UNIVERSITÄT
DUISBURG
ESSEN

Open-Minded

DOCTORAL THESIS

Interactive Methods for Model-based Collaborative Filtering Recommender Systems

by Benedikt Loepp

Interactive Methods for Model-based Collaborative Filtering Recommender Systems

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Informatik und Angewandte Kognitionswissenschaft
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation
von

Benedikt Loepp
aus
Dortmund

1. Gutachter: Prof. Dr.-Ing. Jürgen Ziegler
2. Gutachter: Assoc. Prof. Dr. Ir. Martijn C. Willemsen
Tag der mündlichen Prüfung: 02. März 2021

Abstract

Recommender systems have become very popular for reducing the information overload users are often confronted with in today's web. Collaborative filtering is the method of choice for generating personalized recommendations, supporting users in finding items that best match their preferences, from news articles and movies to all kinds of consumer goods and services. Model-based techniques have achieved great success in terms of recommendation accuracy and algorithmic performance. While there is a large body of research on these aspects, only little effort has been spent on improving user control and experience. As a consequence, users of contemporary systems usually have no other option than rating single items to indicate their preferences and thus to influence the recommendations. In this thesis, we propose a set of *interactive methods for model-based collaborative filtering recommender systems*. With these methods, we aim at providing users richer possibilities to specify their preferences and to control the outcome of the systems according to situational needs. In general, users should be enabled to take a more active role throughout the process of finding suitable items. Guided by a structured model of user interaction, we first present a *choice-based preference elicitation* method. For systems that rely on matrix factorization, one of the most commonly applied techniques in the area of model-based collaborative filtering, this method provides an alternative to rating items in cold-start situations. Furthermore, we describe an algorithmic enhancement, *content-boosted matrix factorization*. Based on the additional item-related information that is considered by this method, we give several examples of advanced *interactive features* that allow users to control the recommendations in an even more expressive manner, also later in the process. Finally, we propose a concept called *blended recommending*. This concept is designed to merge model-based collaborative filtering with other established methods in a way that users can be supported also in complex scenarios with the full range of options they need to reach their search goal. All these methodological contributions are complemented by *empirical evaluations*. Overall, we conducted four user experiments with $n = 35$, 46, 54 and 33 participants, respectively. The results underline that our methods can effectively be implemented in existing recommender systems in order to turn them into fully interactive, user-controlled applications. This is finally confirmed with the help of an *integrated recommendation platform* that demonstrates that all our developments can be combined with each other in a single holistic system.

Keywords Recommender systems, Collaborative filtering, Interactive recommending, Matrix factorization, Empirical studies, User experience, User interfaces.

Acknowledgements

I would like to thank all people who directly or indirectly helped me throughout the process of writing this thesis, but also in the time leading up to this journey.

First, I am very grateful to my advisor, Jürgen Ziegler. Also, I would like to thank my second reviewer, Martijn C. Willemsen. Moreover, I wish to express my gratitude to all my (former) colleagues in the Interactive Systems research group at the University of Duisburg-Essen, in particular, Tim Hussein, who supervised my Master thesis and introduced me to recommender systems, as well as Tim Donkers, Timm Kleemann and Johannes Kunkel, who were supervised by myself and contributed to my research with their student theses.

Also, thanks to my parents, Marianne and Helmut, my sisters, Daniela and Melanie, as well as my friends, especially those who studied with me, Dennis and Sebastian. I would have never made it through this journey without you. Finally, I would like to thank my girlfriend, Melanie, for all her support and understanding.

*— **Benedikt Loepp, December 2020***

Contents

CHAPTER 1

> "Information is a source of learning. But unless it is organized, processed, and available to the right people in a format for decision making, it is a burden, not a benefit."
>
> — *William Pollard, American businessman*

Introduction

Information can be a burden. The above quote captures well the problem that users all too often suffer from in today's web, where they are constantly confronted with tremendous amounts of information. Even on a single platform, users have to decide between a sheer endless number of alternatives, no matter for which task—be it reading the news, watching a movie, or buying a commercial product. Without support, the resulting information overload imposes a cognitive demand that makes it very difficult for users to find the right information in a reasonable time. *Recommender systems* can help in such situations: These systems organize and process the overwhelming amount of information in a way that eventually allows to make the right piece of information available to the right people at the right time, i.e. the most recent news article relevant to the topic the user is interested in, the most enjoyable movie given his or her specific interests, or the product that best matches the current needs. With the support of recommender systems, effective decision making can thus be significantly facilitated, and the availability of all the different options turns from a burden into a benefit.

Around 1990, the idea for the method underlying most of these systems, *collaborative filtering*, was introduced in *Tapestry* [Gol*92], which is often considered the first recommender system. One of the reasons for the success of this method is the fact that the requirements with respect to data availability are rather low: Collaborative filtering leverages the "wisdom of the crowd" to identify *items* that are potentially of personal interest to the current user, from news articles and movies to all kinds of consumer goods and services. From an information provider's perspective, this constitutes a major advantage as only *feedback* the community of users provided for these items—explicitly expressed via ratings or implicitly acquired from observed and logged interaction behavior—is required as input data for the algorithms [JWK14; JJ17]. Beyond that, collaborative filtering is known for its high efficiency. At the same time, the generated recommendations appear very accurate when evaluated in offline experiments [SK09; ERK11]. From a user's perspective, the recommendations can also be considered fitting sufficiently well to the user-item feedback previously collected by the system. Accordingly, collaborative filtering still is the most popular personalized recommendation method [RRS15a].

Existing approaches can be divided into *memory-based* and *model-based* collaborative filtering [BHK98; AT05]. Due to lower memory requirements, less computational efforts, and usually higher recommendation quality, model-based techniques dominate today both in academia [KB15a] and industry [AB15]. For creating the models, it is inevitable to handle incomplete

datasets as they are common for recommendation scenarios, in which each user typically provides feedback only for a small number of items compared to the size of the whole item catalog. *Matrix factorization* is one of the techniques of choice that can be applied exclusively based on these sparse user-item interaction data [KBV09]. Many algorithmic improvements to this and other techniques have been proposed in recent years [KB15a]. However, these improvements primarily address the issue of increasing recommendation accuracy, i.e. the performance measured by objective metrics in retrospective offline experiments [GS15]. This thesis, in contrast, goes beyond the large body of research on algorithmic issues by focusing on *user experience*. Concretely, we propose *interactive methods for model-based collaborative filtering recommender systems* that provide users novel, more advanced possibilities for specifying their preferences, controlling the outcome of the systems according to situational needs, and, overall, taking a more active role throughout the entire process of finding suitable items.

In the following, we formulate the underlying *problems* of state-of-the-art recommender systems in more detail. Next, we derive *goal and objectives* for this thesis, and list the resulting *contributions* as well as *related publications* afterwards. At the end of this chapter, we give an *overview of the structure* of the remainder of this thesis.

1.1 Problem formulation

Especially due to the large research effort during the *Netflix* prize competition in the 2000s [BL07; FHK12], model-based collaborative filtering can be considered quite mature today in terms of recommendation accuracy. This means, applying techniques such as matrix factorization allows to calculate very precisely which items should be recommended to a user, either by predicting ratings the user would likely give or by computing a ranking among the items [KB15a]. However, it has been shown that this does not necessarily lead to a commensurate level of user satisfaction [XB07; KR12; PCH12]. The small incremental improvements that still seem possible are thus unlikely to be exceptionally beneficial for users. Moreover, the question has recently been raised—in context of deep learning algorithms, which in the last few years became increasingly popular also in the area of recommender systems [Zha*19]—whether these improvements progress the field at all [DCJ19]. Consequently, other evaluation metrics such as *coverage*, *diversity*, *novelty* and *serendipity* have been discussed for assessing the quality of the systems' outcome. These metrics are more important with respect to user experience, with much greater potential for improvement when optimizing against them [GDJ10; VC11; CHV15].

Beyond that, another important aspect that is strongly related to the user experience of recommender systems, and may contribute significantly to actual user satisfaction, is the degree of *control* users have over the recommendations [KR12; JJ17; Alv*19]. Nevertheless, the ways to influence the generation of recommendations in contemporary implementations such as the ones of *Amazon* [LSY03; SL17] or *Netflix* [BL07; GH15] are mostly extremely limited: The only means for users to actively affect the results in these highly automated systems is by providing explicit feedback regarding the relevance of single items, i.e. indicating preferences by rating or re-rating them [JWK14]. However, users often prefer comparing items instead of rating them [JBB11; Ngu*13], and are faster in doing so [Car*08; JBB11]. Preferences expressed as ratings tend to be noisy, inaccurate and unstable [Cos*03; Ama*09; APO09; JBB11]. Moreover, the risk of being stuck in a "filter bubble" [Par11] increases as the recommendations are more and more

constrained to items similar to those the current user has evaluated positively in the past. This well-known effect makes it difficult to become aware of hidden alternatives, explore new and diverse areas of potential interest, and adapt the results to situational needs [Par11; NV14]. Overall, it can thus be said that possibilities to improve control are highly valued, especially when they help users in achieving their search goals. At the same time, such possibilities are increasingly expected by data regulation policies such as the *GDPR* [Har*19].

Another problem related to user experience can be seen in the general lack of *transparency* of many contemporary recommender systems, particularly of those relying on model-based collaborative filtering [XB07; PCH12]. The most prevalently used techniques automatically infer abstract models with latent dimensions from the original input data [KBV09; KB15a], making it difficult for users to understand the profiles that are learned to represent their preferences, and consequently, why certain items are recommended. This, in turn, may reduce trust in the systems as well as acceptance of the automatically personalized results [XB07; TM15]. In addition, the fact that model-based systems often act as "black boxes"—a problem that gets even worse with the ever-increasing complexity of the algorithms, currently reaching its peak with the rise of deep learning [Zha*19]—hampers the use of methods for explaining these results [TM15; Rud19]. The attempts to facilitate the users' understanding range from simple textual components such as the well-known "other customers also bought …" explanations by *Amazon* [LSY03; SL17], over social explanations that show what is preferred by friends in social networks [SC13], to complex visualizations of the entire item space, which highlight areas of recommended items and raise awareness of alternatives [Gan*09; KLZ17].

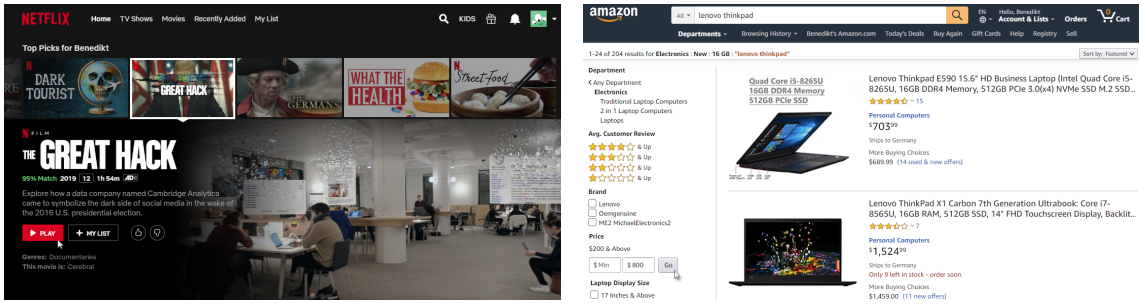

Figure 1.1 Screenshots of *Netflix* and *Amazon*, showing examples of automatically generated recommendations (left) and explicitly filtered content (right).

Information retrieval systems, where users need to enter a query before receiving results from an underlying database that match the specified terms [BR99], as well as related *search and filtering* mechanisms [Hea09; ST09; Dir12; Wei*13], constitute alternatives for users to find relevant content. While less prominently available on platforms such as *Netflix* or *Spotify*, where the content is specifically tailored for each user and served in a highly automated manner, without an expert search or advanced browsing facilities, the picture is different on many e-commerce websites: Here, features of this kind are implemented more frequently, which makes the underlying systems effectively more controllable. Though *Amazon* is also a prominent example of the application of recommendation functionalities, this contrast is well illustrated by the examples shown in Figure 1.1. Beyond that, these systems only perform the actions requested by the user. Whereas this is clearly necessary due to the lack of system-initiated *personalization*, users can

thus also trace system behavior more easily, which usually leads to a better understanding of the results. On the other hand, the lower degree of automation increases the *interaction effort* on part of the users. Accordingly, it is expected that users are aware of their search goal, at best already at the very beginning, and know how to reach this goal using the available options [Kuh91; WKB05; Mic*07]. Yet, as their information need typically changes dynamically as long as new information is picked up [Bat89], it is essential that users can intervene in the search process until the "anomaly in the user's state of knowledge" [BOB82] is finally resolved, i.e. the gap closes between what the user knows about a problem and what he or she needs to solve it.

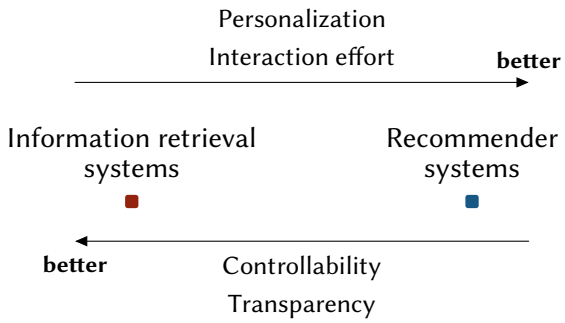

Figure 1.2 Comparison of information retrieval and recommender systems.

Figure 1.2 summarizes the problems of contemporary systems: Recommenders have deficiencies in terms of *controllability* and *transparency*. However, users do not need to know their search goal in order to be provided with adequate results due to the usually high degree of *personalization*. Consequently, the *interaction effort* that needs to be spent for arriving at these results is relatively low. Information retrieval systems are highly controllable and act in a transparent manner. But, data requirements are much higher from an information provider's perspective [WS12]. Also, it is more cumbersome for users to reach their goal due to the lack of personalized results. Hence, proactively taking users by the hand is often demanded in information retrieval research [WKB05]. Thus far, search and filtering mechanisms have, however, primarily been developed and studied independently of recommendation functionalities—despite numerous calls for integrating them more closely with each other [GKP11; KR12; HPV16; JJ17].

Similarly, adding more interactivity to the systems and rendering them more comprehensible to users are increasingly recognized as important goals in recommender research [XB07; KR12; PCH12; KW15; JJ17; Alv*19]. Yet, such aspects related to user experience have only recently begun to attract wider attention [KR12; Kni*12]: *Interactive recommenders* have been proposed that aim at preserving the benefits of automated systems while closing the gap to systems that employ more flexible and controllable methods. Early approaches guide users by asking series of questions [MR09], visualize similarities to other users [Gre*10], or use comparisons instead of ratings for eliciting user preferences [JBB11]. Richer interaction mechanisms allow critiquing recommended items by leveraging predefined metadata [VFP06] or tags [VSR12]. In the former case, explicitly defined catalog descriptions of item content or lists of product attributes are required, but, in the latter case, users themselves generate the necessary information.

Using item-related information in the form of *user-generated data* has the advantage of relying on concepts that appear inherently meaningful to the entire user community. Accordingly, eliciting

preferences via tags has been shown to bear the potential for improving user control and comprehension [Gre*09; SVR09]. However, also most of the interactive recommending approaches have been implemented in an isolated manner, i.e. not integrated with other datasources, preference elicitation techniques, information filtering or recommendation methods. In line with that, systems that use tags are largely independent of established collaborative filtering techniques [e.g. Gre*09; SVR09; Gua*10; VSR12]. For this reason, these systems cannot benefit from existing profiles, which, based on previously collected user-item feedback data, usually reflect the long-term preferences of users and thus allow to personalize the results. The same applies to many of the other, often way more complex, but also more powerful approaches that aim at increasing interactivity [e.g. CP12a; BOH12; PBT14; SSV16; APO16; Car*19].

In summary, the advantages of model-based collaborative filtering techniques are only rarely exploited in interactive recommending research. On the other hand, the models learned by these techniques, such as latent factor models that result from the application of matrix factorization algorithms, are seldom applied for purposes other than improving recommendation accuracy or algorithmic performance. Therefore, there is a need for methods in the middle of the continuum depicted in Figure 1.2. These methods should retain the benefits of modern collaborative filtering models in terms of personalization and efficiency, but, at the same time, allow for the introduction of advanced mechanisms for increasing interactive control and system transparency. They should be available throughout the recommendation process, whenever this is useful for the individual user—holistically integrated with each other, but also with search and filtering mechanisms as they are known from (commercial) real-world systems.

1.2 Goal and objectives

In order to address the aforementioned shortcomings, the overall *goal* of this thesis is to propose *interactive methods* for enhancing contemporary model-based collaborative filtering systems: Users should be enabled to express their preferences more easily and to effectively control the recommendations at all times. The range of mechanisms for making the systems more responsive to situational needs should be broadened and user experience be improved.

For pursuing this goal, we define the following *objectives*, also summarized in Figure 1.3:

- Develop a *model* that illustrates how recommendations can be influenced by the user and helps in structuring methods that may be integrated into or with model-based collaborative filtering systems in order to interact with them.
- Present concrete *methods* in accordance with this model that allow for the implementation of model-based collaborative recommender systems with more advanced interaction mechanisms than currently available.
- Validate the effectiveness of these methods and explore their value in terms of user control and experience by means of empirical *evaluations*.
- Develop an integrated recommendation *platform* under consideration of the evaluation insights for demonstrating the combined potential of these methods.

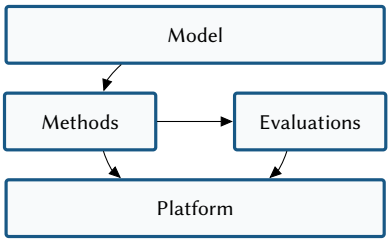

Figure 1.3 Overview of the objectives.

1.3 Contributions and related publications

In this thesis, we directly address these objectives, going beyond prior research that aimed at improving model-based collaborative filtering systems mainly in terms of recommendation accuracy and algorithmic performance. Instead, we contribute to the state of the art by proposing interactive methods that can effectively be implemented in existing recommender systems in order to turn them into fully interactive, user-controlled applications. For this, we explore the following *research questions* based on ideas that build on one another:

RQ1: How to *exploit the semantics* in latent factor models for improving user control and experience?

RQ2: How to *leverage item-related information* in addition to standard collaborative filtering feedback data for this purpose?

RQ3: How to *merge model-based collaborative filtering* with other recommendation and information filtering methods for this purpose?

Addressing these research questions, this thesis contains the following *main contributions* (in addition to the aforementioned model for structuring the underlying suggestions towards more interactive recommender systems from a theoretically informed perspective). Originally proposed in the *publications* that are listed accordingly, these are:

Choice-based preference elicitation A *method* that lets users express their preferences in collaborative filtering systems without requiring them to rate items: Under exploitation of the semantics in the dimensions of a typical latent factor space as spanned by a model-based technique that relies on the application of matrix factorization, users can indicate their preferences in a dialog by choosing between sets of representative sample items.

- **Benedikt Loepp**, Tim Hussein, and Jürgen Ziegler. "Interaktive Empfehlungsgenerierung mit Hilfe latenter Produktfaktoren." In: *Mensch & Computer 2013 – Tagungsband*. München, Germany: Oldenbourg Verlag, 2013, pp. 17–26.
- **Benedikt Loepp**, Tim Hussein, and Jürgen Ziegler. "Choice-Based Preference Elicitation for Collaborative Filtering Recommender Systems." In: *CHI '14: Proceedings of the 32nd ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 3085–3094. **Honorable mention award**.

Content-boosted matrix factorization An algorithmic enhancement to standard matrix factorization: The *method* allows to integrate a latent factor model with any type of content data as side information, this way opening up a variety of options for the implementation of interactive features. Based on user-generated tags as a running example, a *framework* called *TagMF* implements this method. An extensive *offline evaluation* validates its effectiveness.

- Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Merging Latent Factors and Tags to Increase Interactive Control of Recommendations." In: *RecSys '15: Poster Proceedings of the 9th ACM Conference on Recommender Systems*. 2015.
- Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Tag-Enhanced Collaborative Filtering for Increasing Transparency and Interactive Control." In: *UMAP '16: Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2016, pp. 169–173.
- Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Towards Understanding Latent Factors and User Profiles by Enhancing Matrix Factorization with Tags." In: *RecSys '16: Poster Proceedings of the 10th ACM Conference on Recommender Systems*. 2016.
- **Benedikt Loepp**, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Interactive Recommending with Tag-Enhanced Matrix Factorization (TagMF)." in: *International Journal of Human-Computer Studies* 121 (2019), pp. 21–41.

Interactive features based on content-boosted matrix factorization Several *application possibilities* of the extended matrix factorization method for implementing interactive collaborative filtering systems: Leveraging the additional item-related information, users are provided with advanced interaction mechanisms allowing them to indirectly determine their position in the latent factor space. Based on concepts that are inherently meaningful, they can thus steer the recommendations into the direction appropriate to their current situation.

- **Benedikt Loepp**, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Interactive Recommending with Tag-Enhanced Matrix Factorization (TagMF)." in: *International Journal of Human-Computer Studies* 121 (2019), pp. 21–41.

Blended recommending A *concept* for merging model-based collaborative filtering both with other recommendation methods and information filtering methods in a fully user-controlled fashion: Users can thus be supported even in complex scenarios with the full range of options necessary to reach their search goal. The hybrid combination preserves the benefits of the individual methods and leaves room for the consideration of the previously described direct extensions to collaborative filtering systems.

- Katja Herrmanny, Sandra Schering, Ralf Berger, **Benedikt Loepp**, Timo Günter, Tim Hussein, and Jürgen Ziegler. "MyMovieMixer: Ein hybrider Recommender mit visuellem Bedienkonzept." In: *Mensch & Computer 2014 – Tagungsband*. Berlin, Germany: De Gruyter Oldenbourg, 2014, pp. 45–54.
- **Benedikt Loepp**, Katja Herrmanny, and Jürgen Ziegler. "Blended Recommending: Integrating Interactive Information Filtering and Algorithmic Recommender Techniques." In: *CHI '15: Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 975–984.

- **Benedikt Loepp**, Katja Herrmanny, and Jürgen Ziegler. "Merging Interactive Information Filtering and Recommender Algorithms – Model and Concept Demonstrator." In: *i-com – Journal of Interactive Media* 14.1 (2015), pp. 5–17.

Empirical evaluations Four extensive *user experiments* (with $n = 35$, 46, 54 and 33 participants, respectively) for exploring the effectiveness of the aforementioned developments and their effects on user experience: Among others, the results confirm for the first time that considering side information in model-based collaborative filtering systems is also beneficial from a user perspective, which previously has only been shown offline. Furthermore, and more importantly, the findings illustrate the potential of our methods for providing users with more interactive options than rating items to control the recommendations.

- **Benedikt Loepp**, Tim Hussein, and Jürgen Ziegler. "Choice-Based Preference Elicitation for Collaborative Filtering Recommender Systems." In: *CHI '14: Proceedings of the 32nd ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 3085–3094. **Honorable mention award**.
- **Benedikt Loepp**, Katja Herrmanny, and Jürgen Ziegler. "Merging Interactive Information Filtering and Recommender Algorithms – Model and Concept Demonstrator." In: *i-com – Journal of Interactive Media* 14.1 (2015), pp. 5–17.
- **Benedikt Loepp**, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Interactive Recommending with Tag-Enhanced Matrix Factorization (TagMF)." in: *International Journal of Human-Computer Studies* 121 (2019), pp. 21–41.

Integrated recommendation platform A *platform* that combines all the developments in a single system based on a set of seamlessly connected perspectives, and additionally provides access to other recommendation methods as well as search and filtering mechanisms: Several *case studies* with this platform demonstrate that the approaches to interactive recommending we propose in this thesis, taken together, can significantly contribute to making today's largely automated model-based collaborative filtering recommender systems more controllable by users.

- **Benedikt Loepp** and Jürgen Ziegler. "Towards Interactive Recommending in Model-Based Collaborative Filtering Systems." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 546–547.

Several times throughout this thesis, we will touch upon the aspect of *transparency* of recommender systems. However, this aspect is only partly within the scope of this work: While we found several indications of increased transparency, for instance, due to the usage of side information in collaborative filtering algorithms, our focus is on the contributions as stated above. We pursue the goal of providing users richer mechanisms to control the systems, rather than opening up the underlying black-box models or explaining their outcome. We briefly address these issues in context of our content-boosted matrix factorization method and its application possibilities, but only as far as the improvements go hand in hand with higher controllability.[1]

[1] In related work, we directly approached the aspect of transparency several times, even in context of latent factor models as learned by matrix factorization algorithms. For instance, we exploited these models for visualization purposes [KLZ17] or studied their semantics with the help of specifically designed online games [KLZ18a; KLZ18b; Kun*19b]. However, the author of this thesis contributed less substantially to these publications.

Finally, equally outside the scope of this thesis, some other publications are worth mentioning. These publications not only played a role in shaping the research interests of the author of this thesis, but also influenced the presented research to some extent. For instance:

- **Benedikt Loepp**, Catalin-Mihai Barbu, and Jürgen Ziegler. "Interactive Recommending: Framework, State of Research and Future Challenges." In: *EnCHIReS '16: Proceedings of the 1st Workshop on Engineering Computer-Human Interaction in Recommender Systems*. 2016, pp. 3–13.
- Johannes Kunkel, **Benedikt Loepp**, and Jürgen Ziegler. "A 3D Item Space Visualization for Presenting and Manipulating User Preferences in Collaborative Filtering." In: *IUI '17: Proceedings of the 22nd International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2017, pp. 3–15.[2]
- Jan Feuerbach, **Benedikt Loepp**, Catalin-Mihai Barbu, and Jürgen Ziegler. "Enhancing an Interactive Recommendation System with Review-based Information Filtering." In: *IntRS '17: Proceedings of the 4th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*. 2017, pp. 2–9.[2]
- Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Sequential User-Based Recurrent Neural Network Recommendations." In: *RecSys '17: Proceedings of the 11th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2017, pp. 152–160.[2]
- Johannes Kunkel, **Benedikt Loepp**, and Jürgen Ziegler. "Ein Online-Spiel zur Benennung latenter Faktoren in Empfehlungssystemen." In: *Mensch & Computer 2018 – Tagungsband*. Gesellschaft für Informatik, 2018.[2]
- **Benedikt Loepp**, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Impact of Item Consumption on Assessment of Recommendations in User Studies." In: *RecSys '18: Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2018, pp. 49–53. **Best short paper award**.

For a complete publication list, please see the bibliography at the end of this thesis.

1.4 Outline

The remainder of this thesis is organized in chapters as follows:

- Chapter 2 discusses the *state of the art* in the field of recommender systems, focusing on model-based collaborative filtering, algorithmic advances in this area, and existing attempts to increase interactivity.
- Chapter 3 is an *advance organizer* that summarizes the problems and outlines solutions based on the literature review and a model of user interaction. It explains where our research questions stem from and how they are reflected in the succeeding chapters.
- Chapter 4 describes *choice-based preference elicitation* as an alternative to rating items, including the empirical evaluation of this first attempt to use the semantics contained in matrix factorization models for practical user-oriented purposes.

[2] Note that in contrast to the publications that are directly related to the contributions of this thesis (see Section 1.3), the author of this thesis contributed less substantially to this work.

- Chapter 5 treats the succeeding step of *boosting matrix factorization with content information*. This includes the methodological background, a framework for implementing the method, and the results of an extensive offline evaluation.
- Chapter 6 builds on the extended matrix factorization method by presenting a number of *interactive features* based on the considered item-related side information. It also describes the empirical evaluation of these application possibilities.
- Chapter 7 covers the concept we call *blended recommending*, which merges established recommendation and information filtering methods in a hybrid and at the same time user-controlled fashion, as well as the corresponding empirical evaluation.
- Chapter 8 illustrates by means of the *integrated recommendation platform* and several descriptive case studies that all the proposed developments can successfully be combined with each other to come up with a fully interactive, user-controlled system.
- Chapter 9 *concludes* this thesis and provides an *outlook* on future research.

Figure 1.4 repeats the overview of the objectives from Figure 1.3. In addition, it reflects the above structure and indicates which publications served as the main basis for the respective chapters.

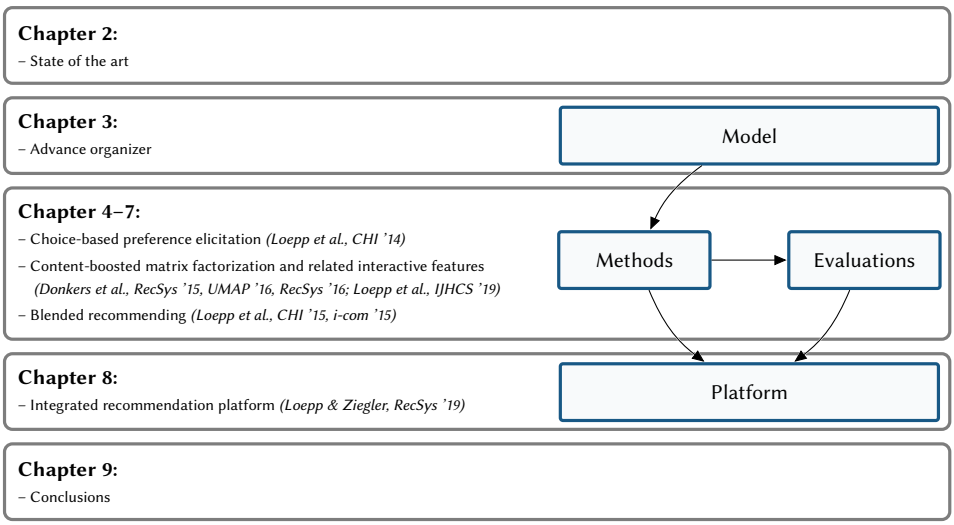

Figure 1.4 Structure of the thesis, including objectives and major publications.

CHAPTER 2

"Research is formalized curiosity.
It is poking and prying with a purpose."

— *Zora Neale Hurston, African-American writer*

State of the art

Recommender systems have become ubiquitous tools that today support users in almost all situations in which they need to find suitable items. They are often defined as "systems that select and present items from confusingly large sets of alternatives, such as news articles, movies, or basically any kind of consumer goods and services, that best match the user's preferences" [cf. SK09; RRS15b]. Research has long been focused on optimizing accuracy and performance of the algorithms that perform this type of recommendation task, usually in a largely automated manner. However, for maximizing user satisfaction, not only automatically addressing individual preferences plays a decisive role, but also contextual information such as the user's location or the goal in the current situation. Yet, only in recent years, a more holistic view has emerged. In line with that, user interaction with the systems and general user experience are increasingly considered as important factors for the success of recommender systems—in addition to algorithms and background data [cf. KR12; KW15; JJ17; Alv*19].

In this chapter, we start by giving a *broad overview* of the research field of recommender systems, in particular, the manifold methods for generating recommendations. Next, due to its relevance for this thesis, we lay our focus on *matrix factorization*, one of the most commonly applied techniques for this purpose: We elaborate on the basic algorithm but also the advances recently made. Finally, we discuss *interactive methods*, including conventional preference elicitation and related work conducted in the area of search and information filtering. More importantly, this discussion also includes recommender research that addresses the above factors, i.e. approaches that put users in control and improve user experience. We structure this review around a model for interactive recommending we have previously proposed [LBZ16].

2.1 Overview of recommendation methods

Automated recommender systems as they are widely used today, for instance, on *Amazon* [LSY03; SL17] or *Netflix* [GH15], generally rely on one of two principles: *personalized* or *non-personalized* recommending. In the first case, recommendations are tailored specifically for each user, taking his or her preferences, individual needs and goals into account (as in the two examples just mentioned). In the other case, all users receive the same recommendations. For example, general popularity determines whether an item is recommended. Under certain circumstances, this non-personalized behavior already produces results of sufficient quality [AB15]. Yet, personalized

approaches are usually more effective, and thus play a much larger role both in academia and industry—although the methods have stronger prerequisites, are more complex, and require an individual representation of each user, often in an underlying model.

Recommendation model Before going into detail, we explain the underlying principle of the generation of recommendations, which is almost always the same, independent of the specific method. The model we present in Figure 2.1 provides an overview: The ■ *user* interacts with a ■ *user interface*, for instance, an online shop or a movie platform. The interaction performed by the user represents so-called *user-item feedback*, explicit ratings provided for the items or implicit actions such as time spent inspecting item details. This feedback serves as input for the ■ *recommender*. In case of personalized recommendations, the algorithm of the recommender (or multiple algorithms, if several methods are combined in a hybrid fashion) uses this input to generate ■ *recommendations* with the help of information about the items from the ■ *item database*. Previously stored information about the user may additionally be taken into account, i.e. his or her representation within the ■ *user model*. This model is responsible for storing the representations of all users of the system and may be updated as new feedback comes in. Information from a ■ *context model* may be used as well. Either way, recommendations are shown in the interface. In few cases, the interactions performed by the user at this stage are again considered, for example, to obtain feedback regarding the relevance of recommended items. Later in this chapter, we discuss attempts for making recommender systems more interactive, diving deeply into this connection (see Section 2.3.2). In most contemporary systems, this kind of relevance feedback is though the only way for users to actually make interventions once the recommendation process has started, if at all.

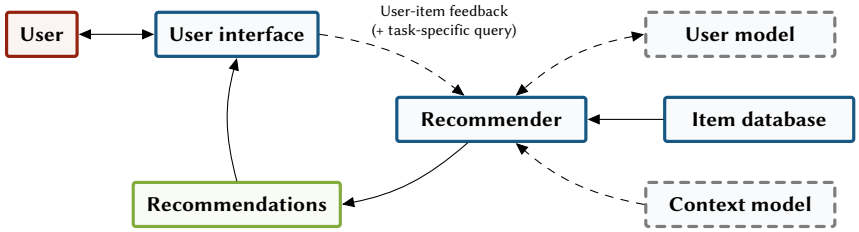

Figure 2.1 Model presenting an overview of the generation of recommendations. Dashed lines denote optional relations or components.

Recommendation function Formally,[3] the output of the algorithm of a recommender is often defined by a function that estimates the preference of a user u from the set U of all users of the system, for an item i from the set of all items I that the system may recommend [RRS15b]:

$$s(i|u) = \hat{r}_{ui} \in \mathbb{R}\,. \tag{2.1}$$

The value of this function is typically calculated based on item feedback r_{ui} provided by the current user, and in some methods, also by other users. As illustrated in Figure 2.1, it may also depend on information about the user's context or task. Then, a parameter x for contextual information or h for a task-specific query is added to the function shown in (2.1). Either way,

[3] In this thesis, we, wherever possible, follow the suggestions by Ekstrand and Konstan [EK19] for notation purposes.

the result is a score $\hat{r}_{ui}$ representing the estimated preference. Accordingly, this function can be used to select the top n items as recommendations ($n \ll |I|$), i.e. to present a sorted list of items to which the highest scores have been assigned. In this way, the system automatically reduces the set of available items to those of the highest utility for the current user.

In the literature, the task of recommender systems is often defined *exclusively* in this way [AT05]. But, as already outlined, this common point of view stops short of taking aspects relevant from a human-computer interaction perspective into account. Next, before addressing these aspects, we provide an overview of the different ways a function as depicted above can actually be implemented, i.e. of different *recommendation methods*.

2.1.1 Collaborative filtering

Collaborative filtering is the most popular method for personalized recommendations. The underlying idea is to exploit the "wisdom of the crowd", i.e. feedback provided for the items by the entire user community. It is assumed that feedback of users who in the past expressed a taste similar to the current user, constitutes a valid means for the prediction of which items this user might prefer in the future [DK10; ERK11; KB15a]. Conventional notation is to represent the existing feedback r_{ui} by a *user-item matrix* $\mathbf{R} \in \mathbb{R}^{|U| \times |I|}$, where rows correspond to users and columns to items [RRS15b; KB15a; EK19]. Table 2.1 shows a toy example based on movies.

Table 2.1 Example of a typical user-item matrix.

	Donnie Darko	Twilight	Hangover	The Dark Knight	Bridget Jones	Braveheart	Bad Boys	Mr. & Mrs. Smith
Amalia	1		4		5			3
Benjamin	1	2	5		4		5	
Charlotte		3	4	1	5	2		
Daniel	1	2		4			5	4
Emily		3	2	2		4	2	
Freddie			4	3		4	5	

The type of this historical user-item feedback is defined by the use case of the recommender system: The cells of $\mathbf{R}$ may contain ratings from the $[1, 5]$ interval as in the example above, from other intervals, or just $\{0, 1\}$ values in case of implicit feedback (see Section 2.3.1 for more details). Either way, the non-empty cells may also be represented by the set R, the ratings of a user u by user vector $\vec{r}_u$, and the ratings for an item i by item vector $\vec{r}_i$. However, each user typically provides feedback only for a small number of items, and each item only receives feedback from a limited number of users. Thus, $\mathbf{R}$ is mostly sparse, often with a very large percentage of empty cells. For instance, sparsity was 98.8 % in case of the dataset released for the *Netflix* prize, the famous competition in the late 2000s that was concerned with improving the accuracy of the movie platform's recommendation algorithm [BL07].

According to these definitions, the task of a collaborative filtering recommender can be subsumed as predicting the values of the empty cells of $\mathbf{R}$, i.e. to determine how users would rate items they have not yet rated. Hence, $\hat{r}_{ui}$ from (2.1) estimates the missing item feedback of the current user.

For this task, multiple techniques exist (see below). The fact that all these techniques exclusively use the feedback provided by other users for the respective items underlines the main advantage of collaborative filtering: System providers do not have to ensure that predefined metadata or expensive expert knowledge is available. Yet, this goes hand in hand with the *cold-start problem*, a difficulty all collaborative filtering systems need to deal with: In situations in which there is no feedback available, for new users or new items, the corresponding rows or columns of the user-item matrix do not contain any entries. Thus, the algorithm cannot generate recommendations for these users or consider these items as suggestions, respectively. Concerning the user side, this problem has mostly been addressed in an algorithmic manner, for instance, by posing initial interview questions or applying active learning techniques and bandit algorithms [e.g. ZYZ11; Kar*12; ZZW13; Rub*15; ERR16; CHR16]. However, even if feedback exists, this is often too sparse, especially when the number of users in the system is (still) small. In addition, users often view only a fraction of all items in ongoing system use, and rate or purchase much less. Thus, many cells of $\mathbf{R}$ remain empty, which makes it even more difficult to produce accurate recommendations.

Since collaborative filtering constitutes the underlying principle of *matrix factorization*, the technique that serves as a basis for the developments we present in this thesis, we now illustrate the two different types of this method in a bit more detail.

2.1.1.1 Memory-based techniques

Memory-based (or neighborhood-based) collaborative filtering techniques first determine *similarities between users or items*, i.e. between rows or columns of $\mathbf{R}$, to then come up with predictions [Sch*07; DK10]. In this way, these techniques very directly reflect the underlying principle of exploiting the "wisdom of the crowd".

User-based collaborative filtering When similarities are calculated between users, the first step is to determine a *neighborhood* $N(u)$ of users who are similar to the active user u in terms of their item feedback. For this, various similarity metrics may be translated into a function $\mathrm{sim}(u, v)$ [cf. ERK11]. Based on this function, the most similar users are chosen as *mentors* [Sch*07]. Next, for estimating $\hat{r}_{ui}$, the average is taken of their feedback for item i, an item the current user u has not yet provided feedback for. This leads to a *recommendation function* defined as follows, where the mentors' feedback is additionally weighted proportionally to similarity:

$$\mathrm{s}(i|u) := \frac{\sum_{v \in N(u)} w_{uv} \cdot r_{vi}}{\sum_{v \in N(u)} |w_{uv}|} \quad \text{with } w_{uv} := \mathrm{sim}(u, v)\,. \tag{2.2}$$

Item-based collaborative filtering Alternatively, it is possible to calculate similarities between items. Since column vectors are much more stable (they contain relatively more entries than a single user can ever provide feedback), this part of the process can be moved into an *offline phase*, without any side effects but significantly improving performance. Again using a weighted arithmetic mean, a score $\hat{r}_{ui}$ for item i can then be predicted based on the current user u's feedback for other items and their similarities to i. Thus, items can be promoted similar to those which already received positive feedback by the active user or which are currently shown in

the user interface. This is widely known from *Amazon*'s "customers who bought this item also bought …" product recommendations [LSY03; SL17].

Apparently, calculating which items might be of interest for the current user is a rather simple task. This ensures applicability in diverse domains and that the algorithms are adoptable without much effort for many use cases. However, recommendation quality heavily *depends on data availability*. For example, in new user cold-start scenarios or when the current user only has provided feedback for a small number of items, the user's neighborhood cannot be adequately determined. Then, few similar users exist, who can hardly be considered appropriate mentors. Beyond that, storing the complete user-item matrix is *memory intensive*, and performing all the above calculations directly on this matrix at runtime *computationally expensive*.

2.1.1.2 Model-based techniques

As a consequence, several attempts have been made to circumvent at least some of these problems, for instance, performance issues by switching to an item-based variant. However, model-based techniques have become more prominent in recent years, being superior in terms of algorithmic performance and scalability: Only at the beginning, access to the high-dimensional user-item matrix is required. Based on empirical observations and the resulting assumption that there exist hidden patterns in the item feedback provided by users due to similarities in taste and behavior, a *statistical model* is then derived from this matrix, significantly reduced in the number of dimensions. While this memory-consuming step, which also requires lot of computational effort, can be performed separately offline, only the model is necessary later in the process when recommendations need to be presented to the user. This significantly speeds up their generation at runtime [KB15a]. Beyond that, many algorithms can effectively be parallelized, allowing to additionally distribute the computational load of the calculations [Zho*08].

In addition to these advantages, the *Netflix* prize competition constituted a major step forward with respect to recommendation accuracy. Basically, the famous blog post by Funk [Fun06] introduced a way for very efficiently approximating a user-item matrix **R**: Under the premise of still being able to reproduce the original data, a small number of *latent factors* is determined from an overall much larger dataset, similar to factor analysis as known from statistics. However, *matrix factorization* algorithms that originate from this idea allow at the same time to find appropriate values for *missing* data points [KBV09]. As a result, these algorithms became frequently used in the remainder of the challenge. Eventually, the winner team's solution incorporated hundreds of them to achieve the required 10 % gain in prediction error [BKV10; FHK12].

Indeed, other model-based techniques may be applied for the same purpose, ranging from early attempts based on clustering [Sar*02] to recent developments building on the advances made in the area of deep learning [SMH07; Alm*15; WWY15; Hid*16; DLZ17; QCJ18; Fan*19; Zha*19]. To this day, however, matrix factorization is one of the most popular approaches in academia and industry [KB15a; AB15], regardless of the fact that systems based on these algorithms *lack interaction possibilities* that go beyond rating single items. Yet, as we discuss later, this is true for all collaborative filtering systems as well as many other recommendation approaches [JWK14]. Especially compared to memory-based variants, model-based techniques additionally *suffer from low transparency*: It is far more difficult to explain recommendations when they are generated based on black-box models, compared to situations in which a small number of specifically determined mentors with preferences very similar to the current user is responsible for the system's

outcome. In the course of this thesis, however, we will see that for common techniques such as matrix factorization, several ways exist to overcome these issues.

2.1.2 Content-based filtering

Content-based filtering does not take data of the user community into account, but the content of the items themselves, or structured or unstructured descriptions of their properties [Gem*15]. The goal is to find items that are similar in terms of their content to those the current user has already seen, rated or bought. In case of text documents, the procedure is closely related to information retrieval [BR99]: After several preprocessing steps, a term vector is derived for each document, representing how often it contains each of the words found in the entire document set. Then, the similarity to all other documents is determined by means of a vector similarity metric. With more advanced techniques, not only the frequency of terms within each single document is taken into account, but also the frequency in the rest of the document set [BR99]. For more complex types of items and items where the content is not equal to the representation in the system, content-based techniques are applicable as well. The only requirement is that properties are available in a structured format (e.g. predefined metadata or user-generated tags) or the content can be converted (e.g. via extraction from free text).

Beyond these lexical approaches, other techniques work on a semantic level: A *bottom-up* example is *latent semantic indexing* [Dee*90]. Here, the content itself is analyzed to identify latent concepts that describe certain document characteristics. To find these concepts and determine their relevance with respect to individual documents, singular value decomposition can be applied on a term-document matrix. More recently, *top-down* variants gained importance due to the increasing availability of machine-readable ontologies and knowledge databases. Often involving *natural language processing* techniques based on state-of-the-art deep learning methods, such attempts rely on external sources to obtain the knowledge that is necessary to represent (also other types of) items and match them on a semantic level with user interests [Gem*15].

Advantages of content-based filtering are largely complementary to those of collaborative filtering: Since it is always possible to determine products similar to the one the active user currently inspects, user-item feedback is not required, which allows the use in new user cold-start situations. However, *making content information available* may be a task of similar difficulty. For instance, the online radio *Pandora* uses an entirely manual annotation process, involving dozens of specifically trained experts who classify each song according to a large set of properties [Jan*10]. Taking 20–30 minutes per song, this illustrates the huge effort if item characteristics cannot immediately be accessed or at least automatically extracted. On the other hand, the fact that content-based results are naturally easier to explain (“this movie is recommended because it contains action and stars your favorite actor”) is another important advantage. Yet, users often suffer from *recommendations that become increasingly constrained* to items similar to those they rated positively in the past—and would have consumed anyway, even without system support. Pariser [Par11] made this effect, which hinders users in exploring the item space and consuming alternatives, widely known as the “filter bubble problem”.

All these are arguments for *merging* content-based techniques with collaborative filtering in so-called hybrid systems (see Section 2.1.5). Or, taking this even further, directly *integrating* model-based collaborative filtering techniques with content information (see Section 2.2.4). Accordingly, we will take up both suggestions again in the course of this thesis.

2.1.3 Graph-based methods

Depending on the underlying data structure, *graph-based methods* for recommender systems may largely overlap with standard collaborative filtering. Desarkar, Sarkar, and Mitra [DSM10], for instance, propose a memory-based algorithm, but instead of user-item feedback in the form of absolute ratings, employ preference graphs expressing which items a user prefers over others. A step further, Tiroshi, Berkovsky, Kaafar, Vallet, and Kuflik [Tir*14] investigate how tag and friendship relationships can additionally be taken into account to achieve even more accurate recommendations. Beyond that, purely graph-based techniques have been suggested. For instance, *spreading activation* determines the relevance of nodes in a way inspired by biological neural circuits. Such techniques are popular in domains where a graph-based structure is inherently given, such as in context of the semantic web [cf. HN10].

Algorithms known from other use cases have also been adopted: Hotho, Jäschke, Schmitz, and Stumme [Hot*06] build upon the famous *PageRank* search algorithm by *Google* [Pag*99; Fra11]. Their *FolkRank* algorithm recommends tags and other resources by considering the underlying folksonomies as tripartite hyper graphs (user, tag, resource). *Twitter* uses a similar variant for suggestions of whom to follow [Gup*13]. *Social recommenders* [Ben*07; Guy15], sometimes also regarded as knowledge-based systems (see below), directly exploit the graph structure of social networks, including friendship relations or communication paths. Relying on people the current user is familiar with has a number of benefits, for instance, higher trustworthiness by establishing a network of trust [MA07; JE09] or more persuasive explanations by indicating who liked the recommended items [SC13]. However, the superordinate problem of all these approaches remains: *relational data are hardly available* and access often interferes with *privacy concerns*.

2.1.4 Knowledge-based methods

Knowledge-based methods may help if other recommendation methods are no longer sufficient: Especially for expensive products such as cars or houses, which are bought at low frequency, availability of user-item feedback is strongly limited. In complex product domains, criteria might play a role which cannot be taken into consideration due to lack of content information. Then, only deeper domain knowledge can help. Established methods include *case-based* techniques, which calculate similarities between requirements explicitly specified by the user (problem description) and item properties (potential solutions), and *constrained-based* techniques, which use similarity metrics and rules manually defined by the system provider [Jan*10; RRS15b].

Against this background, knowledge-based recommender systems can usually be considered more interactive than others, often being called *conversational*: Users define their requirements stepwise, in an interactive fashion. In turn, they are led in a personalized manner through the space of available options to those that best match these requirements [Bur00]. *Critique-based* systems [VFP06; CP12a] follow this principle very closely. They allow users to critique a recommended item based on its properties. Subsequently, they suggest items that are still similar, but fit better in terms of these properties. The same is true for dialog-based *product advisors*: Only more recently, these tools became more popular, asking users of online shops specific questions regarding their demands to incrementally guide them towards their goal [KZ19]. Preference elicitation thus takes place on a higher level, for example, by asking for which purpose a product is intended. Later in this chapter, we will address these advanced approaches in more depth when discussing interactive recommender systems (cf. Section 2.3.2).

Beyond greater interactivity, another advantage is that users can specify their requirements in a uniquely detailed way. At the same time, the systems can provide semantically rich explanations regarding the reasons why items are recommended. Yet, all this *requires up-front availability of rich information* with respect to the items. This directly depends on the domain knowledge of the system provider or makes external, often expensive expert knowledge inevitable. Besides, immediately adapting to updated user preferences is hardly possible, which in other methods happens automatically as soon as new user-item feedback comes in. Finally, without the guidance of critique-based systems or product advisors, *users need to be aware of their demands*, and also able to verbalize them. Especially when users have only limited domain knowledge or are still at the beginning of the decision process, this can be a problem [Jam*15].

Demographic-based approaches are often seen as special cases of knowledge-based recommenders [Bob*13]. They rely on demographic user profiles and knowledge about which items are most suitable, for instance, for a certain age or a region of origin. While popular in marketing literature, such approaches are less explored in recommender research: They often suffer from the problem that not enough user data are available—or using them appears inadequate or even impossible due to privacy reasons. Moreover, demographics alone may foster stereotypes and are rarely sufficient to accommodate individual interests and, in particular, situational needs.

2.1.5 Hybrid methods

Hybrid methods aim at overcoming the disadvantages of the above techniques while taking advantage of their individual benefits [Bur07]. For instance, content-based suggestions can be provided first, but personalized collaborative filtering recommendations later, once user preferences are known. This kind of hybridization can be performed in several ways: The literature distinguishes between *loosely* and *tightly coupling* [Bur07]. With respect to the first case, many strategies exist for an effective combination of two or more recommendation methods. Implemented separately, only the results of the algorithms are combined, for instance, linearly by averaging the individually predicted scores (weighted), or by interleaving the resulting recommendation lists (mixed). Early examples are the systems by Claypool, Gokhale, Miranda, Murnikov, Netes, and Sartin [Cla*99] (weighted) and Cotter and Smyth [CS00] (mixed). A more extensive overview may be found in the survey by Burke [Bur07] or in the work of Hussein, Linder, Gaulke, and Ziegler [Hus*14], who propose a framework for implementing hybrid systems in accordance to these strategies, also taking contextual factors into account. In the second case, algorithms are closely integrated with each other, for instance, taking content attributes and user-item feedback into account at the same time (feature combination). In practice, system providers usually employ such individual solutions: Tighter coupling often appears more meaningful in light of the complex real-world requirements regarding brand image, scalability and existing infrastructure. Examples are the recommenders of *Google News* [Das*07] or *YouTube* [Dav*10].

Techniques of the same nature may be combined as well [Bur07]. Yet, such constellations are usually referred to as *ensembles*. An example is the winner solution of the *Netflix* prize that coupled together hundreds of collaborative filtering models [BKV10; FHK12]. Regardless of the advantages of other methods (and their combination) discussed in this section, this underlines again the popularity of collaborative filtering. Next, given our goal of countering this method's deficiencies in terms interactive control, we consequently turn our attention back to this method, specifically, the technique that forms the basis for our presented developments.

2.2 Recommender systems based on matrix factorization

In this section, we explain *matrix factorization* in more detail, one of the most popular techniques to implement collaborative filtering systems. We describe the *latent factor models* that are learned by the application of matrix factorization algorithms, and how this is done via *objective functions* using specific *optimization methods*. Moreover, we elaborate on *enhancements* proposed to improve the algorithms as well as on the few works that address *further use cases* and exploit the abstract models beyond accurate predictions.

2.2.1 Latent factor models

As mentioned in the previous section, pure model-based collaborative filtering algorithms derive models entirely from user-item feedback as contained in a standard user-item matrix $\mathbf{R}$. Accordingly, this is also true for models that result from the application of matrix factorization algorithms. The nature of these models is entirely statistical, obscuring any meaning the model dimensions might have. This is the reason why it becomes, in contrast to content-based but also memory-based collaborative filtering techniques, much more difficult to explain the recommendations [HKV08]. Nevertheless, it is widely accepted that these dimensions represent real-world concepts, obvious ones such as the degree of humor movies comprise, but also subtle characteristics such as how much users appreciate the existence of a romantic love story [KBV09]. This way, the dimensions span a *latent factor space*, which may look as illustrated in Figure 2.2.

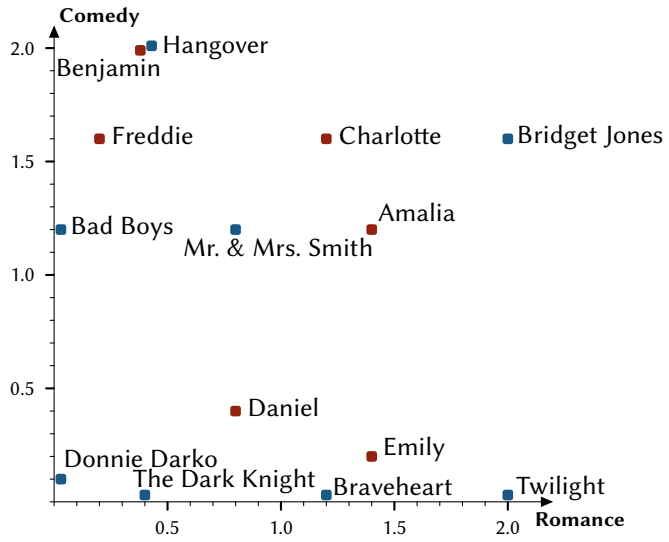

Figure 2.2 Example of a latent factor model with two factors representing the degree of comedy and romance in movies (blue), and the preferences of users (red) regarding these aspects. In contrast to this example, it is usually difficult to assign meaningful labels to latent dimensions.

This example illustrates well that the application of a matrix factorization algorithm embeds users and items into a single common space [KBV09; RK12]. Formally, such a factorization is expressed by two matrices, a *user-factor matrix* $\mathbf{P} \in \mathbb{R}^{|U| \times k}$ and an *item-factor matrix* $\mathbf{Q} \in \mathbb{R}^{|I| \times k}$. The predefined constant k represents the number of dimensions. In our example, we have $k = 2$

factors, whereas in practice, 10 to 100 factors are typically learned for achieving sufficient recommendation accuracy [KBV09; ERR14; KB15a]. Figure 2.3 shows an example that corresponds to the figure above, with **P** and **Q** derived from the user-item matrix shown in Table 2.1. Note that for illustration purposes, we only use positive values here, whereas negative values may equally occur unless a *non-negative algorithm* [LS99] is explicitly used.

$$\mathbf{P} = \begin{bmatrix} 1.4 & 1.2 & 0.4 \\ 0.4 & 2.0 & 1.2 \\ 1.2 & 1.6 & 0.2 \\ 0.8 & 0.4 & 2.0 \\ 1.4 & 0.2 & 0.8 \\ 0.2 & 1.6 & 1.6 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0.0 & 0.1 & 0.4 \\ 2.0 & 0.0 & 0.2 \\ 0.4 & 2.0 & 0.4 \\ 0.4 & 0.0 & 1.6 \\ 2.0 & 1.6 & 0.0 \\ 1.2 & 0.0 & 1.8 \\ 0.0 & 1.2 & 1.8 \\ 0.8 & 1.2 & 1.8 \end{bmatrix}$$

Figure 2.3 Examples of a user-factor and an item-factor matrix.

These matrices form a latent factor model with a much lower dimensionality than the original user-item matrix [KBV09; KB15a]. They describe the interests users have for the aforementioned characteristics and the degree to which items fulfill these characteristics. Thus, a user u's (calculated) interest in a particular factor f is numerically expressed by entry p_{uf} of **P**, whereas entry q_{if} of **Q** describes the (calculated) extent to which item i possesses this factor.

Recommendation function In accordance with these definitions, the inner product of a user-factor vector $\vec{p}_u \in \mathbf{P}$ and an item-factor vector $\vec{q}_i \in \mathbf{Q}$ captures the *interaction between user and item.*[4] A prediction $\hat{r}_{ui}$ for user u and item i can thus be calculated as follows:

$$s(i|u) := \vec{p}_u \cdot \vec{q}_i = \hat{r}_{ui} \,. \tag{2.3}$$

Figuratively speaking, users with a high interest in certain characteristics consequently receive recommendations of items that well represent these characteristics (cf. Figure 2.2). For all users and items in total, this corresponds to multiplying matrix **P** and the (transposed) matrix **Q**, leading to an approximated version $\hat{\mathbf{R}}$ of the original user-item matrix **R**:

$$\mathbf{R} \approx \mathbf{P}\mathbf{Q}^{\mathsf{T}} = \hat{\mathbf{R}} \,. \tag{2.4}$$

Given our sample factorization from Figure 2.3, such an *approximated matrix* $\hat{\mathbf{R}}$ may look as shown in Table 2.2. Inspecting the differences to the matrix **R** in Table 2.1 illustrates that the approximation error appears rather small, i.e. the deviation between estimated values and observed feedback is low. More importantly, $\hat{\mathbf{R}}$ is obviously a complete matrix, i.e. one equally obtains values for the feedback that was initially missing. Consequently, these predictions may be used to select and present the top n items as recommendations.

[4] Note that when referring to the corresponding rows of **P** and **Q**, we also write $\mathbf{p}_u$ and $\mathbf{q}_i$.

Table 2.2 Example of an approximation of a user-item matrix based on a user-factor and an item-factor matrix: Entries that were initially missing, i.e. related to items that may be recommended, are highlighted in bold. A comparison of the other entries with those of the original matrix shown in Table 2.1 allows determining the quality of these predictions.

	Donnie Darko	Twilight	Hangover	The Dark Knight	Bridget Jones	Braveheart	Bad Boys	Mr. & Mrs. Smith
Amalia	0.28	**2.88**	3.12	**1.20**	4.72	**2.40**	**2.16**	3.28
Benjamin	0.68	1.04	4.64	**2.08**	4.00	**2.64**	4.56	**4.88**
Charlotte	**0.24**	2.44	3.76	0.80	4.96	1.80	**2.28**	**3.24**
Daniel	0.84	2.00	**1.92**	3.52	**2.24**	**4.56**	4.08	4.72
Emily	**0.34**	2.96	1.28	1.84	**3.12**	3.12	1.68	**2.80**
Freddie	**0.80**	**0.72**	3.92	2.64	**2.96**	3.12	4.80	**4.96**

2.2.2 Objective functions

Latent factor models consisting of a user-factor matrix $\mathbf{P}$ and an item-factor matrix $\mathbf{Q}$ can be learned by a wide range of factorization methods. In contrast to *singular value decomposition* [FMM77], which has already for a long time been used in the area of information retrieval for dimensionality reduction [Dee*90], the matrix factorization algorithms that are employed in recommender systems *only approximate the original data*, as shown above. If adequately parameterized, this leads to very accurate results [KBV09; KB15a]. At the same time, these algorithms are not exclusively defined for complete matrices: For collaborative filtering recommender systems, it is necessary to efficiently handle sparse user-item matrices as they are typical for this application scenario [KBV09; KB15a]. Under consideration of the goal of approximating *only* the original data, but producing highly accurate predictions in the *other* cases, this can be achieved by means of *objective functions* and *optimization methods* (see next section) that exclusively rely on user-item interaction data that have been observed in the past.

2.2.2.1 Rating prediction

The objective functions that are most frequently used by matrix factorization algorithms try to *minimize the squared error* for the known entries of the user-item matrix $\mathbf{R}$ with the help of the two model matrices $\mathbf{P}$ and $\mathbf{Q}$ [Fun06; KBV09]: between all given ratings $r_{ui} \in R$ and the corresponding predictions $\hat{r}_{ui}$, calculated by dot multiplication of user-factor vectors $\vec{p}_u$ and item-factor vectors $\vec{q}_i$, the (squared) differences $(r_{ui} - \hat{r}_{ui})^2$ should be as small as possible.[5] Put differently, the overall goal is to minimize the *root mean square error* (RMSE). For a long time, this was the most widely used recommendation accuracy metric [GS15], which also served as the optimization target in the *Netflix* prize competition [BL07]. If this goal can be achieved, it is assumed that the resulting model adequately reflects actual user behavior, and more importantly, that the quality of the predictions for the previously unknown entries is sufficiently high. To lay the focus on the *missing* user-item feedback instead of already observed data points, i.e. to

[5] For the sake of simplicity, we use conventional ratings as a running example. In principle, however, any type of user-item feedback (see Section 2.3.1) may be used.

abstract from training data, a *regularization term* is usually added [KBV09]. This way of reducing model complexity is particularly important due to the high sparsity of the user-item matrix, and prevents overfitting. The resulting *objective function* for finding a global minimum may look as follows:

$$\min_{\mathbf{P},\mathbf{Q}} \sum_{r_{ui} \in R} (r_{ui} - \vec{p}_u \cdot \vec{q}_i)^2 + \lambda(\|\vec{p}_u\|^2 + \|\vec{q}_i\|^2)\,, \tag{2.5}$$

where λ, a parameter set using cross validation, controls the regularization. This type of regularization is frequently applied in machine learning. It uses the Euclidean norm $\|\cdot\|$ to punish larger values (also known as *L2-norm* or *Tikhonov regularization*). Yet, it also constitutes the most common type in recommender systems. Other techniques such as Lasso regularization [Tib96] exist, but led to inferior results in prior research [BKV07b; FHK12]. Therefore, we do not address alternative regularization techniques in more detail.

Usually, the variance that can be observed in user-item feedback is not entirely explained by the interaction between users and items as expressed by $\vec{p}_u \cdot \vec{q}_i$, but to some proportions by effects that are directly associated with either users or items. For instance, a specific user may provide on average higher ratings than others, while a certain item usually receives ratings below the arithmetic mean [KBV09]. Accordingly, so-called *biases* are frequently integrated into the previously shown objective function:

$$\min_{\mathbf{P},\mathbf{Q}} \sum_{r_{ui} \in R} (r_{ui} - \vec{p}_u \cdot \vec{q}_i - \mu - b_u - b_i)^2 + \lambda(\|\vec{p}_u\|^2 + \|\vec{q}_i\|^2 + b_u^2 + b_i^2)\,, \tag{2.6}$$

where b_u and b_i represent user and item bias, respectively, and μ the global average rating. Naturally, these biases need to be added back again when calculating the predictions $\hat{r}_{ui}$ at runtime. The updated *recommendation function* from (2.3) may look as follows:

$$s(i|u) \coloneqq \vec{p}_u \cdot \vec{q}_i + \mu + b_u + b_i\,. \tag{2.7}$$

More advanced ways of taking biases into account have been proposed [cf. KB15a]. In general, any kind of normalization of the user-item matrix is useful [KBV09; KB15a], increasing accuracy and leading to performance improvements because the corresponding effects do not need to be captured by the latent factors [Sar*00; Fun06]. Since enhancements concerned with quality, performance or scalability do not have any impact on the applicability of the interactive methods we propose, they are however outside the scope of this thesis. The same applies to other common extensions of the objective function, for instance, regarding implicit feedback or temporal dynamics. Instead, we refer to the literature for more details [KBV09; ERK11; KB15a].

2.2.2.2 Learning to rank

As an alternative to an objective function for rating prediction, Rendle, Freudenthaler, Gantner, and Schmidt-Thieme [Ren*09] proposed *Bayesian personalized ranking*, which in contrast to the previously described standard task focuses on "learning to rank". This means, presenting items in the right order is considered more important than predicting their scores most accurately. While this is often seen as the more realistic task, the use of this kind of function is still more rare.

The original model has been proposed for implicit feedback. Accordingly, the ranking is learned based on pairwise comparisons of items that have been observed by the user, with items that have

not been observed (1 or 0 entries in the user-item matrix). Lerche and Jannach [LJ14] suggested an extension that allows integrating implicit feedback on different levels: In a certain percentage of cases, their altered model takes weights into account (any positive number), for example, to differentiate between clicks and purchases. Thus, the ranking can additionally be learned from item pairs where feedback is available for both items. This way, not only the ranking's quality may be improved because all items can be put into an order, but it becomes possible to directly adopt this approach for explicit feedback: When comparing two items, the one with the higher rating is considered to have a higher weight, and therefore to be on a higher position.

In any case, the Bayesian optimization criterion requires item pairs as training data. User-specific preferences are thus represented as triples (u, i, j), expressing whether a user u prefers an item i over an item j. Hence, i is the positive sample (in the original formulation the item the user has interacted with) and j the negative sample (some unobserved item, or the one with lower weight). With σ being the logistic sigmoid and D a set containing these triples [cf. Ren*09], this results in the following *objective function*:

$$\begin{aligned}\max_{\mathbf{P},\mathbf{Q}} & \sum_{(u,i,j)\in D} \ln \sigma(\hat{r}_{uij}) && - \lambda\left(\|\vec{p}_u\|^2 - \|\vec{q}_i\|^2 - \|\vec{q}_j\|^2 - b_i^2 - b_j^2\right) \\ = & \sum_{(u,i,j)\in D} \ln\left(\frac{1}{1+\mathrm{e}^{-\hat{r}_{uij}}}\right) && - \lambda\left(\|\vec{p}_u\|^2 - \|\vec{q}_i\|^2 - \|\vec{q}_j\|^2 - b_i^2 - b_j^2\right) \\ = & \sum_{(u,i,j)\in D} -\ln(1+\mathrm{e}^{-\hat{r}_{uij}}) && - \lambda\left(\|\vec{p}_u\|^2 - \|\vec{q}_i\|^2 - \|\vec{q}_j\|^2 - b_i^2 - b_j^2\right),\end{aligned} \tag{2.8}$$

with $\hat{r}_{uij}$ capturing the relationship between user u and items i and j according to the underlying model [cf. Ren*09]. With standard matrix factorization, this *estimator* is decomposed as follows:

$$\hat{r}_{uij} \coloneqq \hat{r}_{ui} - \hat{r}_{uj} \text{ with } \hat{r}_{uk} \coloneqq \vec{p}_u \cdot \vec{q}_k + b_k\,. \tag{2.9}$$

Note that global average and user bias cancel out due to the way $\hat{r}_{uij}$ is calculated. The regularization term is subtracted because the objective function gets maximized. As a side note, this optimization task has been shown to be analogous to maximizing the *area under curve* [Ren*09].

2.2.3 Optimization methods

While less effort is required at runtime when using a matrix factorization algorithm ($|I|$ dot multiplications and logarithmic sorting of the results), learning the underlying model is computationally expensive and may actually take some time given the quadratic problem that needs to be solved. Consequently, model training is performed entirely offline, namely with the help of an *optimization method* for one of the objective functions. Since the corresponding optimization criteria are differentiable, algorithms based on *stochastic gradient descent* [Fun06; Zho*08] or *alternating least squares* [Tak*09] constitute a natural choice for the respective minimization or maximization task [Ren*09]. In the following, we detail on the former, as some of the developments we present in this thesis depend on this technique. Before, however, we briefly elaborate on *singular value decomposition* [FMM77] and its relation to matrix factorization.

2.2.3.1 Singular value decomposition

Singular value decomposition is a mathematically well-defined method for factorizing a matrix. The method is widely used in a number of application areas, forming the basis for statistical principal component analysis [cf. FHK12] and various dimensionality reduction techniques [cf. Dee*90; ERK11]. In context of recommender systems, applying singular value decomposition on a user-item matrix $\mathbf{R}$ may be defined as follows:

$$\mathbf{R} \coloneqq \mathbf{X}\Sigma\mathbf{Y}^{\intercal} \text{ with } \mathbf{X} \in \mathbb{R}^{|U|\times|U|}, \Sigma \in \mathbb{R}^{|U|\times|I|}, \mathbf{Y} \in \mathbb{R}^{|I|\times|I|}. \tag{2.10}$$

$\mathbf{X}$ and $\mathbf{Y}$ contain the left- or right-singular vectors, respectively. Σ is a diagonal matrix containing the non-zero *singular values* in descending order. All three matrices have orthogonality constraints. The singular vectors correspond to the column and row spaces of the original matrix $\mathbf{R}$, sorted according to the singular values in descending order. Such a decomposition can be obtained in an exact manner by using numerical techniques. For practical application, the dimensionality can subsequently be reduced. This leads to a more economical decomposition that represents the best possible reconstruction of the original data for a specified rank k in terms of the Frobenius norm. To get an approximated user-item matrix $\hat{\mathbf{R}}$, the matrices $\mathbf{X}$, $\mathbf{Y}$ and Σ are truncated to retain only those entries that correspond to the k largest singular values. This results in an updated problem formulation:

$$\hat{\mathbf{R}} \coloneqq \mathbf{X}_k\Sigma_k\mathbf{Y}_k^{\intercal} \text{ with } \mathbf{X} \in \mathbb{R}^{|U|\times|k|}, \Sigma \in \mathbb{R}^{|k|\times|k|}, \mathbf{Y} \in \mathbb{R}^{|I|\times|k|}. \tag{2.11}$$

Simplification of this formula by multiplying each outer matrix with the square root of the inner matrix leads to a formulation similar to the one presented in (2.4), i.e. with a user-factor and an item-factor matrix [Sar*00]. This allows for predictions as described above. However, singular value decomposition is *only defined for complete matrices*. For content-based techniques (cf. Section 2.1.2), this does not raise any problems, since term-document matrices naturally have no empty cells. Early attempts to model-based collaborative filtering impute missing entries prior to the decomposition [Sar*00; KY05] or derive dense submatrices from the original user-item matrix [Gol*01]. While these approaches improve accuracy in comparison to some memory-based techniques, they are *inferior to modern variants* based on stochastic gradient descent or alternating least squares, i.e. optimization methods that are specifically designed to work *only* on known user-item feedback. Moreover, the imputation mechanisms lead to overfitting and have deficiencies in terms of scalability. Nonetheless, the modern variants are often called "SVD-like" or "regularized SVD" due to the mathematically strong relation, although the factorizations lack orthogonality constraints and do not contain singular values [RS08; NZ13].

2.2.3.2 Stochastic gradient descent

With an objective function as in (2.6), it is possibly to rely exclusively on observed user-item interaction data to come up with a well approximated user-item matrix $\hat{\mathbf{R}}$, namely by minimizing the sum of squared differences to the individually predicted scores. To solve this quadratic optimization problem, a conventional method would use all observed data points for calculating exact gradients, which requires high computational effort. In contrast, *stochastic gradient descent* pursues the idea of taking only *single* observations into account and approximating *local* gradients, which leads to faster convergence [Tak*09; Gem*11]. Building on (2.6), this means that

in each step, only the *squared prediction error* regarding a single rating of user u for item i is considered:

$$e_{ui}^2 \coloneqq (r_{ui} - \vec{p}_u \cdot \vec{q}_i - \mu - b_u - b_i)^2 \,. \tag{2.12}$$

Instead of following the exact gradient, the minimum is then reached by iteratively moving in the direction of the local gradients, i.e. by minimizing each single error calculated as above. For this, *partial derivatives* with respect to the optimization parameters, i.e. all $\vec{p}_u \in \mathbf{P}$ and $\vec{q}_i \in \mathbf{Q}$, or their individual components, respectively, need to be determined:

$$\begin{aligned}
\frac{\partial}{\partial p_{uf}} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2) &= -2e_{ui} \cdot q_{if} + 2\lambda p_{uf} &&\propto -e_{ui} \cdot q_{if} + \lambda p_{uf} \,, \\
\frac{\partial}{\partial q_{if}} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2) &= -2e_{ui} \cdot p_{uf} + 2\lambda q_{if} &&\propto -e_{ui} \cdot p_{uf} + \lambda q_{if} \,, \\
\frac{\partial}{\partial b_u} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2) &= -2e_{ui} + 2\lambda b_u &&\propto -e_{ui} + \lambda b_u \,, \\
\frac{\partial}{\partial b_i} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2) &= -2e_{ui} + 2\lambda b_i &&\propto -e_{ui} + \lambda b_i \,.
\end{aligned} \tag{2.13}$$

These derivatives can be used to formulate *rules for updating* the user-factor and item-factor values in the opposite direction of the local gradient. As a consequence, the prediction error for the corresponding user-item pair gets smaller, to the extent specified by the learning rate (or step size) η, which can also be adjusted dynamically [cf. Gem*11]:

$$\begin{aligned}
p_{uf} &\leftarrow p_{uf} - \eta(-e_{ui} \cdot q_{if} + \lambda p_{uf}) &&= p_{uf} + \eta(e_{ui} \cdot q_{if} - \lambda p_{uf}) \,, \\
q_{if} &\leftarrow q_{if} - \eta(-e_{ui} \cdot p_{uf} + \lambda q_{if}) &&= q_{if} + \eta(e_{ui} \cdot p_{uf} - \lambda q_{if}) \,, \\
b_u &\leftarrow b_u - \eta(-e_{ui} + \lambda b_u) &&= b_u + \eta(e_{ui} - \lambda b_u) \,, \\
b_i &\leftarrow b_i - \eta(-e_{ui} + \lambda b_i) &&= b_i + \eta(e_{ui} - \lambda b_i) \,.
\end{aligned} \tag{2.14}$$

Listing 2.1 shows a typical implementation of this algorithm in pseudo code: After $\mathbf{P}$ and $\mathbf{Q}$ are initially set to random values [cf. Fun06; Tak*09], the algorithm iterates several times over all user-item pairs for which feedback data are available (lines 1–17). The respective prediction error is determined (lines 3–7) and all factor values are set to new values according to the updates rules (lines 9–15, bias updates are not shown). Thus, the prediction quality increases for each individual pair, while the total error is simultaneously reduced in a stepwise manner. The algorithm terminates after a given number of repetitions (`num_iters`), alternatively, as soon as the relative improvement gets smaller than a predefined constant ϵ. Other variations are possible, for instance, regarding the way factors are taken into account [FHK12], i.e. separately one after the other, as shown by Funk [Fun06], or all at once, as shown in the listing.

2.2.3.3 Alternative methods

Stochastic gradient descent most frequently forms the basis for recommender systems that make use of matrix factorization algorithms [KBV09]. This method allows not only very efficiently to find an approximate solution for the underlying optimization problem, but is also straightforward to implement. Other approaches come with other advantages: In *alternating least squares*, either all user-factor vectors or all item-factor vectors are taken as fixed. In each iteration, one of these

Listing 2.1 Pseudo code for matrix factorization with stochastic gradient descent.

```
input:  known_ratings: existing user-item feedback
        p, q: randomly initialized arrays
        k, num_iters, eta, lambda: predefined constants

for iter := 1 to num_iters do
  for r_(u, i) in known_ratings do
    double prediction := 0.0;
    for f := 1 to k do
      prediction := prediction + (p[u][f] * q[i][f]);
    end;
    double err = r_ui - prediction;

    for f := 1 to k do
      double p_uf = p[u][f];
      double q_if = q[i][f];

      p[u][f] := p_uf + eta * (err * q_if - lambda * p_uf);
      q[i][f] := q_if + eta * (err * p_uf - lambda * q_if);
    end;
  end;
end;
```

vector sets is then recomputed in an alternating manner, by solving the resulting, now convex least-squares subproblem. A single iteration is thus computationally more expensive. But, fewer iterations are necessary, efficiency for more densely filled matrices is higher, and parallelization can easily be achieved due to the independent recalculation of $\mathbf{P}$ and $\mathbf{Q}$. On the other hand, it is required that these matrices (on a rotating basis) fit entirely into memory [BKV07a; BK07; Zho*08; KBV09; PZT10]. However, for all the developments we present in this thesis, implementation details are not important because each of the interactive methods is applicable independent of the specific algorithm that is used in the background. Accordingly, we deem it unnecessary to provide a more detailed explanation of alternative methods in this chapter, especially of those with an underlying principle that is similar to stochastic gradient descent. Instead, we again refer to the literature [ERK11; FHK12; KB15a].

Also concerning the Bayesian personalized ranking approach, further details may be of interest. For instance, a sampling technique needs to be implemented, i.e. a negative sample item j has to be selected for each r_{ui}, either an unobserved item [Ren*09; RF14], or an item with lower weight [LJ14]. Moreover, derivatives and corresponding update rules are necessary, as shown for the rating prediction approach. For the same reasons as mentioned above, we omit these details in this chapter, but present the most important ones in Appendix D.

2.2.4 Algorithmic enhancements

The last couple of years have shown that the improvements that still seem possible with respect to recommendation accuracy of model-based collaborative filtering algorithms neither seem particularly beneficial from a subjective user perspective [KR12; PCH12], nor worth the implementation effort for system providers. The latter is one of the key takeaways from the *Netflix* prize competition [AB15], not only true because the company's business case has changed (from movie

rental service to streaming platform), but, in particular, because integrating the winner's solution would not have compensated the impact of other aspects meanwhile considered more important for the success of recommender systems. Nevertheless, research has tried to increase recommendation quality in terms of objective performance metrics even further [GS15; KW15]. In offline experiments, numerous authors have shown that enhancing existing algorithms with side information is one of the most promising avenues for achieving this goal [e.g. Kar*10; ML13; SLH13; NZ13; FC14; Alm*15]. As this also applies to matrix factorization, we now review which *types of information* may be considered, and which *techniques for integrating this information* exist.

Note that many other attempts have been made for improving the accuracy of matrix factorization algorithms. These include pairwise or listwise training [RF14; Ste15; KRT16], transfer of models across domains [PC15], modeling of complex user preferences with the help of feature projection methods [Zha*15] or non-linear factorizations with multiple user-factor vectors [WWY13], and even learning two factor models at once for reciprocal recommendation scenarios such as online dating [NP19]. Yet, we omit a discussion of these algorithmic advances as they have no direct relation to the interactive methods we propose in this thesis.

2.2.4.1 Types of additional information

One of the most promising, and, at the same time, easy-to-implement approaches to increase accuracy is to complement the user-item interaction data that are usually fed into matrix factorization algorithms with *additional information* related to user or items. This may include generic *implicit feedback* [Liu*10], which is typically available in greater amounts than explicit feedback. Furthermore, it may be taken into account in multiple variants at once, for instance, combining binary purchase signals with continuous satisfaction signals automatically determined based on dwell time [LKB19]. Also, *temporal relations* of ratings may be considered [ZI13], which is often beneficial due to the dynamics in user rating behavior [cf. KBV09; KB15a]. On the other hand, more specific datasources may be leveraged as well: The approaches by Karatzoglou, Amatriain, Baltrunas, and Oliver [Kar*10] and Hidasi and Tikk [HT12] rely on *contextual information*, for example, user age or current season. Forbes and Zhu [FZ11] and Nguyen and Zhu [NZ13] exploit *predefined metadata* about movie genres and recipe ingredients, and refer to this approach as "content-boosted" matrix factorization.

Beyond that, more and more authors semantically analyze user-written *product reviews*. Reviews play an important role in buying decisions, forming a highly useful source of information about what users typically like or dislike, especially in case of experience products [CM06; SB11]: Once hidden topics or opinions about items are inferred using content-based techniques such as latent semantic indexing, the derived concepts can be integrated with standard matrix factorization models [ML13; Dia*14; Zha*14; Alm*15]. While in some of these works, the fundamental explainability of recommendations has already been considered as a secondary optimization target, the focus recently shifted completely towards this aspect. For this, review texts are analyzed with natural language processing techniques based on modern deep learning methods [LDS18; Hou*19]. Nonetheless, the output is still combined with regular latent factor models, which are simply learned in parallel. Requiring further preprocessing, *more complex information* has been taken into account as well, for instance, by extracting audio and visual features from movies [DEC16; Del*19] or accessing knowledge databases for cross-domain metadata [Fer*19].

User-generated data in the form of *tags* have for a long time received only little attention [TMS08; ZLY09], if at all, for recommending tags as an aid for users in annotation or search tasks [cf. SNM08; Kam*09; Mar*10]. Tso-Sutter, Marinho, and Schmidt-Thieme [TMS08] suggested for the first time to use this kind of information for the standard recommendation task: As an extension to conventional memory-based collaborative filtering, the authors augment the user-item matrix by user preferences for tags or by tag-based item descriptions, apply user- and item-based collaborative filtering, and finally combine the results. In line with that, only few matrix factorization approaches take advantage of tags. The few exceptions enhance the underlying latent factor models with keywords describing movies, their mood and plot, generic social tags, or tags crawled from recipes and ingredient lists [ZLY09; SLH13; BJG13; FC14; Ge*15]. Yet, these extensions focus again on *improving objective accuracy*. Also, these as well as the approaches mentioned before in this section have primarily been studied in *retrospective offline experiments*. Becerra, Jimenez, and Gelbukh [BJG13] additionally derive tag-based visualizations of user profiles, using a method that would in principle also allow for manipulation of the results. Ge, Elahi, Fernández-Tobías, Ricci, and Massimo [Ge*15] actually enable users to indicate preferences via tags, but consider these preferences only during offline training (due to the integration technique, see next section). These exceptions, at least, contain qualitative analyses [BJG13] or small user studies [Ge*15], but focus on general usability instead of the influence of the additionally considered tags. Hence, empirically investigating the *influence of side information on user experience* is still an open subject.

On a side note, a few approaches have been proposed that *exclusively rely on tags* for generating recommendations, i.e. which do not act as extensions to collaborative filtering techniques. Among others, these approaches use conventional information retrieval methods [Gre*09], graph-based techniques [Gua*10], or directly model user preferences based on item-tag signals [SVR09; NR13]. Naturally, these standalone solutions can *neither benefit from the algorithmic maturity* of model-based collaborative filtering techniques, *nor from the availability of long-term preference profiles* based on implicit or explicit user-item feedback data—for which it has been shown that in terms of accuracy, just collecting more of the same may have a larger impact than metadata [PT09; FO19]. Finally, it is worth mentioning that apart from few exceptions, which we discuss later in context of interactive recommending (see Section 2.3), also these pure tag-based recommender systems have *not been designed for improving user control*.

2.2.4.2 Techniques for integrating additional information

The range of *techniques* for considering side information is very broad as well. For matrix factorization, a straightforward approach is adding *further constraints* to the objective function (cf. Section 2.2.2). This may increase accuracy [KBV09; FC14; Ge*15], but has the consequence that after having been learned, the latent factors *exhibit no interpretable association* with the additional information: The provided data are calculated into the factor values in a way that the relations to the factors, and consequently users and items, cannot be accessed anymore. The same applies to approaches that add *regularization terms* [e.g. ZLY09; ML13; SLH13], even the most recent ones with complex calculations behind these terms [LDS18; CSZ19; Fer*19; Hou*19].

However, other approaches use the information explicitly with the intention of establishing content-related associations with the factors: By proposing a *regression-constrained* formulation, the content-boosted technique by Forbes and Zhu [FZ11] considers the item-factor vectors as

functions of content attributes, which are thus still accessible later in the process. Specifically, the technique replaces the item-factor matrix $\mathbf{Q}$ in the standard matrix factorization formulation shown in (2.4) by $\mathbf{AB}$, resulting in:

$$\mathbf{R} \approx \mathbf{P}\mathbf{Q}^\top = \mathbf{P}(\mathbf{AB})^\top, \tag{2.15}$$

with $\mathbf{A} \in \mathbb{R}^{|I| \times d}$ associating the items with each of the d content attributes, and $\mathbf{B} \in \mathbb{R}^{d \times k}$ containing the values of the k latent factors for these attributes. Since this formulation serves as the basis for some developments we present in this thesis, we refer not only to the original literature [FZ11] and subsequent work [NZ13; Zha*14] for more details, but also to Chapter 5, in particular, for an overview of how the objective function and the optimization method need to be adapted. Somewhat similar is the work by Becerra, Jimenez, and Gelbukh [BJG13], but they completely replace the item-factor matrix by an item-attribute matrix during the learning phase.

Beyond these variants that redefine the standard formulation, more *complex techniques* include extended probabilistic matrix factorization [Dia*14; LKB19]; deep learning [Alm*15; WWY15]; factorization machines [NKB14]; mapping functions [Gan*10; PT09], sometimes keeping the benefits of pure singular value decomposition by jointly factorizing user-item interaction data and side information [FO19]; and the generalized variant of matrix factorization, tensor factorization [Kar*10; HT12]. Again, these techniques hardly make it possible to access the additional information once integrated, in particular, for practical purposes such as exposing this information in the user interface to give users control over the model or to convey its semantics. Furthermore, whereas all mentioned techniques have shown advantages in terms of objective accuracy metrics, there is a *lack of empirical user experiments*. As a consequence, the *effects of integrating collaborative filtering models with additional information* on the subjective assessment of aspects such as recommendation quality and on user experience still need to be investigated.

2.2.5 Further use cases

Up until today, the research efforts to improve model-based collaborative filtering are for the most part targeted at *accurately* and *efficiently* determining which items to recommend. The usage of latent factor models, including the enhancements to the underlying algorithms described in the previous section, is rarely dedicated to other purposes. A reason might be that while manual classification of users or items is a cumbersome process—requiring content-based techniques, expert knowledge, or other expensive and poorly scalable methods—matrix factorization algorithms provide such a result without further ado. However, as in other factor analysis or dimensionality reduction techniques, identifying the meaning of the resulting dimensions is often a problem, especially due to possible incoherence in the underlying semantics and the increasing redundancy with more factors.

Nonetheless, other authors [e.g. Tka*11; Gra11] as well as we ourselves [cf. KLZ18a; KLZ18b; Kun*19b] have successfully investigated the relations of latent factors to both user characteristics and item properties. The results provide evidence for the—already for a long time accepted—assumption that the dimensions of latent factor models actually represent *real-world concepts* (cf. Section 2.2.1). Still, these dimensions have to be considered overall hard to explain due to their statistical nature, especially in a more *automated* manner than in the works just mentioned. This goes back to the fundamental problem of model-based systems, in which users lack a deeper understanding of the underlying mechanisms due to their black-box characteristics. Hence, it is

highly difficult to reveal the inner logic and explain the output [XB07; PCH12; TM15]. Overall, latent factor models therefore have rarely been exploited for practical purposes related to the transition of conventional automated recommender systems to *interactive* and *transparent* applications. However, a few exceptions have shown the potential for improving these user-oriented aspects, beyond increasing accuracy and performance. In the following, we provide an overview of these *alternative use cases*.

2.2.5.1 Support in cold-start situations

Since the *cold-start problem* depicts one of the most severe problems in collaborative filtering systems, many authors have addressed this issue in their work. Mostly, *algorithmic solutions* have been proposed, ranging from simple strategies based on popularity and entropy [Ras*02; RKR08] to more advanced active learning methods [Rub*15; ERR16]. Active learning is a widely used approach that aims at minimizing uncertainty by quickly collecting as much meaningful item feedback from users as possible. It has been adopted to latent factor models as well: Karimi, Freudenthaler, Nanopoulos, and Schmidt-Thieme [Kar*12] suggest implementing an *online updating* mechanism for user-factor vectors. In a stepwise manner, users are asked to rate items that are popular among users who are similar in terms of their latent factors values. Manzato [Man12] proposes to integrate genre information and to overcome the sparsity problem by factorizing a user-genre matrix, allowing for predictions even in case users have not provided feedback for items of certain genres. Other solutions use decision trees [ZYZ11; RK12], multi-armed bandits [ZZW13; CHR16], or alternative definitions of the optimization problem [Sep*18], to determine which items to recommend next or to use in an interview process for eliciting (absolute or relative) user preferences. Often, these solutions come with *high user effort or lack scalability*. For this reason, again with the help of additional information, other authors propose to automatically infer values of user-factor or item-factor vectors. They use functions that map the demographics of new users or metadata of new items to latent factors, either during the optimization process [PT09] or as a postprocessing step [Gan*10].

As it is easy to imagine, a user-factor vector might also be initialized by means of a mapping from personality traits or social media profiles, as well as by specifically asking questions regarding the user's preferences. The latter corresponds to the few exceptions that realize an *interactive elicitation of user preferences* based on the properties of latent factor spaces, instead of trying to alleviate the new user cold-start problem in a fully automated manner as described above: Graus and Willemsen [GW15] propose a choice-based technique where users are confronted in a number of steps with sets of recommended movies, each time being asked to settle on one of them. In a similar fashion, Taijala, Willemsen, and Konstan [TWK18] rely on binary rating feedback. Either way, the user-factor vector of the current user is updated in accordance with the item-factor vector of the selected or rated item, making the user slowly traverse the latent factor space from an initial position towards items he or she more likely prefers. User experiments have shown that users comprehend this kind of navigation and use it in a sensible way [GW15; TWK18]. Without a meaning in the underlying model dimensions, this would not have been possible. Comparable to these examples, other authors took inspiration from the work presented in this thesis: For instance, based on non-negative matrix factorization,[6] Liu, Han, Iserman, and Jin [Liu*18] adopt

[6] For more details on non-negative matrix factorization approaches for recommender systems, we refer to the literature [e.g. Zha*06; Tak*08; RS08; Tak*09; Luo*14].

our approach of presenting sets of representative sample items (see Chapter 4), but determine the factors for the iterative preference collection process in a personalized manner.

Beyond these approaches for *initial* preference elicitation, there are no interactive approaches that allow users to intervene *later* in the recommendation process. In particular, it is *not possible to exert influence* on the underlying latent factor models once a user representation exists, apart from indicating preferences by rating further items. The only exception is the approach by Taijala, Willemsen, and Konstan [TWK18], where it is possible to continue the preference elicitation from a location in the latent space that relates to an existing user-factor vector.

Finally, although not in the focus of this thesis, it is worth mentioning that significant effort has also been spent on alleviating the *new item* cold-start problem: Deldjoo, Dacrema, Constantin, Eghbal-Zadeh, Cereda, Schedl, Ionescu, and Cremonesi [Del*19] present a framework that suggests to first apply collaborative filtering (possibly using a matrix factorization algorithm), and then a content-based method, to recommend new items based on the introduced (latent) features. Aleksandrova, Brun, Boyer, and Chertov [Ale*17] use non-negative matrix factorization to improve explainability (see next section). For this, they identify representative users in the style of neighborhood-based techniques. These users are promoted to seed users who are specifically asked to provide ratings for new items. In contrast to asking top raters or diverse users, this leads more quickly to item vectors that represent well the entire population.

2.2.5.2 Alternative optimization targets

While improving the support in cold-start situations is "only" beneficial for providing new users more quickly with better recommendations, matrix factorization algorithms also bear potential to help all users at any time in the recommendation process. For instance, *optimization criteria* that go beyond accuracy have become increasingly popular: Based on earlier findings that higher model dimensionality goes hand in hand with more accurate predictions for long-tail items [cf. CKT10], Coba, Symeonidis, and Zanker [CSZ19] propose to increase the number of *novel items* in the recommendation sets. Willemsen, Graus, and Knijnenburg [WGK16] aim at *diversification* of these sets. For addressing this criterion, which is particularly important due to its immediate effect on user satisfaction [cf. Bol*10], they maximize the distances of the top n items in the underlying factor space. Khawar and Zhang [KZ18] analyze the relation of standard matrix factorization to eigenvectors and eigenvalues. From a theoretical point of view, they show that removing the top k eigenvectors—which they say correspond to global effects—may lead to recommendations of *less popular* but *more diverse* items.

Beyond these dimensions strongly related to the qualities of recommendation sets [GS15], *explainability* has gained significant attention also in context of matrix factorization algorithms: Already early, Forbes and Zhu [FZ11] and Nguyen and Zhu [NZ13] have shown as a side product of their work that not only objective accuracy may benefit from considering additional information (see Section 2.2.4). But, this may also help to interpret the models and the relationships between items in the resulting factor space, and consequently, to provide explanations. Rossetti, Stella, and Zanker [RSZ13] took a first step towards *automatically* explaining latent factors in textual form by associating them with topics inferred from unstructured item descriptions. However, their approach has never been evaluated with users. Later, review data became the major source of side information, used for increasing the fundamental explainability of recommendations by combining matrix factorization with content-based techniques [ML13; Zha*14] as

well as deep learning models [LDS18; Hou*19]. However, apart from qualitative examinations or online A/B tests [e.g. Zha*14; LDS18], these attempts to make the algorithmic output inherently easier to understand have been *evaluated only in offline experiments*.

Other authors take advantage of more specific matrix factorization variants: Aleksandrova, Brun, Boyer, and Chertov [Ale*14] determine representative users based on user-factor vectors resulting from a *non-negative* algorithm.[6] They present items as recommendations that these users like the most—similar to mentors in memory-based techniques. Khoshneshin and Street [KS10], Yin, Wang, and Yu [YWY12], and Moin [Moi14] replace the inner product in the objective function shown in (2.6) by the *Euclidean distance*. Consequently, users and items become embedded in a Euclidean factor space, which turns out more comprehensible and easier to visualize. Other authors add *regularization terms* that describe how explainable items are [AN16; AN17; CSZ19]. Although proposed under the term "explainable recommendations", all these approaches have in common that they *do not generate explanations* in the true sense of the word.

Note that there also exist a few approaches that learn interpretable models based on the output of matrix factorization algorithms, i.e. which explain the factor models *after* they have been learned [e.g. CR15]. This principle of explaining black-box models a posteriori instead of designing models that are *inherently* interpretable receives more and more criticism, among others, only recently in research on explainable artificial intelligence [cf. Rud19].

2.2.5.3 Visualizations

Beyond these rather straightforward use cases, latent factor models have also shown potential for *visualization purposes*. In general, visualizations may serve as alternatives to textual explanations, and, in particular, to the above approaches that claim to learn easier interpretable models. Visualizations are often more flexible from a system perspective, since they do not necessarily require additional information, which otherwise often is the case [TM15; HPV16]. Moreover, they can increase system transparency, including awareness of hidden alternatives, and provide a basis for implementing interaction mechanisms that allow to manipulate recommendations and explore new and diverse areas of potentially interesting items.

Already Koren, Bell, and Volinsky [KBV09] illustrated in one of the early, frequently cited publications on matrix factorization algorithms for recommender systems, that items are arranged in a meaningful way when they are positioned according to their latent factor values: The authors present an exemplary visualization of the first two dimensions of a factorization of the dataset from the *Netflix* prize competition [BL07]. The inspection of the resulting two-dimensional grid, which depicts sample items for the different areas, shows that it is easily possible to categorize the items, at least for someone who is partially familiar with the movie domain: One dimension contains lowbrow comedy and horror movies for a male audience as well as more serious drama and comedy movies with female lead actors. The other dimension contains quirky movies for very specific target groups as well as mainstream blockbusters. In light of the difficulties discussed with respect to transparency and explainability of model-based techniques, it thus appeared promising to further investigate *visualizations based on latent factor models*.

User and item visualizations Accordingly, Németh, Takács, Pilászy, and Tikk [Ném*13] propose to explain user and item characteristics by changing the standard matrix factorization formulation. They store factor values of users and items, respectively, in *two-dimensional grids*

instead of one-dimensional vectors. Based on these grids, items are then displayed in the user interface as feature maps, with a gray scaling indicating the importance of factors with respect to an item, and keywords correlated with these factors exemplifying their meaning. While there have been attempts to automatically relate factors with topics (see previous section), these keywords are manually determined by identifying common properties of items that have equally low or high factor values. Becerra, Jimenez, and Gelbukh [BJG13] replace the item-factor matrix by metadata. Using *simple additive weighting*, a method from multi-attribute decision making, they represent user ratings as linear combinations of content attributes. Then, they visualize the derived user profiles by means of *line graphs*, which represent the predicted interest or disinterest of users concerning these attributes. Although the latent knowledge gets lost in this way, the weights that result from this method would not only be valuable for visualization purposes, but also to allow for manipulation of the results—though such a functionality has neither been implemented nor evaluated.

Item space visualizations In contrast to these visual representations on the level of single entities (i.e. users or items), other visualizations provide users with an *overview of the entire item space* (or at least large parts). In a two-dimensional visualization, Wegba, Lu, Li, and Wang [Weg*18] arrange movies according to a *single factor* of the underlying model. Within the visualization, the authors indicate for certain ranges of this factor's values, how diverse the items are or how familiar they are to the user. All identified concepts are of similar simplicity. Nevertheless, the movies the current user has already watched, which are shown as examples representing these concepts, illustrate again the meaning behind the factors. The interaction concept is more focused on "storytelling" on part of the system than improving user control. Accordingly, the results of the user study are not particularly impressive, especially in light of the fact that only a comparison with a conventional top n recommendation list is made.

To provide an overview with respect to *all factors* at once, other authors use dimensionality reduction techniques such as multidimensional scaling or force-directed algorithms (see the survey by Kagie, Wezel, and Groenen [KWG10] for an overview of possible techniques) to reduce the high-dimensional latent space to two dimensions: The goal is to obtain a representation of the item space that reflects the item similarities that exist in terms of factor values, i.e. with similar items close to each other, dissimilar ones further apart. Moin [Moi14] uses an adapted objective function to find a Euclidean embedding. As shown in Figure 2.4 (left), the resulting map is yet rather simple and depends on the current user since it is drawn around his or her factor vector. Gansner, Hu, Kobourov, and Volinsky [Gan*09] use standard matrix factorization, but focus on the visualization itself, as visible in Figure 2.4 (middle): Their *TVLand* represents the entire item space, divided by a clustering mechanism into "countries" that reflect commonalities of the items. Areas that could be of particular interest to the current user (i.e. with recommended items) are highlighted within this map, providing navigational aid and improving transparency. Yet, *no interaction possibilities are given*, neither to facilitate the handling of the map, nor to use it as a means for preference elicitation—although the potential of interactive visualizations for large item databases has been shown long ago, especially for experience products [SPG00], and also in context of memory-based collaborative filtering [Gre*10].

Interactive item space visualizations Consequently, in our other work, we proposed to extend this approach: Still based on the map metaphor, we add a third dimension that captures

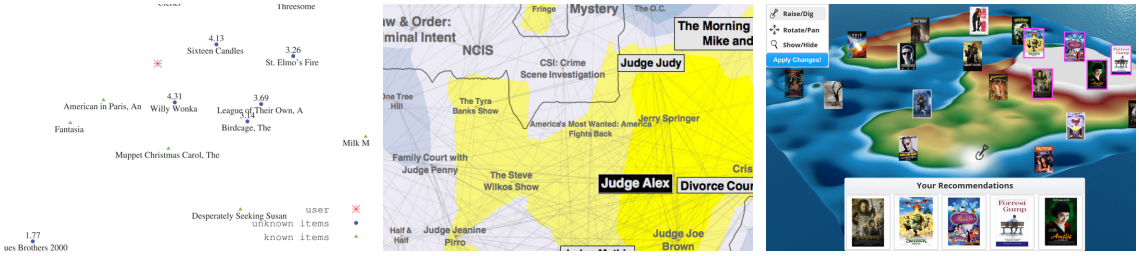

Figure 2.4 Screenshots of visualizations based on latent factor models, ordered by complexity and degree of interactivity: a rather simple map of the current user's direct surroundings, showing items and predicted scores (left) [Moi14]; an excerpt of a map representing the entire item space, divided into countries, highlighting areas of predicted interest (middle) [Gan*09]; and a map with an additional third dimension reflecting the user's preferences he or she can manipulate by means of the provided interaction tools (right) [KLZ17].

the current user's preferences [KLZ17]. In the prototypical implementation that is shown in Figure 2.4 (right), hills represent areas containing items for which highly positive ratings are predicted, valleys represent areas of items the user probably does not like. Interaction tools are provided for reshaping this landscape by scooping or digging, allowing users to interactively express their preferences for entire regions of the item space, instead of only for single items. Each update is immediately reflected back into the user's latent factor vector, triggering a recalculation of all predictions and leading to a new set of recommendations. Reshaping the landscape is possible without a user profile, starting from a flat surface, as well as with a user-factor vector learned by the underlying matrix factorization algorithm, and thus an elevation profile that corresponds to the user's long-term preferences right from the outset.

With this, the entire range of visualizations based on latent factor models is already covered. In general, visualizations in recommender systems are rare, especially in real-world environments [TM15; HPV16]. Nevertheless, the next section includes an overview of the few exceptions that have been proposed (without going into the above approaches in detail again).

2.3 Interactive methods

As should have become clear by now, state-of-the-art recommendation methods can be considered quite successful in finding suitable items with reduced interaction effort and cognitive load. As a consequence of their high degree of automation, users however often feel too much dominated, unable to flexibly specify current interests, and to target, for instance, diverse or novel recommendations. Overall, there are significant limitations in the ways users can exert influence. In many cases, they have no means at all to intervene during the recommendation process—although the aspect of *control* is strongly related to the *user experience* of recommender systems and considerably contributes to user satisfaction [XB07; KR12; PCH12; JJ17; Alv*19].

In typical commercial systems that employ collaborative filtering techniques [e.g. LSY03; BL07; GH15; SL17], the only option for users to actively affect the results is providing explicit feedback in the form of *ratings for single items* [JWK14]. This constitutes a possibility to influence these systems at least to some extent, but amplifies the *filter bubble effect* [Par11]: the user's existing long-term profile is only further refined, despite that the search goal may vary depending on the

current situation, and that hiding alternatives may not only be harmful on an individual level, but even for society. Non-negligible *interaction effort* is thus required in case users want to adjust the recommendations, but also to obtain suggestions in *cold-start situations*, i.e. when no historical data are yet available for a new user entering the system or when a user does not want an existing profile to be applied [SS11; CGT12]. At the same time, ratings have shown to suffer from a number of specific drawbacks [Ama*09; APO09; JBB11]. Beyond that, notably in real-world systems, recommendations are often the result of implicit user-item feedback [JWK14; RRS15b; JJ17]. This certainly has advantages, but also makes the recommendations *less transparent*—despite the fact that data regulation policies such as the *GDPR* increasingly demand the opposite [Har*19]. Moreover, it gets even harder for users to adapt the results to actual preferences and, in particular, situational needs.

Beyond indicating preferences through these types of feedback for single items, there exists a range of more advanced approaches: Relevance feedback mechanisms as introduced in information retrieval allow users to rate the utility of recommendations immediately after their presentation (e.g. "I do (not) find this item interesting") [SB97]. Richer interaction possibilities enable users, for example, to critique recommendations with respect to certain properties of the items (e.g. "I'd like to see movies with more comedy that are less dark") [CP12a]. Highly interactive recommenders even let users control the interplay of different methods and datasources in hybrid configurations [BOH12]. The surveys by He, Parra, and Verbert [HPV16] and Jugovac and Jannach [JJ17] provide extensive overviews of interactive methods for recommender systems. Nonetheless, after briefly summarizing *conventional preference elicitation* methods, we also discuss the most important *interactive recommending* approaches in this section, before finishing with a brief overview of related research in the area of *search and information filtering*.

2.3.1 Conventional preference elicitation

For any system to be effectively informed about the user's interests and needs, an appropriate mechanism for determining user preferences—ideally in a largely automated manner—is an important prerequisite [BB09]. In *decision-support systems*, this is usually addressed by posing a small set of queries regarding the user's preferences he or she needs to answer. Subsequently, the system promotes the user to make those decisions it considers most appropriate in light of the available knowledge, i.e. the preferences expressed in this way. Acquiring enough information for this task means breaking the so-called "preference bottleneck" [BB09].

This can be considered very important in the area of recommender systems as well, key for the success of the entire recommendation process. According to Braziunas and Boutilier [BB09], the pivotal problems in this specific context are: the *enormous number of items* in tandem with their variety of properties, and the difficulty for users to *quantitatively specify their preferences* towards these properties in a way that is meaningful to the system. Pu and Chen [PC09] mention a number of related challenges for obtaining a model of user preferences that is as accurate and complete as possible: initially *motivating users* to express any preferences at all, providing them with interaction possibilities that help *revealing preferences* they may not even know about, dealing with *conflicting preferences*, and *revising preferences* articulated in the process at some later point in time. In addition, it has been found that different users call for different methods to express their preferences [KRW11], and these methods affect the user experience in different ways [KW10]. Accordingly, much work has been done in recommender research on preference

elicitation. Regardless of these efforts, *implicit or explicit feedback* provided by users for single items still is the most common type of input data [JWK14].

Implicit feedback Implicit feedback data describe, for instance, the visit of an item detail page, the duration of such a visit (dwell time), or the final purchase of a product. This form of feedback is *easy to obtain* from a system provider perspective (it is inherently known if a user bought a product or how long he or she watched a movie), and thus often available to a sufficient amount. Yet, the *interpretation may be difficult* as the preferences are derived from user interaction behavior. Here, one distinguishes between [JWK14]:

- *Examining* refers to selecting items or spending time for inspecting them.
- *Retaining* is observed when users bookmark or purchase items.
- *Referencing* happens especially in social networks, by quoting, forwarding or replying.
- *Annotating* means that users rate items or publish something about them.

The latter can also be understood as an explicit interest signal, i.e. consciously provided feedback. The first three categories, in contrast, clearly relate to behaviors users perform without the intention to indicate their preferences. Once observed, these behaviors can yet be used to get an *approximation* of their preferences [JWK14], or depending on the point of view, of the confidence in these preferences [HKV08]. Most existing approaches are limited to user-item interactions in binary form, for example, views or purchases. Even richer feedback (e.g. dwell time) is mostly reduced to a binary form [HKV08; JWK14], or at least heavily quantized [PA11]. Accordingly, user-item matrices that store implicit feedback usually contain only 1s and 0s, but are more dense than matrices that store explicit preference signals. Often, there is a correlation between these two types of feedback [Cla*01; PA11]. For these reasons, implicit feedback has received growing attention—regardless of the aforementioned simplifications. As a consequence, it is today widely accepted that this type of feedback models user behavior more accurately, showing positive effects on *objective* recommendation quality [PA11; JWK14].

Notwithstanding these advantages, many real-world examples (still) make use of explicit feedback. Academic research has for a long time been focused on this type of user-item interaction as well—and still is [SB18]. A reason might be the greater availability of freely accessible datasets, whereas implicit feedback often needs to be made up by translating explicit feedback to so-called "pseudo-implicit" feedback [Kor10]. Moreover, the demand for user control and transparency gets increasingly recognized in academia [XB07; KR12; PCH12; JJ17; Alv*19], while shifting towards implicit feedback in industry has exactly the opposite effect.

Explicit feedback Explicit feedback is elicited by actively asking users to rate items. Thus, user *preferences are directly reflected*, but it is more *difficult to establish a dataset* that is large enough and of reasonable quality for generating accurate recommendations: Users need to be motivated to provide feedback for the products they have bought or the movies they have seen, and then be able to express their preferences regarding these items in an adequate manner. This is usually possible in one of the following ways [SS11; Nob*12; JWK14]:

- *Unary* feedback allows users exclusively to express positive opinions. *Facebook* is a prominent example where users can "like" the content, but have no means to indicate reluctance.

- *Binary* feedback enables users to additionally indicate that they "do not like" an item. Prominent examples are *YouTube* and *Netflix*, where users can vote for or against an item literally by using a "thumbs up" or "thumbs down" button.
- *Ordinal* ratings allow users to express their preferences with higher granularity. The most common way is using a 5-star rating scale as known from *Amazon* or other online shops.
- *Interval* scales go even further, but are rarely seen in practice. Earlier versions of *Google News* are some of the few examples where sliders could be used to tell the system about user preferences in a very fine-grained manner.

All these variants differ in terms of usage effort and cognitive load [SS11; Nob*12; Klu*12]. For settling on one of them, understanding the needs of users and their system-specific interaction behavior is thus highly important [JWK14]. This is reflected by the examples of *YouTube*, which already shifted years ago from ordinal feedback to binary ratings,[7] and *Netflix*, which did the same only more recently:[8] Users did not understand 5-star rating scales as individually predicted scores, since they were accustomed to seeing them solely as representations of average item ratings (e.g. on *Amazon*). Moreover, they could not imagine their ratings would make an impact. In case they eventually provided ratings, they did so mostly using extreme values, leading to U-shaped rating distributions. Consequently, shifting to coarse-grained thumbs up/down ratings quickly led to explicit feedback more aligned with actual viewing behavior.

Still, rating-based feedback can generally be considered reliable in a sense that it allows differentiating *how much a user prefers an item*. There has been a tremendous amount of research on automatically determining those items that contribute the most to accurately modeling user preferences, and should thus be rated next. Solutions range from naively presenting the same popular items in a static manner to all users [Ras*02] and early attempts to active learning [BZM03], later also taking an information-theoretic perspective [RKR08], to dynamic processes that select items with the highest information gain depending on the user's previously provided feedback. We refer to the surveys on active learning for more details [Ela14; ERR14; Rub*15; ERR16].

On the other hand, ratings may not be ideal not for measuring *real preferences and long-term interests* [Pom*12; JWK14]: In general, there are numerous issues regarding labeling, scaling, and granularity [PC00; AF01]. These aspects have also been investigated in context of recommender systems [Cos*03; SS11; Klu*12; Cen*17], where they can heavily affect cognitive load and predictive value, requiring difficult trade-offs by system providers. In addition, rating-based feedback is often noisy, inaccurate and unstable [Cos*03; Ama*09; APO09; JBB11]. In other work [Loe*18], we observed that rating behavior may be affected by consumption of items, though depending on product domain and amount of information present in the user interface. Ratings should thus neither be considered the absolute truth, nor isolated from other forms of feedback that may equally add to modeling user preferences and support decision making.

Fusion of feedback types In light of the aforementioned problems, various attempts have been made for improving the quality of feedback data, for instance, by reducing noise, accounting for variance in ratings or uncertainty in implicit feedback [cf. APO09; SB18]. Integrating the different types into *unified models* can also be seen as a promising means to further improve the algorithms [JWK14], though challenging due to the differences in meaning (quantified preference

[7] `https://youtube.googleblog.com/2010/01/video-page-gets-makeover.html` (visited on July 22, 2020).
[8] `https://media.netflix.com/en/company-blog/goodbye-stars-hello-thumbs/` (visited on July 22, 2020).

vs. confidence [HKV08; JWK14]) and scaling (binary vs. ordinal) as well as the lack of combined datasets. Examples include correlation analyses of browsing behavior and ratings [Cla*01], resulting in linear mappings from one to the other [PA11; Par*11]; integration of implicit feedback in matrix factorization models [HKV08; Kor10], additionally coping with heterogeneous scaling [Liu*10]; and modeling of the recommendation process as a Markov decision problem to take both types of feedback in a sequential manner into account [MBR12]. In general, research in this regard is however limited, especially with respect to the effects from a user perspective. The achievements of the aforementioned works have *only been evaluated in offline experiments*. Using online A/B tests, Tang, Long, Chen, and Agarwal [Tan*16] at least have shown on *LinkedIn* that even simple weighted linear combinations of individually learned models may lead to promising results. Moreover, their work underlines the importance of taking into account different types of feedback in a *graded* manner: Known-item search likely indicates a much stronger interest signal than broadly filtering or browsing. In turn, implicit interactions may be seen as weaker preference indicators. In collaborative filtering, apart from temporal decaying [KBV09; KB15a], all user-item feedback data are in contrast considered *equally* important.

Summary Apparently, fusing different types of feedback, possibly at different levels, is possible even via extensions to existing methods—which of course also applies to the developments we present in this thesis. In general, our interactive methods are independent of both the type and the quality of the underlying user-item feedback data. As a consequence, we omit further details regarding the two types and their integration at this point, but refer to the survey by Jawaheer, Weller, and Kostkova [JWK14]. On the other hand, implementing preference elicitation methods based on conventional ratings already comes with a whole set of challenges itself. Consequently, various alternatives have been suggested that go beyond what we consider *standard* user-item feedback. This starts with multi-criteria recommending approaches that allow users to indicate their item-related preferences in terms of multiple aspects at the same time [AK15]. It ends with feedback that is usually considered only independently in real-world applications, for example, when users submit a query for items with a certain property, filter for broader categories, or navigate through the item space in a particular direction. Yet, we will address these alternatives in the next section, when we focus on the area of interactive recommending.

2.3.2 Interactive recommending

In light of the previously discussed drawbacks, *interactive recommending* approaches have received more and more attention in recent years. Referring back to our recommendation model shown at the beginning of this chapter in Figure 2.1, these approaches continue where most contemporary systems stop: at considering interactions performed by users once recommendations are generated. Instead, they focus specifically on *increasing the level of user control during the process*, which often also improves system transparency, and, as a consequence, the user's trust [KR12; PCH12; HPV16; JJ17; Alv*19]. When we elaborated on further use cases of matrix factorization algorithms, we have seen that only few attempts can be found in the area of model-based collaborative filtering (see Section 2.2.5). With the goal of developing interactive methods for exactly this kind of recommendation method, it thus makes sense to *provide an overview* of the broader landscape of interactive recommending approaches that have been proposed so far.

Structuring research More interactivity in recommender systems may be achieved in many different ways. This raises the question of how the range of options can be mapped out in a systematic manner. Several ways to classify and structure the different aspects that pertain to user interaction have been suggested over the years. The perspectives taken by the authors are related to cognitive activities users perform when moving from an initial intention to a final decision for an item, or to system components and their individual ability to contribute to making the recommendation process more interactive. Related to the former, several models were introduced early on in the area of information retrieval, for examining the information-seeking behavior of users [cf. Kuh91; Mar95]. Yet, these models are not directly applicable to recommender research due to their focus on document collections and explicit search tasks.

Also in the area of recommender systems, McNee, Riedl, and Konstan [MRK06] argued early on the "need [for] a deeper understanding of users and their information-seeking tasks to be able to generate better recommendations". Accordingly, Pu, Faltings, Chen, Zhang, and Viappiani [Pu*10] identified feedback cycles in four phases of the recommendation process: preference specification, recommendation generation, revision of preferences, and final decision. Focusing on basic feedback mechanisms, their model can be seen related to our recommendation model from Figure 2.1. With a similar intention, we ourselves proposed a model comprising three interaction loops, related to user interaction with: recommendations, properties of recommended items, and entire applications [LHZ15b]. Besides, more general models exist with respect to the interaction with conversational [SM03] or critique-based [CP12a] recommender systems. Again on a different level, Jameson, Willemsen, Felfernig, Gemmis, Lops, Semeraro, and Chen [Jam*15] provided an overview of how users make choices and decisions: The *ASPECT* and the *ARCADE* model help taking into account psychological patterns when designing recommender systems and applying strategies that improve user support.

Several literature reviews also make use of classification frameworks: Comprising typical components that allow for extension, the model by He, Parra, and Verbert [HPV16] is in some aspects similar to the framework we use in this thesis. Yet, the authors focus more strongly on visualizations and related aspects, whereas offering users more control at the different stages of the recommendation process plays only a subordinate role. Jugovac and Jannach [JJ17] make a distinction between, on the one hand, (initial) preference elicitation, on the other hand, approaches that aim at increasing control, providing explanations, and visualizing recommendations once they are generated. Then, they structure the overview based on concrete methods. Alvarado Rodriguez, Vanden Abeele, Geerts, and Verbert [Alv*19] take another perspective: Their framework is derived based on an extensive qualitative user study with *Netflix*. It allows assessing requirements for recommender systems and describes design areas ranging from algorithmic transparency to algorithmic awareness. Also, user control is included as one of the opportunities for which guidelines are presented to improve the systems accordingly. Finally, based on focus groups conducted in the news domain, Harambam, Bountouridis, Makhortykh, and Hoboken [Har*19] examine the value of interaction mechanisms at three different stages identified in prior research: manipulating the input, controlling the process, and filtering the output.

Our framework With the practical purpose of providing users a holistic experience, it seems equally meaningful to *focus on the different components* that come into play in the course of the recommendation process. To describe the range of interaction possibilities users can be provided with at the different stages of this process, the framework we originally published in [LBZ16]

takes a broader perspective. As shown in the updated version in Figure 2.5, the model comprises five components: ■ *input data* related to users and items, the ■ *recommender* itself, the ■ *user model* representing user preferences and characteristics, the external ■ *context model* representing aspects of the user's situation, and the system's ■ *presentation layer*.

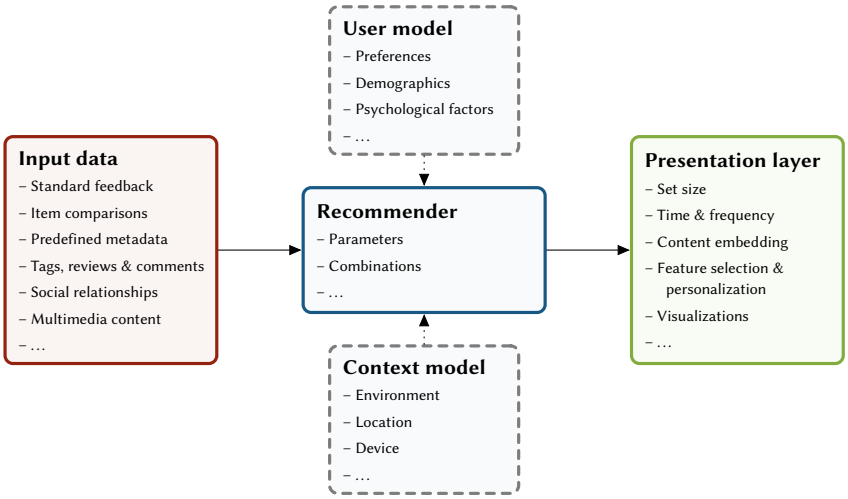

Figure 2.5 Framework for interactive recommending, delineating components that may be extended in order to provide users with additional means for interaction.

The model is aligned with the user-oriented recommendation model in Figure 2.1, but arranged slightly differently due to its alternative purpose and the system-oriented perspective: Arrows visualize the data flow,[9] starting with the selection of appropriate input data for the algorithm of the central recommender component (or the combination of multiple algorithms in case of ensemble methods or hybrid configurations), including possibly preprocessing steps. Once recommendations are generated, the presentation layer is updated, at least with a new recommendation set. For this, the recommender may exploit both user and context data. The user model consists of information provided by the current user (and the user community), whereas the context model is mainly fed by external hardware or software sensors.

For each component, a (non-exhaustive) list of properties that characterize this part of the system is presented. These properties also represent aspects that may be targeted for improving user control and experience. The fact that contemporary recommender systems have severe deficiencies in these regards, particularly those employed in (commercial) real-world settings, is reflected by the lack of attention many of these aspects received from the research community. In the following, we provide an overview of the interactive recommending approaches that are different in that they actually contribute to closing these gaps. We structure this literature review based on the model components. Note that some works may fall in one or the other category. In these cases, we focus on the contributions that are most closely related to this thesis.

2.3.2.1 Input data

The typical input for recommender systems, i.e. data related to users and items, cannot only be used by the algorithms for generating recommendations. We argue that this kind of *input data*

[9] Interaction data collected at the presentation layer are of course used as input data and may also be fed into user or context model. However, for the sake of simplicity, we omit outgoing edges of the last component.

may also contribute to implementing mechanisms that let users influence the system's outcome and improve their understanding. While we addressed the usage of implicitly observed behavioral data and explicitly provided item ratings in the previous section, the examples we discuss in the following are a level above these standard user-item feedback data.

Dialog-based and multi-criteria recommendation Going more into detail, it makes sense to start with *dialog-based recommenders*. Mahmood and Ricci [MR09] present an early example in which users are asked a series of questions regarding their search goal in order to elicit their preferences. However, to generate recommendations, this requires prior modeling of item characteristics and dialog structure, making such an approach costly to develop and limited in flexibility. The same applies to dialog-based *product advisors* as they are used increasingly in online shops [cf. KZ19]. As a consequence, Hurrell and Smeaton [HS13] treat the recommendation process as a conversation between user and system based on collaborative filtering. For this *conversational approach*, the authors use item popularity and average ratings in order to continuously confront users with decisions between two items. This way, their *MovieQuiz* application constrains the results to the desired part of the item space. Evaluated in a user study with a basic questionnaire, the authors showed that participants preferred the interactive preference elicitation over a typical recommendation list. Participants had no difficulties in making their choice, noteworthy without requiring domain knowledge on their part, or content information on part of the system. Apparently, engaging with users in style of a conversation helps to gather the necessary data with less effort. Furthermore, the authors argue that their approach has the potential to improve overall user satisfaction, in particular, due to the transparent manner in which the conversation takes place. On the other hand, the limited possibilities to manipulate the results might have negatively affected recommendation quality. Lacking appropriate baselines, these aspects would have needed further empirical backing.

Likewise based on item-related information, *multi-criteria recommenders* extend the usage of user-item interaction data in a different direction: users cannot only provide one-dimensional feedback regarding the overall quality of an item, but express their preferences on a number of dimensions, each referring to a single *predefined metadata* attribute [AK15]. However, with the main intention of increasing expressiveness to account for more complex preferences, this does not necessarily make recommender systems more interactive. In addition, we deem it more promising to account for such preferences by letting users interactively manipulate the results *at runtime*—also to consider possibly evolving situational preferences. For this reason, we omit further details on multi-criteria approaches, in particular, as we focus on product domains of lower complexity, as typical for collaborative filtering. Besides, many approaches discussed in the literature can be seen as extensions that may equally be applied to systems that integrate our interactive methods. For example, factorization techniques exist that can take multiple criteria into account [cf. LWG08]. Nevertheless, it is worth noting that the underlying principle has paved the way for the increasingly popular technique of critiquing recommendations.

Critique-based recommendation Accordingly, *example critiquing* is one of the most prominent interactive recommending techniques [CP12a], allowing users to iteratively apply critiques to the items shown as recommendations (i.e. the examples). This may be done in terms of one or more characteristics users wish to value higher or lower, resulting in requests for items that are still similar, but fulfill these characteristics to a stronger or weaker degree. Starting from a sug-

gested product that to some extent already appears sufficient, users may indicate preferences for cheaper products, products by other manufacturers, or of a different color. In line with research on information-seeking behavior (see Section 2.3.3 on information filtering), this relies on the assumption that critiquing results is easier for users than forming and expressing a search goal up front, i.e. without any cognitive clues. Whereas users typically have many requirements, they are not able to state them (in advance) unless the presented solutions (i.e. here recommendations) violate these "latent" preferences, especially in case of little domain knowledge [Xie*18].

The first *unit critiquing* system, i.e. which only allowed to critique a single item property at a time, was the *FindMe* system. It provided rather simple, completely predefined critique options, with results exclusively based on tweaking the currently shown example [BHY97]. *Compound critiquing* offers users the possibility to provide feedback simultaneously on multiple dimensions [Rei*05]. Models containing all critiques expressed by the user in the current session [VFP06] and critique options dynamically derived based on the remaining items [McC*04] further facilitate the critiquing process. Moreover, visual representation of critique options and direct manipulation mechanisms may improve comprehensibility and usability while reducing the interaction effort [ZJP08]. Mixed-initiative approaches stimulate users in expressing their preferences by showing certain recommendations specifically for this purpose, which avoids that decision making is driven too much by anchoring effects [VPF08].

Regardless of these advances, *critiquing processes usually depend on metadata* that needs to be modeled a priori. The *MovieTuner* by Vig, Sen, and Riedl [VSR12], which can still be found on the *MovieLens* platform,[10] constitutes an exception that uses a large set of *tags generated by the user community* (see Figure 2.6). In general, tags have shown their potential before [cf. Gre*09; SVR09; Gua*10]. For instance, in *Music Explaura*, tags are provided as a means to interact with the system, displayed for each item in the form of a tag cloud: By increasing or decreasing the size of a tag, users can adjust the vector of the respective item, leading to recommendations (based on standard information retrieval methods) that are related to this updated vector [Gre*09]. Decoupled from individual items, Vig, Sen, and Riedl [VSR10] introduced the so-called *Tag Genome* for the interaction with *MovieTuner*, consisting of vectors computed based on the whole tag dataset. These vectors describe the relevance of all tags concerning the items. Automatically derived, this knowledge base does not require any expert knowledge, but contains terms that are inherently meaningful because they are in the language of the users. As a consequence, the most *descriptive* or *discriminative* tags can easily be determined and suggested to the active user as critique options (see [VSR11] for a user-centric comparison of these two algorithmic variants). Moreover, similar items can be identified using these vectors. As soon as a user applies a critique, the critique satisfaction can be calculated based on the critique distance, i.e. the difference of the items along the critique dimensions (see again [VSR11] for the different models proposed for this purpose). This weights the corresponding tags so that they are more or less strongly represented by the items that are subsequently shown as recommendations. In a user study, participants enjoyed being able to manipulate the recommendations in this way, though some of them indicated that they would have preferred a more-fined grained adjustment [VSR11].

The independence of expensive predefined metadata can be considered a major benefit. However, although less restrictive, fulfilling the requirements in relation to user-generated data also entails non-negligible efforts. Moreover, when implemented as a standalone solution—which

[10] `https://movielens.org/`

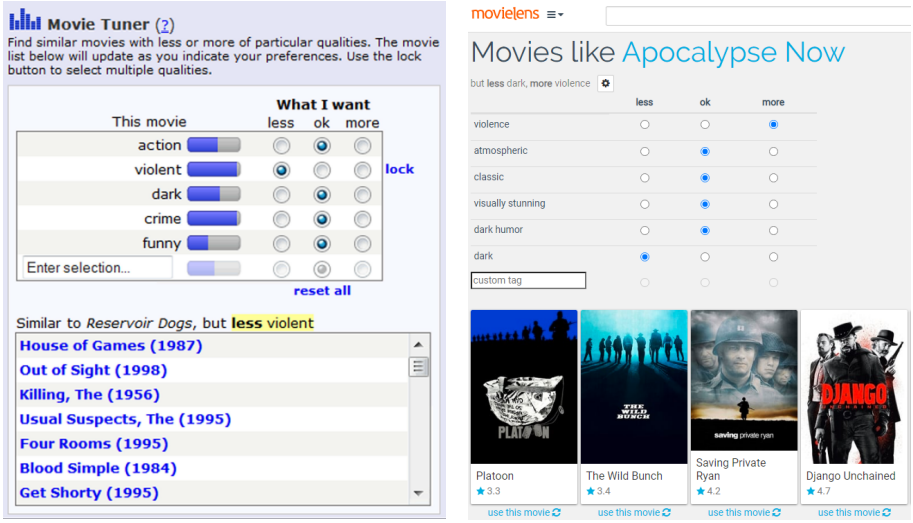

Figure 2.6 Screenshots of *MovieTuner*: The initially proposed variant (left) [VSR12], and the current variant on the *MovieLens* platform[10] (right), both showing that a user applied critiques to a recommended movie in order to receive new recommendations of similar items that represent these tags "less" or "more".

usually is the case—the focus on content information makes it difficult to gear the critiquing process towards the *current user*. For this, only his or her own critiquing behavior has been taken into account, i.e. items this user accepted in the past [MSS10; MF12]. In one of the most recent works on critiquing, a graph-based representation of user sessions is used. In contrast to earlier works, this allows to consider similarity and compatibility of these accepted items. A simulation study based on real-world datasets confirmed the effectiveness in reducing the number of rounds in the critiquing process [Xie*18]. Still, all these approaches have in common that *profiles containing long-term preferences* regarding the items, as they are common in collaborative filtering, are *neither considered for adapting the process* and the presented critique dimensions, *nor do they affect the recommendations* resulting from this process.

Choice-based preference elicitation While critiquing mainly supports users *during* the recommendation process, much effort has been spent on helping users *at the beginning* of this process. Active learning techniques can very well alleviate the problems of rating-based mechanisms in new user cold-start situations [cf. Rub*15; ERR16]. Moreover, we already discussed algorithmic solutions specifically for improving initial preference elicitation in model-based collaborative filtering systems (see Section 2.2.5.1). Also, interactive solutions have been proposed: Chang, Harper, and Terveen [CHT15] let users pick the most suitable ones from *groups* of similar movies, i.e. in contrast to the usual case, the user input is related to more than one item. Neidhardt, Schuster, Seyfang, and Werthner [Nei*14] present *pictures* related to touristic behavioral patterns and generate recommendations depending on the user's selection of these pictures. Most importantly, however, *comparisons of items* have shown their potential already early in the area of information retrieval [cf. Car*08], but also as input to recommender systems: Jones, Brun, and Boyer [JBB11] introduced them as an alternative to standard user-item feedback, showing in a user study the potential for facilitating decision making and obtaining more stable preferences.

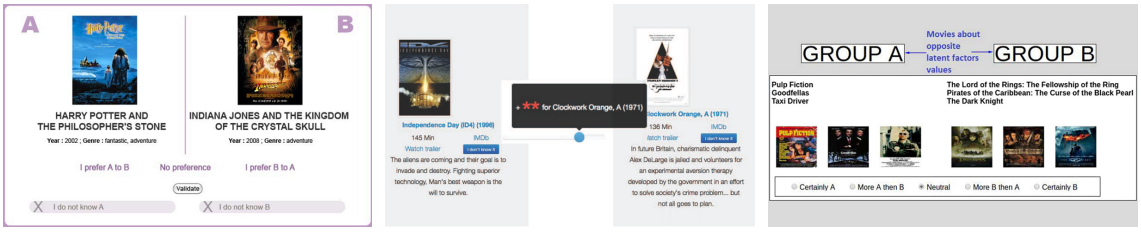

Figure 2.7 Screenshots of interfaces for item comparisons, ordered by year of publication: a binary choice interface introduced as a first alternative to interfaces using rating-based feedback (left) [JBB11]; a similar interface that allows to express preferences in a more fine-grained manner (middle) [BR15]; and a more recent variant that in turn reduces the granularity, but juxtaposes groups of items (right) [Ros*17].

Rokach and Kisilevich [RK12] for the first time proposed such a preference elicitation method *specifically* for matrix factorization algorithms (apart from earlier attempts that just adapted the objective function for pairwise learning, see Section 2.2.2.2). Concretely, they first create a judgment matrix containing pairwise comparisons of all items based on the user-item matrix. To address scalability and to obtain representative items for the comparisons users are confronted with, they suggest a clustering based on latent factor vectors. In each of the corresponding dialog steps, users are then asked to indicate their preference regarding one of these items (or that both are unknown). The authors consider their approach as a promising alternative to typical active learning strategies, as the interaction effects between items are inherently captured by the similarities in the factor space. Such dependencies are usually ignored, which leads to queries for items that contribute only little to accurately modeling user preferences.

Graus and Willemsen [GW15] and Liu, Han, Iserman, and Jin [Liu*18] follow a similar idea as we do in this thesis, namely letting users choose between items (or groups of items) that *directly represent the factors* of a matrix factorization model (see Chapter 4). This enables users to manipulate their user-factor vector, and thus steer the recommendations into any desired direction. Taijala, Willemsen, and Konstan [TWK18] make use of the same approach for implementing a stepwise navigation. Since these approaches directly exploit the characteristics of latent factor models, they have already been part of the discussion of further use cases of matrix factorization algorithms (see Section 2.2.5.1). On the other hand, since users in these approaches can manipulate their usually opaque representation within the user model, they could also be listed below, where we discuss this part of our framework (see Section 2.3.2.3).

Yet, pairwise preference elicitation attracted further attention in relation to matrix factorization algorithms. One of the later examples is the work by Sepliarskaia, Kiseleva, Radlinski, and Rijke [Sep*18], who propose optimizing a different target, namely the expectation of misclassified preferences for one item over another. Although the authors introduce their method as a novel preference elicitation method, in fact, they "only" describe a new technique for coming up with a static set of interview questions. These questions are the same for all users (due to performance concerns), each aiming at finding out which of two items is preferred. Similar to other, more recent approaches [e.g. CHR16; KRT16], this makes this work interesting from a system perspective, showing how to improve the conversation between users and system by asking questions that are more informative for the algorithm. However, the *user perspective is largely overlooked.* For instance, regarding the dialog design, only specific issues have yet been

addressed, such as more fine-grained feedback mechanisms that allow users to indicate on a 5-point bipolar scale or by means of slider widgets the degree to which they tend to one of the juxtaposed (sets of) items [BR15; Ros*16; Ros*17]. The evolution of the corresponding interfaces is illustrated in Figure 2.7. Also, apart from few exceptions [BR15; KRG18], *user-centric evaluation is still uncommon*, although necessary for confirming the effects of comparisons in terms of subjective recommendation quality, interaction with the systems, and user experience.

Summary It appears valid to say that all aforementioned approaches, which exploit the input data in other ways than typical recommender systems that are based on standard user-item feedback, have at least shown potential in motivating users to express their preferences and in keeping them engaged. Data sparsity may thus be reduced while the quality of the results increases in relation to the amount of feedback users provide. However, we also identified a number of deficiencies, most importantly, the independence of collaborative filtering, but also the limited consideration of user-oriented aspects in pairwise preference elicitation approaches that are actually implemented on top of state-of-the-art matrix factorization algorithms. On the other hand, it is worth mentioning that there are some "extreme" cases in which explicitly entered queries serve as additional input data: Dzyabura and Tuzhilin [DT13] combine a search function for desired attributes with a recommendation model in a hybrid configuration. Kveton and Berkovsky [KB15b] pose questions with respect to item properties in order to let users more quickly find the target item within already generated lists of recommendations. Later in this chapter, we detail on related approaches that fully belong to the area of information filtering, but integrate recommendation methods only as a by-product (see Section 2.3.3).

2.3.2.2 Recommender

After having seen that there are numerous alternatives to using input data such as implicit or explicit feedback in a conventional manner, and to add interactivity in this way, we now continue with the four remaining components of our framework. We start with the *recommender* itself: Also the used algorithms offer possibilities to integrate advanced interactive features into the systems. These features may include modifiers for *parameters* of the algorithms such as item popularity and recency [Har*15], but also weighting mechanisms for adjusting the influence of selected keywords on content-based techniques [SSV16] or options to let users choose from a range of different algorithms [Eks*15]. Moreover, systems have been proposed that allow to manipulate the *combination* of algorithms in hybrid scenarios.

TasteWeights by Bostandjiev, O'Donovan, and Höllerer [BOH12] is one of these hybrid systems, with methods based on social (*Facebook*), content (*Wikipedia*), and expert (*Twitter*) data. Sliders allow users to control the individual influence of preferred artists as well as of datasources and associated entities on music recommendations. At the same time, the connections between user profile, datasources, and results are highlighted. A user experiment showed that this process of generating recommendations was perceived as more transparent, and that the visualization helped participants to better understand the system's behavior. The additional interaction capabilities resulted in a considerable gain in perceived recommendation quality and overall satisfaction. Similar results were obtained with several successor systems [cf. BOH13; SHO15].

With *SetFusion*, Parra, Brusilovsky, and Trattner [PBT14] also illustrate how a system that uses a common hybridization strategy can be influenced by users, and how the results of multiple

algorithms can effectively be visualized. The system is designed for the use on conferences and recommends research papers. Again, sliders are provided to individually weight the implemented algorithms. However, changes to this configuration are not directly represented in the result list. In addition, no further interaction is possible, for example, to refine this list. Nevertheless, a user experiment acknowledged a high degree of control. The corresponding visualization in the form of a Venn diagram, which revealed the sources of the entries of the result list (i.e. the algorithms responsible for the recommendations), had a positive effect on transparency. Beyond that, promising findings were reported regarding the engagement of participants, whereas perceived recommendation quality had not been investigated.

TalkExplorer is another research paper recommender. Here, instead of sliders and a Venn diagram, Verbert, Parra, and Brusilovsky [VPB16] use lists of checkboxes (similar to faceted filtering, see Section 2.3.3) and a cluster map. Thus, users should better understand how the different algorithms contribute to the results, for which they can switch these "recommender agents" on or off. This always results in an immediate update of the cluster map that visualizes the relations of the current user to tags and other users, and allows to explore the connections of suggested conference talks to bookmarks and additional social data. A user study confirmed that participants were more active when they had the possibility to use the advanced visualization and the corresponding selection mechanisms. On the other hand, participants who were less tech-savvy needed additional guidance to understand the rationale behind the intersections of recommendation sets that can occur when selecting and deselecting the different algorithms.

In Section 2.3.2.5, we will briefly address related examples that more extensively use information visualization techniques, and put their emphasis on discovery and exploration rather than possibilities to intervene in the algorithms or affect their interplay.

2.3.2.3 User model

The quality of *user models*, which are typically learned as a consequence of the user interaction happening at the front-end, not only is a critical determinant for recommendation accuracy. We argue that it is possible to "intervene in these models" to let users adapt the system's output and to improve their understanding of the system's behavior. If user *preferences* are not only used ad hoc (as in many critique-based systems), but added to a user model (as in case of collaborative filtering), approaches that allow users to manipulate this representation within the model could be listed here. But, we already discussed these approaches in Section 2.3.2.1. Other approaches focus on visualizations, often to provide users insights into their representation. If these approaches include additional interaction mechanisms, they are in turn described later in Section 2.3.2.5.

For this reason, we can here only point out the remaining potential, for instance, when using other methods than collaborative filtering. In content-based filtering, the user model typically is the result of probabilistic or nearest neighbor methods based on items the user has seen, liked, or purchased, i.e. what can be considered as his or her preferences [Gem*15]. Yet, *demographics* [Bob*13] or *psychological factors* [NH12; TC15] can be taken into account as well. For the latter, preference elicitation may take place in the form of personality quizzes or by using gamification [cf. HP09; Tek*16]. However, in all these cases, the *only way to influence the results* once they are generated, and thus, to refine the underlying user model, is again by *giving some kind of relevance feedback* as known from information retrieval [SB97; Gem*15]. Beyond that, many more aspects make this component an interesting subject for further research. For example, privacy concerns

suggest that users should be able to select themselves the information that is stored in the user model [Fri*15]. But, it still needs to be investigated how to support users most effectively in this regard. Moreover, mediating user models, i.e. importing them from other systems [BKR08], seems promising for providing cross-domain recommendations [Can*15]. However, this kind of transfer learning would likely benefit from improving interactivity as well.

2.3.2.4 Context model

Beyond what is represented by the user model, also the context may affect whether an item constitutes an appropriate suggestion at the present time. Yet, although many approaches to context-aware recommending have been proposed [AT15], date and time, weather, company of other people as well as other aspects of the user's current situation, are often disregarded in real-world recommender systems. In fact, many systems do not even integrate a *context model*. Worse, they often make no distinction between long-term and short-term preferences, neglecting that the latter are closely connected with the user's context [Ela*15].

Nevertheless, variables indicated by the user's external context have already been taken into account many times: Accessing the user's *location* led to restaurant or travel recommenders, but also music recommenders specialized for different purposes (in car vs. at the gym) [AT15]. With the advent of handhelds and smartphones, not only increased the interest in adapting the presentation of recommendations according to the currently used *device* [Ard*03], but also in *mobile* context-aware recommender systems. One can easily imagine that it would thus be useful to add interactive features to cope with the higher complexity, and the difficulty of obtaining all necessary information about the user's context. However, as is generally the case in recommender research, *most work has been done on part of the algorithms* so far (details can be found in the survey by Adomavicius and Tuzhilin [AT15]). In contrast, only little attention has been paid to increasing user control: Loosely related, some conversational systems adapt the order of dialog steps implicitly based on interaction sequences [MR09]. Other approaches try to capture changes in the user's interests [HMB14]. Similar to the user model, the context model can also be influenced more directly, for instance, to refine contextual factors by filtering out items that do not suit the current situation [AT15]. Visualizations have been proposed to account for concept drifts in the user's search goal, allowing to revise true positive and mark false positive recommendations [Kan*16]. However, the contextual information is mostly derived automatically [AT15]. If users are allowed to manually specify this information, this requires doing *everything* manually up front [cf. CMO15; NWB16]. Whereas some approaches use contextual information to explain recommendations [NWB16; Hie*16], the system presented by Baltrunas, Ludwig, Peer, and Ricci [Bal*11] is one of the few exceptions in which users can actually decide which contextual factors should be considered in the process of generating recommendations. Yet, this is limited to switching them on or off. Accordingly, coming up with improvements for this part of our framework seems to be another fruitful area for future research: Mechanisms are required that enable users to directly intervene in the context model, or at least to *adjust recommendations according to situational needs* at any time.

2.3.2.5 Presentation layer

Overall, the *presentation of recommendations* in the user interface has received relatively little attention. Basic questions have been addressed, such as: what is the ideal recommendation *set*

size [Bol*10], at which point in *time* should recommendations be presented [DSR15], how helpful is it to embed additional *content* such as images [NLF10], which *features* should be shown alongside recommended items concerning the user's interests [BZ17] and context [Ard*03; NWB16]. In addition, there are some more sophisticated approaches inspired by *information visualization techniques* as they are comprehensively reviewed in [Ker*08; HBO10]. However, as we will see below, their underlying goal is mainly improving transparency, *without considering interactivity a major factor*. While only partly within the scope of this thesis, we still provide a brief overview of these approaches, but refer to the literature for more details [HPV16].

Explanations Several times throughout this thesis, we highlighted the importance of explanations, especially given the black-box characteristics that increasingly come to light due to the higher complexity of state-of-the-art model-based algorithms. At the same time, system-generated explanations still appear mostly of inferior quality when compared to explanations made by humans [Kun*19a]. This is especially true in practice, where mainly simple *textual variants* can be found [TM15]. These include the well-known "other customers also bought …" explanations provided by *Amazon* for collaborative filtering data [LSY03; SL17], and explanations that highlight what is preferred by friends or relatives based on graph-based social network data [SC13]. Visualizations are more rare, although they appear particularly promising for improving the user's understanding of the system and its output. This becomes visible even with early variants that just depict the rating distribution among similar users [HKR00]. Some approaches are located at the intersection of textual and visual components, using *line charts* to visualize the user's predicted interest in tags [BJG13] or *tag clouds* to present the top n tags in relation to their individual importance [Gre*09; GGJ11]. Preparing textual data in these ways turned out superior to presenting plain lists of tags, such as in the work by Vig, Sen, and Riedl [VSR09]. More complex variants include flow charts [Jin*16] or Sankey diagrams [BZ18]. Yet, providing such explicit explanations is only one of many cases in which visualizations may contribute to user experience. Nevertheless, fostering transparency and explainability also plays a role in the other cases, which we address in the following.

Item space visualizations In the area of information retrieval, a wide range of methods exists for visualizing large-scale document collections [cf. AS94; Shn94; Hea09; ST09; AB13]. *Map-based visualizations* have been shown particularly useful for facilitating the browsing and exploration process. In recommender systems, maps may be equally helpful for visualizing the space of available items (but also the user's preferences and the system's results, see below). Users can be made aware of regions of the item space they would not have considered otherwise, or retrace their own search behavior. This can in turn increase transparency of the recommendations and engagement of users when interacting with the systems [Gan*09; Far*10].

For preference profiles stored as high-dimensional vectors, dimensionality reduction techniques have been used successfully to project them onto a two-dimensional item space representation. Items with highly positive rating predictions are then positioned close to the point that represents the respective user within the underlying user model [cf. KS10; Far*10; Moi14]. Similarly, Andjelkovic, Parra, and O'Donovan [APO16] in their *MoodPlay* system use a mapping of both artists and moods into a common information space in which the current user is represented by an avatar. A user study showed that participants were able to better understand the reasons songs were recommended by inspecting the position of this avatar in the latent space. However,

for maps of this kind, items need to be arranged specifically for each user. Consequently, they cannot be generated without a sufficient amount of information about the current user. Moreover, showing distances inversely proportional to predicted ratings usually requires an adapted version of the underlying recommendation algorithm (see Section 2.2.5.2).

Conventional systems often make it difficult to grasp which alternatives are available due to their use of plain recommendations lists. The aforementioned approaches that visualize *only a part* of the item space may also be prone to this problem. Several attempts have thus been made to reduce the lack of diversity and to mitigate the risk of users becoming stuck in a filter bubble. Often taking up the bubble metaphor, visualizations have been proposed that show what like-minded people prefer in comparison to others [e.g. Won*11; NV14; WV14]. Beyond that, *TVLand* by Gansner, Hu, Kobourov, and Volinsky [Gan*09], but also our own approach presented in related work [KLZ17], visualize the *whole* item space based on *standard* matrix factorization. This way, it can still be shown how the current user is positioned in relation to recommended items. But, an overview is also provided that exhibits how these items are positioned in relation to all remaining items. However, we have already discussed visualizations based on matrix factorization algorithms in Section 2.2.5.3. Accordingly, it is only worth mentioning here that, apart from our own work, users typically cannot control the recommendations directly from within the map.

Presentation of preferences and results In addition to map-based approaches, which often also depict the user's preference profile, other approaches that may be used for this purpose are focus-and-context lists [UK03], icon-based avatars [Bog*13], hyperbolic and multi-modal views [LFK08] or graph embeddings [VS12; VS13]. Only few exceptions allow users modifying their profile, for instance, by moving keywords over a radial display that visualizes the user's representation within the user model [Bak*13; Kan*16], or by selecting nodes in a network visualization of this model [Crn*11]. When it comes to the presentation of the system's results, possible visualizations again include some of the map-based approaches, but also Venn diagrams or cluster maps (see the description of *SetFusion* and *TalkExplorer* in Section 2.3.2.2). Also in this context, only few systems have been proposed that use *interactive* visualization techniques: *uRank* by Sciascio, Sabol, and Veas [SSV16] is a research paper recommender that introduces *stacked bar charts* as a means to indicate the relevance of items in relation to selected keywords. The system allows to weight the influence of these keywords on the underlying content-based recommendation method by means of sliders. *IntersectionExplorer* by Cardoso, Sedrakyan, Gutiérrez, Parra, Brusilovsky, and Verbert [Car*19] builds on the success of *SetFusion* and *TalkExplorer*, but uses a complex *set-based matrix visualization*. Thus, users can explore the intersections of the result sets generated by the underlying methods, see immediately which methods are responsible for certain results, and how many have considered an item to be relevant.

Beyond that, several *graph-based approaches* have emerged. These approaches can be applied directly on standard collaborative filtering data without reducing the number of dimensions. *PeerChooser* [ODo*08] and *SmallWorlds* [Gre*10] are examples that explain the output of memory-based collaborative filtering by showing the active user's neighbors as connected nodes, with distances that reflect their similarity. In a user experiment with *SmallWorlds*, a music artist recommender implemented by Gretarsson, O'Donovan, Bostandjiev, Hall, and Höllerer [Gre*10] based on *Facebook*, the collaborative filtering process was thus easier to understand and participants were more satisfied. Being able to adjust the importance of the mentors by moving

the associated nodes, hereby changing their weights for the rating prediction task, additionally contributed to system transparency and user satisfaction.

Summary Overall, visualizations are still underrepresented in the area of recommender systems. Up until today, most visual recommendation approaches are standalone solutions, not being integrated into the user's typical browsing and exploration process. Moreover, the literature review has shown that even in the examples presented in this section, the interaction with the systems rarely goes beyond what is already possible in established (commercial) systems. This again underlines the need to *improve user interaction* in contemporary recommender systems.

2.3.3 Search and information filtering

While recommender systems can be helpful tools, there is a broad range of alternatives that may equally lead users to suitable items. In certain situations, *search and information filtering* methods may be more effective, with complementary advantages: Being on the other end of the spectrum than conventional recommender systems (cf. Figure 1.2), the approaches proposed in context of information retrieval research are usually highly *controllable*, and thus inherently *transparent*, since they behave exactly as indicated by the user's actions. In the previous section, we have seen how these two aspects are increasingly taken into consideration by the recommender research community to eliminate the weaknesses of recommender systems in these regards. On the other hand, also the disadvantages are complementary: The *interaction effort* is naturally higher, the search and filtering process cognitively more demanding due to the lack of *personalization*. Users are required to know their search goal at least to some extent, and to be able to express the corresponding information need given the options offered by the system. Since users typically start with an ambiguous information need that evolves as long as new information is picked up [Bat89], this is often a more severe problem. As one of the consequences, users often misuse keyword-based search mechanisms for orientation purposes, instead of directly targeting their search goal [Tee*04]. Thus, it seems necessary to support users in making something out of the growing amount of information they gather, without forcing them to use existing mechanisms in different ways than intended. This is of particular importance in domains that are large and unknown or contain experience products—factors that make it more difficult to mentally form a search goal, i.e. where automated recommender systems can play out their strengths.

Accordingly, extended search mechanisms, hierarchical navigation aids as well as advanced methods for browsing and filtering large result sets have been proposed. For example, the *FilmFinder* by Shneiderman [Shn94] established the concept of dynamic queries already years ago, supporting continuous manipulation of filter settings while providing users with immediate feedback within a visualization of the item space. However, this and subsequent attempts to make the systems in this area more interactive—which from a human-computer interaction perspective are often still limited—are outside the scope of this thesis. Thus, we refer to the literature for more details [Tun09; Hea09; ST09; Wei*13]. Nevertheless, we take a look at one of the most successful information filtering techniques, *faceted filtering*, as we consider this technique as a point of departure for one of the methods we present in this thesis.

Introduction to faceted filtering For the application of faceted filtering, also known as faceted search or navigation, the entire item set is classified according to a number of dimensions that represent certain properties of the items. For this, a single source of data is usually exploited,

either unstructured texts or predefined taxonomies. The resulting classification is then translated into *facets* and *facet values* which are presented in the user interface as filter criteria, allowing users to indicate their preferences with respect to these properties. This constrains the results in a stepwise manner to items whose properties match the selected values. This way, faceted filtering allows (even non-expert) users to explore item sets of nearly any size and discover items relevant with respect to their current needs. At the same time, it may help users to understand the structure of the item space and to become aware of alternatives that do not match the currently specified criteria [Yee*03; Tun09; Hea09; ST09; Wei*13]. Providing more effective information-seeking support than conventional search mechanisms [cf. Yee*03; Dir12; NH14], faceted filtering is often used in digital libraries or online shops as an additional aid that offers users a more flexible browsing experience (see the *Amazon* screenshot in Figure 7.1).

Limitations of early research Yet, as often in information filtering, the filter criteria are mostly predefined, fixed, and a hard Boolean filtering logic is implemented based on conjunctive queries. All criteria are hereby considered with equal importance, and an exact matching with the properties of the items is performed. Due to this strict logical, very restrictive query processing, users may quickly over-constrain their search, which makes retracing the effects of their actions more difficult, especially in case the facet values they select are mutually exclusive [cf. Sac06; Tva*08; TRH12]. In general, many manual exploration techniques suffer from the system-related drawback of high data requirements, especially in comparison to the methods used in automated recommender systems [TDG08]. Usually, this includes an expensive data aggregation and extraction process as well as a transformation into a structured format [WS12]. Moreover, it has to be determined which facets and facet values to present to the user, and how to preview their effects [TDG08].

Improvements in example systems Accordingly, more recent approaches often exploit alternative datasources to automatically extract facets and facet values, and implement adaptive techniques to facilitate the user's task of achieving a meaningful selection of filter criteria [TB07; DI08; Tva*08; Li*10; CAS11]. *DocuBrowse* by Girgensohn, Shipman, Chen, and Wilcox [Gir*10] supports faceted browsing of large document collections based on automatically identified genres. To deal with missing metadata, the authors propose to exploit the file hierarchy. In this case an effective solution, this highly depends on the specific item type. Still, the system is one of the exceptions that apply fuzzy techniques to deal with misspellings and similar but not identical values. Consequently, when users select a value, relevance of documents is indicated by colors. Recommendations are offered as well, but without any integration into the filtering process.

RevMiner by Huang, Etzioni, Zettlemoyer, Clark, and Lee [Hua*12] is one of the examples that use a *social datasource*: The system extracts attribute-value pairs from user-written restaurant reviews (e.g. "delicious pizza"), associates each value with a sentiment score, and groups attributes that belong together. Eventually, these attributes are presented in the form of facets and facet values. When users select these values, the results are ranked according to sentiment, strength and frequency. Again, some recommendation functionality is implemented, but only for suggesting places with similar attributes. To further reduce the data requirements, Koren, Zhang, and Liu [KZL08] adopt the idea of collaborative filtering. They propose a personalized mechanism that automatically selects facets and facet values according to *explicitly provided user feedback*. Hussein and Münter [HM10] use *semantic models* for creating a faceted navigation, entirely inde-

pendent of content and model structure. In all these approaches, however, content attributes of some kind are still used for presenting the facets in the user interface. Moreover, the possibilities to influence the current filter setting remain very limited.

Addressing some of these challenges, *VizBoard* by Voigt, Werstler, Polowinski, and Meißner [Voi*12] suggests not only facets and facet values, but enables users to prioritize selected criteria. Thus, users can order the results according to their own situational needs while receiving support to avoid the exclusion of relevant items. Thai, Rouille, and Handschuh [TRH12] focus even more strongly on user experience: Their *IVEA* system uses a multi-dimensional matrix visualization that displays documents and their relevance according to the user's selection based on a TF-IDF heuristic [BR99]. Users can sort the underlying facets, which are derived from a user-built ontology. To allow for the comparison of document sets in large collections, relevance values are then shown in the interface. However, they are not used for establishing a personalized ranking of the results.

Summary Recent research has produced a range of promising search and information filtering methods. However, methods such as faceted filtering have thus far not been extensively considered for combination with recommendation methods. This is particularly inexplicable since faceted filtering has been shown useful for complex tasks in which users target exploration instead of pursuing a functional goal [NH14; KFK14], but fails at providing the same level of support in many *other* situations. Of course, a few examples exist at the intersection of information filtering and recommender systems, such as the approaches discussed in Section 2.3.2.2 and 2.3.2.5. However, state-of-the-art model-based collaborative filtering techniques only play a minor role in these approaches. Worse, in real-world scenarios, filtering and recommending usually happens *entirely* independent of each other, as illustrated by Figure 1.2 at the beginning of this thesis. This shows that further research is required to come up with more *holistic solutions*, which always provide users the *full range of options* they need to reach their search goal.

CHAPTER 3

"The organization of information actually creates new information."

— *Richard S. Wurman, American architect and designer*

Methods for interactive model-based collaborative filtering systems

The literature review has shown that contemporary recommendation methods have significant deficiencies. Highly accurate and efficient, model-based systems in particular lack mechanisms for users to control the recommendations. On the other hand, there exists a range of approaches for increasing the interactivity of recommender systems. However, these approaches are usually independent of established collaborative filtering techniques. In this thesis, we aim at closing this gap by introducing interactive methods for model-based collaborative filtering recommender systems. Before addressing these contributions in detail, it first makes sense to take a theoretically informed look at this gap and ways to close it. For this, we begin this chapter with a *model* of user interaction. We use this model as a foundation for summarizing the problems identified in the literature review and proposing enhancements for each phase of the recommendation process: to provide users control on different levels, depending on individual needs, and in accordance with the current situation. From this point of departure, we elaborate on the consecutive nature of the *research questions* we have posed at the beginning of this thesis, and relate the proposed model to specific approaches for getting closer to our overarching goal. Finally, we give an overview of how these approaches are reflected in the actual contributions, thereby providing an *advance organizer* for the remainder of this thesis.

3.1 Model of user interaction

In personalized recommender systems, the interaction performed by the ■ *user* serves as input data, as it can already be seen in the basic recommendation model in Figure 2.1. The model in Figure 3.1 outlines this as well, but specifically for systems that rely on a ■ *latent factor model* for generating ■ *recommendations* (cf. Section 2.2.1). In addition, it illustrates what prevents such systems from being more interactive: As typical for collaborative filtering, the interaction is limited to *standard feedback* expressed with respect to single items (dotted line, cf. Section 2.3.1). In memory-based variants, this feedback immediately leads to updated results, due to their lazy nature. With model-based techniques, however, the feedback is reflected back into the model only during offline training, unless an online updating mechanism is available. ■ *Content- or knowledge-based recommendation techniques* allow for user interaction with the help of *item-related information*, for example, by indicating preferences or critiquing results based on prede-

fined metadata or user-generated tags (see Section 2.1.2 and 2.1.4). Yet, as shown in the literature review, this kind of input data plays no role when it comes to manipulating collaborative filtering models (dashed line). Combinations of different methods, however, are frequently used in order to take advantage of the most suitable method for generating recommendations at each stage of the recommendation process (see Section 2.1.5). Users can manipulate the results of these hybrid solutions only in rare cases, for example, by selecting individual methods or weighting their influence on the final outcome of the system (dash-dotted line). Worse, many real-world systems are entirely limited to (model-based) collaborative filtering (as indicated by the dashed boxes), so that even if hundreds of such algorithms are combined to ensembles, users cannot benefit from other methods. This includes ■ *information filtering methods*, which are frequently employed (see Section 2.3.3), but mostly completely decoupled from recommendation components.

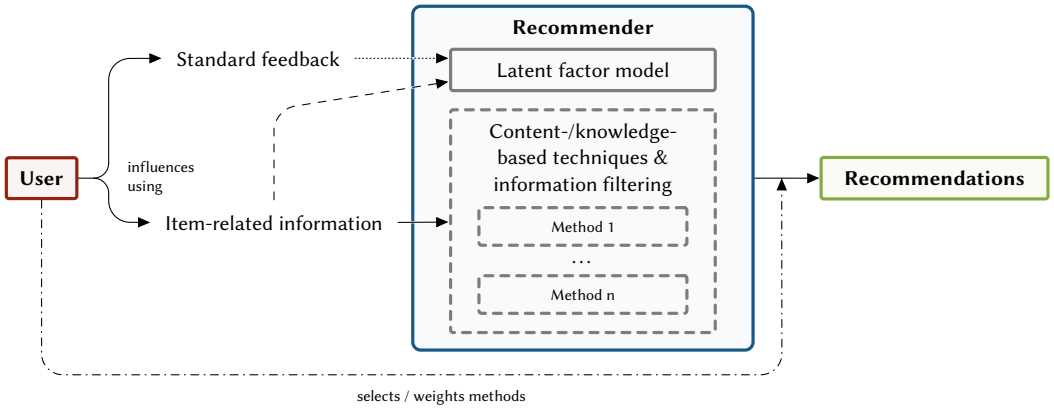

Figure 3.1 Model of user interaction with systems that use model-based collaborative filtering, highlighting their limitations with respect to controllability.

In the following, we address the question of how a recommender based on a latent factor model, which is only fed by user-item interaction data, can be turned into a fully interactive, user-controlled system. For this, we go through the different phases of the recommendation process. Against the background of the literature review, we summarize the problems users face in each phase and point out possible solutions to overcome the limitations indicated by our model. Note that the phases may overlap, are not sequential, and may be defined differently.

3.1.1 Elicitation of initial preferences

As indicated in Figure 3.1, the only way for users to express preferences in collaborative filtering systems is providing ratings for single items—if this is possible at all. Often, implicit feedback is used exclusively, undermining user integrity by making it even more difficult for users to steer the recommendations into a certain direction and to understand the consequences of their actions [JWK14; JJ17]. This is a problem especially in *cold-start situations*: Entering a system or being confronted with a new decision problem, users often do not know for what they are looking. Their information need is vague, developing only as new information is picked up [Bat89]. Then, being limited to rating items may be cognitively too demanding as it requires translating a mental judgment into a numerical value. Moreover, especially in large and unknown domains or in case of experience products, showing items users potentially do not know, and usually cannot consume right away [Loe*18], does not contribute to becoming aware of the search goal. Given

all the criticism this kind of preference elicitation has received (see Section 2.3.1), it thus appears necessary to strengthen this connection in our model (dotted line), enabling users to indicate preferences in systems based on latent factor models in other ways than by standard feedback.

Jameson, Willemsen, Felfernig, Gemmis, Lops, Semeraro, and Chen [Jam*15] argue in context of the *ARCADE* model that representing the choice situation by organizing the necessary information in a more appropriate way may help facilitate the decision process. Letting users state relative preferences for items in a joint evaluation has shown to be more accurate and supports users even if they are not able to ascribe absolute values to separately presented items [RK12; Jam*15]. Thus, one of the natural alternatives is using *comparisons*: No one would rate products in a brick-and-mortar store, but make a purchase decision after comparing them [HT00]. Reflecting this customer behavior, pairwise comparisons consequently enjoy high popularity in decision making and marketing techniques such as *analytic hierarchy processing* [Saa08] or *conjoint analysis* [GS78]. Also in the area of information retrieval, the potential of comparisons was known before it became accepted that users of recommender systems equally prefer comparing items instead of rating them (see Section 2.3.2.1). However, when it comes to model-based collaborative filtering, comparisons still play a minor role. In case of matrix factorization, the exceptions are mainly related to algorithms that optimize a ranking objective, i.e. only compare automatically sampled items in the background (see Section 2.2.2). Providing a user interface that directly reflects how pairwise preferences are processed, yet is considered quite important [RK12; KRG18]. As we will see below, our *first research question* addresses exactly this issue.

3.1.2 Control over the systems

Also later in the process, the only possibility to actively affect the recommendations is providing new ratings or changing and revoking existing ones (see again Section 2.3.1). For this reason, being limited to standard user-item feedback as shown in Figure 3.1 remains a problem, especially as the feedback data are considered as long-term preferences, thus also affecting all future sessions. With matrix factorization algorithms, a single vector is usually persisted to represent these preferences (see Section 2.2.4). Retraining this vector is widely established for online updating [RS08], much effort has been spent on active learning [Rub*15; ERR16], and variants with multiple vectors have been proposed [WWY13]. Nevertheless, users often have the desire to more actively *control the systems* in order to account for situational needs [KR12; PCH12]. Given the success of systems that make use of specific item-related information (see Section 2.1.2, 2.1.4 and 2.3.3), it thus seems promising to let users influence latent factor models in a similar, more expressive way, i.e. to strengthen this connection in our model as well (dashed line).

In line with that, Jameson, Willemsen, Felfernig, Gemmis, Lops, Semeraro, and Chen [Jam*15] suggest based on their *ARCADE* model to combine all available data and perform computations on this basis to improve user support during the decision process. The application of *critiquing*, a technique that has gained considerable popularity for this purpose in interactive recommending research, depends on the availability of predefined metadata, apart from few exceptions such as *MovieTuner*, which rely on user-generated data (see Section 2.3.2.1). Collaborative filtering data, however, are rarely exploited, and thus neither the accuracy-related advantages of model-based algorithms, nor the long-term preference profiles that allow for personalization. Another option, common in interactive hybrid systems, is the use of *weighting* mechanisms: Not particularly useful for conventional preference elicitation [SS11], positive effects on perceived recommendation

quality and system transparency were found in cases in which the weights could be applied on a level above standard user-item feedback (see Section 2.3.2.2). These examples show that collaborative filtering itself rarely serves as a basis for interactive features, or that only the interplay between different methods is affected. The *SmallWorlds* system is an exception in which users can actively take part in the collaborative filtering process, though only for memory-based rating prediction (see Section 2.3.2.5). With respect to the application of matrix factorization, there exist approaches that visualize user profiles or results, but none that allow users to *adjust* the results stemming from their representation within the factor model. Notwithstanding the aforementioned strategy, also the integration of additional information has not yet been exploited for this purpose (see Section 2.2.5). As we will see below, our *second research question*, in contrast, takes this strategy explicitly into account for the implementation of novel interaction mechanisms.

3.1.3 Manipulation in complex scenarios

Finally, as highlighted by Pu, Faltings, Chen, Zhang, and Viappiani [Pu*10], users like to "state preferences on any attributes they choose". In many systems, the opposite is the case: As illustrated in Figure 3.1, only preferences expressed as ratings affect the underlying collaborative filtering models. Above, we argued that with the help of advanced interaction mechanisms, users can be enabled to influence these models more directly. Nevertheless, it may still be difficult to accommodate all situational needs in this way, i.e. ultimately based on collaborative filtering. Consequently, allowing users to *manipulate the results even in more complex scenarios*, in which they are composed by a variety of methods, can be considered another important requirement, with the potential to strengthen the remaining connection in our model (dash-dotted line).

Several attempts have been made for the elicitation of user preferences based on multiple attributes (see Section 2.3.2.1). While these approaches allow users to specify their needs on a superordinate level, therefore capable of modeling preferences in complex domains, most research has been conducted on part of the algorithms [AK15]: Users do not receive any particular support *during* the recommendation process—apart from critique-based systems, which, however, rely on their own recommendation methods (see above). More importantly, in all these approaches, the (mostly predefined) item properties are processed by a *single* method. In contrast, *hybrid systems* combine multiple methods (see Section 2.1.5), and interactive variants demonstrate that composing their output can successfully be put in the user's hands: In *TasteWeights* [BOH12], users can indicate interest in certain topics mined from Wikipedia or give more weight to the opinion of some Facebook friends than of others, according to their situational needs (see Section 2.3.2.2). *IntersectionExplorer* [Car*19] additionally introduces a matrix visualization that allows turning individual methods on or off (see Section 2.3.2.5), similar to interactive approaches in the area of information filtering. In this area, *faceted filtering* is known as a very intuitive and efficient technique for the exploration of large item spaces (see Section 2.3.3), particularly useful if basic mechanisms are not sufficient due to the complexity of the search goal [NH14; KFK14]. However, regardless of the need to help users in such situations, in which they have to compensate themselves the lack of proactive support [WKB05], there is still no closer combination of the initially contrasting approaches of (manual) filtering and (automated) recommending [GKP11]. Especially in commercial real-world systems, users can typically use only one mechanism at a time. Yet, as we will also see below, our *third research question* aims at helping users in the best possible way at all times, by model-based collaborative filtering, but also content- or knowledge-based techniques as well as information filtering methods.

3.2 Derivation of research questions

Having summarized the problems of contemporary systems and discussed possible solutions to overcome the limitations indicated by our model of user interaction, we will now focus on how the underlying research gap may actually be closed. For this, we reflect on our research questions posed at the beginning of this thesis (see Section 1.3), and explain how we derived them based on three ideas that build on one another to bring us closer to our overall goal: improving user control and experience at all stages of the recommendation process by means of interactive methods—as discussed in the previous section.

3.2.1 Exploiting semantics in latent factor models

To improve the elicitation of user preferences in model-based collaborative filtering systems, the most straightforward approach would be to stick with the feedback data that are anyway used by these systems (as indicated in Figure 3.1), but try to take advantage of the patterns hidden in these data. This idea originates from the literature review, showing that latent factor models are primarily used for improving accuracy, without exploiting the widely accepted assumption that the dimensions, which are derived from these data, relate to actual real-world concepts. Given the corresponding evidence provided in Section 2.2.5, the question thus arises:

RQ1: How to *exploit the semantics* in latent factor models for improving user control and experience?

We assume that the meaning in the model dimensions can be conveyed by letting users *compare representative items*. Then, the system should be able to understand the relative preferences expressed in this way. In line with Section 3.1.1, we expect this to be particularly beneficial for *eliciting initial preferences*, i.e. at cold start, when item-related preferences need to be obtained as accurately and quickly as possible, but ratings come with a set of drawbacks.

3.2.2 Leveraging item-related information

While it appears promising to exploit their semantics, natural limitations exist when latent factor models are exclusively derived from regular user-item feedback. In particular, there is always the necessity to fall back on items themselves to exert control. In the literature, it is often suggested to boost the models with content information. Whereas this is again mostly done for improving accuracy, approaches in which item-related information represents the *main* datasource usually offer higher interactivity, for example, using content- or knowledge-based techniques or information filtering methods (as illustrated in Figure 3.1). Concerning model-based collaborative filtering systems, this leads to the question of:

RQ2: How to *leverage item-related information* in addition to standard collaborative filtering feedback data for improving user control and experience?

In case the latent factors become accessible in this way, we assume that more advanced interaction mechanisms can be offered, enabling users to indirectly manipulate their position in the factor space, and thus the recommendations. As outlined in Section 3.1.2, users could *apply critiques* while the recommendations would still reflect their long-term preferences expressed in the past through conventional ratings. Moreover, users could *specify weights* for determining

the influence of certain factors. Overall, users would not be limited to expressing their preferences with respect to items as it is usually the case (even if comparisons take place as discussed above). Instead, they could use the item-related information for *controlling the system's outcome*, remarkably without affecting their representation within the factor model for future sessions.

We expect this to be particularly beneficial for users who want to make adjustments based on their *current situation*, which allows to keep the typical representation in the form of single vectors. Still being able to personalize the results via rating-based feedback can at the same time be considered a significant advantage over pure content- and knowledge-based techniques, and also most of the approaches to interactive recommending, which do not build on the benefits of model-based collaborative filtering. On the other hand, no further information about the current user would be needed due to the focus on *item-related* information. Using information provided by the user community, however, would still be possible—and advisable given the potential shown for interactive recommending in Section 2.3.2.1. In addition, some works mentioned in Section 2.2.5.2 have shown that including concepts in the language of the users may contribute to *opening up the black boxes* latent factor models still constitute.

3.2.3 Merging recommendation and information filtering methods

Finally, there are scenarios in which it is not sufficient to follow the aforementioned approaches as they "only" allow users to intervene in the underlying model. However, there is often an interplay between such a model and other methods (as also illustrated in Figure 3.1). Attempts to make these hybrid solutions more interactive are rare. Methods that are more interactive by nature, such as from the area of information filtering, appear mostly decoupled from recommendation components, especially from state-of-the-art model-based variants. In light of these issues, to ultimately get to our overarching goal as close as possible, the question comes up:

RQ3: How to *merge model-based collaborative filtering* with other recommendation and information filtering methods for improving user control and experience?

Given the considerations in Section 3.1.3, we expect that a common *hybridization strategy* with a front-end based on *faceted filtering* may be an appropriate point of departure for providing users the possibility to adjust the final system output in the most holistic manner. The combination of the right methods from the whole range of options, and handing control of this combination over to users, should enable them to *manipulate the results even in complex scenarios*: Not being limited to using individual methods separately, and, in particular, to just influencing a latent factor model that represents only one part of the system, they could always take advantage of functionalities diametrically different to collaborative filtering. Still, they could benefit from its accuracy when it comes to recommendations personalized according to long-term preferences, as well as from the direct extensions discussed in the previous sections.

3.3 Contributions in context

Having explained how users can be supported in the different phases of the recommendation process, and laid the ground for our three research questions, we now describe how we concretely address these questions in the following chapters. With this overview, we provide an *advance organizer* for the main contributions of this thesis (cf. Section 1.3):

- In Chapter 4, we address our first research question and describe a novel *choice-based preference elicitation method* for model-based collaborative filtering systems. This method, initially presented in [LHZ13; LHZ14], has no additional data requirements, but *exploits the semantics in latent factor models*. By picking up on the idea of using comparisons as described in Section 3.2.1, this shows that *initial preference elicitation* can thus be made more effective.

- Next, in Chapter 5 and 6, we address our second research question and explain how *leveraging item-related information* can be a benefit also for other purposes than increasing accuracy, namely for improving user control and experience even further. Based on the method and the corresponding framework we present for *boosting matrix factorization with content information*, we propose *advanced interactive features* that can be implemented as extensions to existing collaborative filtering systems. For this purpose, we initially suggested to integrate the underlying models with user-generated tags [DLZ15; DLZ16a; DLZ16b; Loe*19b]. Whereas any type of attribute may be used, this kind of item-related information represents a meaningful running example, illustrating that users can successfully be enabled to directly *exert control over the systems* in a more expressive manner, as described in Section 3.2.2.

- Finally, in Chapter 7, we address our third research question and present the concept of *blended recommending*. First proposed in [Her*14; LHZ15a; LHZ15b], this envisages to *merge model-based collaborative filtering* with other established methods in a fully user-controlled fashion. Using faceted filtering as described in Section 3.2.3, users can thus be enabled to *manipulate the results even in more complex scenarios*, while maintaining the individual benefits of the (in principle easily exchangeable) methods that are responsible for these results. Remarkably, this includes model-based collaborative filtering components, possibly extended with the other novel interaction mechanisms.

After these methodological contributions, it makes sense to briefly elaborate on their *evaluation*: Provoked by taking a human-computer interaction perspective on past research, in which retrospective offline experiments were mostly used (cf. Chapter 2), and in light of our main goal, we follow a user-centric approach. For this, the *framework for user-centric evaluation* introduced by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12] can be seen as a meaningful tool, allowing to measure the impact of our proposed methods in terms of user control and experience, i.e. in line with the research questions.

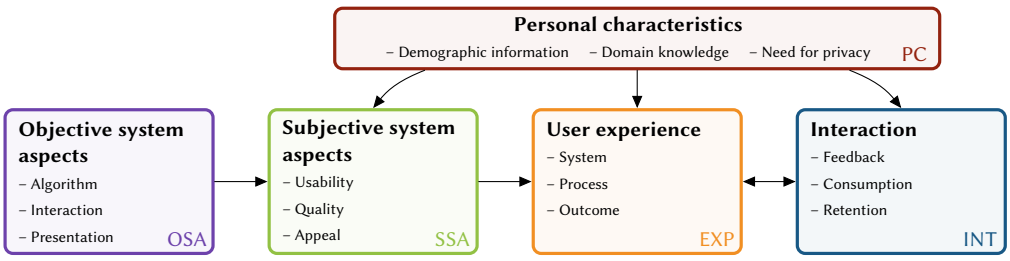

Figure 3.2 Framework by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12], showing constructs for the user-centric evaluation of recommender systems.

As Figure 3.2 illustrates, this framework describes how *objective system aspects* (OSA) such as used recommender algorithms or interaction mechanisms affect *subjective system aspects* (SSA),

including general usability, but also perceived recommendation quality and diversity. The assessment of these aspects has an influence on constructs related to *user experience* (EXP), for example, perceived effectiveness of the system, effort during the recommendation process, and satisfaction with the items finally chosen. Eventually, user experience is strongly related to the *interaction* (INT) of users with the system, which is in turn influenced by *personal characteristics* (PC), as is their attitude towards the other aspects mentioned before [KWK11; Kni*12; KW15].

With this, we can proceed with the remaining contributions:

- Completing Chapter 4, 6 and 7, we present *empirical evaluations*, each comprising one or multiple user experiments. Consistently relying on the user-centric evaluation framework, we vary several objective aspects of model-based collaborative filtering systems in order to validate the effectiveness of our proposed developments and to study their impact on the subjective assessment of relevant system aspects and on user experience. Although these experiments are exploratory in nature, this allows us, on the one hand, to use hypotheses to describe our expectations and guide the analyses, on the other hand, to apply statistical methods to gain insights that help us answer our research questions. Accordingly, we first compare our *choice-based preference elicitation method* against typical baselines (Chapter 4), exploring whether the semantics in latent factor models can actually contribute to improving user control and experience (RQ1). Next, we compare the performance of *content-boosted matrix factorization* with a regular model, and study its application for the implementation of *interactive features* (Chapter 6). This way, we address the general question of the effect of considering side information from a user perspective, and, more specifically, investigate whether leveraging item-related information can help to further improve user control and experience (RQ2). Finally, we compare an interface implemented according to our concept of *blended recommending* with a baseline filtering interface (Chapter 7), focusing again on possible benefits regarding these aspects (RQ3). In the sections that describe these experiments, more details can be found regarding the methodology, including prototype systems (screenshots in Appendix A) and questionnaires (complete overview in Appendix B). Succeeding the description of each experiment and the presentation of the results, we discuss our findings and elaborate on how they contribute to answering the corresponding research question.
- Next, in Chapter 8, with the intention of tying together the ideas behind our research questions, we present an *integrated recommendation platform*. Initially presented in [LZ19b], this platform combines all our developments in a single system. This system showcases how the individual methods eventually all contribute to the main goal of this thesis, supporting users in a holistic manner during the entire recommendation process with adequate interaction possibilities. To illustrate that the approaches—when implemented in the form of a set of seamlessly connected perspectives—may help users reach typical search goals, we additionally present several illustrative *case studies*.
- Finally, in Chapter 9, we *conclude* the thesis with a discussion of the results in relation to the research questions and our original goal. Furthermore, we provide an *outlook* on how future recommender systems may be improved and made even more interactive, starting from one of the current contributions.

CHAPTER 4

"All your life, you will be faced with a choice.
You can choose love or hate…
I choose love."

— *Johnny Cash, American singer-songwriter*

Choice-based preference elicitation

In this chapter, as a first step towards increasing the level of user control in model-based collaborative filtering recommender systems, we propose a novel method for eliciting the user's (initial) preferences. In Section 3.1.1, we have illustrated that comparisons might be a promising alternative to rating single items, the option most frequently offered in cold-start scenarios. Now, we present a *choice-based preference elicitation method* [LHZ13; LHZ14] that allows to integrate any recommender system that relies on matrix factorization with an interactive dialog for capturing preferences via comparisons. With this, we directly address our first research question: As outlined in Section 3.2.1, we make use of the semantics contained in latent factor models derived from conventional user-item feedback, and present sets of sample items that represent the dimensions of these models in a stepwise manner. Entirely relying on the inherent properties of the models and without posing additional data requirements, users can thus be enabled to influence the recommendations right from the outset by expressing their preferences for these sets according to their situational needs. In the following, we elaborate on the *background* of this method, describe the *method* itself, and finally present an *empirical evaluation* that explores its effectiveness by means of a comparison against several baselines [LHZ14].

4.1 Background

The method we suggest aims at combining the benefits of interactive recommending approaches with those of model-based collaborative filtering. Particularly inspired by systems such as *MovieTuner* [VSR12], the main objective is to intuitively *guide the user* through the preference elicitation process, achieving a good trade-off between system support and user control. In contrast to *MovieTuner* and many other interactive approaches (cf. Section 2.3.2.1), this should not imply that rich information regarding the items is required, for example, predefined metadata or user-generated tags. This voluntary restriction serves the practical purpose that collaborative filtering data are often more readily available. Still, users should be able to actively take part in the process without having to rate any items or to know the details of the underlying algorithm. To make user interaction easy and intuitive, we thus propose to show *system-selected examples* of typical items from which users can choose the ones they prefer. As in critique-based approaches, we assume that anchoring the preference elicitation process to examples makes it easier for users to express their often unclear or even unconscious preferences (cf. Section 2.3.2.1). In case of movies, they should be able to tell the recommender that they want "something animated such

as 'Toy Story' or 'Cars'" or a "dark sci-fi movie such as 'Alien' or 'Blade Runner'". For this, however, we *only* consider availability of a standard user-item matrix as a prerequisite, i.e. the data requirements are the same as for any standard collaborative filtering system.

Inspiration from conjoint analysis This idea is inspired by *conjoint analysis* [GS78], a survey-based technique often used in marketing research for statistically determining customer preferences for product attributes. While the products originally had to be rated or ranked on an individual basis, *choice-based* conjoint analysis, the most popular variant today, confronts customers with sets of products [Hub05]. These products are made up from all considered product attributes. Step by step, all combinations are then compared with each other, and users asked to indicate which combinations they prefer. Figure 4.1 shows an example.[11]

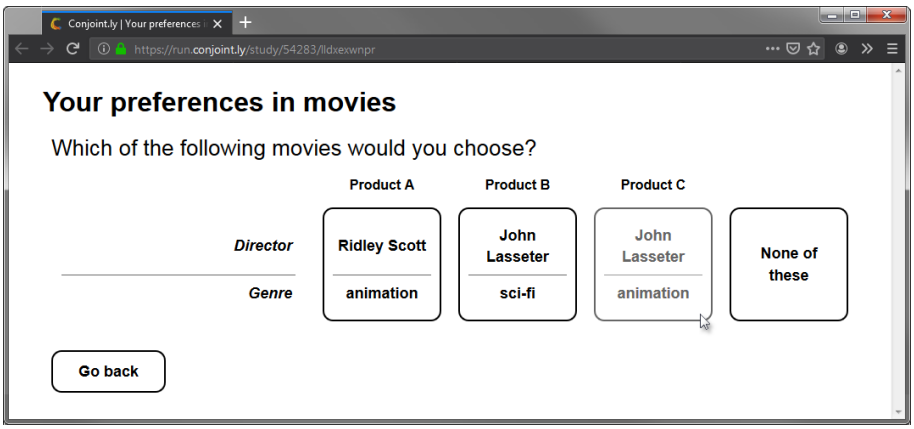

Figure 4.1 Example of one step of a conjoint analysis on movie preferences, conducted using the online service of *conjoint.ly*.

For such an analysis, it is generally necessary that a set of predefined attributes exists, and an appropriate way to present the products can be found. Even if these requirements are fulfilled, the analysis may be prone to errors due to the hypothetical nature of the products: The realism of the stimuli has shown to be an important factor for the validity of conjoint analyses [EHS17], especially for assessing the emotional and situational value of a product. For example, it could be difficult to imagine an animated kids' movie by the "Alien" and "Blade Runner" director Ridley Scott as shown in Figure 4.1. In addition, the customer's experience in the respective domain affects whether he or she can interpret and evaluate all attributes and their corresponding values [SM18]. Finally, a full factorial design as described above, in which all possible combinations occur, drastically increases the effort for taking the survey, requiring participants to pay additional attention when the number of attributes is large. For these reasons, conjoint analysis has yet only rarely been applied in recommender research. An exception is the work by Bruyn, Liechty, Huizingh, and Lilien [Bru*08], who use conjoint analysis to build a preference model ex ante. This model subsequently serves as a means to determine an optimal sequence of questions for eliciting the user's preferences. Whereas this turned out effective for products where preferences depend on objective attributes, the authors themselves considered it difficult to apply conjoint analysis techniques for experience products.

[11] This toy example can be accessed here: `https://run.conjoint.ly/study/54283/lldxexwnpr`

Realization with matrix factorization Thus, our goal is to exploit a *standard* recommendation algorithm, and enable users to express their preferences with respect to *actual* items, instead of fictional ones as created for conjoint analyses. To further facilitate decision making, we aim at limiting the process to *binary* choices. This is in line with findings from conjoint analysis research, which emphasize the efficiency of pairwise comparisons when there are many attributes, and suggest to use this type of comparison for difficult or emotional choices [MH16]. From a recommender systems perspective, the advantages of comparisons to represent the user's choice situation have already been discussed in Section 3.1.1.

To generate an *interactive dialog* that meets these demands, the only requirement is that a vector of latent factor values is assigned to each item. This can be done with any matrix factorization algorithm (see Section 2.2). Then, the common procedure for generating recommendations is to multiply the vectors of users and items. As indicated in our model of user interaction (cf. Figure 3.1), this requires that the target user has already provided feedback for a sufficient number of items, so that a user-factor vector can be derived. However, we instead suggest to position the user directly within the k-dimensional vector space in which the items are arranged according to their item-factor vectors (where k is the number of factors to be learned). For this, only sample items need to be identified in a way that they match the characteristics represented by certain areas of the factor space. Driven by examples, the *semantics in the derived dimensions* can then be conveyed without explicitly describing the factors (which may still be difficult, see Section 2.2.1 and 2.2.5). Using *sets* of items furthermore avoids the need for users to comprehend the underlying concepts based on single factor representatives. This reduces the risk of misinterpretation due to specific item properties and increases the likelihood that users are familiar with at least one of the items. The latter appears especially important in light of our related work, where we have shown that sometimes *only* item consumption enables users to adequately approximate the value of a recommended item [Loe*18].

Either way, a variety of sampling techniques may be used, including item clustering as suggested in other works [cf. Gan*09; RK12]. However, to model the interactive dialog as a series of *binary choices between representative items*, we deem it more promising to assess the user's preferences with regard to single factors. Figure 4.2 illustrates this with two factors, each represented by an axis. Taking again the example of movies, the first factor might represent the degree of humor, the second factor the relation to sci-fi. For each factor one after the other, two sets of sample items are then shown in the dialog: One set comprises items that are selected as representatives because of their low values for the current factor f, whereas the items in the other set score highly for this factor. In each interaction step, the user is asked to choose either the set with low values S_{fa} or the set with high values S_{fb}. In the example, the user first decided for funny, then for sci-fi movies (red dots in both steps). Based on the position that is incrementally determined in this way, he or she can then be presented with recommendations, here for sci-fi comedies such as "Back to the Future" or "Space Balls" (green dots in the last step).

Such a *conversational* interaction can generally be considered helpful for users to make more informed and accurate decisions when asked for their preferences [Xie*18]. Nevertheless, with the conventional factor model in the background, capturing preferences in the form of ratings is still possible. However, whereas historical feedback by the user community is necessary for learning the model, the target user does *not* need to provide item feedback him or herself, which makes our approach particularly useful at cold start and in case he or she does not want a long-

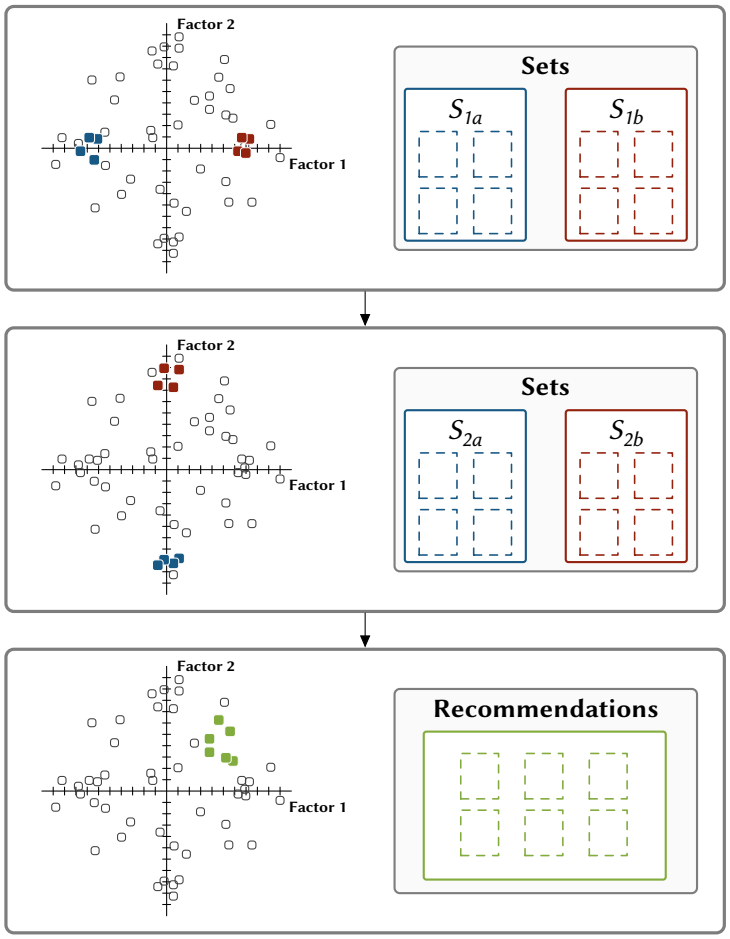

Figure 4.2 Example of the preference elicitation process with our choice-based method: For two factors, sets of representative items S_{fa} and S_{fb} are presented to the user, either related to low (blue) or high (red) factor values. Here, the user chooses the set with high factor values in each step, leading to the items being recommended that are finally highlighted (green).

term profile to be applied. Note that with comparisons directly based on the factor space, it is also not required to continuously access the user-item matrix as in the approach by Rokach and Kisilevich [RK12], the most comparable one to ours (cf. Section 2.3.2.1).

4.2 Method

In the following, we elaborate in more detail on how to *select and order the factors* for the choice-based dialog, and how to subsequently *determine representative items* for these factors. Finally, we describe how *recommendations* can actually be generated according to the user's decisions.

4.2.1 Selecting and ordering factors

In order to position the user as precisely as possible in the latent factor space, i.e. to determine his or her preferences most adequately, one would in principle need to iterate over all factors. This is not advisable from a user perspective as matrix factorization models in recommender systems

often consist of up to 100 and more factors in order to achieve maximum performance in terms of objective accuracy metrics. With one interaction step per factor, users would thus become bored or impatient. Also, distinguishing between the sets of sample items would become more and more difficult. Similar problems causing increased cognitive load have been found in context of conjoint analysis [HNM16; SM18]. Accordingly, a trade-off needs to be established between a sufficiently good positioning and an acceptable number of interaction steps.

When asking users to rate items in standard collaborative filtering systems, eliciting 5 to 20 ratings is generally considered sufficient, with 10 ratings constituting an adequate trade-off between accuracy and effort (at least in the movie domain) [cf. CGT12; ERR14]. Later, it has been shown that in practice, preference elicitation via pairwise comparisons does not need more data than rating-based variants [BR15]. Theoretically, given the binary decision process as described above, one could thus argue that 5 dialog steps would be an appropriate minimum: If the item sets were as small as possible (i.e. comparison of one item vs. another), users would be able to express their preferences with respect to 10 items—or a few less if items show up for multiple factors or appear unknown. With larger sets, users would express their preferences towards more items. This could ease making decisions and eventually improve the quality of the results, but at the same time raises the chance that not all items are known to the user. Worse, within these sets, items might be perceived to be in conflict. This particularly applies as we suggest to determine their similarities based on their latent nature. However, the results of preliminary experiments we conducted with early prototypes suggest that users can well distinguish between larger sets without too much effort if the number of decisions is limited to 5–10, leading to recommendations that fit reasonably well to their preferences. This was confirmed in one of our prior studies conducted with $n = 14$ participants and a dialog with exactly 5 steps, each showing sets of size 4 [LHZ13]. Beyond that, we found that a higher number of steps in the interactive dialog may lead to items being selected as representatives for multiple factors at the same time, thus increasing the difficulty of understanding the differences between the sets, but not necessarily the recommendation quality. Accordingly, while the number of factors could in principle also be determined dynamically, showing representative item sets for a fixed number of 5 factors seems to be an overall reasonable trade-off. Besides, this number is well in line with the number of interaction cycles users need in critique-based systems [VFP06; ZJP08; CP12b], and with the work of other authors who later adopted our approach [Ros*16; Liu*18].

Consequently, given the high dimensionality of factor models, it is necessary to identify the most important factors in order to limit the interaction steps accordingly. These factors should differentiate between the items as well as possible, so that comparisons are easier than with examples of less distinctive factors. This problem can be approached in several ways: With so-called *factorwise* matrix factorization algorithms (see Section 2.2.3.2), the factors are learned one after the other in descending order of explained variance [BKV07a]. Thus, they are already ordered by their distinctiveness [FHK12]. The same is true for the content-boosted matrix factorization method we describe in Chapter 5, which inherently provides information on the importance of the factors. By relying on one approach or the other, we can thus limit the factors for the choice-based dialog to the most important ones in all cases relevant for this thesis.

However, in case one would apply a factorization technique that does not directly provide such insights, other solutions may be used. These include naively selecting a few factors at random, but also feature selection techniques [TAL14], exploitation of the properties of non-negative

matrix factorization [Liu*18], and automatic identification of semantic concepts [Weg*18]. As we have recently proposed in related work, one could also determine the distinctiveness of factors via games-with-a-purpose [KLZ18a; KLZ18b; Kun*19b]. Finally, outside cold-start scenarios, the next most important factor could be identified under consideration of the items previously shown to the user or based on his or her existing long-term preference profile.

4.2.2 Determining factor representatives

Next, to determine sample items that adequately represent the previously selected factors, it is first important to decide *how many* representatives should be shown for each factor: Chang, Harper, and Terveen [CHT15] propose to use groups of items for eliciting user preferences. However, for groups of three movies, they received feedback from participants of their user experiment that one or two movies more would have been a better choice. Such a number would also fall in the range of items users can handle cognitively [Mil56; Cow10], and which are shown in most example-critiquing approaches [CP12a]. Thus, also based on insights gained from the preliminary experiments, we show 4 items per set, i.e. overall 8 items in each step of our dialog. Initially considering a larger number of items for each set (e.g. 25), to then select randomly the desired number of items for the presentation at the front-end, additionally reduces the probability that the dialog looks exactly the same when the user returns at a later time.

Regarding the question *which* items to show, previous research has indicated that the items for an initial preference elicitation process need to be both popular and controversial [RK12]. As mentioned earlier, various sampling techniques may be used. But, with a binary decision process in which users are asked for preferences regarding a single factor in each step, the most natural way to convey the semantics of a factor would be to present items that are located at the extrema of its scale, i.e. which have either very low or high values in the respective component of the factor vector, and thus very different characteristics. However, as confirmed by our early pilot studies and informal interviews with test users, naively selecting items that possess minimum or maximum values does not yield well discriminable representatives. Also as a result of these experiments, we instead suggest *three requirements* for composing the sets of sample items.

Popularity of items First, to ensure that users are able to make qualified judgments, we propose to focus on *popular items*, depending on the total number of ratings $|R_i|$ an item i has received:

$$\text{pop}(i) := |R_i| \,. \tag{4.1}$$

Additionally, the average rating (as in formula (7.1), also used by the *Internet Movie Database* (IMDb) to calculate the top 250 movie charts[12]) or the rating entropy [cf. Ras*02] may be taken into account. This information is part of the user-item matrix, and thus easily applicable to filter out all items below a specified threshold, as illustrated in Figure 4.3. Furthermore, old items could be filtered out or considered with reduced weight in order to get a list of items that are generally well-known. This is especially important in case the average rating is (at least partially) considered, as it usually increases with item age [Das*10]. Besides, newer items have a disadvantage per se, as less user-item feedback is naturally available for items that were recently added to the underlying dataset. Although considering an item's age implies leveraging

[12] https://www.imdb.com/chart/top/

additional information (i.e. which is not contained in the user-item matrix), this does not restrict the general applicability of our approach, but rather constitutes a domain-specific optimization step that may be omitted. Moreover, at least some basic metadata are almost always available. Finally, while user-item interaction data are not required for the current user, this knowledge could in principle be taken into account as well: If available, items could be constrained to those that the user already knows about, or at least to similar ones. Since item familiarity has been shown to affect the way users evaluate recommendation sets [JLJ15a; JLJ15b], one can assume that this would also make the resulting sets of sample items easier to assess, though it may introduce additional bias that would need to be considered.

Distinctness of sets Next, users should not only be familiar with the factor representatives in order to make informed and qualified choices. More importantly, the representatives need to be selected in a way that the resulting sets are highly distinguishable with regard to the current factor. Otherwise, it would hardly be possible for users to clearly state a preference. On the other hand, the items should still be comparable, i.e. not too different [KRG18]. Moreover, extreme factor values might distort the decision because the items corresponding to these values need to be considered as outliers in typical item distributions. Thus, we first remove the items in the lower and upper fifth percentile for each factor. Afterwards, the items that remain for each factor can be partitioned by dividing the item space into 4 equally-sized intervals. While the inner segments, which contain items with rather neutral values, are ignored, items in the lower and upper 25 % value interval, which we call segment a and b, are subsequently chosen as candidate representatives for factor f. In Figure 4.3, this is illustrated for the first factor. Additionally, weights can be assigned to these items based on the respective factor value, representing their individual *relevance*:

$$\mathrm{rel}(i, f) \coloneqq |q_{if}| \,. \tag{4.2}$$

While segment size is also subject to parameterization, using 4 intervals as described above ensures in typical item distributions that the segments are large enough to contain a sufficient number of items for the final selection of representatives that is shown to the user.

Isolation of factors Eventually, it is not only important to ensure that the items are diverse with respect to the factor at hand, but as neutral as possible with respect to all other dimensions. Otherwise, there might be various, possibly conflicting options to interpret the differences between the sets, and difficulties to comprehend the relationships of the items within. Therefore, after focusing on items with dissimilar values for the current factor f, we need to isolate this factor. For this purpose, we construct average item-factor vectors $\vec{q}'_{fa}$ and $\vec{q}'_{fb}$ for both of its segments a and b: The components that represent the current factor are set to the mean value for this factor over the items in the respective segment, whereas the components that represent the other factors are filled with the mean values over all the remaining items:

$$\vec{q}'_{fs_k} \coloneqq \begin{cases} \sum_{i \in I_s} q_{ik} / |I_s| & \text{if } k = f, \\ \sum_{i \notin I_s} q_{ik} / |I \setminus I_s| & \text{else,} \end{cases} \tag{4.3}$$

with I_s being the set of items contained in segment $s \in \{a, b\}$.

By this means, these artificial item-factor vectors are positioned at the centers of the segments, while the distance to the average remains small in all other dimensions. The exclusion of items

from the lower and upper percentile in the previous step prevents that these vectors are shifted too much towards outliers. Accordingly, by determining their distance to the vectors $\vec{q}_i$ of all candidate items, a *specificity* weight can be assigned to all candidate items as follows:

$$\text{spec}(i, f, s) := \frac{1}{\|\vec{q}_i - \vec{q}'_{fs}\|} . \tag{4.4}$$

Now, we can select the candidate items from segment a and b with the highest weights to show them as representatives for the current factor, i.e. add them to the sets S_{fa} and S_{fb}. Figure 4.3 also illustrates this final step of the *successive application* of our three criteria.

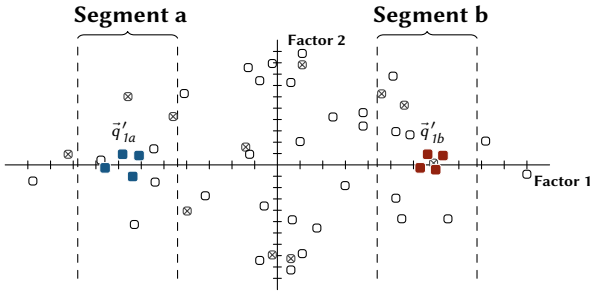

Figure 4.3 Schematic example of the selection of factor representatives: Items not popular enough are ignored (crossed-out gray). The item space is then divided into segments for the currently presented factor (here factor 1), of which only those with items that have either very low or very high factor values are used, ensuring that the representatives are sufficiently different with respect to this factor. Afterwards, items near to the average item-factor vectors are selected (blue and red), so that their other characteristics are as neutral as possible (here only factor 2).

Alternatively, one can also calculate an *overall score* for all items i in the respective segment s of factor f by taking all criteria into account at once:

$$\text{score}(i, f, s) := w_p \cdot \text{pop}(i) + w_r \cdot \text{rel}(i, f) + w_s \cdot \text{spec}(i, f, s) , \tag{4.5}$$

where w_p, w_r and w_s are application-specific parameters that may be determined empirically. Subsequently, the n items with the highest scores can be added as representatives to the sets S_{fa} and S_{fb}. Note that for calculating the overall score in (4.5) it makes sense to additionally scale or normalize the individual components ex ante (e.g. via log transformation).

These three criteria have shown to be useful heuristics for sampling the item space in a way that the representatives are likely known to most users, well distinguishable, and very characteristic for the factor currently under consideration. In addition to these criteria and the factor selection described before, other important considerations need to be made when implementing our choice-based preference elicitation method: There are phenomena in decision psychology such as the tendency of people to lay their focus on the first item in comparison tasks, which gives items shown on the left-hand side an inherent advantage [DS92]. Accordingly, we randomize which set of representatives is shown on the left, which on the right. Moreover, the availability of a “do not care” option may foster choice deferral, and thus influence the decision process as

well [Dha97] (not only when users select the no-choice option, but even due to its mere presence [PS11]). Consequently, designing the steps of the interactive dialog in a meaningful way goes further than carefully determining factor representatives. But, this is difficult to generalize as these considerations highly depend on domain and application.

4.2.3 Generating recommendations

Finally, recommendations need to be generated according to the user's decisions. For this, a new user-factor vector is set up at the beginning of the choice process and iteratively updated with each interaction step: If the user chooses the set with low values S_{fa} when the representatives for a factor f are shown in one of the steps, the f-th component of this vector is set to the center point of the corresponding interval. For instance, in case segment a spans over factor values from -1 to -0.5, the value for this segment is -0.75. In contrast, the f-th component is set to the center point of the interval containing the items in segment b, if the user settles on the set with high values S_{fb}. More formally, this can be expressed as follows:

$$\vec{p}'_{u_f} := d_{sl} + \frac{d_{sr} - d_{sl}}{2}, \tag{4.6}$$

where $s \in \{a, b\}$ refers to the segment from which the user preferred the representatives for factor f, and d_{sl} and d_{sr} define the left and right edges of this segment. If the user prefers not to make a decision, the corresponding dimension is ignored in the remainder of the process. For new user cold-start scenarios, for which we have proposed our choice-based preference elicitation method originally, using the no-choice option thus means that the preference vector's f-th component is set to zero or a neutral value. The same is true for vector components that correspond to factors that were left out when determining the most important dimensions as described before.

Taijala, Willemsen, and Konstan [TWK18], who propose to interactively generate recommendations depending on the user's binary rating feedback for single items (see Section 2.3.2.1), start the preference elicitation process from a location in the latent factor space that is chosen in a similar fashion, namely, using a non-personalized vector averaging over all other user-factor vectors. However, they address not only cold-start situations, but also suggest two alternatives to continue the collection of user preferences later on: Unsurprisingly, in their user experiment, participants with a starting location based on a regular $\vec{p}_u$ vector were less active in the system as they immediately received personalized recommendations according to their long-term preferences. On the other hand, participants with a starting location based on a short-term preference vector (made up by averaging the vectors of items they rated most recently with a high score) or a non-personalized variant (similar to our approach), were more appropriately supported in discovering novel items and obtaining recommendations in line with situational needs. Nevertheless, also in our approach, it would effectively be possible to start with an individual $\vec{p}_u$ vector that corresponds to long-term preferences, for instance, by inheriting its values to $\vec{p}'_u$ to determine a personalized location.

Either way, the incrementally adjusted vector $\vec{p}'_u$ can finally be used as usual to generate recommendations via dot multiplications, i.e. with the standard matrix factorization *recommendation function* $\mathrm{s}(i|u)$ shown in (2.3) in Section 2.2.1. Alternatively, the items whose latent factor vectors $\vec{q}_i$ are spatially the most similar ones to this vector may be recommended.

4.3 Empirical evaluation

In order to evaluate the effectiveness of our choice-based preference elicitation method in terms of improving user control and experience, we compared it in an exploratory user study with several baselines. For an interactive preference elicitation approach, this is more important than results of objective performance metrics. Accordingly, we developed a prototypical recommender system, using movies as a running example, in which we implemented our choice-based method as well as three common alternatives for eliciting user preferences at cold start. With this prototype, we then carried out the experiment with $n = 35$ participants, who were asked to test the different methods and fill in a questionnaire.

In the following, we describe the *goals* and list the *hypotheses* for this experiment, which was first reported in [LHZ14]. Next, we explain the evaluation *method*, including the prototype system and the underlying datasets, the questionnaire, and the exact procedure. Afterwards, we report the quantitative *results* and finally conclude the chapter on choice-based preference elicitation with a *discussion* of these results in light of our first research question.

4.3.1 Goals and hypotheses

Since we proposed our method as an alternative to rating-based preference elicitation, an obvious goal was to evaluate its effectiveness from a user perspective in comparison to a model-based collaborative filtering recommender that implements such a typical preference elicitation process. We assumed that this baseline, in which recommendations are *automatically generated based on ratings*, would be perceived as more effortful due to the cognitive demand of ascribing absolute values to the items, and less trustworthy because of the low transparency. Moreover, we assumed that preferences would be captured more effectively via comparisons, leading to better recommendations. In addition, we expected that users would perceive this kind of interaction as more adequate, and would thus have a greater feeling of control. At the same time, the fact that the corresponding decisions are likely made based on items the users are familiar with, should not restrict the novelty of the results, because ratings equally are provided for known items only.

Despite the exploratory character of the experiment, we posed the following *hypotheses* to test these assumptions in a structured manner and to guide our analysis:

H1 Choice-based preference elicitation leads to recommendations of higher perceived *quality*.
H2 Choice-based preference elicitation has no negative impact on *novelty* of recommendations.
H3 Choice-based preference elicitation positively affects *trustworthiness*.
H4 Choice-based preference elicitation improves perceived *interaction adequacy*.
H5 Choice-based preference elicitation improves perceived *effectiveness* of the system.
H6 Choice-based preference elicitation increases the feeling of *control*.
H7 Choice-based preference elicitation reduces perceived *usage effort*.

To be able to make a more qualified assessment, we aimed at considering two additional baselines: First, a complete opposite method, supporting *manual exploration* with well-known search and filtering mechanisms. Second, a simple in-between solution, providing *popularity-based* recommendations, likely good enough for the majority of users, but with no interaction at all.

4.3.2 Method

We conducted the experiment as a user study under controlled laboratory conditions. We recruited $n = 35$ participants (11 female, 24 male) with an average age of 29.54 years ($SD = 7.81$). The experiment was guided by a supervisor who handed out task descriptions and questionnaire in paper form. Participants used a desktop PC with 24" LCD (1920×1200 px resolution) and a common web browser to interact with the prototype system we implemented for the study.

Note that we ran several pretests with test users, and conducted a user experiment similar to the one reported here. For this experiment, which can be considered a prestudy in a user-centered design process, we only had $n = 14$ participants (7 female, 7 male), with an average age of 34.50 years ($SD = 14.10$). The design was overall the same and led to results very similar to the ones reported in this section. More details can be found in [LHZ13].

Prototype To compare our approach with the three baselines, we set up a web application with four different interfaces based on the following methods:

- A *popularity-based recommender*, simply returning the most popular items in the dataset by means of the function shown in (7.1), which is similar to the formula used by the *Internet Movie Database* (IMDb) to calculate the top 250 movie charts.[12]

- A *manual exploration interface* that allowed users to freely interact with conventional search and filtering mechanisms: As shown in Figure A.1 in Appendix A, users were able to explore the space of available items, apply filter criteria, and inspect item detail pages. Hyperlinks were provided to facilitate the filtering process, enabling users to obtain lists of movies assigned with a specific tag, directed by a certain director, or starring a preferred actor. A shopping cart functionality was additionally integrated for the purpose of the study.

- A *rating-based collaborative filtering recommender* relying on standard matrix factorization: Because of its support for online updating, we employed the `MatrixFactorization` algorithm from the *MyMediaLite* recommender library [Gan*11]. We used 10 factors, which is usually considered sufficient [cf. KBV09; ERR14]. With 30 iterations and $\lambda = .03$, we obtained a *root mean square error* (RMSE) of 0.858 after extensive pretesting using 10-fold cross validation on the *MovieLens 1M* dataset,[13] i.e. performance was up to standard.

- The *choice-based preference elicitation method* as described in Section 4.2: We used a matrix factorization algorithm similar to the one mentioned above, but with *factorwise* learning as proposed in [BKV07a]. Concretely, we employed the `FactorWiseMatrixFactorization` algorithm from the *MyMediaLite* library with 10 factors. The standard parameterization led to a similar RMSE of 0.864, and, according to the pretests mentioned above, meaningful sets of representatives. We restricted the candidates for these sets to the 150 most frequently rated items, and filtered out movies released before 1960 (see Section 4.2.2 for details on these additional restrictions). Figure A.2 shows an example of one step of the resulting dialog.

[13] The *MovieLens 1M* dataset contains about 1 million ratings from more than 6 000 users for over 4 000 movies. It can be found here: `https://grouplens.org/datasets/movielens/1m/`

Datasets As background data for implementing the four interfaces, we used the *MovieLens 10M* dataset for user-item feedback.[14] At the time we conducted the study, this dataset was widely considered one of the standard datasets for implementing and evaluating collaborative filtering systems in recommender research. Due to the domain independence of collaborative filtering, we thus expected to ensure a sufficient degree of generalizability while being able to focus on a single domain. To implement the manual exploration interface and to provide users with informative and appealing item presentations, we enriched the dataset, which itself only comprises basic item data and associated user ratings. For this purpose, we used the *HetRec '11* dataset[15] and imported additional data from the *Internet Movie Database* (IMDb)[16]. This allowed us to present metadata such as genre, cast and director, but also plot descriptions and tags as well as movie posters. To facilitate decision making, we additionally provided tag clouds for the movies and the sets of representatives based on terms that were now available in the dataset.

Questionnaire Due to the lack of established questionnaires at the time we conducted the experiment, all questionnaire items were developed by ourselves. For this, we took into account prior research, so that the items finally reflected constructs that today are well established. Figure 3.2 provides an overview of the *subjective system aspects* (SSA) and the aspects related to *user experience* (EXP) as suggested by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12] for the evaluation of recommender systems. With respect to the former, we measured ■ *perceived recommendation quality*, ■ *perceived recommendation novelty*, ■ *trustworthiness*, and ■ *interaction adequacy*. With respect to the latter, we took into account ■ *perceived system effectiveness*, ■ *perceived control*, and ■ *usage effort*.

In addition, we assessed more *general aspects* (GEN): the ■ *overall satisfaction* of participants with each method, the ■ *suitability for different usage scenarios*, i.e. whether participants would like to use a method with or without a search goal, and the ■ *intention to use again* one of the methods. With respect to *personal characteristics* (PC), we also asked participants to provide ■ *demographic information* and to state their ■ *domain knowledge* regarding movies. To measure these constructs, we used again self-generated statements. For all items, we used 7-point Likert response scales. An overview of all constructs and items can be found in Appendix B.

Procedure First, we collected demographic data and asked participants about their interest in and familiarity with movies. After filling in this part of the *questionnaire* (step 1 in Figure 4.4), the *experimental phase* started. To ensure that all participants would interact with each of the four interfaces of the web application, we considered the underlying method as an *objective system aspect* (OSA), and set up the following conditions in a within-subject design:

POP In this condition, recommendations were generated using the ■ *popularity-based recommender*. Without further interaction, participants were immediately presented with a non-personalized set of the 6 most popular (and thus likely known) movies, for example, "Schindler's List", "Pulp Fiction", and "The Matrix".

[14] The *MovieLens 10M* dataset contains about 10 million ratings from more than 70 000 users for over 10 000 movies. It can be found here: `https://grouplens.org/datasets/movielens/10m/`

[15] The *HetRec '11* dataset extends the *MovieLens 10M* dataset and can be found here: `http://ir.ii.uam.es/hetrec2011/datasets.html`

[16] `https://www.imdb.com/`

MAN In this condition, participants were presented with the ■ *manual exploration interface*. We instructed them to use all available navigation aids, search and filtering mechanisms, to find movies they would like to watch. To be able to compare the quality of the resulting item set, 6 movies needed to be added to the shopping cart (see Figure A.1 in Appendix A), i.e. the same number as recommendations were generated in the other conditions.

RAT In this condition, the ■ *rating-based collaborative filtering recommender* was responsible for the results. For this, participants were first asked to rate 10 movies out of the 30 most popular items in the dataset on a 5-star rating scale. This is a common number of ratings for such a baseline in active learning experiments [cf. CGT12; ERR14]. Without any further interaction, the online updating mechanism processed these ratings, so that it was possible to present the top 6 personalized matrix factorization recommendations.

CHB In this condition, participants were presented with the dialog based on the ■ *choice-based preference elicitation method* (see Figure A.2). We elicited their preferences with respect to the 5 most important latent factors, i.e. participants were asked five times to choose an item set (or to indicate to have no preference regarding the current comparison). This differentiated well the items and turned out acceptable in terms of effort in the pretests (cf. Section 4.2.1). After participants completed the binary decision process, the top 6 recommendations were shown accordingly.

The four interfaces were presented to participants in counterbalanced order. Once they performed the respective *task* as explained above (2a) and obtained the corresponding *results* in the form of a selection of six movies (2b), they were asked to fill in the part of the *questionnaire* that was designed to measure the dependent variables for the respective method (2c).

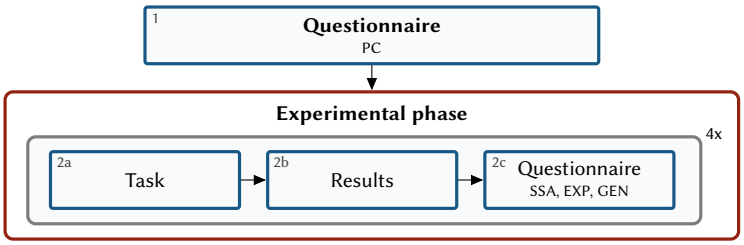

Figure 4.4 Overview of the procedure. See the text for a detailed description of the steps 1 and 2a–2c.

4.3.3 Results

In the following, we describe the *quantitative results* obtained through the questionnaire. We start with details on domain knowledge and overall satisfaction in the different conditions. Then, we step through all subjective system aspects as well as the aspects related to user experience. At the end of the section, we present further general results.

Quantitative results Regarding their domain knowledge, most participants reported that they like movies a lot: about 85 % agreed or totally agreed to the corresponding statement of the questionnaire, i.e. they provided a rating of at least 5 on the 7-point scale. Moreover, participants on average stated to watch 7.89 movies per month ($SD = 5.88$), rated their knowledge in general rather high ($M = 4.34$, $SD = 1.31$), and regarding recent movies a bit lower ($M = 3.83$, $SD = 1.64$).

Table 4.1 presents mean values and standard deviations with respect to the questionnaire constructs that we used to subjectively assess system aspects, user experience, and the suitability for different usage scenarios. These results already show the effects of the objective system aspect, outlining strengths and weaknesses of the individual methods. To confirm the indicated differences between the four conditions, we used one-factorial repeated-measures analyses of variance.[17] We observed p-values $< .001$ for all constructs, using Greenhouse-Geisser correction whenever sphericity was violated. These results are also shown in Table 4.1.

Table 4.1 Analysis of variance results ($df_1 = 3$, $df_2 = 102$)[18] for a comparison of the conditions in terms of subjective system aspects, user experience, and suitability for different usage scenarios. Higher values indicate better results on 7-point Likert response scales (*usage effort* is reversed accordingly). The values for the two additional baselines are grayed out, the best values in the two other conditions are highlighted in bold. η_p^2 represents effect size.

Construct		POP	MAN	RAT	CHB	F	p	η_p^2
Perceived recommendation quality	*M*	4.57	6.17	4.71	**5.54**	16.98	<.001	0.33
	SD	1.29	1.18	1.51	1.04			
Perceived recommendation novelty	*M*	2.91	2.91	**4.86**	4.80	16.18	<.001	0.32
	SD	1.77	1.84	1.70	1.69			
Trustworthiness	*M*	3.17	5.86	4.69	**5.31**	27.38†	<.001	0.45
	SD	1.65	1.68	1.41	1.23			
Interaction adequacy	*M*	6.77	5.37	6.20	**6.71**	18.97‡	<.001	0.36
	SD	0.65	1.37	1.08	1.67			
Perceived system effectiveness	*M*	1.63	1.66	4.94	**5.46**	96.15††	<.001	0.74
	SD	1.06	1.33	1.55	1.22			
Perceived control	*M*	1.92	6.31	4.51	**5.60**	110.49	<.001	0.77
	SD	0.71	1.30	1.72	1.33			
Usage effort	*M*	6.89	3.49	5.17	**6.20**	52.32†‡	<.001	0.61
	SD	0.32	1.93	1.60	0.83			
Suitability								
with a search goal	*M*	2.54	5.31	3.63	**4.14**	31.19	<.001	0.48
	SD	1.70	1.45	1.54	1.33			
without a search goal	*M*	4.91	3.34	5.14	**5.95**	21.71‡‡	<.001	0.39
	SD	1.70	2.04	1.38	0.97			

To analyze the differences in more depth and, in particular, to address our hypotheses, we subsequently performed pairwise comparisons of CHB with RAT, but also POP and MAN, by applying post hoc tests with Bonferroni correction. The results of these comparisons are reported below.

■ **Overall satisfaction** Before addressing the comparisons for the specific aspects, it is worth mentioning that the four interfaces were rated differently already with respect to overall satisfaction. A one-factorial repeated-measures analysis of variance yielded a main effect for condition, $F(3, 102) = 7.76$, $p < .001$, $\eta_p^2 = 0.19$. As illustrated in Figure 4.5, participants were more satisfied

[17] Note that in this thesis, we follow suggestions by, among others, the *American Statistician* regarding the dichotomization of p-values [WSL19]. Accordingly, we report only exact values and do not declare our findings as "significant", but take a more holistic approach when analyzing and interpreting the results.

in the CHB ($M = 5.43$, $SD = 1.31$) than in the POP ($M = 4.11$, $SD = 1.59$; $p = .002$), and, to a certain degree, in the RAT condition ($M = 4.54$, $SD = 1.76$; $p = .231$). On the other hand, there seemed to be no difference to the MAN condition ($M = 5.43$, $SD = 1.58$; $p = 1.000$).

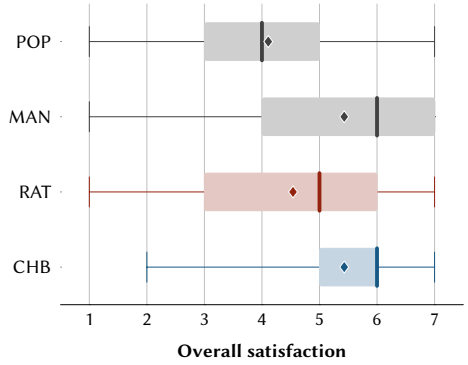

Figure 4.5 Box plot depicting the overall satisfaction of participants with the different methods: The thick vertical lines represent medians, the diamond signs mean values. Boxes extend from the 25th to the 75th percentile, whiskers correspond to minimum and maximum values.

■ **Perceived recommendation quality** The pairwise comparisons for perceived quality of the results provided first evidence of the superiority of our novel preference elicitation approach in terms of more specific aspects: As visible in Table 4.1, CHB achieved better results for this subjective system aspect than POP, yielding $p = .005$ in the post hoc test with medium effect size ($d_z = 0.61$), and RAT, yielding $p = .082$ and small to medium effect size ($d_z = 0.44$). Thus, H1 can be confirmed. Not surprisingly, self-chosen items in the MAN condition led to higher ratings than in the CHB condition, with $p = .100$ and small to medium effect size ($d_z = -0.43$).

■ **Perceived recommendation novelty** As indicated in Table 4.1, CHB performed very similar to RAT with respect to perceived novelty of the recommended items ($p = 1.000$, $d_z = -0.03$), which supports H2. As expected, CHB was rated much better in comparison to POP ($p < .001$, $d_z = 0.80$) and MAN ($p < .001$, $d_z = 0.79$).

■ **Trustworthiness** Also as expected, the manual exploration interface, which only did exactly what participants asked for, received better results with respect to trustworthiness than the more automated choice-based dialog (cf. Table 4.1). Still, the post hoc test indicated only a slight difference between MAN and CHB ($p = .520$, $d_z = -0.30$). The results of the two other baselines were in turn considerably in favor of CHB, with $p < .001$ for POP ($d_z = 1.14$) and $p = .030$ for RAT ($d_z = 0.53$). Thus, we can accept H3.

■ **Interaction adequacy** If user interaction was not necessary or reduced to few dialog steps, respectively, the adequacy of the interaction possibilities appeared very similar, with $p = 1.000$ for the comparison of POP and CHB ($d_z = -0.04$). However, in comparison to the entirely manual exploration in the MAN condition ($p < .001$, $d_z = 0.64$), and, to a certain degree, also to the rating-based preference elicitation in the RAT condition ($p = .036$, $d_z = 0.31$), the results in the CHB condition were superior (cf. Table 4.1). The latter confirms H4.

[18] Except for † ($df_1 = 2.48$, $df_2 = 84.30$), ‡ ($df_1 = 2.21$, $df_2 = 75.10$), †† ($df_1 = 2.52$, $df_2 = 85.74$), †‡ ($df_1 = 2.12$, $df_2 = 72.13$), and ‡‡ ($df_1 = 2.07$, $df_2 = 70.30$), adjusted due to violation of sphericity.

■ **Perceived system effectiveness** To operationalize the perception of system effectiveness, we asked participants whether they felt that the system learns their preferences. As suggested by an inspection of the mean values in Table 4.1, the post hoc tests confirmed the expected differences between CHB and POP ($p < .001$, $d_z = 2.00$) as well as between CHB and MAN ($p < .001$, $d_z = 1.92$). However, according to the post hoc comparison of CHB and RAT, the rating-based procedure for learning user preferences seemed to be perceived only slightly less effective, with $p = .711$ and small effect size ($d_z = 0.27$). For this reason, H5 cannot be accepted.

■ **Perceived control** With respect to the feeling of control over the system, results in the CHB condition were much better than in the POP ($p < .001$, $d_z = 2.09$), and also better than in the RAT condition ($p = .012$, $d_z = 0.57$). Thus, we can accept H6. As expected, the manual exploration interface received better results, even though mean values (cf. Table 4.1), post hoc test ($p = .086$), and effect size ($d_z = -0.43$) indicated only a small to medium difference between CHB and MAN.

■ **Usage effort** Conversely, the effort was perceived as much higher in the manual exploration interface than in our choice-based dialog (cf. Table 4.1). The post hoc comparison of CHB and MAN yielded $p < .001$ and large effect size ($d_z = 1.40$). Also, participants in the RAT condition felt not only less in control, but, at the same time, perceived the effort considerably worse than in the CHB condition ($p = .001$, $d_z = 0.72$). Thus, we can also accept H7. Since no interaction was required, POP unsurprisingly achieved a much better score ($p < .001$, $d_z = -0.83$).

■ **Suitability for different usage scenarios** Beyond that, Table 4.1 shows that the four methods were perceived differently well suited depending on whether users already formed a search goal or not: *With* a search goal, the suitability was rated much better in the CHB condition than in the POP condition ($p < .001$, $d_z = 0.91$), but only slightly better than in the RAT condition ($p = .374$, $d_z = 0.32$). In the MAN condition, in contrast but as expected, a superior result was achieved ($p = .003$, $d_z = -0.65$). On the other hand, *without* a search goal, CHB scored much better than all other methods, POP ($p = .009$), MAN ($p < .001$), and RAT ($p = .017$). As expected, the difference was particularly large between CHB and MAN, with an effect size of $d_z = 1.27$, as opposed to $d_z = 0.58$ or $d_z = 0.55$, respectively, when comparing CHB with POP and RAT.

■ **Intention to use again** Finally, we asked participants regarding their intention to use one of the interfaces again, respectively, more often if they were available. Figure 4.6 shows the agreement to the corresponding statements of the questionnaire. A one-factorial repeated-measures analysis of variance indicated considerable differences, $F(3, 102) = 21.14$, $p < .001$, $\eta_p^2 = 0.38$. Post hoc tests confirmed that CHB ($M = 5.74$, $SD = 0.82$) outperformed POP ($M = 3.43$, $SD = 1.63$; $p < .001$), MAN ($M = 3.23$, $SD = 1.75$; $p < .001$), and RAT ($M = 4.51$, $SD = 1.48$; $p = .002$).

4.3.4 Discussion

Overall, our novel preference elicitation dialog achieved better results than the three alternative methods in 15 out of 21 pairwise comparisons that were related to the specific system aspects and user experience. The disadvantages in the other comparisons were often small or to be expected. Also with respect to more general aspects such as participants' overall satisfaction and their intention to use one of the methods again, the choice-based method clearly outperformed the other

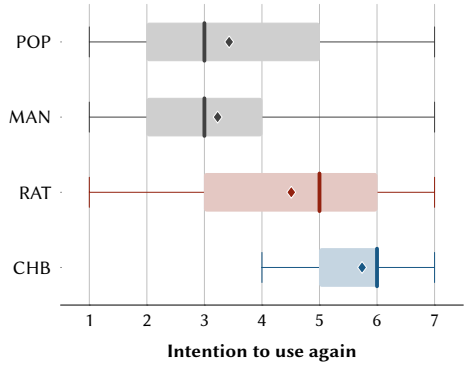

Figure 4.6 Box plot depicting the intention of participants to use again one of the methods: The thick vertical lines represent medians, the diamond signs mean values. Boxes extend from the 25th to the 75th percentile, whiskers correspond to minimum and maximum values.

interfaces. Most importantly, however, the comparisons with the interface that implemented the rating-based preference elicitation process supported almost all exploratory hypotheses.[19]

Quality and novelty As expected, the manual exploration interface showed advantages with respect to perceived quality, trustworthiness, perceived control, and suitability with a search goal. However, all differences seemed rather negligible, and the scores for the choice-based method were still in the upper end of the scale. On the other hand, the *recommendation quality* of the choice-based method was superior to the popularity-based and, in particular, the rating-based recommender (H1). Nevertheless, it is worth mentioning that the difference to the condition with the rating-based recommender, together with the standard deviation and the result of the post hoc test, suggest that for some participants, the recommendations also in this condition appeared to be of sufficient quality. Interestingly, the score of the popularity baseline was not much lower either. This confirms the results from earlier research that non-personalized recommendations often constitute an effective, easy-to-implement alternative [AB15].

In contrast, but as expected, the *novelty* of the mainstream items that were presented in this baseline condition was by definition much lower. The same was true for the items participants had in their shopping cart after using the manual exploration interface, as they had to find these items themselves. The rating-based recommender received the highest scores in this dimension. This is in line with literature showing that matrix factorization recommender systems include less popular, and thus more novel items in the result sets [Eks*15]. However, our approach performed only marginally worse (H2). Regardless of this result, both scores were relatively low compared to the other constructs. This may be explained by the fact that neither the rating-based nor the choice-based method were targeted at producing novel recommendations. Moreover, there is a well-known popularity bias in historical user-item interaction data [Ste11], and also, when users can choose between items to express their preferences [GW15; GW16].

[19] Note that we did not adjust for multiple hypothesis testing, which may have inflated the type-I error rate. However, these adjustments may be omitted in exploratory research such as ours, and are often considered impractical [cf. BL01; Rub17]. This particularly applies as we took into account more variables than we formulated hypotheses, and did not specify a priori the number of tests to be applied. For these reasons, the statistical findings can still be considered valuable for providing insights that help to answer our research questions. Nonetheless, further (confirmatory) research that accounts for such effects is clearly needed.

The results for novelty were in line with those for *system effectiveness*. To some degree, this appeared to be a consequence of the questionnaire item we used to measure this construct: Participants only felt that the system learns their preferences if they provided feedback in the form of ratings or comparisons, which led to high scores in this dimension. Within the sets of consequently suggested items, the number of novel items was naturally higher than within the sets of popular or manually selected items, i.e. in the two additional baseline conditions. Therefore, results might have been different with a different operationalization of this construct. On the other hand, more importantly, this definition emphasizes the success of our method in learning actual preferences, with at least similar performance as when using rating-based feedback (H5).

Trustworthiness and control For *trustworthiness* and *perceived control*, the picture was similar to perceived recommendation quality. The manual exploration interface performed best, the popularity-based recommender worst. Whereas this was expected—on the one hand, due to the direct influence participants had on the system's behavior, on the other hand, due to the presentation of popular items without any possibility to intervene—the lower scores achieved by the rating-based recommender again underlined the benefits of using comparisons: Apparently, participants were better able to attribute the results to their behavior in the choice-based dialog. In contrast, they had less trust in the standard matrix factorization recommender due to its black-box characteristics in relation to the personalization of the results based on the provided data (H3). At the same time, they felt more in control, although the interaction was in fact still limited in comparison to the manual exploration interface (H6).

Effort and interaction In terms of *perceived effort*, the scores achieved by the different methods were ordered the other way around: Only the non-interactive popularity baseline performed better than our choice-based dialog. Manually exploring the item database, but also providing ratings to single items, was perceived to require much more effort (H7). The results with respect to *interaction adequacy* were similar, though with smaller differences (H4). In particular, the difference between our method and the interface with the popularity-based recommender was almost negligible. However, given there were no interaction possibilities, it actually made no sense to ask regarding their adequacy or the required effort. In contrast, it is noteworthy that our novel dialog achieved better results in comparison to the interaction that is typical for initial preference elicitation: While frequently used in active learning [Rub*15; ERR16], in our experiment, the lower interaction adequacy of the rating-based mechanism came hand in hand with greater effort, although the number of interaction steps was in fact similar. This supports the assumption that regardless of the actual interaction, expressing preferences in the form of comparisons appears cognitively less demanding. At the same time, the increased effort did not lead to recommendations of higher quality. However, more advanced active learning approaches have been proposed in recent years [Rub*15; ERR16], even based on comparisons of pairs or groups of items [cf. RK12; BC12; CHT15; BR15; GW15]. Even though other authors in later studies were able to show that the aforementioned differences persist when taking into account these approaches as additional baselines [KRG18], future research is therefore clearly necessary.

Beyond that, the advantage of our method over manual exploration needs to be seen with a caveat: To compare the quality of the results, we asked participants in the manual condition to select six items, i.e. the same number of items as in the recommendation sets obtained in the other conditions. In real-world scenarios, however, users would have stopped searching as

soon as they found an appropriate item, i.e. effort would per se have been lower. Thus, further investigation is also needed to account for different task settings.

Limitations As seen before, not all differences between the choice-based method and the rating-based collaborative filtering recommender were large. Accordingly, the slightly negative assessment of the baseline in terms of perceived recommendation quality or system effectiveness might have been just the result of limited statistical power or lack of parameter optimization. Moreover, with more ratings available ex ante, recommendations would likely have been better and participants would have felt more strongly that the system learns their preferences. But, this also applies to our method, which would equally benefit from additional data.

Besides, several design decisions we made when implementing the choice-based dialog for this experiment need to be reconsidered: Although grounded theoretically and based on pilot studies, this includes the selection and ordering of the factors, the method for determining the factor representatives and their number, as well as the availability of the no-choice option. Also, we did not adapt the order of the side-by-side comparisons as described in Section 4.2.2, which may have distorted the results for some participants. While all these aspects are highly domain-specific, their improvement has the potential to positively affect the assessment of our method in future comparisons. For this, the usage of well-founded metrics in offline experiments might be a meaningful alternative to user experiments [cf. ERR14], as shown later by other authors who adopted the choice-based method for artwork recommendation [Ros*16]. Irrespective of the evaluation method, however, other active learning approaches (see above) need to be taken into account in order to evaluate the effectiveness of our method also in comparison to more advanced baselines.

Summary In light of the fact that the current baselines were taken from a wide range of established solutions for cold-start situations, it is yet safe to say—regardless of possible implementation issues—that the choice-based method produces recommendations that match the user's preferences very well. This is reflected in subjective system aspects such as perceived recommendation quality, and in aspects related to user experience, for example, perceived effectiveness. While the findings are exploratory, they suggest that participants appreciated the interactive dialog, in particular, in comparison to the elicitation of preferences via ratings, thanks to the increased trustworthiness and the effective interaction that did not come at the expense of higher usage effort. In line with later research [e.g. KRG18], this especially applied to situations without a search goal, although our novel method appeared useful in other scenarios as well. Together with the more general results regarding overall satisfaction and intention to use the method again, we can thus conclude that it is actually possible to *exploit the semantics* in latent factor models for the purpose of *improving user control and experience* to a much larger degree than currently practiced in model-based collaborative filtering (RQ1).

CHAPTER 5

> "We're entering a new world in which data may be more important than software."
>
> — *Tim O'Reilly, Irish businessman*

Boosting matrix factorization with content information

In this chapter, we lay the ground for implementing extensions to model-based collaborative filtering systems that will help us to further improve user control and experience. The choice-based method introduced in the previous chapter has illustrated that the semantics contained in latent factor models allow eliciting user preferences in a more interactive fashion. The same applies to some of the approaches proposed by others as well as ourselves in later work (cf. Section 2.2.5). Whereas these approaches have the same minimal data requirements as collaborative filtering in general, we have argued in Section 3.1.2 that leveraging item-related information in addition to standard collaborative filtering feedback data may contribute to a higher degree of user control and more expressive interaction mechanisms, not only to recommendation accuracy. Yet, before we address these mechanisms, we present the basis for implementing them, *content-boosted matrix factorization* [DLZ15; DLZ16a; DLZ16b]: This method enables us to integrate latent factor models with any type of item-related side information, and to subsequently associate the considered content attributes with each individual user. By making the latent factors in this way accessible from the user interface, we obtain a starting point for addressing our second research question, as described in Section 3.2.2. Next, we first describe the *background*, explain the *method* in more detail, and introduce an implementation *framework called TagMF* [Loe*19b]. In addition, to show the general effectiveness of our method, we describe a series of *offline experiments*, including a qualitative analysis of a resulting factor model.

5.1 Background

Melville, Mooney, and Nagarajan [MMN02], who presented one of the first hybrid recommender systems following the earliest attempts to hybridization (see Section 2.1.5), coined the term of "content-boosted" collaborative filtering [Bur07]. For them, this meant filling the sparse user vectors of a typical user-item matrix with ratings by means of a simple, separately trained bag-of-words model. Then, they generated recommendations by applying a standard memory-based collaborative filtering algorithm on the resulting (now dense) user-item matrix.

Inspiration from the potential of item-related information This example of feature augmentation emphasized early the potential for improving recommendations by considering side

information. Moreover, the necessity of finding neighbors who have ratings in common, a problem often associated with conventional memory-based techniques, vanished without further ado. However, performance and scalability issues remained the same—or became even worse due to the higher density, and thus the need to process a larger amount of (calculated) user-item interaction data. On the other hand, there also exists a broad range of model-based techniques. The usage of these techniques naturally circumvents most of the problems related to performance and scalability, and is by itself beneficial in terms of objective recommendation quality.

Many experiments have shown that integrating additional information constitutes one of the most promising avenues for improving the state of the art in this area even further—at least when it comes to accuracy as measured in offline experiments (see Section 2.2.4). Yet, there is still more potential, which has been acknowledged whenever *latent factor models* were used for other purposes, such as alleviating the cold-start problem [PT09; Gan*10], increasing explainability [ML13; Zha*14] or providing visualizations [Ném*13; BJG13] (see Section 2.2.5). Given our main goal, we also target *this* kind of models. However, the literature review equally shows that most enhancements, in contrast to the aforementioned exceptions, have been proposed without taking a user-oriented perspective: Focusing only on the quality of the models, the authors made no effort to interweave the additionally provided data and the latent factors in a way that the relations in between are still accessible after the offline learning phase is over. Worse, the effects of side information on the subjective assessment of aspects such as recommendation quality or on user experience have not yet even been investigated.

Realization with matrix factorization On the other hand, there is a wide and, in particular, very diverse range of interactive recommending approaches that rely *exclusively* on other types of input data than standard user-item feedback (see Section 2.3.2.1). These approaches enable users to control the recommendations without the necessity (but also the possibility) of providing, for example, rating-based feedback. Given the success of our interactive preference elicitation dialog for matrix factorization systems introduced in the previous chapter, designed *without* requiring anything other than plain user-item feedback, it thus appears to be the next most natural step to leverage item-related information in addition to this typical, but often criticized form of feedback, in order to add further interactive features to these systems. In addition, being able to take advantage of the maturity of model-based collaborative filtering algorithms, and consequently, long-term user profiles, should allow for supporting users not only in cold-start situations, but also *at any point* later in the recommendation process.

For this, we consider the approach proposed by Forbes and Zhu [FZ11] as a promising point of departure: As described in Section 2.2.4.2, their method equally boosts matrix factorization with content attributes, yet with the benefit of maintaining accessible relations between these attributes and the factors thanks to its *regression-constrained formulation* of the optimization problem. Thus, these content-related associations may be exposed in the user interface for practical purposes such as letting users intervene in the underlying model or conveying its semantics. Accordingly, when we talk in this thesis about "content boosting", we refer to this method. However, as this method as well as the enhancements we are going to present are independent of algorithmic details, we point to Section 2.2 for aspects such as regularization or biases.

5.2 Method

In the following, independent of the particular goal of content boosting, we first explain how to actually *learn such a latent factor model*. Subsequently, we also address the question of how to *associate users with the considered attributes*, which is necessary for our purposes under the assumption that the employed side information is only available in the form of attributes that describe the content of the items. Later, we use tags assigned to the items by the user community. Here, we describe these two steps on a more general level, underlining that *any* type of content data may be used for implementing the interactive features we describe in the next chapter.

5.2.1 Learning a content-boosted model

First, we follow the underlying approach by Forbes and Zhu [FZ11] to integrate a standard matrix factorization model with item-related side information. For this, we use a set of attributes H, and define ${}^{\text{i}}\mathbf{H} \in \mathbb{R}^{|I| \times |H|}$ as a matrix representing how strongly each item is related to these attributes. Accordingly, an entry h_{ia} of ${}^{\text{i}}\mathbf{H}$ describes on a continuous scale from 0.0 (not relevant) to 1.0 (very relevant) the degree to which attribute a is relevant for item i.

Redefining matrix factorization Next, in the underlying approach, the item side is altered by extending the item-factor matrix $\mathbf{Q}$ with these attributes, as visible in the updated matrix factorization formulation in (2.15) in Section 2.2.4.2. We, in addition, extend the user-factor matrix $\mathbf{P}$ and define ${}^{\text{u}}\mathbf{H} \in \mathbb{R}^{|U| \times |H|}$. This way, we are also able to represent the relations between users and attributes, i.e. how relevant each attribute a is for a user u. From that, we *redefine* the original matrix factorization model given in (2.4) as follows:

$$\mathbf{R} \approx \mathbf{P}\mathbf{Q}^{\mathsf{T}} = {}^{\text{u}}\mathbf{H}\,{}^{\text{u}}\boldsymbol{\Theta}({}^{\text{i}}\mathbf{H}\,{}^{\text{i}}\boldsymbol{\Theta})^{\mathsf{T}}, \tag{5.1}$$

where ${}^{\text{u}}\boldsymbol{\Theta} \in \mathbb{R}^{|H| \times k}$ associates the attributes with the factors as seen from the user side, and ${}^{\text{i}}\boldsymbol{\Theta} \in \mathbb{R}^{|H| \times k}$ is the item equivalent. This again represents a regression-constrained formulation of the matrix factorization problem, where each of the k factors is a function of the attributes.

Item-related information In Section 2.2.4.1, we provided an overview of the variety of data that has been fed successfully into matrix factorization algorithms in addition to standard collaborative filtering data. However, apart from the decision on the right data, an important consideration is that side information might be available only for users *or* for items. Forbes and Zhu [FZ11] and Nguyen and Zhu [NZ13] used predefined metadata for the items. With our goal of using *item-related* information to provide users the option to influence latent factor models in a more expressive manner, we similarly assume that some kind of content data is known *a priori*, and a corresponding matrix ${}^{\text{i}}\mathbf{H}$ can either be derived separately, or is available right up front. The only requirement is that a numerical representation can be determined so that the entries of this matrix eventually hold the relevance scores for all items and attributes on a continuous scale, and that the attributes relate to the item information space but also to the user information space in a meaningful way.

User-related information In contrast to matrix ${}^{\mathrm{i}}\mathbf{H}$, we consider the corresponding matrix for users, ${}^{\mathrm{u}}\mathbf{H}$, to be *unknown*. In many scenarios, this seems to be an important consideration with a valuable result: While information regarding the items is often available—be it in the form of predefined metadata, expert knowledge or user-generated tags—users are often unknown, for example, in cold-start situations, when they do not want to provide the required information, or cannot easily be motivated to do so. Yet, information specific for users is naturally not required by the original method, but neither by our proposed enhancements: We treat the whole term ${}^{\mathrm{u}}\mathbf{H}{}^{\mathrm{u}}\mathbf{\Theta}$ implicitly at this step by just learning the user-factor matrix $\mathbf{P}$ as described for standard matrix factorization in Section 2.2.2.1. With the resulting constrained equation, we can formulate the following *minimization problem* as done by Forbes and Zhu [FZ11]:

$$\begin{aligned} \min_{\mathbf{P},{}^{\mathrm{i}}\mathbf{\Theta}} \sum_{r_{ui}\in R} e_{ui}^2 + \lambda\Big(\sum_{u\in U}\|\mathbf{p}_u\|^2 + \|{}^{\mathrm{i}}\mathbf{\Theta}\|^2 + \sum_{u\in U}\|b_u\| + \sum_{i\in I}\|b_i\|\Big)\,, \\ \text{with } e_{ui}^2 \coloneqq (r_{ui} - \mathbf{p}_u^{\mathsf{T}}\,{}^{\mathrm{i}}\mathbf{\Theta}^{\mathsf{T}}\mathbf{h}_i - \mu - b_u - b_i)^2\,, \end{aligned} \tag{5.2}$$

λ controlling the regularization, and R being the set of all user-item tuples for which feedback exists. Note that we use matrix notation here, i.e. vectors are represented as column matrices.

Performing the minimization Next, to be able to apply stochastic gradient descent in the subsequent step for minimizing the squared prediction error, we need to build the *partial derivatives* with respect to the optimization parameters. This can be done as follows:

$$\begin{aligned} \frac{\partial}{\partial p_{uf}} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|{}^{\mathrm{i}}\mathbf{\Theta}\|^2 + b_u^2 + b_i^2) &= -2e_{ui}\cdot[{}^{\mathrm{i}}\mathbf{\Theta}^{\mathsf{T}}\mathbf{h}_i]_f + 2\lambda p_{uf} &&\propto -e_{ui}\cdot[{}^{\mathrm{i}}\mathbf{\Theta}^{\mathsf{T}}\mathbf{h}_i]_f + \lambda p_{uf}\,, \\ \frac{\partial}{\partial {}^{\mathrm{i}}\theta_{af}} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|{}^{\mathrm{i}}\mathbf{\Theta}\|^2 + b_u^2 + b_i^2) &= -2e_{ui}\cdot p_{uf}\cdot h_{ia} + 2\lambda\,{}^{\mathrm{i}}\theta_{af} &&\propto -e_{ui}\cdot p_{uf}\cdot h_{ia} + \lambda\,{}^{\mathrm{i}}\theta_{af}\,, \\ \frac{\partial}{\partial b_u} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|{}^{\mathrm{i}}\mathbf{\Theta}\|^2 + b_u^2 + b_i^2) &= -2e_{ui} + 2\lambda b_u &&\propto -e_{ui} + \lambda b_u\,, \\ \frac{\partial}{\partial b_i} e_{ui}^2 + \lambda(\|\mathbf{p}_u\|^2 + \|{}^{\mathrm{i}}\mathbf{\Theta}\|^2 + b_u^2 + b_i^2) &= -2e_{ui} + 2\lambda b_i &&\propto -e_{ui} + \lambda b_i\,. \end{aligned} \tag{5.3}$$

From this, we can define the *update rules* for adjusting each vector $\vec{p}_u$ and the entire matrix ${}^{\mathrm{i}}\mathbf{\Theta}$ for each given $r_{ui}\in R$. As described in Section 2.2.3.2, these adjustments are done in a stepwise manner in the opposite direction of the gradient, scaled by the learning rate η:

$$\begin{aligned} p_{uf} &\leftarrow p_{uf} - \eta\big(-e_{ui}\cdot[{}^{\mathrm{i}}\mathbf{\Theta}^{\mathsf{T}}\mathbf{h}_i]_f + \lambda p_{uf}\big) &&= p_{uf} + \eta\big(e_{ui}\cdot[{}^{\mathrm{i}}\mathbf{\Theta}^{\mathsf{T}}\mathbf{h}_i]_f - \lambda p_{uf}\big)\,, \\ {}^{\mathrm{i}}\theta_{af} &\leftarrow {}^{\mathrm{i}}\theta_{af} - \eta\big(-e_{ui}\cdot p_{uf}\cdot h_{ia} + \lambda\,{}^{\mathrm{i}}\theta_{af}\big) &&= {}^{\mathrm{i}}\theta_{af} + \eta\big(e_{ui}\cdot p_{uf}\cdot h_{ia} - \lambda\,{}^{\mathrm{i}}\theta_{af}\big)\,, \\ b_u &\leftarrow b_u - \eta\big(-e_{ui} + \lambda b_u\big) &&= b_u + \eta\big(e_{ui} - \lambda b_u\big)\,, \\ b_i &\leftarrow b_i - \eta\big(-e_{ui} + \lambda b_i\big) &&= b_i + \eta\big(e_{ui} - \lambda b_i\big)\,. \end{aligned} \tag{5.4}$$

Note that each update of a $\vec{p}_u$ vector is done by updating its components for each factor f. Matrix ${}^{\mathrm{i}}\mathbf{\Theta}$ is updated based on each single attribute a. Therefore, the update rules are shown for each individual entry, p_{uf} and ${}^{\mathrm{i}}\theta_{af}$, respectively.

As an alternative to such an objective function for rating prediction, one can also use a "learning to rank" criterion as described in Section 2.2.2.2. The *partial derivatives* for an adapted version of Bayesian personalized ranking may look as follows:

$$
\begin{aligned}
&\frac{\partial}{\partial p_{uf}} - \ln(1 + \mathrm{e}^{-\hat{r}_{uij}}) - \lambda(\|\mathbf{p}_u\|^2 - \|{}^{\mathrm{i}}\boldsymbol{\Theta}\|^2 - b_i^2 - b_j^2) \\
&\quad \propto \quad \frac{\mathrm{e}^{-\hat{r}_{uij}}}{1 + \mathrm{e}^{-\hat{r}_{uij}}} \cdot \left([{}^{\mathrm{i}}\boldsymbol{\Theta}^{\mathsf{T}}\mathbf{h}_i]_f - [{}^{\mathrm{i}}\boldsymbol{\Theta}^{\mathsf{T}}\mathbf{h}_j]_f\right) - \lambda p_{uf} = \frac{1}{1 + \mathrm{e}^{\hat{r}_{uij}}} \cdot \left([{}^{\mathrm{i}}\boldsymbol{\Theta}^{\mathsf{T}}\mathbf{h}_i]_f - [{}^{\mathrm{i}}\boldsymbol{\Theta}^{\mathsf{T}}\mathbf{h}_j]_f\right) - \lambda p_{uf}\,, \\
&\frac{\partial}{\partial {}^{\mathrm{i}}\theta_{af}} - \ln(1 + \mathrm{e}^{-\hat{r}_{uij}}) - \lambda(\|\mathbf{p}_u\|^2 - \|{}^{\mathrm{i}}\boldsymbol{\Theta}\|^2 - b_i^2 - b_j^2) \\
&\quad \propto \quad \frac{\mathrm{e}^{-\hat{r}_{uij}}}{1 + \mathrm{e}^{-\hat{r}_{uij}}} \cdot (p_{uf} \cdot h_{ia} - p_{uf} \cdot h_{ja}) - \lambda {}^{\mathrm{i}}\theta_{af} = \frac{1}{1 + \mathrm{e}^{\hat{r}_{uij}}} \cdot (p_{uf} \cdot h_{ia} - p_{uf} \cdot h_{ja}) - \lambda {}^{\mathrm{i}}\theta_{af}\,, \\
&\quad \ldots
\end{aligned}
\tag{5.5}
$$

The remaining derivatives for biases can be found in Appendix D. The estimator $\hat{r}_{uij}$ is used as defined in (2.9), but with $\hat{r}_{uk} \coloneqq \mathbf{p}_u {}^{\mathrm{i}}\boldsymbol{\Theta}^{\mathsf{T}}\mathbf{h}_k + b_k$. For stochastic gradient ascent, the *update rules* can be defined as follows, allowing to adjust the vector $\vec{p}_u$, the matrix ${}^{\mathrm{i}}\boldsymbol{\Theta}$, and the item biases for each sampled triple that represents whether a user u prefers an item i over an item j:

$$
\begin{aligned}
p_{uf} &\leftarrow p_{uf} + \eta\left(\frac{1}{1 + \mathrm{e}^{\hat{r}_{uij}}} \cdot \left([{}^{\mathrm{i}}\boldsymbol{\Theta}^{\mathsf{T}}\mathbf{h}_i]_f - [{}^{\mathrm{i}}\boldsymbol{\Theta}^{\mathsf{T}}\mathbf{h}_j]_f\right) - \lambda p_{uf}\right), \\
{}^{\mathrm{i}}\theta_{af} &\leftarrow {}^{\mathrm{i}}\theta_{af} + \eta\left(\frac{1}{1 + \mathrm{e}^{\hat{r}_{uij}}} \cdot (p_{uf} \cdot h_{ia} - p_{uf} \cdot h_{ja}) - \lambda {}^{\mathrm{i}}\theta_{af}\right), \\
b_i &\leftarrow b_i + \eta\left(\frac{1}{1 + \mathrm{e}^{\hat{r}_{uij}}} - \lambda b_i\right), \\
b_j &\leftarrow b_j + \eta\left(\frac{-1}{1 + \mathrm{e}^{\hat{r}_{uij}}} - \lambda b_j\right).
\end{aligned}
\tag{5.6}
$$

5.2.2 Associating users with content attributes

At this point, the semantics contained in the latent dimensions have already been transferred into a more comprehensible information space by utilizing the regression-constrained approach on the item side. Thanks to the enhancements we proposed to the original method by Forbes and Zhu [FZ11], we can now associate the content attributes with *users*, although we considered the additional information to be available only for items. This is possible as the way matrix factorization models are learned ensures per definition that both users and items are mapped into a joint factor space, i.e. the characteristics reflected by a factor f are equally related to users and items (cf. Section 2.2.1). The regression coefficients hence describe attribute-factor relations in general, for users as well as for items. Accordingly, the previously implicitly assumed matrix ${}^{\mathrm{u}}\boldsymbol{\Theta}$ is equivalent to matrix ${}^{\mathrm{i}}\boldsymbol{\Theta}$, such that:

$$
{}^{\mathrm{u}}\boldsymbol{\Theta} = {}^{\mathrm{i}}\boldsymbol{\Theta} =: \boldsymbol{\Theta}\,. \tag{5.7}
$$

Since ${}^{\mathrm{u}}\mathbf{H}$ is consequently the only unknown left in (5.1), we can now solve for this matrix in order to obtain the equivalents of the scores stored for the items in ${}^{\mathrm{i}}\mathbf{H}$. For this, given $\boldsymbol{\Theta}$ is

generally not a square matrix, we need to apply singular value decomposition to calculate its pseudoinverse $\boldsymbol{\Theta}^+$, i.e. the Moore-Penrose generalization of the inverse matrix [Moo20; Pen55]. This yields $\mathbf{X} \in \mathbb{R}^{|H|\times|H|}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{|H|\times k}$ and $\mathbf{Y} \in \mathbb{R}^{k\times k}$. Hence, $\boldsymbol{\Theta}^+$ is defined by $\mathbf{Y}\boldsymbol{\Sigma}^+\mathbf{X}^\mathsf{T}$, such that:

$$\begin{aligned} \mathbf{P} &= {}^{\mathrm{u}}\mathbf{H}\boldsymbol{\Theta} \quad \Leftrightarrow \\ \mathbf{P} &= {}^{\mathrm{u}}\mathbf{H}\mathbf{X}\boldsymbol{\Sigma}\mathbf{Y}^\mathsf{T} \Leftrightarrow \\ {}^{\mathrm{u}}\mathbf{H} &= \mathbf{P}\mathbf{Y}\boldsymbol{\Sigma}^+\mathbf{X}^\mathsf{T} \Leftrightarrow \\ {}^{\mathrm{u}}\mathbf{H} &= \mathbf{P}\boldsymbol{\Theta}^+ . \end{aligned} \tag{5.8}$$

As intended, it is thus effectively possible to determine ${}^{\mathrm{u}}\mathbf{H}$, which then represents the interest of all users with respect to all attributes, i.e. basically the *calculated* counterpart of the *given* item-related information integrated as explained in the previous section. Note this step is somewhat similar to the approach by Becerra, Jimenez, and Gelbukh [BJG13], but we retain the latent knowledge instead of completely replacing the item-factor matrix by an item-attribute matrix.

Since $\boldsymbol{\Theta}\boldsymbol{\Theta}^\mathsf{T}$, holding the general attribute-factor relations, is a square diagonalizable matrix, we can finally apply *eigendecomposition* to represent our redefined model from (5.1) as follows:

$$\begin{aligned} \mathbf{R} \approx {}^{\mathrm{u}}\mathbf{H}\boldsymbol{\Theta}({}^{\mathrm{i}}\mathbf{H}{}^{\mathrm{i}}\boldsymbol{\Theta})^\mathsf{T} &= {}^{\mathrm{u}}\mathbf{H}\boldsymbol{\Theta}\boldsymbol{\Theta}^\mathsf{T}{}^{\mathrm{i}}\mathbf{H}^\mathsf{T} \\ &= {}^{\mathrm{u}}\mathbf{H}\mathbf{X}\boldsymbol{\Sigma}\mathbf{Y}^\mathsf{T}\mathbf{Y}\boldsymbol{\Sigma}^\mathsf{T}\mathbf{X}^\mathsf{T}{}^{\mathrm{i}}\mathbf{H}^\mathsf{T} \\ &= {}^{\mathrm{u}}\mathbf{H}\mathbf{X}\boldsymbol{\Sigma}\mathbf{1}\boldsymbol{\Sigma}^\mathsf{T}\mathbf{X}^\mathsf{T}{}^{\mathrm{i}}\mathbf{H}^\mathsf{T} \\ &= {}^{\mathrm{u}}\mathbf{H}\mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\mathsf{T}{}^{\mathrm{i}}\mathbf{H}^\mathsf{T} . \end{aligned} \tag{5.9}$$

Consequently, the *eigenvectors* in $\mathbf{E}$ hold the importance of every attribute with respect to a certain direction. The diagonal matrix $\boldsymbol{\Lambda}$ contains the *eigenvalues* of $\boldsymbol{\Theta}\boldsymbol{\Theta}^\mathsf{T}$ in non-increasing order. Since $\boldsymbol{\Theta}\boldsymbol{\Theta}^\mathsf{T}$ is symmetric, eigenvectors are chosen orthogonal to each other. Latent factors are thus incorporated into the attribute information space by stretching it along the eigenvector directions according to the magnitude of the corresponding eigenvalues.

For generating recommendations, a function similar to $s(i|u)$ from (2.3) in Section 2.2.1 can then be used. But, in the next chapter, we will see that the redefined matrix factorization model also allows to realize more interactive model-based collaborative filtering systems by using an adapted recommendation function. First, however, we present a *framework* for implementing such a model, and describe an *offline evaluation* to validate the general effectiveness of our method—in quantitative and qualitative manner—before applying it in a user-oriented environment.

5.3 Framework

Next, to allow for an effective development of model-based collaborative filtering recommender systems based on our content-boosted matrix factorization method, we present a *Java software framework* for implementing the method exactly as described in the previous section. While the framework can be used with any kind of attributes—provided a numerical representation exists or can be derived—we rely in this thesis for all demonstration purposes on user-generated tags. For this reason, the framework is called *TagMF*. It consists of the following three packages:

CORE The main algorithmic package contains the *recommendation* functionality, implemented on top of the *Apache Mahout* recommender library.[20]

EVAL The *evaluation* package, implemented on top of the *RiVal* benchmarking toolkit,[21] can be utilized to run structured offline experiments.

DEMO The *demonstration* package contains a prototypical web application that showcases the interactive features that become possible by means of our method.

Not only is the framework completely generic with respect to the attributes describing the items, but also the domain of these items, due to the independence of the underlying collaborative filtering approach. Without loss of generality, the web application included in the *demo* package as well as all subsequent prototypes we present in this thesis are, however, implemented based on movies. The availability of the *MovieLens* datasets,[22] large and representative collections of item data, associated user ratings, and, in particular, tags and corresponding relevance values, was the main driver behind this decision. Also the examples provided in the *core* and *eval* package make use of movies and tags. Yet, they are implemented in a way that setting up a recommender or running an evaluation is easily possible with other background data as well.

In the following, we briefly describe how to *initialize and use a recommender* with the help of the *core* package, and how to subsequently *conduct an offline evaluation* to study its performance using the functionality provided by the *eval* package. Since the integrated recommendation platform we present later in this thesis is essentially an extension of the web application contained in the *demo* package, we omit details regarding this package, but refer to Chapter 8.

5.3.1 Initializing and using a recommender

As a point of departure for the *core* package, we used the stochastic gradient descent implementation called `ParallelSGDFactorizer` from the *Apache Mahout* recommender library.[20] This implementation represents a typical matrix factorization algorithm as described in Section 2.2, specifically the one presented by Takács, Pilászy, Németh, and Tikk [Tak*09]. We extended this implementation according to our method exactly as described in this chapter. Moreover, we added online updating as proposed in [RS08]. As a by-product, we also set up a Bayesian personalized ranking variant in accordance with [Ren*09]. The *initialization* of the resulting `RegressionConstrainedSGDFactorizer` may be done as shown in Listing 5.1.

Listing 5.1 Initialization of a recommender.

```
input:  userItemModel: model of existing user-item feedback
        itemAttributes: item-attribute array
        numFactors, lambda, numIters: predefined constants

1 factorizer  =   new RegressionConstrainedSGDFactorizer(userItemModel,
2                   itemAttributes, numFactors, lambda, numIters);
3
4 CandidateItemsStrategy candidateItemsStrategy
5             =   new AllUnknownItemsCandidateItemsStrategy();
6 PersistenceStrategy persistenceStrategy
7             =   new NoPersistenceStrategy();
```

[20] https://mahout.apache.org/

[21] http://rival.recommenders.net/

[22] https://grouplens.org/datasets/movielens/

```
8 RegressionConstrainedFactorization factorization
9             =   factorizer.factorize();
10 recommender =   new ContentBoostedSVDRecommender<String>(userItemModel,
11                   candidateItemsStrategy, persistenceStrategy, factorization);
```

First, the small data requirements of our content boosting method need to be fulfilled. For this, the `factorizer` is initialized with a `userItemModel` that represents a regular user-item matrix $\mathbf{R}$, and an array of `itemAttributes` that is equal to the matrix ${}^{\text{i}}\mathbf{H}$ (lines 1–2). Once this is done and some library-specific assignments are made (lines 4–7), the `factorize()` method is called, which is responsible for the actual optimization task (lines 8–9). This method iterates over all user-item pairs for which feedback data are available and determines the prediction errors as shown in pseudo code in Listing 2.1 in Section 2.2.3.2. But, the updates of the user-factor vectors $\vec{p}_u$ and of matrix ${}^{\text{i}}\mathbf{\Theta}$ are performed according to the novel rules given in (5.4). Afterwards, the `RegressionConstrainedFactorization`, i.e. the new model consisting of ${}^{\text{u}}\mathbf{H}$, ${}^{\text{i}}\mathbf{H}$ and ${}^{\text{i}}\mathbf{\Theta}$, is handed over to a `ContentBoostedSVDRecommender` (lines 10–11). This is an extension of the standard `SVDRecommender` that is used by *Apache Mahout* for matrix factorization recommendations (despite its name not to be confused with an actual singular value decomposition, cf. Section 2.2.3.1). Given this `recommender`, which works with any type of attribute (including `String` values that may represent user-generated tags), one can now *ask for recommendations* for any particular user, as illustrated in Listing 5.2.

Listing 5.2 Generation of recommendations and weighting of attributes.

```
  input:  recommender: recommender initialized as shown before
          attributes: list of integrated attributes
          userId: id of the current user
          numRecs: predefined constant

1 recommendations = recommender.recommend(userId, numRecs);
2 print(recommendations);
3
4 Map<Attribute<String>, Double> weightedAttributes = new HashMap<>();
5 for (Attribute<String> attribute : attributes)
6   if(attribute.getValue().equals("sci-fi"))
7     weightedAttributes.put(attribute, 1.0);
8   else
9     weightedAttributes.put(attribute, 0.0);
10 recommender.updateWeights(userId, weightedAttributes);
11
12 recommendations = recommender.recommend(userId, numRecs);
13 print(recommendations);
```

First, recommendations are generated for the user with the specified `userId` (line 1) and printed in some way in the user interface (line 2). In the background, this essentially calls the standard recommendation function $s(i|u)$ as shown in (2.3) in the introduction to matrix factorization in Section 2.2.1. Accordingly, the recommendations are the result of dot multiplications of the user's latent factor vector and the vectors of all items he or she has not yet rated. However, with content-boosted matrix factorization, we can additionally set weights for the attributes integrated into the underlying latent factor model and adjust the vector representation of the user's long-term preferences according to individual short-term goals. While in real-world scenarios, these weights would be set in the interface, they are defined manually here: with the maximum

value for "sci-fi", and the minimum value for all other tags (lines 4–9). Then, based on these `weightedAttributes`, the call of the `updateWeights()` method (line 10) performs exactly what is described as one of the application possibilities in the next chapter (see Section 6.2.2). Note that for some of the other possible applications, there is also a more generic variant of this method, which directly takes as input an adjusted user-attribute vector $\vec{h}_u$ instead of a set of weights. Either way, updated recommendations can be obtained afterwards. These recommendations now reflect both the user's existing profile and the provided ad hoc preferences, such as here the weights for the considered content attributes (lines 12–13).

5.3.2 Conducting an offline evaluation

The *RiVal* benchmarking toolkit introduced by Said and Bellogín [SB14b] allows running offline experiments with all kinds of recommendation algorithms in a structured and automated manner.[21] The toolkit implements common metrics such as *root mean square error* (RMSE), one of the most widely used, yet increasingly criticized variants for measuring objective accuracy, and *normalized discounted cumulative gain* (NDCG), a popular variant from information retrieval for determining the quality of a ranking [GS15]. In addition, mechanisms are provided for efficiently handling large datasets as well as for cross validation and taking care of other aspects that help ensure experimental validity. We wrapped the entire functionality in the *eval* package in order to facilitate the interplay with *Apache Mahout* recommender algorithms in general, and our implementations from the *core* package in particular. Listing 5.3 shows a corresponding *evaluation protocol* for a `RegressionConstrainedSGDFactorizer` as used in the example above, including the setup of the whole experiment and the subsequent execution of several evaluation tasks.

Listing 5.3 Setup and execution of an experiment.

```
  input:  attributes: list of attributes to consider

1 RivalEvaluator.getInstance().prepareSplits();
2
3 configuration = new RivalConfiguration();
4 configuration.setFactorizer("[...].tagmf.eval.recommender
5         .RivalMahoutRegressionConstrainedSGDFactorizer");
6 configuration.setAttributes(attributes);
7 configuration.setNumFactors(5);
8
9 evaluationResults = RivalEvaluator.getInstance().runEvaluation(configuration);
10 print(evaluationResults);
11
12 configuration.setNumFactors(10);
13 evaluationResults = RivalEvaluator.getInstance().runEvaluation(configuration);
14 print(evaluationResults);
```

The first step follows the typical procedure of the *RiVal* framework for conducting cross-validated experiments: The `prepareSplits()` method is called at the beginning (line 1). This method prepares the input data and creates a series of training and test datasets for the cross validation folds (using the provided `IterativeCrossValidationSplitter`). This requires that the number of folds, but also the dataset files for user-item feedback and item-attribute relevance scores, are set in advance, which is here omitted for the sake of simplicity. Afterwards, the `RivalConfiguration` is set up to parameterize the `RivalEvaluator` (line 3). In the example,

we declare the factorizer, the content attributes that should be integrated when the latent factor model is learned, and the number of dimensions with which this should happen (lines 4–7). The factorizer represents an adopted version of our previously explained `RegressionConstrainedSGDFactorizer`, specifically tailored for the *RiVal* environment. The `runEvaluation()` method of the `RivalEvaluator`, which is called next (line 9), encapsulates the steps that follow according to the typical *RiVal* procedure: 1) calculate recommendations, which in our case includes learning a new factor model for each fold and calculating the predictions, 2) write strategy files for arranging the predicted scores in a way that allows to use them in the next step more efficiently, and 3) compute a set of metrics, in our case, *MAE*, *RMSE*, *NDCG*, *MAP*, precision and recall [cf. GS15]. Once these steps are completed, the results of the computed metrics are printed out (line 10). Afterwards, to examine a different parameterization, we change the configuration and run the same task again, but with a larger number of latent dimensions (lines 12–14).

5.4 Offline evaluation

Before addressing the application of content-boosted matrix factorization for establishing interactive features, and investigating its benefits from a user perspective, it is necessary to validate the general effectiveness. Accordingly, we present a *performance analysis* based on our *TagMF* framework to test whether the findings of other authors regarding the usefulness of side information in collaborative filtering settings (cf. Section 2.2.4) can be confirmed when analyzing our method by means of typical accuracy metrics in offline experiments. Moreover, we present a *qualitative analysis* of one of the resulting factor models to add evidence to the assumption that our way of content boosting helps convey the semantics contained in the latent dimensions. This appears essential before making further use of this advantage.

In the following, we first describe the *setup* for these analyses, proceed with the *results*, and finally conclude the chapter with a *discussion* of the insights gained in these experiments [Loe*19b].

5.4.1 Setup

To run the *performance analysis*, we set up both a standard matrix factorization recommender as a baseline, and a recommender based on content-boosted matrix factorization, using the *eval* package of our *TagMF* framework as described in Section 5.3.2. For objectively comparing our method with the baseline, as well as testing different parameterizations, we executed an evaluation protocol as shown in Listing 5.3, yielding results in terms of RMSE and NDCG@10. Details and parameters are reported below for each individual part of the analysis. To perform the *qualitative analysis* after a number of preliminary tests (see [DLZ16b] for a similar earlier analysis), we used a representative example from the range of latent factor models that we generated in the course of our experiments.

As background data for both analyses, we used established datasets to ensure generalizability. Concretely, we used the *MovieLens 20M* dataset[23] for user-item feedback from a popular domain (i.e. movies), and the *MovieLens Tag Genome* dataset[24] for item-related side information in

[23] The *MovieLens 20M* dataset contains about 20 million ratings from more than 138 000 users for over 27 000 movies. It can be found here: `https://grouplens.org/datasets/movielens/20m/`

[24] The *MovieLens Tag Genome* dataset contains item-related tag relevance scores for over 10 000 movies and 1 100 user-generated tags. It can be found here: `https://grouplens.org/datasets/movielens/tag-genome/`

the form of tags generated by a large user community. However, these datasets cover slightly different sets of movies. Therefore, we created an intersection, leaving us with a combined dataset of 10 370 movies that were included in both original datasets, 19 800 443 corresponding user ratings, and 11 697 360 relevance scores for associated tags. For standard matrix factorization, of course, only the user-item feedback dataset was required and used.

5.4.2 Results

First, we lay our focus on *analyzing the performance* of our content-boosted matrix factorization method (in the following just referred to as *TagMF*) in comparison to a standard matrix factorization algorithm in terms of objective recommendation accuracy.

Accuracy-related aspects We start by examining the influence of different basic configurations. For this, we trained a standard matrix factorization model and several content-boosted models with 20 factors, which turned out to be a meaningful number in earlier experiments. We used 10 % subsamples of users from the underlying user-item feedback dataset and 5-fold cross validation. For *TagMF*, we considered the relevance scores for the 50 most popular tags as additional training data. Figure 5.1 shows the experimental results in terms of RMSE and NDCG@10 when varying the *number of iterations* and the *regularization parameter* λ for model training. According to these results, *TagMF* showed an overall superior performance. Furthermore, the results obtained with *TagMF* were rather stable. In contrast, iterating more often over the training data decreased the performance of standard matrix factorization.

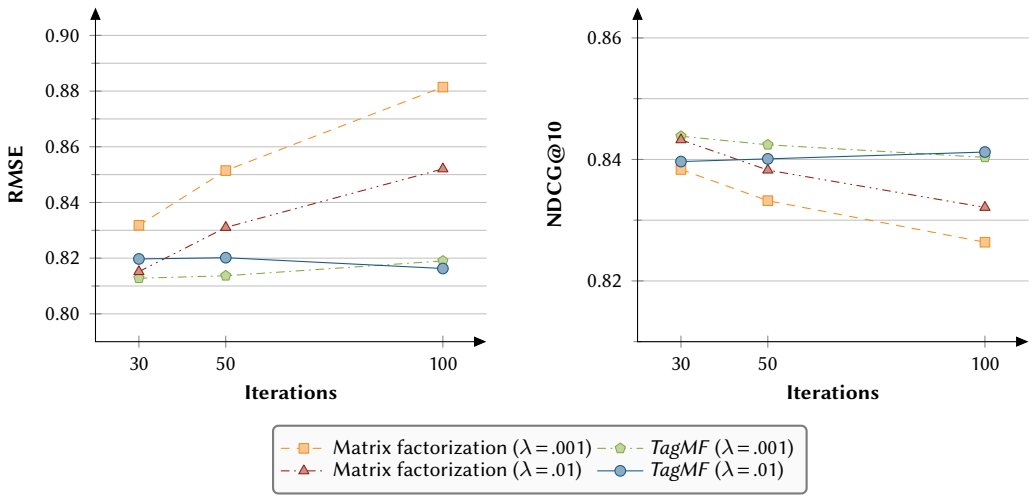

Figure 5.1 Comparison of standard matrix factorization and *TagMF* with 10 % subsamples of users in terms of RMSE and NDCG@10 for different numbers of iterations and values for λ.

Next, we look at the *number of latent factors* learned and the *number of tags* considered by *TagMF*. Following further pretests, we used 30 iterations and set $\lambda = .03$, now with 1 % subsamples of users. The RMSE and NDCG@10 results after again performing 5-fold cross validation are shown in Figure 5.2. Overall, the positive effects of considering additional information were visible again: When using 50 tags or more, RMSE was lower for *TagMF* (i.e. better), independently of the number of latent factors. NDCG@10 showed a similar behavior (higher values are better).

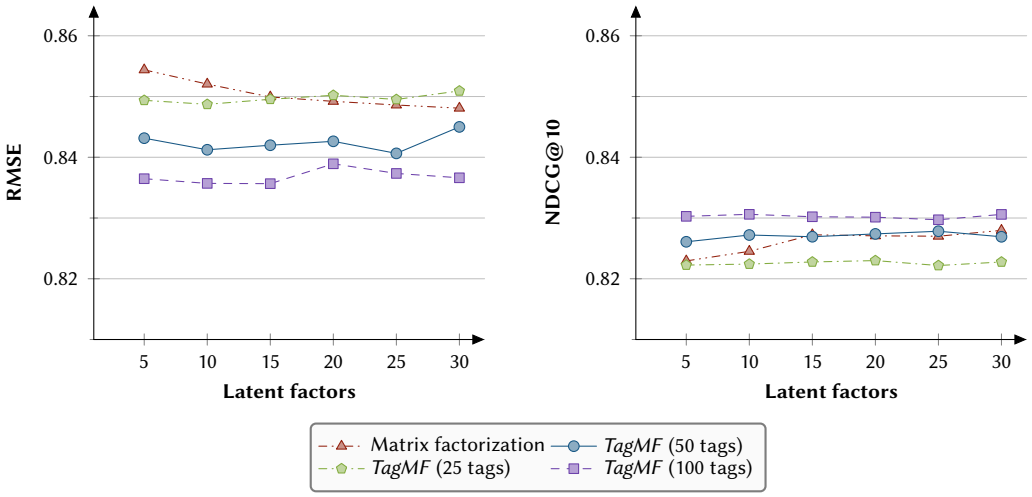

Figure 5.2 Comparison of standard matrix factorization and *TagMF* with 1 % subsamples of users in terms of RMSE and NDCG@10 for different numbers of latent factors and tags.

Finally, we address the use of a different *objective function*. For this, we used the implementation of Bayesian personalized ranking (see Section 2.2.2.2) as integrated in our framework. With 1 % subsamples of users, we again set the number of factors to 20 and used grid search for the remaining parameters. As a consequence, we ran 40 iterations with $\lambda = .003$, and used the improved sampling method by Lerche and Jannach [LJ14] in 85 % of all cases. Without content boosting, 5-fold cross validation left us with an RMSE of 1.085 and an NDCG@10 of 0.819. Once we additionally considered the 50 most popular tags as described at the end of Section 5.2.1, we obtained an RMSE of 1.441 and an NDCG@10 of 0.827.

Since content-boosted matrix factorization apparently achieved highly competitive results, and model quality actually seemed to benefit from integrating additional information, we now shift our focus to the *qualitative analysis* of a typical model generated by our method.

Qualitative aspects The application of eigendecomposition as described in Section 5.2.2 allows us to gain insights into the importance of each dimension of a latent factor space and its relation to the additionally considered content attributes. Consequently, examining the most negatively and most positively related attributes, respectively, may provide a more general understanding of what is expressed by the dimensions of an automatically learned matrix factorization model. Table 5.1 shows an example. For this, we applied our method as described above, but on the complete dataset, with 20 factors and under consideration of the 20 most popular tags: Rows represent factors ordered by descending importance values, as represented by the entries of matrix $\sqrt{\mathbf{\Lambda}}$ from our redefined model shown in (5.9) at the end of Section 5.2.2. These values are depicted in the left-most column. Tags are shown in alphabetical order in columns. The negative or positive values from matrix $\mathbf{E}$ displayed in the cells depict direction and strength of the relations determined in between, expressing how strongly certain characteristics described by the tags are represented within the factors, thus denoting their individual meaning.

As a consequence, the semantics contained in the latent dimensions can now be interpreted more easily: For instance, both factor 4 and 5 seem strongly related to the tag “fantasy”, whereas factor 4 has a very negative, and factor 5 a very positive relation to the tag “action”, i.e. these

Table 5.1 Example of automatically determined relations between latent factors (rows) and user-generated tags (columns): The five most important factors are shown together with negatively (blue) and positively (red) related tags, as indicated by $\mathbf{E}$. The factor importance values (in brackets in the left-most column) are equal to the entries of $\sqrt{\mathbf{\Lambda}}$. Representatives for each factor are automatically determined by extracting the movies (with at least 10 000 ratings) that score highest for the respective factor in the item-factor matrix ${}^{\mathrm{i}}\mathbf{H}\mathbf{E}\sqrt{\mathbf{\Lambda}}$.

Factor	action	atmospheric	based on a book	classic	comedy	dark comedy	disturbing	dystopia	fantasy	funny	psychology	quirky	romance	sci-fi	surreal	time travel	thought-provoking	twist ending	violence	visually appealing	Representatives
1 (1.66)	0.25	0.38	-0.14	0.47	-0.20	0.16	0.14	0.04	-0.27	-0.15	-0.09	0.17	-0.26	-0.03	0.15	-0.19	0.06	-0.36	0.11	0.24	The Shining, Taxi Driver, A Clockwork Orange
2 (1.51)	-0.11	-0.12	0.02	-0.34	0.12	0.21	0.26	0.12	0.27	-0.13	-0.02	0.14	-0.30	0.22	0.36	-0.51	-0.06	0.09	0.14	-0.21	Natural Born Killers, Brazil, Beetlejuice
3 (1.30)	0.10	0.11	-0.13	-0.63	-0.10	0.11	-0.06	0.07	-0.16	0.06	-0.07	0.21	-0.03	-0.16	0.18	0.17	-0.11	-0.05	-0.07	0.59	Amélie, Sin City, Magnolia
4 (1.21)	-0.39	-0.06	0.24	0.12	-0.16	0.00	0.03	-0.12	0.50	-0.22	0.05	0.14	0.29	-0.17	0.17	0.02	-0.08	-0.04	-0.48	0.19	Wizard of Oz, Willy Wonka & the Chocolate Factory, The NeverEnding Story
5 (1.17)	0.44	0.17	0.01	0.13	-0.11	-0.29	-0.16	0.01	0.44	0.10	-0.12	-0.27	-0.42	0.15	0.02	-0.16	0.01	-0.05	-0.20	0.28	Star Wars: Episode IV – A New Hope, Hobbit: An Unexpected Journey, Thor: The Dark World

factors correspond to very different kinds of fantasy movies. The right-most column shows sample movies for each factor, selected similar to the procedure we described in Section 4.2.2 for our choice-based preference elicitation method (there based on a *pure* matrix factorization model). Concretely, these representatives are movies with at least 10 000 ratings, which have the highest values in the item-factor matrix ${}^{\mathrm{i}}\mathbf{H}\mathbf{E}\sqrt{\mathbf{\Lambda}}$ from (5.9) for the respective factor. Accordingly, movies such as "Wizard of Oz" (factor 4) and "Star Wars: Episode IV – A New Hope" (factor 5) are clearly in line with the aforementioned observations (some others are not, but this is discussed below).

On a more general level than these factor representatives, the regression-constrained formulation also enables us to shed light on how users and items are positioned inside the resulting information space, and, in particular, which role the latent factors play in this context. For items, Figure 5.3 illustrates this by means of an example based on two tags, which we used for learning a simple two-factorical content-boosted model: In the plot on the left-hand side, movies are shown with respect to (normalized) tag relevance scores, i.e. values from their *Tag Genome* vectors (cf. Section 2.3.2.1), or, put differently, based on the row vectors of our item-attribute matrix ${}^{\mathrm{i}}\mathbf{H}$. On the right-hand side, movies are instead arranged according to their (likewise normalized) latent factor values, i.e. based on the vectors from our item-factor matrix ${}^{\mathrm{i}}\mathbf{H}\mathbf{E}\sqrt{\mathbf{\Lambda}}$. Comparing these two plots shows that the similarities between items in terms of content attributes can still be found even if their original positions are translated by taking into account the latent knowledge derived from user-item interaction data. This becomes especially visible with the small set of tags we used in both cases for the sake of demonstration, equal in size to the number of factors.

5.4.3 Discussion

With the offline evaluation, we pursued the goal of validating our method's general effectiveness. Specifically, we aimed at investigating whether we can observe the same positive effects other authors found in their retrospective offline experiments when taking additional information into

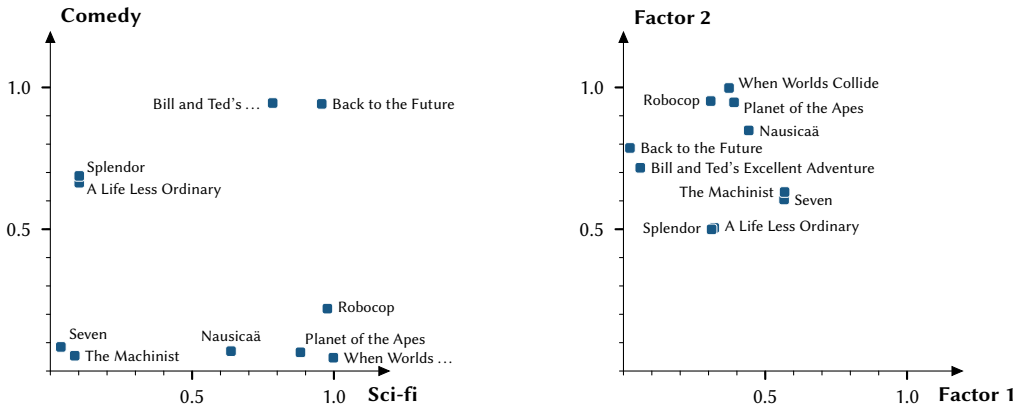

Figure 5.3 Examples of (normalized) positions of movies, once determined based on tag relevance scores (left), and once based on latent factor values (right).

account (see Section 2.2.4), and, whether the content-related associations determined by our method are not only always accessible, but also as interpretable as desired. Before investigating the potential of our method in the next chapter also from a user perspective, we first discuss the findings we obtained with regard to these aspects.

Parameterization First, given the limited subsamples of user-item interaction data we used in our performance analysis, it appears noteworthy that the decreasing accuracy of standard matrix factorization with a higher number of iterations compared to the largely stable results of *TagMF* might be attributed to overfitting: As visible in Figure 5.1, additional tag-based information seemed to contribute more to control overfitting than increasing the regularization parameter λ did for standard matrix factorization. On the other hand, it must be noted that, in general, a larger amount of user-item feedback (i.e. here larger subsamples and thus more rating data) often has more impact than leveraging side information [cf. PT09; FO19]. This could mean that tags may actually have less impact than observed in this experiment. However, this finding is only related to accuracy measured in offline experiments, whereas integrating item-related side information may have other advantages. In particular, there might be a larger impact from a user perspective, for instance, when offering interaction mechanisms that allow for a richer experience than rating-based mechanisms right from the outset.

Yet, as can be seen in Figure 5.2, the positive effects were also observed when investigating the influence of the number of latent factors: The results of standard matrix factorization improved with more factors. But, in accordance with suggestions for an optimal number of factors [cf. KBV09; ERR14], they became stable already with 15 to 20 factors. In contrast, this parameter did not seem to affect *TagMF* to a large degree. Instead, with the amount of training data we used, the number of incorporated tags seemed to be the predominant factor for model quality. Nevertheless, in cases with few tags (25 and 50), RMSE for *TagMF* went up slightly when increasing the number of factors. Apparently, tags cannot cover sufficiently the variance in the model when there are fewer tags than factors. This is in line with the aforementioned ability of tag-based information to control overfitting. Thus, more factors seem to require considerably more tags to ensure consistently high model quality (the example in Table 5.1 shows accordingly that each factor is strongly related to multiple tags). However, parameter tuning and, in particular, deter-

mining an optimal ratio of factors to tags, was not within the scope of our experiments. Besides, the fact that the observed differences were *always* in favor of our method, already confirms its general effectiveness. Beyond that, whereas these differences were rather small, independently of the number of tags taken into account (see again Figure 5.2), one can expect that they would increase considerably just by using a larger set of ratings, i.e. in real-world settings. Only then, as the data-generating function gets more complex, parameter tuning and including more model dimensions can be assumed to have a larger impact.

Alternative optimization targets Also, it should be noted that the implementation of Bayesian personalized ranking—more a by-product intended to demonstrate the possibility of transferring our approach to other algorithmic variants—led to results similar to the "standard" implementation of *TagMF*: The RMSE values must be considered meaningless due to the fact that the optimization was not performed against an error-related criterion. This becomes even more apparent when comparing the values without and with content boosting, showing that the consideration of tags completely distorted the results according to this metric. The NDCG@10 results, however, which are actually relevant, shed a positive light on this variant—noteworthy with a dataset of ratings, i.e. not the ideal form of user-item feedback for "learning to rank": The results were in the same range as before, again acknowledging the positive effect of side information.

Consequently, using *other* objective functions for content-boosted matrix factorization seems to be of interest for further research. Nevertheless, in the remainder of this thesis, we focus on the standard variant, in particular, because the differences in performance appear to be negligible, we always use the same background data, and the interactive features we propose in the next chapter are more straightforward to implement in this case. Moreover, an adoption of these features to all potential variants of the underlying algorithm does not seem to be of relevance to illustrate their general potential in terms of user control and experience. The same is true for more advanced algorithmic approaches, such as the non-linear matrix factorization method by Weston, Weiss, and Yee [WWY13]: This method can represent a variety of user interests by learning *multiple* latent factor vectors per user, and thus, consider the user's needs and goals more strongly. However, the user is not actually put into the loop, which, as we have shown in the literature review in Chapter 2, often is the case in recommender research. In contrast, we deem it more promising to use our (more basic) enhancements to account for varying interests at runtime, and provide options to let the user *actively* influence his or her (single) latent factor vector. Nonetheless, extending our content boosting approach to more advanced (matrix factorization) algorithms needs to be considered in future work.

Model interpretation Against this background, it appeared even more important to gain qualitative insights into the content-boosted models learned by our method instead of further exploring their algorithmic performance: The example depicted in Table 5.1 shows that it is possible to make sense of the relations between the dimensions of these models and the attributes considered as side information. Given some domain knowledge, this becomes clear, in particular, when examining the characteristics of the items we automatically selected as factor representatives. Moreover, Figure 5.3 highlights the role of latent knowledge: conveying semantics that exist in the patterns in the item feedback provided by the user community, representing *more subtle* item characteristics than can be expressed through explicit tag-based information. Of course, these examples leave room for interpretation—not only because both the way we derive the relations

and the results of the eigendecomposition are ambiguous: For instance, at first sight, the representatives listed for factor 3 in Table 5.1 appear to have nothing in common (e.g. "Amélie" vs. "Sin City"). Here, one must note that there had been better representatives if movies with fewer ratings were taken into account. Also, the small or negative relations to tags such as "romance" or "classic" seem counterintuitive. However, "romance" is not much related to any factor, whereas factor 3 is strongly related to the tag "visually appealing", which in turn applies to both aforementioned movies. Nevertheless, even under consideration of the bigger picture, some relations may remain unclear due to the underlying linear approach, which is still statistical in nature, and may not even be able to capture all relations. In line with that, our related work has shown that latent factors may contain semantics that can be hard to distinguish, possibly confounded by domain, dataset, and even parameterization [cf. KLZ18a; KLZ18b; Kun*19b].

Summary Nevertheless, we consider it safe to conclude that our method may serve as an appropriate starting point for implementing more advanced interactive features as extensions to conventional collaborative filtering systems. First, by using common accuracy metrics, we were able to confirm the positive effects on objective recommendation quality—independent of the algorithmic variant and the parameterization, but consistent with earlier work [e.g. Kar*10; ML13; SLH13; NZ13; FC14; Alm*15]. Second, the qualitative analysis made clear that the underlying semantics may effectively be brought to light. Since the required item-related information is often available in addition to standard user-item interaction data, for example, in the form of inherently meaningful concepts such as user-generated tags, we thus see no reason not to leverage this information also for other purposes than just improving recommendation accuracy. In particular, there seems to be potential for increasing the level of user control and providing more expressive interaction mechanisms on a level above standard user-item feedback, as well as for opening up the black boxes latent factor models usually constitute.

CHAPTER 6

> "An interface is humane if
> it is responsive to human needs
> and considerate of human frailties."
>
> — *Jef Raskin, American computer scientist*

Interactive recommending with content-boosted matrix factorization

Our interest in boosting model-based collaborative filtering with content information as described in the previous chapter has been driven by the idea of using side information not only for achieving higher recommendation accuracy. As indicated in Section 3.1.2, taking into account all available data is considered to allow for richer mechanisms to influence the recommendations in a more expressive manner than by providing standard feedback with respect to single items. Accordingly, we present in this chapter a set of *interactive features* that can be implemented as extensions to model-based collaborative filtering systems due to the availability of our content-boosted matrix factorization method [DLZ16a; Loe*19b]. By building on this method in order to further improve user control and experience, we directly address our second research question: As suggested in Section 3.2.2, we exploit that item-related information is leveraged in addition to standard collaborative filtering feedback data, and integrated with the underlying factor vectors so that users can be enabled to indirectly update their position in the latent space based on the considered content attributes. First, however, we explain the *background* in more detail, before we describe the corresponding *application possibilities* of our method. The larger part of this chapter deals with the *empirical evaluation* in the form of two user experiments we conducted to study the proposed extensions in comparison to typical baseline systems, but also to explore the general advantages of content boosting from a user perspective [DLZ16a; Loe*19b].

6.1 Background

Similarly to the motivation for coming up with the choice-based preference elicitation approach in Chapter 4, the fundamental idea behind the *application* of content-boosted matrix factorization is to combine the benefits of state-of-the-art recommendation algorithms with those of interactive recommending approaches. Yet, our earlier goal was to directly exploit the *semantics* contained in the latent dimensions of *standard* matrix factorization models. Accordingly, we presented users with system-selected examples for these dimensions, i.e. items themselves, without requiring any side information. This turned out successful for finding out about their preferences at cold start. However, we assume that users would need richer interaction mechanisms to control the results in the *ongoing* recommendation process in an equally adequate manner.

Inspiration from other interactive recommending approaches Only being able to provide item feedback leads to a variety of problems for users of collaborative filtering systems: In addition to all the specific issues of rating-based mechanisms (see Section 2.3.1), they may find it generally difficult to articulate their (possibly evolving) information need by means of such limited interaction possibilities. Moreover, the resulting feedback data are only beneficial in scenarios in which users want to see their *general* interests reflected, as these data usually represent only long-term preferences. In light of the fact that user profiles often comprise data collected over many years, it thus appears particularly important to offer mechanisms to intervene in the collaborative filtering process for pursuing *short-term* goals. As we have argued in Section 3.1.2, using other types of input data, which currently only play a role as side information in collaborative filtering, may have potential in this regard: Many of the more interactive recommending approaches proposed in the literature let users manipulate the results according to situational needs. By *leveraging specific item-related information*, the weighting or critiquing mechanisms that are often employed in these cases are much more *expressive*. On the other hand, these approaches are limited in taking into account historical user-item feedback data due to their dependency on other techniques than collaborative filtering for determining the relevance of items.

Realization of content boosting with tags Consequently, our main objective is to add more advanced interaction mechanisms to collaborative filtering systems, allowing users not only to provide explicit rating-based feedback or to choose from juxtaposed sample items. For this, given the promising findings reported in the previous chapter, we already have the right vehicle, namely our extended matrix factorization method: By exploiting the content-related associations that this method establishes with the underlying latent factors, users can be provided with options to adjust their own user-factor vector. As a result, we expect to reach a level of interactivity similar to the interactive recommending approaches mentioned above. Originating in collaborative filtering, it should however still be possible to provide users the rating-based mechanisms they are familiar with, and thus personalized results at all times.

Yet, for making the novel interactive features as expressive as possible, it is important to choose the *right type* of side information for content boosting. *Tags* generated by the user community have a number of advantages, especially in comparison to the (possibly abstract) knowledge of experts or to predefined metadata: First, user-generated data are often more readily available. Second, tags represent concepts in the language of the users: Inherently comprehensible, they describe items on a "local" rather than a "global" level, i.e. individually for each user, while other attributes are the same for all [TMS08]. Third, tags already have shown much potential for improving transparency, and, particularly important for us, controllability (cf. Section 2.3.2.1). However, only few approaches exist that feed tags directly into matrix factorization algorithms, of which the performance has—as often in recommender research—only been evaluated in offline experiments (cf. Section 2.2.4.1).

For these reasons, we use tags as a *running example* for exploring the effects of content boosting, not only in relation to the implementation of novel interaction mechanisms, but, for the first time, also on general user experience. Note that this does not require to have a priori knowledge about the relevance of the tags for the *current* user: Thanks to the enhancements we made to the underlying approach by Forbes and Zhu [FZ11] (see Section 5.2.1), each user may benefit from the improvements without ever having assigned tags him or herself. Note further that our method paves the way for a whole range of mechanisms, instead of supporting only *one specific*

type of interaction, as is the case with most purely tag-based approaches: Users may choose the mechanisms at their convenience, including interactive features that are usually available only outside collaborative filtering environments. While this becomes evident in Chapter 8, where we present our integrated recommendation platform, we describe the application possibilities of our method that we present as examples separately below to lay the focus on each individual feature that can now be implemented. Finally, keep in mind that whenever we refer to tags, *any* other type of attribute may equally be used due to the small data requirements posed by our method.

6.2 Application possibilities

In the following, we describe possible applications of our content-boosted matrix factorization method: The first three examples show how applying this method in combination with user-generated tags may improve user interaction at the different stages of the recommendation process, from *indicating preferences at cold start*, over *adjusting recommendations* according to situational needs, to *critiquing specific items*. In addition, the fourth example shows how our method may likewise contribute to *explaining user profiles*.

6.2.1 Indicating preferences at cold start

The first example refers to the need to better support users of collaborative filtering systems when it comes to *eliciting initial preferences*. In Section 3.1.1, we discussed this issue, which is also illustrated by our model of user interaction with these systems (see Figure 3.1): Usually, the current user has to rate a certain number of items before his or her interests can reliably be predicted. With our content-boosted matrix factorization method, however, only item feedback of *other* users is necessary as input data. A new user u can instead just be asked to *select a small number of tags*, similar to content- or knowledge-based approaches. As a result, a representation of this user can be created in the underlying latent factor model as otherwise by exploiting item ratings. Still, it is possible to provide ratings at any time. Regarding the tags, it is on the other hand not required that the user selects them him or herself: User interaction can entirely be avoided by choosing them automatically, for instance, based on a social media profile. Either way, user u can immediately be provided with recommendations of items that represent model dimensions with highly positive relations to these tags. Picking up the example of movies again, "comedy" and "sci-fi" would thus lead to a film such as "Ghostbusters" being recommended.

Creating a user-tag vector Regardless of how and how many tags are selected from the set H, which holds all the tags as described in Section 5.2.1, a new user-tag vector $\vec{h}'_u \in \mathbb{R}^{|H|}$ needs to be initialized as a replacement for the vector $\vec{h}_u \in$ "$\mathbf{H}$ that is otherwise derived as described in Section 5.2.2. This can be done as follows:

$$\vec{h}'_{u_a} := \begin{cases} 1 & \text{if tag } a \text{ has been selected by/for user } u, \\ 0 & \text{else.} \end{cases} \tag{6.1}$$

Next, by multiplying this vector with $\mathbf{E}\sqrt{\mathbf{\Lambda}}$, holding the tag-factor relations in our redefined matrix factorization model shown in (5.9), we can obtain a new user-factor vector:

$$\vec{p}'_u := \vec{h}'_u \mathbf{E}\sqrt{\mathbf{\Lambda}}\,. \tag{6.2}$$

Generating recommendations Now, to generate recommendations, this substitute vector can be used in the same way as a vector $\vec{p}_u \in \mathbf{P}$ that is derived by regular matrix factorization, i.e. exclusively based on user-item interaction data. This means, we calculate its inner product with the item-factor vectors $\vec{q}_i \in \mathbf{Q}$ as in the original recommendation function $s(i|u)$ shown in (2.3) in the introduction to matrix factorization in Section 2.2.1:

$$s(i|u) := \vec{p}'_u \cdot \vec{q}_i \,. \tag{6.3}$$

6.2.2 Adjusting recommendations

The second application possibility that we present here addresses the issue of being able to *control the system* also at any point later in the recommendation process. As discussed in Section 3.1.2, users who return to a collaborative filtering recommender encounter problems as soon as their situational needs are not in line with their long-term profile, or their preferences have generally changed over time: Then, as indicated in our model of user interaction, their only means to influence the recommendations is to (re-)rate single items (cf. Figure 3.1). With our content-boosted matrix factorization method, it is still possible for a user u to fall back on a rating-based profile whenever this sufficiently fits his or her needs. But, given the content-related associations established by this method with the factors, he or she can be enabled to temporarily update this profile to accommodate the current situation or to obtain alternative suggestions if there is a lack of diversity or novelty. Concretely, once the counterpart of the given item-related information has been calculated as described in Section 5.2.2, i.e. a user-tag vector $\vec{h}_u \in {}^{u}\mathbf{H}$ exists, an option to *interactively weight specific tags* can be provided to allow the user manipulate the result set that is produced by the recommender at this point in time. For example, if the user is generally interested in comedy, but selects and weights the tag "sci-fi", his or her recommendations would shift towards movies such as "Ghostbusters". When the user also wants a little more *black* humor than in the movies usually recommended to him or her, he or she may additionally select the tag "black humor", and set the weights to 1.0 and 0.5, respectively. In turn, this would lead to movies such as "Brazil" being recommended.

Creating a weighting vector First, a weighting vector $\vec{w}_u \in [0,1]^{|H|}$ needs to be defined to capture the user u's feedback in the form of weights for the tags from the set H, where 0.0 means no and 1.0 maximal interest in a tag. Then, provided the user has an option to manipulate the values of $\vec{w}_u$ via the user interface, he or she can continuously adjust the set of recommendations in realtime, allowing to observe the effects of different preference settings while exploring the whole range of available items.

Generating recommendations Regardless of how the weights stored in the components of $\vec{w}_u$ are specified by the user, this vector can be added to the derived user-tag vector $\vec{h}_u$ to calculate recommendations based upon this temporary update to the user profile. Consequently, we extend the original recommendation function $s(i|u)$ given in (2.3) based on our redefined model from (5.9) as follows:

$$\begin{aligned} s(i|u) &:= (\vec{h}_u + d \cdot \vec{w}_u)\mathbf{E}\mathbf{\Lambda}\mathbf{E}^{\mathsf{T}}\vec{h}_i \\ &\text{with } d := \begin{cases} \|\vec{h}_u\|/n \cdot \|\vec{w}_u\| & \text{if user } u \text{ applied weights for } n > 0 \text{ tags,} \\ 0 & \text{else.} \end{cases} \end{aligned} \tag{6.4}$$

Here, d represents the degree to which the weights are taken into consideration that user u assigned to the n tags he or she has selected $n \ll |H|$. As a consequence, both vectors $\vec{h}_u$ and $\vec{w}_u$ are of equal length when the user sets all weights to the maximum value, i.e. the weighting vector gets the same influence as the user-tag vector itself. In this way, *actual* ratings are only predicted at the beginning, when none of the tags available in the system are selected. Then, the recommendation function in (6.4) effectively approximates r_{ui}, since the weighting vector contains only zeros and d is set accordingly, so that the product on side of the user, $\vec{h}_u \mathbf{E}\sqrt{\mathbf{\Lambda}}$, equals a standard user-factor vector $\vec{p}_u \in \mathbf{P}$. Consequently, the recommendations are solely based on the user's profile, even though this is now the result of latent knowledge *and* tag-based information, as explained in Section 5.2.2. Otherwise, as soon as tags are selected, this representation of the user's long-term preferences is combined with the operationalization of his or her current interests and situational needs $\vec{w}_u$, which he or she has expressed with respect to these tags by interacting with the system.

6.2.3 Critiquing specific items

The third example is also related to the application of our method for supporting users who want to have more *control over the systems*. Above, we have already seen that our method lets users actively take part in the collaborative filtering process based on the additionally considered content attributes. As we have argued in Section 3.1.2, critiquing has a similar potential to strengthen the respective connection in our model of user interaction (see Figure 3.1). Yet, it takes a different angle: Users can provide feedback in a more discrete fashion, in particular, with respect to a *specific item*, which may consequently serve as a cognitive anchor. This may be helpful when they have already adjusted the recommendations, for example, using the aforementioned weighting mechanism, but still need to settle on one of the suggested items. For this, in *MovieTuner*, the current user u can request items that are similar to an item j, but represent some selected tags more or less strongly [VSR12]. For example, when "Apocalypse Now" is shown as a recommendation (or selected by the user because he or she wants to watch something similar), applying the tag-based critique "less dark" could lead to "Saving Private Ryan" being suggested (as in the screenshot in Figure 2.6). With our method, *critiquing items based on tags* becomes equally possible in collaborative filtering systems. But, building on the latent knowledge derived from historical user-item interaction data, it is ensured that the user u's long-term preferences inferred from his or her own item feedback are additionally taken into account, as it is customary in these systems. At the same time, more subtle item characteristics may come into play due to the way the item-factor vectors are now composed. As a consequence, if the user generally enjoys comedy more than other genres, he or she would instead be presented, for example, with the movie "M*A*S*H" as a new recommendation.

Determining critique dimensions Before we describe how the critiques can be applied, it is worth having a look at the selection of critique dimensions to support the user in starting or continuing the critiquing process. The method used by *MovieTuner* shows tags in the user interface based on their utility for critiquing the respective item j, their popularity and diversity [VSR12]. Accordingly, the only requirement is the availability of item-related tag relevance scores, which corresponds to our *given* matrix ${}^{\text{i}}\mathbf{H}$ (cf. Section 5.2.1). Thus, the critiquing process is completely geared towards the critiqued item, but ignores the relevance of the critique dimensions for the current user. With our content-boosted matrix factorization method, however, we can exploit

that relevance scores for all tags provided by the community also exist for each user represented within the underlying model, notably even for users who only provided user-item feedback. Therefore, we can immediately obtain a personalized set of tags for a user u by considering those with the highest scores in the *derived* vector $\vec{h}_u \in {}^{u}\mathbf{H}$ (cf. Section 5.2.2). As a consequence, critique dimensions can be presented by blending the two resulting sets of tags together: the set with tags that are particularly meaningful for critiquing item j, determined according to the *MovieTuner* method, and the set that is targeted to support the user through tags that are personally relevant for him or her.

Creating an adjusted user-tag vector Eventually, on condition that tags are selected as critique dimensions either by the system or via the user interface, and that the current user u has applied one or more critiques based on these tags to the recommended or shown item j, it is necessary to update the recommendation set. For this purpose, we need a new user-tag vector $\vec{h}'_u \in \mathbb{R}^{|H|}$ that reflects the interests of user u regarding the tags, but also the characteristics of item j. This can be achieved by performing the following steps:

1) We initialize $\vec{h}'_u$ by setting this vector to the vector of item j, but scale it to the length of the original user-tag vector as follows:

$$\vec{h}'_u := \vec{h}_j \cdot (\|\vec{h}_u\| / \|\vec{h}_j\|) \,. \tag{6.5}$$

This ensures that, in the end, we can use $\vec{h}'_u$ on the user side for generating recommendations in the same way as the original vector.

2) Assuming that user u likes very specific characteristics of item j, we keep only values of $\vec{h}'_u$ from (6.5) that are two standard deviations from the mean of this vector:

$$\vec{h}'_{u_a} := \begin{cases} \vec{h}'_{u_a} & \text{if } \vec{h}'_{u_a} > \text{mean}(\vec{h}'_u) + 2 \cdot \text{sd}(\vec{h}'_u) \,, \\ 0 & \text{else.} \end{cases} \tag{6.6}$$

Preliminary tests have shown that too homogeneous entries in the final vector can be avoided in this way. This would be different if the vector from (6.5) and the original $\vec{h}_u$ vector were directly combined in the next step, leading to results neither related to item j's characteristics nor user u's profile.

3) We determine a weighted combination of the $\vec{h}'_u$ vector from (6.6), which at this point essentially still represents the vector of item j, with the *actual* user-tag vector $\vec{h}_u$, considering the latter with a predefined weight $d \in \mathbb{R}$:

$$\vec{h}'_u := (1 - d) \cdot \vec{h}'_u + d \cdot \vec{h}_u \,. \tag{6.7}$$

Further pretests have shown that a small weight such as $d = 0.4$ ensures that item j's similarity to the items in the final recommendation set is adequately reflected. As the critiquing process continues, d may be adjusted dynamically, for example, by decreasing its value under the assumption that user-related information becomes less and less relevant with each item for which the user applies critiques. In general, however, d constitutes an application-specific parameter that needs to be determined empirically.

Generating recommendations After these steps, the new $\vec{h}'_u$ vector includes information from the vector of item j, and resembles at the same time a typical user-tag vector. Consequently, this vector can be used in the same way as the user-tag vectors in the sections before, where we adapted the recommendation function $s(i|u)$ from (2.3) for cold-start situations or the consideration of weights, respectively. However, the user's critiques need to be fulfilled in addition. For this, we employ the *linear-sat* variant of the critique distance as proposed by Vig, Sen, and Riedl [VSR12], and adjust the score calculated by $s(i|u)$ based on the degree to which the respective item i satisfies the critiques user u has expressed to item j. Building on our redefined matrix factorization model from (5.9), this may look as follows:

$$\begin{aligned} s(i|u) &:= (\vec{h}'_u \mathbf{E}\mathbf{\Lambda}\mathbf{E}^\intercal \vec{h}_i) \cdot dist_{uij} \text{ with } dist_{uij} := \text{linear-sat}(u,i,j)\,, \\ \text{linear-sat}(u,i,j) &:= \sum_{a \in H} \max(0, (h_{ia} - h_{ja}) \cdot \text{crit}(u,j,a))\,, \end{aligned} \tag{6.8}$$

and $\text{crit}(u,j,a) \in \{-1,0,1\}$ representing the user's feedback in the form of a critique based on tag a. Using the differences along the considered critique dimensions thereby assumes that critique satisfaction increases linearly to critique distance. In the end, the obtained recommendations are thus similar to the critiqued item, reflect the user's current opinion regarding the characteristics of this item, and are in line with his or her long-term profile.

6.2.4 Explaining user profiles

The fourth and final example describes a by-product of the application of our method: improving transparency instead of controllability. Despite the semantics contained in the latent dimensions of typical matrix factorization models (cf. Section 2.2.1), the attempts to increase the fundamental explainability of the resulting recommendations (cf. Section 2.2.5.2), and the approaches in which these models are exploited for visualization purposes (cf. Section 2.2.5.3), the representation of users within these models is still largely opaque. As often in model-based collaborative filtering systems, this makes it difficult to understand the reasons why items are recommended. In the qualitative analysis of a *content-boosted* model in Section 5.4, we have however seen that our method may help to gain a more thorough understanding of the underlying semantics: Items are positioned so that they represent well the different dimensions and the relations of these dimensions to the additionally considered content attributes (cf. Table 5.1 and Figure 5.3). Since our method relates the latent factors with the information space spanned by these attributes via eigenvectors and eigenvalues, the positions of users can be translated into this information space in the same way. Consequently, associating users with content attributes as described in Section 5.2.2 allows to automatically describe the representation of each user within the underlying model in a meaningful way.

For this, the fact can be exploited that a user-tag vector $\vec{h}_u \in {}^{\text{u}}\mathbf{H}$ is per se available for each user that exists in the model, independent of the tags this user actually has assigned him or herself. Accordingly, we can select the n tags scoring highest in the vector of the current user, similarly to the personalized selection of critique dimensions described in the previous section. As suggested in related work, yet outside the context of matrix factorization (see Section 2.3.2.5), these tags can immediately be used to *explain the user's profile* in textual form, even though this representation of his or her long-term preferences stems exclusively from user-item interaction data.

6.3 Empirical evaluation

To gain insights into the effectiveness of our content-boosted matrix factorization method, and, in particular, to validate the previously proposed application possibilities, we carried out another empirical evaluation. With a focus on user control and experience, we designed two exploratory user studies in which we compared the performance of interactive recommending approaches based on our *TagMF* framework with typical baseline systems, and studied the usage of the novel interactive features. First, we present an experiment with $n = 46$ participants, which was originally published in [DLZ16a; Loe*19b]. We conducted this experiment to fill the gap in prior research with respect to the impact of side information from a user perspective, and to test a first subset of interactive features that can be implemented on top of our extended matrix factorization method. Afterwards, we present a second experiment with $n = 54$ participants, which was first reported in [Loe*19b]. We performed this experiment to complement the first study by examining the influence of the latent knowledge, which is derived from the underlying collaborative filtering data, during the recommendation process. Moreover, we wanted to further explore the value of content boosting for implementing more advanced interaction mechanisms.

In the following, both experiments are presented in individual sections. We start each section by describing the *goals* of the experiment, including the specific *hypotheses*. Then, we explain the respective *method* and provide details on the prototype system and the datasets we used, the questionnaire, and the exact procedure. Subsequently, we present the *results* and conclude each part with a *discussion* in light of our second research question.

6.3.1 Part I

In the first part of the empirical evaluation, we laid our focus on the benefits of additional item-related information, and, in particular, on the more expressive interaction mechanisms that can thus be implemented as extensions to collaborative filtering systems. For this purpose, we implemented a prototypical recommender system for movies and conducted an exploratory study with $n = 46$ participants [DLZ16a; Loe*19b]. As motivated at the beginning of this chapter, we used user-generated tags as a running example. Consequently, participants were asked to interact either with a variant of the system based on standard matrix factorization, or with a content-boosted variant based on our *TagMF* framework, and to fill in a questionnaire.

6.3.1.1 Goals and hypotheses

Based on the assumption that leveraging item-related information in addition to collaborative filtering data also has advantages from a user perspective, one of the overarching goals was to complement prior offline experiments. We assumed that compared to a *standard matrix factorization recommender* as a baseline, not only objective accuracy, but also perceived recommendation quality and aspects related to user experience would benefit from using a content-boosted model. At the same time, we expected that the tags would introduce a meaning into the result sets that makes it easier to understand why items are recommended and to settle on one of the items. Ultimately still based on collaborative filtering, this should however not limit the diversity of the result sets, which often is a problem if making use of content information alone.

Since we proposed content boosting particularly as a vehicle for offering novel interactive features as extensions to conventional collaborative filtering systems, another goal was to validate the related application possibilities of our method. We expected that in comparison to the aforementioned rating-based baseline, users would benefit from the more expressive interaction mechanisms, allowing to use tags for indicating preferences at cold start and for adjusting the recommendations according to situational needs. Despite the higher complexity of these additional mechanisms described in Section 6.2.1 and 6.2.2, we expected that the intuitive interaction on a level above standard user-item feedback would not negatively affect the perceived effort.

To test these assumptions in a structured manner, we formulated the following *hypotheses*. Although the experiment had an exploratory character, this was useful for contrasting a baseline recommender relying on standard matrix factorization, and a more interactive recommender based on content-boosted matrix factorization:

H1 Content boosting leads to recommendations of higher perceived *quality*.
H2 Content boosting has no negative impact on *diversity* of recommendations.
H3 Content boosting improves *transparency*.
H4 Content boosting improves *satisfaction* with the chosen item.
H5 Content boosting reduces the *difficulty* to choose an item.
H6 Content boosting has no negative impact on perceived *usage effort*.

6.3.1.2 Method

The experiment was designed as a user study under controlled conditions. We recruited $n = 46$ participants (33 female, 13 male) with an average age of 22.89 years ($SD = 6.88$), most of them students (85 %). A supervisor was present, but participants were guided via the online tool *SosciSurvey*, which also served for presenting the questionnaire.[25] To answer the questionnaire items and to interact with the prototype system we developed for this experiment, participants used a common web browser at a desktop PC with 24" LCD (1920×1200 px resolution).

Prototype For the comparison with the baseline, we implemented the prototype system as a web-based movie recommender in two variants, corresponding to the following methods:

- A typical collaborative filtering *recommender based on standard matrix factorization*. Users were only able to rate items, without any interface elements related to tags being present. We used the same stochastic gradient descent implementation as in the offline evaluation reported in Section 5.4, i.e. the `ParallelSGDFactorizer` from the *Apache Mahout* recommender library[20] based on [Tak*09]. Pretests similar to this offline evaluation, but based on the *MovieLens 10M* dataset[14] (at the time of the study, not all data were released for the newer, much larger *MovieLens 20M* dataset), suggested to use 20 factors, 40 iterations, and $\lambda = .001$. We implemented online updating of user-factor vectors as described in [RS08].
- A *recommender based on content-boosted matrix factorization* that offered additional tag-based interaction mechanisms. We adapted the algorithm from above as described in context of our *TagMF* framework in Section 5.3. The aforementioned pretests also suggested to consider the 25 most popular tags from the underlying dataset as additional training data. We implemented the interactive features as described in Section 6.2.1 and 6.2.2. Figure A.3 in

[25] `https://www.soscisurvey.de/`

Appendix A shows the resulting interface. This interface was identical to that of the other variant with the exception of the interface elements related to tags.

Datasets To implement recommendation functionalities and interactive features, we used the well-known *MovieLens 10M* dataset[14] for basic item data as well as associated user ratings, and the *MovieLens Tag Genome* dataset[24] for associated tags. We created an intersection of these datasets. This left us with 8 429 items that were included in both datasets, 9 964 745 ratings and 9 507 912 item-related tag relevance scores. Note that in this way we used scores that were precomputed as described in [VSR10], based on the underlying dataset. In our prototype, it was not possible for users to create tags themselves, but only to use the tags contained in this dataset. However, as *TagMF* can be set up with any set of attributes, this could easily be handled differently, for example, by calculating scores based on tags assigned by users of the system at hand.

For providing users with informative and visually appealing item presentations, we used the *HetRec '11* dataset[15] and imported additional data from the *Internet Movie Database* (IMDb)[16]. The resulting dataset included metadata such as genre, cast and director information, but also plot descriptions and tags as well as movie posters. It complemented the *MovieLens 10M* dataset and was similar to the one we used for the user study described in Section 4.3.2.

Questionnaire and log data The questionnaire was primarily based on the pragmatic evaluation procedure for recommender systems proposed by Knijnenburg, Willemsen, and Kobsa [KWK11], with items related to *subjective system aspects* (SSA) and *user experience* (EXP). This framework is based on the work by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12], but reduced to stable operationalizations of the constructs, which, after repeatedly being validated, appeared to measure the qualities of recommender systems reasonably well with a limited number of questionnaire items (see Figure 3.2 for an overview of the aspects and their relations). We used this framework to assess ■ *perceived recommendation quality* and ■ *perceived recommendation diversity*. With the help of an additional item from the evaluation framework proposed by Pu, Chen, and Hu [PCH11], we also measured recommendation ■ *transparency*. To specifically analyze the ■ *usability* of the interaction mechanisms in the content-boosted variant of our prototype system, we used the *system usability scale* (SUS) by Brooke [Bro96] and the *user experience questionnaire* (UEQ) by Laugwitz, Held, and Schrepp [LHS08]. In addition, we used again items by Pu, Chen, and Hu [PCH11] to assess the ■ *interface adequacy* in this variant. For both variants, we measured ■ *choice satisfaction*, ■ *choice difficulty*, and ■ *usage effort* by means of items from the framework of Knijnenburg, Willemsen, and Kobsa [KWK11].

We also developed items to assess more *general aspects* (GEN): the ■ *suitability for different usage scenarios*, i.e. with search goal, with a vague search goal, or without a search goal, as well as the ■ *intention to use again* one of the two variants of the prototype. Besides, we collected information about *personal characteristics* (PC) of participants, including ■ *demographic data*, their ■ *domain knowledge*, i.e. interest in movies and familiarity with the movie domain, and their ■ *trust in technology*. Apart from UEQ (7-point bipolar scale ranging from −3 to 3), all items had 5-point Likert response scales. An overview of all constructs and related questionnaire items can be found in Appendix B. We also collected qualitative feedback: An open-ended question asked participants to report suggestions and complaints. Moreover, we measured task times and logged *interaction* (INT) behavior. In particular, we asked participants to rate the recommended items separately, yielding the ■ *mean item rating* for each participant.

Procedure First, participants were asked to fill in the general part of the *questionnaire* in which they needed to provide demographic information and indicate their interest in and familiarity with movies (step 1 in Figure 6.1). Next, participants were asked to complete two *preliminary tasks* using the prototype system (2). In counterbalanced order, we elicited their initial preferences, once in the form of numerical ratings, once in the form of tags:

- Participants had to *rate 10 movies* out of the 30 most popular movies in the dataset on a 5-star rating scale. This usually leads to appropriate results [cf. CGT12; ERR14]. Items were shown in random order and could be skipped when unknown.
- Participants had to *select 3 tags* they liked out of the 20 most popular tags in the dataset, also shown in random order. We chose the number of 3 tags by analyzing the general interest in tags of all users in the dataset as stored in $^{u}\mathbf{H}$, derived according to Section 5.2.2. We assumed that the tags with the highest influence would have a value at least one standard deviation above the mean of $^{u}\mathbf{H}$, which left us with 3.46 tags on average per user.

Next, the *experimental phase* started. Based on the two system variants implemented in our web application and the two preliminary tasks, we defined the underlying recommendation method and the initial preference elicitation method as *objective system aspects* (OSA). From this, we assigned participants in counterbalanced order to the following three different conditions in a within-subject design:

SMF In this condition, participants had to use the ■ *recommender based on standard matrix factorization*. Initially, recommendations were generated via online updating based on the ■ *ratings* provided in the preliminary task. The only possibility to interact with the system was to rate more items, i.e. to refine the rating-based preference profile. Participants were able to rate recommended movies and to search for movies in order to rate them.

TMFR In this condition, participants were confronted with the ■ *recommender based on content-boosted matrix factorization*. The recommendations initially shown were generated as described above, i.e. based on the ■ *ratings* provided in the preliminary task. Participants were again able to rate more items, but, in addition, to select and weight tags. The screenshot in Figure A.3 in Appendix A shows more details regarding the possible interaction.

TMFT In this condition, participants also had to use the ■ *recommender based on content-boosted matrix factorization*. Yet, initial recommendations were generated based on the ■ *tags* selected in the other preliminary task, using a user-tag vector $\vec{h}'_u$ initialized as described in Section 6.2.1. Interaction mechanisms were equivalent to the previous condition.

In each condition, participants were initially presented with the top 6 *results* of the respective algorithm,[26] generated as explained above (3a). First, they were asked to choose one movie from these results that they would like to watch. Second, they had to rate their satisfaction with each movie on a 5-point Likert response scale. Finally, they had to fill in the method-specific part of the *questionnaire* concerning their subjective assessment (3b). Afterwards, the interface of the system variant that corresponded to the respective condition was shown. There, participants had the *task* to use the provided interaction possibilities to refine the recommendations and obtain a result set that better matches their interests. During this interaction phase (3c), the top 10

[26] Although research has suggested to use sets of 7 to 10 items [cf. Bol*10], we decided to display 6 movies in our interactive setting as participants were able to adapt the recommendations at all times, and were encouraged to do so because of this limitation. Nevertheless, we increased set size in later experiments (see Section 6.3.2.2).

recommendations were displayed, showing movie title and release year, poster and plot description. In the content-boosted variant of the prototype, the most relevant tags were additionally shown (see Figure A.3). Each interaction immediately led to an update of the result set, providing direct feedback regarding the effects of the preference settings. Participants were allowed to finish the interaction phase at their own discretion. Then, the top 6 *results* from the adjusted recommendation set were presented again (3d). As before, participants had to settle on one movie, rate their satisfaction with each movie, and fill in the corresponding part of *questionnaire* (3e). For each condition, the dependent variables were thus assessed at two different points in time: before and after the interaction phase. Only with respect to the performed interaction, additional questionnaire items were presented afterwards. Eventually, once participants completed the task in all conditions, they were asked to fill in the remainder of the questionnaire, primarily concerned with usability of the content-boosted variant and other more general aspects (4).

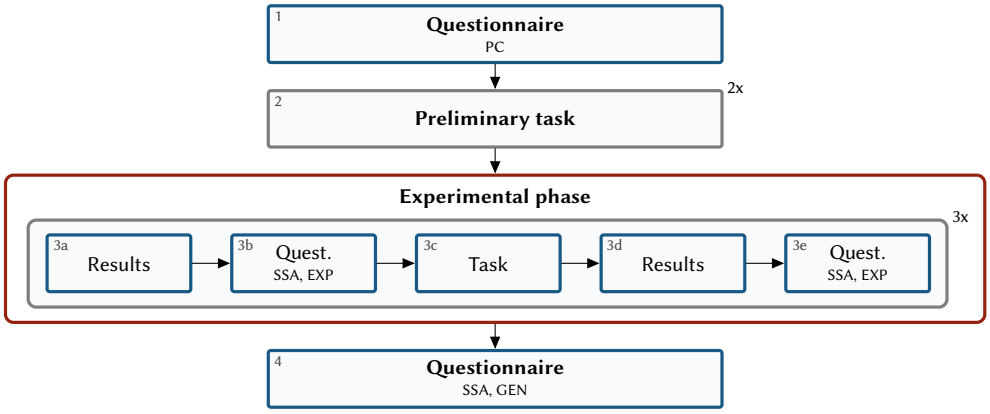

Figure 6.1 Overview of the procedure. See the text for a detailed description of the steps 1–4.

6.3.1.3 Results

In the following, we report the *quantitative results* of the experiment, including domain knowledge of participants, but, in particular, the subjective assessment of system aspects and user experience. Afterwards, we present some more general results before we proceed with the *structural equation modeling* for a detailed analysis of cold-start situations.

Quantitative results With respect to domain knowledge, participants reported that they like movies a lot ($M = 4.22$, $SD = 0.63$) and watch a fairly large number ($M = 3.72$, $SD = 0.83$). They had average knowledge about movies in general ($M = 3.07$, $SD = 0.80$) and recent movies ($M = 2.93$, $SD = 0.98$). They also stated to trust in technology ($M = 4.01$, $SD = 0.83$).

To address our hypotheses, we conducted two-way repeated-measures analyses of variance (if not indicated otherwise) to explore the effects of the objective system aspects in the three conditions (SMF, TMFR, TMFT) and the effects of the point in time (before or after the interaction phase, i.e. the respective task referred to as 3c in Figure 6.1) on the dependent variables.[17] For the comparison between conditions, mean values and standard errors are reported in Table 6.1.

Next, we detail on the differences suggested by these results, for which we performed post hoc comparisons with Bonferroni correction. Also, we elaborate on the differences we found with

Table 6.1 Mean values and standard errors for a comparison of the different conditions in terms of subjective system aspects and user experience. Higher values indicate better results on 5-point Likert response scales (*choice difficulty* and *usage effort* are reversed accordingly), except for time values. The best values are highlighted in bold.

	SMF		TMFR		TMFT	
Construct	*M*	*SE*	*M*	*SE*	*M*	*SE*
Perceived rec. quality	3.16	0.11	3.31	0.13	**3.65**	0.10
Mean item rating	3.11	0.10	3.29	0.11	**3.55**	0.10
Perceived rec. diversity	3.65	0.12	**3.78**	0.11	3.49	0.11
Transparency	3.20	0.15	3.41	0.15	**3.73**	0.13
Choice satisfaction	4.00	0.10	4.10	0.13	**4.35**	0.09
Choice difficulty	3.19	0.15	3.03	0.15	**3.30**	0.15
	33.82 sec	3.09	**28.41 sec**	2.60	28.48 sec	2.37
Usage effort	3.77	0.13	**3.84**	0.10	3.64	0.11
	2.76 min	0.28	3.75 min	0.33	3.19 min	0.32

respect to the point in time. Note that an analysis of the interaction terms of the two factors did not suggest meaningful effects, so that we omit them in the following.

■ **Perceived rec. quality and ■ mean item rating** Concerning the subjective assessment of the recommendations, there was a large effect of condition, $F(2, 90) = 7.40$, $p < .001$, $\eta_p^2 = 0.14$. Post hoc tests confirmed that the mean value for TMFT reported in Table 6.1 appeared much higher, with $p = .028$ in comparison to TMFR and $p < .001$ in comparison to SMF. In turn, TMFR performed only slightly better than SMF ($p = .885$). Still, these results support H1. We observed highly similar results with respect to the ratings participants provided in step 3a and 3d for each of the recommended items, again with large effect size, $F(2, 88) = 11.19$, $p < .001$, $\eta_p^2 = 0.20$. The mean item rating in the TMFT condition was much higher than in the two other conditions, TMFR ($p = .025$) and SMF ($p < .001$). Overall, we can thus accept H1.

With respect to the point in time, we found no considerable differences between before and after the interaction phases (3c), neither with respect to perceived quality, $F(1, 45) = 0.02$, $p = .904$, $\eta_p^2 = 0.01$, nor individual ratings, $F(1, 44) = 0.02$, $p = .885$, $\eta_p^2 = 0.01$.

■ **Perceived recommendation diversity** In terms of perceived diversity of the recommended items, we found a medium effect of condition, $F(2, 90) = 3.02$, $p = .053$, $\eta_p^2 = 0.06$. Post hoc testing suggested a difference in the mean values between TMFT and TMFR reported in Table 6.1 ($p = .070$). However, with a mean value in between these two conditions, SMF was perceived as positive as TMFT ($p = .673$) and TMFR ($p = .620$), so that we can accept H2. We also found a difference with respect to the point in time, $F(1, 45) = 5.91$, $p = .019$, $\eta_p^2 = 0.12$. Before the interaction phases (3c), recommendations were perceived slightly more diverse ($M = 3.73$, $SE = 0.10$) than afterwards ($M = 3.55$, $SE = 0.90$).

■ **Transparency** We noted a medium to large effect of condition on transparency, $F(2, 90) = 6.22$, $p = .003$, $\eta_p^2 = 0.12$. As it can be seen by inspecting the mean values in Table 6.1, recommendations in the SMF condition were perceived as less transparent than in the TMFT condition ($p = .003$).

This confirms H3, even though the post hoc test indicated no notable difference between SMF and TMFR ($p = .565$). In addition, there was a small difference between the two *TagMF* conditions ($p = .083$), but none between the two points in time, $F(1, 45) = 0.01$, $p = .948$, $\eta_p^2 = 0.01$.

In light of these findings, we wanted to analyze in more depth how content boosting, and, in particular, the initial preference elicitation method, contribute to transparency, and which effects this variable has on other relevant aspects. However, before we describe the structural equation models that we used for this purpose, we address the remaining subjective system aspects as well as the various aspects related to user experience.

■ **Usability and ■ interface adequacy** First, we elaborate on the usability of the content-boosted variant of our prototype system.[27] With a SUS score of 78, it was rated as "good" according to [BKM09]. Values between 0.95 and 1.96 on the different subscales of the UEQ can be considered equally promising, which is visible in Figure 6.2 in comparison to the benchmark values. In particular, the subscale for perspicuity, a pragmatic quality aspect, yielded an "excellent" score ($M = 1.96$) according to [SHT17]. This provided further evidence that this variant was easy to understand and appeared transparent, which additionally supports H3. Overall attractiveness ($M = 1.24$), but also efficiency ($M = 1.16$), another pragmatic quality aspect, as well as stimulation ($M = 1.07$) and novelty ($M = 0.95$), two hedonic quality aspects, were rated "above average". These results corresponded well with the very positive assessment of the interface's adequacy ($M = 4.13$, $SD = 0.48$). Only for the remaining pragmatic quality aspect, dependability, the score was "below average" ($M = 0.99$). However, this was expected because the result set changed a lot in the interaction phases (3c) whenever participants provided new ratings, which was difficult to predict and *appeared* to be out of their control. Qualitative comments supported this assumption, underlining that our method was not responsible for this result.

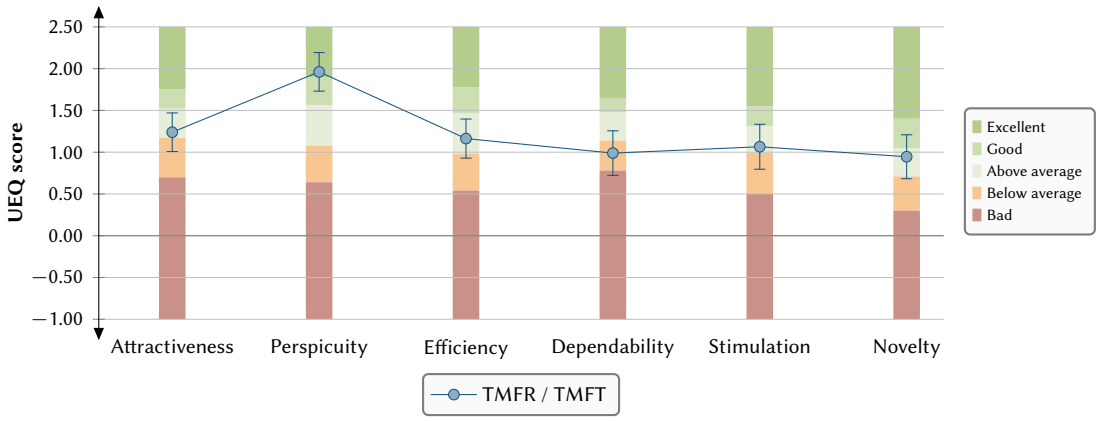

Figure 6.2 UEQ results including 5 % confidence intervals. Benchmark values are from [SHT17].

■ **Choice satisfaction** According to the mean values reported in Table 6.1, participants' satisfaction with the movie they finally selected from the recommendations was equally high in all conditions. However, the statistical test supported the existence of differences, with medium

[27] Note that we asked specific questions regarding usability only for this variant in order to reduce participants' workload in the within-subject design. Also, the interaction in the other variant of our system was limited to rating items (cf. Section 6.3.1.2), which minimized the need for a dedicated usability evaluation.

effect size, $F(2, 90) = 4.72$, $p = .011$, $\eta_p^2 = 0.10$. A post hoc test indicated an advantage of TMFT over SMF ($p = .009$), which confirms H4. Again, the difference was smaller between TMFR and TMFT ($p = .091$), but, in particular, between TMFR and SMF ($p = 1.000$). Moreover, the test suggested a difference with respect to the point in time, with medium effect size, $F(1, 45) = 5.07$, $p = .029$, $\eta_p^2 = 0.10$. In the part of the questionnaire presented before the interaction phases (3b), participants were more satisfied ($M = 4.28$, $SE = 0.10$) with the selected movie than in the part (3e) shown afterwards ($M = 4.02$, $SE = 0.11$).

■ **Choice difficulty** Regarding the difficulty to choose a movie from the recommendations before or after the interaction phases (i.e. in step 3a or 3d), we neither found an effect of condition, $F(2, 90) = 1.20$, $p = .307$, $\eta_p^2 = 0.03$, nor point in time, $F(1, 45) = 1.60$, $p = .212$, $\eta_p^2 = 0.03$. As indicated in Table 6.1, TMFT was rated best, but only slightly better than SMF and TMFR.[28] To investigate this aspect in more depth, we additionally operationalized the difficulty to choose an item as the total time participants spent for settling on a movie from the respective list (see also Table 6.1): In contrast to participants' perception, the analysis of the within-subjects main effect for this objective measurement revealed differences with medium effect size for condition, $F(2, 88) = 5.34$, $p = .006$, $\eta_p^2 = 0.11$. Participants needed more time in the SMF condition compared to the TMFR ($p = .015$) and the TMFT condition ($p = .050$). The difference between the two *TagMF* conditions appeared in contrast negligible ($p = 1.000$). Given the questionnaire results, H5 is yet only partially supported. With respect to the point in time, we also found a considerable difference, with large effect size, $F(1, 44) = 28.03$, $p < .001$, $\eta_p^2 = 0.39$. Before the interaction phases (3c), decisions took longer ($M = 34.66$ sec, $SE = 2.88$) than afterwards ($M = 25.81$ sec, $SE = 2.31$).

■ **Usage effort** For perceived effort, the results shown in Table 6.1 were rather similar across conditions.[28] Participants were asked to assess this aspect only after the interaction phases (i.e. in step 3e). Therefore, we used a one-way analysis of variance, which also indicated a negligible effect, $F(2, 90) = 1.40$, $p = .253$, $\eta_p^2 = 0.03$. Yet, we again operationalized this construct more objectively: Concerning the total time it took participants to complete the tasks, we found a medium-sized effect using a one-way analysis of variance, $F(2, 90) = 3.34$, $p = .040$, $\eta_p^2 = 0.07$. According to the mean values reported in Table 6.1, the interaction phases (3c) were longer for both TMFR and TMFT than for SMF. However, post hoc tests indicated that participants only in the TMFR condition spent considerably more time ($p = .040$). Thus, in combination with the absence of notable differences in the questionnaire results, H6 can still be confirmed.

■ **Suitability for different usage scenarios** Similar to the usability-related constructs, we assessed the general construct of the suitability for different usage scenarios only for the content-boosted variant of our system. Overall, it was rated useful *without* ($M = 3.78$, $SD = 0.99$) and with a *vague* search goal ($M = 3.89$, $SD = 1.02$). In contrast, but as expected, participants indicated lower suitability for scenarios *with* a search goal in mind ($M = 2.52$, $SD = 1.50$).

■ **Intention to use again** When we finally asked participants to compare the two variants with respect to their intention to use them again, they seemed to clearly prefer the content-boosted variant ($M = 3.76$, $SD = 1.02$) over the baseline interface ($M = 2.83$, $SD = 1.00$), which was confirmed by a paired t-test ($t(45) = 4.15$, $p < .001$; $d = 0.61$). Figure 6.3 underlines these results. In addition, the qualitative statements supported that participants "did not only want to use star ratings, but

[28] Note that higher values indicate better results.

rate several aspects, so that the system can provide better movie recommendations" and that they "really liked selecting the tags and using the sliders".

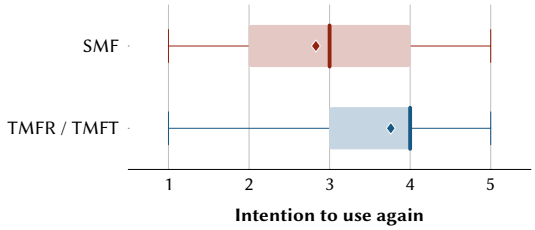

Figure 6.3 Box plot depicting the intention of participants to use again one of the methods: The thick vertical lines represent medians, the diamond signs mean values. Boxes extend from the 25th to the 75th percentile, whiskers correspond to minimum and maximum values.

Structural equation modeling As already outlined above, we were interested in exploring the role of transparency in more depth. Given that a major difference between conditions was the preference elicitation in the preliminary tasks in step 2, we expected to gain further insights by focusing on cold-start situations. For this, we used structural equation modeling, a multivariate analysis technique that allows to investigate the influence of individual aspects and their relationships. Although considered particularly useful for evaluating aspects related to user experience [Kni*12; KW15], this technique has rarely been applied in recommender research. Exceptions include analyses of the effects of objective system aspects on the perception of recommendations [Kni*12; Eks*14], of the influence of choice-based preference elicitation in comparison to rating-based mechanisms [GW15], of the number of recommendations in relation to choice difficulty and satisfaction [Bol*10], and of the impact of diversification based on latent factors models on these aspects [WGK16]. However, as there are no user experiments regarding the effects of integrating these models with additional information (cf. Section 2.2.4.2), there are also no investigations in this regard that use structural equation modeling.

To close this gap, we again relied on the framework by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12]. In line with the overview in Figure 3.2, we hypothesized that manipulating *objective system aspects* (OSA) would have an effect on *user experience* (EXP), possibly mediated by *subjective system aspects* (SSA), which represent the perception of these aspects. In turn, we expected *interaction* (INT) behavior to be strongly related to user experience, and all these aspects to be moderated by *personal characteristics* (PC) [Kni*12; KW15]. Against this background, with the knowledge that changes to algorithms and interaction mechanisms substantially affect the subjective assessment [cf. KW10; Kni*12; CP12a; Ngu*13; Eks*14; KW15; Eks*15], we again defined ■ *recommendation method* (*standard vs. content-boosted matrix factorization*) and initial ■ *preference elicitation method* (*ratings vs. tags*) as implemented in our three conditions as objective system aspects. We considered ■ *perceived recommendation quality*, and, most important for the question at hand, ■ *transparency*, as subjective system aspects. As one of the most fundamental constructs from a user perspective, we included ■ *choice satisfaction*. We complemented this more general assessment of the initial recommendations by capturing the interaction behavior in the form of the ratings provided for each movie in step 3a, i.e. we also included the ■ *mean item rating*. In addition, we took into account personal characteristics to deduce assumptions about the influence of different dispositions, ■ *domain knowledge* and ■ *trust in technology*.

Based on these definitions, we set up a *first theoretical model*, which is shown in Figure 6.4. In line with the regular way of representing structural equation models, the diagram contains boxes for constructs (colored in accordance with the framework) and edges in between that highlight their relationships. Arrows with one head represent causal regression coefficients. For these edges, standardized regression weights and p-values are shown. Arrows with double heads represent correlation coefficients, which are displayed accordingly. As it is common practice, we omit non-significant relationships for the sake of clarity.

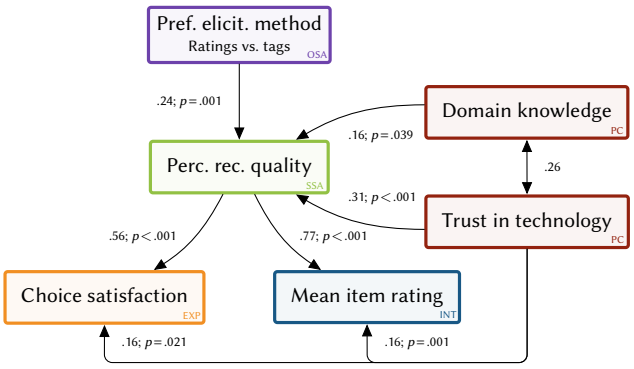

Figure 6.4 Structural equation model for comparing the influence of initial preference elicitation via ratings or tags. On the edges between the constructs, standardized regression weights and p-values are shown.

Direct effects of varying the *recommendation method* across conditions were not significant for any dependent variable or the mediator. Thus, the objective system aspect of using either standard or content-boosted matrix factorization for generating recommendations was eventually not integrated in our model, yielding a good fit with the data ($\chi^2(7) = 8.246$, $p = .311$, $CFI = .995$, $TLI = .989$, $RMSEA = .032$). The model also explained a large amount of variance regarding our dependent variables, *choice satisfaction* ($R^2 = .408$) and *mean item rating* ($R^2 = .698$), as well as about 20 % of the considered mediator, *perceived recommendation quality* ($R^2 = .208$).

In contrast to the *recommendation method*, the *preference elicitation method* seemed to account for a significant explanation of *perceived recommendation quality*. Further analysis showed that this subjective system aspect completely mediated the otherwise significant predictive power of varying the *preference elicitation method*, i.e. whether initial preferences were captured in the form of ratings or tags. Also, this more general variable was a strong predictor for the specific aspects related to user experience (*choice satisfaction*) and interaction behavior (*mean item rating*). Regarding personal characteristics, *domain knowledge* appeared to have an effect only on *perceived recommendation quality*, whereas *trust in technology* affected all dependent variables.

Circling back to *transparency*, we integrated this subjective system aspect as an additional mediator in a *second theoretical model*, which is shown in Figure 6.5. Overall, the new model again fitted the data well ($\chi^2(12) = 13.669$, $p = .322$, $CFI = .995$, $TLI = .989$, $RMSEA = .032$), and explained a large proportion of variance regarding *choice satisfaction* ($R^2 = .401$), *mean item rating* ($R^2 = .693$), and *perceived recommendation quality* ($R^2 = .523$). Moreover, it achieved a reasonable amount of explained variance with regard to *transparency* ($R^2 = .234$).

The second model indicated that the predictive power of *perceived recommendation quality* on the dependent variables was still there. However, we found shifts in the relationships between

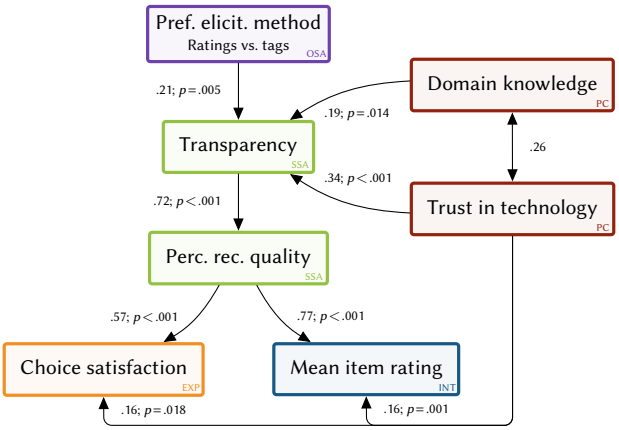

Figure 6.5 Structural equation model for the influence of the preference elicitation method, mediated by transparency. On the edges between the constructs, standardized regression weights and *p*-values are shown.

the variables: *Transparency* appeared as a substantial causal factor for *perceived recommendation quality*, which in turn acted as a complete mediator for the effects on the more specific variables. In fact, *transparency* turned out to be a regressor fully mediating the direct effect of the *preference elicitation method* on *perceived recommendation quality* that was found in the first model. Besides, *transparency* appeared to partially mediate the personal characteristics *domain knowledge* and *trust in technology*. Overall, the structural equation modeling in this way underlined the role of content boosting in explaining the differences we already found using the analysis of variance: If initial recommendations were generated based on selected tags, there was a direct effect on transparency, which eventually supports H3 also for the special case of cold start.

6.3.1.4 Discussion

The variant of the prototype system that was implemented based on our content-boosted matrix factorization method received higher scores than the baseline with a standard matrix factorization recommender in relation to subjective system aspects and user experience. Beyond that, the comparison of the two conditions based on this method illustrated the benefits of using tags for the elicitation of initial preferences. The structural equation modeling confirmed this effect of the corresponding objective system aspect, but also emphasized the important role of transparency, which had a positive influence on a range of important constructs, including perceived recommendation quality. This underlines that content boosting may even help indirectly in conveying the semantics in the dimensions of latent factor models. Overall, the experiment thus allowed us to accept all our exploratory hypotheses,[19] with only one minor exception.

Quality and transparency In some cases, however, the differences between the condition based on standard matrix factorization and the lower rated *TagMF* condition (mostly the condition with initial recommendations based on ratings) were only small. Nevertheless, the results were almost always in favor of our method, though some scores were not as high as expected. Since this applied to all conditions, we assume that the dataset (only movies released before 2008) and our particular sample (more females, rather young, average domain knowledge) were responsible for this observation. Answers to the open-ended question as well as better results in

the second user experiment (newer dataset, more heterogeneous sample) support this assumption. Nonetheless, especially the results with respect to *recommendation quality* appeared very promising, and more importantly, superior in both *TagMF* conditions. The ratings participants provided for the recommended movies were in line with these questionnaire results (H1). Regarding *transparency*, but also *choice satisfaction*, an indicator of user experience, we found the same tendencies. In particular, the baseline recommender was clearly outperformed in case our method was fed with an initial set of preferences based on tags (H3, H4).

We did not find meaningful effects between the different points in time, neither for recommendation quality nor transparency. Given the absence of interaction effects, we deduce that this applied to all conditions. On the other hand, participants were less satisfied with the chosen movie after the interaction phases. Also, there was a considerable effect for the time required to make a choice. The latter was expected as participants likely decided for an item already during the interaction phases, and were therefore able to choose faster afterwards (as the questionnaire results were similar at both points in time, this apparently had no impact on the *perceived* difficulty). The former may be attributed to user behavior as well: Already during the interaction phases, participants rated many of the movies they (at least partially) knew. Consequently, the shown recommendation set changed a lot, eventually comprising items that were not as easy to assess at first sight. One participant explicitly mentioned that the final set "would have better fitted [his or her] taste if movies [he or she] rated highly had not been removed immediately".

However, the fact that meaningful differences were already visible before the interaction phases suggested that indicating preferences via tags in the preliminary task already improved the subjective assessment of system aspects and user experience. Given these differences were still visible afterwards, the positive effects seemed to persist until participants arrived at the final set of recommendations. Most noteworthy, this was also true for *transparency*, even though participants did not know that only the few tags they selected up front were initially responsible for the recommendations. Boosting matrix factorization with content information thus seemed to help participants in judging the output of the algorithm—independent of any later interaction (H3). This is a particularly important consideration in view of real-world scenarios, in which initial preference elicitation can be seen as part of regular system use.

Structural equation modeling Because of these findings, we further examined the role of transparency. Our first structural equation model confirmed that selecting tags is as a promising alternative to rating items in cold-start situations, improving the *perceived quality of recommendations* (H1). Including transparency in the second model increased the amount of explained variance concerning recommendation quality from 21 % to 52 %. With a high standardized regression weight, transparency appeared to be a substantial predictor for this variable. In turn, varying the preference elicitation method contributed to the explanation of *transparency*. Moreover, the effects of this objective system aspect were now fully mediated by transparency. Apparently, content boosting led to more comprehensible results in the first place (H3), which improved their quality, and ultimately led to higher *satisfaction with the chosen items* (H4). Thus, we deduce that the user-generated tags we considered as side information import semantics into recommendation sets, which are more natural to understand than a meaning that needs to be derived from sets that are exclusively based on standard user-item interaction data.

Nevertheless, it must be noted that the structural equation modeling showed no considerable

effect of varying the recommendation method. However, this is in line with recent research stating that objectively better algorithms do not necessarily produce results that are better from a user perspective [XB07; KR12; PCH12; Eks*14]: Although it can achieve high accuracy scores, a list of items detached from a superordinate context might not be satisfactory for users. But, in light of the fact that varying the preference elicitation method, in contrast, caused a difference, it seems that by associating the latent factors with tags, the recommendation sets do actually exhibit some kind of inner consistency, and thus appear more transparent.

On a side note, while recommendation quality was indeed the main predictor for aspects related to user experience and interaction behavior, *personal characteristics* played a considerable role as well. For example, the structural equation models suggested that our method is particularly helpful for users with little domain knowledge, making it easier to comprehend why certain items are recommended. Noteworthy, our method was also rated as more useful for situations without or with a vague search goal. On the other hand, the influence of trust in technology was only partially mediated by transparency. Therefore, one can assume that personal characteristics may alter the way perceived recommendation quality is translated by users into the numerical ratings they usually need to provide for single items: Users who do not trust in technology likely provide lower ratings in more technically-oriented systems. This constitutes another argument for coming up with more natural ways to interact with collaborative filtering systems.

Usability and effort However, more advanced interactive features can increase interaction effort. At the same time, if these features are made possible by content boosting, the results can become too narrow, as known from conventional content-based filtering (cf. Section 2.1.2). Eventually, this might even increase the difficulty for users to make a decision [cf. Bol*10]. In contrast to these possible negative outcomes, the content-boosted variant of our system actually obtained very positive feedback in terms of general *usability*.[27] Some participants had suggestions (e.g. "full text search should be integrated") or complaints (e.g. "movies cannot be excluded from the results without rating them"). Yet, these qualitative comments addressed specific usability issues of the prototypical implementation. These issues should be considered in future work, but were not related to our method. In line with that, participants preferred the variant based on our method when they were directly asked. This preference might be a reason why they spent more time for the tasks in the corresponding conditions. On the other hand, the higher complexity of the novel features, together with the familiarity with rating-based mechanisms, may have accounted for this finding as well. Nevertheless, the *interaction effort* appeared similar, suggesting that content boosting actually had no negative impact (H6). The fact that the effort was perceived as highly acceptable was also supported by the UEQ results for the pragmatic quality aspect of efficiency, as well as the results in relation to interface adequacy.

Diversity and choice difficulty Also with respect to perceived *diversity*, the concerns were unfounded: The score in the baseline condition was between the scores in the *TagMF* conditions (H2). Still, the expected side effect was visible in the comparison of these two experimental conditions: In case initial recommendations were generated based on ratings, the resulting item sets seemed more diverse than with a tag-based preference profile. Here, recommendations were focused very much on concepts reflected by selected tags. However, this might be confounded by the fact that in the preliminary task, participants provided ratings for a broad range of items, including items they did not like. In contrast, tags were all shown at once, and participants were

asked to select only the ones they prefer. Either way, it is worth mentioning that diversity was always rated higher after interaction phases, which is in line with literature on the increasingly important role of this aspect [VC11; CHV15]. Finally, with respect to the *difficulty of choosing an item*, we also did not find a negative effect of content boosting—but no positive either: Although participants needed considerably less time to choose an item in the corresponding conditions, the questionnaire results did not allow to conclude that it became easier to settle on one of the recommended items (H5). Against our assumption, the greater homogeneity of the item sets appeared to contribute more to choice difficulty than the positive effects caused by transparency were able to reduce it. Accordingly, further research is required in this regard.

Summary All in all, our method seems valuable for alleviating the new user cold-start problem, but also for increasing the degree of control users have over the recommendations throughout the process. The positive results with respect to the preference elicitation via tags are in line with earlier research on tag-based profiles [BJG13]. Nevertheless, given the recent advances with respect to active learning [Rub*15; ERR16], we want to remark that the rating-based baseline we used in this experiment relied on a rather basic method, with much room left for improvement. However, this also applies to our tag-based approach. Despite the exploratory nature of the experiment, it therefore appears safe to say that users can successfully be provided with more expressive options to influence latent factor models, allowing them to adjust the resulting recommendations according to situational needs at all times. In this way, the experiment also helped to validate both application possibilities of our content-boosted matrix factorization method that we wanted to examine in this part of the evaluation. In addition, it showed for the first time that considering side information is beneficial with respect to the subjective assessment of recommendations. Thus, we can conclude that compared to a typical model-based collaborative filtering recommender, both *user control and experience* clearly benefit from *leveraging item-related information* by means of our method (RQ2).

6.3.2 Part II

In the second part of the empirical evaluation, we laid our focus again on the possibility to add more expressive interaction mechanisms to collaborative filtering systems, and, in particular, on the specific role of the latent knowledge that is derived from historical user-item interaction data. To investigate this role in relation to user control and experience, we set up an interactive recommender, again using movies and user-generated tags, and conducted another exploratory study with $n = 54$ participants [Loe*19b]. Participants were assigned to different variants of this prototypical system, either with interaction possibilities implemented on top of a pure content-based technique, or on top of our *TagMF* framework, and asked to fill in a questionnaire.

6.3.2.1 Goals and hypotheses

To complement the first part of the evaluation, the main goal of this experiment was to examine the advantages of boosting a latent factor model with content information in comparison to using content information alone. The remaining example of the application possibilities of our extended matrix factorization method described in Section 6.2.3 introduced the option to critique specific items via tags. Therefore, it appeared obvious to address this goal by means of a comparison with a *conventional critique-based recommender system*. Such a content-based baseline also benefits from the availability and comprehensibility of tags, but, being independent of established

collaborative filtering techniques, fails at taking into account the preference profiles that usually allow for personalization—a problem of many interactive recommending approaches. With content boosting, we expected that users would value that in addition to their applied critiques, long-term preferences inferred from previously provided item feedback can be considered. This should contribute to perceived quality of the recommendations and aspects related to user experience. At the same time, we did not expect that the latent knowledge would interfere with comprehensibility or that diversity would be constrained, which often is the case in pure content-based filtering. On the contrary, we assumed that the interaction would be perceived as more adequate and less effortful because of the personalization of the critiquing process.

To guide the analysis of these expected differences between a regular critique-based recommender and one that uses content-boosted matrix factorization, in this way continuing the validation of the application possibilities, we defined another set of exploratory *hypotheses*:

H1 Content boosting leads to recommendations of higher perceived *quality*.
H2 Content boosting increases *diversity* of recommendations.
H3 Content boosting has no negative impact on *transparency*.
H4 Content boosting positively affects perceived *interaction adequacy*.
H5 Content boosting improves *satisfaction* with the chosen item.
H6 Content boosting reduces the *difficulty* to choose an item.
H7 Content boosting reduces perceived *usage effort*.

6.3.2.2 Method

Also this experiment was designed as a controlled laboratory user study. We recruited $n = 54$ participants (37 female, 17 male) with an average age of 27.89 years ($SD = 10.30$), a small majority of them students (57 %). While a supervisor was present, participants were guided through the experiment via *SosciSurvey*.[25] We used this tool also to set up the questionnaire. To fill in this questionnaire and interact with the prototype system we implemented for this study, they had to use a common web browser running on a desktop PC with 24" LCD (1920×1200 px resolution).

Prototype Again, we implemented the prototypical movie recommender system as a web application in two variants, based on the following methods:

- A typical interactive *recommender with critiquing based on tags* that implemented the method behind *MovieTuner* as described in [VSR12], with an interface similar to its integration in the *MovieLens* platform.[10] For this, we relied on the 50 most popular tags from the underlying dataset. Critique dimensions were shown by the system based on item-related tag relevance scores according to the method described in [VSR12], i.e. depending on tag utility, popularity, and diversity. Based on prior testing, we chose the *linear-sat* metric for computing critique satisfaction. Recommendations were then generated based on similarity and critique distance to the current item, again in terms of tag relevance scores. Further parameters were set as suggested in the literature [VSR10; VSR11; VSR12].
- A typical collaborative filtering *recommender with critiquing based on content-boosted matrix factorization*. Figure A.4 in Appendix A shows the front-end, nearly identical to the other system variant (the only difference was the dialog with the user profile). As a point of departure, we used again the stochastic gradient descent implementation from the *Apache Mahout* recommender library,[20] i.e. the `ParallelSGDFactorizer` based on [Tak*09]. We adapted the

algorithm as described in context of our *TagMF* framework in Section 5.3, i.e. in the same way as for the offline evaluation and the first user experiment (see Section 5.4 and 6.3.1). As a result of the offline evaluation, we used 20 factors, 30 iterations, and $\lambda = .001$. As additional training data, we leveraged the 50 most popular tags from the underlying dataset. Critique dimensions were suggested based on item-related as well as user-related tag relevance scores, i.e. as described above, but with half of the tags replaced by a set personalized as described in Section 6.2.3. Recommendations were generated as described in this section as well.

Datasets For basic item data, user ratings, and item-related tag relevance scores, we used the same intersected dataset based on the *MovieLens 20M* dataset[23] and the *MovieLens Tag Genome* dataset[24] as in the offline experiments reported in Section 5.4. This dataset contained 10 370 movies, 19 800 443 ratings, and 11 697 360 tag relevance scores, with which we implemented the recommendation and critiquing functionalities. As in the first part of the evaluation, all tags came from the underlying tag dataset. However, since *TagMF* can be used with any set of attributes, users could also be enabled to create new tags themselves. Instead of precomputed relevance scores, tags assigned by users of the system would then be used to calculate the scores.

For item presentation, we additionally gathered data from *The Movie Database* (TMDb).[29] In case certain information was not available, we used the *Open Movie Database* (OMDb).[30] Both websites are collaborative alternatives to the *Internet Movie Database* (IMDb)[16] and provide access via programming interfaces. As a result, we got metadata for each movie in the *MovieLens 20M* dataset, including genre information, plot descriptions, lists of directors, cast members and keywords, titles in different languages as well as links to posters, images and trailers.

Questionnaire and log data As in the first part of the evaluation, the questionnaire was primarily based on the framework by Knijnenburg, Willemsen, and Kobsa [KWK11], containing items related to *subjective system aspects* (SSA) and *user experience* (EXP), as shown in the overview in Figure 3.2. With respect to the former, we used this framework to measure ■ *perceived recommendation quality* and ■ *perceived recommendation diversity*. Moreover, we used an item from the evaluation framework proposed by Pu, Chen, and Hu [PCH11] to assess ■ *transparency* of recommendations. To measure general ■ *usability*, we used the *system usability scale* (SUS) by Brooke [Bro96] and the *user experience questionnaire* (UEQ) by Laugwitz, Held, and Schrepp [LHS08]. More specifically, we also assessed ■ *interface adequacy* and ■ *interaction adequacy*, again using items suggested by Pu, Chen, and Hu [PCH11]. To examine the ■ *critiquing mechanism* in more detail, we integrated additional items into our questionnaire that were used by Vig, Sen, and Riedl [VSR11] in the evaluation of *MovieTuner*. With respect to user experience, we assessed ■ *choice satisfaction*, ■ *choice difficulty*, and ■ *usage effort*, again with the help of the framework by Knijnenburg, Willemsen, and Kobsa [KWK11]

Using items by Pu, Chen, and Hu [PCH11], we also assessed more *general aspects* (GEN): the ■ *overall satisfaction* of participants and their ■ *intention to use again* the respective system variant. In terms of *personal characteristics* (PC), we gathered ■ *demographic information* and asked participants regarding their familiarity with the movie domain, i.e. we assessed their ■ *domain knowledge*. Apart from UEQ (7-point bipolar scale ranging from −3 to 3), all items had a 5-point Likert response scale. An overview of all questionnaire items can be found in Appendix B. We

[29] `https://www.themoviedb.org/`
[30] `http://www.omdbapi.com/`

also collected qualitative feedback: An open-ended question asked participants to report suggestions and complaints. Furthermore, we measured task times and logged *interaction* (INT) behavior. Concretely, we asked participants to rate each of the recommended items, which gave us the ■ *mean item rating* for each participant.

Procedure As a first step, participants were asked to fill in the part of the *questionnaire* that was concerned with demography and knowledge with respect to the movie domain (step 1 in Figure 6.6). Afterwards, they needed to complete a *preliminary task* with our prototype system to provide an initial set of preferences (2a). Movies were presented one after the other based on popularity and rating entropy as proposed in [Ras*02]. Items were separated into blocks of 25, shuffled to eliminate sequence effects. Unknown movies could be skipped. Participants had to rate 10 movies on a 5-star rating scale. This is known to produce recommendations of reasonable quality [cf. CGT12; ERR14]. With the help of online updating, implemented according to [RS08] into the recommender based on our content-boosted matrix factorization method, this feedback was used to initialize a user-factor vector and to generate recommendations: The top 15 *results* were presented in the form of a list that could be expanded up to a maximum of 30 movies (2b). From this list, participants had to choose one movie they felt familiar with, and would like to see as a starting point for a possibly succeeding critiquing process.

Next, in correspondence with the two system variants, we defined the underlying recommendation method as an *objective system aspect* (OSA). From this, participants were assigned in counterbalanced order to one of the two following conditions in a between-subject design (yielding $n = 27$ participants per condition):

TAG In this condition, participants had to use the ■ *recommender with critiquing based on tags*. As in *MovieTuner*, they were able to interactively select tags, apply critiques, and switch the critiqued item to update the recommendations.

TMF In this condition, participants had to use the prototype variant that implemented the ■ *recommender with critiquing based on content-boosted matrix factorization*. Figure A.4 in Appendix A shows the interface, which was equivalent to the other condition.

Independent of the assigned condition, participants had to fulfill several tasks in the *experimental phase*. In each task, they had to interact with the respective system variant by applying critiques and switching the critiqued item. Each interaction immediately led to a new set of recommendations, offering participants direct feedback regarding the effects of their preference settings. Recommendations were presented based on movie title, release year, poster, plot description as well as associated tags (see again Figure A.4). All participants started with the same task:

Task I Participants were again confronted with the item they had chosen after the preliminary task, now as a starting point for the first critiquing process. Beginning from the top 9 recommendations produced accordingly by the underlying method, participants had the *task* to find a movie in line with their *personal preferences* (3a). Recommendations were always based on the critiqued item, and, in the TMF condition, the user-factor vector learned for the respective participant based on the ratings elicited up front.

After finishing task I, participants were confronted with task IIa and IIb in random order:

Task IIa The movie chosen after the preliminary task served again as a starting point for the critiquing process, and was used by the underlying method to generate the initially shown

top 9 recommendations. The *task* was to find a movie that participants would like to watch when going out on *a date with someone* (4a). Thus, they were not only required to take their own interests into account, but, in addition, the interests of the fictitious date, which were not explicitly given. Recommendations were generated as in the first task.

Task IIb This time, a representative horror movie served as a starting point for the critiquing process. The initial top 9 recommendations were generated accordingly by the underlying method. The *task* was to find a movie for the given situation that an *adult horror movie fan* wants to watch a movie together with *a 9-year-old child* (4a). Thus, participants were required to assume a high interest in horror movies and take the interests of the child into account, which were not explicitly given. Recommendations were always based on the critiqued item, and, in the TMF condition, an artificial user profile we created by training a user-factor vector with typical ratings of a horror movie enthusiast.

Participants were able to finish each task at their own discretion. Then, they were again presented with the top 9 *results* from the end of the respective critiquing process (3b/4b). First, they were asked to choose the movie they found most suitable for the given task. Second, they had to rate their satisfaction with each recommended item on a 5-point Likert response scale. Finally, they were asked to fill in the part of the *questionnaire* on the task they just finished (3c/4c). Eventually, after participants finished all tasks, they were asked to fill in the task-independent part of the *questionnaire* related to general aspects, usability, and the critiquing mechanism (5).

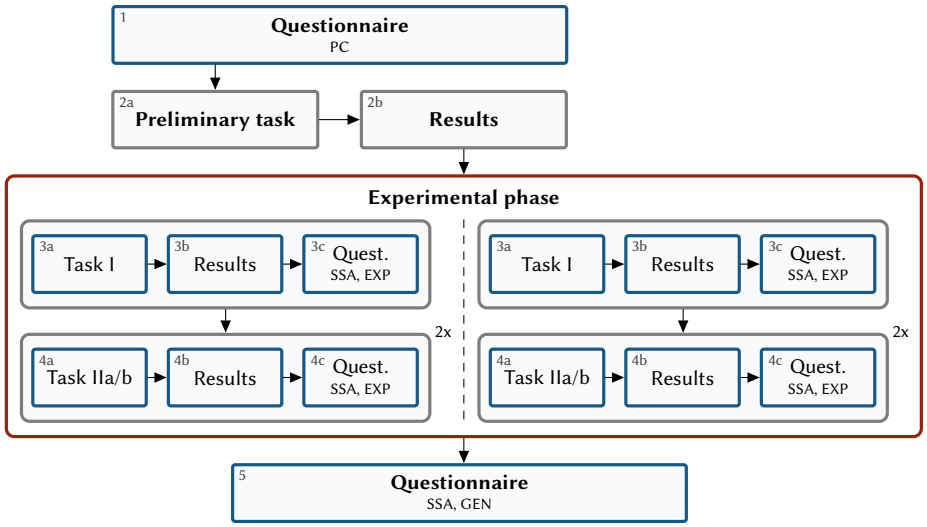

Figure 6.6 Overview of the procedure. See the text for a detailed description of the steps 1–5.

6.3.2.3 Results

In the following, we report the *quantitative results* obtained through the questionnaire. We start with domain knowledge of participants and their overall satisfaction. Then, we step through the subjective system aspects as well as the aspects related to user experience. Finally, we present the results with respect to the intention to use one of the system variants again.

Quantitative results Regarding the knowledge of participants in the domain of movies, they reported to like movies ($M = 3.44$, $SD = 1.02$) and to watch an average number compared to their

friends ($M = 3.04$, $SD = 1.03$). Moreover, they did not see themselves as movie experts ($M = 2.57$, $SD = 0.96$). The movie chosen in step 2b was rated very positively ($M = 4.65$, $SD = 0.68$), while most participants had (as intended) seen it before (94 %).

With respect to our dependent variables, it was rather meaningless to make a comparison between tasks because of their different nature. Thus, in contrast to the first part of the evaluation, we did not use repeated-measures analyses. Instead, for our specific directional hypotheses, we conducted one-tailed t-tests to compare the two conditions (TAG and TMF). If we did not hypothesize a direction, we conducted two-tailed t-tests, but this is always explicitly mentioned below. For the constructs for which we were interested in the effect of the objective system aspect for each individual task (i.e. collected in step 3d or 4d), the results are shown in Table 6.2. In these cases, we used Benjamini-Hochberg adjustment ($FDR < .05$) to account for multiple tests of the same hypothesis. In the remainder of this section, we elaborate more extensively on the observed differences.[17] In addition, we report the results for constructs for which we were interested in a task-independent assessment of the two system variants (i.e. collected only once, in step 5).

■ **Overall satisfaction** Before we address the specific constructs, the overall satisfaction of participants already sheds a positive light on the content-boosted variant of the prototype system: As underlined by Figure 6.7, participants were in general more satisfied in the TMF ($M = 4.48$, $SD = 0.75$) than in the TAG condition ($M = 4.11$, $SD = 0.80$), with medium effect size ($d = 0.48$). In line with the specific hypotheses, we applied a one-tailed t-test, which confirmed this finding ($t(52) = 1.75$, $p = .043$). The qualitative comments provided further support: For example, one participant in the experimental group explicitly stated that he or she "enjoyed using the system", which did not become apparent in the control condition.

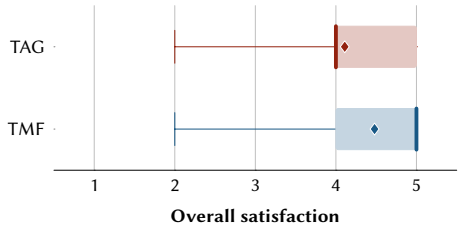

Figure 6.7 Box plot depicting the overall satisfaction of participants with the different methods: The thick vertical lines represent medians, the diamond signs mean values. Boxes extend from the 25th to the 75th percentile, whiskers correspond to minimum and maximum values.

■ **Perceived rec. quality and ■ mean item rating** Also in terms of the more specific aspects, the content-boosted variant frequently received better scores. Starting with the perceived quality of the recommendations, the mean values shown in Table 6.2 indicated an advantage of TMF. Apparently, there was a considerable effect of condition in task I and task IIb, with medium to large size, and still a small to medium effect in task IIa. The individual ratings provided in step 3c/4c for the recommended items draw a slightly different picture: The largest effect was now observed in task IIa. In task IIb, there was still a considerable difference, with medium to large effect size, but the effect in task I was much smaller. However, since all these results were still in favor of TMF, they complement well the questionnaire results, so that we can accept H1.

Table 6.2 *t*-test results ($df = 52$)[31] for a comparison of the conditions in terms of subjective system aspects and user experience. Higher values indicate better results on 5-point Likert response scales (*choice difficulty* and *usage effort* are reversed accordingly). The best values are highlighted in bold. *d* represents Cohen's effect size value.

		TAG		TMF				
Construct & Task		*M*	*SD*	*M*	*SD*	*T*	*p*	*d*
Perceived rec. quality	Task I	3.67	0.84	**4.20**	0.67	2.59	.009	0.70
	Task IIa	3.87	0.93	**4.19**	0.86	1.30	.100	0.36
	Task IIb	3.26	0.81	**4.02**	0.88	3.29	.003	0.90
Mean item rating	Task I	3.61	0.55	**3.83**	0.66	1.32	.097	0.36
	Task IIa	3.45	0.49	**3.86**	0.57	2.75	.012	0.77
	Task IIb	3.27	0.55	**3.65**	0.64	2.30	.020	0.64
Perceived rec. diversity	Task I	3.67	0.92	**4.07**	0.83	1.71	.141	0.46
	Task IIa	3.89	0.89	**4.19**	0.62	1.42	.123	0.39
	Task IIb	3.81	0.79	**4.11**	0.75	1.42	.082	0.39
Choice satisfaction	Task I	4.59	0.50	**4.78**	0.64	1.18	.121	0.33
	Task IIa	4.56	0.64	**4.81**	0.48	1.68†	.075	0.44
	Task IIb	4.00	0.83	**4.52**	0.64	2.56	.039	0.70
Choice difficulty	Task I	**3.59**	1.01	3.22	1.28	-1.18	.366	-0.32
	Task IIa	**3.37**	1.15	3.33	1.33	-0.11	.457	-0.03
	Task IIb	2.89	1.09	**3.19**	1.30	0.91	.276	0.25
Usage effort	Task I	3.98	0.60	**4.06**	0.80	0.39	.351	0.11
	Task IIa	3.89	0.87	**4.09**	0.75	0.92	.270	0.25
	Task IIb	3.46	0.63	**3.72**	0.94	1.19‡	.363	0.32

■ **Perceived recommendation diversity** In all tasks, the diversity within the sets of recommended items was perceived slightly higher in the TMF than in the TAG condition, always with medium effect size (cf. Table 6.2). Note that the difference between conditions in task IIa and IIb was highly similar, but smaller than in task I. However, this was expected as the search goal was less specific in the first task. Overall, the results provide only partial support for H2.

■ **Transparency** Once after completing all tasks, participants had to indicate how they perceived the transparency of the recommendations (i.e. in step 5). They provided higher scores in the TMF ($M = 4.22$, $SD = 0.89$) than in the TAG condition ($M = 4.15$, $SD = 0.82$), even though the effect size was small ($d = 0.08$), and a two-tailed *t*-test did not suggest that this difference was actually meaningful ($t(52) = 0.32$, $p = .752$). This, however, confirms H3.

■ **Usability,** ■ **interface and** ■ **interaction adequacy** Also independently of the tasks, participants had to assess the usability in step 5. In both conditions, it was rated as "good" according to the adjective rating scale from [BKM09], with a SUS score of 87 for TMF and 84 for TAG. Because of the nearly identical interfaces, we had no a priori assumption on the direction. Therefore, we used a two-tailed *t*-test for a statistical comparison, which also showed no meaningful effect ($t(52) = 1.12$, $p = .269$; $d = 0.30$). This was well aligned with the very positive assessment of interface adequacy, with $M = 4.44$ ($SD = 0.57$) in the TMF and $M = 4.20$ ($SD = 0.53$) in the TAG condition.

[31] Except for † ($df = 48.36$) and ‡ ($df = 45.51$), adjusted due to unequal variances.

At least, the effect was a bit larger for this more specific construct ($t(52) = 1.55$, $p = .128$; $d = 0.44$). In terms of interaction adequacy, the mean values indicated that participants favored the TMF ($M = 4.15$, $SD = 0.76$) over the TAG condition ($M = 3.63$, $SD = 0.69$). Here, we however *expected* an advantage due to the direct relation of this aspect to the critiquing process. Hence, we used a one-tailed t-test, which actually confirmed a medium to large effect ($t(52) = 2.63$, $p = .006$; $d = 0.72$). Thus, we can accept H4.

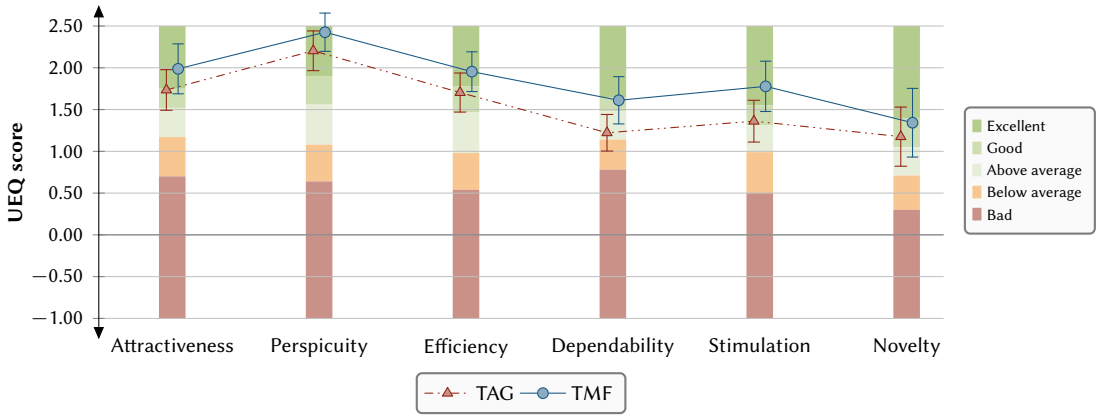

Figure 6.8 Comparison of the conditions with respect to the UEQ subscales, including 5 % confidence intervals. Benchmark values are from [SHT17].

Regarding the UEQ, values between 1.34 and 2.43 in the TMF condition on the different subscales appeared very promising as well, which is underlined by Figure 6.8: The scores for overall attractiveness, but, in particular, perspicuity and efficiency, were "excellent" according to [SHT17]. The latter provided further evidence that participants were able to learn how to use the system, and to do so in a very efficient manner, which additionally supports H3. Moreover, dependability was rated as "good". In all dimensions, scores were higher than in the TAG condition, where values ranged from 1.18 to 2.20: Overall attractiveness and efficiency were only rated as "good", dependability as "above average". This indicated that participants felt less efficient and less in control, which went hand in hand with the interaction adequacy results, and also supports H4. Especially for the pragmatic quality aspect of dependability and the hedonic quality aspect of stimulation, two-tailed t-tests together with medium effect sizes confirmed these findings. Table C.1 in Appendix C shows the results of these statistical tests.

■ **Critiquing mechanism** To investigate the impact of content boosting on the quality of the critiquing process, we defined the perception of the underlying mechanism (content-based or on top of *TagMF*) as a subjective system aspect. We applied a multivariate analysis of variance to aggregate the questionnaire items we used to assess this aspect once participants completed all tasks (i.e. in step 5). This analysis indicated no considerable difference between conditions, $F(12, 41) = 0.68$, $p = .761$, $\eta_p^2 = 0.17$. Accordingly, the individual results per item were equally positive in both conditions (cf. Table 6.3): All participants understood the critique dimensions and the effects of their application on the recommendations, which additionally supports H3. Moreover, they liked applying critiques in the form of user-generated tags. In qualitative feedback, one participant answered to the open-ended question that it was "clear and straightforward to point the system in the direction of movies [he or she] would like to watch". However, others commented

that it "would have been helpful to see a list of all tags because of the difficulty to come up with the right terms" (note that autocompletion was in fact available) and that they "missed a broader range of tags to select from". Nevertheless, in spite of the lack of high effect sizes and meaningful differences (see the one-tailed *t*-tests reported in Table 6.3), the mean values, which were always slightly in favor of TMF, additionally support H4, especially under consideration of the minor differences in the method for determining the critique dimensions.

Table 6.3 *t*-test results ($df = 52$) for a comparison of the conditions with respect to the critiquing mechanism. Higher values indicate better results on 5-point Likert response scales. The best values are highlighted in bold. *d* represents Cohen's effect size value.

	TAG		TMF				
Questionnaire item	*M*	*SD*	*M*	*SD*	*T*	*p*	*d*
The tags made sense to me	4.22	0.75	**4.48**	0.75	1.27	.106	0.35
The tags shown helped me learn about the movie	4.00	0.73	**4.26**	0.76	1.27	.105	0.35
I liked having the ability to specify critiques	4.52	0.64	**4.67**	0.68	0.82	.207	0.23
Movies displayed in response to my critique made sense	3.67	1.04	**3.89**	1.12	0.76	.227	0.20

■ **Choice satisfaction** With respect to user experience, we noted that participants in the TMF condition were more satisfied with the movies they chose from the final set of recommendations (in step 3c/4c) in all tasks. In line with the differences in mean values shown in Table 6.2, we even found statistical evidence in task IIb, and, to a limited extent, in task IIa, with medium to large effect size. Together with the equally positive result in task I, these findings support H5.

■ **Choice difficulty** On the other hand, we did not find any notable difference with respect to the difficulty of making a choice. Thus, we have to reject H6. In two cases, TAG even received better results,[28] though the effect was small in task I and negligible in task IIa (see Table 6.2).

■ **Usage effort** In terms of perceived effort, TMF scored only slightly better than TAG.[28] Especially in task I, the difference appeared negligible, while there was at least a small effect in the two other tasks (cf. Table 6.2). In turn, with respect to task times, the baseline method led to a 14.73 % improvement in task I and a 16.07 % improvement in task IIb (cf. Table 6.4). Yet, given the large standard deviations, *t*-test results and effect sizes, this more objective measurement did not appear to vary systematically across conditions. In combination with the only partially consistent questionnaire results, we thus have to reject H7.

Table 6.4 *t*-test results ($df = 52$) for a comparison of the conditions with respect to task times. The best values are highlighted in bold. *d* represents Cohen's effect size value.

	TAG		TMF				
Task	*M*	*SD*	*M*	*SD*	*T*	*p*	*d*
Task I	**7.21 min**	3.23	8.46 min	4.11	1.24	.333	0.34
Task IIa	4.37 min	2.19	**4.22 min**	2.16	-0.25	.401	-0.07
Task IIb	**5.08 min**	2.96	6.06 min	3.73	1.06	.220	0.29

■ **Intention to use again** While TMF outperformed TAG in many of the above dimensions, a two-tailed t-test did not show a better result regarding participants' intention to use this system variant again ($t(52) = 0.53$, $p = .589$; $d = 0.15$), with $M = 4.17$ ($SD = 0.97$) in the TMF and $M = 4.04$ ($SD = 0.77$) in the TAG condition. This is reflected accordingly in Figure 6.9.

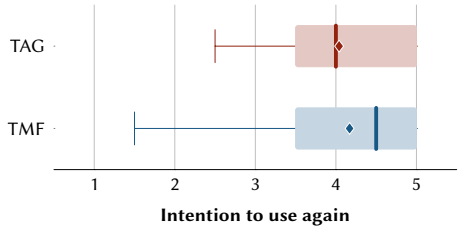

Figure 6.9 Box plot depicting the intention of participants to use again one of the methods: The thick vertical lines represent medians, the diamond signs mean values. Boxes extend from the 25th to the 75th percentile, whiskers correspond to minimum and maximum values.

6.3.2.4 Discussion

The comparison between the system variant implemented based on content-boosted matrix factorization, and the baseline with critiquing implemented exclusively based on tags, led to positive results in terms of overall satisfaction and, in particular, several more specific dimensions. Related to both subjective system aspects and user experience, these findings provided support for 5 out of our 7 exploratory hypotheses.[19] In the other cases, there were understandable reasons why our expectations regarding the impact of latent knowledge were not met. Also, the results were still promising, showing no negative impact of content boosting on the critiquing process.

Quality and diversity With respect to *perceived recommendation quality*, the very positive assessment already became apparent in the preliminary task (H1). There, participants selected a movie from the initial set of recommendations, which was generated in both conditions by content-boosted matrix factorization. This corroborated findings from the first user experiment, underlining that our method leads to high quality recommendations even before any interaction takes place. Based on the results from the three main tasks, we were also able to draw conclusions in relation to the comparison with the purely content-based baseline: The value of content boosting for the critiquing process became most evident in case participants had to find movies in line with their own interests (i.e. in task I). Apparently, the consideration of their preference profile, i.e. the user-factor vector learned during the preliminary task via conventional preference elicitation, led to recommendations that not only matched their situational needs (here induced by the task descriptions), but also their general interests, as it is customary in collaborative filtering systems. This effect was less visible in task IIa. Potentially, it was harder for participants to determine whether the items were in line with the task description, since the interests of the fictitious date had to be taken into account in order to pursue the given short-term goal. Overall, however, the outcome of the critiquing process was perceived as very positive in each task (H1). This was reflected in the more specific constructs, both related to interaction behavior, *mean item rating*, and user experience, *choice satisfaction* (H1, H5). Beyond that, whereas approaches that exclusively rely on content-based techniques (such as our baseline) are known to tend to over-specialization [Iaq*08], we actually found (small) improvements in *diversity* (H2). This is

well in line with other works that exploit latent factors to diversify the output of recommender systems or to address the filter bubble problem [e.g. WGK16; KLZ17].

Critiquing Besides aspects related to the final outcome, we also investigated the impact of the latent knowledge on the critiquing process itself. The questionnaire results suggested that participants did not become confused by the less item-oriented presentation of critique dimensions: In general, they rated *transparency* higher in the experimental condition, and found, more specifically, that the personalized selection of tags made more sense (H3). Participants also provided more positive feedback regarding the *interaction adequacy* (H4). Note that some of the differences between conditions were not large. But, we blended together the sets of tags to equal proportions, i.e. only 3 of the 6 tags were actually determined differently, and thus more user-oriented. This minor difference at the front-end, together with the between-subject design, may have diminished the effect of personalization. In addition, while effect sizes were still small to medium, the explanatory power was limited by sample size. On the other hand, participants were confronted with the corresponding questionnaire items only once, after completing all tasks. Thus, it may also have distorted the results that there were tasks in which they had to consider the interests of others (i.e. in task IIa and IIb). The answers to the open-ended question support the assumption that personalized critique dimensions were less useful in these cases: One participant mentioned that it was "difficult to change the direction of the recommendations [from horror to comedy] in order to obtain movies for a 9-year-old". Yet, he or she explicitly added that "adapting to the user profile was the very purpose of the system".

In contrast to the first study, it was not possible to use structural equation modeling because of sample size and experimental design. Further research is thus necessary to explore the effects of our method on the subjective assessment of the critiquing mechanism in more depth. Still, we deduce that these effects already became visible in the results we obtained for overall satisfaction and some other, more general constructs: As mentioned above, the content-boosted variant outperformed the baseline in terms of *interaction adequacy* (H4) without requiring more *effort* (H7), which can clearly be attributed to an improved perception of the critiquing mechanism.

Transparency and effort Returning to these more general constructs, it is worth mentioning that we found only marginal improvements with respect to *transparency* (H3). Bearing in mind that latent knowledge came into play, this finding, however, even sheds a positive light on our method: It would not have been surprising if the variant that exclusively relied on well understandable tag-based information had facilitated the comprehension of recommendations. In principle, the same applied to *perceived effort*, and the related objective measurement, time spent for tasks. Yet, we initially hypothesized that the consideration of long-term preference profiles would improve efficiency when navigating through the information space during the critiquing process. In contrast to this assumption, the subjective results showed only slight improvements, whereas task times even tended in the opposite direction (H7). In terms of *choice difficulty*, we did not find a positive effect either, also against our expectation (H6): While we assumed that the personalization of the result set would reduce the difficulty to choose a movie in the experimental condition, the system variant that implemented the method behind *MovieTuner* actually made it easier in two of the tasks to settle on one of the items.[32] However, a confounding fac-

[32] Note that in the first part of the evaluation, we additionally measured how long it took participants to make a choice. Due to differences in experimental design and task descriptions, this was not possible this time.

tor might have been the high quality of this set in the content-boosted variant, which is known to complicate decision making [Bol*10]. Structural equation modeling could help to get a better understanding of the relationships between these constructs, especially in light of the (more positive) results with respect to *recommendation diversity* (H2) and *choice satisfaction* (H5).

Usability and interaction Notwithstanding some unfulfilled expectations related to specific aspects of user experience, the general *usability* assessment indicated that participants were in favor of the content-boosted variant of the system. Similar to transparency, we did not even expect this. As in the first user experiment, most usability-related comments were independent of the method: In their qualitative feedback, participants indicated that they wanted to "directly search for movies" and to "exclude bad, but keep good movies over several critiquing cycles". These issues need to be considered in future iterations of the prototype system. But, they are more related to system use in real-world settings. More importantly, participants also found the *interaction much more adequate* in the experimental condition (H4). This underlines that gearing the entire process towards the current user may considerably contribute to a more positive image of the interaction in recommender systems, regardless of required effort or choice difficulty.

Summary Taken all together, our method again showed its potential for adding more expressive interaction possibilities to collaborative filtering systems, without losing the ability to generate recommendations based on regular user-item feedback. Since we designed the second user experiment as a comparison between two system variants, one using a content-boosted latent factor model, the other the pure content-based technique of the well-known *MovieTuner*, the value of the latent knowledge for implementing an interactive recommending approach became clearly visible: Independent of task-specific circumstances, we found positive results in relation to subjective systems aspects, user experience, and interaction behavior. Given the exploratory nature of the experiment with its limited sample size, the few visible differences between the system variants, and the confusion that could have been caused by the latent factors, even the absence of larger effects in some of the dimensions cannot detract from the value of content boosting. Nonetheless, larger user studies, but also simulation studies as performed with other critique-based approaches [cf. Xie*18], are required to confirm the current findings, examine the effects of specific parameters (e.g. number of critique dimensions), and obtain more objective results (e.g. with respect to the duration of the critiquing process). In summary, however, the remaining application possibility of our content-boosted matrix factorization method that we wanted to address in this part of the evaluation can be considered successfully validated. Again, we can conclude that *leveraging item-related information* in addition to standard collaborative filtering feedback data is an effective means to improve *user control and experience* (RQ2).

CHAPTER 7

> "No intelligent idea can gain general acceptance unless some stupidity is mixed in with it."
>
> — *Fernando Pessoa, Portuguese poet*

Blending recommendation methods with information filtering

In this chapter, we finally propose a concept to improve user control also in cases in which it is hardly possibly to guide users to their search goal exclusively relying on collaborative filtering, even if the advanced interaction mechanisms proposed in the previous chapters are available. As we have argued in Section 3.1.3, these mechanisms allow users to directly intervene in the underlying model, but may be insufficient as scenarios become more complex: If multiple methods are responsible for the system's outcome, their influence is limited to the model-based algorithm. Methods that are more interactive by nature, for example, from information filtering, mostly stand on their own. For these reasons, we introduce *blended recommending* [Her*14; LHZ15a; LHZ15b], directly addressing our third research question: As suggested in Section 3.2.3, we employ a common hybridization strategy as a means to merge model-based collaborative filtering with other recommendation methods, but hand control of the resulting combination over to users by designing the front-end using faceted filtering. This concept enables users to adjust the final outcome in a holistic manner, while the benefits of each individual method are preserved, including those of model-based collaborative filtering components, which possibly implement the other enhancements. In the following, we describe the *background* in more detail. Afterwards, we explain the *method* and present an *empirical evaluation* we conducted to study the effectiveness of our concept by comparing it with a baseline filtering interface [LHZ15a; LHZ15b].

7.1 Background

The motivation for the concept of blended recommending arises from the situation users often face in real-world systems: On an e-commerce website such as *Amazon*, users may flexibly browse through the item database, sort the results, and apply filter criteria to constrain the product set (see the screenshot in Figure 1.1 in the introductory chapter). Within such a set, however, recommendations play a minor role, for instance, in the form of items that are specifically featured or advertised. Actual recommendations usually appear separately, mostly only as suggestions for similar products on item detail pages. On a platform such as *Netflix*, in contrast, users are presented with a personal homepage, proactively tailored to their interests by showing numerous rows of items the system considers to be of relevance (see again Figure 1.1). There, in turn, it is neither possible nor intended to let users search, sort, or filter. Yet, only allowing for

consumption of recommendations may not account for the complexity of the decision process. Especially in case of experience products or products users only rarely (or never) are confronted with (e.g. high-risk products), but also in case of high domain expertise or very specific situational needs, this may be a problem. Consequently, our main objective is to support users also in such more complex situations, in which system-initiated personalization based on state-of-the-art collaborative filtering methods is not sufficient—not even if the preference elicitation methods and interactive features we proposed as extensions come into play.

Inspiration from hybrid recommender systems Accordingly, the first requirement is to make other recommendation methods available in an adequate manner. For this, numerous strategies exist (see Section 2.1.5). As we have illustrated in Section 3.1.3, systems such as *TasteWeights* [BOH12] already implement mechanisms for controlling the interaction between algorithms and the impact of underlying datasources. In some works, users can even switch from method to method [e.g. Eks*15] (cf. Section 2.3.2.2). Yet, from a user perspective, it might not be of interest to weight or select a specific algorithm. Instead, to strengthen the corresponding connection in our model of user interaction (see Figure 3.1), it should be possible to manipulate the influence of individual methods on the final output of the system *without* the need to deal with its inner workings, or to know about technical details—neither of the methods, nor of the way they are combined. Against this background, it seems promising to step on a level superordinate to that of algorithms and datasources. In terms of movies, for example, users should be able to request recommendations similar to "Pulp Fiction", from another decade, starring a certain actor, or with elements of specific genres.[33] This is not possible only with the help of extensions directly to model-based collaborative filtering: On the one hand, because the information that would be needed cannot be put into relation to the dimensions of a typical collaborative filtering model in the same way as the specific item-related information is by means of our content-boosted matrix factorization method. On the other hand, because such a model often represents only one part of the whole system, so that other parts would remain outside the user's control.

Realization with faceted filtering Consequently, the second challenge is to find a basis for the manipulation of the *combined* results based on items, their properties, and related entities. Basic information filtering methods may help narrow down the item space by eliminating results that do not match exactly the criteria specified by the user. This limits the degrees of freedom in which results can be presented and ordered. Moreover, the available criteria have to cover the dimensions that are relevant for the user in light of his or her search goal. If this goal is more complex and users need to actively explore in order to acquire further knowledge on how it can be reached (i.e. the opposite of known-item search), these basic methods may not be sufficient [NH14; KFK14]. Interfaces using *faceted filtering*, on the other hand, are known to provide more effective information-seeking support, even for exploration and discovery of very large item spaces (cf. Section 2.3.3). Several attempts have been made for integrating intelligent methods, for instance, with regard to the extraction, adaptation, or suggestion of facets and facet values [e.g. KZL08; Tva*08; CAS11; Voi*12]. However, this has not yet led to a combination with modern methods for generating personalized *item* recommendations, although we have argued in Section 3.1.3 that this would contribute to support users even in complex scenarios with the options they need to reach their search goal. Taken for itself, faceted filtering on the other hand

[33] This example corresponds to the screenshot in Figure A.6 in Appendix A, showing the blended recommending interface of the prototype system that we implemented for the corresponding user experiment (see Section 7.3).

enjoys great popularity for interactive information retrieval, and has been successfully applied in digital libraries or online shops. Figure 7.1 illustrates this again with the example of *Amazon*.

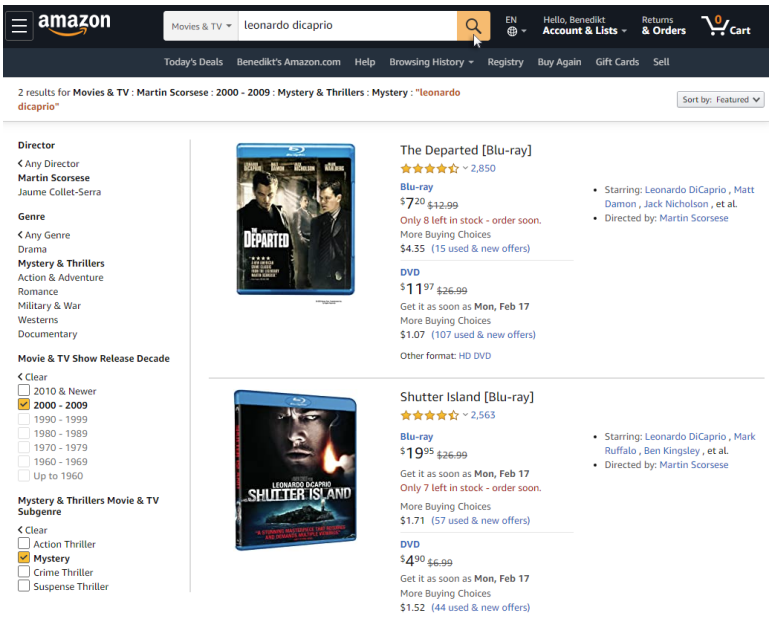

Figure 7.1 Example of a faceted search on *Amazon* for mystery movies from the 2000s, starring Leonardo DiCaprio and directed by Martin Scorsese.

In light of the call for convergence of search and filtering mechanisms with recommendation functionalities [GKP11], it thus appears to be a natural step to use faceted filtering, despite its limitations (see also Section 2.3.3), as a point of departure for *merging collaborative filtering with other recommendation methods*: By implementing each method in the background of a certain facet, meaningful criteria may be shown at the front-end, based on items, their properties, and related entities. Then, the recommendation functionalities have the potential to alleviate the otherwise cognitively demanding task of mentally forming a search goal, especially in large or unknown domains or in case of experience products. By offering additional options to weight the criteria, we eventually aim at providing users a list of recommendations that is always ranked according to their actual preferences, without requiring them to deal with algorithms or data-sources. At the same time, the hybrid recommendation approach may avoid the strict conjunctive application of filter criteria as known from most attempts to information filtering. In contrast to real-world systems, which process queries in a strictly logical manner, this should prevent users from over-constraining their search, and systems from producing empty result sets.

7.2 Method

In the following, we elaborate on how to *design an interface* in line with the above considerations for combining model-based collaborative filtering with content- and knowledge-based techniques as well as information filtering methods. Next, we present further details on our concept. Concretely, we describe how to *implement the facets based on arbitrary recommendation methods*, and how *recommendations can then be generated* in accordance with selected criteria.

7.2.1 Designing the interface

To fulfill the requirements we discussed above, we propose the following design for the user interface of a system that implements our concept of blended recommending: As illustrated in Figure 7.2, the *facets*, which the user may use for setting the filter criteria, are shown on the left-hand side. In the schematic example, all facets but the first are displayed collapsed. For the expanded facet, the corresponding facet values are shown. Other facets may be opened at any time as well. However, users should not only be able to *select* filter criteria, i.e. values from these facets, but also to *weight* the influence of the underlying recommendation methods on the final results. Consequently, they are not provided with lists of checkboxes as in conventional faceted filtering. Instead, users can drag the facet values they would like to select and drop them into the *working area* shown in the middle of the screen. There, each value is accompanied by a *slider* to adjust the weight of the corresponding method. In the example, two values have been selected from the first facet, which accordingly do not appear anymore within the facet widget.

Using drag and drop allows to deviate from the typical presentation of facet values. Moreover, showing the interaction widgets for specifying the weights within the working area avoids cluttering the facets area. As a consequence, values can be displayed as *tiles*, showing meaningful images or pictograms. In the domain of movies, this may include movie posters, actor portraits, or genre illustrations, thereby reducing recognition time and cognitive load in comparison to a pure textual variant. Yet, for facets with a large number of values, only the most relevant ones can thus be shown. For these cases, we suggest to add a search function, or a function that automatically suggests values the system considers helpful for refinement.

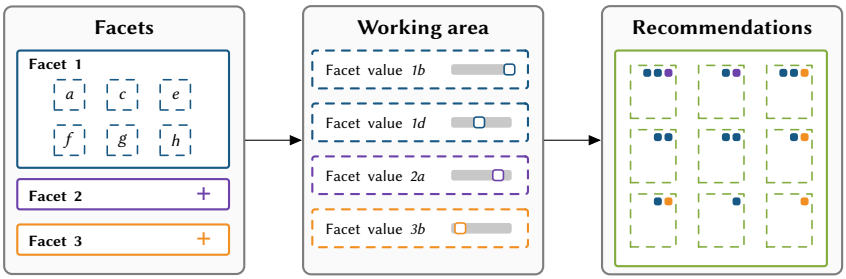

Figure 7.2 Schematic example of the interface design suggested by our concept of blended recommending: Facets, which are initially collapsed, and corresponding values are shown on the left-hand side. If facet values are selected, they are displayed in the working area in the middle of the screen, accompanied by sliders. There, users can adjust the weight of each selected value. Correspondingly, the recommendations are updated, which are shown on the right-hand side together with visual clues indicating their source within the hybrid configuration.

Based on the output of the methods that are related to selected criteria, and under consideration of the assigned weights, the *recommendations* that try to satisfy the ad hoc preferences expressed in this way are shown on the right-hand side. Since all the above interactions of the user are directly reflected back into the underlying algorithms, he or she can continuously adapt this set in realtime, allowing to explore the effects of different values and weights. In addition, as illustrated by the consistent use of colors in Figure 7.2, visual clues may be given to convey the source of each recommendation and to facilitate the user's understanding of the results.

It would, of course, be possible to take into account the user's long-term profile instead of only ad hoc preferences. However, this is neither required for our approach, nor would it make much of a difference for the actual implementation, because collaborative filtering is per se considered to be part of the hybrid configuration. In the following, for the sake of simplicity, we thus refer only to explicit input given by the user during the *current* session.

7.2.2 Implementing facets and recommendation methods

Because of the broad range of recommendation methods, a system that implements our concept needs to be capable of providing facets of different types. For this, however, the only requirement is that for each facet value the active user has selected, the relevance of all items can somehow be determined. We define a set C to represent all selected values, and $\text{rel}(i, c) \in [0, 1]$ to describe the degree to which item i fulfills a criterion $c \in C$. Then, the exact calculation depends on the underlying method and the data used by this method. In the following, we use facets of movies for illustration purposes. Without loss of generality, the differences between these facets allow us to show the computation of relevance scores based on many common recommendation methods.

Boolean filtering First, if users select a value from a facet such as movie genre, director, or age rating, each movie with *exactly this value* needs to be considered in the results, others to be ignored. This hard filtering would lead to a large number of items receiving a relevance score $\text{rel}_{\text{bool}}(i, c)$ of 1.0, i.e. the maximum value. For example, all items from the action genre would be ranked equally. To avoid this, we assume that more popular items are more important from a user perspective. Accordingly, we adjust the relevance scores determined with respect to a criterion c by artificially ordering all items with the same score. For this, based on insights gained from preliminary experiments, we make use of the formula the *Internet Movie Database* (IMDb) uses for its top 250 movie charts:[12]

$$\text{pop}(i) := \frac{\overline{r_i} \cdot |R_i| + \overline{r} \cdot d}{|R_i| + d}, \tag{7.1}$$

taking into account both the number of ratings $|R_i|$ of an item i and its average rating $\overline{r_i}$. $\overline{r}$ represents the mean rating across all items. Applying this formula instead of a simpler, more common definition of popularity has the effect of adjusting the average ratings towards the global mean. The constant $d \in \mathbb{R}$ allows to control this adjustment, and needs to be determined empirically.

Fuzzy filtering In some cases, however, users may be uncertain about the value they should select. This calls for a soft filtering of criteria, which is why we use *fuzzy logic* [Zad65] in order to calculate the relevance scores for certain facets, thus avoiding that users need to find exact matches as in most information filtering systems. Instead, for a facet such as the year a movie was released, they can select a criterion c that is related to a specific decade such as the 2000s. Then, movies from this decade are included, but also movies released a few years before or afterwards, though with linearly decreasing scores. Consequently, a movie from 1999 would not be completely ignored as in the case of Boolean filtering. The same is applicable to movie length, allowing users to select criteria that represent "short", "normal", or "extra long" movies, corresponding to, for instance, < 80, $90 - 120$, or > 130 minutes in length. Then, using a *fuzzy membership function*, movies falling within these intervals receive a maximal relevance score $\text{rel}_{\text{fuzzy}}(i, c)$, while movies in between, i.e. which only partly satisfy the respective criterion, are considered less relevant. Figure 7.3 illustrates such a fuzzy membership function μ.

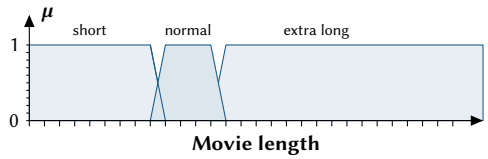

Figure 7.3 Illustration of a fuzzy membership function that allows to determine the relevance of a movie in terms of the desired length.

Content-based filtering For considering criteria beyond the basic item properties addressed above, it is required to implement facets based on *content-based* recommendation methods. For instance, if users select a facet value that corresponds to a certain keyword, the relevance score $\text{rel}_{\text{cbf}}(i, c)$ with respect to this criterion c may be calculated using a *TF-IDF heuristic* [BR99]. For this, inspired by tag-based recommender systems such as *MovieTuner* [VSR12], we consider tags as terms, the sets of tags associated with movies as documents, and subsequently calculate the relative importance of each tag a for a movie i as follows:

$$\text{tf-idf}(a, i) \coloneqq \text{tf}(a, i) \cdot \text{idf}(a) = \frac{\text{freq}(a, i)}{\max(i)} \cdot \log \frac{|I|}{|I_a|}, \tag{7.2}$$

with $\text{freq}(a, i)$ returning the frequency of tag a for movie i, $\max(i)$ being the number of times the most frequent tag was assigned to this movie, and I_a representing the set of all movies associated with tag a. Carrying out this calculation for all tags and movies results in TF-IDF vectors that allow determining the most relevant movies by comparing them with a vector that corresponds to criterion c. In principle, however, *any* other content-based method may be used as well: On the one hand, $\text{rel}_{\text{cbf}}(i, c)$ may be set simply by using predefined values from the underlying dataset, for instance, in case it is known for actors how important their role is. On the other hand, as we have proposed in related work [Feu*17], a complex pipeline may be used to derive the necessary data from user-written product reviews by means of natural language processing techniques.

Collaborative filtering In addition to all the superordinate criteria that may be taken into account based on the aforementioned facet types, users might still want to indicate preferences on an *item level*. Therefore, we also offer an "items similar to" facet, allowing users to tell, for example, that they would prefer a movie similar to "Pulp Fiction". For this, items need to be presented as facet values from which users can select those they know and like. This may be done based on popularity, the current recommendation set, or the user's long-term profile. Either way, as soon as the user selects one of these items as a criterion c, movies that received similarly positive feedback by the user community can be assigned a high relevance score. To determine $\text{rel}_{\text{cf}}(i, c)$, we rely on matrix factorization, and compute the similarities between each item i, and the respective item j, in terms of latent factor vectors as follows:

$$\text{sim}_{\text{dist}}(i, j) \coloneqq \frac{1}{\|\vec{q}_i - \vec{q}_j\|}. \tag{7.3}$$

Other collaborative filtering techniques may be used as well. However, the application of matrix factorization serves as a basis for all developments presented earlier in this thesis. To finally merge the methods mentioned before with collaborative filtering, we thus decided to use this kind

of algorithm also in context of blended recommending. In any case, this goes beyond conventional information filtering—which always requires some kind of content attributes—and may be beneficial for users since they do not need to know exactly what they are looking for: Similar to critique-based approaches or our choice-based preference elicitation method, they are provided a starting point from which further exploration of the recommendations is possible.

7.2.3 Generating recommendations

The final step is to come up with a set of recommendations that combines the results of the individual methods under consideration of the weights specified by the current user as described in Section 7.2.1. To make our concept independent of specific methods, we use a *loosely-coupled* hybridization approach (cf. Section 2.1.5). Concretely, we determine an overall relevance score for each item i using a weighted arithmetic mean. In line with *multi-attribute utility theory* [BB09], this score aggregates the relevance scores $\mathrm{rel}(i, c)$ calculated separately as illustrated in the previous section, weighted according to the assignments made by the current user u, for which we define $\mathrm{weight}(u, c) \in [0, 1]$. Thus, the *recommendation function* may look as follows:

$$\mathrm{s}(i|u) := \frac{\sum_{c \in C} w_{uc} \cdot \mathrm{rel}(i, c)}{\sum_{c \in C} w_{uc}} \quad \text{with } w_{uc} := \mathrm{weight}(u, c)\,. \tag{7.4}$$

All items can now be sorted in descending order. Table 7.1 shows a toy example: The user searches for a movie directed by Martin Scorsese, and specifies a weight of 1.0 for this criterion. Moreover, the user wants another criterion to be fulfilled, namely that the movie should be from the 2000s, but with a lower weight of 0.5. For demonstration purposes, we assume that the dataset consists only of three items and dispense ordering the items in case of equal relevance scores.

Table 7.1 Example showing how the recommendation function in blended recommending is applied.

	Director $\mathrm{rel}_{\mathrm{bool}}(i, \textit{scorsese})$	**Release year** $\mathrm{rel}_{\mathrm{fuzzy}}(i, \textit{2000s})$	**Overall relevance** $\mathrm{s}(i\|u)$
The Departed (Scorsese, 2006)	1.0	1.0	1.00
Bringing out the Dead (Scorsese, 1999)	1.0	0.5	0.83
Inglourious Basterds (Tarantino, 2009)	0.0	1.0	0.33

Subsequently, the items with the highest scores can be presented as recommendations. This does not require that the user knows what happened on part of the system, i.e. which individual methods were responsible for the aggregated results. From the perspective of system providers, this has the advantages that the algorithms are easily exchangeable and can be chosen according to domain-specific requirements, as shown in the previous section for movies. Note, however, that *before* the aggregation takes place in (7.4), it may be necessary to scale or normalize the values calculated by the $\mathrm{rel}(i, c)$ functions in case they are on different scales. Alternatively, the items may be sorted with respect to each single criterion in order to adjust the relevance scores according to the position of the items in the respective result list. Yet, as this is an application-specific problem, we omit further details at this point.

Note further that with this ranking approach, empty result sets cannot occur. Still, certain filter settings may lead to few results. This particularly applies if non-matching items must be excluded entirely from the final output due to hard Boolean filtering (e.g. in case of age rating restrictions). In these rare situations, we extend the recommendation set dynamically: Items are added, which are both popular and similar in terms of latent factors as shown in (7.3), to the set of already recommended items, or to the items from the current user's long-term profile, if this exists.

7.3 Empirical evaluation

The last user experiment we report in this thesis is concerned with the evaluation of the effectiveness of blended recommending and the user experience that results from interacting with a system that implements this concept. For this, we developed a prototypical recommender system, again with movies as a running example. Using this system, we conducted an exploratory study with $n = 33$ participants as a last step of an extensive user-centered design process. Participants were asked to use either a variant of this system that embedded a conventional filtering interface as a baseline, or a variant that followed our novel concept, and to fill in a questionnaire.

In the following, we describe the underlying *goals* as well as the *hypotheses* we derived accordingly. Afterwards, we explain the *method*, including a brief summary of the earlier stages of the user-centered design process (reported in [Her*14]), the prototype system and the datasets, the questionnaire, and the procedure of the main experiment, which was originally published in [LHZ15a; LHZ15b]. Finally, we present the *results* and conclude the chapter with a *discussion* that takes up our third research question.

7.3.1 Goals and hypotheses

Our concept builds on the assumption that faceted filtering is an appropriate point of departure for merging model-based collaborative filtering with other recommendation methods in order to support users even in more complex scenarios. For this reason, our main goal was to evaluate effectiveness and user experience in comparison to a *system that uses only conventional faceted filtering*, as known from many real-world applications. Since we described our concept with a focus on ad hoc preferences from the current session, this appeared to be a natural baseline with a similarly high level of interactivity.[34] However, we hypothesized that a system based on blended recommending would give users an even stronger feeling of control. At the same time, we expected better results because of the advanced options to indicate individual preferences. Due to the integration of collaborative filtering, and thus the possibility to indicate some of these preferences on an item level, we assumed this would be the case especially in situations with less specific search goals. Consequently, such a system should also be perceived as more effective, among others, because over-constraining the search as with strict logical query processing is not possible. In line with this, we expected that the interaction possibilities would appear more adequate. On the other hand, they could also be perceived as more effortful. Yet, we considered browsing and filtering to be equally cumbersome when there are only conventional mechanisms available, whereas potential deficiencies of blended recommending should at least

[34] A comparison with typical recommending approaches would have been unfair, especially given that the tasks had no relation to long-term preferences. Thus, we dismissed the idea of using additional baselines.

be compensated by the constantly present recommendations. Similarly, we expected that any negative effects on general usability would appear negligible in light of the other advantages.

While our research design was again exploratory, we considered it useful to translate these assumptions into the following *hypotheses* in order to investigate the possible advantages over a baseline filtering interface in a more structured manner:

H1 Blended recommending leads to recommendations of higher perceived *quality*.
H2 Blended recommending leads to similar system *usability*.
H3 Blended recommending improves perceived *interaction adequacy*.
H4 Blended recommending improves perceived *effectiveness* of the system.
H5 Blended recommending increases the feeling of *control*.
H6 Blended recommending reduces perceived *usage effort*.

7.3.2 Method

Again, the user experiment took place in a controlled laboratory setting. We recruited $n = 33$ participants (13 female, 20 male) with an average age of 27.18 years ($SD = 6.46$). A supervisor was present, but participants went through the experiment on their own using *SosciSurvey*, which we also used to present the questionnaire.[25] Participants used a desktop PC with 24" LCD (1920×1200 px resolution) and a common web browser to answer the questionnaire items and to interact with the prototype system that we implemented for this experiment.

The development of this system followed a user-centered design process with several prestudies: First, we conducted a preliminary study with $n = 22$ participants (10 female, 12 male; average age of 30.31 years, $SD = 12.72$) to evaluate several options with respect to the ordering of facets and the presentation of their values by means of mockups. Second, we implemented a basic prototype system and ran a follow-up study with $n = 30$ participants (17 female, 13 male; average age of 23.97 years, $SD = 2.98$) to capture the visual impression of an interface designed according to our concept. Third, with the insights gained in these experiments, we implemented a first running version of *MyMovieMixer*, a movie recommender based on blended recommending, already quite similar to the one described in this section. We performed another user study with $n = 30$ participants (14 female, 16 male; average age of 28.26 years, $SD = 10.18$) to analyze its usability and the general acceptance of the underlying concept. More details can be found in [Her*14].

Prototype In light of the aforementioned goal, we implemented the prototype system for the main experiment again as web application in two different variants:

- A standard *faceted filtering interface* implemented as described in the literature [cf. Yee*03; Hea09]: As shown in Figure A.5 in Appendix A, participants were able to freely interact with this system variant to explore the space of available items. We used the underlying metadata dataset to represent typical movie properties as facets and facet values. Initially, the result table was ordered with respect to item popularity. As it is common practice, selected criteria were combined using Boolean AND operations. Hyperlinks were provided to facilitate the filtering process, allowing participants, for example, to click on a person's name in order to apply the corresponding filter criterion. For certain columns of the result table, filtering mechanisms and sorting functionalities were provided in the column head.

- An interface implemented according to the concept of *blended recommending*: This system variant represented another iteration of *MyMovieMixer*, designed based on the insights gained in the prestudies. Figure A.6 shows the front-end, including the different areas as suggested in Section 7.2.1. We adopted the design of the baseline and implemented all features as similar as possible, but integrated facets and facet values as suggested in Section 7.2.2. Using the same metadata allowed us to offer the same facets (genres, actors, directors, keywords, release year, movie length, age rating). To compute item similarities based on latent factor vectors and to offer an additional "items similar to" facet, we employed the `FactorWiseMatrixFactorization` algorithm from the *MyMediaLite* recommender library [Gan*11] with 10 factors (as in the first user study, see Section 4.3.2). For the facets based on content-based filtering, we calculated relevance scores either by means of fuzzy membership functions (release year and movie length) or a TF-IDF heuristic (actors and keywords). The initially shown top 9 recommendations were again ordered with respect to item popularity, but afterwards, with respect to aggregated relevance scores under consideration of assigned weights as described in Section 7.2.3.

For the purpose of the study, we added a shopping cart functionality to both system variants.

Datasets As in the first two user experiments reported in this thesis, we used the *MovieLens 10M* dataset to implement the collaborative filtering functionalities.[14] Since this dataset was considered the de facto reference dataset at the time of the study, we expected to ensure a sufficient degree of generalizability in this way. To implement the other facets, but also to present items in an adequate manner, we gathered additional metadata, including detailed information on genre, director and cast, but also plot descriptions and tags as well as movie posters. For this, we used the *HetRec '11* dataset[15] and imported missing data from the *Internet Movie Database* (IMDb)[16]. This left us with a dataset similar to the one we described in Section 4.3.2 and 6.3.1.2.

Questionnaire and log data Again, the questionnaire was primarily based on the work by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12] and Pu, Chen, and Hu [PCH11], who suggested items for the assessment of *subjective system aspects* (SSA) and *user experience* (EXP). Figure 3.2 provides an overview of these aspects and their relations with each other. More concretely, we used items by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12] to measure ■ *perceived recommendation quality* and ■ *perceived recommendation diversity*. To assess general ■ *usability*, we used the *system usability scale* (SUS) by Brooke [Bro96] and the *user experience questionnaire* (UEQ) by Laugwitz, Held, and Schrepp [LHS08]. In addition, we used items suggested by Pu, Chen, and Hu [PCH11] to measure ■ *interface adequacy* and ■ *interaction adequacy*. On a more specific level, we also asked participants several questions regarding the ■ *sliders and visual clues* that were exclusively available in the blended recommending interface. With respect to user experience, we assessed ■ *perceived system effectiveness* and ■ *perceived control* by means of items suggested by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12], and ■ *usage effort* by means of a self-generated item.

With respect to more *general aspects* (GEN), we assessed the ■ *overall satisfaction* of participants, again using an item by Pu, Chen, and Hu [PCH11]. We developed additional items to let participants rate the ■ *suitability for different usage scenarios* of the respective system variant, i.e. with search goal, with a vague search goal, or with no search goal. With respect to *personal characteristics* (PC), we collected ■ *demographic data* and asked participants regarding their movie

■ *domain knowledge*. Apart from UEQ (7-point bipolar scale ranging from −3 to 3), all items had a 5-point Likert response scale. An overview of all constructs and the exact questionnaire items can be found in Appendix B. We also collected qualitative feedback using an open-ended question. In addition to the questionnaire, we measured task times and recorded screencasts of the *interaction* (INT), among others, to analyze the ■ *number of filter criteria* used by participants.

Procedure Based on the system variants, participants were assigned to one of the following conditions, for which we considered the respective interface as an *objective system aspect* (OSA):

FFI In this condition, participants were presented with the system variant based on the conventional ■ *faceted filtering interface*. Participants were allowed to use all available navigation aids, search and filtering mechanisms (see Figure A.5 in Appendix A).

BRI In this condition, the system variant based on ■ *blended recommending* was shown to participants. They were also allowed to use all available mechanisms, including tile-based interaction, weighting of criteria, and the "items similar to" facet (see Figure A.6).

Since participants' use of one system variant would have too much influenced their behavior with the other, we opted for a between-subject design and randomly assigned participants to conditions (yielding $n = 16$ in the FFI, $n = 17$ in the BRI condition). Based on this assignment, they received a brief introduction to the experiment and the respective system variant by the supervisor. Then, the *experimental phase* started, in which participants were asked in both conditions to perform the following tasks in the given order:

Task I The first *task* was designed as a *training trial*, allowing participants to learn about the respective interface (step 1a in Figure 7.4). This way, we wanted to ensure that participants in both conditions were able to start from the same level in the second task. They had to assume to buy a DVD as a gift for a friend who prefers action and romance movies, and likes the actor Brad Pitt. Appropriate items should be added to the shopping cart.

Task II In the main *task*, participants were asked to find movies in line with their *personal preferences* (1b). The only requirement was to put at least one movie they would like to watch into the shopping cart, or more, if they wanted to.

Participants were able to finish each task at their own discretion. After finishing both tasks, they had to assess the system and the *results* of the main task, i.e. the final shopping cart contents (1c). For this, they were asked to leave the web application and fill in the questionnaire (2).

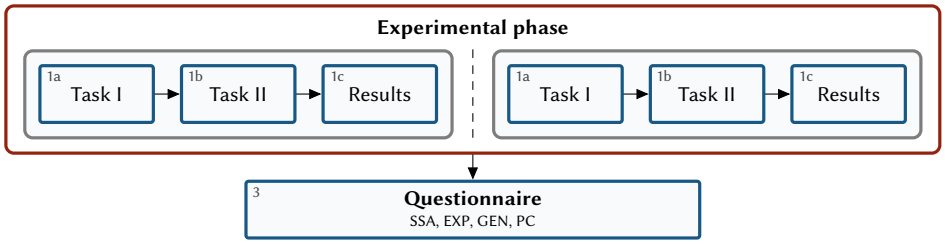

Figure 7.4 Overview of the procedure. See the text for a detailed description of the steps 1a–1c and 2.

7.3.3 Results

In the following, we briefly report the findings from the *prestudies*. Then, we present the *quantitative results* of the main experiment. We start with domain knowledge and overall satisfaction, continue with subjective system aspects and aspects related to user experience, and finish with general results. Afterwards, we present an *interaction analysis* that we additionally carried out.

Prestudies As outlined at the beginning of this section, the development of the experimental system variant followed a user-centered design process. In the *first prestudy*, we asked participants to rank criteria in terms of their usefulness for the recommendation process and to compare several layout options for different facet types. The results were used for the implementation of the basic prototype system for the *second prestudy*. Using the *VisAWI* questionnaire for measuring the visual aesthetics of websites [MT10], we obtained promising results and valuable feedback that contributed greatly to the version of *MyMovieMixer* that we used in the *third prestudy*. There, participants indicated a high usability and responded positively to questionnaire items regarding ease of use and comprehensibility of tiles as well as accompanying sliders, and regarding the quality of the resulting recommendations. Nonetheless, qualitative comments suggested potential for further improvements. This led us eventually to the version of *MyMovieMixer* that we used for the *main experiment*, the results of which are reported below.

Quantitative results With respect to domain knowledge, most participants indicated on a 4-point scale (from "few" to "very many") that they knew "many" movies ($M = 2.82$, $SD = 0.73$).

For the directional hypotheses, we conducted one-tailed t-tests to explore the effects of the objective system aspect on the dependent variables. The results are shown in Table 7.2. In the following, we elaborate in more detail on the comparison of the FFI and the BRI condition.[17] In addition, we report differences in the results for the BRI condition with regard to domain knowledge. In this and in some other cases, we did not hypothesize a direction, so that we conducted two-tailed t-tests. This is always indicated below.

■ **Overall satisfaction** First, however, we address the more general construct of overall satisfaction. As shown in Figure 7.5, participants were satisfied with the respective system variant both in the FFI ($M = 3.69$, $SD = 0.87$) and the BRI condition ($M = 3.76$, $SD = 1.03$). In line with the specific hypotheses, we applied a one-tailed t-test, which did not indicate a considerable difference ($t(31) = 0.23$, $p = .410$). Also, we noted only a very small effect ($d = 0.07$).

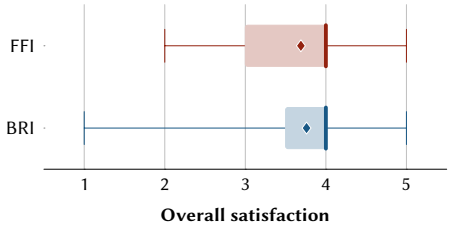

Figure 7.5 Box plot depicting the overall satisfaction of participants with the different methods: The thick vertical lines represent medians, the diamond signs mean values. Boxes extend from the 25th to the 75th percentile, whiskers correspond to minimum and maximum values.

Table 7.2 *t*-test results ($df = 31$)[35] for a comparison of the conditions in terms of subjective system aspects, user experience, and suitability for different usage scenarios. Higher values indicate better results on 5-point Likert response scales (*usage effort* is reversed accordingly). The best values are highlighted in bold. *d* represents Cohen's effect size value.

	FFI		BRI				
Construct	*M*	*SD*	*M*	*SD*	*T*	*p*	*d*
Perceived recommendation quality	**4.15**	0.48	3.99	0.45	-0.96	.173	-0.34
Perceived recommendation diversity	**3.41**	0.96	3.15	0.88	-0.81	.214	-0.28
Interface adequacy	3.86	0.60	**4.07**	0.40	1.21	.117	0.41
Interaction adequacy	3.13	1.00	**3.94**	0.53	2.90†	.004	1.02
Perceived system effectiveness	3.45	0.45	**3.66**	0.51	1.29	.103	0.44
Perceived control	3.85	0.99	**4.43**	0.50	2.10‡	.024	0.75
Usage effort	4.25	0.58	**4.47**	0.72	0.97	.170	0.34
Suitability							
with a search goal	**3.50**	1.27	2.47	1.46	-2.16	.020	-0.75
with a vague search goal	**4.31**	0.70	4.24	0.66	-0.32	.374	-0.10
without a search goal	2.80	1.27	**4.13**	1.09	3.13††	.002	1.13

■ **Perceived recommendation quality** Regarding the more specific constructs, we start by noting that the quality of the final outcome of the main task actually appeared slightly worse in the BRI than in the FFI condition: As visible in Table 7.2, the effect size was rather small, but the mean value was in fact lower. Therefore, we cannot accept H1. Still, the scores were overall satisfactory in both conditions, which was reflected in qualitative comments: Even in the BRI condition, some participants attested that "the selection of movies was surprisingly good".

Beyond that, we found that participants with poor domain knowledge, i.e. who stated to know only "few" or "rather few" movies (see the questionnaire item in Appendix B), rated the recommendation quality in the BRI condition lower ($M = 3.63$, $SD = 0.36$) than those with more expertise, i.e. who stated to know "many" or "very many" movies ($M = 4.24$, $SD = 0.40$). A two-tailed *t*-test confirmed a large effect ($t(12) = 2.81$, $p = .016$; $d = 1.58$).

■ **Perceived recommendation diversity** We found the same tendencies for the other subjective system aspect that was directly related to the final outcome: As shown in Table 7.2, FFI slightly outperformed BRI in terms of perceived diversity. However, the results were rather average this time, which is in accordance with the fact that we had no expectations regarding this variable. Also with respect to differences caused by domain knowledge, the results were similar, with $M = 2.60$ ($SD = 0.82$) for participants classified as having low domain knowledge, $M = 3.33$ ($SD = 0.87$) for others. Whereas the effect size was large ($d = 0.86$), a two-tailed *t*-test did not indicate a meaningful difference ($t(12) = 1.54$, $p = .149$).

■ **Usability,** ■ **interface and** ■ **interaction adequacy** SUS scores of 84 in the FFI and 82 in the BRI condition were equally "good" according to [BKM09]. As we had no directional hypothesis with respect to general usability, we used a two-tailed *t*-test to analyze this result: Also in a statistical sense, the scores appeared to be on the same level ($t(31) = -0.26$, $p = .796$; $d = -0.09$).

[35] Except for † ($df = 22.47$) and ‡ ($df = 21.82$) adjusted due to unequal variances, and †† ($df = 29$) with missing answers.

As illustrated in Figure 7.6, both system variants also achieved positive results with respect to the more specific subscales of the UEQ. Thus, H2 can already be confirmed. However, BRI actually outperformed FFI in all UEQ dimensions. Especially in terms of attractiveness, with a score "above average" in the BRI, but "below average" in the FFI condition according to [SHT17], as well as stimulation and novelty, two-tailed t-tests together with medium to large effect sizes underlined the positive influence of blended recommending. Whereas BRI obtained "good" results for the two hedonic quality aspects, the FFI scores were "below average" or "bad", respectively. The differences for the pragmatic quality aspects, perspicuity, efficiency, and dependability, were much smaller. Apparently, participants valued the novel interface, but felt capable of performing their task in both conditions, which was in line with the aforementioned subjective system aspects. Table C.2 in Appendix C shows the exact mean values and standard deviations for the six UEQ subscales, including the results of the statistical tests.

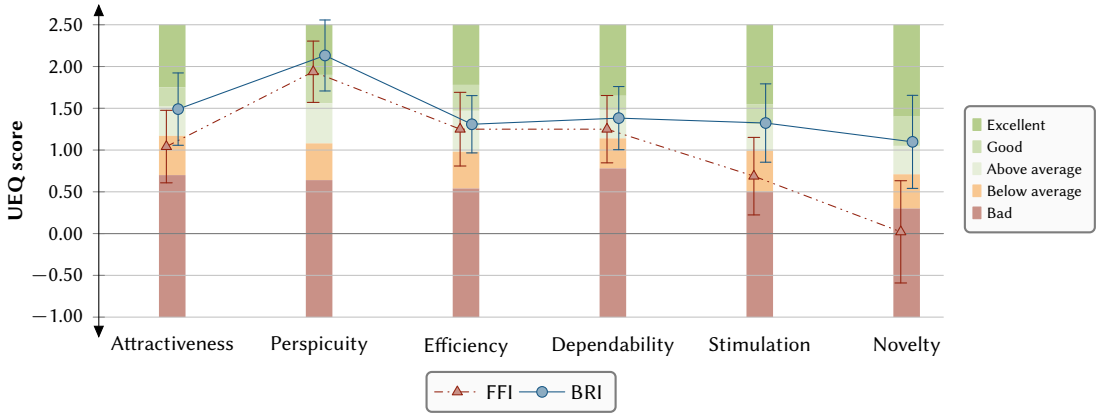

Figure 7.6 Comparison of the conditions with respect to the UEQ subscales, including 5 % confidence intervals. Benchmark values are from [SHT17].

Regarding the two more specific constructs, interface and interaction adequacy, we also noted a considerable effect of condition on the latter, while the mean values for the former were on the same level (see Table 7.2): The interfaces as such appeared highly adequate in both conditions, only with a small effect in favor of BRI. Given the between-subject design, this was not surprising as participants found and understood everything they needed for the task also in the standard interface. In contrast, the adequacy of the interaction possibilities provided in the blended recommending interface was rated as superior, with large effect size. This confirms H3. None of the results for BRI showed a difference with regard to domain knowledge.

■ **Sliders and visual clues** To examine the novel interface in the BRI condition in more detail, we confronted participants with additional questions. These questions were related to the perception of the sliders, which were presented for each selected value in the working area, and of the visual clues, which were shown to convey the sources of the recommendations. Apparently, participants found the sliders quite helpful (M = 3.41, SD = 1.23). Also, they liked the bubble visualizations that indicated how many items fulfilled the criteria (M=3.13, SD=1.41). With the help of an aided question, we tested whether participants understood the effects of the sliders: All participants chose the correct out of three predefined answers. With the help of an unaided question, we tested the same for the visualizations: 88 % explained the visual clues correctly in

their own words. The rest also seemed to have understood their meaning, but their explanations were not clear enough to draw a conclusion. Overall, these results additionally support H3.

■ **Perceived system effectiveness** The questionnaire results for user experience provided further, more general evidence of the potential of blended recommending. For the first construct, perceived system effectiveness, the mean value shown in Table 7.2 was however only slightly higher. Since effect size and t-test also indicated only a rather small difference, H4 is only partially supported. The results for domain knowledge were similar to those for the subjective system aspects mentioned at the beginning: Participants with poor domain knowledge found the system in the BRI condition less effective ($M = 3.27$, $SD = 0.65$) than those with more expertise ($M = 3.87$, $SD = 0.37$). A two-tailed t-test confirmed this effect ($t(12) = 2.24$, $p = .045$; $d = 1.25$).

■ **Perceived control** In contrast, with respect to the feeling of control, the results in the BRI condition were clearly superior to the results in the FFI condition (see Table 7.2). The statistical test confirmed that there was a large effect of condition. Thus, we can accept H5. This time, however, participants with *poor* domain knowledge provided slightly higher scores ($M = 4.47$, $SD = 0.70$) compared to those with more expertise ($M = 4.30$, $SD = 0.42$), even though this difference appeared negligible ($t(12) = -0.58$, $p = .574$; $d = -0.32$).

■ **Usage effort** Not only did participants in the FFI condition feel less in control, they also perceived the effort to be higher than participants in the BRI condition.[28] Yet, according to Table 7.2, there was only a small effect and the statistical comparison did not indicate a notable difference. Thus, H6 is also only partially supported. With respect to domain knowledge, we again observed the same tendencies as before: A two-tailed t-test ($t(12) = 3.25$, $p = .007$) suggested that participants with low expertise found the required effort less acceptable ($M = 4.00$, $SD = 0.71$) than those with high domain knowledge ($M = 4.89$, $SD = 0.33$), with large effect size ($d = 1.81$).

■ **Suitability for different usage scenarios** In addition to the specific constructs, we used three more general questionnaire items to assess the suitability of the two interfaces for different usage scenarios (see Table 7.2): As expected, we noted a preference for the conventional filtering interface *with* a search goal. In contrast, the blended recommending interface received considerably higher scores for scenarios *without* a search goal in mind. The distances between the mean values were similar for these two diametrically different scenarios, but the effects were in opposite directions. On the other hand, for the in-between scenario, i.e. with a *vague* search goal, we found no meaningful difference in mean values, which were equally high in both conditions.

Interaction analysis Because of the comparison of two complex interfaces, we complemented the questionnaire-based assessment by a more objective analysis of the interaction behavior of participants. In the following, we report the results obtained through recorded log data and screencasts. To explore the statistical relationships, we used again two-tailed t-tests.

First, regarding the *number of movies* participants had in their shopping cart at the end of the main task, we did not find a considerable difference ($t(29) = -0.02$, $p = .986$; $d = -0.01$): In the FFI condition, on average 7.21 items were in the cart ($SD = 6.02$), in the BRI condition, 7.18 items ($SD = 5.81$). Also, the *duration of the main task* was similar ($t(29) = 1.00$, $p = .327$; $d = 0.36$): Participants spent only slightly less time in the FFI ($M = 5.37$ min, $SD = 2.28$) compared to the BRI condition ($M = 6.18$ min, $SD = 2.25$). Putting these results in relation to each other, we found that

participants needed almost the same amount of *time to select a movie* in both conditions, namely 1.25 minutes, with $SD = 1.23$ in the FFI and $SD = 0.78$ in the BRI condition ($t(29) = -0.01$, $p = .992$; $d = 0.00$). Similar to the questionnaire results for usage effort, these time measurements do not support H6. In contrast to these findings for task II, there was a considerable difference with respect to the *duration of the training trial*: In task I, participants were much faster in the FFI ($M = 0.97$ min, $SD = 0.30$) than in the BRI condition ($M = 2.52$ min, $SD = 1.53$). A t-test confirmed this finding ($t(17.55) = -4.03$, $p < .001$; $d = -1.34$). In combination with the lack of differences in the task that followed, this however supports both H2 and H3, related to usability and interaction adequacy, as participants seemed to learn quickly how to use the richer interface.

In both conditions, participants selected on average roughly the same *overall number of filter criteria*, with $M = 9.92$ ($SD = 3.73$) in the FFI and $M = 8.21$ ($SD = 2.91$) in the BRI condition ($t(24) = -1.31$, $p = .204$; $d = -0.52$). Since facet values that were used multiple times counted towards this number, we laid our focus on the *number of active criteria* at the time participants put a movie into the shopping cart: Then, they had selected only $M = 2.22$ ($SD = 0.83$) criteria in the FFI, but $M = 4.21$ ($SD = 2.51$) criteria in the BRI condition. A statistical test also indicated a considerable difference ($t(24) = 2.61$, $p = .015$; $d = 1.03$). For the individual facets, in contrast, there were no considerable differences in relative numbers between conditions. However, the "items similar to" facet, which was only available in the blended recommending interface, was the second most frequently used facet in the corresponding condition, with 23 %. Also, values from this facet were more frequently selected than values from (almost) all facets in the control condition. The only exception was the genre facet, which was most frequently used in both conditions, with 43 % in the FFI and 57 % in the BRI condition. In the FFI condition, the second most frequently used filter criterion was the prominently placed input field for release year, though with only 17 %. Figure 7.7 shows the usage frequency for all facets.[36]

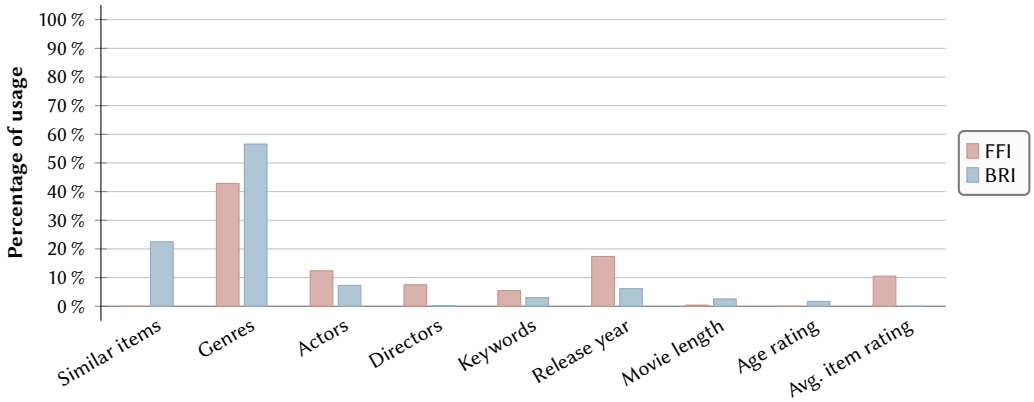

Figure 7.7 Comparison of the conditions with respect to the usage of different filter criteria.

By analyzing the screencasts captured for each session, we found that there were no effects over time in the BRI condition: Participants had selected the same number of filter criteria every time they chose a movie, and they kept using the same types of facets. Whereas individual interaction behavior was constant over time, we found differences between participants: 2 of them used on average less than two values when settling on an item, 2 others more than five. The remaining

[36] Filtering based on average item ratings was only possible in the FFI condition, via a drop-down menu in the table head (see Figure A.5 in Appendix A), but omitted in the other condition in favor of the "items similar to" facet.

majority of 71 %, however, placed two to five tiles in the working area.

For the standard interface, we observed that participants often reached combinations of values that caused empty result sets. Then, they tended to deselect arbitrary criteria, select others, and examine the effects of their settings on the results. As a consequence, one participant even explicitly expressed the need for Boolean OR operations, writing: "the possibility of ORing selections from the genre facet would have been quite useful". Participants in the BRI condition equally tried out several combinations of criteria (even with a higher number of values, see above). But, they were not able to over-constrain their search thanks to the underlying ranking approach. In addition, they even made extensive use of the sliders to adjust their preference settings.

Finally, we also examined the effects of *domain knowledge* in the experimental group: Participants with high expertise spent much less time to select a movie ($M = 0.87$ min, $SD = 0.43$) compared to those with low expertise ($M = 2.05$ min, $SD = 0.85$), which was confirmed by a t-test ($t(12) = -3.429$, $p = .005$; $d = -1.96$). At the same time, they added more movies to the shopping cart ($M = 7.89$, $SD = 3.92$) than others ($M = 3.60$, $SD = 1.52$), also confirmed by a statistical test ($t(12) = 2.32$, $p = .039$; $d = 1.29$). As expected, they most frequently used criteria from the genre or the "items similar to" facet. In contrast, as shown in Figure 7.8, participants with poor domain knowledge had selected criteria from a broader range of facets when they settled on a movie, and also a larger number, namely $M = 5.50$ ($SD = 3.16$) as opposed to $M = 3.49$ ($SD = 1.90$). This was again reflected statistically, at least to some extent ($t(12) = -1.51$, $p = .158$; $d = -0.84$).

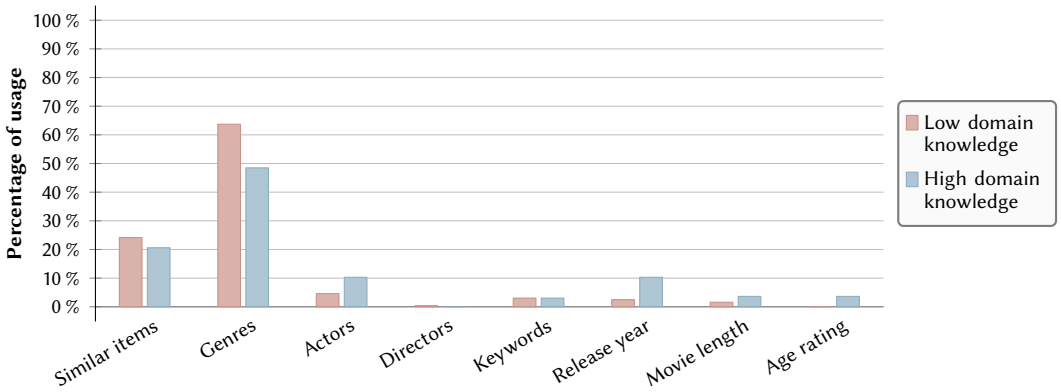

Figure 7.8 Comparison of the usage of different filter criteria in terms of domain knowledge.

Overall, participants with high domain knowledge also created more than twice ($M = 2.33$, $SD = 2.24$) as many new tiles as others ($M = 1.00$, $SD = 1.41$), i.e. they used the search function that was provided for facets with a large number of values more frequently ($t(12) = 1.20$, $p = .255$; $d = 0.66$). As expected, this was a result of the "items similar to" facet: The same comparison *without* considering values from this facet led to a much smaller difference ($t(12) = 0.16$, $p = .876$; $d = 0.09$), with the same mean value of 1.00 for participants with poor domain knowledge ($SD = 1.41$), but only with a mean value of 1.11 for those with more expertise ($SD = 1.17$).

7.3.4 Discussion

The comparison of the interface based on blended recommending with a conventional filtering interface showed that it is possible to provide users the flexibility of manual exploration in combination with the advantages of state-of-the-art recommendation methods—without having to

accept a loss of usability. We observed benefits in terms of subjective system aspects, but especially aspects related to user experience. In cases in which the benefits were smaller than expected, the results were still promising in both conditions. Accordingly, we found support for almost all our exploratory hypotheses,[19] throwing a positive light on our novel concept.

Quality The only, though important, exception was the hypothesis concerned with *perceived recommendation quality* (H1). Here, we did not find the positive effect that we had expected. Yet, the negative effect was small, and the level of quality high in both conditions. Concerning diversity, we obtained similar results. In this case, however, the rather average results suggested that none of the two interfaces was particularly useful—in line with the fact that we had no hypothesis in this regard. A possible explanation for the lack of differences in recommendation quality might be the nature of the baseline interface: Similar to the manual exploration interface in the study reported in Section 4.3, the movies chosen manually by means of faceted filtering were of course close to the preferences of participants. This may have equalized the effects of the recommendation methods in the blended recommending interface, in particular, since these methods did not personalize the results based on long-term preferences due to the experimental setting. Furthermore, we expected a positive effect of the "items similar to" facet. However, steering the recommendations into the direction of similar movies might in turn have had a negative impact on their diversity, which is known to be related to recommendation quality [Bol*10]. Overall, the results can still be considered promising, especially in light of the fact that these aspects only address the final outcome of the system, but stop short of taking into account the process of getting there. Nevertheless, further experiments with improved study design (more nuanced differences between conditions), but, in particular, with larger samples, are clearly necessary. Then, structural equation modeling could help to explore the influence of individual methods in more depth (e.g. collaborative filtering in the background of the "items similar to" facet), and thus, to improve their interplay.

Usability and interaction On the other hand, in the more general dimensions, in particular, those stronger related to user experience, we already saw the positive effects of blended recommending. First, with respect to system *usability*, it is yet worth mentioning that we would not have been surprised by a negative impact due to the richer functionality and the novelty of the interface. On the other hand, we assumed that the theoretical grounding and the user-centered design process would pay off. In accordance with these assumptions, the SUS results actually indicated that the usability was on the level of the more familiar faceted filtering interface (H2). However, the picture was even better for the more specific subscales of the UEQ: Here, we found positive effects, especially in terms of hedonic quality aspects. Apparently, conventional filtering was (no longer) perceived as stimulating, emphasizing the need for novel, more engaging approaches to search and filtering. Together with the smaller differences for pragmatic quality aspects, this showed again that participants were in principle able to fulfill their task in the control condition (in line with the results mentioned above), but could have benefited from a better user experience, as in the experimental condition. This was also reflected in the assessment of interface and interaction adequacy: Both interfaces appeared comprehensible and useful, and therefore received similarly high scores in terms of interface adequacy. But, when using the provided interaction possibilities, the positive effects of blended recommending came to light, shown by the results for *interaction adequacy*, but also for *perceived control* (H3, H5).

However, at the same time, the usefulness of the interfaces seemed to differ considerably depending on the situation: As expected, the conventional interface was rated as appropriate for targeted searches. For exploratory tasks with no or vague search goal, the interface based on blended recommending was rated as more suitable, i.e. for those usage scenarios for which it was intended. Apparently, participants appreciated that they were not required to specify precisely the desired properties of items and related entities. The interaction analysis underlined the positive impact of the tile-based representation of criteria, including the option to indicate preferences on an item level (reflected by the frequent usage of the "items similar to" facet), as well as of the possibility to weight the influence of the underlying methods on the final results.

Effectiveness and effort A related advantage was that participants did not have to deal with the hard Boolean filtering logic as known from faceted search. This may have contributed to the overall more positive assessment in terms of *perceived system effectiveness* and *usage effort* (H4, H6): The interaction analysis showed that participants in the experimental condition equally tried out several combinations of criteria. But, they did not need to observe the implications of the queries they indirectly formulated in this way in order to avoid empty result sets (some participants actually selected combinations although it was immediately clear that not all the criteria could be satisfied). The fact that they had much more criteria selected at the time they put items into the shopping cart also suggested that they appreciated the wide range of options to specify their preferences. On the other hand, this might also be an explanation for the rather small advantage we found with respect to perceived effort. However, the *overall* number of filter criteria they used was quite similar. The exploratory screencast analysis showed that participants changed or reset the criteria much more often in the baseline condition (to prevent mutual exclusions), which counted towards this number. In the other condition, participants usually kept the criteria in the working area, so that once selected, they contributed more "productively" towards the final results. Of course, the sliders may have biased this analysis as participants were able to manipulate the results even without changing the selection of criteria.

The longer duration of the training trial in the experimental condition suggested that the novel interface required a learning phase. Given that this difference vanished in the main task, this phase however seemed to be rather short and already over when they started this task. Remarkably, this was the case despite the larger functionality. On the other hand, the issue that participants often over-constrained their search with the baseline interface, forcing them to backtrack and change their filter settings, might have compensated for the time they gained in the corresponding condition due to the simpler and more familiar interaction. Tedious backtracking could of course have been avoided, for instance, by query previews [Hea09] or dynamic taxonomies [ST09]. But, due to the underlying logic that is typical for information filtering systems, also these features cannot eliminate the necessity of testing different combinations of filter criteria. Exactly this, however, turned out inherently easier in our blended recommending interface.

Nonetheless, we initially expected greater benefits in terms of effectiveness and effort, both subjectively and objectively. Contrary to our exploratory hypotheses, the simpler and more familiar interaction in the control condition probably balanced out the advantages of blended recommending more than we had expected. Beyond that, also the blended recommending interface had some issues. Participants indicated, for example, that "specifying weights for the individual criteria was more cumbersome than just deciding on an ordering". Also, they stated that "knock-out criteria would have been helpful", and that they wanted to "specify which kind of movie they

do not want to see". These are not only interesting topics for future research (the question of what should *not* be recommended has hardly been investigated so far), but also aspects that may have confounded the current results for effectiveness and efficiency.

Personal characteristics In light of the rich interaction possibilities, we studied the influence of domain knowledge in more detail. Apparently, participants with more expertise specified their preferences more precisely: With fewer criteria, but more self-created values, they obtained recommendations they perceived to be of higher quality than participants with less domain knowledge. The interaction analysis indicated that these participants, in turn, relied on facet values that were visible right from the outset, and needed a larger number. We observed similar differences in the subjective assessment. These results, however, might have been confounded by participants who were not able to adequately assess the recommended items because they did not know them (in other work, we have shown that item consumption is a decisive factor in user studies for the approximation of the actual value of recommendations [Loe*18]). Beyond that, the results need to be taken with a grain of salt because of the large number of statistical tests, and because of the overall small sample size, and thus the limited number of participants in the experimental condition. Nonetheless, it became clear that there is room left for improving specifically the experience of less knowledgeable users. On the other hand, the blended recommending interface was, in fact, targeted for users with high domain knowledge, and power users in general. Still, the support for users with poor domain knowledge was already better than in the conventional interface. Nevertheless, further investigation of the influence of personal characteristics is clearly necessary, which is in line with other calls to take individual needs more serious when designing recommender systems [cf. KRW11; KWB14; JTV18; Car*19].

Summary Despite its exploratory nature, the experiment allows to conclude that the richer interaction possibilities of the novel interface do not introduce any disadvantages. In contrast, users are supported by the permanent availability of a ranked list of recommendations that matches the selected criteria as well as possible. Thereby, since control of the combination of the methods that are responsible for this list is put into their hands, users can always use the system in accordance with the complexity of their situation and their individual expertise: relying on collaborative filtering recommendations if the search goal is vague, using content-based techniques if desired item properties are known. Nevertheless, it remains an open question which methods are preferred exactly in which situation. Therefore, not only confirming the findings of the current study, but also gaining deeper insights, are important subjects of future work. Moreover, whereas the more holistic support for adjusting the system's final outcome became visible in the subjective assessment of aspects such as interaction adequacy, and, in particular, in dimensions related to user experience, a more thorough investigation is necessary with respect to the transparency of the process: While there was immediate feedback whenever participants changed their preference settings, including visual clues, it is still unclear whether the comprehensibility that is typical for information filtering methods was preserved by using faceted filtering as a point of departure. In summary, however, there are already enough indications that *merging model-based collaborative filtering* with other recommendation and information filtering methods is another promising means to improve *user control and experience*, and blended recommending thus an important step to eventually get to our overarching goal as close as possible (RQ3).

CHAPTER 8

> "True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information."
>
> — *Winston Churchill, British politician*

Integrated platform for interactive model-based collaborative filtering

In this chapter, we bring together all the approaches to interactive recommending proposed in this thesis. Each of the underlying methods aims at turning model-based collaborative filtering systems into fully user-controlled applications. In a series of user experiments, we have explored the potential of these methods for filling the gaps indicated in Section 3.1. Now, we present an *integrated recommendation platform* [LZ19b][37] that implements these methods in the form of a set of seamlessly connected perspectives. This enables us to showcase that the ideas behind our three research questions described in Section 3.2 ultimately all contribute to our main goal. For this, we first present an *overview* of the system and the different perspectives. Afterwards, we describe a set of illustrative *case studies* to demonstrate that our methods can effectively be integrated with each other, and that users can thus be provided at all stages of the recommendation process with interaction mechanisms that support them in reaching typical search goals.

8.1 Overview

In the following, we describe *implementation details* and provide an overview of the *perspectives* that we used for a holistic integration of our interactive methods.

8.1.1 Implementation details

For implementing the recommendation platform, we set up a *JavaEE* web application with a user interface based on the *JavaServer Faces* standard,[38] using the *PrimeFaces* component library.[39] We used the *Apache Mahout* recommender library[20] in combination with our *TagMF* framework as presented in Section 5.3. In fact, the platform constitutes a major extension of the *demo* package that is part of this framework. Due to the same arguments as presented in this context, we

[37] This publication was accepted for the demo track of the *13th ACM Conference on Recommender Systems* (`https://recsys.acm.org/recsys19/`), held 2019 in Copenhagen, Denmark, where attendees could try out an earlier version of this interactive recommender system. Later, the same version was also exhibited at the *GI Human-Computer Interaction Symposium: AI for Humans* (`https://fb-mci.gi.de/veranstaltung/symposium-mensch-computer-interaktion-ki-fuer-den-menschen/`), held 2019 in Berlin, Germany.

[38] `https://javaee.github.io/javaserverfaces-spec/`

[39] `https://www.primefaces.org/`

assumed there would be no loss of generality when making use of background data from the domain of movies. Consequently, we chose the same dataset as in most of the experiments with our content-boosted matrix factorization method, i.e. based on the *MovieLens 20M* dataset[23] and the *MovieLens Tag Genome* dataset[24] (see Section 5.4.1 and 6.3.2.2). In addition, we used an updated version of the metadata dataset that we initially created for the second user experiment: We again gathered data from *The Movie Database* (TMDb)[29] and the *Open Movie Database* (OMDb)[30]. The resulting dataset contained metadata for each movie in the *MovieLens 20M* dataset, in particular, titles in different languages, links to posters, images and trailers, plot descriptions and genres, as well as lists of directors, cast members and keywords (see again Section 6.3.2.2).

This provided us a basis to implement all methods for improving user control and experience, which we introduced separately in the previous chapters, in a single system. Based on a theoretical analysis of possible connections between these methods and expected user behavior if all the related interaction mechanisms were available at the same time, we used multiple perspectives for this purpose. We integrated these perspectives as outlined in Figure 8.1: All users ■ *start* from an initial view. From there, they can navigate to a typical ■ *item list* and proceed to related ■ *item details* (and vice versa), or inspect their ■ *user profile*, if existing. At cold start, users can try one of the novel ■ *preference elicitation* methods. Subsequently, they are provided with ■ *recommendations*. Everything else takes place in the same view (or in similar perspectives, not shown in Figure 8.1 for the sake of simplicity). This includes the usage of standard recommendation functionalities (providing ratings, changing or revoking them), of the remaining features based on content boosting (weighting or critiquing), and of a blended recommending interface.

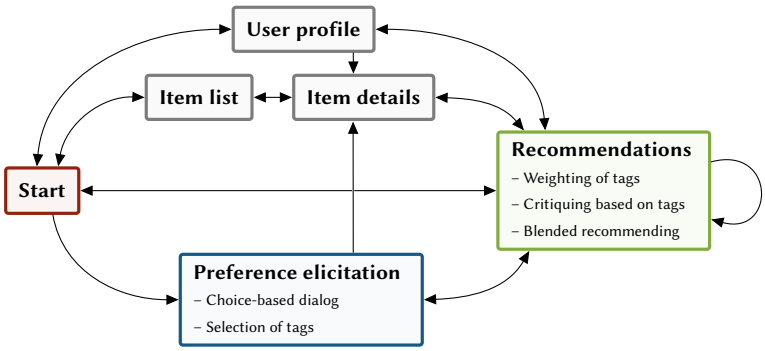

Figure 8.1 Overview of the perspectives we used for implementing both the interactive methods and the other functionalities of our platform.

For ensuring a seamless experience, other connections in Figure 8.1 are of practical importance as well: First, users can ask for recommendations immediately upon entering the system. Then, a non-personalized variant of this perspective is delivered with results based on item popularity. Moreover, from the recommendations perspective, users can return to the choice-based dialog for an adaptation of the current result set, or to the selection of tags for requesting a new one. Finally, it is possible to access the item database and corresponding detail pages from all other perspectives. In turn, users can jump (back) from item detail pages into the recommendation process, either by asking for similar items or by applying critiques.

When we implemented the perspectives and their connections, we followed typical user-oriented design guidelines for recommender systems, as suggested by Pu, Faltings, Chen, Zhang, and

Viappiani [Pu*10], Cremonesi, Elahi, and Garzotto [CEG17], and Alvarado Rodriguez, Vanden Abeele, Geerts, and Verbert [Alv*19]. Moreover, we took into account the qualitative feedback gathered in the user experiments as well as the quantitative results we obtained with respect to the usability of the corresponding prototype systems.

8.1.2 Perspectives

Next, we detail on perspectives that most directly illustrate how we implemented the different methods in our platform. We highlight noteworthy *connections between these perspectives* as well as typical instances of the implementation of *guidelines and usability-related findings* as described above. Note that the system contains further perspectives, for the standard functionalities mentioned in the previous section (see Appendix A for screenshots), but also for development and experimentation purposes as well as alternative component arrangements.

Choice-based preference elicitation As a first example, the perspective shown in Figure 8.2 implements the novel preference elicitation method described in Section 4.2. This time, the *choice-based dialog* is set up on top of a content-boosted model learned by means of our *TagMF* framework instead of a standard matrix factorization model as in the user experiment reported in Section 4.3. At the front-end, this does not make any visible difference: The dialog shows a series of binary choices between sets of four items, where each step represents a single factor. In the example, the set on the left-hand side contains low-brow action movies situated in the present (a), the set on the right-hand side much more serious, dark sci-fi movies (b). Movies are presented as in the prototype from the user experiment. Further details are available upon request. The main difference is that the tag clouds underneath each set are omitted to ensure that users ■ *focus on the examples*, even though the tag-based information is still indirectly present thanks to content boosting.[40] Buttons allow users to indicate which set is preferred, or to use the no-choice option in case they cannot decide or do not know enough about the items (c). As an improvement to the earlier implementation, all ■ *comparisons are counterbalanced*, mitigating negative effects for users who tend to choose sets only because of their position. Moreover, each set now comprises a randomized selection of items from the larger set of candidate representatives. This improves the experience when using the dialog again. In addition, users can go backwards to ■ *revise preferences*. This may be necessary when they want to reconsider a decision in light of a succeeding step or of the final results, but is also useful for exploring alternatives. A ■ *progress indicator* informs users about the number of remaining steps, avoiding that they lose interest too early (d). Finally, as soon as they reach the recommendations perspective, this does not mark the end of the process anymore. Instead, there are several ■ *options to continue*, based on the close connection to the other mechanisms: ■ *weighting tags* or ■ *applying critiques* is not only possible for recommendations based on an existing user profile (as proposed in Section 6.2.2 and 6.2.3), but also based on the latent factor vector established during the choice process.

Indicating preferences at cold start via tags As an alternative, the perspective shown in Figure 8.3 implements the elicitation of initial preferences with the help of a *selection of tags* as described in Section 6.2.1: Based on our content-boosted matrix factorization method, all users without a profile can use this interaction mechanism, which leads in the background to a new

[40] Informal interviews with test users who interacted with variants without and with content boosting confirmed that both consistency within and diversity between the sets benefit from the additional tag-based information.

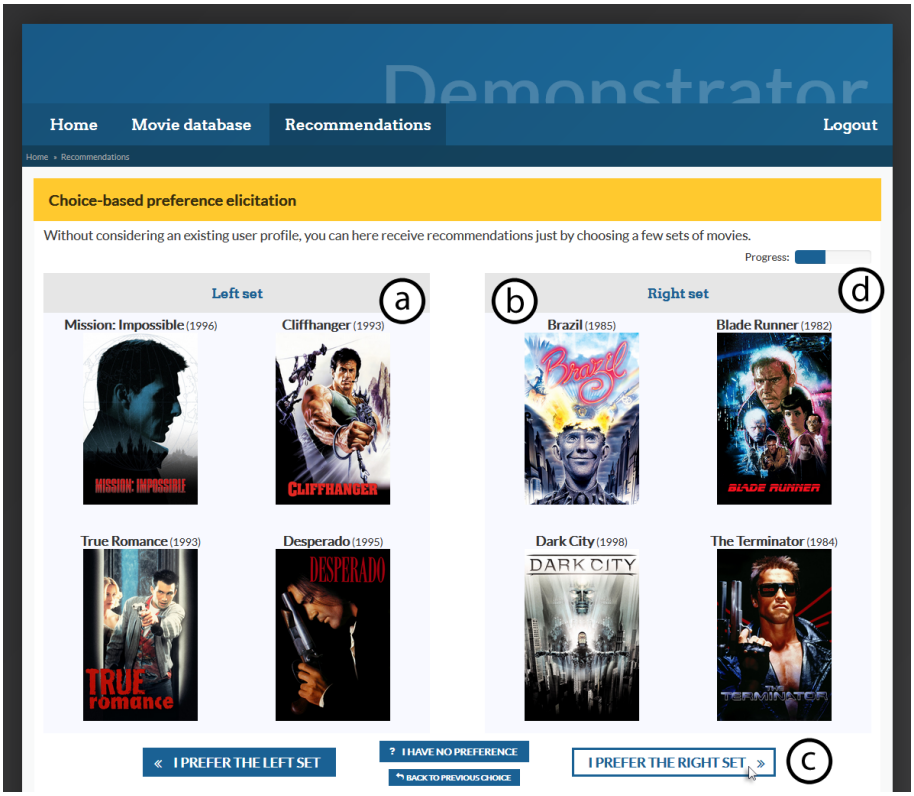

Figure 8.2 Screenshot of the perspective that implements the choice-based preference elicitation dialog: The user can choose between the left (a) and the right (b) set of movies, which represent a certain factor of the underlying model, using the buttons below (c). A progress bar (d) indicates the number of steps he or she needs to complete before the corresponding recommendations are finally shown.

user-tag vector being created. In the example, the tags "action", "drugs", and "comedy" have been selected by using the input field at the top (a). Enhanced with ■ *autocompletion*, this element allows searching within all tags considered by the system. The recommendations shown below fit to the preferences expressed in this way, comprising movies such as "Hot Fuzz" or "Rush Hour" (b). To continue, users can either try alternative tags or ■ *proceed to another perspective*: Again, it is possible to ■ *adjust the result set* by assigning weights to the tags or to ■ *apply tag-based critiques* to one of the contained items. Under exploitation of the content-related associations that are established by our extended matrix factorization method, these options let users adjust their user-factor vector, i.e. here the substitute vector created from the new user-tag vector. In addition, users can ■ *re-rank or filter the results* based on criteria that cannot be taken into account solely based on collaborative filtering, but with the help of ■ *blended recommending*. Next, we continue with three perspectives that implement exactly these interactive features.

Adjusting recommendations via tags The perspective shown in Figure 8.4 allows the *weighting of tags* as described in Section 6.2.2: Based on our content-boosted matrix factorization method, tags can be placed in the area at the top to take into account situational needs (a). There, users can manipulate their weights using the attached sliders. In contrast to the aforementioned perspectives for cold start, this allows for an adjustment of recommendations that are generated

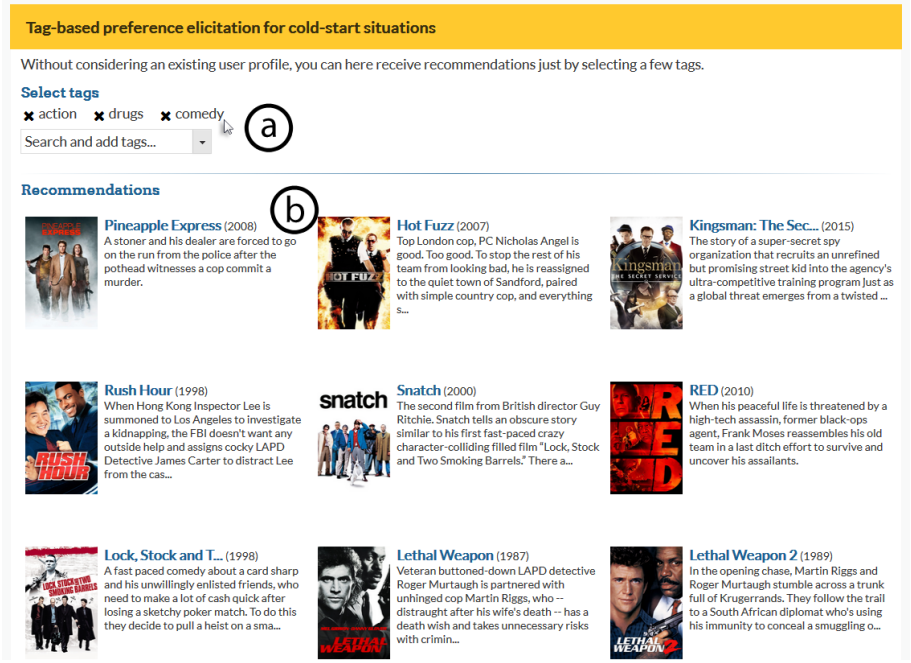

Figure 8.3 Screenshot of the perspective that allows to indicate initial preferences via tags: The user can select tags using the search functionality (a), leading to the recommendations that are shown below (b).

based on an existing rating-based profile, but also, on the input collected by one of the novel ■ *preference elicitation methods*. In the example, the tags "disturbing" and "violence" already have been selected. Maximum weight has been assigned to the former, whereas the weight of the latter is currently under change. Each interaction is immediately reflected back into the weighting vector, which gets added to the user-tag vector that is responsible together with the latent knowledge for the recommendations. An input field allows to manually search for tags, supported by ■ *autocompletion* (b). In addition, ■ *attribute suggestions* are presented to facilitate both starting and continuing the search process (c): Initially, the most popular tags are shown, i.e. tags assigned most often by other users. This gives the current user an impression of what is generally of interest. However, as soon as he or she has selected tags and applied weights, tags are shown that are most similar to the tags already in use (in terms of their relevance for the items). This helps ■ *refine the result set*, which is shown below (d). The most relevant tags according to the respective item-tag vector are shown alongside each recommendation. These tags may be selected as well, providing another ■ *option for refinement*, but directly from within the result set, based on tags just identified to be of interest. As already outlined above, ■ *immediate feedback* ensures that the effects of these preference settings are made clear.

Critiquing specific items via tags Based on the extended matrix factorization method, it is also possible to express feedback in a more discrete fashion. Reachable from ■ *any location* in our platform where items are shown, the perspective shown in Figure 8.5 allows *critiquing based on tags*: In the top-left corner, the currently critiqued movie (recommended or manually selected) is presented (a). The other side of the screen contains the critiquing area, comprising tags for the application of critiques (b). Each tag is accompanied by radio buttons that allow to request a new set of recommendations with items that are more or less strongly related to this tag. In

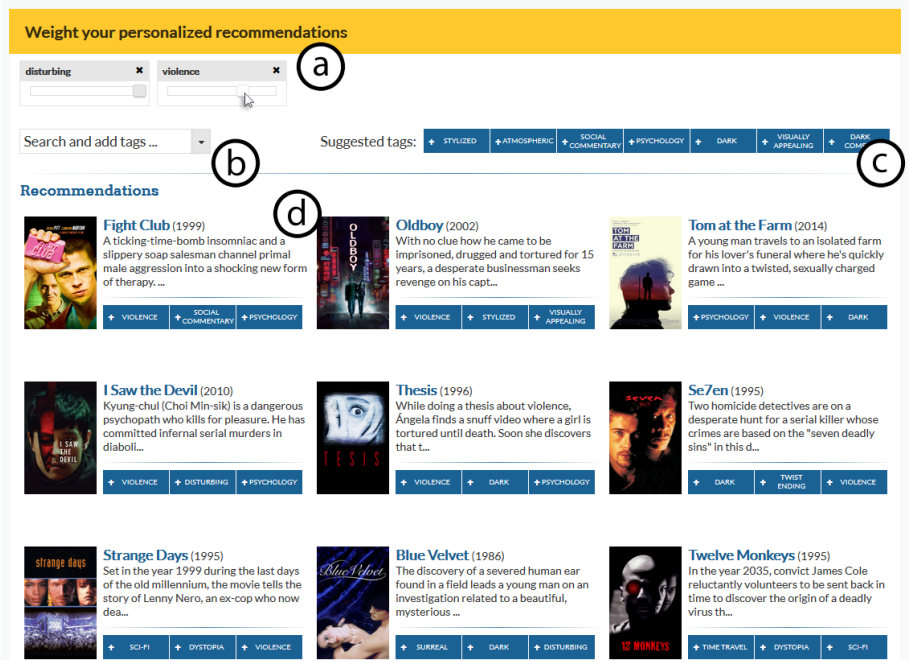

Figure 8.4 Screenshot of the perspective that allows to adjust recommendations via tags: The tags selected by the user are shown at the top (a), where he or she can set their weights. Tags may be added by searching manually (b) or following the suggestions (c). Item recommendations appear below (d), representing the user's long-term preferences, weighted according to his or her situational needs.

the example, it has already been indicated that the results should contain "less sci-fi" than "The Matrix", whereas the request for "more disturbing" content is about to be issued. As in *MovieTuner*, tags are automatically suggested as critique dimensions by the system. However, given that our method determines user-tag vectors for all users, not only the relevance for the respective item is taken into account for these suggestions, but also the relevance for the current user. Following the description in Section 6.2.3 with the same parameterization as in the corresponding user experiment reported in Section 6.3.2, this introduces more ■ *personalization* to the critiquing process. Nevertheless, ■ *user-initiated critiques* may be applied as well, using the input field with ■ *autocompletion* at the bottom of the critiquing area (c).

The rest of the screen contains the recommended items (d). In contrast to common practice in example critiquing, not only the respective item and the applied critiques are responsible for these results. But, the general preferences of the current user are additionally taken into account, i.e. his or her long-term profile derived from item feedback he or she has provided while interacting with the platform. Next to each movie, the most relevant tags are shown. Upon selection, they are added as critique dimensions for ■ *further refinement*. The "critique this movie" button allows to start a new cycle in the critiquing process, setting the respective movie as the new item to critique. As an addition to the prototype from the user experiment, a slider below the set of recommendations (not visible in the screenshot) allows to ■ *vary the extent* to which the user's original user-tag vector is considered, i.e. to adjust the corresponding parameter described in Section 6.2.3: A small value means that only content data play a role (as in *MovieTuner*). A large value means that only the user's interests contribute to the vector that eventually serves

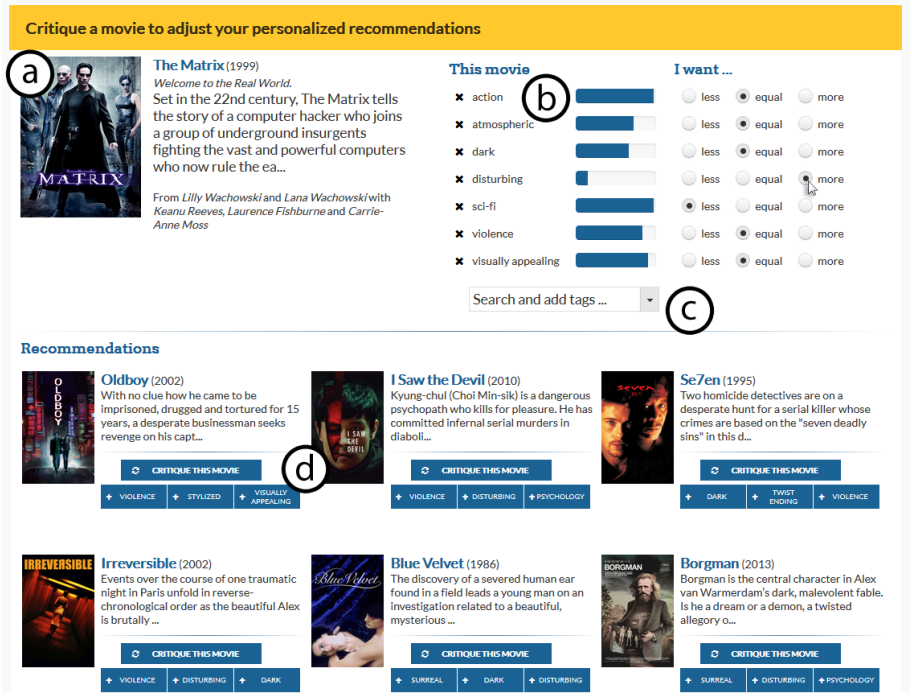

Figure 8.5 Screenshot of the perspective that allows critiquing via tags: The user can apply critiques to the current movie (a), using the tags shown as critique dimensions (b). In case these dimensions are not sufficient, he or she may add further tags (c). The recommended movies below, accompanied by buttons for starting a new critiquing cycle, match the critiques, but also the user's long-term preferences (d).

for generating the recommendations, but none of the item's characteristics. This way, users can indicate how important they find their long-term preferences in the current situation. Given the ■ *immediate feedback* provided by the system, they can also examine the influence of the vector representation of these preferences on the results.

Blended recommending First proposed in a different stream of research, the *blended recommending* concept we described in Section 7.2 finally allows to merge the results of content-boosted matrix factorization everywhere in our platform with those of other recommendation methods. Yet, whereas ■ *any perspective* may be implemented according to this concept, Figure 8.6 shows a perspective dedicated specifically to showcasing its implementation, adopting the suggested interface design as closely as possible (cf. Figure 7.2): Facets are presented as widgets on the left-hand side of the screen (a). Initially, all facet are collapsed. As soon as a facet is expanded, a number of rectangular tiles with images or pictograms is displayed. These tiles represent the corresponding facet values from which users may choose the criteria they prefer and want to have considered in the results (b). For facets with a large number of values (similar items, directors, actors, keywords), only the most popular entities are shown at the beginning. This ensures that users do not become overwhelmed, but can ■ *focus on generally known entities*. A search function (not visible in the screenshot) is provided to look for other values. Once selected, they are displayed as tiles as well. Beyond that, users can ■ *circle through all values* of a facet (via a button that is also not visible in the screenshot). In contrast to the original mechanism

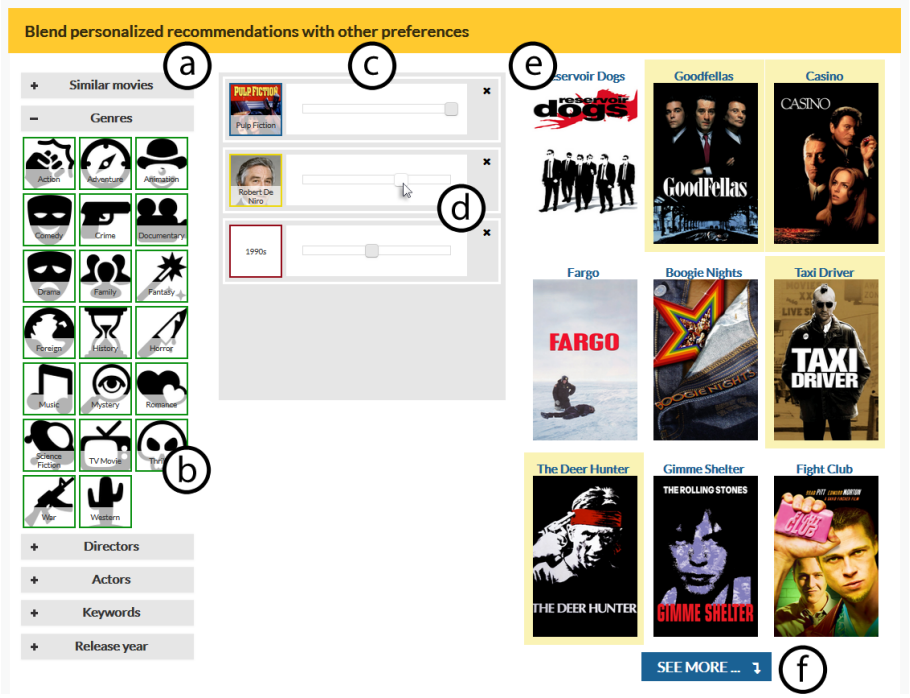

Figure 8.6 Screenshot of the perspective implemented according to the blended recommending concept: The user can select from a set of facets (a) the values he or she wants to see considered (b). Then, these values are shown in the working area (c) for adjusting their weights (d). Recommendations appear in a ranked list (e), in which movies that correspond to the currently hovered or weighted criterion are highlighted. This list may be extended by clicking the button below (f).

that suggested further values in the prototype for the user experiment reported in Section 7.3, this avoids that users constrain their results too much on what already has been recommended.

The interaction via ■ *drag and drop* is generally the same as in the earlier prototype: Users can drag tiles from the facet widgets and drop them into the working area (c). In the example, it has been indicated that recommended movies should be similar to "Pulp Fiction", with Robert de Niro, and from the 1990s. Weights can be manipulated by means of the associated sliders, allowing to vary the influence of the underlying methods (d). Right next to the working area, the resulting list of recommendations is shown, based on the weighted overall relevance scores calculated for each item (e). Adding and removing criteria as well as changing their weights updates this list in realtime, so that users receive ■ *instantaneous feedback* when they adjust their preference settings. The list, here showing movies such as "Reservoir Dogs" (highly similar to "Pulp Fiction") and "Goodfellas" (a 1990 movie with Robert de Niro), may be extended at any time by clicking the "see more" button below (f). However, in contrast to the description in Section 7.2 and the user experiment, not only explicit input provided during the current session is used for generating the recommendations. Instead, an adapted recommendation function rescores the results according to the predictions of the content-boosted matrix factorization method. This introduces ■ *personalization* based on the user's long-term profile in addition to ad hoc preferences. Since popularity is used as a fallback for coming up with the ranking, the system's functionality is not limited if the current user has not yet provided any ratings in our platform (or only very few).

The "items similar to" facet benefits as well, as recommended movies can be ■ *set much easier into relation* to values selected from this facet (i.e. to other movies): The tags considered as side information convey the meaning in the model dimensions more effectively, which consequently also affects the similarity calculation. The remaining differences to the earlier prototype are mostly related to ■ *general usability*: Certain facets (e.g. age rating and movie length) or features (e.g. custom time spans for the release year facet or drag and drop from the result set) are removed because of their infrequent usage, or are handled differently because they caused confusion.

In addition to these interactive features, mechanisms to ■ *support user comprehension* are available as well. By clicking on a recommended movie, users can proceed to the ■ *item detail perspective* or the ■ *critiquing perspective*. However, they are also presented with an explanation that shows which criteria could be fulfilled, and thus, why the item appears in the result set. The other way around, when hovering a criterion in the working area or changing its weight, the respective items in the result set are highlighted in the same color as the criterion (movies starring Robert de Niro in Figure 8.6). Although we left out other visual clues that were present in the earlier prototype for the sake of simplicity (e.g. bubble visualizations visible in Figure A.6 in Appendix A), the perspective in this way helps better ■ *understand the sources* of the recommendations. This is particularly useful if the hybrid combination gets more complex with more criteria, always providing users with hints for ■ *preference refinement*.

8.2 Case studies

As part of our final contribution, the goal of the descriptive case studies we present in this section is to provide insights into the general effectiveness of our platform, and, in particular, the full potential of the individual methods when they are holistically integrated in a single system as proposed in the previous section. For this, we take up the model of user interaction from Chapter 3 (see Figure 3.1). There, we used this model to take a theoretically informed look on the possibilities to support users in each phase of the recommendation process. In the succeeding chapters, we addressed the resulting research questions and were able to show that the underlying ideas bring us closer to interactive recommender systems that fully adhere to this model. Figure 8.7 shows an updated version, in which the corresponding improvements are highlighted accordingly (by bold colored lines): In addition to standard user-item feedback, users are enabled to indicate their preferences in systems based on latent factor models with the help of a) *item comparisons* (cf. Chapter 4). The application of our extended matrix factorization method (cf. Chapter 5) allows for more advanced interaction mechanisms based on b) *item-related information* (cf. Chapter 6). Building on our concept of blended recommending, the output of the potentially hybrid configuration of these systems may furthermore be manipulated by c) *selecting and weighting* the corresponding methods in a more interactive fashion (cf. Chapter 7).

The case studies are designed to illustrate these improvements. For this, we refer to the phases of the recommendation process in the same way as in Chapter 3. For each phase, we describe possible user behavior in relation to typical search goals, using a number of sample users, for whom we created rating-based profiles in accordance with the user-item matrix shown in Table 2.1 in Section 2.1. We highlight, how each method may contribute to achieving these goals, using the same colors as above for the formerly incomplete connections in our model. Also, we provide references to our research questions, and explain how addressing them has strengthened these

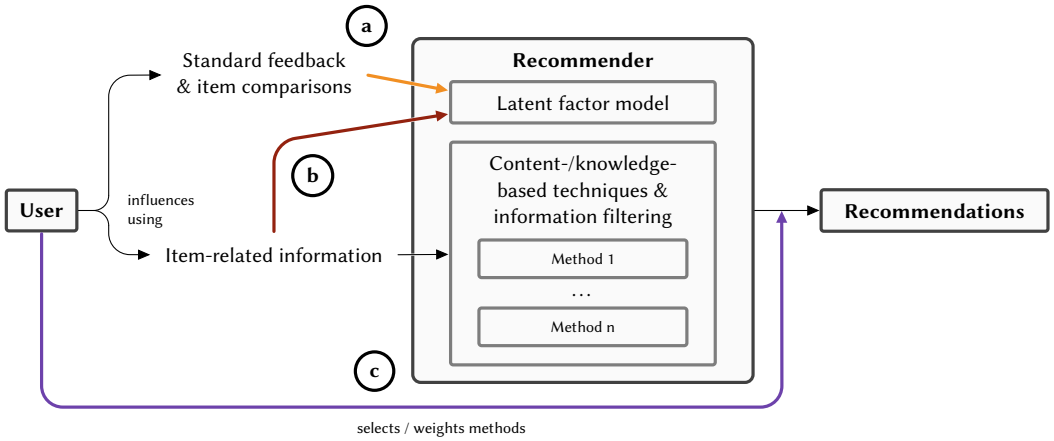

Figure 8.7 Updated model of user interaction with systems that use model-based collaborative filtering, highlighting the improvements we were able to achieve by means of our methods. See the text for a detailed description of how the respective connections could be strengthened (a, b and c), and how this is illustrated by the case studies.

connections. Note that there are many other cases, and also other ways to reach the given goals. Still, we are confident that it is possible to draw general conclusions from our case studies.

8.2.1 Elicitation of initial preferences

First, we look at cold-start situations. Users may have a concrete search goal or not. This means, when starting to use the system, they need to *initially express their preferences* before they can effectively be provided with recommendations, but either know or do not know for what they are looking. To illustrate these cases, we consider the two following example users:

Greta wants to enjoy an evening in front of the TV. She has some general knowledge about movies, but no idea which light movie might cheer her up. Thus, she enters the platform to receive suggestions without having a concrete goal.

Hendrik almost certainly knows what he is looking for: He wants to follow a suggestion by a friend who told him about a movie of which he forgot the title, but remembered that it stars Brad Pitt and is similar to "Donnie Darko".

Both users are enabled to articulate their needs in an adequate manner, regardless of their different background: Whereas Greta in a real-world system would only have the possibility to rate movies, our integrated platform allows her to use, for example, the ■ *choice-based dialog*. There, the choice process gives her a first impression of the available options, including factor representatives that would already be appropriate for her: from musical fantasy films over Monty Python comedies to animated movies. Eventually, however, she receives a set of feel-good movies depending on all her choices, i.e. based on the position within the factor space to which she is assigned only by a more extensive *exploitation of the semantics* contained in the latent dimensions (RQ1). Yet, she may find the choice process too difficult, for example, because she does not know enough representative movies (such as in case of a comparison as shown in Figure 8.2). Then, indicating preferences by ■ *selecting a small number of tags* may be a meaningful alternative (as illustrated in Figure 8.3): Few tags such as "comedy", "funny", and "music" already lead to recommendations of movies such as "Hangover", "Blues Brothers", or "Shrek". Accordingly,

leveraging item-related information in addition to standard collaborative filtering data represents another possibility to fulfill Greta's needs without requiring her to rate items (RQ2).

Users with a specific goal, but likewise without an existing preference profile, can skip these mechanisms. Thus, Hendrik may proceed immediately to the list view that provides an overview of the item database: Searching for "Donnie Darko" leads him to this movie's detail page, from where he requests similar movies (in terms of latent factor vectors, see the screenshot in Figure A.8 in Appendix A). Then, he starts ■ *blending in content-based recommendations* by selecting Brad Pitt as a facet value and giving this criterion a high weight (similar to Figure 8.6). Thanks to the *merging of model-based collaborative filtering with other methods*, "Twelve Monkeys" is thus brought up as a recommendation, which is likely the movie he was told about (RQ3). Up to this point, Hendrik's behavior represents a simple look-up task [cf. Mar06]. But, users often switch from browsing to searching (and vice versa) as long as their information need evolves [Dir12], so that known-item search can always turn into an exploratory task [LRS06; Mar06]. Consequently, Hendrik might also continue browsing through the recommendations instead of settling on "Twelve Monkeys", or request other movies with Brad Pitt that reflect his own interests more strongly: Selecting genres such as action or romance and playing with the weights (which resembles task I in the user experiment on blended recommending, cf. Section 7.3.2) may lead him to movies such as "Mr. & Mrs. Smith" or "The Curious Case of Benjamin Button".

8.2.2 Control over the systems

Another typical situation is that users have a long-term profile. Personalized recommendations can then be presented immediately, but more direct *control over the systems* is required to satisfy situational needs, with the option to provide feedback in a more expressive manner than on the level of items. Let us consider, for example, the two following users, who already rated some items and consequently have a representation within the content-boosted model of our platform:

Amalia is interested in comedies and romance movies (see Table 2.1). However, she frequently watches movies with friends or relatives. Thus, she likes to take into account not only her personal interests when she enters the platform, but also those of others.

Emily has yet provided rather average ratings. The only exception is "Braveheart" because of her passion for British history (see again Table 2.1). She wants to explore whether the platform can recommend something similar, but more suitable for a girls' night.

These cases are partly inspired by task IIa of our second user experiment on content-boosted matrix factorization, in which participants were also asked to take into account the preferences of other persons (see Section 6.3.2.2). As in this task, Amalia might use the ■ *critiquing mechanism* to find movies in accordance with both long-term preferences and short-term goals. For this kind of interaction, it is particularly useful that tags in the language of users may be *leveraged as additional item-related information*, and consequently allow to intervene in the underlying model (RQ2). However, this mechanism requires that users know a movie to start with, which is more related to the second case. Therefore, Amalia more likely goes straight to the recommendations perspective every time she enters the platform. Figure 8.8 shows the preference profile (a) that is responsible for the results initially presented to her in this view (b). Likewise based on specific *item-related information* (RQ2), she adjusts these results by ■ *selecting and weighting tags* that seem appropriate for the current situation, either to obtain suggestions for an "action" movie to watch with her boyfriend (c), or a "classic" movie "based on a book" to watch with her mom (d).

Consequently, "Pirates of the Caribbean" or "The Wizard of Oz", respectively, are recommended, i.e. movies in line with her own taste, but accommodating the preferences of her company. None of her actions in these recurring situations affects her rating-based profile, so that these sessions have no effect on the personalization of the results when she comes back another time.

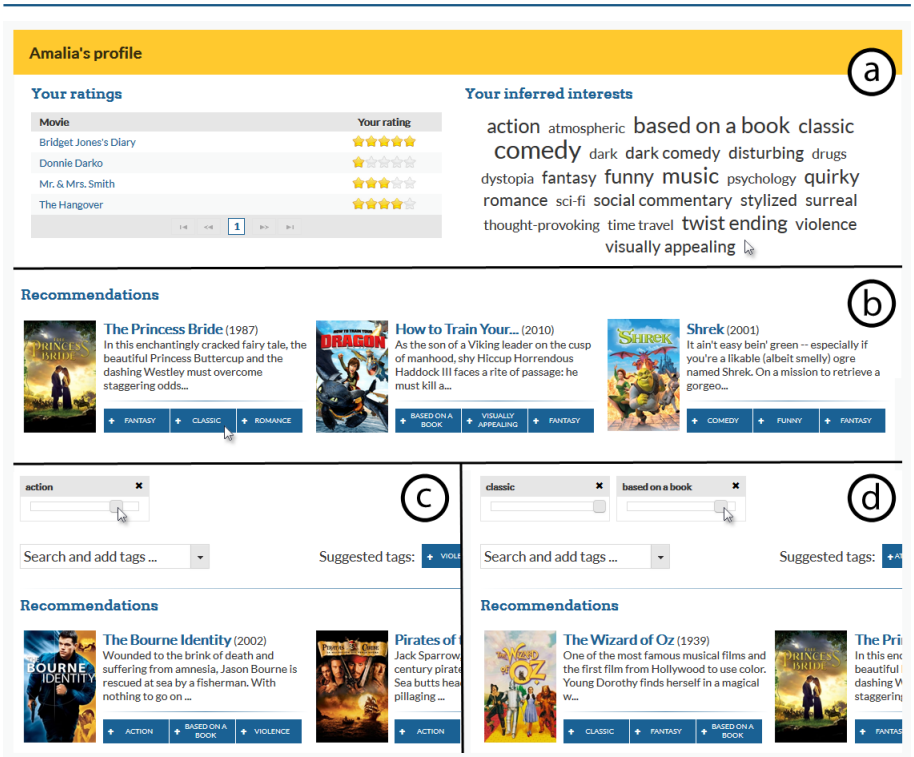

Figure 8.8 Screenshots for one of our case studies: At the top, next to a corresponding tag cloud visualization, the ratings provided by the user are shown (a). The recommendations are displayed below (b), which are then adjusted by weighting different tags (c and d).

As outlined above, the ■ *critiquing mechanism* may be more appropriate for users who know where to start the recommendation process. For example, Emily selects "Braveheart", one of her favorite movies, after visiting her profile page to review which movies she has already rated. Then, she starts applying critiques to accommodate the preferences of her friends (similar to Figure 8.5): She indicates that she wants to focus less on the "historical" aspect, but more on "romance". The British drama and love film "The Young Victoria" is consequently recommended—still similar, but representing the tags chosen as critique dimensions less or more strongly, respectively. At first sight, this resembles a typical subject search [cf. LRS06]. However, due to the *integration of content information* in addition to regular collaborative filtering data (RQ2), she does not need to explicitly indicate her general interests as would be necessary in systems implemented independent of personalized recommendation methods. In line with that, less realistic romance movies are not recommended to Emily because of her low rating for "Twilight" (cf. Table 2.1). Thus, it is rather an exploratory task [cf. Mar06; Dir12], combined with effective recommender functionality. Accordingly, Emily might also influence the results on a lower level (i.e. of items themselves), using the ■ *choice-based dialog* that exploits the *semantics contained in the underlying latent factor model* (RQ1), as well as on a higher level (i.e. of properties of items and

related entities), relying on the concept of ■ *blended recommending* in order to get the results of this model *merged with those of content- and knowledge-based techniques* (RQ3).

8.2.3 Manipulation in complex scenarios

Finally, there are *more complex scenarios*, in which especially (but not only) users with high domain knowledge, and power users in general, would benefit from options to *manipulate the results* in a more holistic manner. This particularly applies to systems that do not use only a single factor model, but rely on an interplay of such a model with other methods. Again, this concerns users who already have an existing profile, but also new users or users with incomplete profiles. For illustration purposes, we consider the following two cases:

Benjamin prefers comedy and romance (similar to Amalia), but also likes action (see Table 2.1). With his broad interests and good domain knowledge, he is not satisfied with what is provided by typical recommender systems. Therefore, he looks forward to using the advanced features of our platform to explore for alternative recommendations.

Freddie has an interest in action and horror movies, though he has not yet provided any ratings for the latter (see again Table 2.1). He spends an evening together with his nephew. For this, he seeks a spooky movie that matches his own preferences, and, at the same time, is appropriate for a 9-year-old child.

The first case represents an exploratory learning task [cf. Mar06; KFK14], similar to the open-ended (main) tasks in the user experiments reported in Section 6.3.1 and 7.3. The other case corresponds to task IIb of the experiment reported in Section 6.3.2, but is extended to illustrate the richer functionality of our platform in comparison to the earlier prototype. In both scenarios, it comes in handy that users in our platform cannot only use either search and filtering mechanisms *or* recommendation functionalities: Since *model-based collaborative filtering is merged* with other methods based on the ■ *blended recommending* concept, they can instead use interaction mechanisms from a much broader range of options (RQ3). Accordingly, Benjamin is most satisfied with the rich interface shown in Figure 8.6. Because of his movie expertise, tech-savviness, and maximization behavior [cf. PDF07], he does not get overwhelmed or frustrated. Instead, he enjoys using the different facets and their values to indicate his preferences, and plays with the weights to adjust the system's final outcome as long as his information need evolves. The whole time, this outcome is automatically geared towards his general interests, which he highly appreciates. Occasionally, he also navigates to other perspectives (e.g. ■ *choice-based dialog*) or uses other mechanisms (e.g. ■ *weighting of tags* or ■ *application of critiques*).

Freddie's goal is more specific, but he is less experienced. His user profile is displayed at the top in Figure 8.9, together with a visualization of his user-tag vector based on the application of our extended matrix factorization method (a). Being the most obvious solution for him, he immediately starts to use the faceted filtering functionality in the recommendations perspective. Focused on the genre facet, he ■ *blends some content-based results* into his recommendations, which currently contain action-oriented movies because of his rating-based profile (as visible in the tag cloud): The recommendations resulting from the selection of the "family" genre are still in line with this profile thanks to the rescoring mechanism described in Section 8.1.2. Yet, Freddie's other preferences continue to be ignored due to the lack of ratings for horror movies (cf. Table 2.1). Since he does not know how to cope with this issue, he drags another tile from the genre facet into the working area. Unfortunately, his selection of the "horror" criterion leads to very heterogeneous

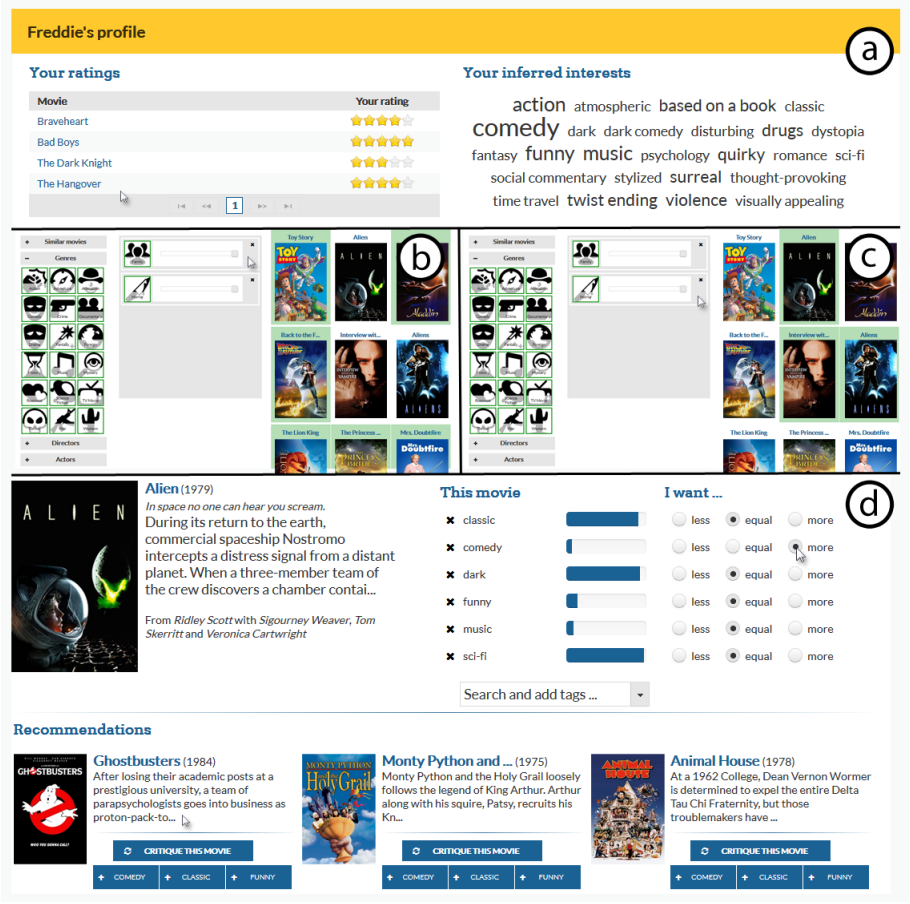

Figure 8.9 Screenshots for another case study: At the top, the user's profile is shown, again with ratings and tag cloud (a). The recommendations below additionally fulfill the selected criteria, but consist of disjunct item sets, which becomes visible when hovering one criterion (b) or the other (c). Critiques are consequently applied to one of the suggested movies, eventually yielding meaningful results (d).

results because of the weighted average that is calculated, with "Toy Story" on the one hand (b), "Alien" on the other (c). As a consequence, Freddie concentrates on those movies he personally likes: He proceeds to the item detail page of "Alien", and then to the ■ *critiquing perspective* (d). There, the suggestions are per se related to the critiqued movie, but also to the representation of his long-term preferences within the factor model. While horror and action are thus considered, the need for family-oriented movies got overruled. However, by applying a critique for "more comedy" movies, Freddie stumbles upon "Ghostbusters", which he thinks is an appropriate movie for his nephew, he will also enjoy himself (note that the ■ *choice-based dialog* would have allowed to steer the recommendations into a similar direction). Overall, these last two cases underline that in some situations, a sufficient degree of user control and experience may only be reached if all proposed methods are available, based on the *semantics contained in the underlying latent factor model* (RQ1), the *integration of item-related information* (RQ2), and the *combination with other recommendation and information filtering methods* (RQ3).

CHAPTER 9

> "Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience."
>
> — *Roger Bacon, English philosopher*

Conclusions

Recommender systems have reached a popularity that it is impossible to imagine video and music streaming platforms such as *Netflix* or *Spotify*, but also e-commerce websites such as *Amazon*, without these tools. Personalized methods have shown to reduce significantly the information overload users would otherwise be confronted with. Research has achieved tremendous success in improving accuracy and performance of these methods. But, deficiencies in controllability and transparency have been largely ignored. State-of-the-art collaborative filtering techniques amplify these deficiencies by using increasingly complex algorithms that learn highly abstract models in a completely automated manner. With the recognition that user-oriented aspects count equally or more to the success of recommender systems, interactive recommending approaches have therefore gained attention. However, as the literature review in Chapter 2 has shown, most of these approaches do not exploit the advantages of model-based collaborative filtering. At the same time, users are not supported by alternative methods in accordance with their current situation—be it from the same end of the spectrum that is illustrated in Figure 1.2, recommender systems, or from the other end, information retrieval systems, and consequently, more flexible mechanisms for browsing and filtering large result sets. We set us the goal to retain the benefits of modern collaborative filtering models in terms of personalization and efficiency, but to fill this research gap by letting users express their preferences more effectively, providing a broader range of mechanisms to always influence the recommendations according to situational needs, and thus, improving user experience overall. In this final chapter, we discuss the *contributions* we made towards this goal in relation to our initially posed research questions (see Section 1.3). In addition, we address possible *limitations* and provide an outlook on *future research*.

9.1 Contributions to the research questions

To achieve our overarching goal, we initially defined four objectives, which we addressed in the course of this thesis (cf. Figure 1.3 and 1.4): First, in Chapter 3, we introduced a *model* for structuring interactive methods that may be integrated into or with recommender systems based on latent factor models, i.e. typical implementations of model-based collaborative filtering. Based on this model of user interaction, which is depicted in Figure 3.1, we derived three research questions. In the succeeding chapters, we addressed these questions by presenting several *methods* to overcome the limitations indicated by this model. In a series of empirical *evaluations*, we were

able to show the value of these methods for improving user control and experience in the different phases of the recommendation process. By means of case studies based on the integrated recommendation *platform* presented in Chapter 8, we were moreover able to illustrate that the methods, taken together, bring us closer to interactive systems that fully adhere to our model. Put differently, our developments showed the potential for strengthening the formerly incomplete connections in this model, which is illustrated by the updated version in Figure 8.7. In the following, we discuss these findings in light of our research questions in more detail.

9.1.1 Exploiting semantics in latent factor models

First, the widely accepted assumption that the latent dimensions of matrix factorization models contain semantics related to real-world concepts, led us to the following research question:

RQ1: How to *exploit the semantics* in latent factor models for improving user control and experience?

In Section 3.1.1, we discussed the potential of pairwise comparisons as an alternative to rating-based preference elicitation. In line with this, we picked up the idea behind the above research question, which is explained in detail in Section 3.2.1, and proposed the *choice-based preference elicitation method* in Chapter 4 to strengthen the respective connection in our model of user interaction (dotted line in Figure 3.1). In the related user experiment, which is described in Section 4.3, we were able to show that this method effectively enables users to incrementally express their initial preferences—without any additional requirements on part of the system, only based on inherent properties of the underlying model. The exploratory comparison with several baselines, including a standard rating-based matrix factorization recommender, yielded superior results in all relevant dimensions. In context of the case studies presented in Section 8.2, it furthermore became clear that the method may be used successfully in the ongoing recommendation process. Regardless of these achievements, it has to be mentioned that the underlying assumption was similarly exploited in other (parallel) works, for instance, for visualization or explanation purposes [Ném*13; RSZ13]. Also, in our own (later) work, we presumed—and in one way or the other confirmed—the existence of relations to real-world concepts [KLZ17; KLZ18a; KLZ18b; Kun*19b]. However, our choice-based method showed for the first time that the patterns hidden in the user-item interaction data, which are used to learn the models of collaborative filtering systems, can effectively be exploited for practical user-oriented purposes.

In contrast to related approaches that also rely on comparisons [e.g. JBB11; RK12], our method *directly* exploits the underlying latent factor model, both for the comparisons that are shown to the user *and* for finding out about his or her relative preferences. This way, the computational effort remains the same, as expensive steps such as conversion of user ratings or item clustering can be avoided. Still, we were inspired by these works, but also findings from conjoint analysis [GS78; Hub05], and thus took into account a variety of aspects that are relevant from a user perspective for implementing the comparisons in an effective but also intuitive fashion. This led to the conversational interaction with binary choices between sets of sample items. Sampling the factor space in this manner accommodates for the interaction effects that typically exist among items, and thus automatically circumvents the problem inherent to active learning strategies based on ratings, i.e. determining informative items to confront the user with [cf. Rub*15; ERR16].

All these advantages should hold regardless of the underlying factorization technique—as long as users and items become embedded in a joint factor space. This property is fulfilled by many model-based collaborative filtering algorithms. Moreover, the criteria we suggested to determine the factor representatives are universally applicable. Thus, it should be relatively easy to integrate our interactive dialog in most contemporary systems, i.e. on top of the models that are anyway used by these systems. This is underlined by our own implementations: on the one hand, for the user experiment, based on standard matrix factorization and *MyMediaLite*, on the other hand, for the integrated recommendation platform, based on content-boosted matrix factorization and *Apache Mahout*. While it is known that even standard algorithms differ in accuracy due to low-level implementation differences [SB14a], the potential of our method was visible under *all* conditions. In line with that, over the years, others were inspired by our approach and achieved similar results [e.g. BR15; Liu*18], even if the idea was interpreted slightly different [e.g. GW15; TWK18] or applied in less frequently addressed domains [Ros*16]. In recent work, we even found indications that the idea can be transferred to deep learning [Tön19].

Summarizing these findings, it seems valid to give a positive answer to our first research question: A more extensive exploitation of the *semantics contained in latent factor models* can considerably contribute to *improving user control and experience* of model-based collaborative filtering systems. Consequently, our method can be seen as a promising, easy-to-implement vehicle to improve the *elicitation of (initial) preferences* in these systems, and thus as a first step towards our main goal.

9.1.2 Leveraging item-related information

In addition, we wanted to provide more expressive interaction mechanisms that do not require interaction on the level of items. In light of the potential shown by approaches that specifically use item-related information, we thus formulated the second research question as follows:

RQ2: How to *leverage item-related information* in addition to standard collaborative filtering feedback data for improving user control and experience?

Inspired by the approaches from interactive recommending research, we discussed in Section 3.1.2 the use of critiquing and weighting mechanisms to exert control over recommender systems. Then, in Section 3.2.2, we elaborated on the idea behind the above research question and presented a *content-boosted matrix factorization method* in Chapter 5. The specific way in which we extended the method of Forbes and Zhu [FZ11] made the latent factors accessible from the user interface. This paved the path for more advanced *interactive features* as extensions to collaborative filtering systems, based on concepts that are inherently meaningful to the user community. We described examples of these application possibilities of our method in Chapter 6. With tags as a running example, these showed the potential for strengthening the corresponding connection in our model of user interaction (dashed line in Figure 3.1): In two exploratory user studies, described in Section 6.3.1 and 6.3.2, as well as in the case studies presented in Section 8.2, we were able to confirm the improvements in the different phases of the collaborative filtering process, in particular, for accommodating situational needs without requiring users to (re-)rate single items, and without affecting their representation within the underlying model.

Yet, it makes sense to start the detailed discussion by highlighting that our method also contributes to opening up the black-box models of contemporary collaborative filtering systems. While several approaches aim at improving transparency, for instance, by visualizations based

on latent factors (see [Gan*09; Ném*13; Weg*18] as well as our other work [KLZ17]), this was not within the scope of this thesis.[1] Nevertheless, in contrast to other works in which side information is used for increasing accuracy of recommendations (cf. Section 2.2.4), or more rarely, their fundamental explainability (cf. Section 2.2.5), our experiments suggest that content boosting has potential in this regard as well: The qualitative analysis we performed as part of the offline evaluation illustrated that the additionally considered tags can effectively bring to light the meaning of the model dimensions. In the first user experiment, structural equation modeling highlighted the mediating role of transparency in case initial preferences are elicited via tags instead of ratings. In the second experiment, we observed a positive effect on the comprehensibility of the implemented critiquing process and its results. In both studies, participants' behavior indicated that they valued the tag cloud visualization of their profile. The successful implementation of these tag clouds, also in the integrated platform, underlines that our extended matrix factorization method can help explain the vector representations of long-term preferences. Remarkably, this is also true for users who have not (yet) provided any tags, but only conventional item feedback. However, explainability was not within the scope of this thesis either, so that further investigation of our method's contribution in this regard is also left to future work.

More interesting in light of the second research question is the contribution to controllability and user experience in general. Using a very diverse range of approaches, many improvements have been made regarding the objective accuracy of matrix factorization algorithms (cf. Section 2.2.4). As in these works, we analyzed the effectiveness of our method in Chapter 5 by means of offline experiments: We were able to confirm earlier results that showed the accuracy-related advantages of considering side information [cf. Kar*10; ML13; SLH13; NZ13; FC14; Alm*15]. However, our *empirical* evaluation for the first time gave an indication of the advantages from a user perspective. The first study compared the usage a content-boosted and a standard matrix factorization model: Participants were able to decide faster and were more satisfied with their chosen items. They preferred the interaction via tags in general, but, in particular, for expressing their *initial* preferences. The second study complemented the user-centric evaluation by a comparison against another established baseline, namely, an interactive, but purely content-based recommending approach: Participants appreciated the personalization of the critiquing process, which was possible thanks to the latent knowledge that became available through our method.

At first sight, these results appear to contradict the finding that a "few ratings are more valuable than [additional] metadata" [PT09; FO19]. However, this finding relies on a system perspective, whereas our different implementations alone underline the potential of content boosting for increasing interactivity—with features that usually require content- or knowledge-based techniques [e.g. CP12a; VSR12], or only affect the interplay between methods without providing an option to intervene in the underlying models [e.g. BOH12; Car*19]. Moreover, in addition to overcoming some of the widely discussed drawbacks in terms of controllability and transparency, these features seem to contribute to users' motivation to provide feedback. This can (partially) compensate the sparsity of user-item interaction data. On the other hand, availability of content information becomes a new requirement. In our prototypes, users were not able to add to this information themselves (which would be different in real-world systems). But, since only a numerical representation of the content attributes needs to be available, and we used datasets from the virtually similar *MovieLens* platform,[10] this was not even necessary. Therefore, neither the focus on tags, nor the specific datasets, are likely to have affected generalizability.

Consequently, we think it is safe to say that our experiments were successful in validating the usefulness of our method for providing users a higher degree of *control over the systems*, allowing them to steer the recommendations into the direction appropriate to their current situation. Accordingly, we can also positively answer the second research question: If model-based collaborative filtering algorithms are employed, *leveraging item-related information* in addition to standard feedback data can be a valuable means to further *improve user control and experience*.

9.1.3 Merging recommendation and information filtering methods

Despite the availability of these direct extensions to systems based on latent factor models, some scenarios called for additional methods to account for the complexity of the user's decision process. To finally achieve our goal, we thus formulated the third research question as follows:

RQ3: How to *merge model-based collaborative filtering* with other recommendation and information filtering methods for improving user control and experience?

In Section 3.1.3, we discussed the potential of the few interactive approaches to hybrid recommender systems as well as the fact that the interactive methods from the area of information filtering are usually decoupled from recommendation components. We picked up the idea behind the above research question, which is explained in detail in Section 3.2.3, and proposed the concept of *blended recommending* in Chapter 7, in this way addressing the remaining connection in our model of user interaction (dash-dotted line in Figure 3.1). In the user experiment presented in Section 7.3, we were able to show that by building upon faceted filtering, this concept enables users to take advantage of multiple recommendation methods at the same time. Questionnaire results and interaction analysis, but also the case studies presented in Section 8.2, shed a positive light on the interfaces that we implemented accordingly, in particular, for situations in which collaborative filtering alone is not sufficient to reach the search goal. Despite the lower complexity, entirely manual exploration, as in the faceted filtering interface that we used as a baseline in the exploratory user study, simultaneously appeared to have *no* usability advantage.

In general, we argued a lot about the need to get rid of the requirement of providing item ratings. However, as soon as alternative methods come into play, our other enhancements are stretched to their limits. Relying exclusively on content attributes, on the other hand, may not be appropriate either, in particular, if the respective interface elements have no connection with each other. Then, users are forced to keep using a specific, possibly limited decision strategy [Jam*15], even as their information need evolves [Bat89]. In large and unknown domains or in case of experience products, this can make the task of finding suitable items cognitively overly exhausting. But, in line with calls for providing users with different algorithms [Eks*15] and preference elicitation methods [KRW11], our hybrid, and, at the same time, fully user-controlled approach, enables users to manipulate the final outcome of the system on a *superordinate* level. This includes using collaborative filtering whenever necessary (by means of the "items similar to" facet, i.e. without having to rate items), but also content- and knowledge-based techniques in case properties of items or related entities are known already. In contrast to other approaches that allow selecting or weighting different methods (see Section 2.3.2.2), this does not require users to deal directly with algorithms or datasources. These components are also easily exchangeable thanks to the common hybridization strategy, which is well illustrated by the completely new implementation in the integrated recommendation platform. Another consequence of this strategy is that users can refine the results without spending too much effort on observing the logical implications of

their actions: In contrast to conventional filtering approaches (Section 2.3.3), empty result sets cannot occur. In addition, the visual clues that are provided have shown to contribute to revealing the sources of recommendations, which is highly important in hybrid systems [cf. PBT14].

Noteworthy, we primarily targeted ad hoc preference elicitation, though our integrated platform already demonstrates that existing user profiles may be taken into account as well, in fact, just by adding another hybridization step. In the user experiment, however, a persistent profile was neither required nor created. Nonetheless, the results showed that participants were able to obtain recommendations in line with their personal preferences, i.e. blended recommending alleviated the problem of acquiring user-item interaction data. Moreover, the system's outcome seemed to keep up with the complexity of the decision process, which is often not the case in pure collaborative filtering systems, no matter how well they are personalized or which enhancements are made. Notwithstanding these results, the consideration of long-term preferences needs further investigation, then of course also with a conventional recommender as a baseline. Beyond that, the potential has yet been shown for exploratory learning tasks only, i.e. situations in which users have at most a vague search goal [cf. Mar06; KFK14]. With respect to the suitability for *directed* searches, the current baseline interface, in contrast, achieved higher scores. This also underlines the need for additional experiments, namely to examine whether this advantage persists if users are allowed to switch to the other proposed features depending on what they think is most appropriate in the current phase of the recommendation process. While we expect a positive effect of our developments, the potential of a holistic integration of the underlying methods has yet only been shown anecdotally (e.g. case studies, comments at the *RecSys* conference[37]).

Nonetheless, our exploratory findings illustrate sufficiently that blended recommending allows for the *manipulation in complex scenarios* in which neither the functionalities provided by systems that rely on standard mechanisms, nor our enhancements, provide enough support. The gap to systems that employ more flexible and controllable methods seems closed while the benefits of automated systems are preserved. Therefore, we cannot only answer the third research question, but also conclude this thesis on a positive note: *Merging recommendation and information filtering methods* finally contributes to *improving user control and experience* of model-based collaborative filtering systems to an extent that we think it is safe to say that we achieved our main goal and can provide users at all times, in one way or the other, with adequate interaction possibilities.

9.2 Limitations and future research

Although we came to the conclusion that we made model-based collaborative filtering systems as interactive as possible in the context of our research, we want to point out that all methods proposed in this thesis have limitations, could be implemented differently, or replaced by alternative solutions. Moreover, we acknowledge as a limitation that all our experiments were exploratory in nature. In the following, we discuss some of these aspects in more detail and provide an outlook on future work. Note that we do not address very specific limitations that we have observed during the evaluations and therefore discussed in previous chapters.

Choice-based preference elicitation In a first step, we are interested in *comparing* our novel preference elicitation method with more sophisticated baselines, in particular, active learning techniques based on comparisons of pairs or groups of items [e.g. RK12; BR15; CHT15; Liu*18].

In addition to further user experiments, novel offline evaluation methodologies for active learning [cf. ERR14; CB18] might be used to get a more solid foundation for some of the decisions we made when designing the interactive dialog. Concerning the number of dialog steps, some insights can be found in later works. For example, the group-based approach [cf. CHT15] required fewer steps, but increased cognitive load [Ros*17]. However, since users had to choose between multiple item clusters, the binary choice interface might not have been the most appropriate baseline in this case. Also in experiments on the number of latent factors that leads to high quality recommendations at minimum effort, our method served as a baseline [Liu*18]. Again, the results were of limited value, but the proposed *dynamic selection* of the next most important factor is indeed an interesting topic for future research, in particular, as the order of the factors, in our approach, is currently the same for all users, at least in cold-start situations.

In this context, it must be noted that the current experiment was limited to exactly these situations. In fact, we proposed our method without *personalization*. In the meantime, however, other authors addressed situations in which user profiles exist. They found that it is sufficient to initialize user-factor vectors just by choosing different starting locations in the latent space [cf. TWK18; Liu*18]. Still, empirical investigation is necessary, especially as we would like to gear the *entire* choice process towards the current user. For example, raising the weight of familiar items could reduce the dependence on general popularity as a criterion for sampling the item space. Moreover, given that users only get a superficial impression of a few sample items, we are interested in taking into account *personal characteristics* such as decision style [cf. Kah11], as they could have a considerable impact on their behavior. In this regard, it may be worth investigating whether preferences can also be elicited on a level above these items. For movies, this could mean extracting imagery from the source material [cf. Del*19] to display compositions of pivotal scenes instead of movie posters. Such an "experiential impression" might also be useful for other domains, from hotel recommendations, with pictures of amenities and surroundings, to suggestions for digital cameras, with photos taken with the cameras themselves.

Content-boosted matrix factorization and related interactive features Also concerning the interactive features we proposed based on our extended matrix factorization method, numerous improvements can be made. Yet, even though we address generalizability more extensively below, the *choice of background data* has to be mentioned first, given its particular importance for content boosting: The positive effects of the specific method have already been shown in a variety of domains [FZ11; NZ13; Zha*14]. However, since our user experiments were the first that confirmed these effects from a user perspective, it is still necessary to test whether the subjective advantages can be transferred to other domains. Besides, it cannot be taken for granted that user-generated tags always depict the most valuable source of side information: First, despite our efforts to find an ideal parameterization in offline experiments, other (larger) subsets or completely different datasets might lead to better results. Second, other types of content data might contribute more to the comprehensibility of the associations established by our method with the factors. For example, extracting attributes directly from user-written product reviews (as in our other work [Feu*17]) could be a useful alternative in certain domains. Finally, while tags are generally well understood, it has been found that domain knowledge affects the ability to interpret them [KFK10]. Also for these reasons, further research in other domains and with other types of data is clearly necessary. In addition, this will allow for *structural equation modeling*, which may reveal possible relationships between the assessment of system aspects and personal character-

istics such as domain knowledge, and thereby explain the usefulness of the novel interaction mechanisms for individual types of users.

Based on structural equation models, we already observed the mediating effect of *transparency*. However, we only briefly touched upon this aspect in the rest of this thesis:[1] In the user experiments, tag cloud visualizations of the usually opaque user representations were visible. This yielded positive qualitative feedback, but, none of the experiments was targeted explicitly at *explainability*. Similarly, the examples of the application of our method were focused on eliciting preferences and increasing control, whereas providing explanations was only a by-product. Therefore, a dedicated experiment, including a comparison with other tag-based explanation approaches [e.g. VSR09; GGJ11], is also on our research agenda. Nevertheless, the tag-based visualizations are examples of the many indications that the dimensions of latent factor models bear an intelligible meaning. In this context, it has to be noted that the factorizations are not unambiguous. As a consequence, not only the effects of domain and dataset need to be investigated, but also how algorithm and parameterization affect quality and comprehensibility of recommendations, and thus, the effectiveness of the novel interaction mechanisms. Also for this reason, we plan to study the *application of content boosting* with other model-based techniques. At the same time, one can easily imagine a sheer endless number of *alternative interaction mechanisms*, from minor variations to completely new features (so far, we only proposed three examples). For instance, if attributes were extracted from product reviews, those reviews related to the attributes selected by the user (and their sentiments) could be included both for exploring the results and clarifying the algorithmic reasoning (as we suggested in other work [DLZ18][2]).

All this enlarges the space of design variables to an extent that it becomes not feasible to explore all solutions in statistically sound user experiments, in particular, as the application context also plays a non-negligible role. Thus, we plan to use *simulation studies* (as conducted recently for critique-based approaches [cf. Xie*18]) to justify parameterizations and to run comparisons against other methods in a more economical manner. In general, structured evaluation plans will be useful for further research on interactive recommending [cf. OG08; GS15; Kon18]—similar to the efforts of making offline experiments more comparative [SB14a; SB14b], but taking into account the caveats of user experiments, which we discussed in other work [Loe*18; LZ19a].

Blended recommending Also with respect to our last methodological contribution, one of the most obvious directions for further development is the *exploitation of user-generated data*: The techniques we used to implement blended recommending were yet limited to background data in the form of user-item feedback and predefined metadata. In related work, we have already shown that the concept is in principle capable of implementing a natural language processing pipeline [Feu*17]. However, the personal relevance of statements that can be found in product reviews concerning domain-specific aspects of items and related entities, as well as the sentiments of other users regarding these aspects [cf. DZ20; DKZ20], have not yet been considered. Accordingly, we plan to exploit these data to provide richer facets. In addition, social network data could be of interest, for instance, to allow for the selection of other persons as criteria [cf. BOH12; Car*19], or to leverage the social graph structure, including (direct and indirect) relationships between users, to improve the calculation of relevance scores.

In general, more extensively making use of *personalization* constitutes an important topic for future research. In the current experiment, we obtained promising results even without tak-

ing long-term preference profiles into account. However, in the time since we proposed our concept, we extended the hybrid configuration to include predictions of content-boosted matrix factorization as available in our integrated platform. Whereas informal interviews with test users confirmed a positive effect of incorporating general interests in this way (which would otherwise have to be specified explicitly in addition to short-term goals), this still needs to be investigated empirically. Moreover, from a methodological point of view, the question remains whether and how the actions performed by users *within* the facet widgets should be reflected back into user profiles. With the richer facets mentioned above, their dynamic adaptation also needs to be reconsidered, beyond the early attempts to suggest facet values in *MyMovieMixer* or to show alternatives in the recommendation platform. Overall, we are thus interested in exploring further options to support users in reaching their search goal with less effort.

While one can experiment with the implementation of specific components (style of visual clues, scale and range of sliders), this also applies to the *general layout*: Using the proposed working area could be a problem in real-world scenarios (e.g. on mobiles). Therefore, users should be provided automatically with the most adequate interaction mechanisms depending on their current progress and the complexity of the situation—different from our integrated recommendation platform, where they have to decide on their own. Moreover, all types of input data should be considered in line with their specificity: Not only differ facets in relative importance, but selecting a facet value also represents a stronger preference signal than a pairwise comparison of sample items, and a weaker one than keyword-based search. The interaction analysis we performed as part of the exploratory study on blended recommending already provided some insights for addressing these aspects. However, a larger amount of interaction data is required to get a deeper understanding of user behavior. For this, our platform is exactly the right tool. In future research, it will also help us to confirm the current findings and to add further *comparisons*—with alternative filtering interfaces, but also automated recommender systems. In this way, we eventually want to find out which of our proposed methods are most appropriate for which type of user, not only depending on individual needs and current situation, but also personal characteristics such as decision style and maximization behavior [cf. SB95; PDF07; Kah11; HSM16].

Replicability and generalizability Eventually, with respect to all our contributions, one has to keep in mind that the positive image primarily stems from a limited number of specific user experiments we conducted to address our research questions. Most of our conclusions were thus drawn from exploratory results obtained under controlled laboratory conditions. Although we argued that most of our findings should be generalizable, in particular, because of the underlying principle of collaborative filtering, and we were able to replicate earlier successes whenever possible, it therefore is essential to conduct *follow-up studies*—in line with the general movement towards reproducible and replicable recommender research [cf. SB14a; Bee*16; DCJ19].

Most importantly, the application of our methods needs to be validated in *other domains* and with *different datasets*. So far, we only used explicit user-item feedback data, although implicit feedback has shown to model user behavior more accurately [PA11; JWK14]. Due to the ability of collaborative filtering to generalize, we assumed that our enhancements would perform similarly well with other datasources. But, the novel interactive features come with an even increased amount of explicit feedback, so that we still support the calls for further research on the use of this kind of input data [cf. SB18]. Beyond that, we only used datasets from the movie domain. We argued that this should not be a problem either, in particular, as movies are popular in

recommender research and can be considered representative of other experience products. Nevertheless, it appears at least questionable, for example, whether our choice-based dialog would perform equally well in domains of higher complexity or with search products. However, this concern is true for collaborative filtering in general, which is highlighted by the fact that content- and knowledge-based techniques increasingly gain importance in such scenarios (cf. our work on blended recommending and on related topics [NLZ20][2]). Notwithstanding these considerations, further experiments with other types of items (and other item-related information) are clearly needed in order to confirm the effectiveness of our interactive methods. Running these experiments will neither be a problem from a technical point of view, nor require much effort, thanks to the flexible implementation based on our *TagMF* framework.

Of course, some of the empirical findings reported in this thesis may be spurious because of the exploratory nature of the current experiments, in particular, given the large number of statistical tests we conducted without accounting for testing multiple hypotheses.[19] Furthermore, larger samples would have led to more meaningful results and enabled us to use advanced analysis techniques more often. For the sake of comparison, more consistent questionnaires would have been required, which currently varied across studies simply due to the fact that our research was conducted over several years. For these reasons, *further (confirmatory) analyses* are clearly necessary, possibly using Bayesian statistics [cf. Gel*13]. Nonetheless, we are confident that the experiments were sufficient to demonstrate the effectiveness of our interactive methods, and, given the holistic approach we used to analyze and interpret the results, to explore their potential for improving user control and experience, i.e. in line with the research questions. The illustrative case studies can be seen as a first—though qualitative—step towards an even *broader evaluation* of these methods. To investigate how users actually go back and forth between specific and more general methods, depending on long-term preferences and short-term goals, but also personal characteristics, empirical user testing is however required. With our integrated platform, we already have a vehicle for in-depth user experiments and even field studies. This will also enable us to corroborate our *model of user interaction*, which served us to structure this research, by analyzing real usage data once the system is deployed online. In this way, we expect to gain insights into the usefulness of the individual interaction mechanisms if users can choose from all methods described in this thesis at their convenience and over a longer period of time.

Closing remarks Regardless of the effectiveness of the proposed methods for enhancing collaborative filtering systems and the promising results in terms of user control and experience, one cannot ignore that other model-based techniques have gained popularity. The underlying problem of low interactivity remains, so that the implementation of our novel interaction mechanisms on top of these techniques is certainly a topic for future research. However, more recent approaches, for example, based on deep learning, are seen increasingly critical, both with respect to algorithmic progress [DCJ19] and interpretability as well as explainability [Rud19]. Matrix factorization, on the other hand, is well understood and widely used, often with a performance that is similar to or even better than the performance of much more complex techniques. In light of the above arguments concerning generalizability, we thus encourage all readers, even those who prefer the application of other algorithms, to see our findings at least as indications that using collaborative filtering does not preclude from taking user-oriented aspects into account. Also, this thesis hopefully shows that interactive recommending research does not have to stop as soon as model-based algorithms—of any kind—come into play. Wherever that may lead…

Bibliography

[AB13] Jae-Wook Ahn and Peter Brusilovsky. "Adaptive Visualization for Exploratory Information Retrieval." In: *Information Processing & Management* 49.5 (2013), pp. 1139–1164.

[AB15] Xavier Amatriain and Justin Basilico. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Recommender Systems in Industry: A Netflix Case Study, pp. 385–419.

[AF01] Taiwo Amoo and Hershey H. Friedman. "Do Numeric Values Influence Subjects' Responses to Rating Scales?" In: *Journal of International Marketing and Marketing Research* 26 (2001), pp. 41–46.

[AK15] Gediminas Adomavicius and YoungOk Kwon. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Multi-Criteria Recommender Systems, pp. 847–880.

[Ale*14] Marharyta Aleksandrova, Armelle Brun, Anne Boyer, and Oleg Chertov. "Search for User-Related Features in Matrix Factorization-Based Recommender Systems." In: *ECML-PKDD '14: Proceedings of the 2014 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. 2014.

[Ale*17] Marharyta Aleksandrova, Armelle Brun, Anne Boyer, and Oleg Chertov. "Identifying Representative Users in Matrix Factorization-Based Recommender Systems: Application to Solving the Content-Less New Item Cold-Start Problem." In: *Journal of Intelligent Information Systems* 48.2 (2017), pp. 365–397.

[Alm*15] Amjad Almahairi, Kyle Kastner, Kyunghyun Cho, and Aaron Courville. "Learning Distributed Representations from Reviews for Collaborative Filtering." In: *RecSys '15: Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2015, pp. 147–154.

[Alv*19] Oscar Luis Alvarado Rodriguez, Veronika Vanden Abeele, David Geerts, and Katrien Verbert. ""I Really Don't Know What 'Thumbs Up' Means": Algorithmic Experience in Movie Recommender Algorithms." In: *Human-Computer Interaction — INTERACT 2019*. Ed. by David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris. Vol. 11748. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2019, pp. 521–541.

[Ama*09] Xavier Amatriain, Josep M. Pujol, Nava Tintarev, and Nuria Oliver. "Rate it Again: Increasing Recommendation Accuracy by User Re-Rating." In: *RecSys '09: Proceedings of the 3rd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2009, pp. 173–180.

[AN16] Behnoush Abdollahi and Olfa Nasraoui. "Explainable Matrix Factorization for Collaborative Filtering." In: *WWW '16: Proceedings of the 25th International Conference Companion on World Wide Web*. New York, NY, USA: ACM, 2016, pp. 5–6.

[AN17] Behnoush Abdollahi and Olfa Nasraoui. "Using Explainability for Constrained Matrix Factorization." In: *RecSys '17: Proceedings of the 11th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2017, pp. 79–83.

[APO09] Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. "I Like it... I Like it not: Evaluating User Ratings Noise in Recommender Systems." In: *UMAP '09: Proceedings of the 17th International Conference on User Modeling, Adaptation and Personalization*. Berlin, Germany: Springer, 2009, pp. 247–258.

[APO16] Ivana Andjelkovic, Denis Parra, and John O'Donovan. "Moodplay: Interactive Mood-based Music Discovery and Recommendation." In: *UMAP '16: Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2016, pp. 275–279.

[Ard*03] Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. "INTRIGUE: Personalized Recommendation of Tourist Attractions for Desktop and Hand Held Devices." In: *Applied Artificial Intelligence* 17.8-9 (2003), pp. 687–714.

[AS94] Christopher Ahlberg and Ben Shneiderman. "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays." In: *CHI '94: Proceedings of the 12th ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1994, pp. 313–317.

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." In: *IEEE Transactions on Knowledge and Data Engineering* 17.6 (2005), pp. 734–749.

[AT15] Gediminas Adomavicius and Alexander Tuzhilin. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Context-Aware Recommender Systems, pp. 191–226.

[Bak*13] Fedor Bakalov, Marie-Jean Meurs, Birgitta König-Ries, Bahar Sateli, René Witte, Greg Butler, and Adrian Tsang. "An Approach to Controlling User Models and Personalization Effects in Recommender Systems." In: *IUI '13: Proceedings of the 18th International Conference on Intelligent User Interfaces*. Visualization: ACM, 2013, pp. 49–56.

[Bal*11] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. "Context-Aware Places of Interest Recommendations and Explanations." In: *Joint Proceedings of the 1st Workshop on Decision Making and Recommendation Acceptance Issues in Recommender Systems (DEMRA '11) and the 2nd Workshop on User Models for Motivational Systems: The Affective and the Rational Routes to Persuasion (UMMS '11)*. Vol. 740. 2011, pp. 19–26.

[Bat89] Marcia J. Bates. "The Design of Browsing and Berrypicking Techniques for the Online Search Interface." In: *Online Information Review* 13.5 (1989), pp. 407–424.

[BB09] Darius Braziunas and Craig Boutilier. "Elicitation of Factored Utilities." In: *AI Magazine* 29.4 (2009), pp. 79–92.

[BC12] Suhrid Balakrishnan and Sumit Chopra. "Two of a Kind or the Ratings Game? Adaptive Pairwise Preferences and Latent Factor Models." In: *Frontiers of Computer Science* 6.2 (2012), pp. 197–208.

[Bee*16] Joeran Beel, Corinna Breitinger, Stefan Langer, Andreas Lommatzsch, and Bela Gipp. "Towards Reproducibility in Recommender-Systems Research." In: *User Modeling and User-Adapted Interaction* 26.1 (2016), pp. 69–101.

[Ben*07] David Ben-Shimon, Alexander Tsikinovsky, Lior Rokach, Amnon Meisles, Guy Shani, and Lihi Naamani. "Recommender System from Personal Social Networks." In: *Advances in Intelligent Web Mastering*. Ed. by Katarzyna M. Wegrzyn-Wolska and Piotr S. Szczepaniak. Vol. 43. Advances in Soft Computing. Springer, 2007, pp. 47–55.

[BHK98] John S. Breese, David Heckerman, and Carl Kadie. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering." In: *UAI '98: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann, 1998, pp. 43–52.

[BHY97] Robin Burke, Kristian J. Hammond, and Benjamin Young. "The FindMe Approach to Assisted Browsing." In: *IEEE Expert* 12.4 (1997), pp. 32–40.

[Bie*17] Kai Biefang, Johannes Kunkel, **Benedikt Loepp**, and Jürgen Ziegler. "Eine Sandbox zur physisch-virtuellen Exploration von Ausgrabungsstätten." In: *Mensch & Computer 2017 – Workshopband*. Gesellschaft für Informatik, 2017.

[BJG13] Claudia Becerra, Sergio Jimenez, and Alexander Gelbukh. "Towards User Profile-Based Interfaces for Exploration of Large Collections of Items." In: *Decisions@RecSys '13: Proceedings of the 3rd Workshop on Human Decision Making in Recommender Systems*. 2013, pp. 9–16.

[BK07] Robert M. Bell and Yehuda Koren. "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights." In: *ICDM '07: Proceedings of the 7th IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE, 2007, pp. 43–52.

[BKM09] Aaron Bangor, Philip Kortum, and James Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale." In: *Journal of Usability Studies* 4.3 (2009), pp. 114–123.

[BKR08] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. "Mediation of User Models for Enhanced Personalization in Recommender Systems." In: *User Modeling and User-Adapted Interaction* 18.3 (2008), pp. 245–286.

[BKV07a] Robert M. Bell, Yehuda Koren, and Chris Volinsky. "Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems." In: *KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2007, pp. 95–104.

[BKV07b] Robert M. Bell, Yehuda Koren, and Chris Volinsky. *The BellKor Solution to the Netflix Prize*. Tech. rep. AT&T Labs Research, 2007.

[BKV10] Robert M. Bell, Yehuda Koren, and Chris Volinsky. "All Together Now: A Perspective on the Netflix Prize." In: *CHANCE* 23.1 (2010), pp. 24–29.

[BL01] Ralf Bender and Stefan Lange. "Adjusting for Multiple Testing – When and How?" In: *Journal of Clinical Epidemiology* 54.4 (2001), pp. 343–349.

[BL07] James Bennett and Stan Lanning. "The Netflix Prize." In: *Proceedings of the KDD Cup and Workshop 2007*. 2007.

[Bob*13] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. "Recommender Systems Survey." In: *Knowledge-Based Systems* 46 (2013), pp. 109–132.

[BOB82] Nicholas J. Belkin, Robert N. Oddy, and Helen M. Brooks. "ASK for Information Retrieval: Part I. Background and Theory." In: *Journal of Documentation* 38.2 (1982), pp. 61–71.

[Bog*13] Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Anna Xambó, Emilia Gómez, and Perfecto Herrera. "Semantic Audio Content-Based Music Recommendation and Visualization Based on User Preference Examples." In: *Information Processing & Management* 49.1 (2013), pp. 13–33.

[BOH12] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. "TasteWeights: A Visual Interactive Hybrid Recommender System." In: *RecSys '12: Proceedings of the 6th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2012, pp. 35–42.

[BOH13] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. "LinkedVis: Exploring Social and Semantic Career Recommendations." In: *IUI '13: Proceedings of the 18th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2013, pp. 107–116.

[Bol*10] Dirk Bollen, Bart P. Knijnenburg, Martijn C. Willemsen, and Mark P. Graus. "Understanding Choice Overload in Recommender Systems." In: *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010, pp. 63–70.

[BR15] Laura Blédaité and Francesco Ricci. "Pairwise Preferences Elicitation and Exploitation for Conversational Collaborative Filtering." In: *HT '15: Proceedings of the 26th ACM Conference on Hypertext and Social Media*. New York, NY, USA: ACM, 2015, pp. 231–236.

[BR99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. New York, NY, USA: ACM, 1999.

[Bro96] John Brooke. "SUS – A Quick and Dirty Usability Scale." In: *Usability Evaluation in Industry*. London, UK: Taylor & Francis, 1996, pp. 189–194.

[Bru*08] Arnaud de Bruyn, John C. Liechty, Eelko K.R.E. Huizingh, and Gary L. Lilien. "Offering Online Recommendations with Minimum Customer Input Through Conjoint-Based Decision Aids." In: *Marketing Science* 27.3 (2008), pp. 443–460.

[Bur00] Robin Burke. "Knowledge-Based Recommender Systems." In: *Encyclopedia of Library and Information Systems* 69.32 (2000), pp. 180–201.

[Bur07] Robin Burke. "Hybrid Web Recommender Systems." In: *The Adaptive Web. Methods and Strategies of Web Personalization*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Vol. 4321. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2007, pp. 377–408.

[BZ17] Catalin-Mihai Barbu and Jürgen Ziegler. "Towards a Design Space for Personalizing the Presentation of Recommendations." In: *EnCHIReS '17: Proceedings of the 2nd Workshop on Engineering Computer-Human Interaction in Recommender Systems*. 2017, pp. 10–17.

[BZ18] Catalin-Mihai Barbu and Jürgen Ziegler. "Designing Interactive Visualizations of Personalized Review Data for a Hotel Recommender System." In: *RecTour '18: Proceedings of the 3rd Workshop on Recommenders in Tourism*. 2018, pp. 7–12.

[BZM03] Craig Boutilier, Richard S. Zemel, and Benjamin Marlin. "Active Collaborative Filtering." In: *UAI '03: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann, 2003, pp. 98–106.

[Can*15] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Cross-Domain Recommender Systems, pp. 919–959.

[Car*08] Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Dumais. "Here or There – Preference Judgments for Relevance." In: *ECIR '08: Proceedings of the 30th European Conference on Information Retrieval*. Berlin, Germany: Springer, 2008, pp. 16–27.

[Car*19] Bruno Cardoso, Gayane Sedrakyan, Francisco Gutiérrez, Denis Parra, Peter Brusilovsky, and Katrien Verbert. "IntersectionExplorer, a Multi-Perspective Approach for Exploring Recommendations." In: *International Journal of Human-Computer Studies* 121 (2019), pp. 73–92.

[CAS11] Ilknur Celik, Fabian Abel, and Patrick Siehndel. "Towards a Framework for Adaptive Faceted Search on Twitter." In: *DAH '11: Proceedings of the 2nd International Workshop on Dynamic and Adaptive Hypertext*. 2011, pp. 11–22.

[CB18] Diego Carraro and Derek Bridge. "A More Comprehensive Offline Evaluation of Active Learning in Recommender Systems." In: *REVEAL '18: Proceedings of the Workshop on Offline Evaluation for Recommender Systems*. 2018.

[CEG17] Paolo Cremonesi, Mehdi Elahi, and Franca Garzotto. "User Interface Patterns in Recommendation-Empowered Content Intensive Multimedia Applications." In: *Multimedia Tools and Applications* 76.4 (2017), pp. 5275–5309.

[Cen*17] Federica Cena, Cristina Gena, Pierluigi Grillo, Tsvi Kuflik, Fabiana Vernero, and Alan J. Wecker. "How Scales Influence User Rating Behaviour in Recommender Systems." In: *Behaviour & Information Technology* 36.10 (2017), pp. 985–1004.

[CGT12] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. "User Effort vs. Accuracy in Rating-Based Elicitation." In: *RecSys '12: Proceedings of the 6th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2012, pp. 27–34.

[CHR16] Konstantina Christakopoulou, Katja Hofmann, and Filip Radlinski. "Towards Conversational Recommender Systems." In: *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016, pp. 815–824.

[CHT15] Shuo Chang, F. Maxwell Harper, and Loren G. Terveen. "Using Groups of Items for Preference Elicitation in Recommender Systems." In: *CSCW '15: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*. New York, NY, USA: ACM, 2015, pp. 1258–1269.

[CHV15] Pablo Castells, Neil J. Hurley, and Saul Vargas. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Novelty and Diversity in Recommender Systems, pp. 881–918.

[CKT10] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. "Performance of Recommender Algorithms on Top-N Recommendation Tasks." In: *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010, pp. 39–46.

[Cla*01] Mark Claypool, Phong Le, Makoto Wased, and David Brown. "Implicit Interest Indicators." In: *IUI '01: Proceedings of the 6th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2001, pp. 33–40.

[Cla*99] Mark Claypool, Anuja Gokhale, Tim Miranda, Paul Murnikov, Dmitry Netes, and Matthew Sartin. "Combining Content-Based and Collaborative Filters in an Online Newspaper." In: *Proceedings of the ACM SIGIR Workshop on Recommender Systems*. 1999.

[CM06] Judith A. Chevalier and Dina Mayzlin. "The Effect of Word of Mouth on Sales: Online Book Reviews." In: *Journal of Marketing Research* 43.3 (2006), pp. 345–354.

[CMO15] Victor Codina, Jose Mena, and Luis Oliva. "Context-Aware User Modeling Strategies for Journey Plan Recommendation." In: *UMAP '15: Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization*. Berlin, Germany: Springer, 2015, pp. 68–79.

[Cos*03] Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. "Is Seeing Believing? How Recommender Interfaces Affect Users' Opinions." In: *CHI '03: Proceedings of the 21st ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2003, pp. 585–592.

[Cow10] Nelson Cowan. "The Magical Mystery Four: How is Working Memory Capacity Limited, and Why?" In: *Current Directions in Psychological Science* 19.1 (2010), pp. 51–57.

[CP12a] Li Chen and Pearl Pu. "Critiquing-Based Recommenders: Survey and Emerging Trends." In: *User Modeling and User-Adapted Interaction* 22.1-2 (2012), pp. 125–150.

[CP12b] Li Chen and Pearl Pu. "Experiments on User Experiences with Recommender Interfaces." In: *Behaviour & Information Technology* 33.4 (2012), pp. 372–394.

[CR15] Ivan Sanchez Carmona and Sebastian Riedel. "Extracting Interpretable Models from Matrix Factorization Models." In: *CoCo '15: Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*. 2015.

[Crn*11] Tarik Crnovrsanin, Isaac Liao, Yingcai Wuy, and Kwan-Liu Ma. "Visual Recommendations for Network Navigation." In: *EuroVis '11: Proceedings of the 13th Eurographics / IEEE VGTC Conference on Visualization*. Chichester, UK: The Eurographs Association & John Wiley & Sons, Ltd., 2011, pp. 1081–1090.

[CS00] Paul Cotter and Barry Smyth. "PTV: Intelligent Personalised TV Guides." In: *AAAI '00: Proceedings of the 17th National Conference on Artificial Intelligence*. Palo Alto, CA, USA: AAAI Press, 2000, pp. 957–964.

[CSZ19] Ludovik Coba, Panagiotis Symeonidis, and Markus Zanker. "Personalised Novel and Explainable Matrix Factorization." In: *Data & Knowledge Engineering* 122 (2019), pp. 142–158.

[Das*07] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. "Google News Personalization: Scalable Online Collaborative Filtering." In: *WWW '07: Proceedings of the 16th International Conference on World Wide Web*. New York, NY, USA: ACM, 2007, pp. 271–280.

[Das*10] Sudipto Das, Yannis Sismanis, Kevin S. Beyer, Rainer Gemulla, Peter J. Haas, and John McPherson. "Ricardo: Integrating R and Hadoop." In: *SIGMOD '10: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2010, pp. 987–998.

[Dav*10] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. "The YouTube Video Recommendation System." In: *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010, pp. 293–296.

[DCJ19] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. "Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 101–109.

[DEC16] Yashar Deldjoo, Mehdi Elahi, and Paolo Cremonesi. "Using Visual Features and Latent Factors for Movie Recommendation." In: *CBRecSys '16: Proceedings of the 3rd Workshop on New Trends in Content-based Recommender Systems*. 2016, pp. 15–18.

[Dee*90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. "Indexing by Latent Semantic Analysis." In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407.

[Del*19] Yashar Deldjoo, Maurizio Ferrari Dacrema, Mihai Gabriel Constantin, Hamid Eghbal-Zadeh, Stefano Cereda, Markus Schedl, Bogdan Ionescu, and Paolo Cremonesi. "Movie Genome: Alleviating New Item Cold Start in Movie Recommendation." In: *User Modeling and User-Adapted Interaction* 29.2 (2019), pp. 291–343.

[Dha97] Ravi Dhar. "Consumer Preference for a No-Choice Option." In: *Journal of Consumer Research* 24.2 (1997), pp. 215–231.

[DI08] Wisam Dakka and Panagiotis G. Ipeirotis. "Automatic Extraction of Useful Facet Hierarchies from Text Databases." In: *ICDE '08: Proceedings of the 24th IEEE International Conference on Data Engineering*. Washington, DC, USA: IEEE, 2008, pp. 466–475.

[Dia*14] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. "Jointly Modeling Aspects, Ratings and Sentiments for Movie Recommendation (JMARS)." In: *KDD '14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2014, pp. 193–202.

[Dir12] Abdigani Mohamed Diriye. "Search Interfaces for Known-Item and Exploratory Search Tasks." Doctoral dissertation. London, UK: University College London, 2012.

[DK10] Christian Desrosiers and George Karypis. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Berlin, Germany: Springer, 2010. Chap. A Comprehensive Survey of Neighborhood-Based Recommendation Methods, pp. 107–144.

[DKZ20] Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Explaining Recommendations by Means of Aspect-Based Transparent Memories." In: *IUI '20: Proceedings of the 25th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2020, pp. 166–176.

[DLZ15] Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Merging Latent Factors and Tags to Increase Interactive Control of Recommendations." In: *RecSys '15: Poster Proceedings of the 9th ACM Conference on Recommender Systems*. 2015.

[DLZ16a] Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Tag-Enhanced Collaborative Filtering for Increasing Transparency and Interactive Control." In: *UMAP '16: Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2016, pp. 169–173.

[DLZ16b] Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Towards Understanding Latent Factors and User Profiles by Enhancing Matrix Factorization with Tags." In: *RecSys '16: Poster Proceedings of the 10th ACM Conference on Recommender Systems*. 2016.

[DLZ17] Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Sequential User-Based Recurrent Neural Network Recommendations." In: *RecSys '17: Proceedings of the 11th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2017, pp. 152–160.

[DLZ18] Tim Donkers, **Benedikt Loepp**, and Jürgen Ziegler. "Explaining Recommendations by Means of User Reviews." In: *ExSS '18: Proceedings of the 1st Workshop on Explainable Smart Systems*. 2018.

[DS92] Ravi Dhar and Itamar Simonson. "The Effect of the Focus of Comparison on Consumer Preferences." In: *Journal of Marketing Research* 29.4 (1992), pp. 430–440.

[DSM10] Maunendra Sankar Desarkar, Sudeshna Sarkar, and Pabitra Mitra. "Aggregating Preference Graphs for Collaborative Rating Prediction." In: *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010, pp. 21–28.

[DSR15] Nofar Dali Betzalel, Bracha Shapira, and Lior Rokach. ""Please, Not Now!": A Model for Timing Recommendations." In: *RecSys '15: Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2015, pp. 297–300.

[DT13] Daria Dzyabura and Alexander Tuzhilin. "Not by Search Alone: How Recommendations Complement Search Results." In: *RecSys '13: Proceedings of the 7th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2013, pp. 371–374.

[DZ20] Tim Donkers and Jürgen Ziegler. "Leveraging Arguments in User Reviews for Generating and Explaining Recommendations." In: *Datenbank-Spektrum* 20 (2020), pp. 181–187.

[EHS17] Felix Eggers, John R. Hauser, and Matthew Selove. *Scale Matters: How Craft in Conjoint Analysis Affects Price and Positioning Strategies*. Tech. rep. MIT Sloan School of Management, 2017.

[EK19] Michael D. Ekstrand and Joseph A. Konstan. *Recommender Systems Notation - Proposed Common Notation for Teaching and Research*. Tech. rep. Boise State University & University of Minnesota, 2019.

[Eks*14] Michael D. Ekstrand, F. Maxwell Harper, Martijn C. Willemsen, and Joseph A. Konstan. "User Perception of Differences in Recommender Algorithms." In: *RecSys '14: Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2014, pp. 161–168.

[Eks*15] Michael D. Ekstrand, Daniel Kluver, F. Maxwell Harper, and Joseph A. Konstan. "Letting Users Choose Recommender Algorithms: An Experimental Study." In: *RecSys '15: Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2015, pp. 11–18.

[Ela*15] Mehdi Elahi, Mouzhi Ge, Francesco Ricci, Shlomo Berkovsky, and Massimo David. "Interaction Design in a Mobile Food Recommender System." In: *IntRS '15: Proceedings of the 2nd Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*. 2015, pp. 49–52.

[Ela14] Mehdi Elahi. "Empirical Evaluation of Active Learning Strategies in Collaborative Filtering." Doctoral dissertation. Bozen-Bolzano, Italy: Free University of Bozen-Bolzano, 2014.

[ERK11] Michael D. Ekstrand, John Riedl, and Joseph A. Konstan. "Collaborative Filtering Recommender Systems." In: *Foundations & Trends in Human-Computer Interaction* 4.2 (2011), pp. 175–243.

[ERR14] Mehdi Elahi, Francesco Ricci, and Neil Rubens. "Active Learning Strategies for Rating Elicitation in Collaborative Filtering: A System-Wide Perspective." In: *ACM Transactions on Intelligent Systems and Technology* 5.1 (2014), 13:1–13:33.

[ERR16] Mehdi Elahi, Francesco Ricci, and Neil Rubens. "A Survey of Active Learning in Collaborative Filtering Recommender Systems." In: *Computer Science Review* 20 (2016), pp. 29–50.

[Fan*19] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. "Deep Social Collaborative Filtering." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 305–313.

[Far*10] Siamak Faridani, Ephrat Bitton, Kimiko Ryokai, and Ken Goldberg. "Opinion Space: A Scalable Tool for Browsing Online Comments." In: *CHI '10: Proceedings of the 28th ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2010, pp. 1175–1184.

[FC14] Ignacio Fernández-Tobías and Iván Cantador. "Exploiting Social Tags in Matrix Factorization Models for Cross-Domain Collaborative Filtering." In: *CBRecSys '14: Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems*. 2014, pp. 34–41.

[Fer*19] Ignacio Fernández-Tobías, Iván Cantador, Paolo Tomeo, Vito Walter Anelli, and Tommaso Di Noia. "Addressing the User Cold Start with Cross-Domain Collaborative Filtering: Exploiting Item Metadata in Matrix Factorization." In: *User Modeling and User-Adapted Interaction* 29.2 (2019), pp. 443–486.

[Feu*17] Jan Feuerbach, **Benedikt Loepp**, Catalin-Mihai Barbu, and Jürgen Ziegler. "Enhancing an Interactive Recommendation System with Review-based Information Filtering." In: *IntRS '17: Proceedings of the 4th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*. 2017, pp. 2–9.

[FHK12] Andrey Feuerverger, Yu He, and Shashi Khatri. "Statistical Significance of the Netflix Challenge." In: *Statistical Science* 27.2 (2012), pp. 202–231.

[FMM77] George Elmer Forsythe, Michael A. Malcolm, and Cleve B. Moler. "Computer Methods for Mathematical Computations." In: Englewood Cliffs, NJ, USA: Prentice Hall, 1977. Chap. Least Squares and the Singular Value Decomposition.

[FO19] Evgeny Frolov and Ivan Oseledets. "HybridSVD: When Collaborative Information is Not Enough." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 331–339.

[Fra11] Massimo Franceschet. "PageRank: Standing on the Shoulders of Giants." In: *Communications of the ACM* 54.6 (2011), pp. 92–101.

[Fri*15] Arik Friedman, Bart P. Knijnenburg, Kris Vanhecke, Luc Martens, and Shlomo Berkovsky. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Privacy Aspects of Recommender Systems, pp. 649–688.

[Fun06] Simon Funk. *Netflix Update: Try This at Home*. 2006. URL: `http://www.sifter.org/~simon/journal/20061211.html` (visited on July 15, 2020).

[FZ11] Peter Forbes and Mu Zhu. "Content-Boosted Matrix Factorization for Recommender Systems: Experiments with Recipe Recommendation." In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 261–264.

[Gan*09] Emden R. Gansner, Yifan Hu, Stephen Kobourov, and Chris Volinsky. "Putting Recommendations on the Map: Visualizing Clusters and Relations." In: *RecSys '09: Proceedings of the 3rd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2009, pp. 345–348.

[Gan*10] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. "Learning Attribute-to-Feature Mappings for Cold-Start Recommendations." In: *ICDM '10: Proceedings of the 10th IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE, 2010, pp. 176–185.

[Gan*11] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. "MyMediaLite: A Free Recommender System Library." In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 305–308.

[GDJ10] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. "Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity." In: *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010, pp. 257–260.

[Ge*15] Mouzhi Ge, Mehdi Elahi, Ignacio Fernández-Tobías, Francesco Ricci, and David Massimo. "Using Tags and Latent Factors in a Food Recommender System." In: *DH '15: Proceedings of the 5th International Conference on Digital Health*. New York, NY, USA: ACM, 2015, pp. 105–112.

[Gel*13] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Boca Raton, FL, USA: CRC Press, 2013.

[Gem*11] Rainer Gemulla, Erik Nijkamp, Peter J. Haas, and Yannis Sismanis. "Large-Scale Matrix Factorization with Distributed Stochastic Gradient Descent." In: *KDD '11: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2011, pp. 69–77.

[Gem*15] Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Semantics-Aware Content-Based Recommender Systems, pp. 119–159.

[GGJ11] Fatih Gedikli, Mouzhi Ge, and Dietmar Jannach. "Explaining Online Recommendations Using Personalized Tag Clouds." In: *i-com – Journal of Interactive Media* 10.1 (2011), pp. 3–10.

[GH15] Carlos A. Gomez-Uribe and Neil Hunt. "The Netflix Recommender System: Algorithms, Business Value, and Innovation." In: *ACM Transactions on Management Information Systems* 6.4 (2015), 13:1–13:19.

[Gir*10] Andreas Girgensohn, Frank Shipman, Francine Chen, and Lynn Wilcox. "DocuBrowse: Faceted Searching, Browsing, and Recommendations in an Enterprise Context." In: *IUI '10: Proceedings of the 15th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2010, pp. 189–198.

[GKP11] Hector Garcia-Molina, Georgia Koutrika, and Aditya Parameswaran. "Information Seeking: Convergence of Search, Recommendations, and Advertising." In: *Communications of the ACM* 54.11 (2011), pp. 121–130.

[Gol*01] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. "Eigentaste: A Constant Time Collaborative Filtering Algorithm." In: *Information Retrieval* 4.2 (2001), pp. 133–151.

[Gol*92] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. "Using Collaborative Filtering to Weave an Information Tapestry." In: *Communications of the ACM* 35.12 (1992), pp. 61–70.

[Gra11] Mark P. Graus. "Understanding the Latent Features of Matrix Factorization Algorithms in Movie Recommender Systems." Master thesis. Eindhoven, The Netherlands: Eindhoven University of Technology, 2011.

[Gre*09] Stephen J. Green, Paul Lamere, Jeffrey Alexander, François Maillet, Susanna Kirk, Jessica Holt, Jackie Bourque, and Xiao-Wen Mak. "Generating Transparent, Steerable Recommendations from Textual Descriptions of Items." In: *RecSys '09: Proceedings of the 3rd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2009, pp. 281–284.

[Gre*10] Brynjar Gretarsson, John O'Donovan, Svetlin Bostandjiev, Christopher Hall, and Tobias Höllerer. "SmallWorlds: Visualizing Social Recommendations." In: *Computer Graphics Forum* 29.3 (2010), pp. 833–842.

[GS15] Asela Gunawardana and Guy Shani. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Evaluating Recommender Systems, pp. 265–308.

[GS78] Paul E. Green and V. Srinivasan. "Conjoint Analysis in Consumer Research: Issues and Outlook." In: *Journal of Consumer Research* 5.2 (1978), pp. 103–123.

[Gua*10] Ziyu Guan, Can Wang, Jiajun Bu, Chun Chen, Kun Yang, Deng Cai, and Xiaofei He. "Document Recommendation in Social Tagging Services." In: *WWW '10: Proceedings of the 19th International Conference on World Wide Web*. New York, NY, USA: ACM, 2010, pp. 391–400.

[Gup*13] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. "WTF: The Who to Follow Service at Twitter." In: *WWW '13: Proceedings of the 22nd International Conference on World Wide Web*. New York, NY, USA: ACM, 2013, pp. 505–514.

[Guy15] Ido Guy. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Social Recommender Systems, pp. 511–543.

[GW15] Mark P. Graus and Martijn C. Willemsen. "Improving the User Experience During Cold Start Through Choice-Based Preference Elicitation." In: *RecSys '15: Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2015, pp. 273–276.

[GW16] Mark P. Graus and Martijn C. Willemsen. "Can Trailers Help to Alleviate Popularity Bias in Choice-Based Preference Elicitation?" In: *IntRS '16: Proceedings of the 3rd Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*. 2016, pp. 22–27.

[Har*15] F. Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren G. Terveen. "Putting Users in Control of Their Recommendations." In: *RecSys '15: Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2015, pp. 3–10.

[Har*19] Jaron Harambam, Dimitrios Bountouridis, Mykola Makhortykh, and Joris van Hoboken. "Designing for the Better by Taking Users into Account: A Qualitative Evaluation of User Control Mechanisms in (News)

Recommender Systems." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 69–77.

[HBO10] Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky. "A Tour Through the Visualization Zoo." In: *ACM Queue* 53.6 (2010), pp. 59–67.

[Hea09] Marti A. Hearst. *Search User Interfaces*. Cambridge, UK: Cambridge University Press, 2009.

[Her*14] Katja Herrmanny, Sandra Schering, Ralf Berger, **Benedikt Loepp**, Timo Günter, Tim Hussein, and Jürgen Ziegler. "MyMovieMixer: Ein hybrider Recommender mit visuellem Bedienkonzept." In: *Mensch & Computer 2014 – Tagungsband*. Berlin, Germany: De Gruyter Oldenbourg, 2014, pp. 45–54.

[Hid*16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. "Session-Based Recommendations with Recurrent Neural Networks." In: *ICLR '16: Proceedings of the 4th International Conference on Learning Representations*. 2016.

[Hie*16] Patrick Hiesel, Wolfgang Wörndl, Matthias Braunhofer, and Daniel Herzog. "A User Interface Concept for Context-Aware Recommender Systems." In: *Mensch & Computer 2016 – Tagungsband*. Gesellschaft für Informatik, 2016.

[HKR00] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. "Explaining Collaborative Filtering Recommendations." In: *CSCW '00: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM, 2000, pp. 241–250.

[HKV08] Yifan Hu, Yehuda Koren, and Chris Volinsky. "Collaborative Filtering for Implicit Feedback Datasets." In: *ICDM '08: Proceedings of the 8th IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE, 2008, pp. 263–272.

[HM10] Tim Hussein and Daniel Münter. "Automated Generation of a Faceted Navigation Interface Using Semantic Models." In: *SEMAIS '10: Proceedings of 1st Workshop on Semantic Models for Adaptive Interactive Systems*. 2010.

[HMB14] Negar Hariri, Bamshad Mobasher, and Robin Burke. "Context Adaptation in Interactive Recommender Systems." In: *RecSys '14: Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2014, pp. 41–48.

[HN10] Tim Hussein and Sebastian Neuhaus. "Explanation of Spreading Activation Based Recommendations." In: *SEMAIS '10: Proceedings of 1st Workshop on Semantic Models for Adaptive Interactive Systems*. 2010.

[HNM16] Rubén Huertas-García, Ana Nuñez-Carballosa, and Paloma Miravitlles. "Statistical and Cognitive Optimization of Experimental Designs in Conjoint Analysis." In: *European Journal of Management and Business Economics* 25.3 (2016), pp. 142–149.

[Hot*06] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. "Information Retrieval in Folksonomies: Search and Ranking." In: *ESWC '06: Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications*. Berlin, Germany: Springer, 2006, pp. 411–426.

[Hou*19] Yunfeng Hou, Ning Yang, Yi Wu, and S. Yu Philip. "Explainable Recommendation with Fusion of Aspect Information." In: *World Wide Web* 22.1 (2019), pp. 221–240.

[HP09] Rong Hu and Pearl Pu. "A Comparative User Study on Rating vs. Personality Quiz Based Preference Elicitation Methods." In: *IUI '09: Proceedings of the 14th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2009, pp. 367–372.

[HPV16] Chen He, Denis Parra, and Katrien Verbert. "Interactive Recommender Systems: A Survey of the State of the Art and Future Research Challenges and Opportunities." In: *Expert Systems with Applications* 56 (2016), pp. 9–27.

[HS13] Eoin Hurrell and Alan F. Smeaton. "A Conversational Collaborative Filtering Approach to Recommendation." In: *Advances in Visual Informatics*. Ed. by Halimah Badioze Zaman, Peter Robinson, Patrick Olivier, Timothy K. Shih, and Sergio Velastin. Vol. 8237. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2013, pp. 13–24.

[HSM16] Katherine Hamilton, Shin-I Shih, and Susan Mohammed. "The Development and Validation of the Rational and Intuitive Decision Styles Scale." In: *Journal of Personality Assessment* 98.5 (2016), pp. 523–535.

[HT00] Gerald Häubl and Valerie Trifts. "Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids." In: *Marketing Science* 19.1 (2000), pp. 4–21.

[HT12] Balázs Hidasi and Domonkos Tikk. "Fast ALS-Based Tensor Factorization for Context-Aware Recommendation from Implicit Feedback." In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Peter

A. Flach, Tijl Bie, and Nello Cristianini. Vol. 7524. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2012, pp. 67–82.

[Hua*12] Jeff Huang, Oren Etzioni, Luke Zettlemoyer, Kevin Clark, and Christian Lee. "RevMiner: An Extractive Interface for Navigating Reviews on a Smartphone." In: *UIST '12: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2012, pp. 3–12.

[Hub05] Joel Huber. *Conjoint Analysis: How We Got Here and Where We Are (An Update)*. Tech. rep. Duke University & Sawtooth Software, Inc., 2005.

[Hus*14] Tim Hussein, Timm Linder, Werner Gaulke, and Jürgen Ziegler. "Hybreed: A Software Framework for Developing Context-Aware Hybrid Recommender Systems." In: *User Modeling and User-Adapted Interaction* 24.1-2 (2014), pp. 121–174.

[Iaq*08] Leo Iaquinta, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino. "Introducing Serendipity in a Content-Based Recommender System." In: *HIS '08: Proceedings of the 8th International Conference on Hybrid Intelligent Systems*. Washington, DC, USA: IEEE, 2008, pp. 168–173.

[Jam*15] Anthony Jameson, Martijn C. Willemsen, Alexander Felfernig, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Li Chen. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Human Decision Making and Recommender Systems, pp. 611–648.

[Jan*10] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge, UK: Cambridge University Press, 2010.

[JBB11] Nicolas Jones, Armelle Brun, and Anne Boyer. "Comparisons Instead of Ratings: Towards More Stable Preferences." In: *WI-IAT '11: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*. Washington, DC, USA: IEEE, 2011, pp. 451–456.

[JE09] Mohsen Jamali and Martin Ester. "Using a Trust Network to Improve Top-N Recommendation." In: *RecSys '09: Proceedings of the 3rd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2009, pp. 181–188.

[Jin*16] Yucheng Jin, Karsten Seipp, Erik Duval, and Katrien Verbert. "Go With the Flow: Effects of Transparency and User Control on Targeted Advertising Using Flow Charts." In: *AVI '16: Proceedings of the 13th International Conference on Advanced Visual Interfaces*. New York, NY, USA: ACM, 2016, pp. 68–75.

[JJ17] Michael Jugovac and Dietmar Jannach. "Interacting with Recommenders – Overview and Research Directions." In: *ACM Transactions on Interactive Intelligent Systems* 7.3 (2017), 10:1–10:46.

[JLJ15a] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. "Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation." In: *i-com – Journal of Interactive Media* 14.1 (2015), pp. 29–39.

[JLJ15b] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. "Item Familiarity Effects in User-Centric Evaluations of Recommender Systems." In: *RecSys '15: Poster Proceedings of the 9th ACM Conference on Recommender Systems*. 2015.

[JTV18] Yucheng Jin, Nava Tintarev, and Katrien Verbert. "Effects of Personal Characteristics on Music Recommender Systems with Different Levels of Controllability." In: *RecSys '18: Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2018, pp. 13–21.

[JWK14] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. "Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback." In: *ACM Transactions on Interactive Intelligent Systems* 4.2 (2014), 8:1–8:26.

[Kah11] Daniel Kahneman. *Thinking, Fast and Slow*. New York, NY, USA: Farrar, Straus and Giroux, 2011.

[Kam*09] Yvonne Kammerer, Rowan Nairn, Peter Pirolli, and Ed H. Chi. "Signpost from the Masses: Learning Effects in an Exploratory Social Tag Search Browser." In: *CHI '09: Proceedings of the 27th ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009, pp. 625–634.

[Kan*16] Antti Kangasrääsiö, Yi Chen, Dorota Głowacka, and Samuel Kaski. "Interactive Modeling of Concept Drift and Errors in Relevance Feedback." In: *UMAP '16: Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM, 2016, pp. 185–193.

[Kar*10] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. "Multiverse Recommendation: N-Dimensional Tensor Factorization for Context-Aware Collaborative Filtering." In: *RecSys '10:*

Proceedings of the 4th ACM Conference on Recommender Systems. New York, NY, USA: ACM, 2010, pp. 79–86.

[Kar*12] Rasoul Karimi, Christoph Freudenthaler, Alexandros Nanopoulos, and Lars Schmidt-Thieme. "Exploiting the Characteristics of Matrix Factorization for Active Learning in Recommender Systems." In: *RecSys '12: Proceedings of the 6th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2012, pp. 317–320.

[KB15a] Yehuda Koren and Robert M. Bell. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Advances in Collaborative Filtering, pp. 77–118.

[KB15b] Branislav Kveton and Shlomo Berkovsky. "Minimal Interaction Search in Recommender Systems." In: *IUI '15: Proceedings of the 20th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2015, pp. 236–246.

[KBV09] Yehuda Koren, Robert M. Bell, and Chris Volinsky. "Matrix Factorization Techniques for Recommender Systems." In: *IEEE Computer* 42.8 (2009), pp. 30–37.

[Ker*08] Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, eds. *Information Visualization: Human-Centered Issues and Perspectives*. Berlin, Germany: Springer, 2008.

[KFK10] Ruogu Kang, Wai-Tat Fu, and Thomas George Kannampallil. "Exploiting Knowledge-In-The-Head and Knowledge-In-The-Social-Web: Effects of Domain Expertise on Exploratory Search in Individual and Social Search Environments." In: *CHI '10: Proceedings of the 28th ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2010, pp. 393–402.

[KFK14] Kristof Kessler, Luanne Freund, and Richard Kopak. "Does the Perceived Usefulness of Search Facets Vary by Task Type?" In: *IIiX '14: Proceedings of the 5th Information Interaction in Context Symposium*. New York, NY, USA: ACM, 2014, pp. 267–270.

[Klu*12] Daniel Kluver, Tien T. Nguyen, Michael D. Ekstrand, Shilad Sen, and John Riedl. "How Many Bits Per Rating?" In: *RecSys '12: Proceedings of the 6th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2012, pp. 99–106.

[KLZ15] Johannes Kunkel, **Benedikt Loepp**, and Jürgen Ziegler. "3D-Visualisierung zur Eingabe von Präferenzen in Empfehlungssystemen." In: *Mensch & Computer 2015 – Tagungsband*. Berlin, Germany: De Gruyter Oldenbourg, 2015, pp. 123–132.

[KLZ17] Johannes Kunkel, **Benedikt Loepp**, and Jürgen Ziegler. "A 3D Item Space Visualization for Presenting and Manipulating User Preferences in Collaborative Filtering." In: *IUI '17: Proceedings of the 22nd International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2017, pp. 3–15.

[KLZ18a] Johannes Kunkel, **Benedikt Loepp**, and Jürgen Ziegler. "Ein Online-Spiel zur Benennung latenter Faktoren in Empfehlungssystemen." In: *Mensch & Computer 2018 – Tagungsband*. Gesellschaft für Informatik, 2018.

[KLZ18b] Johannes Kunkel, **Benedikt Loepp**, and Jürgen Ziegler. "Understanding Latent Factors Using a GWAP." In: *RecSys '18: Poster Proceedings of the 12th ACM Conference on Recommender Systems*. 2018.

[Kni*12] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. "Explaining the User Experience of Recommender Systems." In: *User Modeling and User-Adapted Interaction* 22.4-5 (2012), pp. 441–504.

[Kon18] Joseph A. Konstan. "From User Experience to Offline Metrics and Back Again – A Research Agenda." In: *REVEAL '18: Proceedings of the Workshop on Offline Evaluation for Recommender Systems*. 2018.

[Kor10] Yehuda Koren. "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering." In: *ACM Transactions on Knowledge Discovery from Data* 4.1 (2010), pp. 1–24.

[KR12] Joseph A. Konstan and John Riedl. "Recommender Systems: From Algorithms to User Experience." In: *User Modeling and User-Adapted Interaction* 22.1-2 (2012), pp. 101–123.

[KRG18] Saikishore Kalloori, Francesco Ricci, and Rosella Gennari. "Eliciting Pairwise Preferences in Recommender Systems." In: *RecSys '18: Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2018, pp. 329–337.

[KRT16] Saikishore Kalloori, Francesco Ricci, and Marko Tkalčič. "Pairwise Preferences Based Matrix Factorization and Nearest Neighbor Recommendation Techniques." In: *RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2016, pp. 143–146.

[KRW11] Bart P. Knijnenburg, Niels J. M. Reijmer, and Martijn C. Willemsen. "Each to His Own: How Different Users Call for Different Interaction Methods in Recommender Systems." In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 141–148.

[KS10] Mohammad Khoshneshin and W. Nick Street. "Collaborative Filtering via Euclidean Embedding." In: *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010, pp. 87–94.

[Kuh91] Carol C. Kuhlthau. "Inside the Search Process: Information Seeking from the User's Perspective." In: *Journal of the American Society for Information Science* 42.5 (1991), pp. 361–371.

[Kun*19a] Johannes Kunkel, Tim Donkers, Lisa Michael, Catalin-Mihai Barbu, and Jürgen Ziegler. "Let Me Explain: Impact of Personal and Impersonal Explanations on Trust in Recommender Systems." In: *CHI '19: Proceedings of the 37th ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2019.

[Kun*19b] Johannes Kunkel, **Benedikt Loepp**, Esther Dolff, and Jürgen Ziegler. "LittleMissFits: Ein Game-with-a-Purpose zur Evaluierung subjektiver Verständlichkeit von latenten Faktoren in Empfehlungssystemen." In: *Proceedings of the 2nd Gam-R Workshop – Gamification Reloaded*. Gesellschaft für Informatik, 2019.

[KW10] Bart P. Knijnenburg and Martijn C. Willemsen. "The Effect of Preference Elicitation Methods on the User Experience of a Recommender System." In: *CHI '10: Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2010, pp. 3457–3462.

[KW15] Bart P. Knijnenburg and Martijn C. Willemsen. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Evaluating Recommender Systems with User Experiments, pp. 309–352.

[KWB14] Bart P. Knijnenburg, Martijn C. Willemsen, and Ron Broeders. "Smart Sustainability through System Satisfaction: Tailored Preference Elicitation for Energy-Saving Recommenders." In: *AMCIS '14: Proceedings of the 20th Americas Conference on Information Systems*. 2014.

[KWG10] Martijn Kagie, Michiel van Wezel, and Patrick J. F. Groenen. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Berlin, Germany: Springer, 2010. Chap. Map Based Visualization of Product Catalogs, pp. 547–576.

[KWK11] Bart P. Knijnenburg, Martijn C. Willemsen, and Alfred Kobsa. "A Pragmatic Procedure to Support the User-Centric Evaluation of Recommender Systems." In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 321–324.

[KY05] Dohyun Kim and Bong-Jin Yum. "Collaborative Filtering Based on Iterative Principal Component Analysis." In: *Expert Systems with Applications* 28.4 (2005), pp. 823–830.

[KZ18] Farhan Khawar and Nevin L. Zhang. "Matrix Factorization Equals Efficient Co-Occurrence Representation." In: *RecSys '18: Poster Proceedings of the 12th ACM Conference on Recommender Systems*. 2018.

[KZ19] Timm Kleemann and Jürgen Ziegler. "Integration dialogbasierter Produktberater in Filtersysteme." In: *Mensch & Computer 2019 – Tagungsband*. New York, NY, USA: ACM, 2019, pp. 67–77.

[KZL08] Jonathan Koren, Yi Zhang, and Xue Liu. "Personalized Interactive Faceted Search." In: *WWW '08: Proceedings of the 17th International Conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 477–486.

[LBZ16] **Benedikt Loepp**, Catalin-Mihai Barbu, and Jürgen Ziegler. "Interactive Recommending: Framework, State of Research and Future Challenges." In: *EnCHIReS '16: Proceedings of the 1st Workshop on Engineering Computer-Human Interaction in Recommender Systems*. 2016, pp. 3–13.

[LDS18] Yichao Lu, Ruihai Dong, and Barry Smyth. "Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews." In: *WWW '18: Proceedings of the 2018 World Wide Web Conference*. Geneva, Switzerland: International World Wide Web Conference Committee, 2018, pp. 773–782.

[LFK08] Hangzai Luo, Jianping Fan, and Daniel A. Keim. "Personalized News Video Recommendation." In: *MM '08: Proceedings of the 16th ACM International Conference on Multimedia*. New York, NY, USA: ACM, 2008, pp. 1001–1002.

[LHS08] Bettina Laugwitz, Theo Held, and Martin Schrepp. "Construction and Evaluation of a User Experience Questionnaire." In: *HCI and Usability for Education and Work*. Ed. by Andreas Holzinger. Vol. 5298. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2008, pp. 63–76.

[LHZ13] **Benedikt Loepp**, Tim Hussein, and Jürgen Ziegler. "Interaktive Empfehlungsgenerierung mit Hilfe latenter Produktfaktoren." In: *Mensch & Computer 2013 – Tagungsband*. München, Germany: Oldenbourg Verlag, 2013, pp. 17–26.

[LHZ14] **Benedikt Loepp**, Tim Hussein, and Jürgen Ziegler. "Choice-Based Preference Elicitation for Collaborative Filtering Recommender Systems." In: *CHI '14: Proceedings of the 32nd ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2014, pp. 3085–3094.

[LHZ15a] **Benedikt Loepp**, Katja Herrmanny, and Jürgen Ziegler. "Blended Recommending: Integrating Interactive Information Filtering and Algorithmic Recommender Techniques." In: *CHI '15: Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2015, pp. 975–984.

[LHZ15b] **Benedikt Loepp**, Katja Herrmanny, and Jürgen Ziegler. "Merging Interactive Information Filtering and Recommender Algorithms – Model and Concept Demonstrator." In: *i-com – Journal of Interactive Media* 14.1 (2015), pp. 5–17.

[Li*10] Chengkai Li, Ning Yan, Senjuti B. Roy, Lekhendro Lisham, and Gautam Das. "Facetedpedia: Dynamic Generation of Query-dependent Faceted Interfaces for Wikipedia." In: *WWW '10: Proceedings of the 19th International Conference on World Wide Web*. New York, NY, USA: ACM, 2010, pp. 651–660.

[Liu*10] Nathan Nan Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. "Unifying Explicit and Implicit Feedback for Collaborative Filtering." In: *CIKM '10: Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2010, pp. 1445–1448.

[Liu*18] Yuhong Liu, Yue Han, Kirk Iserman, and Zhigang Jin. "Minimizing Required User Effort for Cold-Start Recommendation by Identifying the Most Important Latent Factors." In: *IEEE Access* 6 (2018).

[LJ14] Lukas Lerche and Dietmar Jannach. "Using Graded Implicit Feedback for Bayesian Personalized Ranking." In: *RecSys '14: Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2014, pp. 353–356.

[LKB19] Gal Lavee, Noam Koenigstein, and Oren Barkan. "When Actions Speak Louder than Clicks: A Combined Model of Purchase Probability and Long-Term Customer Satisfaction." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 287–295.

[Loe*18] **Benedikt Loepp**, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Impact of Item Consumption on Assessment of Recommendations in User Studies." In: *RecSys '18: Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2018, pp. 49–53.

[Loe*19a] **Benedikt Loepp**, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Impact of Consuming Suggested Items on the Assessment of Recommendations in User Studies on Recommender Systems." In: *IJCAI '19: Proceedings of the 28th International Joint Conference on Artificial Intelligence*. IJCAI Organization, 2019, pp. 6201–6205.

[Loe*19b] **Benedikt Loepp**, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. "Interactive Recommending with Tag-Enhanced Matrix Factorization (TagMF)." In: *International Journal of Human-Computer Studies* 121 (2019), pp. 21–41.

[LRS06] Jin Ha Lee, Allen Renear, and Linda C. Smith. "Known-Item Search: Variations on a Concept." In: *Proceedings of the American Society for Information Science and Technology* 43.1 (2006), pp. 1–17.

[LS99] Daniel D. Lee and H. Sebastian Seung. "Learning the Parts of Objects by Non-Negative Matrix Factorization." In: *Nature* 401.6755 (1999), pp. 788–791.

[LSY03] Greg Linden, Brent Smith, and Jeremy York. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering." In: *IEEE Internet Computing* 7.1 (2003), pp. 76–80.

[Luo*14] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. "An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems." In: *IEEE Transactions on Industrial Informatics* 10.2 (2014), pp. 1273–1284.

[LWG08] Qiudan Li, Chunheng Wang, and Guanggang Geng. "Improving Personalized Services in Mobile Commerce by a Novel Multicriteria Rating Approach." In: *WWW '08: Proceedings of the 17th International Conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 1235–1236.

[LZ14] **Benedikt Loepp** and Jürgen Ziegler. "Komplexe Präferenzprofile für intermodale Navigation." In: *Proceedings of the 3rd Workshop on Automotive HMI*. Berlin, Germany: De Gruyter Oldenbourg, 2014, pp. 191–198.

[LZ17a] **Benedikt Loepp** and Jürgen Ziegler. "Empirische Bedarfsanalyse zur intermodalen Navigation und dem Einsatz von Informationssystemen zur Förderung ihrer Attraktivität." In: *Innovative Produkte und Dienstleistungen in der Mobilität: Technische und betriebswirtschaftliche Aspekte*. Ed. by Heike Proff and Thomas Martin Fojcik. Wiesbaden, Germany: Springer Gabler, 2017, pp. 409–426.

[LZ17b] **Benedikt Loepp** and Jürgen Ziegler. "On User Awareness in Model-Based Collaborative Filtering Systems." In: *AWARE '17: Proceedings of the 1st Workshop on Awareness Interfaces and Interactions*. 2017.

[LZ18] **Benedikt Loepp** and Jürgen Ziegler. "Recommending Running Routes: Framework and Demonstrator." In: *ComplexRec '18: Proceedings of the 2nd Workshop on Recommendation in Complex Scenarios*. 2018.

[LZ19a] **Benedikt Loepp** and Jürgen Ziegler. "Measuring the Impact of Recommender Systems – A Position Paper on Item Consumption in User Studies." In: *ImpactRS '19: Proceedings of the 1st Workshop on Impact of Recommender Systems*. 2019.

[LZ19b] **Benedikt Loepp** and Jürgen Ziegler. "Towards Interactive Recommending in Model-Based Collaborative Filtering Systems." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 546–547.

[MA07] Paolo Massa and Paolo Avesani. "Trust-Aware Recommender Systems." In: *RecSys '07: Proceedings of the 1st ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2007, pp. 17–24.

[Man12] Marcelo G. Manzato. "Discovering Latent Factors from Movies Genres for Enhanced Recommendation." In: *RecSys '12: Proceedings of the 6th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2012, pp. 249–252.

[Mar*10] Leandro B. Marinho, Alexandros Nanopoulos, Lars Schmidt-Thieme, Robert Jäschke, Andreas Hotho, Gerd Stumme, and Panagiotis Symeonidis. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Berlin, Germany: Springer, 2010. Chap. Social Tagging Recommender Systems, pp. 615–644.

[Mar06] Gary Marchionini. "Exploratory Search: From Finding to Understanding." In: *Communications of the ACM* 49.4 (2006), pp. 41–46.

[Mar95] Gary Marchionini. *Information Seeking in Electronic Environments*. New York, NY, USA: Cambridge University Press, 1995.

[MBR12] Omar Moling, Linas Baltrunas, and Francesco Ricci. "Optimal Radio Channel Recommendations with Explicit and Implicit Feedback." In: *RecSys '12: Proceedings of the 6th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2012, pp. 75–82.

[McC*04] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. "On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems." In: *AH '04: Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. Berlin, Germany: Springer, 2004, pp. 279–304.

[MF12] Monika Mandl and Alexander Felfernig. "Improving the Performance of Unit Critiquing." In: *UMAP '12: Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization*. Berlin, Germany: Springer, 2012, pp. 176–187.

[MH16] Harmen Oppewal Meissner Martin and Joel Huber. "How Many Options? Behavioral Responses to Two Versus Five Alternatives per Choice." In: *Proceedings of the 19th Sawtooth Software Conference*. Sequim, WA, USA: Sawtooth Software, 2016, pp. 19–36.

[Mic*07] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. "Personalized Search on the World Wide Web." In: *The Adaptive Web. Methods and Strategies of Web Personalization*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Vol. 4321. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2007, pp. 195–230.

[Mil56] George A. Miller. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." In: *Psychological Review* 63.2 (1956), pp. 81–97.

[ML13] Julian McAuley and Jure Leskovec. "Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text." In: *RecSys '13: Proceedings of the 7th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2013, pp. 165–172.

[MMN02] Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. "Content-Boosted Collaborative Filtering for Improved Recommendations." In: *AAAI '02: Proceedings of the 18th National Conference on Artificial Intelligence*. Palo Alto, CA, USA: AAAI Press, 2002, pp. 187–192.

[Moi14] Afshin Moin. "A Unified Approach To Collaborative Data Visualization." In: *SAC '14: Proceedings of the 29th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2014, pp. 280–286.

[Moo20] E. Hastings Moore. "On the Reciprocal of the General Algebraic Matrix." In: *Bulletin of the American Mathematical Society* 26 (1920), pp. 394–395.

[MR09] Tariq Mahmood and Francesco Ricci. "Improving Recommender Systems with Adaptive Conversational Strategies." In: *HT '09: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*. New York, NY, USA: ACM, 2009, pp. 73–82.

[MRK06] Sean M. McNee, John Riedl, and Joseph A. Konstan. "Making Recommendations Better: An Analytic Model for Human-Recommender Interaction." In: *CHI '06: Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2006, pp. 1103–1108.

[MSS10] Kevin McCarthy, Yasser Salem, and Barry Smyth. "Experience-Based Critiquing: Reusing Critiquing Experiences to Improve Conversational Recommendation." In: *ICCBR '10: Proceedings of the 18th International Conference on Case-Based Reasoning*. Berlin, Germany: Springer, 2010, pp. 480–494.

[MT10] Morten Moshagen and Meinald T. Thielsch. "Facets of Visual Aesthetics." In: *International Journal of Human-Computer Studies* 68.10 (2010), pp. 689–709.

[Nei*14] Julia Neidhardt, Rainer Schuster, Leonhard Seyfang, and Hannes Werthner. "Eliciting the Users' Unknown Preferences." In: *RecSys '14: Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2014, pp. 309–312.

[Ném*13] Bottyán Németh, Gábor Takács, István Pilászy, and Domonkos Tikk. "Visualization of Movie Features in Collaborative Filtering." In: *SoMeT '13: Proceedings of the 12th IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques*. 2013, pp. 229–233.

[Ngu*13] Tien T. Nguyen, Daniel Kluver, Ting-Yu Wang, Pik-Mai Hui, Michael D. Ekstrand, Martijn C. Willemsen, and John Riedl. "Rating Support Interfaces to Improve User Experience and Recommender Accuracy." In: *RecSys '13: Proceedings of the 7th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2013, pp. 149–156.

[NH12] Maria Augusta S. N. Nunes and Rong Hu. "Personality-Based Recommender Systems: An Overview." In: *RecSys '12: Proceedings of the 6th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2012, pp. 5–6.

[NH14] Xi Niu and Bradley Hemminger. "Analyzing the Interaction Patterns in a Faceted Search Interface." In: *Journal of the Association for Information Science and Technology* 66.5 (2014), pp. 1030–1047.

[NKB14] Trung V. Nguyen, Alexandros Karatzoglou, and Linas Baltrunas. "Gaussian Process Factorization Machines for Context-Aware Recommendations." In: *SIGIR '14: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2014, pp. 63–72.

[NLF10] Theodora Nanou, George Lekakos, and Konstantinos Fouskas. "The Effects of Recommendations' Presentation on Persuasion and Satisfaction in a Movie Recommender System." In: *Multimedia Systems* 16.4-5 (2010), pp. 219–230.

[NLZ20] Sidra Naveed, **Benedikt Loepp**, and Jürgen Ziegler. "On the Use of Feature-based Collaborative Explanations: An Empirical Comparison of Explanation Styles." In: *ExUM '20: Proceedings of the International Workshop on Transparent Personalization Methods based on Heterogeneous Personal Data*. New York, NY, USA: ACM, 2020, pp. 226–232.

[Nob*12] Syavash Nobarany, Louise Oram, Vasanth Kumar Rajendran, Chi-Hsiang Chen, Joanna McGrenere, and Tamara Munzner. "The Design Space of Opinion Measurement Interfaces: Exploring Recall Support for Rating and Ranking." In: *CHI '12: Proceedings of the 30th ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2012, pp. 2035–2044.

[NP19] James Neve and Ivan Palomares. "Latent Factor Models and Aggregation Operators for Collaborative Filtering in Reciprocal Recommender Systems." In: *RecSys '19: Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2019, pp. 219–227.

[NR13] Tien T. Nguyen and John Riedl. "Predicting Users' Preference from Tag Relevance." In: *UMAP '13: Proceedings of the 21st International Conference on User Modeling, Adaptation and Personalization*. Berlin, Germany: Springer, 2013, pp. 274–280.

[NV14] Sayooran Nagulendra and Julita Vassileva. "Understanding and Controlling the Filter Bubble Through Interactive Visualization: A User Study." In: *HT '14: Proceedings of the 25th ACM Conference on Hypertext and Social Media*. New York, NY, USA: ACM, 2014, pp. 107–115.

[NWB16] Shabnam Najafian, Wolfgang Wörndl, and Matthias Braunhofer. "Context-Aware User Interaction for Mobile Recommender Systems." In: *HAAPIE '16: Proceedings of the 1st International Workshop on Human Aspects in Adaptive and Personalized Interactive Environments*. 2016.

[NZ13] Jennifer Nguyen and Mu Zhu. "Content-Boosted Matrix Factorization Techniques for Recommender Systems." In: *Statistical Analysis and Data Mining* 6.4 (2013), pp. 286–301.

[ODo*08] John O'Donovan, Barry Smyth, Brynjar Gretarsson, Svetlin Bostandjiev, and Tobias Höllerer. "PeerChooser: Visual Interactive Recommendation." In: *CHI '08: Proceedings of the 26th ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008, pp. 1085–1088.

[OG08] Félix Hernández del Olmo and Elena Gaudioso. "Evaluation of Recommender Systems: A New Approach." In: *Expert Systems with Applications* 35.3 (2008), pp. 790–804.

[PA11] Denis Parra and Xavier Amatriain. "Walk the Talk: Analyzing the Relation Between Implicit and Explicit Feedback for Preference Elicitation." In: *UMAP '11: Proceedings of the 19th International Conference on User Modeling, Adaptation and Personalization*. Berlin, Germany: Springer, 2011, pp. 255–268.

[Pag*99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Tech. rep. Stanford InfoLab, 1999.

[Par*11] Denis Parra, Alexandros Karatzoglou, Xavier Amatriain, and Idil Yavuz. "Implicit Feedback Recommendation via Implicit-to-Explicit Ordinal Logistic Regression Mapping." In: *CARS '11: Proceedings of the 3rd Workshop on Context-Aware Recommender Systems*. 2011.

[Par11] Eli Pariser. *The Filter Bubble: What the Internet is Hiding From You*. New York, NY, USA: Penguin Press, 2011.

[PBT14] Denis Parra, Peter Brusilovsky, and Christoph Trattner. "See What You Want to See: Visual User-driven Approach for Hybrid Recommendation." In: *IUI '14: Proceedings of the 19th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2014, pp. 235–240.

[PC00] Carolyn C. Preston and Andrew M. Colman. "Optimal Number of Response Categories in Rating Scales: Reliability, Validity, Discriminating Power, and Respondent Preferences." In: *Acta Psychologica* 104.1 (2000), pp. 1–15.

[PC09] Pearl Pu and Li Chen. "User-Involved Preference Elicitation for Product Search and Recommender Systems." In: *AI Magazine* 29.4 (2009), pp. 93–103.

[PC15] Rohit Parimi and Doina Caragea. "Cross-Domain Matrix Factorization for Multiple Implicit-Feedback Domains." In: *Machine Learning, Optimization, and Big Data*. Ed. by Panos Pardalos, Mario Pavone, Giovanni Maria Farinella, and Vincenzo Cutello. Vol. 9432. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2015, pp. 80–92.

[PCH11] Pearl Pu, Li Chen, and Rong Hu. "A User-Centric Evaluation Framework for Recommender Systems." In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 157–164.

[PCH12] Pearl Pu, Li Chen, and Rong Hu. "Evaluating Recommender Systems from the User's Perspective: Survey of the State of the Art." In: *User Modeling and User-Adapted Interaction* 22.4-5 (2012), pp. 317–355.

[PDF07] Andrew M. Parker, Wändi Bruine De Bruin, and Baruch Fischhoff. "Maximizers versus Satisficers: Decision-making Styles, Competence, and Outcomes." In: *Judgment and Decision Making* 2.6 (2007), p. 342.

[Pen55] Roger Penrose. "A Generalized Inverse for Matrices." In: *Mathematical Proceedings of the Cambridge Philosophical Society* 51.3 (1955), pp. 406–413.

[Pom*12] Alina Pommeranz, Joost Broekens, Pascal Wiggers, Willem-Paul Brinkman, and Catholijn M. Jonker. "Designing Interfaces for Explicit Preference Elicitation: A User-Centered Investigation of Preference Representation and Elicitation Process." In: *User Modeling and User-Adapted Interaction* 22.4-5 (2012), pp. 357–397.

[PS11] Jeffrey R. Parker and Rom Y. Schrift. "Rejectable Choice Sets: How Seemingly Irrelevant No-Choice Options Affect Consumer Decision Processes." In: *Journal of Marketing Research* 48.5 (2011), pp. 840–854.

[PT09] István Pilászy and Domonkos Tikk. "Recommending New Movies: Even a Few Ratings Are More Valuable Than Metadata." In: *RecSys '09: Proceedings of the 3rd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2009, pp. 93–100.

[Pu*10] Pearl Pu, Boi Faltings, Li Chen, Jiyong Zhang, and Paolo Viappiani. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Berlin, Germany: Springer, 2010. Chap. Usability Guidelines for Product Recommenders Based on Example Critiquing Research, pp. 511–545.

[PZT10] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. "Fast ALS-Based Matrix Factorization for Explicit and Implicit Feedback Datasets." In: *RecSys '10: Proceedings of the 4th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010, pp. 71–78.

[QCJ18] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. "Sequence-Aware Recommender Systems." In: *ACM Computing Surveys* 51.4 (2018), 66:1–66:36.

[Ras*02] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. "Getting to Know You: Learning New User Preferences in Recommender Systems." In: *IUI '02: Proceedings of the 7th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2002, pp. 127–134.

[Rei*05] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. "Explaining Compound Critiques." In: *Artificial Intelligence Review* 24.2 (2005), pp. 199–220.

[Ren*09] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. "BPR: Bayesian Personalized Ranking from Implicit Feedback." In: *UAI '09: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. Corvallis, OR, USA: AUAI Press, 2009, pp. 452–461.

[RF14] Steffen Rendle and Christoph Freudenthaler. "Improving Pairwise Learning for Item Recommendation from Implicit Feedback." In: *WSDM '14: Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2014, pp. 273–282.

[RK12] Lior Rokach and Slava Kisilevich. "Initial Profile Generation in Recommender Systems Using Pairwise Comparison." In: *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews* 42.6 (2012), pp. 1854–1859.

[RKR08] Al Mamunur Rashid, George Karypis, and John Riedl. "Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach." In: *SIGKDD Explorations Newsletter* 10.2 (2008), pp. 90–100.

[Ros*16] Silvia Rossi, Francesco Barile, Davide Improta, and Luca Russo. "Towards a Collaborative Filtering Framework for Recommendation in Museums: From Preference Elicitation to Group's Visits." In: *DaMIS '16: Proceedings of the 2016 International Workshop on Data Mining on IoT Systems*. 2016, pp. 431–436.

[Ros*17] Silvia Rossi, Francesco Barile, Sergio Di Martino, and Davide Improta. "A Comparison of Two Preference Elicitation Approaches for Museum Recommendations." In: *Concurrency and Computation: Practice and Experience* 29.11 (2017).

[RRS15a] Francesco Ricci, Lior Rokach, and Bracha Shapira, eds. *Recommender Systems Handbook*. Berlin, Germany: Springer US, 2015.

[RRS15b] Francesco Ricci, Lior Rokach, and Bracha Shapira. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Recommender Systems: Introduction and Challenges, pp. 1–34.

[RS08] Steffen Rendle and Lars Schmidt-Thieme. "Online-Updating Regularized Kernel Matrix Factorization Models for Large-Scale Recommender Systems." In: *RecSys '08: Proceedings of the 2nd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2008, pp. 251–258.

[RSZ13] Marco Rossetti, Fabio Stella, and Markus Zanker. "Towards Explaining Latent Factors with Topic Models in Collaborative Recommender Systems." In: *DEXA '13: Proceedings of the 24th International Workshop on Database and Expert Systems Applications*. 2013, pp. 162–167.

[Rub*15] Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Active Learning in Recommender Systems, pp. 809–846.

[Rub17] Mark Rubin. "Do p Values Lose Their Meaning in Exploratory Analyses? It Depends How You Define the Familywise Error Rate." In: *Review of General Psychology* 21.3 (2017), pp. 269–275.

[Rud19] Cynthia Rudin. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead." In: *Nature Machine Intelligence* 1 (2019), pp. 206–215.

[Saa08] Thomas L. Saaty. "Decision Making with the Analytic Hierarchy Process." In: *International Journal of Services Sciences* 1.1 (2008), pp. 83–98.

[Sac06] Giovanni M. Sacco. "Dynamic Taxonomies and Guided Searches." In: *Journal of the American Society for Information Science and Technology* 57.6 (2006), pp. 792–796.

[Sar*00] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. "Application of Dimensionality Reduction in Recommender System – A Case Study." In: *WebKDD '00: Proceedings of the Workshop on Web Mining for E-Commerce*. 2000.

[Sar*02] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. "Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering." In: *ICCIT '02: Proceedings of the 5th International Conference on Computer and Information Technology*. 2002.

[SB11] Beverley A. Sparks and Victorica Browning. "The Impact of Online Reviews on Hotel Booking Intentions and Perception of Trust." In: *Tourism Management* 32.6 (2011), pp. 1310–1323.

[SB14a] Alan Said and Alejandro Bellogín. "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks." In: *RecSys '14: Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2014, pp. 129–136.

[SB14b] Alan Said and Alejandro Bellogín. "RiVal: A Toolkit to Foster Reproducibility in Recommender System Evaluation." In: *RecSys '14: Proceedings of the 8th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2014, pp. 371–372.

[SB18] Alan Said and Alejandro Bellogín. "Coherence and Inconsistencies in Rating Behavior: Estimating the Magic Barrier of Recommender Systems." In: *User Modeling and User-Adapted Interaction* 28.2 (2018), pp. 97–125.

[SB95] Susanne G. Scott and Reginald A. Bruce. "Decision-Making Style: The Development and Assessment of a New Measure." In: *Educational and Psychological Measurement* 55.5 (1995), pp. 818–831.

[SB97] Gerard Salton and Chris Buckley. "Improving Retrieval Performance by Relevance Feedback." In: *Readings in Information Retrieval*. San Francisco, CA, USA: Morgan Kaufmann, 1997, pp. 355–364.

[SC13] Amit Sharma and Dan Cosley. "Do Social Explanations Work? Studying and Modeling the Effects of Social Explanations in Recommender Systems." In: *WWW '13: Proceedings of the 22nd International Conference on World Wide Web*. New York, NY, USA: ACM, 2013, pp. 1133–1144.

[Sch*07] J. Ben Schafer, Dan Frankowski, Jonathan L. Herlocker, and Shilad Sen. "Collaborative Filtering Recommender Systems." In: *The Adaptive Web. Methods and Strategies of Web Personalization*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Vol. 4321. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2007, pp. 291–324.

[Sep*18] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. "Preference Elicitation as an Optimization Problem." In: *RecSys '18: Proceedings of the 12th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2018, pp. 172–180.

[Shn94] Ben Shneiderman. "Dynamic Queries for Visual Information Seeking." In: *IEEE Software* 11.6 (1994), pp. 70–77.

[SHO15] James Schaffer, Tobias Höllerer, and John O'Donovan. "Hypothetical Recommendation: A Study of Interactive Profile Manipulation Behavior for Recommender Systems." In: *FLAIRS '15: Proceedings of the 28th International Florida Artificial Intelligence Research Society Conference*. 2015.

[SHT17] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. "Construction of a Benchmark for the User Experience Questionnaire (UEQ)." In: *International Journal of Interactive Multimedia and Artificial Intelligence* 4.4 (2017), pp. 40–44.

[SK09] Xiaoyuan Su and Taghi M. Khoshgoftaar. "A Survey of Collaborative Filtering Techniques." In: *Advances in Artificial Intelligence* 2009 (2009), 4:1–4:19.

[SL17] Brent Smith and Greg Linden. "Two Decades of Recommender Systems at Amazon.com." In: *IEEE Internet Computing* 21.3 (2017), pp. 12–18.

[SLH13] Yue Shi, Martha Larson, and Alan Hanjalic. "Mining Contextual Movie Similarity with Matrix Factorization for Context-Aware Recommendation." In: *ACM Transactions on Intelligent Systems and Technology* 4.1 (2013), 16:1–16:19.

[SM03] Barry Smyth and Lorraine McGinty. "An Analysis of Feedback Strategies in Conversational Recommenders." In: *AICS '03: Proceedings of the 14th Irish Artificial Intelligence and Cognitive Science Conference*. 2003, pp. 211–216.

[SM18] Michael Steiner and Martin Meißner. "A User's Guide to the Galaxy of Conjoint Analysis and Compositional Preference Measurement." In: *ZFP – Journal of Research and Management* 40.2 (2018), pp. 3–25.

[SMH07] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. "Restricted Boltzmann Machines for Collaborative Filtering." In: *ICML '07: Proceedings of the 24th International Conference on Machine Learning*. New York, NY, USA: ACM, 2007, pp. 791–798.

[SNM08] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. "Tag Recommendations Based on Tensor Dimensionality Reduction." In: *RecSys '08: Proceedings of the 2nd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2008, pp. 43–50.

[SPG00] Pieter Jan Stappers, Gert Pasman, and Patrick J. F. Groenen. "Exploring Databases for Taste or Inspiration with Interactive Multi-Dimensional Scaling." In: *HFES '00: Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Santa Monica, CA, USA: The Human Factors Society of the USA, 2000, pp. 575–578.

[SS11] E. Isaac Sparling and Shilad Sen. "Rating: How Difficult is it?" In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 149–156.

[SSV16] Cecilia di Sciascio, Vedran Sabol, and Eduardo E. Veas. "Rank As You Go: User-Driven Exploration of Search Results." In: *IUI '16: Proceedings of the 21st International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2016, pp. 118–129.

[ST09] Giovanni Maria Sacco and Yannis Tzitzikas, eds. *Dynamic Taxonomies and Faceted Search – Theory, Practice, and Experience*. Berlin, Germany: Springer, 2009.

[Ste11] Harald Steck. "Item Popularity and Recommendation Accuracy." In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 125–132.

[Ste15] Harald Steck. "Gaussian Ranking by Matrix Factorization." In: *RecSys '15: Proceedings of the 9th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2015, pp. 115–122.

[SVR09] Shilad Sen, Jesse Vig, and John Riedl. "Tagommenders: Connecting Users to Items Through Tags." In: *WWW '09: Proceedings of the 18th International Conference on World Wide Web*. New York, NY, USA: ACM, 2009, pp. 671–680.

[Tak*08] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. "Investigation of Various Matrix Factorization Methods for Large Recommender Systems." In: *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*. New York, NY, USA: ACM, 2008.

[Tak*09] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. "Scalable Collaborative Filtering Approaches for Large Recommender Systems." In: *Journal of Machine Learning Research* 10 (2009), pp. 623–656.

[TAL14] Jiliang Tang, Salem Alelyani, and Huan Liu. "Feature Selection for Classification: A Review." In: *Data Classification: Algorithms and Applications* (2014), pp. 37–64.

[Tan*16] Liang Tang, Bo Long, Bee-Chung Chen, and Deepak Agarwal. "An Empirical Study on Recommendation with Multiple Types of Feedback." In: *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016, pp. 283–292.

[TB07] Michal Tvarožek and Mária Bieliková. "Personalized Faceted Navigation for Multimedia Collections." In: *SMAP '07: Proceedings of the 2nd International Workshop on Semantic Media Adaptation and Personalization*. Washington, DC, USA: IEEE, 2007, pp. 104–109.

[TC15] Marko Tkalcic and Li Chen. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Personality and Recommender Systems, pp. 715–739.

[TDG08] Jaime Teevan, Susan T. Dumais, and Zachary Gutt. "Challenges for Supporting Faceted Search in Large, Heterogeneous Corpora Like the Web." In: *HCIR '08: Proceedings of the 2nd Workshop on Human-Computer Interaction and Information Retrieval*. 2008, pp. 6–8.

[Tee*04] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. "The Perfect Search Engine is Not Enough: A Study of Orienteering Behavior in Directed Search." In: *CHI '04: Proceedings of the 22nd ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2004, pp. 415–422.

[Tek*16] Feben Teklemicael, Yong Zhang, Yongji Wu, Yanshen Yin, and Chunxiao Xing. "Toward Gamified Personality Acquisition in Travel Recommender Systems." In: *Human Centered Computing – HCC 2016*. Ed. by Qiaohong Zu and Bo Hu. Vol. 9567. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2016, pp. 375–385.

[Tib96] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso." In: *Journal of the Royal Statistical Society. Series B*. 58 (1996), pp. 267–288.

[Tir*14] Amit Tiroshi, Shlomo Berkovsky, Mohamed Ali Kaafar, David Vallet, and Tsvi Kuflik. "Graph-Based Recommendations: Make the Most Out of Social Data." In: *UMAP '14: Proceedings of the 22nd International Conference on User Modeling, Adaptation and Personalization*. Berlin, Germany: Springer, 2014, pp. 447–458.

[Tka*11] Marko Tkalčič, Andrej Košir, Štefan Dobravec, and Jurij Tasič. "Emotional Properties of Latent Factors in an Image Recommender System." In: *Elektrotehniški Vestnik* 78.4 (2011), pp. 177–180.

[TM15] Nava Tintarev and Judith Masthoff. "Recommender Systems Handbook." In: ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA, USA: Springer US, 2015. Chap. Explaining Recommendations: Design and Evaluation, pp. 353–382.

[TMS08] Karen H. L. Tso-Sutter, Leandro B. Marinho, and Lars Schmidt-Thieme. "Tag-Aware Recommender Systems by Fusion of Collaborative Filtering Algorithms." In: *SAC '08: Proceedings of the 2008 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2008, pp. 1995–1999.

[Tön19] Lennart Zacharias Tönnissen. "Auswahlbasierte Präferenzerhebung auf Basis von Deep Learning." Bachelor thesis. University of Duisburg-Essen, 2019.

[TRH12] Vinh Tuan Thai, Pierre-Yves Rouille, and Siegfried Handschuh. "Visual Abstraction and Ordering in Faceted Browsing of Text Collections." In: *ACM Transactions on Intelligent Systems and Technology* 3.2 (2012), 21:1–21:24.

[Tun09] Daniel Tunkelang. "Faceted Search." In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1.1 (2009), pp. 1–80.

[Tva*08] Michal Tvarožek, Michal Barla, György Frivolt, Marek Tomša, and Mária Bieliková. "Improving Semantic Search via Integrated Personalized Faceted and Visual Graph Navigation." In: *Theory and Practice of Computer Science – SOFSEM 2008*. Ed. by Viliam Geffert, Juhani Karhumäki, Alberto Bertoni, Bart Preneel, Pavol Návrat, and Mária Bieliková. Vol. 4910. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2008, pp. 778–789.

[TWK18] Taavi T. Taijala, Martijn C. Willemsen, and Joseph A. Konstan. "MovieExplorer: Building an Interactive Exploration Tool from Ratings and Latent Taste Spaces." In: *SAC '18: Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2018, pp. 1383–1392.

[UK03] James Uther and Judy Kay. "VlUM, a Web-Based Visualisation of Large User Models." In: *User Modeling*. Ed. by Peter Brusilovsky, Albert Corbett, and Fiorella de Rosis. Vol. 2702. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2003, pp. 198–202.

[VC11] Saúl Vargas and Pablo Castells. "Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems." In: *RecSys '11: Proceedings of the 5th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2011, pp. 109–116.

[VFP06] Paolo Viappiani, Boi Faltings, and Pearl Pu. "Preference-Based Search Using Example-Critiquing with Suggestions." In: *Journal of Artificial Intelligence Research* 27.1 (2006), pp. 465–503.

[Voi*12] Martin Voigt, Artur Werstler, Jan Polowinski, and Klaus Meißner. "Weighted Faceted Browsing for Characteristics-Based Visualization Selection Through End Users." In: *EICS '12: Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. New York, NY, USA: ACM, 2012, pp. 151–156.

[VPB16] Katrien Verbert, Denis Parra, and Peter Brusilovsky. "Agents vs. Users: Visual Recommendation of Research Talks with Multiple Dimension of Relevance." In: *ACM Transactions on Interactive Intelligent Systems* 6.2 (2016), 11:1–11:42.

[VPF08] Paolo Viappiani, Pearl Pu, and Boi Faltings. "Preference-Based Search with Adaptive Recommendations." In: *AI Communications* 21.2-3 (2008), pp. 155–175.

[VS12] Michail Vlachos and Daniel Svonava. "Graph Embeddings for Movie Visualization and Recommendation." In: *Joint Proceedings of the 1st International Workshop on Recommendation Technologies for Lifestyle Change*

(LIFESTYLE '12) and the 1st International Workshop on Interfaces for Recommender Systems (InterfaceRS '12). 2012, pp. 56–59.

[VS13] Michail Vlachos and Daniel Svonava. "Recommendation and Visualization of Similar Movies Using Minimum Spanning Dendrograms." In: *Information Visualization* 12.1 (2013), pp. 85–101.

[VSR09] Jesse Vig, Shilad Sen, and John Riedl. "Tagsplanations: Explaining Recommendations Using Tags." In: *IUI '09: Proceedings of the 14th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2009, pp. 47–56.

[VSR10] Jesse Vig, Shilad Sen, and John Riedl. *Computing the Tag Genome*. Tech. rep. University of Minnesota, 2010.

[VSR11] Jesse Vig, Shilad Sen, and John Riedl. "Navigating the Tag Genome." In: *IUI '11: Proceedings of the 16th International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2011, pp. 93–102.

[VSR12] Jesse Vig, Shilad Sen, and John Riedl. "The Tag Genome: Encoding Community Knowledge to Support Novel Interaction." In: *ACM Transactions on Interactive Intelligent Systems* 2.3 (2012), 13:1–13:44.

[Weg*18] Kodzo Wegba, Aidong Lu, Yuemeng Li, and Wencheng Wang. "Interactive Storytelling for Movie Recommendation Through Latent Semantic Analysis." In: *IUI '18: Proceedings of the 23rd International Conference on Intelligent User Interfaces*. New York, NY, USA: ACM, 2018, pp. 521–533.

[Wei*13] Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Xiaoyu Fu, and Boqin Feng. "A Survey of Faceted Search." In: *Journal of Web Engineering* 12.1-2 (2013), pp. 41–64.

[WGK16] Martijn C. Willemsen, Mark P. Graus, and Bart P. Knijnenburg. "Understanding the Role of Latent Feature Diversification on Choice Difficulty and Satisfaction." In: *User Modeling and User-Adapted Interaction* 26.4 (2016), pp. 347–389.

[WKB05] Ryen W. White, Bill Kules, and Ben Bederson. "Exploratory Search Interfaces: Categorization, Clustering and Beyond." In: *SIGIR Forum* 39.2 (2005), pp. 52–56.

[Won*11] David Wong, Siamak Faridani, Ephrat Bitton, Björn Hartmann, and Ken Goldberg. "The Diversity Donut: Enabling Participant Control over the Diversity of Recommended Responses." In: *CHI '11: Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011, pp. 1471–1476.

[WS12] Jörg Waitelonis and Harald Sack. "Towards Exploratory Video Search Using Linked Data." In: *Multimedia Tools and Applications* 59.2 (2012), pp. 645–672.

[WSL19] Ronald L. Wasserstein, Allen L. Schirm, and Nicole A. Lazar. "Moving to a World Beyond "p<.05"." In: *The American Statistician* 73.sup1 (2019), pp. 1–19.

[WV14] Wesley Waldner and Julita Vassileva. "A Visualization Interface for Twitter Timeline Activity." In: *IntRS '14: Proceedings of the 1st Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*. 2014, pp. 45–52.

[WWY13] Jason Weston, Ron J. Weiss, and Hector Yee. "Nonlinear Latent Factorization by Embedding Multiple User Interests." In: *RecSys '13: Proceedings of the 7th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2013, pp. 65–68.

[WWY15] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. "Collaborative Deep Learning for Recommender Systems." In: *KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2015, pp. 1235–1244.

[XB07] Bo Xiao and Izak Benbasat. "E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact." In: *MIS Quarterly* 31.1 (2007), pp. 137–209.

[Xie*18] Haoran Xie, Debby D. Wang, Yanghui Rao, Tak-Lam Wong, Lau Y. K. Raymond, Li Chen, and Fu Lee Wang. "Incorporating User Experience Into Critiquing-Based Recommender Systems: A Collaborative Approach Based on Compound Critiquing." In: *International Journal of Machine Learning and Cybernetics* 9.5 (2018), pp. 837–852.

[Yee*03] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti A. Hearst. "Faceted Metadata for Image Search and Browsing." In: *CHI '03: Proceedings of the 21st ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2003, pp. 401–408.

[YWY12] Li'ang Yin, Yongqiang Wang, and Yong Yu. "Collaborative Filtering via Temporal Euclidean Embedding." In: *Web Technologies and Applications*. Ed. by Quan Z. Sheng, Guoren Wang, Christian S. Jensen, and Guandong Xu. Vol. 7235. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2012, pp. 513–520.

[Zad65] Lotfi A. Zadeh. "Fuzzy Sets." In: *Information and Control* 8.3 (1965), pp. 338–353.

[Zha*06] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. "Learning from Incomplete Ratings Using Non-negative Matrix Factorization." In: *SDM '06: Proceedings of the 6th SIAM International Conference on Data Mining*. Philadelphia, PA, USA: SIAM, 2006, pp. 549–553.

[Zha*14] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. "Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis." In: *SIGIR '14: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2014, pp. 83–92.

[Zha*15] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H. Chi. "Improving User Topic Interest Profiles by Behavior Factorization." In: *WWW '15: Proceedings of the 24th International Conference on World Wide Web*. New York, NY, USA: ACM, 2015, pp. 1406–1416.

[Zha*19] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. "Deep Learning Based Recommender System: A Survey and New Perspectives." In: *ACM Computing Surveys* 52.1 (2019), 5:1–5:38.

[Zho*08] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. "Large-Scale Parallel Collaborative Filtering for the Netflix Prize." In: *Algorithmic Aspects in Information and Management*. Berlin, Germany: Springer, 2008, pp. 337–348.

[ZI13] Qianru Zheng and Horace H. S. Ip. "Effectiveness of the Data Generated on Different Time in Latent Factor Model." In: *RecSys '13: Proceedings of the 7th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2013, pp. 327–330.

[ZJP08] Jiyong Zhang, Nicolas Jones, and Pearl Pu. "A Visual Interface for Critiquing-Based Recommender Systems." In: *EC '08: Proceedings of the 9th ACM Conference on Electronic Commerce*. New York, NY, USA: ACM, 2008, pp. 230–239.

[ZL20] Jürgen Ziegler and **Benedikt Loepp**. "Empfehlungssysteme." In: *Handbuch Digitale Wirtschaft*. Ed. by Tobias Kollmann. Wiesbaden, Germany: Springer Gabler, 2020, pp. 717–741.

[ZLY09] Yi Zhen, Wu-Jun Li, and Dit-Yan Yeung. "TagiCoFi: Tag Informed Collaborative Filtering." In: *RecSys '09: Proceedings of the 3rd ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2009, pp. 69–76.

[ZYZ11] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. "Functional Matrix Factorizations for Cold-Start Recommendation." In: *SIGIR '11: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2011, pp. 315–324.

[ZZW13] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. "Interactive Collaborative Filtering." In: *CIKM '13: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2013, pp. 1411–1420.

List of figures

List of tables

List of listings

List of equations

Screenshots

In this section of the appendix, screenshots are shown that belong to the prototype systems we have developed for and used in the user experiments reported in this thesis, or to the integrated recommendation platform we have used for the case studies.

Experiment on choice-based preference elicitation

In the user experiment on choice-based preference elicitation (see Section 4.3), we used the prototype system that is shown in Figure A.1 and A.2.

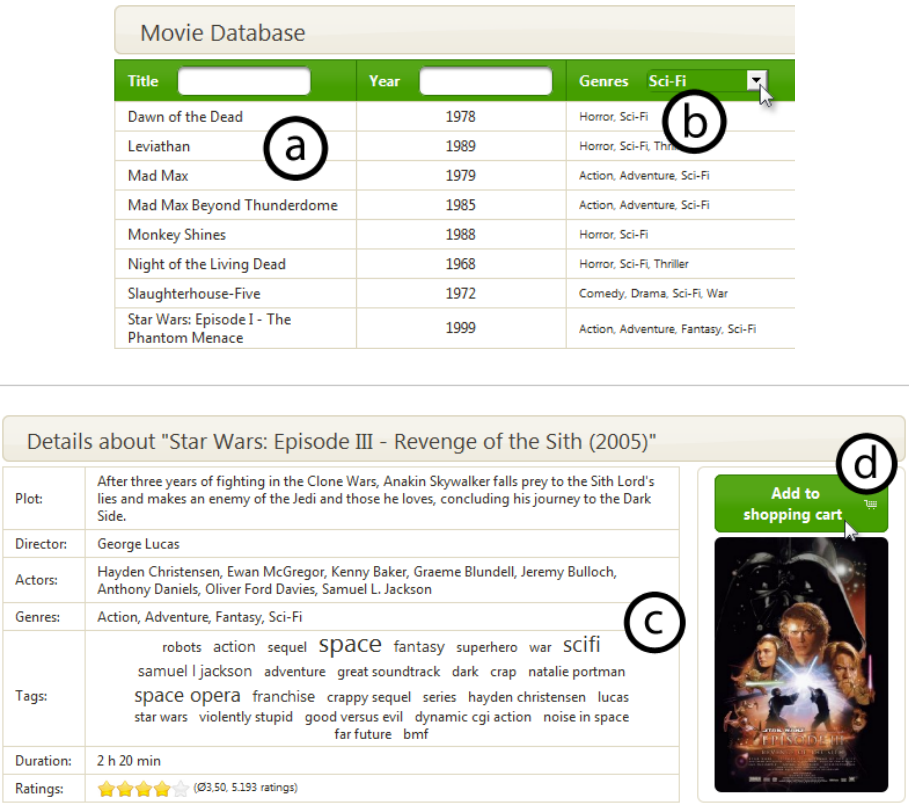

Figure A.1 Screenshots of the manual exploration interface: The upper image shows a list of movies (a). In the table head, search and filtering mechanisms are provided (b). By clicking on a title, users can proceed to the corresponding detail page. The lower image shows such a page, including a plot description and metadata (c). Most of the elements are hyperlinks that may be used to navigate back to the item list, but with a filter applied to show only movies, for example, from a specific genre or by a certain director. For the purpose of the experiment, items can be added to a shopping cart (d).

Figure A.2 Screenshot of the interactive dialog that implements the choice-based preference elicitation method: Two sets of items that represent a dimension of the underlying latent factor model are shown. As described in Section 4.2, the items differ strongly with respect to the values of this factor: While the set on the left-hand side contains serious, rather dark movies (a), the set on the right-hand side contains animated movies and comedies (b).

For each movie, title and poster are shown. Clicking on a title opens a dialog with further metadata. In addition, the tag clouds below provide a description of the movies of the respective set (c). Users can express their preferences by choosing one of the sets, or use the "don't care" option if they cannot decide or do not know the movies (d).

Experiment on content-boosted matrix factorization I

In the first user experiment on our content-boosted matrix factorization method and its application possibilities (see Section 6.3.1), we used the prototype system shown in Figure A.3.

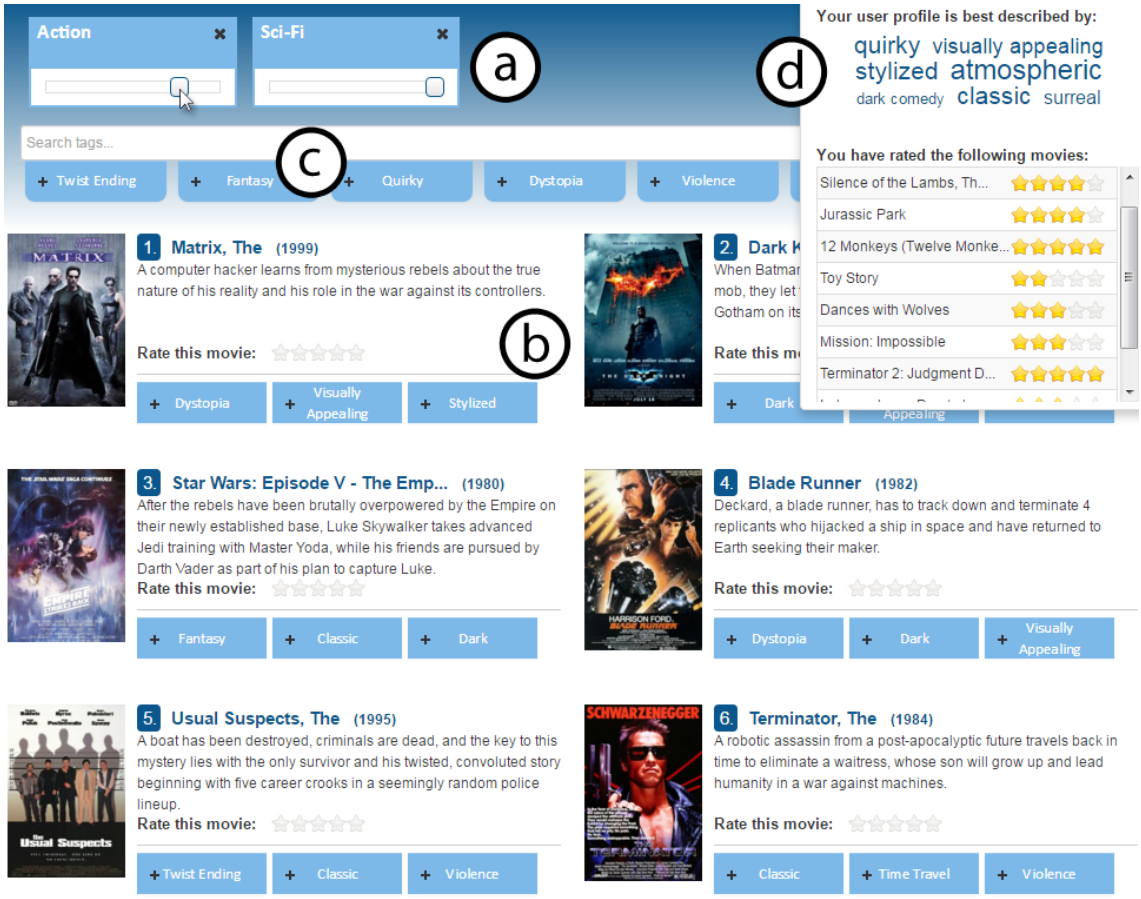

Figure A.3 Screenshot of the variant of the prototype system with a recommender based on content-boosted matrix factorization: At the top, selected tags are shown (a). Users can adjust their weights using the attached sliders, which is reflected back into the weighting vector and influences the recommendations as described in Section 6.2.2.

The top 10 recommendations are shown below (b). For each movie, title, poster and metadata are presented. To refine their profile, users may rate these movies or search manually for other movies to rate them. Next to each movie, the 3 most relevant tags are shown, which may also be selected and weighted. In addition, users can search for tags, supported by autocompletion, or get inspiration from suggested tags (c). For this, the 7 most popular tags are shown initially. As soon as weights are applied to selected tags, tags that are similar in terms of item-tag vectors are shown instead.

The dialog in the top-right corner presents users with a tag cloud that describes their representation within the underlying model as described in Section 6.2.4 (d).

Experiment on content-boosted matrix factorization II

In the second user experiment on our content-boosted matrix factorization method and its application possibilities (see Section 6.3.2), we used the prototype system shown in Figure A.4.

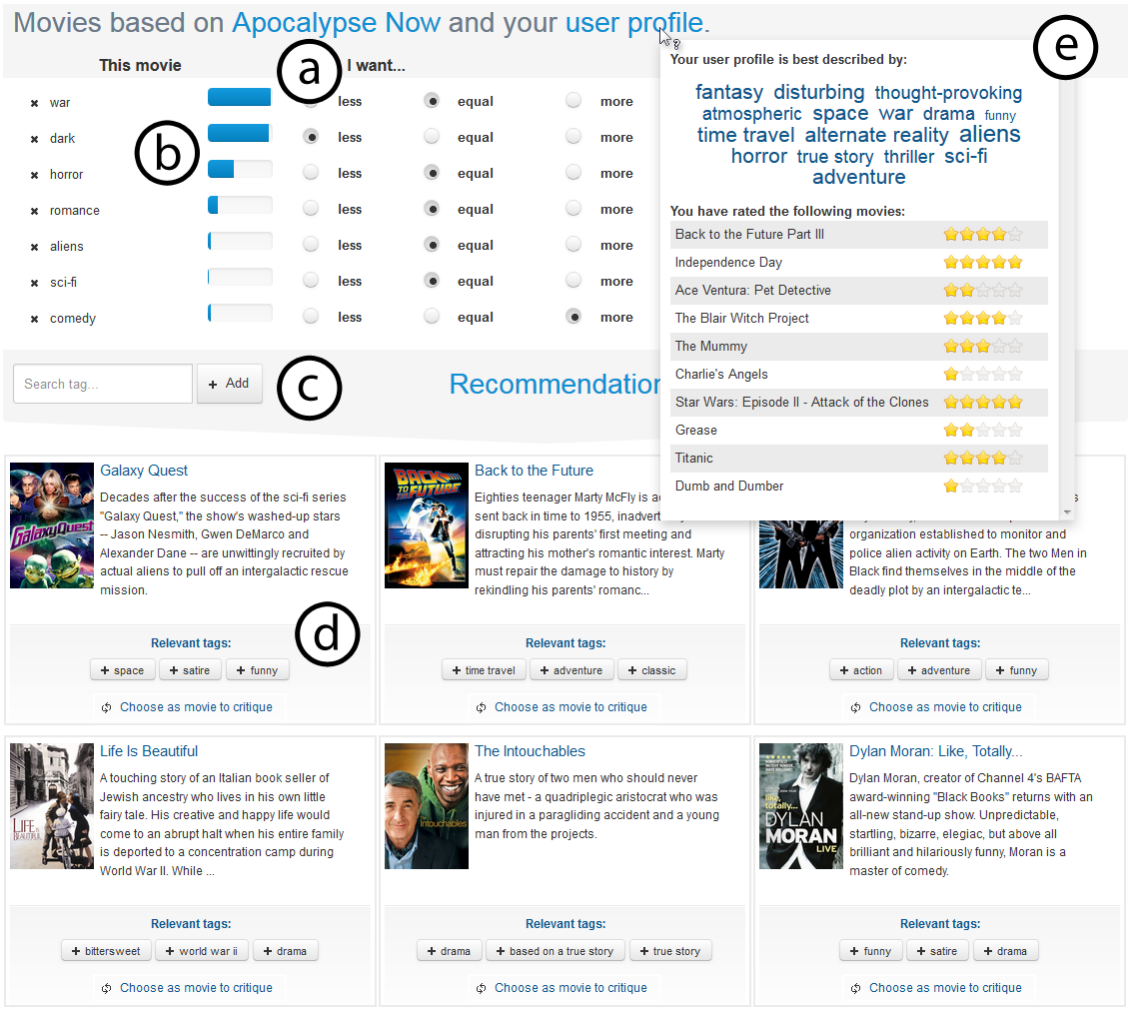

Figure A.4 Screenshot of the variant of the prototype system with critiquing based on content-boosted matrix factorization: At the top, the title of the currently critiqued movie is shown (a). A click opens a dialog with more details. The critiquing area is shown below (b). Initially, it contains 6 critique dimensions: 3 tags chosen by the system as described in the literature, 3 tags according to the method described in Section 6.2.3. For each tag, the relevance with respect to the critiqued movie is shown. The radio buttons allow to request new recommendations with less, equal or more relevance. Users can add further tags as critique dimensions using the input field underneath, supported by autocompletion (c).

The top 10 recommendations are shown below (d). For each movie, the 3 most relevant tags are shown, which may also be selected as critique dimensions. Another button allows to set the respective movie as the new item to critique and to start a new cycle in the critiquing process. Whereas the critiques represent situational needs, the general interests of users are visualized by the tag cloud shown in the dialog in the top-right corner, which is determined as described in Section 6.2.4 (e).

Experiment on blended recommending

In the user experiment we conducted to evaluate our concept of blended recommending (see Section 7.3), we used a prototype system in two variants, as shown in Figure A.5 and A.6.

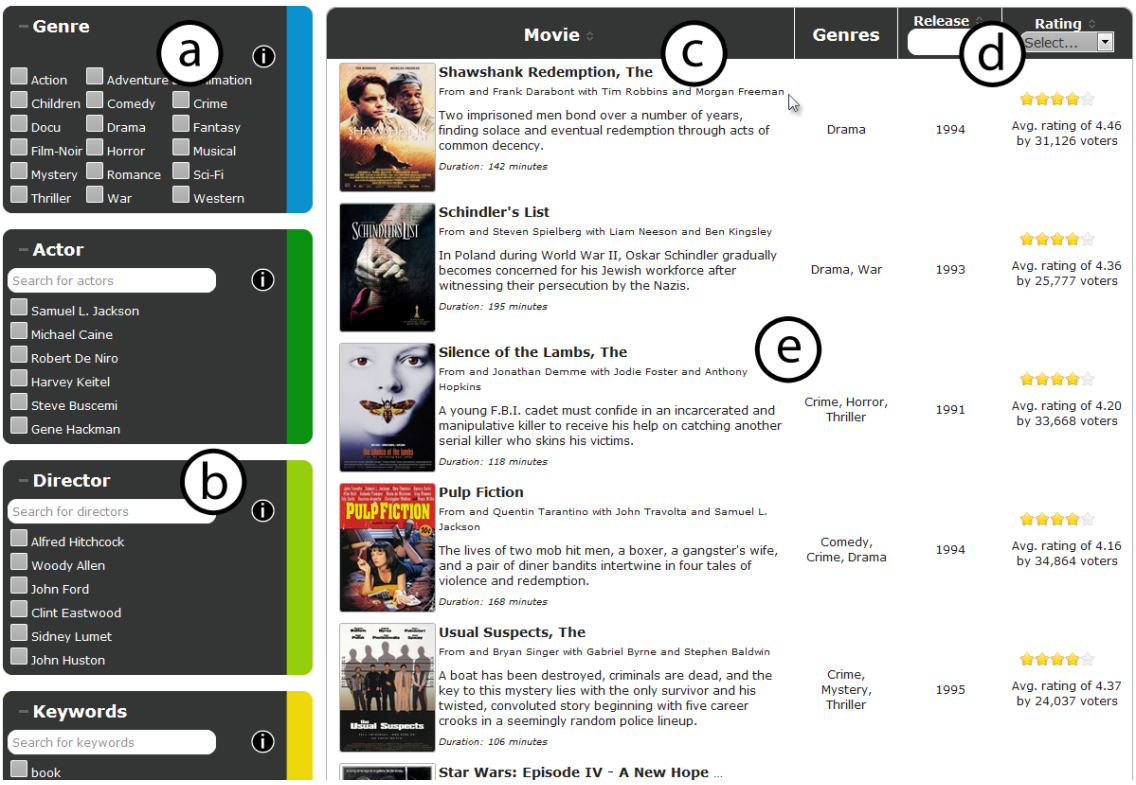

Figure A.5 Screenshot of the standard faceted filtering interface: On the left-hand side, there is a typical representation of facets and facet values (a). If the number of facet values is too large to be displayed, an input field with autocompletion allows to manually search for other values (b). The rest of the screen contains the result table: This table can be sorted by clicking on the column heads (c). There, additional search and filtering mechanisms are offered as well (d). Each row corresponds to a movie that satisfies the selected criteria (e). The movie poster, a short plot description, and relevant metadata are shown. A click on the title opens a dialog with more details. Director and actor names, genres as well as the release year are hyperlinks that may be used to apply further filter criteria.

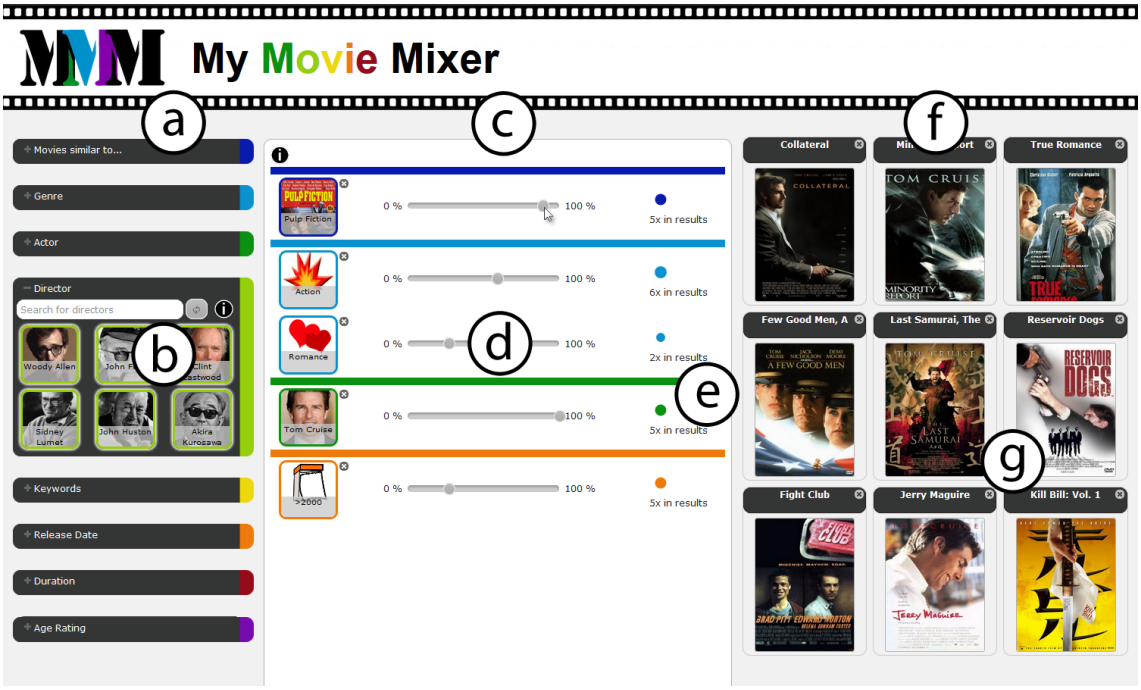

Figure A.6 Screenshot of the interface implemented according to the concept of blended recommending as proposed in Section 7.2: The area on the left-hand side contains the facets (a). Initially, all facets are displayed collapsed. Once opened, the facet values are displayed as tiles (b). Users can use a search function to look for other values, supported by autocompletion, or request suggestions based on frequent item properties in the current result set by using the "refresh" button. Users can drag tiles and drop them into the working area in the middle of the screen (c). There, a slider is attached to each criterion, allowing users to adjust the influence of the underlying recommendation method (d). Each interaction leads to an immediate update of the result set, providing users direct feedback regarding the effects of their preference settings. In addition, the bubbles provide visual clues to the number of items in the result set that fulfill the respective criterion (e).

The top 9 recommendations are shown on the right-hand side (f). For each movie, title and poster are presented. Clicking on a title opens a dialog with additional metadata as well as an explanation that indicates which criteria could be fulfilled. Each recommendation can be dismissed if users are not satisfied or already familiar with the item (g).

Integrated recommendation platform

Our integrated platform contains a few more perspectives than described in Chapter 8, among others, for presenting lists of items and item details, as shown in Figure A.7 and A.8.

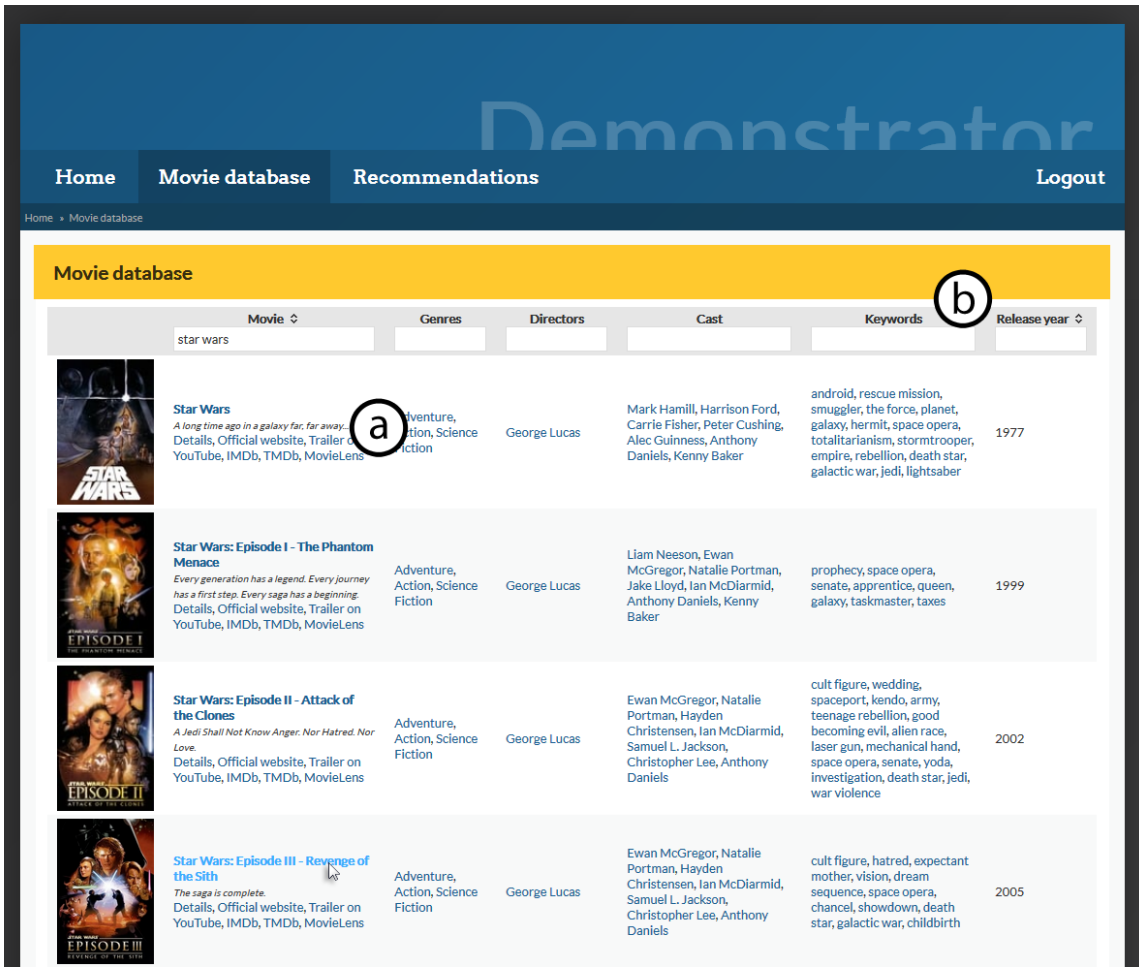

Figure A.7 Screenshot of one of the standard perspectives of the integrated recommendation platform: The item list perspective uses a table to display the movies from the underlying dataset. For each movie, metadata and links to external resources are presented (a). Clicking on a title or the name of a person opens a dialog with additional information. Users can also proceed to the corresponding item detail page. In the table head, search and filtering mechanisms as well as sorting functionalities are provided (b). Most of the elements in the table are hyperlinks that may be used to set values for these mechanisms.

Figure A.8 Screenshot of another standard perspective of the integrated recommendation platform: The item detail perspective includes a short plot description, metadata such as genre, length, age rating and budget (a), as well as details on director and cast (b). Most of these elements are hyperlinks that may be used to navigate back to the item list, but with a filter applied to show only movies, for example, from a specific genre or by a certain director. In addition, there is a typical widget that shows recommendations of similar items (c).

On the right-hand side, the movie poster is shown. In the area below, users can provide a rating for the movie and inspect the average rating by other users of the platform or by users of the *Internet Movie Database* (IMDb)[16], respectively. Moreover, a button allows to proceed to the critiquing perspective (d).

APPENDIX B

Questionnaires

In this part of the appendix, we present the complete questionnaires used in the empirical evaluations described in Section 4.3, 6.3 and 7.3. Colors indicate how constructs and corresponding items are related to the framework shown in Figure 3.2.[41]

Experiment on choice-based preference elicitation

We presented the following self-generated questionnaire items in German language to participants of the user experiment on choice-based preference elicitation (see Section 4.3). We translated them for the presentation in this thesis. Details on the questionnaire and how it was presented can be found in Section 4.3.2.

Overall satisfaction

- "I was satisfied with the result achieved."

Perceived recommendation quality

- "The selection matched very well with my movie interests."

Perceived recommendation novelty

- "The selection contained movies, which I probably would never have found otherwise."

Trustworthiness

- "I trust the system that it takes only my needs into account and not the goals of the system provider."

Interaction adequacy

- "Using the system was straightforward and easily comprehensible."

Perceived system effectiveness

- "I always had the feeling that the system learns my preferences."

Perceived control

- "I felt that I was in control of the selection process at all times."

Usage effort

- "The effort necessary to obtain a selection was acceptable."

Suitability for different usage scenarios

- "I would use the system if I already had a search goal in mind."
- "I would use the system if I had no search goal in mind."

[41] Items colored in gray are related to more general aspects.

Intention to use again

- "I would like to use the list of popular movies more frequently."
- "I would like to use the manual exploration interface more frequently."
- "I would like to use the recommendations generated based on ratings more frequently."
- "I would like to use the interactive recommendation dialog more frequently."

Domain knowledge

- "I watch about [. . .] movies per month."
- "I love movies."
- "I know a lot about movies."
- "I have a good overview of current movies."

Experiment on content-boosted matrix factorization I

We used the following self-generated statements and statements taken from the evaluation frameworks by Knijnenburg, Willemsen, and Kobsa [KWK11] and Pu, Chen, and Hu [PCH11]. We additionally used usability questionnaires by Brooke [Bro96] and Laugwitz, Held, and Schrepp [LHS08]. We translated all questionnaire items to present them in German (sometimes with slightly adapted formulations) to participants of the first user experiment on our content-boosted matrix factorization method and its application possibilities (see Section 6.3.1). Details on the questionnaire and how it was presented can be found in Section 6.3.1.2.

Perceived recommendation quality

- "I liked the movies recommended by the system." [KWK11]
- "The recommended movies fitted my preferences." [KWK11]

Perceived recommendation diversity

- "The recommendations contained a lot of variety." [KWK11]

Transparency

- "I understood why the movies were recommended to me." [PCH11]

Usability

- *System usability scale* (SUS) [Bro96]
- *User experience questionnaire* (UEQ) [LHS08]

Interface adequacy

- "The labels of the system interface are clear." [PCH11]
- "The labels of the system interface are adequate." [PCH11]
- "The layout of the system interface is attractive." [PCH11]
- "The layout of the system interface is adequate." [PCH11]

Choice satisfaction

- "I like the movie I have chosen." [KWK11]

Choice difficulty

- "Making a choice was an overwhelming task." [KWK11]

Usage effort
- "The system is convenient." [KWK11]
- "I had to invest a lot of effort in the system." [KWK11]

Suitability for different usage scenarios
- "I would use the system if I already had a search goal in mind."
- "I would use the system if I had only a vague search goal in mind."
- "I would use the system if I had no search goal in mind."

Intention to use again
- "I would like to use the system that only allowed to rate movies more frequently."
- "I would like to use the system that allowed to rate movies and to select and weight tags more frequently."

Domain knowledge
- "I love movies."
- "I watch many movies."
- "I know a lot about movies."
- "I have a good overview of current movies."

Trust in technology
- "Technology never works." [KWK11]
- "I am less confident when I use technology." [KWK11]

Experiment on content-boosted matrix factorization II

We used the following self-generated statements and statements taken from the evaluation frameworks by Knijnenburg, Willemsen, and Kobsa [KWK11] and Pu, Chen, and Hu [PCH11]. We additionally used items proposed by Vig, Sen, and Riedl [VSR11], and usability questionnaires by Brooke [Bro96] and Laugwitz, Held, and Schrepp [LHS08]. We translated all questionnaire items to present them in German (sometimes with slightly adapted formulations) to participants of the second user experiment on our content-boosted matrix factorization method and its application possibilities (see Section 6.3.2). Details on the questionnaire and how it was presented can be found in Section 6.3.2.2.

Overall satisfaction
- "Overall, I am satisfied with the system." [PCH11]

Perceived recommendation quality
- "I liked the movies recommended by the system." [KWK11]
- "The recommended movies fitted my preferences." [KWK11]

Perceived recommendation diversity
- "The recommendations contained a lot of variety." [KWK11]

Transparency
- "I understood why the movies were recommended to me." [PCH11]

Usability
- *System usability scale* (SUS) [Bro96]
- *User experience questionnaire* (UEQ) [LHS08]

Interface adequacy
- "The labels of the system interface are clear." [PCH11]
- "The labels of the system interface are adequate." [PCH11]
- "The layout of the system interface is attractive." [PCH11]
- "The layout of the system interface is adequate." [PCH11]

Interaction adequacy
- "The system allows me to tell what I like/dislike." [PCH11]
- "I found it easy to tell the system what I like/dislike." [PCH11]
- "I found it easy to inform the system if I dislike/like the recommended item." [PCH11]

Critiquing mechanism
- "The tags made sense to me." [VSR11]
- "The tags shown helped me learn about the movie." [VSR11]
- "I liked having the ability to specify critiques." [VSR11]
- "Movies displayed in response to my critique made sense." [VSR11]

Choice satisfaction
- "I like the movie I have chosen." [KWK11]

Choice difficulty
- "Making a choice was an overwhelming task." [KWK11]

Usage effort
- "The system is convenient." [KWK11]
- "I had to invest a lot of effort in the system." [KWK11]

Intention to use again
- "I will use this system again." [PCH11]
- "I will use this system frequently." [PCH11]

Domain knowledge
- "I love movies."
- "Compared to my friends, I watch many movies."
- "Compared to my friends, I am a movie expert."

Experiment on blended recommending

We used the following self-generated statements and statements taken from the evaluation frameworks by Knijnenburg, Willemsen, Gantner, Soncu, and Newell [Kni*12] and Pu, Chen, and Hu [PCH11]. We additionally used usability questionnaires by Brooke [Bro96] and Laugwitz, Held, and Schrepp [LHS08]. We translated all questionnaire items to present them in German (sometimes with slightly adapted formulations) to participants of the user experiment on blended recommending (see Section 7.3). Details on the questionnaire and how it was presented can be found in Section 7.3.2.

Overall satisfaction

- "Overall, I am satisfied with the system." [PCH11]

Perceived recommendation quality

- "I liked the movies shown by the system." [Kni*12]
- "The shown movies fitted my preference." [Kni*12]
- "The shown movies were well-chosen." [Kni*12]
- "The shown movies were relevant." [Kni*12]
- "The system showed too many bad movies." [Kni*12]
- "I did not like any of the shown movies." [Kni*12]

Perceived recommendation diversity

- "The recommendations contained a lot of variety." [Kni*12]
- "All the recommended movies were similar to each other." [Kni*12]

Usability

- *System usability scale* (SUS) [Bro96]
- *User experience questionnaire* (UEQ) [LHS08]

Interface adequacy

- "The labels of the system interface are clear." [PCH11]
- "The labels of the system interface are adequate." [PCH11]
- "The layout of the system interface is attractive." [PCH11]
- "The layout of the system interface is adequate." [PCH11]

Interaction adequacy

- "The system allows me to tell what I like/dislike." [PCH11]
- "I found it easy to tell the system what I like/dislike." [PCH11]
- "I found it easy to inform the system if I dislike/like the recommended item." [PCH11]

Sliders and visual clues

- "I found using the sliders helpful for manipulating the system."
- "Using a slider, I can indicate [the degree to which the criterion should be fulfilled in each of the recommended movies/in how many of the results the criterion should be considered/how important I find the criterion and how strongly it should overall be considered in the result set]."
- "I found the visual clues that indicated how often criteria could be considered in the results helpful."

Perceived system effectiveness

- "The system is useful." [Kni*12]
- "I would recommend the system to others." [Kni*12]
- "The system has no real benefit for me." [Kni*12]
- "I can save time using the system." [Kni*12]
- "I can find better movies without the help of the system." [Kni*12]
- "The system is recommending interesting content I had not previously considered." [Kni*12]

Perceived control

- "I feel in control of modifying my taste profile." [PCH11]
- "The system allows me to modify my taste profile." [PCH11]
- "I found it easy to modify my taste profile in the system." [PCH11]

Usage effort

- "The effort necessary to obtain a selection was acceptable."

Suitability for different usage scenarios

- "I would use the system if I already had a search goal in mind."
- "I would use the system if I only had a vague search goal in mind."
- "I would use the system if I had no search goal in mind."

Domain knowledge

- "I know [few/rather few/many/very many] movies."

Additional experimental results

In this part of the appendix, we present some additional experimental results that we omitted in Chapter 6 and Chapter 7 for the sake of compactness.

Experiment on content-boosted matrix factorization II

Table C.1 presents the exact UEQ results of the second user experiment on our content-boosted matrix factorization method. More details and other results can be found in Section 6.3.2.

Table C.1 *t*-test results ($df = 52$) for a comparison of the conditions with respect to the UEQ subscales for overall attractiveness, pragmatic and hedonic qualities. Higher values indicate better results on the 7-point bipolar scale. The best values are highlighted in bold.

	TAG		TMF				
Subscale	*M*	*SD*	*M*	*SD*	*T*	*p*	*d*
Attractiveness	1.73	0.65	**1.99**	0.79	1.28	.206	0.36
Perspicuity	2.20	0.63	**2.43**	0.61	1.32	.194	0.37
Efficiency	1.70	0.62	**1.95**	0.63	1.47	.148	0.40
Dependability	1.22	0.58	**1.61**	0.75	2.13	.038	0.58
Stimulation	1.36	0.66	**1.78**	0.80	2.09	.042	0.57
Novelty	1.18	0.94	**1.34**	1.09	0.60	.550	0.16

Experiment on blended recommending

Table C.2 presents the exact UEQ results of the user experiment on blended recommending. More details and other results can be found in Section 7.3.

Table C.2 *t*-test results ($df = 31$) for a comparison of the conditions with respect to the UEQ subscales for overall attractiveness, pragmatic and hedonic qualities. Higher values indicate better results on the 7-point bipolar scale. The best values are highlighted in bold.

	FFI		BRI				
Subscale	*M*	*SD*	*M*	*SD*	*T*	*p*	*d*
Attractiveness	1.04	0.89	**1.49**	0.91	1.43	.162	0.50
Perspicuity	1.94	0.75	**2.13**	0.89	0.68	.504	0.23
Efficiency	1.25	0.90	**1.31**	0.72	0.21	.837	0.07
Dependability	1.25	0.82	**1.38**	0.80	0.47	.642	0.16
Stimulation	0.69	0.95	**1.32**	0.99	1.89	.069	0.65
Novelty	0.02	1.25	**1.10**	1.17	2.56	.016	0.89

Details on matrix factorization

In this part of the appendix, we present some details on matrix factorization that we omitted in Chapter 2 and Chapter 5 for the sake of compactness.

Bayesian personalized ranking

Here, as mentioned in Section 2.2.3, we present the *partial derivatives* for Bayesian personalized ranking in line with (2.8). For the sake of clarity, we again use $\hat{r}_{uij}$ as shown in (2.9):

$$
\begin{aligned}
&\frac{\partial}{\partial p_{uf}} - \ln(1+\mathrm{e}^{-\hat{r}_{uij}}) - \lambda(\|\mathbf{p}_u\|^2 - \|\mathbf{q}_i\|^2 - \|\mathbf{q}_j\|^2 - b_i^2 - b_j^2) \\
&\qquad = \frac{-1}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot -(q_{if} - q_{jf}) \cdot \mathrm{e}^{-\hat{r}_{uij}} - 2\lambda p_{uf} \\
&\qquad \propto \frac{\mathrm{e}^{-\hat{r}_{uij}}}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot (q_{if} - q_{jf}) - \lambda p_{uf} = \frac{1}{1+\mathrm{e}^{\hat{r}_{uij}}} \cdot (q_{if} - q_{jf}) - \lambda p_{uf}\,, \\
&\frac{\partial}{\partial q_{if}} - \ln(1+\mathrm{e}^{-\hat{r}_{uij}}) - \lambda(\|\mathbf{p}_u\|^2 - \|\mathbf{q}_i\|^2 - \|\mathbf{q}_j\|^2 - b_i^2 - b_j^2) \\
&\qquad = \frac{-1}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot -p_{uf} \cdot \mathrm{e}^{-\hat{r}_{uij}} - 2\lambda q_{if} \\
&\qquad \propto \frac{\mathrm{e}^{-\hat{r}_{uij}}}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot p_{uf} - \lambda q_{if} \qquad = \frac{1}{1+\mathrm{e}^{\hat{r}_{uij}}} \cdot p_{uf} - \lambda q_{if}\,, \\
&\frac{\partial}{\partial q_{jf}} - \ln(1+\mathrm{e}^{-\hat{r}_{uij}}) - \lambda\|\mathbf{p}_u\|^2 - \lambda\|\mathbf{q}_i\|^2 - \lambda\|\mathbf{q}_j\|^2 - \lambda b_i^2 - \lambda b_j^2 \\
&\qquad = \frac{-1}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot -(-p_{uf}) \cdot \mathrm{e}^{-\hat{r}_{uij}} - 2\lambda q_{jf} \\
&\qquad \propto \frac{\mathrm{e}^{-\hat{r}_{uij}}}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot -p_{uf} - \lambda q_{jf} \qquad = \frac{1}{-1+\mathrm{e}^{\hat{r}_{uij}}} \cdot p_{uf} - \lambda q_{jf}\,, \\
&\frac{\partial}{\partial b_i} - \ln(1+\mathrm{e}^{-\hat{r}_{uij}}) - \lambda\|\mathbf{p}_u\|^2 - \lambda\|\mathbf{q}_i\|^2 - \lambda\|\mathbf{q}_j\|^2 - \lambda b_i^2 - \lambda b_j^2 \\
&\qquad = \frac{-1}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot -1 \cdot \mathrm{e}^{-\hat{r}_{uij}} - 2\lambda b_i \\
&\qquad \propto \frac{\mathrm{e}^{-\hat{r}_{uij}}}{1+\mathrm{e}^{-\hat{r}_{uij}}} - \lambda b_i \qquad = \frac{1}{1+\mathrm{e}^{\hat{r}_{uij}}} - \lambda b_i\,, \\
&\frac{\partial}{\partial b_j} - \ln(1+\mathrm{e}^{-\hat{r}_{uij}}) - \lambda\|\mathbf{p}_u\|^2 - \lambda\|\mathbf{q}_i\|^2 - \lambda\|\mathbf{q}_j\|^2 - \lambda b_i^2 - \lambda b_j^2 \\
&\qquad = \frac{-1}{1+\mathrm{e}^{-\hat{r}_{uij}}} \cdot -(-1) \cdot \mathrm{e}^{-\hat{r}_{uij}} - 2\lambda b_j \\
&\qquad \propto \frac{-\mathrm{e}^{-\hat{r}_{uij}}}{1+\mathrm{e}^{-\hat{r}_{uij}}} - \lambda b_j \qquad = \frac{-1}{1+\mathrm{e}^{\hat{r}_{uij}}} - \lambda b_j\,.
\end{aligned}
\tag{D.1}
$$

Based on the derivatives presented in (D.1), the following *update rules* may be used to adjust the factor values in the direction of the gradient:

$$\begin{aligned}
p_{uf} &\leftarrow p_{uf} + \eta\left(\frac{1}{1+\mathrm{e}^{\hat{r}_{uij}}} \cdot (q_{if} - q_{jf}) - \lambda p_{uf}\right), \\
q_{if} &\leftarrow q_{if} + \eta\left(\frac{1}{1+\mathrm{e}^{\hat{r}_{uij}}} \cdot p_{uf} - \lambda q_{if}\right), \\
q_{jf} &\leftarrow q_{jf} + \eta\left(\frac{-1}{1+\mathrm{e}^{\hat{r}_{uij}}} \cdot p_{uf} - \lambda q_{jf}\right), \\
b_i &\leftarrow b_i + \eta\left(\frac{1}{1+\mathrm{e}^{\hat{r}_{uij}}} - \lambda b_i\right), \\
b_j &\leftarrow b_j + \eta\left(\frac{-1}{1+\mathrm{e}^{\hat{r}_{uij}}} - \lambda b_j\right).
\end{aligned} \tag{D.2}$$

Bayesian personalized ranking for content-boosted matrix factorization

Continuing (5.5), these are the remaining *partial derivatives* for item biases:

$$\begin{aligned}
&\frac{\partial}{\partial b_i} - \ln(1+\mathrm{e}^{-\hat{r}_{uij}}) - \lambda\|\mathbf{p}_u\|^2 - \lambda\|{}^{\mathrm{i}}\boldsymbol{\Theta}\|^2 - \lambda b_i^2 - \lambda b_j^2 \\
&\quad\propto \quad \frac{\mathrm{e}^{-\hat{r}_{uij}}}{1+\mathrm{e}^{-\hat{r}_{uij}}} - \lambda b_i = \frac{1}{1+\mathrm{e}^{\hat{r}_{uij}}} - \lambda b_i\,, \\
&\frac{\partial}{\partial b_j} - \ln(1+\mathrm{e}^{-\hat{r}_{uij}}) - \lambda\|\mathbf{p}_u\|^2 - \lambda\|{}^{\mathrm{i}}\boldsymbol{\Theta}\|^2 - \lambda b_i^2 - \lambda b_j^2 \\
&\quad\propto \quad \frac{-\mathrm{e}^{-\hat{r}_{uij}}}{1+\mathrm{e}^{-\hat{r}_{uij}}} - \lambda b_j = \frac{-1}{1+\mathrm{e}^{\hat{r}_{uij}}} - \lambda b_j\,.
\end{aligned} \tag{D.3}$$

DOI: 10.17185/duepublico/74289
URN: urn:nbn:de:hbz:464-20210510-140502-3