Designing and Evaluating Recommender Systems With the User in the Loop

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Michael Jugovac

Dortmund

2019

Tag der mündlichen Prüfung: 16.08.2019

Dekan: Prof. Dr.-Ing. Gernot A. Fink

Gutachter:
Prof. Dr. Dietmar Jannach
Prof. Dr. Jens Teubner

Abstract

On many of today's most popular Internet service platforms, users are confronted with a seemingly endless number of options to choose from, such as articles to purchase on online shopping sites, music to listen to on online streaming platforms, or posts to read on social media. As a solution to this choice overload problem, recommender systems have been integrated into more and more websites and applications to help users find items that they might like or that could be useful in their current choice situation.

In recent decades, research on recommender systems has mostly been driven by offline performance comparisons, in which each new approach is compared to the state of the art in terms of its ability to retroactively predict user preferences in historical data sets. However, such a purely algorithmic research approach can only capture one of the many factors that contribute to a useful and engaging recommendation experience from a user perspective. In fact, a variety of aspects can influence how recommendations affect users' decision-making processes and how users perceive recommendations, including details regarding the recommender system's user interface or subconscious cognitive effects evoked by the recommendations.

In this thesis by publication, selected works of the author are presented that investigate different aspects pertaining to the design and evaluation of recommender systems from a more user-focused perspective. The first part of the thesis outlines each of these publications and positions them within the research context. The presented works investigate (i) how recommender systems interact with their users, (ii) how recommender systems should be evaluated with the user in mind, (iii) possible biases in user studies, (iv) an algorithmic strategy to re-rank recommendation lists according to individual user tendencies, and (v) two phenomena based on which recommendations can subconsciously influence user decision-making processes. The second part of the thesis, the appendix, contains the aforementioned publications in full. The presented studies demonstrate that it is imperative to design and evaluate recommender systems with the user in mind, taking into account the intricacies of interaction details, recommendation list composition, user context, and decision-making processes.

Contents

1 Introduction

"The purpose of computing is insight, not numbers."

Richard Hamming

What started out well over three decades ago as a vision in the mind of Tim Berners Lee of how humans could exchange information via a worldwide computer network [Ber89] has fundamentally changed the way humans live and interact with each other. Today, social media enables people worldwide to communicate in real time and even face-to-face via video chats. The Internet has not only connected humans in a way never before thought possible but also enabled completely new forms of e-commerce and online entertainment. Instead of choosing movies from a limited set of options in a TV guide, thousands of movies are available at consumers' fingertips to stream on televisions, laptops, or even smartphones. Maybe most impressively, however, today's consumers can buy almost anything online from the comfort of their home and have it delivered to their doorstep, sometimes even on the same day.

Although the Internet is nearly universally hailed as one of humanity's most influential achievements, not all developments brought about by the Internet are necessarily a blessing. Take, for example, a challenge that consumers are increasingly facing: the ever-growing size of item catalogs, such as in online purchase scenarios or on streaming websites. More options to choose from should, in general, be an advantage for consumers. However, when the number of options becomes too vast to inspect manually, consumers can experience adverse effects, hindering their ability to decide in a reasonable amount of time and reducing their satisfaction with their eventual choice [SGT10]. In this context, recommender systems (RSs) can help consumers by filtering the available option space and identifying a set of items that they would most likely be interested in based on, for example, knowledge about item features, past user actions, or community trends. Given the usefulness of recommendations not only as decision aids for consumers but also as marketing tools for retailers, recommender systems have experienced a sharp growth, in recent decades, alongside the popularization of the Internet itself. Apart from online shops, which regularly feature one or even multiple different forms of automatically generated suggestions, these recommender systems can also be found in online streaming services, for example, for music or movies; in social media; or even online dating services. Furthermore, the application of recommendations is not limited to traditional purchase scenarios. On the contrary, in settings where consumers pay a flat-rate subscription fee to access a certain service, such as movie streaming sites, service providers are also increasingly employing state-of-the-art recommender systems to make their extensive item catalogs more accessible to users. This, in turn, reduces decision effort, which makes consumers more satisfied with the provided service, ultimately serving the provider's goal of customer retention [Gar+14; GH15].

Just as recommender systems have become more common in the online world, so too has research on this topic increased over recent decades. Today, the field offers many avenues for research, such as user modeling aspects, RS interface design, and the application of recommendation in novel domains. However, by far the most common research topic of academic RS publications is the improvement of existing recommendation *algorithms* to achieve higher performance, i.e., matching users' tastes more accurately. In this context, a common research setup is to propose a new or improved algorithmic approach for generating recommendations and to subsequently evaluate the performance of this approach against state-of-the-art baselines on historical data sets. These historical data sets usually comprise user click logs or item ratings collected, for example, from movie streaming websites or online shops. To compare algorithm performance, this data is partly revealed to the recommendation algorithms to train their prediction model and partly hidden to evaluate the algorithms' ability to predict the users' hidden preferences or click actions [Her+04].

Over time, this style of *offline evaluation* has become the de facto standard for academic performance estimation of RS algorithms, which has given researchers a well-defined goal and a simple means of testing their ideas in a comparable way. As a consequence, increasingly complex recommender algorithms have been proposed over the years, ranging from simple rule-based strategies [Bur00], traditional nearest-neighbor methods [Sar+01], item feature-matching approaches [PB07], and latent feature factorization schemes [Fun06; KBV09], to the recently emerging field of deep neural network recommendations [CAS16; Hid+16]. This constant search for more accurate RS algorithms has undeniably driven the field forward in a major way. In the past, algorithmic proposals were often conceptually fundamentally different from previous approaches, which led to large performance increases over previous state-of-the-art baselines. However, in today's research publications, algorithmic proposals are often only of an incremental nature—changing minute details of previous implementations—and thus sometimes lead to only minor performance increases in offline experiments. Additionally, only a few studies [Bro+16; CGT13] actually show a connection between the offline performance of a recommender algorithm and its *online success*, and many studies indicate that offline and online results are essentially unrelated [BL15; Eks+14; GH15; JH09].

Given this discrepancy, it is concerning that the RS literature still largely focuses on offline experiments with sometimes only marginal performance increases. In contrast, even small changes in the RS user interface design can have a fundamental effect on the user's perception and acceptance of the recommendations. For example, in the experiments by Garcin et al., changing the position of the recommendation component on the website from the bottom of the page to the side resulted in a 125 % increase in the click-through rate [Gar+14]. Such examples serve as a reminder that offline experiments can create a disconnect between researchers and the target audience of the products being studied. In algorithmic research papers, users are often treated as a mere preference vector, which reduces the human component to an easily quantifiable system component. While such an abstraction level might be desired if pure algorithmic accuracy is the research goal, it is questionable whether such research can have any real-life impact for users of recommender systems and system providers.

In contrast, studies that evaluate proposed approaches in a way that considers online success or user satisfaction—for example, by deploying the proposed approaches on real-world websites or by conducting laboratory studies—are still underrepresented at conferences such as the ACM Conference on Recommender Systems [Per+18]. Similarly, studies on topics such as RS user interface design or domain-specific interaction challenges are often relegated to workshop publications because of challenging publication requirements for non-quantitative studies. This situation further incentivizes researchers to use offline evaluations for their performance analyses instead of complex setups involving human subjects. Finally, another often overlooked research topic is the effect of recommender systems on human decision making. In this context, numerous publications [Ado+12; Ado+13; Köc+16; Köc+18; NLF10] have shown that the presence of a recommender system, as well as the items it recommends, can affect the user's decision behavior, for example, by having a persuasive effect or by influencing the user's post-decision confidence and satisfaction. However, along with the above-mentioned user interaction topics, this research sub-field is still largely absent from the main tracks of major conferences.

This thesis investigates how to design and evaluate recommender systems from a more holistic perspective. Based on selected studies, a vision of recommender systems is presented that keeps the *user in the loop*, for example, by evaluating recommender systems with the user in mind instead of simply working with historical data sets, by considering cognitive effects that recommender systems might have on users' decisions, and by including user preferences in a more multifaceted way in RS algorithm designs.

1.1 Outline of the Thesis

The thesis is organized as follows. The remainder of this introductory section presents the author's peer-reviewed publications on which the thesis is conceptually based. In Chapter 2, the fundamentals of RS design and evaluation are illustrated, with an emphasis on the sub-fields relevant to the thesis, which include user-focused evaluation and interaction aspects. Then, Chapter 3 shows how novel application domains of recommender systems can be explored with the user in mind, i.e., by combining results from offline evaluations on historical data sets with practical insights from controlled user studies and online deployments. As an example, a systematic evaluation in a novel RS application domain is presented. Here, users of an interactive machine learning tool are assisted via recommendations of context-appropriate machine learning or data-processing building blocks. In Chapter 4, the importance of considering user characteristics during evaluation is highlighted. To this end, the chapter presents an exemplary user study investigating how a specific user bias can influence study outcomes if not accounted for. Given the evidence that considering the user is crucial when *evaluating* recommender systems, Chapter 5 takes a deeper look at how users can be more strongly considered during the *design* of recommender systems. In this context, a novel post-processing strategy is presented which considers user preferences in a multifaceted way during the recommendation process. Finally, Chapter 6 examines how recommender systems can affect users in their decision-making processes, for example, by subconsciously biasing them toward certain item attributes.

1.2 Publications

This thesis by publication includes six of the author's peer-reviewed publications. In the following, the author's individual contribution to each publication is stated. A complete list of publications can be found in the appendix.

Interacting With Recommenders—Overview and Research Directions. This paper was a joint work with Dietmar Jannach. The author of this thesis performed the initial literature search and categorization. He also wrote major parts of the manuscript.

Michael Jugovac and Dietmar Jannach. "Interacting With Recommenders—Overview and Research Directions." In: *Transactions on Interactive Intelligent Systems* 7.3 (2017), pp. 10:1–10:46

Supporting the Design of Machine Learning Workflows With a Recommendation System. This journal article was written in collaboration with Dietmar Jannach and Lukas Lerche. A previously existing small code base [JF14] was substantially extended by the author of this thesis to accommodate a much wider range of (contextualized) algorithms. The offline experiments and real-world log data analyses presented in the paper were conducted by the author of this thesis, and he also contributed to the writing of the paper.

Dietmar Jannach, Michael Jugovac, and Lukas Lerche. "Supporting the Design of Machine Learning Workflows With a Recommendation System." In: *Transactions on Interactive Intelligent Systems* 6.1 (2016), pp. 8:1–8:35

Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation. This journal article was a joint effort with Dietmar Jannach and Lukas Lerche. The author of this thesis collaboratively developed the recommender system used in the user study with Lukas Lerche. The user study was supervised by the author of the thesis, and he also wrote parts of the text.

Dietmar Jannach, Lukas Lerche, and Michael Jugovac. "Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation." In: *i-com* 14.1 (2015), pp. 29–39

Efficient Optimization of Multiple Recommendation Quality Factors According to Individual User Tendencies. This paper was a joint work with Dietmar Jannach and Lukas Lerche. The author of this thesis designed and implemented the `Personalized Ranking Adjustment` algorithm and wrote major parts of the manuscript. Lukas Lerche assisted with the experimental evaluation, and Dietmar Jannach wrote and improved parts of the paper.

Michael Jugovac, Dietmar Jannach, and Lukas Lerche. "Efficient Optimization of Multiple Recommendation Quality Factors According to Individual User Tendencies." In: *Expert Systems With Applications* 81 (2017), pp. 321–331

New Hidden Persuaders: An Investigation of Attribute-Level Anchoring Effects of Product Recommendations. This journal article was a joint work with Sören Köcher, Dietmar Jannach, and Hartmut H. Holzmüller. The author of this thesis performed the log data analysis described in *Study 1*; contributed to the literature review (as part of the section *Conceptual Background*); and conducted a preliminary eye-tracking study, which eventually led to the experiments detailed in *Study 3*. He also helped to improve parts of the manuscript, which was primarily written by Sören Köcher.

Sören Köcher, Michael Jugovac, Dietmar Jannach, and Hartmut H. Holzmüller. “New Hidden Persuaders: An Investigation of Attribute-Level Anchoring Effects of Product Recommendations.” In: *Journal of Retailing* 95.1 (2018), pp. 24–41

Investigating the Decision-Making Behavior of Maximizers and Satisficers in the Presence of Recommendations. The research presented in this paper was conducted in collaboration with Ingrid Nunes and Dietmar Jannach. With the help of Ingrid Nunes, who preprocessed the training data, the author of this thesis implemented the recommendation component employed in the user study as well as the user study frontend itself. He supervised the study, and he also prepared and analyzed the resulting data. Writing the paper was a collaborative effort among all authors.

Michael Jugovac, Ingrid Nunes, and Dietmar Jannach. “Investigating the Decision-Making Behavior of Maximizers and Satisficers in the Presence of Recommendations.” In: *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP ’18)*. 2018, pp. 279–283

Background 2

As a foundation for later chapters, this chapter briefly introduces the field of recommender systems. In Section 2.1, the main design philosophies of commonly used recommender systems are explained. Section 2.2 presents an overview of how the real-life usage of recommender systems (e.g., in industrial settings) differs from academic research and how this disconnect increasingly affects the progress in the field. Finally, Section 2.3 surveys the literature for studies that try to bridge this gap by considering human-recommender interactions as a key component of their research.

2.1 Design Philosophies of Recommenders

From a technical perspective, RS design has changed tremendously in recent decades, partially due to key discoveries of novel algorithmic approaches. In addition, given the more widespread usage of recommender systems on a variety of websites, novel use cases for recommendation algorithms have also changed how recommender systems are designed. In this section, the motivators of RS design and key design philosophies are introduced.

2.1.1 Use Cases That Drive Algorithm Design

Traditionally, recommender systems have often been used as a tool to generate suggestions based on *long-term preferences* of users. In this context, an often-mentioned example are video streaming platforms, where users watch videos on a regular basis [GH15]. In this case, a sizable user profile is available, either as a simple watch history or as a set of ratings if the platform offers this functionality. Based on this historical user profile data, RS algorithms can generate suggestions in preparation for the next user visit. Such recommendations are commonly displayed within a front-page widget with a title such as “recommended for you.”

In contrast to the application scenario mentioned above, some use cases feature no previous information about the user’s long-term preferences. For example, when a customer wants to buy a new car, the decision is normally not based on stable long-term preferences, but on the customer’s current requirements and desires. In this case, recommender systems can still be applied, but they have to collect preference information on the spot, which is commonly accomplished by *asking the customer a set of questions*. The answers can then, for example, be checked against a knowledge base and a set of rules to determine the most fitting car choice [Bur00].

While the formerly-mentioned use case of recommending based on long-term preferences is still among the most popular RS applications, over time, other use cases have gained popularity. For example, as catalogs of online retailers become larger and larger, displaying just a single list of recommended products that does not consider the current user context can be insufficient. Instead, websites like Amazon[1] display a variety of recommendation widgets on each product page that are tailored to the user's situation. These include (i) alternatives to the currently inspected product, (ii) complementary products, and (iii) product recommendations based on recently inspected items. In the last case, a range of different approaches have recently been proposed to address the problem of adapting recommendations to the user's short-term preferences—called *session-based recommendation* schemes [Hid+16; JL17].

In other domains, specific use cases can also dictate algorithm design. For example, on streaming platforms, recommendations solely based on long-term preferences might lead to a rather monotonous listening or watching experience. To achieve a more pleasant user experience, several more specific RS applications are possible on such sites. A music recommender approach might, for example, be used to (i) provide a list of song suggestions that exhibits a certain level of genre diversity [JKL17; KJ17], (ii) allow the user to interactively discover new tracks [Zha+12], or (iii) automatically generate coherent playlists for specific occasions [JLK15].

In addition to the use cases mentioned so far, Jannach et al. list a variety of further application scenarios both from the user's and the provider's viewpoint—which they term *purposes* of recommender systems [JA16]. These include proactive recommendations, which have become more prevalent recently in the form of mobile push notifications, as well as group recommendations, in which a recommender system can assist a group of people in achieving consensus. In Figure 2.1, an overview of the mentioned use cases is provided.

As demonstrated by these examples, different application scenarios impose different requirements on the design of recommendation algorithms. Understanding the application scenario should thus be considered essential not only to the design process of commercial recommender systems but also to academic research.

[1] https://amazon.com

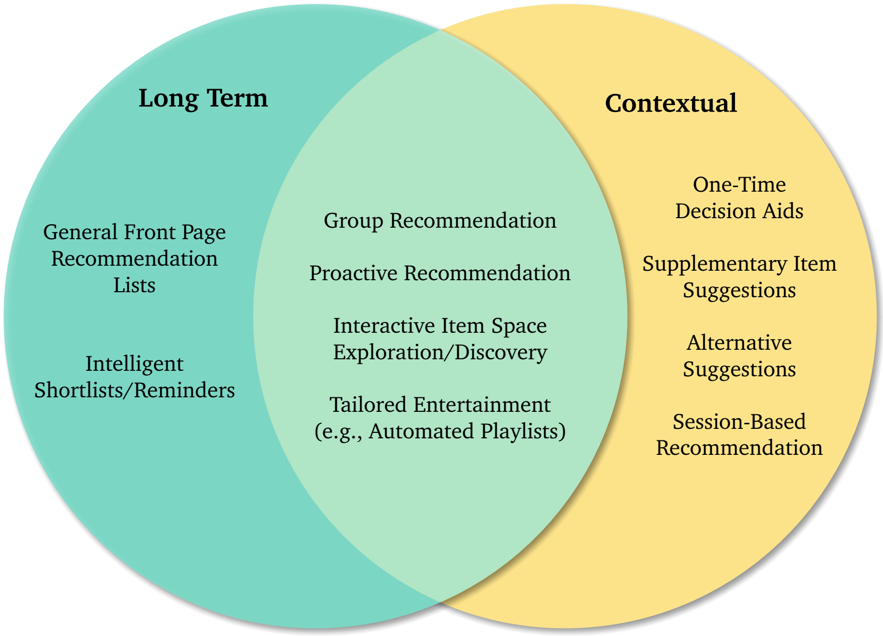

Long Term
General Front Page Recommendation Lists
Intelligent Shortlists/Reminders
Group Recommendation
Proactive Recommendation
Interactive Item Space Exploration/Discovery
Tailored Entertainment (e.g., Automated Playlists)
Contextual
One-Time Decision Aids
Supplementary Item Suggestions
Alternative Suggestions
Session-Based Recommendation

Figure 2.1: Use cases of recommender systems.

2.1.2 Common Recommendation Approaches

Given the growing variety of application scenarios for recommender systems, a wide range of technical approaches for generating recommendations have been proposed and successfully applied. In the literature, these approaches are generally grouped into three families that distinguish themselves from each other mainly in terms of which data sources are used.

Content-based filtering (CBF) works by exploiting item features, e.g., by comparing the features of the items the target user previously consumed with other potential candidate items' features [PB07]. This kind of recommendation approach can also be useful for those previously mentioned use cases in which alternative items are displayed in the context of a focus item. For example, when a customer is shopping for a laptop, the currently inspected laptop's features (e.g., processing power or RAM size) could be used to find contextually suitable alternative products from different brands.

Where content-based filtering focuses on items, *collaborative filtering* (CF) harnesses user behavior or preference statements to generate recommendations. As the term "collaborative" suggests, the idea is to use information about one user to predict the preferences of another user. A simple example of such an approach can be explained based on the viewing behavior of users on a movie streaming platform. In this context, a so-called nearest-neighbor approach [Ama+11] can be used to compare the viewing history of a target user with the viewing histories of other users to find a set of users with similar preferences—called *neighbors*. After these neighbors have been identified, their viewing histories can be scanned again to find movies that the target user has not watched in the hope that the target user might like these movies. However, collaborative-filtering approaches are not limited to use cases in which the target user's consumption, purchase, or rating history is available. They can also be applied on a smaller scale, for example, in the context of a target item for which complementary items are to be displayed. In this case, a trivial strategy could collect co-purchase statistics [SKR01], which might, for example, reveal that laptop X is often bought together with mouse Y, leading to recommendations with the well-known label "Customers who bought...also bought..."

Collaborative-filtering approaches have proven to be among the most versatile recommendation strategies. This is due to, on the one hand, their superior prediction performance and, on the other hand, the growing availability of click, rating, and purchase log data. Unsurprisingly, this has led to the creation of a wide range of CF approaches, from (i) the simple ones mentioned above; to (ii) more complex matrix factorization strategies [KBV09], in which user behavior data is decomposed into latent factor vectors (akin to principal component analysis); to (iii) highly complex deep learning schemes [CAS16; Hid+16].

Lastly, *knowledge-based* systems, which are mostly reserved for one-time, premium purchase experiences, work based on item features, a set of rules, domain knowledge, and the user's expressly stated requirements [Bur00]. To this end, users have to complete questionnaires to assess their requirements. For example, when buying a car, users might be asked about their yearly mileage and transportation needs, instead of having to make direct decisions on complex features (e.g., engine displacement volume). Consequently, such approaches can make complicated purchase decisions easier, especially for people unfamiliar with the technicalities of the domain. However, to generate useful recommendations based on the user's requirements, a well-maintained knowledge base is necessary, which makes this type of recommender system exceedingly rare, especially compared to near-zero human-maintenance CF approaches.

Note that, while the categorization of approaches into families is helpful in academia, practically applied algorithms rarely fall strictly into one of the above-mentioned categories. Instead, RS designers often combine the advantages of different approaches to create *hybrid* recommenders [Bur02]. For example, system providers might combine a content-based system, which does not suffer from cold-start problems since only item features are necessary, with a collaborative system, which can improve overall system performance as soon as enough user feedback is available.

2.2 Success in Industry and Academia

Considering the multitude of RS use cases as well as the technical approaches that address these use cases, unsurprisingly, numerous RS success stories have been recorded over the years by both industrial system providers and academic researchers [Dav+10; Gar+14; GH15; JH09]. Success, however, is not always defined and quantified in the same way, and evaluation methodologies can differ quite strongly. This can, on the one hand, lead to a disconnect between research and reality, making research results unrepresentative of real-life recommendation "performance." On the other hand, operating with a purely business-oriented mindset can make system providers lose sight of individual users' requirements, desires, and emotional needs.

2.2.1 Industry Success

In industrial settings, RS success is often defined in terms of indirect or direct business value. On the one hand, direct business value can be quantified with business metrics, such as conversion rates, gross sales volumes, churn rates, or—less commonly—revenue. In certain situations, system providers might additionally be incentivized to push specific products or product ranges that yield higher profit margins with the help of recommendations. Indirect business value, on the other hand, is typically quantified via proxies such as click-through rates, i.e., clicks on recommended items, or other forms of user engagement, such as website visit durations.

Regardless of the chosen metric, most reports from industrial system providers follow a similar evaluation methodology. First, initial performance evaluations are conducted in an offline setting, where a range of potentially viable algorithmic approaches are compared based on historically collected data, such as click or purchase logs. After one or more well-performing algorithms have been identified in this way, an A/B test is conducted to gauge the online success of the approaches. In such an evaluation setting, novel approaches are compared to an already existing recommendation approach or a non-recommendation condition by exposing part of the user base to the experimental recommender. While this evaluation strategy is not without its flaws (e.g., susceptibility to seasonal effects), it is, to date, the most reliable way of measuring the true online success of a recommendation technique [SG11].

2.2.2 Academic Success

Academic research, in contrast, is driven by different motivators and can often not achieve the same realistic evaluation settings since most researchers do not have experimental access to real-life large-scale recommender systems. As a result, A/B tests are extremely rare in academic literature, and a large fraction of today's research on recommender systems is validated by offline evaluation. To this end, proposed recommendation approaches are compared against state-of-the-art baselines in terms of prediction accuracy on historical data sets. This evaluation approach allows for rapid prototyping and, notably, accelerated publication cycles. However, it is not always fully clear whether the offline accuracy of such algorithms translates into practical recommendation success [BL15; Eks+14; GH15; JH09].

Seemingly in response to this limitation of offline experiments, the volume of RS research that focuses on user studies has recently increased to a certain degree (see Section 2.3). While such studies cannot capture the real-life success of a recommender system as realistically as industrial A/B tests, they can be used to investigate qualitative aspects of the user's perception of the recommender system. For example, whereas A/B testers can only speculate that users are more satisfied with a system because they spend more time on the website, researchers using qualitative approaches can ask users directly about their satisfaction or ease of use. These aspects might not be goals that system providers want to pursue, which could explain why industrial publications rarely report on such qualitative studies. However, evaluations solely based on business-related metrics might also prove insufficient, as performance measures such as user satisfaction can have trickle-down effects, for example, by improving brand recognition or recommend-to-friend rates, which cannot be captured by quantitative A/B tests alone.

2.2.3 Finding a Middle Ground

Overall, both academic and industrial evaluation approaches have their advantages and disadvantages. Academic research mostly suffers from a lack of access to real-life systems to test approaches realistically, as well as from general pressure to publish, which favors evaluations using historical data over time-consuming user studies. Industrial research, in contrast, enjoys access to online evaluation platforms to perform A/B tests that can realistically gauge user feedback and thus business effects. However, while realistic, such experiments can lose sight of the individuals for whom the system is designed.

To address this disconnect between industrial and academic research, a methodological shift toward a middle ground is necessary that takes both business success and individual users into account. This entails a more holistic strategy of evaluating recommender systems in which (i) offline experiments exist as a tried-and-trusted *first-level tool* to roughly estimate recommendation performance; in which (ii) industry cooperates with academia to enable more researchers to evaluate their approaches

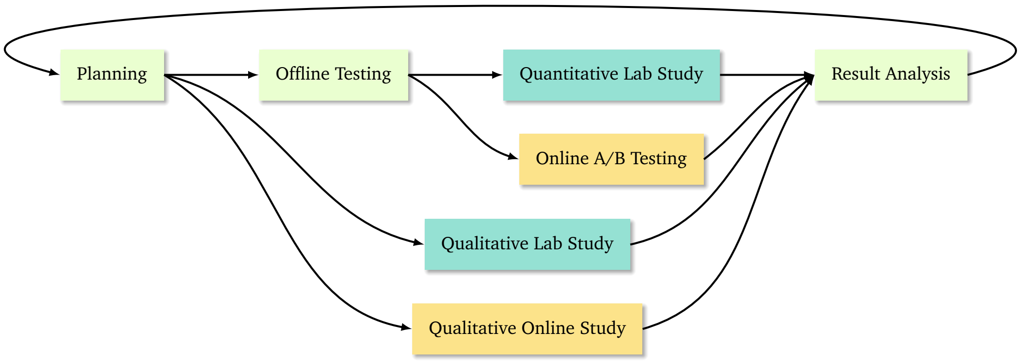


Figure 2.2: The ideal lifecycle of an RS evaluation project. Steps without user involvement are in orange, steps that require a live online system in green, and steps that require lab participants in blue.

in online A/B tests; and in which (iii) qualitative studies routinely accompany quantitative experiments. A visualization of this ideal evaluation lifecycle is provided in Figure 2.2.

2.3 Human-Recommender Interaction

When comparing everyday practices of evaluation with best practices, *the user* is all too often overlooked, which can lead to results that are not representative of real-world effects. However, while it is important to consider the user during the evaluation of recommender systems, it is equally crucial do so during the *design* of recommender systems, instead of just focusing on the design of the most accurate recommendation algorithm.

Recommendation algorithms depend on components around them, such as interfaces that allow users to enter their preferences; visual design elements that present the recommended items to the user; and, in certain cases, further interaction elements that enable users to provide feedback on the recommendations [XB07]. These individual components can influence each other, making it impossible to estimate the true value of a recommender system without looking at the system as a whole [JA16; Jan+].

This section presents an overview of possible RS interaction mechanisms, which is fundamentally based on a previous literature survey [JJ17]. To this end, both traditional systems and more experimental forms of recommender systems that focus on user interaction aspects in their system design are investigated. The section has two parts. In the first part, different ways in which user preferences can be elicited are presented. The second part focuses on approaches for presenting recommenda-

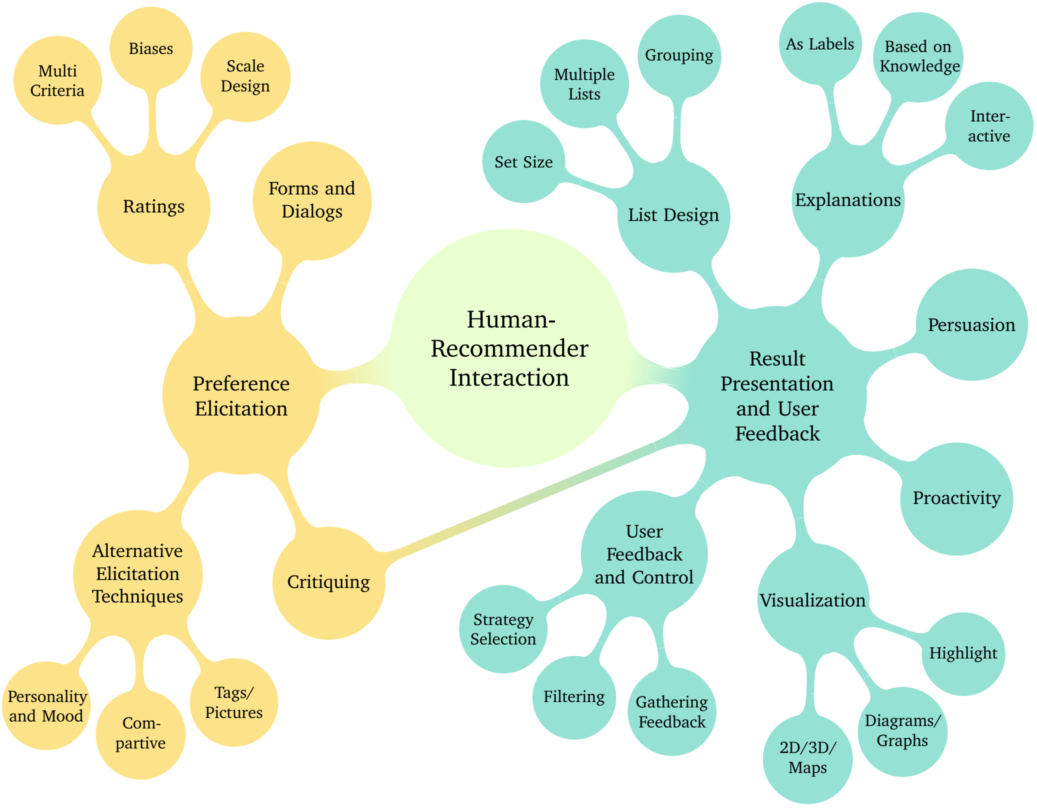


Figure 2.3: Concepts of human-recommender interaction.

tions to users and allowing them to provide feedback. An overview of the concepts presented in this section is visualized in Figure 2.3.

2.3.1 Preference Elicitation

Much of today's RS research assumes that the input data for the recommendation process is collected in the background, e.g., by observing users' access patterns on a shopping website or by tracking their listening history in a music streaming application. In fact, many real-world systems use this approach, as in this way, recommendations can be provided without any active user input. However, since users are not asked directly about their preferences, the amount of *useful* information available to the RS can be limited. For example, if a user repeatedly listens to folk music, the system cannot judge whether it would be useful to recommend rock music. In contrast, allowing users to specify their preferences explicitly—possibly in an interactive or creative way—can enable recommenders to gain deeper insight into users' preferences, while providing users with a more engaging and potentially more transparent experience.

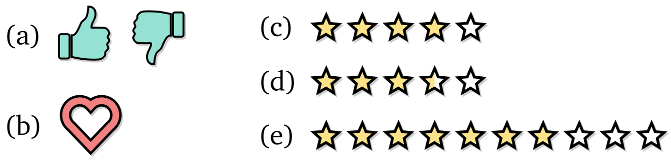


Figure 2.4: Commonly used rating inputs: binary (a) and unary (b) scales, as well as rating scales with different options and step sizes (c–e).

Ratings

The most basic way how users can directly communicate their preferences to a recommender system is by rating items. Such item ratings can be expressed on either a continuous or, more commonly, a discrete scale (e.g., from one to five stars). As this is one of the most common forms of data collection in the RS literature, a whole range of rating interaction methods have been investigated over the years. In general, these systems share the design goal of allowing users to express their preference in a convenient and reliable way—i.e., without any unwanted biases due to, for example, an ambiguous rating scale design.

To achieve these goals, several factors have to be considered, the most obvious being the design of the rating scale itself. In academic research, scales are commonly numerical. However, even among simple numerical scales, a variety of options exist, for instance, regarding the number of possible rating values. Common forms include scales from 1 to 5 or 1 to 10, often in increments of 1, or sometimes 0.5. However, recently, simple binary options (thumbs up/down) or unary systems with "like" as the only available action have become more common. Figure 2.4 illustrates these most common forms of rating inputs. Users are, in general, familiar with such rating methods, which makes them suitable for acquiring reliable preference data. However, system providers should not underestimate the intricacies of these seemingly simple input mechanisms. Studies have, for example, found evidence that option labels of such rating inputs can introduce biases. For instance, Amoo et al. identified that a scale from -5 to 5 is not comparable to a scale from 0 to 10 [AF01]. Furthermore, users react differently to scales with an even number of options than to those with an odd number of options. In general, the literature suggests that finer scales are preferred by users because they feel more in control [Cos+03]. However, system providers should also consider that users take more time to rate an item on a finer scale [SS11], which means that, in the end, trade-offs are inevitable.

As rating data is frequently used in RS research, several studies have also been carried out with more experimental rating mechanisms, such as sliders [SS02], pinching and tilting gestures on mobile systems [WWL13], and even emotional rating scales [Pom+12]. Notably, such experimental approaches are not always relegated to academic literature. For example, in 2017, Facebook extended its long-used "like button" mechanic for posts with emotional feedback in the form of smileys, called *Reactions*.[2]

In terms of reliability, rating systems can suffer from biases originating on both the user side and the system side. For example, users' ratings may differ from their actual opinions, or a scale may not be well designed, as already discussed. To ensure that a rating system is as reliable as possible, providers should be aware of the user's context (e.g., desktop, mobile, or television [Klu+12]) and of the intended trade-off between user effort, accuracy, and noise in the collected data. Additionally, from a user perspective, it cannot be assumed that users can accurately express their enjoyment of a product on a numerical scale, not least because users cannot always easily break their preferences down into a single rating dimension. In this context, multi-criteria ratings can allow users to express themselves along multiple scales, e.g., in terms of the cleanliness, service quality, and price of a hotel room. Several studies have shown that such rating methods can indeed produce more useful data for recommendation purposes [AK07; FZ12; NHA09]. However, as always, the additional scale complexity also increases user effort.

Another reason why users might not rate items accurately is that they are unconsciously influenced in their judgment. They might, for example, be primed by initially seeing a community rating before giving their own rating, which could bias their opinion (see Chapter 6). Additionally, especially in academic laboratory studies, users might be asked to rate items that they have not experienced fully [Loe+18]. They might, for example, be asked to rate their "potential enjoyment" of a certain movie based on just its description and trailer. While such data collection methods are commonplace in RS studies, researchers should consider that users rate items differently depending on if they are already familiar with them (see Chapter 4).

Forms and Dialogs

Although at the moment, ratings and implicit data collection methods, such as click tracking, are by far the most prominent forms of input for recommender systems, other forms of user interactions have also received attention in research and practice over the years. Among these are preference entry forms and adaptive dialog systems. Together, these preference elicitation methods offer users a more explicit way of expressing their preferences than item ratings alone, which is one reason they are used in situations in which reliable feedback is necessary. Examples include cold-start scenarios or situations in which expensive goods, such as cars, are recommended.

[2] https://en.facebookbrand.com/assets/reactions/

With *preferences forms*, websites can allow their users to enter a set of important preferences in a short amount of time. For example, when initially entering a music website as a new member, users might complete a form about their favorite genres and artists. In contrast to item ratings, the limited selection of entities (e.g., genres) makes it easier to design a questionnaire that users are likely to answer confidently and quickly. Additionally, the broader range of preferences that can be collected with a form-based system can also allow the recommender system to generate meaningful recommendations with fewer interaction steps.

In academia, such systems have received little research attention. However, a few examples exist, such as the energy-saving recommender application by Knijnenburg et al., in which users enter their energy needs in a form [KRW11]. In practice, such form-based techniques are quite widespread as an elicitation tool for new users in music and movie streaming systems. However, other websites also sometimes allow users to access a preference dialog to tweak their preference profile. For example, on the Google News[3] aggregation site, users could, until recently, access a settings menu to enter preferences on a slider for certain news topics, such as world news or politics.

In contrast to the static nature of preference forms, *preference dialogs* represent a much more adaptive and, sometimes, even personalized way of eliciting user preferences. Such interaction methods are mostly used when long-term preferences do not play a large role in the choice process, for instance, in purchase scenarios for expensive goods, such as digital cameras. In such cases, the immediate short-term needs of the user are more important, which is why an adaptive preference elicitation system can be a suitable option. Such dialog-based systems are usually based on explicit domain knowledge (e.g., digital camera features) and a set of reasoning rules. The user's requirements are then elicited step by step and are fed into an automated inference process that considers the domain knowledge to identify which item features satisfy the user's requirements (e.g., "wants to print photographs in poster format" → "high-resolution camera sensor"). In some more advanced systems, the next set of questions can be chosen adaptively based on the user's previous answers so that the process of finding the right recommendation becomes as efficient as possible.

While such dialog-based systems were once popular in the context of high-involvement goods, they have mostly disappeared from online stores, as (i) such systems require too much user effort and (ii) system providers are not willing to create and maintain the complex underlying knowledge bases. However, in academic research, dialog-based systems still fill a large niche area, starting with early approaches

[3] https://news.google.com

for travel planner systems [GT00; LHL97] and culminating in more sophisticated approaches that provide a high degree of dialog flexibility [Fel+07; JK05; JK07; JWB05; KKP01; Lee04; Rut+08; Yah+15]. These more advanced systems can adapt the navigation structure, content depth, or presentation format itself, for example, based on the user's current search goals or their expertise. Recently, the field has seen a resurgence in research interest, as technologies such as chat bots and voice assistants, which could be used as interfaces for dialog-based RSs, have been popularized in mobile operating systems [Arg+18; Hol16; NR17].

Critiquing

As a mixture of dialog-based approaches and item rating systems, critiquing also follows an iterative approach of guiding users to the best-matching recommendation. However, in contrast to dialog-based RSs, the user is, from the beginning, exposed to an item recommendation, which can then be "criticized" to reach progressively more satisfying recommendations. Depending on the domain, the available feedback options for item critiques differ. For example, in an apartment recommender system, users might be presented with feedback options such as "larger," "cheaper," or "closer to..." [Bur00]. After every user critique, a new apartment is recommended until a satisfactory choice can be made.

Since critiquing combines aspects of preference elicitation, result presentation, and feedback, it can be classified as both an elicitation strategy and a feedback mechanism (see Figure 2.3). While critiquing offers the advantages of being both simple to understand and gratifying for users because of the instant feedback-response mechanism, interactions might become repetitive quickly. Furthermore, the process can be lengthy, as each refinement step might not always have a strong effect on the recommender model. A solution for the latter point can be *compound critiques*, in which multiple feature changes to the currently selected item are combined into a single feedback cycle [CP06; McC+04; Rei+04; Rei+05; ZP06]. A compound critiquing system could, for example, offer users a critiquing option along the lines of "a laptop with more RAM but less CPU power" (see Figure 2.5 [ZJP08]). It is, however, unclear whether compound critiquing can actually decrease user effort, as studies offer conflicting results [CP06; ZP06]. Lastly, as dialog-based RSs, critiquing systems are mostly geared toward single-use recommendations, and only a few proposals have tried to re-incorporate users' critiquing preferences into a long-term model.

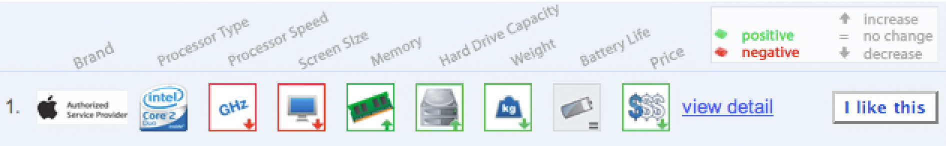


Figure 2.5: Compound critiquing system by Zhang et al. [ZJP08]. ©Zhang et al.

Alternative Elicitation Techniques

The methods presented so far form the backbone of most academic RS research on eliciting user preferences. However, over the years, there have also been ventures into more experimental elicitation techniques. Among these are emotion or personality quizzes, in which the user's current mood or general attitude toward certain emotional content is exploited in the recommendation process [NH12]. Specifically, in emotion-driven domains such as music [Per+04], user personality (e.g., extroversion level) or mood can be used to tailor recommendations [Ela+13]. To this end, the well-established five-factor model [Gol93] is often used as a basis for questionnaires to elicit the user's personality. Such personality or mood information can complement explicit preference data in a mixed-data recommendation model, or the user's personality model can be used in cold-start situations, in which other implicit or explicit feedback is not yet available.

In addition, a whole range of other approaches try to ascertain user preferences in more unconventional ways. For example, Loepp et al. tried to "gamify" the elicitation process in their study by letting users decide between two sets of movies [LHZ14]. Furthermore, several techniques have been studied in RS literature on letting users rate or select more engaging domain entities than the items themselves. For example, selecting pictures has been explored as an alternative form of elicitation in the domain of travel recommendation [Nei+15], and in the movie domain, letting users rate community tags has garnered a substantial amount of research attention [Ela+15; GJ13; IGÖ15; SSV15; SVR09].

2.3.2 Result Presentation and User Feedback

As shown in the previous sections, RS designers are able to tap a vast range of interactive preference elicitation techniques, which can greatly improve the user experience. However, input data collection is not the only opportunity for recommenders to interact with their users. After a suitable set of recommendations has been generated, the results are presented to the user. In this *presentation phase*, user interaction considerations comprise, for example, recommendation list design, dynamic visualizations, and user control mechanisms, which are discussed below.

Recommendation List Design

The most basic form of interaction a recommender system can engage in after generating recommendations is displaying these recommendations to the user in the form of a list of items. Even though this might seem like a straightforward task, a number of design considerations can affect the user experience in this stage. Simple, yet often crucial aspects of recommendation list design include the label (e.g., "customers also bought"), the position of the list in the website's layout, the degree of descriptive detail for each list item, and the visibility conditions of the list (e.g., in case of a pop-up solution). While all these features can, from a system designer's perspective, be altered quite easily, they can have a strong effect on each user's acceptance of recommendations and engagement with the system. This is not least confirmed by the fact that, in real-world systems, such details are often honed to perfection over years of A/B testing [GH15].

In addition, another obvious point of consideration when designing a recommendation list is its size, i.e., how many items are displayed at any given moment. On the one hand, too many items might overwhelm users and thus defeat the purpose of employing a recommender system. On the other hand, too few options might limit the user's choice to the point that no satisfactory item can be found and trust in the recommender system suffers. In fact, the problem of finding a balance between too few and too many options has long been a topic of research in the decision-making literature [IL00; Sch04; SW07]. And, subsequently, an optimal number of options has also been shown to exist in the context of recommendations [BGW12; RH09; SW07]. System designers should keep this issue in mind yet be aware that there is no magical number of options that works equally well in every domain [RH09] or with every individual. For example, it is well known in the decision-making literature that some decision makers—called *maximizers*—want to consider the whole item space, while others—called *satisficers*—usually choose an option that they deem sufficient after only inspecting a few items [Sch+02] (see also Chapter 6).

Apart from rendering a simple list of item recommendations based on the supposed preference match between the items and the user's profile, considering other factors can also increase user engagement. For example, when the user's goal is not to find the perfect choice but to explore the item space, novelty or diversity can be incorporated into an accuracy-optimized list to improve the user experience [Her+04; MRK06] (see also Chapter 5).

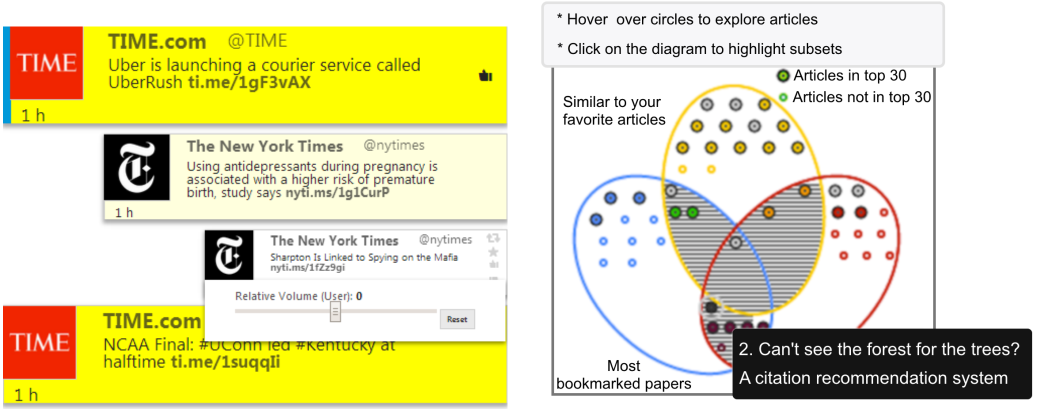
Finally, recommendation lists should not always be considered as isolated interaction components in a website's layout. Recently, recommender systems have become increasingly complex, and website operators have started to incorporate more and more recommendations. A standout example is the user interface of Netflix[4], which consists almost entirely of personalized recommendation lists. In such situations, designers should decide which items to present in which list and, possibly, how to group certain items together. For example, in the movie streaming domain, items can be grouped as "action movies recommended for you," which can help the user navigate through the personalized recommendation content more easily [CP08; NLF10]. In addition, sites such as Amazon often follow an approach to recommendation grouping that can help users understand how the recommendations were generated, for example, by clustering them into groups such as "customers who bought... also bought..." or "similar items" [SKR01]. Such grouping approaches are more common in practical applications, which seem to focus on these big-picture considerations more often than academic systems.

Visualization

While lists are still by far the most common form of presenting recommendations, several proposals, especially in academia, have considered how to visualize a set of item recommendations in a more engaging way. One of the easiest approaches to move from a simple list presentation to a more engaging format is to highlight certain items or vary the amount of item detail. For example, in the Twitter[5]-based social recommender by Waldner et al., important items in the user's Twitter feed are highlighted by either changing their size or coloring them (see Figure 2.6a) [WV14]. Furthermore, visualizing items to highlight certain features can be achieved by augmenting the displayed item information, e.g., by displaying tags that describe the item's importance to the user [SSV15], by adding reputation information (e.g., community ratings) [Kar+10], or by illustrating items with pictures or videos [NLF10; YY10].

However, all of these ideas still assume a basic list format to present recommendations, which is not necessarily the best solution in domains in which relations *between* items are an important feature that users might be interested in (e.g., in bibliographic networks). In such cases, diagrams and graphs can display not only each item's properties but also interrelations between domain entities. Recommended items can, for example, be connected in a graph via edges and thus clustered according to certain group attributes, such as genres in the movie domain [VS12]. Additionally, diagram-like visualizations can help users to understand how recommendations were generated. For example, in the approach by Parra et al., recom-

[4] https://netflix.com
[5] https://twitter.com



TIME.com @TIME
Uber is launching a courier service called UberRush ti.me/1gF3vAX
1 h

The New York Times @nytimes
Using antidepressants during pregnancy is associated with a higher risk of premature birth, study says nyti.ms/1g1CurP
1 h

The New York Times @nytimes
Sharpton Is Linked to Spying on the Mafia nyti.ms/1fZz9gi

Relative Volume (User): 0

TIME.com
NCAA Final: #UConn led #Kentucky at halftime ti.me/1suqqIi
1 h

* Hover over circles to explore articles
* Click on the diagram to highlight subsets

2. Can't see the forest for the trees?
A citation recommendation system

(a) Visual highlighting system by Waldner et al. [WV14]. ©Waldner et al.

(b) Venn diagram visualization by Parra et al. [PBT14]. ©Parra et al.

Figure 2.6: Academic approaches to recommendation presentation.

mendations are shown in a Venn diagram [PBT14], which groups items based on the recommendation (sub)strategy that they originated from (see Figure 2.6b).

While such structural forms of visualization can certainly express much more information about the recommendation space than simple lists, they lack a sense of quantitative relations between items. In contrast, 2D or 3D representations can place recommended items within a coordinate system to convey such links. Most commonly, such approaches are used in hotel or point-of-interest recommendations, with suggestions often placed on a 2D city map [Dal+14; Yah+15]. In more sophisticated proposals, 2D or 3D spaces are exploited to a greater extent, for instance, by showing recommended products in a cost/benefit coordinate system [BGG11] or by organizing music recommendations in a 2D mood space extracted from latent factor analysis [APO19]. Finally, in addition to positioning the recommended movies in a latent 2D space, the system proposed by Kunkel et al. allows users to manipulate the third dimension to express their likes or dislikes for certain preference "areas" of the visualization [KLZ15].

Explanations

Another way in which recommendations can become more engaging is by providing explanations that help the user understand why something has been recommended to them. Explanations for recommendations can have a multitude of goals [NJ17], which cannot be discussed in detail here. However, most commonly, explanations are employed to make users feel more confident in their decisions and to promote user trust through a better understanding of the recommendation process.

As already mentioned, the simplest form of explanation that can be provided for a list of recommendations is a label above the list along the lines of "because you watched. . . ," "trending," or "inspired by your browsing history." For specific types of recommender algorithms, explanations can go beyond vague hints as to the origin of a whole list, for example, in the case of knowledge-based recommendations. In this case, the algorithm's internal rules can be converted to natural language text to form an explanation of why a certain product was selected from a catalog [Fel+07]. However, for other types of recommendation approaches—namely, content-based and collaborative-filtering strategies—generating such expressive explanations is often difficult, as the recommendation logic can be highly complex. While labels such as "because you watched. . . " can be a useful hint for the user as to the recommendations' origin, they only represent a small part of the actual reasoning. Even for comparably simple approaches, such as CF schemes based on neighborhood models, a complex graph visualization is necessary to explain the algorithm's reasoning to the user, which is why such forms of explanations have, so far, been exclusively used in academia [ODo+08; SHO15]. For more complex models, such as latent factor models, explaining the recommendation logic directly can be even more difficult [RSZ13]. In these cases, an auxiliary explanation technique can be used post hoc to find plausible explanations, for instance, based on tags [BK14; GJG14; VSR09]; metadata, such as movie actors [SNM09]; or keywords [NFT13].

However, explanations do not just serve as a passive way of informing users. They can also allow users to interact with the system by giving feedback and correcting reasoning assumptions expressed in explanations. For example, if a recommendation for a pair of jeans were accompanied by an explanation stating "because you liked red shoes," the user could scrutinize this explanation [LAW14]. The user could then correct the given preference statement, e.g., by stating that they actually do not like red shoes, that they are no longer interested in them, or that they bought them for a friend. Such approaches can be found not in academic studies but also on websites like Amazon. Although slightly hidden in a sub-menu of the user's profile, Amazon customers can open a page where their personalized recommendations are explained to them—usually based on previous purchases— and they can then provide feedback, e.g., by instructing the system not to make recommendations based on a certain product in their purchase profile.

User Feedback and Control

As mentioned in the previous section, interactive explanations can be a useful starting point for recommender systems to empower users to provide feedback on system assumptions and, ultimately, take control of the recommendation process. However, explanations are not the only way users can exert control during the recommendation presentation phase.

For example, a simple, yet widely used form of recommendation feedback in real-world RSs is a like or dislike button next to recommended items, as used on platforms such as Spotify's[6] or Pandora's[7] personalized radios. One downside of such simple feedback approaches is, however, that it is not always clear to users what effects their actions have. In the example of Spotify's personalized radio, a dislike action removes the song in question but does not result in an immediate change to rest of the playlist, which could prompt users to stop using the feedback option even though it might have some positive (yet nontransparent) long-term effects. In academia, several attempts at including such feedback options have been made [KLZ15; NFT13; NT14], some of which have demonstrated that such controls can have a positive effect on user satisfaction [HKN12].

Going a step further, some systems, mostly from the realm of academia, allow users more fine-grained and powerful forms of control of their recommendations. Such interactive control mechanisms can, for example, take the form of filtering options for otherwise static recommendation lists, e.g., based on genres or tags [SKR02; SSV15]. Depending on the complexity of the presentation itself, more and more sophisticated forms of user manipulation are possible. In the graph-based visualization system by Chau et al., recommendations can be re-arranged spatially and categorized in groups [Cha+11]. Finally, as the ultimate form of user empowerment, some systems allow users to choose the underlying recommendation strategy themselves [DPM14; Eks+15; PB15; PBT14; Sha13]. However, as this form of user control might be too complicated and nontransparent for non-technologically minded users, such approaches are mostly found in academia.

Persuasion

Most of the previous interaction approaches have been described as mechanisms that *help users*, e.g., to make more informed decisions or receive better suggestions. However, recommender systems can also be a tool for system providers to persuade users, e.g., by directing them toward products with higher sales margins. To achieve such persuasive effects, system providers can rely on a range of interaction cues. For example, explanations can not only exhaustively inform users but also selectively omit negative information; focus on positive item characteristics, such as a high

[6] `https://spotify.com`

[7] `https://pandora.com`

community rating [GJG14; HKR00]; or use persuasive language [GL14; TM07], such as “recommended to you because this item will soon be discontinued.”

In contrast to such explicit forms of persuasion, a more discrete way of convincing users to choose certain items is to exploit well-known decision-making phenomena that can unconsciously bias users toward a recommended item. To this end, primacy and recency effects can be used, i.e., target items can be placed at the beginning or end of a list to ingrain them in the user’s memory [Fel+08]. Furthermore, a so-called decoy item, whose features and price are dominated by the target item, can be placed in the recommendation list to convince the user that the target item is objectively better than other items [TFI11]. And finally, system providers can also exploit anchoring effects, i.e., they can use recommendations as a stimulus that the user inadvertently remembers when making a decision later on [Ado+12; Ado+13; Cos+03]. For example, when entering a site, the user could be presented with a recommendation for a rather expensive pair of shoes. The assumption would then be that, even though the user might not purchase this particular pair of shoes, they might later be biased toward a more expensive item purchase (see Chapter 6 for more detail on such decision-making effects in the context of RS).

Proactive Recommendations

So far, the presented interaction methods have been based on the assumption that recommendations are provided as website components that users can click on if interested. However, with the growing use of smartphones, recommender systems can now also reside within mobile applications that can alert the user via notifications. This active form of interaction—often called *proactive recommendation*—means that recommender systems need to not only identify useful item suggestions but also decide whether newly discovered items are interesting enough to warrant disturbing the user via a notification.

While many modern applications (e.g., for news, shopping, or video and music streaming) include such a notification feature, academic research on this topic is still rather limited. Most approaches employ simple criteria to decide when to notify users about recommendations, for example, based on their current location or depending on whether they are currently using their phone [Höp+10; Wör+11]. However, few studies have focused on more elaborate approaches for automating this decision. One example is the work by Lacerda et al., whose shopping deal platform first pushes novel deals to a limited user base to estimate the deal’s suitability for use in subsequent notifications [LVZ13].

2.3.3 Summary

Today's RS designers can draw from a large variety of interaction techniques to create engaging and user-friendly interfaces for their recommenders. To elicit user preferences, system providers can passively monitor user behavior; allow users to rate items; or offer more interactive means of preference disclosure, such as dialog-based systems. Similarly, on the presentation end, recommendations can be displayed as a traditional list, in which case each item's visual presentation and potential explanatory information can vary. Alternatively, recommendations can be integrated into more elaborate graphical representations, such as 3D or graph models, or they can even be delivered as push notifications. As illustrated, these interaction approaches all come with their own advantages and limitations, which are mostly determined based on the user's effort while using the interface, as well as the designer's time spent creating and maintaining such mechanisms. In the end, system providers should be aware of the available interaction opportunities and the finer details of their chosen interaction methods, since even small changes can have a strong impact on users' decision-making processes and the user experience in general, as discussed in the following chapters.

3 Exploring New Domains With the User in Mind

Today, recommender systems are applied to a seemingly endless variety of domains and use cases, as explained in Section 2.1. However, with different domains come different requirements for the recommender system, not only because of data-related aspects (e.g., novel item features) but also because of changes in user behavior. For example, in one of the most thoroughly investigated RS application domains—movies—users exhibit relatively stable preferences, which makes the application of long-term user models viable. In contrast, in domains such as music recommendation, listening preferences can depend strongly on the user's current mood or emotional state, which requires the use of more complex prediction models. Finally, domains such as e-commerce are almost entirely dependent on the user's current context, i.e., their short-term shopping goals, as long-term preference models are often not available. Consequently, recommender systems must be tailored to the domain, and, most importantly, the effectiveness of a recommendation approach must be tested *per domain*, because there is no catch-all recommendation strategy that will satisfy user needs in every domain.

Thus, whenever recommender systems are considered for use in new domains, a thorough evaluation scheme is necessary to determine what recommendation strategies to apply. Evaluations should follow a holistic approach in which both provider goals and user requirements are assessed in offline and online settings, as well as in qualitative user studies (as detailed in Section 2.2). In this chapter, a case study is presented in which such a comprehensive evaluation approach was applied to gauge the usefulness of different recommendation strategies in a novel domain from a variety of perspectives. The rest of the chapter is fundamentally based on the insights from said case study [JJL16].

3.1 Domain and Use Case Description

For the case study presented in this chapter, an investigation was conducted into the design of a recommender system for a software tool called RapidMiner[8]. Similar

[8] https://rapidminer.com

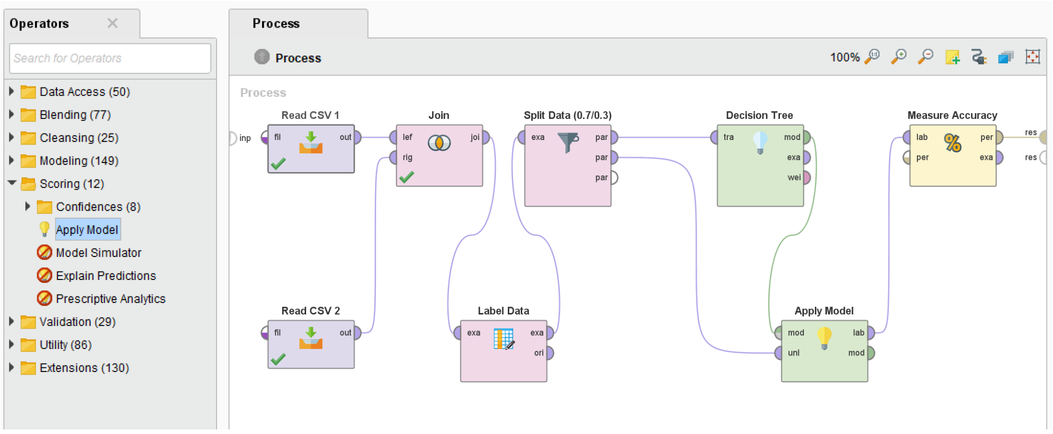


Figure 3.1: The RapidMiner user interface with its operator selection menu (left) and an example process model (right). In the example process, data is loaded from two CSV files, then joined, and a label assigned. A classification performance evaluation is conducted by splitting the data into training and test sets, learning a decision tree model, applying the model, and finally measuring the model's classification accuracy. In the operator selection menu on the left, hundreds of operators are available, nested in a tree structure up to six levels deep.

to tools such as scikit-learn[9] or WEKA[10], RapidMiner allows users easy access to powerful machine learning functions, pre-processing scripts, and output visualizations. However, not all data analysts who want to use machine learning techniques can program, and thus, RapidMiner offers users the ability to drag and drop machine learning functions into a visual development environment akin to workflow modeling. Each step in such a workflow (e.g., data loading/filtering or k-means clustering) is visualized as a graph node, called an *operator*. A machine learning workflow—also called a *process*—modeled in RapidMiner is shown in Figure 3.1.

Even though this form of modeling simplifies the machine learning analysis task tremendously for users who are not technically minded, finding the right operator for the task at hand can still be challenging. RapidMiner features a tree-based operator menu with hundreds of built-in operators and the option of adding custom operators based on plug-ins (see Figure 3.1). For experts, navigating such a complicated menu might become tedious over time, while for beginners, it can be overwhelming.

Consequently, the goal of the case study [JJL16] was to find a way of helping users to create machine learning workflows by recommending useful operators with the help of an additional UI widget inside RapidMiner. Such operator recommendations should be helpful to the user in their current task based on their partly modeled workflow, and the recommendation widget itself should be perceived as a worthwhile addition to the software by the user base. For privacy reasons, this goal must

[9] https://scikit-learn.org
[10] https://www.cs.waikato.ac.nz/ml/weka/

be accomplished based on a partly modeled workflow, i.e., without knowledge of the workflow history of the current user.

It is important to note that previous research studies have already investigated similar use cases, for example, in the context of business process modeling [KHO11; Li+14; LMR11; MM09] or Bayesian networks [Bob+13]. However, business process modeling entities are quite different from those in machine learning tasks, and none of the aforementioned studies have investigated the viability of more recent recommendation strategies.

3.2 Investigated Recommendation Methods

In the case study presented here, a range of recommendation approaches were tested. In addition to a trivial popularity-based baseline method, the investigated approaches fall into two categories: basic collaborative-filtering methods and context-aware methods. All of these more complex approaches rely on a training set of RapidMiner processes, which were downloaded from an online workflow-sharing site (details regarding the data sets are explained later).

Basic CF Methods. As a trivial collaborative-filtering method, a co-occurrence model (`CoOccur`; as described, e.g., in [SZ08]) was designed that ranks operators based on how often they appeared in the training set together with operators from the current partial workflow. In addition, a traditional CF approach in the form of a k-nearest neighbor scheme (`kNN`; see, e.g., [Sch+07]) was implemented. To this end, the current partial process is compared to the training processes in terms of their operator overlap. Given the k nearest neighbors in terms of this similarity measure, the neighboring processes can be scanned for potentially useful operators that are not yet included in the target process.

Context-Aware Methods. Given the specific use case of assisting users in completing a partially modeled process, it seems intuitive that recommendation strategies should consider which operators the user has been working on most recently, as these might provide an indication of the short-term modeling goals of the user. To this end, several context-aware methods were devised that take into account which path of operators in the modeling graph the user has most recently worked on.

As a first step, the two existing CF strategies were slightly altered so as to consider the user context. For the `CoOccur` method, a variant called `CoOccur-CTX` was created that only considers co-occurrence frequencies for operators from the current editing context of the target process. In contrast, a contextualized version of the `kNN` strategy, called `kNN-CTX`, featured a modified similarity function that gives more weight to the overlap of operators on the context path of the target process.

In addition to these modified versions of traditional CF methods, two more strategies were specifically devised to take advantage of the graph structure of RapidMiner processes. On the one hand, the `Link-CTX` strategy works similarly to `CoOccur-CTX`. However, instead of counting how often operators occurred together, this strategy ranks operators based on how often they were *linked* together in a training process, thereby capturing sequential relationships between operators. On the other hand, a method geared toward linked paths of operators, called `Chain-CTX`, was created to focus even more on the assumption that users are creating processes in a sequential manner. To this end, occurrence frequencies of chains—i.e., uninterrupted sequences of linked operators—are counted in training processes. Then, sub-chains of the current context path are compared to known chains to identify possible completion candidates.

3.3 Evaluation

As already mentioned, when evaluating a recommender system in a novel domain, researchers should not rely on a single experiment type or only a few quantitative performance measures. Instead, recommender systems—and the algorithms used within them—should be analyzed from a variety of perspectives, both quantitative and qualitative. In the case study, three evaluation types were chosen to accomplish the following evaluation goals:

1. **Offline Performance Analysis:** The above-mentioned algorithms were compared with each other in an offline setting based on historical RapidMiner processes to estimate their ranking accuracy based on typical *information-retrieval* performance metrics.

2. **User Study:** A laboratory study was conducted in which participants completed a workflow modeling task with the help of operator recommendations. In contrast to the first experiment, this laboratory study aimed at providing insights into how users are affected by recommendations in the RapidMiner UI, for instance, in terms of their efficiency or confidence.

3. **Online Experiments and Replay Analysis:** Finally, recommendations were deployed to the production version of RapidMiner, which allowed for a more

in-depth analysis of how a large user base would interact with the system. Additionally, the data collected in this field study was subsequently used to conduct more realistic offline "replay" analyses.

In the following sections, the above-mentioned analyses are described in detail.

3.3.1 Offline Analysis

The main goal of the offline evaluation was to gain an initial understanding of what kinds of algorithms could be applied to RapidMiner recommendations. To this end, a standard cross-validation performance analysis on historical data was performed to roughly estimate each algorithm's ranking accuracy and computational efficiency.

Data

Commonly, RS offline experiments focus on predicting user preferences by training algorithms on a set of user profiles and then testing the algorithm's ability to predict hidden preferences of a separate set of test users. However, as discussed in Section 3.2, for privacy reasons, it is unreasonable to assume that a complete history of workflow models for each user will be available to the recommender system. Instead, the recommendation component would likely only have access to the currently modeled process workflow within the RapidMiner environment.

Thus, a data set was created by downloading process models from a community website called myExperiment[11], on which users can publicly share their RapidMiner workflows. After removing duplicates and restricting the size of the processes, the data set comprised around 5,400 process models. Collectively, these processes contained over 700 different operators. However, on average, each process only contained nine operators (seven unique), giving a preliminary indication of the choice overload that RapidMiner users face when creating processes.

Evaluation Setup

The field of machine learning workflows is, in many aspects, different from other RS application domains, which must be considered when designing an offline evaluation setup. The following evaluation protocol was adopted: The data set described

[11] https://www.myexperiment.org/

in the previous section—comprising around 5,400 process models—was split randomly into 10 equal-sized bins to facilitate a 10-fold cross-validation scheme. That is, in 10 iterations, 90 % of the process models in the data set were allocated to the training set, with which the algorithms could train their models, and the remaining 10% of processes were used as a test set to measure the ranking performance of the algorithms. From each process model in the test set, a certain amount of operators were revealed to the algorithms. Based on this partial process model, the algorithms were then tasked with generating a ranked list of operator recommendations. These recommendations were compared to the remaining hidden operators of the given process to determine the ranking accuracy of the algorithms.

Several strategies were used to determine which operators to reveal in the partial model and which to use as the target operators to be predicted by the algorithms, i.e., the ground truth. One of these strategies, called `Given-N`, was designed to simulate a cold-start scenario, in which the user has just begun modeling the process. It revealed the "first" N operators of the process. As the insertion order of the processes in the data set is unknown, a heuristic was devised that selected the first N operators *from the longest path* of the process, as this was assumed to be the path that the user would most likely start and end their modeling task with. As ground truth, the immediately subsequent operator on the longest path was selected. A second strategy[12], called `Hide-Last-Two`, revealed all operators in the process model to the algorithms except for the last two operators. Again, the longest path was used as a heuristic for determining the most likely insertion order of the operators. As ground truth, the second-to-last operator was chosen.[13] This evaluation scenario was designed to mimic a recommendation query at the end of the user's modeling task in which help might be needed to complete a more complex process model.

In general, the goal of the evaluation setup was to simulate scenarios that RapidMiner users are likely to face regularly and that might cause difficulties for them when choosing the right operator. A similar user-oriented evaluation approach was also be applied with respect to the choice of evaluation metrics: The obvious goals of the recommender system would be (i) to determine useful operators for the current modeling situation and (ii) to rank these useful operators as highly as possible in the recommendation list. Thus, the recommendation measures Recall@10 and Mean Reciprocal Rank@10 (MRR) were chosen as hit rate and ranking metrics, respectively.

[12] Not all evaluation schemes from the original paper are reported, as the results were mostly similar.
[13] The last operator was not chosen as ground truth since it is nearly always a simple result output operator and not a machine learning function.

Table 3.1: Recall and MRR (@10) results for the offline evaluation; best result in bold.

	Hide-Last-Two		Given-1		Given-3		Given-5	
Algorithm	Recall	MRR	Recall	MRR	Recall	MRR	Recall	MRR
`CoOccur`	0.421	0.132	0.499	0.151	0.447	0.160	0.452	0.178
`MostFreq`	0.425	0.089	0.429	0.116	0.430	0.110	0.434	0.120
`CoOccur-Ctx`	0.468	0.158	0.499	0.151	0.468	0.183	0.495	0.187
`Link-Ctx`	0.690	0.419	**0.644**	**0.297**	0.690	0.389	0.706	0.391
`kNN`	0.725	0.222	0.440	0.106	0.592	0.164	0.679	0.173
`kNN-Ctx`	0.832	0.566	0.271	0.136	0.604	0.370	0.732	0.444
`Chain-Ctx`	**0.852**	**0.705**	**0.644**	**0.297**	**0.852**	**0.652**	**0.893**	**0.763**

Results

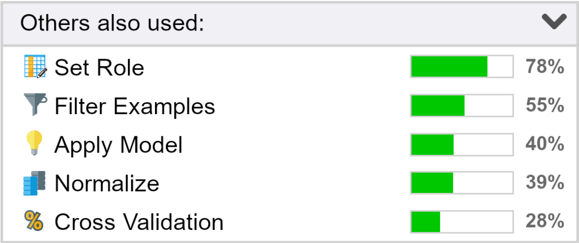
Table 3.1 shows the results of the above-described evaluation procedure. In the `Hide-Last-Two` setting, the contextualized methods were nearly always able to outperform the non-context-aware strategies with an average Recall of up to 85 %. This confirms that the user's current modeling context is an important indicator of their subsequent operator choices. Furthermore, the `Chain-CTX` method achieved the highest Recall and MRR values, which suggests that considering longer operator structures can be a worthwhile strategy. However, as the insertion order is only assumed from a longest-path heuristic, the results might be an overestimation of this method's true usefulness, which is investigated in more depth in the following sections.

In the `Given-N` evaluation scheme, algorithm performances were mostly comparable. Again, contextualized methods performed best. Interestingly, while the `CoOccur` method was not even able to outperform the popularity-based baseline in the `Hide-Last-Two` setting, it achieved slightly better results in this cold-start setting.[14] Finally, all methods were able to generate recommendations quickly; even the slowest method, `Chain-CTX`, needed less than 150 ms on average.

3.3.2 User Study

After gaining a quantitative understanding of how users might benefit from the various recommendation strategies in RapidMiner, the goal of the follow-up user study was to investigate the potential of using recommendations in this domain

[14] More detailed result statistics can be found in the original paper [JJL16]. As the focus of this thesis is more on *how* evaluations should be conducted with the user in mind and less on the actual results of the specific studies, detailed results are omitted at this point.



Others also used:

Set Role	78%
Filter Examples	55%
Apply Model	40%
Normalize	39%
Cross Validation	28%

Figure 3.2: RapidMiner user interface component used to display recommendations.

from a qualitative point of view. A controlled laboratory study offers researchers the most flexibility in observing how recommendations can affect the user experience, for example, in terms of the users' work efficiency or confidence when modeling processes.

Evaluation Setup

As a first step toward this laboratory study, a software architecture was devised with which recommendations based on the currently modeled partial workflow could be displayed in the RapidMiner UI. To this end, a REST[15]-based server environment was set up, in which a recommender algorithm was trained with the data from the first study. The algorithm could then be queried from the client's RapidMiner software via the REST interface to generate recommendations. Finally, the recommendations were displayed within the RapidMiner interface in a plug-in component, shown in Figure 3.2. This recommendation list component, from which users could drag and drop operators into their current workflow, was triggered to update itself every time the process model changed.

For the experiment, 28 university students were recruited, of which six had previous experience with RapidMiner. To keep conditions as controlled as possible, a between-subjects design was chosen. That is, half of the participants were able to use the recommendation component, while the other half used RapidMiner in its original form. Furthermore, to focus the experiment more on examining the effect of recommendations in RapidMiner in general than on identifying the best recommendation strategy, only one algorithm was used in the experiment. Due to its quick response times and simple, yet effective reasoning strategy, the `kNN` method was chosen for this task.

The following evaluation procedure was used for the laboratory experiment:

[15] Representational state transfer, see `https://tools.ietf.org/html/rfc7231`.

1. **Tutorial phase:** To account for individuals who had not used RapidMiner before, a script was devised based on which participants were instructed to build a basic machine learning workflow in RapidMiner as a training exercise.

2. **Main modeling exercise:** Given a partial process with four operators, participants were instructed to complete the process based on a detailed textual description of what the process should accomplish. During the exercise, a number of quantitative performance measurements were taken in the background (e.g., the amount of time each participant took and how many clicks they needed).

3. **Questionnaire:** After completing the modeling task, participants answered a questionnaire asking them to qualitatively assess different aspects of the modeling experiments, such as how confident they were in their own solution. Additionally, the questionnaire for the group that was aided by recommendations contained another set of questions, for example, geared toward how useful the recommendations seemed to them.

Results

As the methodology reflected a fundamentally different evaluation perspective than the initial offline tests, the laboratory study provided additional valuable insights into how recommendations in RapidMiner affect users. Most notably, participants who were aided by the recommendation component were significantly faster (on average 2 m compared to 5 m) and needed fewer clicks to accomplish the modeling exercise. Note that even though this finding is quantitative, it could not have been observed via an offline study, due to the absence of observable user behavior, or an online study, due to the inability to control experiments in a way such that two groups model exactly the same process under the exact same conditions.

Additionally, the answers to the post-task questionnaire were analyzed. However, no significant differences between participants aided by recommendations and regular users were found, for example, in terms of perceived task complexity or the participants' confidence in their own solution. Only in terms of the perceived comprehension of the task, a difference was observed; recommendation users thought that they understood the task more clearly. Furthermore, as could be expected, novice users found the recommendations more useful than experienced users. Interestingly, however, the average relative improvement in terms of task completion time was slightly higher for experienced users, which could indicate that experienced users underestimated the usefulness of the recommendations. Yet again, this latter point shows that both qualitative and quantitative measurements are necessary to identify such deviations between user perceptions and actual user behavior.

3.3.3 Online Experiment and Replay Analysis

As a final step in the evaluation plan, an online experiment was carried out in collaboration with the creators of RapidMiner to identify how well the recommendations would be accepted by a large user base. To this end, a recommendation architecture similar to the one described in the laboratory study was deployed so that the recommendation UI component could be integrated into the RapidMiner software. The recommendation feature was then released to the whole user base, and users who updated to the newest version were made aware of the new functionality by a pop-up. This pop-up indicated that users had to agree to a privacy policy if they wanted to use the new feature, as the recommendation server needed access to the partial process model currently developed in the user's modeling environment.

As with any online test in a real-world system, the provider company imposed certain restrictions. The company favored the `CoOccur` strategy, because of its low computational and memory requirements. Furthermore, instead of a more controlled A/B test, the recommendation component was pushed to the whole user base, ruling out the possibility of comparing user behavior with and without tool assistance. However, as mentioned earlier, restrictions like these are commonplace and represent a trade-off that researchers must accept to access valuable online testing data.

Analysis of Collected Online Usage Data

After the tool's initial deployment, usage data was collected over a few weeks. To avoid privacy concerns, the data shared with the research team did not contain any user-specific information. Instead, to allow an analysis of the user interaction behavior, snapshots of processes were collected each time a process was altered. That is, for each completed process, a list of partial snapshots was recorded on the server side. Additionally, every time a recommendation request was made, the recommended operators were recorded along with whether the user dragged any of the recommendations into the workflow. The research team filtered the collected data slightly, e.g., to remove very small, and thus likely unfinished, processes, resulting in a data set that contained around 3,000 processes with an average size of 11 operators. For each process, an average of 43 snapshots was recorded, and overall, around 80,000 recommendations were made by the plug-in.

Based on the collected data, a number of analyses were conducted to gauge the user base's acceptance of the recommendations. First, the rate of applied operator recommendations was calculated—i.e., the percentage of cases in which an operator from a list of recommendations was dragged into the user's currently modeled process. This acceptance rate was measured at a promising 7.8 %, which means that roughly every 13th operator insertion originated from a recommendation list.

However, looking solely at how often recommendations were applied might give a false indication of their usefulness. Thus, a second analysis was conducted in which the rate of removals of recommended operators was measured. After each insertion of a recommended operator in the data set, the subsequent snapshots were scanned to identify whether the operator was later removed. This recommendation removal rate was calculated to be 34.2 %, which is slightly higher than the removal rate of manually inserted operators (24.5 %). However, a reason for this higher removal rate might be that users were actively experimenting with the new UI element to test its capabilities.

Replay Analysis Based on New Data

Finally, complementary to the online test, the newly collected data was used to conduct an additional, more realistic offline performance analysis. Since the new data set contained snapshots of each process model, the insertion order of the operators could be ascertained. Consequently, the algorithms' performances could be measured by "replaying" the process creation scenario as it occurred in real time. To this end, the algorithms from the initial offline study were again compared against each other with the new data set. To measure Recall and MRR in the test set, the operators of each process were gradually revealed to the algorithms in the order in which they were inserted by the respective user, and the next operator in line to be inserted was used as ground truth. Furthermore, at this stage, contextualized methods, such as `Link-CTX`, also used the most recently inserted operators as the modeling context instead of relying on the longest path heuristic.

Some of the methods that performed particularly well in the initial offline tests performed quite poorly in this replay evaluation—specifically the previously best-performing `Chain-CTX` strategy. However, other methods, such as `kNN` and `kNN-CTX`, performed well in both evaluation settings. Additionally, the initially quite poorly performing `CoOccur-CTX` achieved the highest overall MRR score in the replay analysis. Once more, these results demonstrate that different evaluation methods can lead to vastly different insights.

3.4 Implications

The case study in the domain of machine learning workflow recommendations highlighted several important factors that should be considered when investigating the application of recommender systems in novel domains. Offline experiments yielded initial insights into the comparative performance of the algorithmic strategies and revealed that in the specific domain, the user's context situation should be considered to achieve accurate results. The controlled environment of the laboratory study, in contrast, allowed for an in-depth analysis of how the proposed recommender system would affect users in terms of their work experience and productivity. Finally, the performance ranking of the online data analyses showed that not all algorithms performed as well as could be expected based on the preliminary offline experiments. This finding highlights once more that RS evaluation should not focus on a single experiment type but instead a combination of different approaches.

4 Considering the User When Evaluating

The systematic evaluation presented in the previous chapter illustrated that user studies are crucial to gain a qualitative understanding of how recommender systems affect users. However, as such studies usually deal with relatively fewer data points than offline evaluations, which feature millions of samples, it is important to be aware of possible biases introduced by the individual participants.

For example, in a laboratory study on recommendations in RapidMiner, users might differ strongly in the degree of previous knowledge about the software. Given an incidentally skewed distribution of experts and novices within the treatment groups, the results of such a user study might be biased, leading to incorrect conclusions about algorithm performance. Thus, well-known potential sources of biases, such as varying levels of domain expertise, should be accounted for. In the RapidMiner study, this was done by asking participants to self-report their expertise and checking that none of the treatment groups were dominated by either experts or novices.

However, not all sources of biases are so obvious. For example, similar to the way that results can be biased because of varying degrees of domain expertise, participants might also cause a bias because of their level of *familiarity* with the recommended items. If, for instance, a recommender system in a study about movie recommendations were to recommend an obscure Spanish silent movie, most participants would likely rate it poorly, even though the movie could be a secret gem that they might enjoy. If by chance, however, a participant had watched this movie before, the likelihood of a good rating would be higher due to that person's familiarity with the item.

Note that this bias can only occur in user studies in which participants are asked to state their "potential" enjoyment of the recommended item, which is a common experimental approach in the field. This is because, in domains such as movies, it is impossible to allow study participants to watch the whole movie and then rate it. Instead, they are usually asked to rate the item according to their hypothetical enjoyment based on supplementary data, such as item descriptions, metadata, or trailers.

Based on this problem setting, the research question investigated in the study presented in this chapter [JLJ15b] was the following: *Are study participants' ratings of movie recommendations biased based on whether they have previously watched the respective movies?*

4.1 Study Setup

To investigate this question, a three-step study design was adopted, similar to previous studies in the field [CGT13; Eks+14; Sai+13]. In the first phase, participants could enter their movie preferences into the system. To this end, they had to rate 15 movies on a scale of 1 to 5 stars with half-star increments. Next, each participant was presented with 10 movie recommendations, for which supplementary information was provided, such as the actors, the genre, a plot synopsis, and a trailer. For each of these recommendations, participants had to assign a star rating and state whether they had watched the movie previously. Finally, the participants answered a questionnaire about the recommender system as a whole, for instance, regarding recommendation diversity, surprise, transparency, and ease of use.

4.1.1 Algorithms

The study followed a between-subjects design with five widely used algorithms for comparison, from which one was randomly assigned to each participant. The algorithms employed in the experiment were the following:

- A non-personalized popularity-based baseline;
- `SlopeOne`, a simple and efficient CF method [LM05];
- `FunkSVD`, a CF approach based on matrix factorization [Fun06];
- `Bayesian Personalized Ranking` (`BPR`), a more recent CF strategy that optimizes item rankings [Ren+09]; and
- A content-based approach utilizing IMDb[16] content data.

4.1.2 Data Set

For training purposes, the above-mentioned CF methods require not only the participants' movie ratings but also a set of community ratings to learn a model. To this end, the well-known MovieLens 10M data set[17] was used, which contains 10

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

[17] `https://grouplens.org/datasets/movielens/10m/`

million movie ratings. A subset was created by removing obscure movies, resulting in around 400,000 ratings. For the remaining movies in the subset, content data was crawled from IMDb to (i) calculate movie similarity for the content-based approach and (ii) display descriptive information about the recommended movies to participants.

4.1.3 Participants

The experiment was conducted on the crowdsourcing platform Mechanical Turk[18], where work-from-home employees offer their time to potential employers in exchange for money. Recently, hiring crowd workers to participate in online studies has become increasingly common [BTG18]. In the study, the compensation for each worker was set to US$ 1.50, and since the platform has a reputation of attracting careless workers, a few countermeasures were taken to ensure that participants were paying attention. For example, non-existent movies were included in the initial preference elicitation phase. If participants rated one of these fake movies, they were excluded from the study. Overall, 175 participants accepted the task, of which 96 passed all attention checks, resulting in around 20 participants per treatment condition.

4.2 Results

Even though the performance results of the individual algorithms are not the focus of this thesis, a short summary of key insights is presented here as context for the following analyses. Afterward, observations regarding the familiarity bias are discussed.

4.2.1 General Observations

In terms of the average ratings assigned by the participants to the recommended movies, surprisingly, the non-personalized recommendations of the popularity baseline were the clear (and statistically significant) winner (see Figure 4.1). The `BPR` approach, which is also known to recommend rather popular movies, placed second. This might indicate that in the given study setup and with the sample of crowd workers, item popularity plays an unusually important role in participants' perceived recommendation accuracy. The content-based strategy and the other two CF approaches performed notably more poorly, which is surprising, as `FunkSVD` is

[18] `https://www.mturk.com`

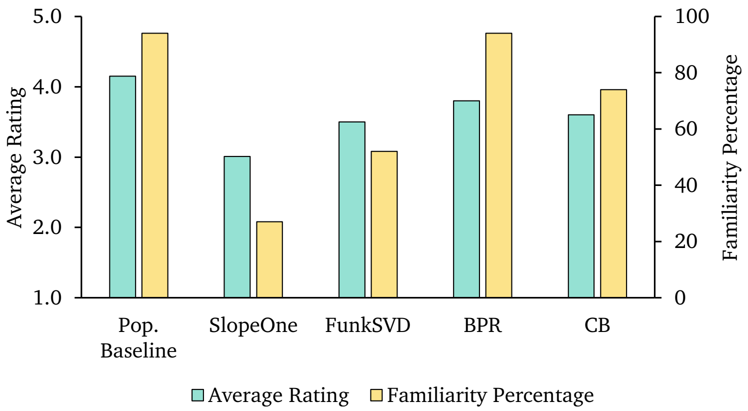


Figure 4.1: Results of the user study. The diagram illustrates, side by side, the average ratings assigned by the participants to the recommended movies, along with the percentage of recommended movies that had previously been watched. The correlation between the two data series is $\rho = 0.6$.

specifically known as a consistently well-performing strategy in terms of predicting what users like, particularly in the movie domain.

Additionally, based on the participants' answers to the final questionnaire, qualitative observations regarding the algorithms' levels of diversity, surprise, and perceived transparency can be made. In terms of perceived transparency, the algorithm ranking mirrored the accuracy ranking. In contrast, in terms of diversity and the surprise factor, the BPR approach and the popularity baseline scored the lowest, thus resulting in a nearly reversed ranking. A possible interpretation of these results is that—while approaches that rely on more popular movie recommendations cannot provide the most explorative experience—their reasoning is easy to understand for users. This higher level of perceived transparency, in turn, might ultimately be a contributing factor to the algorithms' higher average accuracy ranking.

4.2.2 Effect of Familiarity Bias

While the popularity of recommendations might have contributed to the higher accuracy ratings for algorithms such as BPR due to the general mass appeal of the recommended movies, another reason participants perceived these recommendations as a better fit could be rooted in the *recognizability* of such popular movies. In fact, coincidentally, the two strategies that received the highest ratings also recommended movies that participants had already watched in more than 90 % of the cases. In contrast, 74 % of the recommendations of the content-based approach had been watched by the participants and only 52 % and 27 % for FunkSVD and SlopeOne, respectively. These results are, again, shown side-by-side in Figure 4.1.

Noticeably, the ranking of the algorithms according to this familiarity percentage falls in line with the ranking according to the average accuracy ratings. A possible interpretation could be that, in the given study setting, in which participants' ratings were not based on the actual consumption of the movie but on an informed decision based on the provided description, the participants might have inadvertently assigned higher ratings to movies that they had already watched. In contrast, movies that were new to them (but might actually be a suitable fit for their preferences) could have been at a disadvantage, as reading a description or watching a trailer cannot replace the positive sensation of actually watching the whole movie. Consequently, as supported by the qualitative results from the questionnaire, unfamiliar movies might have been perceived as slightly more surprising or niche than they objectively "deserved," which would have introduced a familiarity bias into the data.

Based on the collected rating data, another more detailed analysis was possible. When splitting the participants' perceived accuracy ratings into those for movies that had and had not been previously watched, an interesting effect was observed. For some algorithms, the difference in accuracy for familiar and unfamiliar recommendations was not very high, while for others—specifically `FunkSVD`—the accuracy for unfamiliar movies was much lower. This corroborates the observation that `FunkSVD` tends to recommend niche movies, which might be more prone to receiving a poor rating if the only form of familiarization is a description and trailer. Finally, as expected, the algorithms' familiarity levels correlated highly with the assigned ratings ($\rho = 0.60$), as well as with further qualitative variables from the questionnaire, such as intention to reuse ($\rho = 0.40$), tendency to recommend the system to a friend ($\rho = 0.55$), and the overall perceived preference match of the system as a whole ($\rho = 0.68$).

4.3 Implications

Overall, the results show that item familiarity can be a biasing—and potentially confounding—factor in user studies on recommender systems. Care should be taken to account for the user's familiarity with each of the recommended items when analyzing the collected data. However, given the potential severity of this bias in any given domain, accounting for it might not be enough. In fact, a similar study was recently conducted in the domains of music and movies [Loe+18], whose results are in line with the study presented here [JLJ15b]. Overall, evidence indicates that the results of studies that ask users to rate recommended items without consuming them should be considered carefully, as there might be hidden biases.

To avoid such biases, study designs should take them into account from the beginning. For example, in the music domain, the study setup could allocate extra time to let users actually listen to each recommended music track in its entirety to avoid a familiarity bias. In other domains, such as movies, consuming the recommended items during the study is often not feasible. Thus, the only option that remains is for researchers to be aware of the bias and control it as much as possible.

5 Modeling Multifaceted User Preferences

Previous chapters have shown that on the one hand, certain user characteristics can introduce biases into study results and that on the other hand, a number of RS *design factors* can affect the user experience. For example, in the RapidMiner evaluations described in Chapter 3, algorithms that took domain characteristics into account (e.g., in terms of the current user context) achieved better results.

However, designing algorithms with the single goal of providing *accurate* recommendations might not actually lead to the most enjoyable user experience. For example, in a music recommendation scenario, a simple approach that recommends only Michael Jackson music might generally fit the preferences of certain users, but, over time, these users could easily become tired of such one-dimensional recommendation lists. Instead, users often prefer a more *diverse* set of recommendations, which provides a varied listening experience as well as the potential for discovery of *novel* songs and artists.

As music is not the only domain in which considering multiple quality aspects is necessary to provide an enjoyable user experience, over the years, numerous algorithmic proposals sought to integrate alternative quality factors into recommender systems [AK12; BS01; JW10; Kap+15; Oh+11; Rib+14; Sai+13; Shi+12; VCV11; ZH08; Zha+12; Zho+10; Zie+05]. However, one downside of nearly all of these proposals is that they are fundamentally built on a specific recommendation algorithm, which is then extended to also consider, e.g., diversity or novelty. Consequently, these approaches are often limited to a certain domain or recommendation use case.

In contrast, the approach presented in this chapter [JJL17], called `Personalized Ranking Adaptation` (`PRA`), was designed as a versatile post-processing strategy that can be used to re-arrange recommendation lists produced by any state-of-the-art recommendation approach, for example, to make them more diverse. In fact, the proposed approach goes one step further. Keeping with the example of diversity, `PRA` is built on the assumption that *more* diversity is not *always* better. Instead, the algorithm uses *each individual user's* previous preferences as a template to determine the appropriate level of such quality factors in the recommendation list. For example, in the music domain, the algorithm could try to achieve the same level of artist diversity in the recommendation lists as in each user's individual listening history.

Depending on the domain, such adjustments cannot be achieved without trading off a certain amount of accuracy. For example, in a movie recommendation scenario, changing the popularity characteristics of the recommendation list can increase the risk of recommending obscure movies, which users might not be willing to try. Thus, experiments are necessary to determine the proposed strategy's potential to keep accuracy high while achieving the desired effect in terms of optimizing alternative quality measures. This chapter describes the algorithm's design and the results of empirical evaluations to test its effectiveness.

5.1 Algorithmic Approach

The algorithm presented here mainly sets itself apart from previous strategies by its ability to consider individual user tendencies toward, for example, diversity. Thus, before the actual optimization procedure is explained, an overview of how such tendencies can be quantified and utilized is provided.

5.1.1 Design Rationale

Generally, the idea behind the PRA strategy is to assess a user's tendencies toward certain quality factors so that the recommendations for that specific user may reflect these tendencies. In the best case, such user tendencies could be captured by asking users directly about, e.g., their desired level of diversity in a given situation, as proposed in several studies mentioned in Section 2.3.2. However, as many of today's recommender systems do not support elaborate forms of user interaction, these tendencies must be captured automatically instead. For this purpose, the system must extract the user's tendencies based on historical data, such as the user's set of top-rated items or the playlist they listened to most recently. By analyzing the characteristics of items in this *preference sample set*, the algorithm can then estimate the user's tendencies toward certain quality factors, such as diversity.

In the next step, the proposed PRA strategy takes an accuracy-optimized list generated by any state-of-the-art recommendation method as a starting point and rearranges its items so that the list's quality characteristics match the user's tendencies more closely. For example, if a user mostly assigned high ratings to less popular movies in the past, PRA could move such movies up the list to match the user's tendency toward less popular movies.

To quantify user tendencies based on a sample set of items, different approaches are possible. For one-dimensional quality factors, such as item popularity, the mean and/or standard deviation of the items in the set can be a meaningful indicator. However, this approach only captures how popular the items are overall. To better compare the distribution of popularity levels among (i) the user's sample items and (ii) the recommendation list, a measure such as the earth mover's distance (EMD) [LO07] can be used. The EMD calculates the difference between two distributions based on the number of moving operations required to transform one distribution into the other. Lastly, depending on the quality measure, more tailored quantification approaches might be necessary. For example, to estimate the level of diversity, the inverse of the intra-list similarity (ILS) measure [Zie+05], which is based on the pairwise similarities of items in a list, can be a suitable choice.

5.1.2 Procedure

Based on the above-described general goal of the algorithm, the following formal definitions can be made. For each user, a preference sample set S_u is used in conjunction with a function $\mathcal{P}(S_u)$, which calculates the level of a given quality factor, such as diversity, within the set of items S_u. Similarly, the top-n recommendation list is defined as T_u, based on which the function $\mathcal{R}(T_u)$, again, quantifies the specific quality level. Formally stated, the goal of the algorithm is to minimize the difference

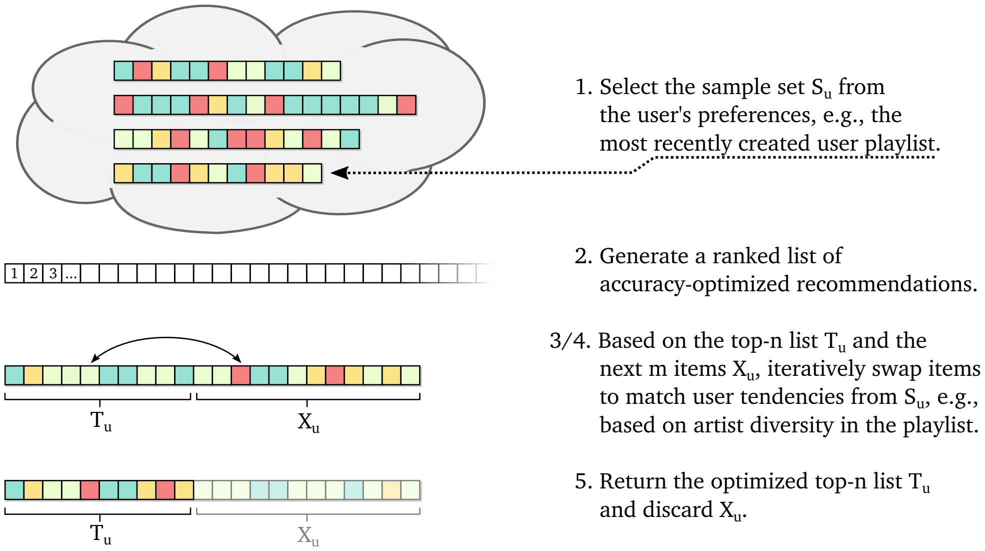
$$\min d(\mathcal{P}(S_u), \mathcal{R}(T_u)),$$

where $d(.)$ is a function that returns the real-valued difference between the two tendency-quantifying functions $\mathcal{P}(S_u)$ and $\mathcal{R}(T_u)$.

For example, given the quality factor *item popularity*, $\mathcal{P}$ and $\mathcal{R}$ could be set to capture the mean popularity of the user's top-rated items and the recommendation list, respectively. Then, d could simply be the absolute difference between the two values, so that PRA effectively tries to match the mean popularity of the items in the recommendation list with the mean popularity of the user's sample items.

To this end, the post-processing scheme executes the following steps each time a recommendation list is needed for a specific user (see also Figure 5.1, which displays these algorithm steps based on the example of optimizing for diversity tendencies):

1. Determine the user's sample set S_u based on their interaction history: for example, the music playlist they most recently listened to or simply their top-rated items.



1. Select the sample set S_u from the user's preferences, e.g., the most recently created user playlist.

2. Generate a ranked list of accuracy-optimized recommendations.

3/4. Based on the top-n list T_u and the next m items X_u, iteratively swap items to match user tendencies from S_u, e.g., based on artist diversity in the playlist.

5. Return the optimized top-n list T_u and discard X_u.

Figure 5.1: Steps in the optimization procedure of the PRA algorithm based on the example of diversity optimization (based on [JJL17]). Boxes in the cloud at the top represent songs organized in user-created playlists. The colors of the boxes indicate the respective artists.

2. Generate a ranked recommendation list with the help of an accuracy-optimized baseline algorithm.

3. From this ranked list, designate the first n elements as T_u, where n is the desired length of the final recommendation list to be displayed to the user. Additionally, define the exchange list X_u as the next m items in the accuracy-optimized list immediately after T_u.

4. Iteratively exchange items between T_u and X_u to create a modified recommendation list T_u^* that minimizes the optimization goal $d(\mathcal{P}(S_u), \mathcal{R}(T_u))$. Per iteration, one item from T_u is swapped with one item from X_u. The two items selected are those that if exchanged, would yield the greatest improvement in the optimization goal function. See also Algorithm 1 for further details about this specific step.

5. If the goal function cannot be improved further or if the error reduction rate Δe falls below a certain threshold e_t, return T_u^* as the list of recommended items to be displayed to the user.

Algorithm Configuration. The algorithm can be configured based on the use case at hand to achieve the desired effect. First, by determining a suitable *sample set* S_u, an engineer with domain knowledge can make sure that the algorithm accurately captures the user's tendencies toward the individual quality factors. Second, choos-

Algorithm 1: PRA variant which terminates when the improvement falls below a certain threshold e_t (based on [JJL17]). In each iteration, all possible swaps between T_u and X_u are evaluated, and from those, the swap with the strongest improvement of the desired quality aspect is executed. The notation $T_u[i \rightarrow j]$ indicates a list T_u in which item i is swapped with item j from X_u.

```
input  : real $e_t$; Array $S_u$, $T_u$, $X_u$
output : Array $T_u$ as $T_u^*$

repeat
    $e_{base} \leftarrow d(\mathcal{R}(T_u), \mathcal{P}(S_u))$
    $\Delta_e \leftarrow 0$
    for $i \in T_u$, $j \in X_u$ do
        $e_{new} \leftarrow d(\mathcal{R}(T_u[i \rightarrow j]), \mathcal{P}(S_u))$
        if $e_{base} - e_{new} > \Delta_e$ then
            $i^* \leftarrow i$
            $j^* \leftarrow j$
            $\Delta_e \leftarrow e_{base} - e_{new}$
        end
    end
    if $\Delta_e \geq e_t$ then
        $T_u \leftarrow T_u[i^* \rightarrow j^*]$
        $X_u \leftarrow X_u[j^* \rightarrow i^*]$
    end
until $\Delta_e < e_t$;
```

ing a suitable *size of the exchange list* X_u can limit accuracy losses by preventing the algorithm from swapping items from low positions into the final list. For the *stopping criterion*, different choices are possible. The recommendation list T_u^* could, for example, be returned when a certain number of swaps have been executed, when the error reduction rate Δ_e falls below a certain threshold e_t, or when no further improvement is possible.[19]

Balancing Multiple Goals. To simultaneously address multiple optimization goals (e.g., diversity *and* popularity), the error calculation (of e_{new} in Algorithm 1) must consider multiple distance functions $d(.)$ at the same time. This can be done by aggregating the distance values of the individual quality criteria, for example, in a simple sum or in a weighted or normalized way, depending on the domain requirements. Additionally, more complex error aggregation schemes are possible, for instance, by allowing the improvement of one factor only if it does not affect another. In the experiments conducted for the original paper [JJL17], a normalized sum was chosen when optimizing multiple goals at the same time.

[19] Even in the last case, the resulting recommendation list T_u^* might not represent the optimal solution due to the greedy nature of the swapping mechanism described in Algorithm 1 (see discussion in Section 5.2.2).

5.2 Evaluation

To understand how effectively the proposed algorithm matches user tendencies with recommendation list characteristics, a set of empirical evaluations was conducted. In the course of these experiments, different optimization targets were tested including individual as well as multiple simultaneous quality dimensions. Furthermore, the algorithm was compared to an optimal re-ranking strategy and to a previously proposed post-processing method from the RS literature.

Data. For the empirical evaluations, two data sets from different domains were used. Specifically, the MovieLens movie rating data set, which was also employed in the study described in Chapter 4, with roughly 1 million ratings was used, as was a music data set from last.fm[20], which contained around 3,500 playlists with 22,000 tracks created by 511 users. The latter domain was chosen because it offers a wide range of quality factors, such as tempo and artist diversity, that might be considered important by users of a music recommender system.

Baseline Algorithms. As the `PRA` method requires an accuracy-optimized list for re-ranking, a number of state-of-the-art recommender algorithms had to be chosen as baselines whose outputs would then be post-processed. For the movie domain, two methods discussed in Chapter 4, `FunkSVD` and `BPR`, were selected. Additionally, the more recent `Factorization Machines` (`FM`) technique [Ren10], which is based on a combination of feature engineering and matrix factorization, was chosen. In the music domain, a `kNN` scheme based on cosine similarities between playlists was applied. Finally, a simple algorithm called `Collocated Artists Greatest Hits` (`CAGH`) was selected. This strategy recommends the greatest hit songs of artists similar to those artists in the current user's playlist, an approach shown to work particularly well for shorter playlists [BJ14].

Evaluation Procedure. For each domain, the available data was split user-wise into five bins to facilitate a standard five-fold cross-validation procedure. The recommendation list length was set to 10 for all experiments, entailing $|T_u^*| = 10$. As quality factors to optimize in both domains, list diversity, item popularity, and item release dates were chosen. To determine diversity, the inverse of the already mentioned ILS measure [Zie+05] was used. In the movie domain, item similarity was calculated based on TF-IDF vectors [Aiz03] of plot descriptions, and in the music domain, artist information was used. To determine item popularity, the number of

[20] `https://last.fm`

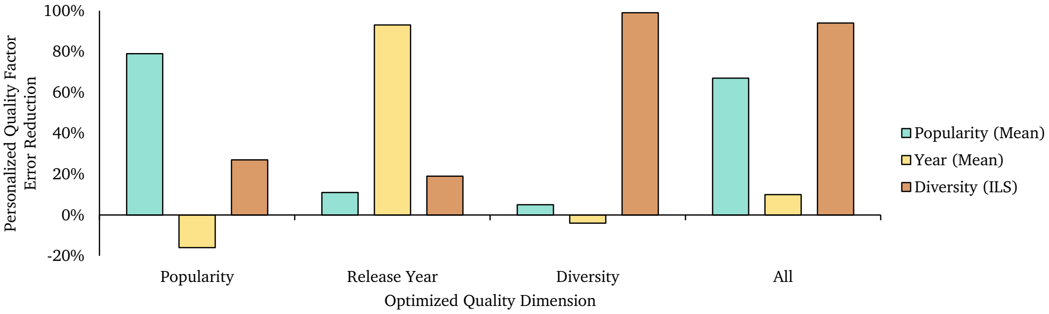


Figure 5.2: Selected results of the effectiveness analysis of the `PRA` algorithm for the MovieLens data set with the `FunkSVD` baseline method.

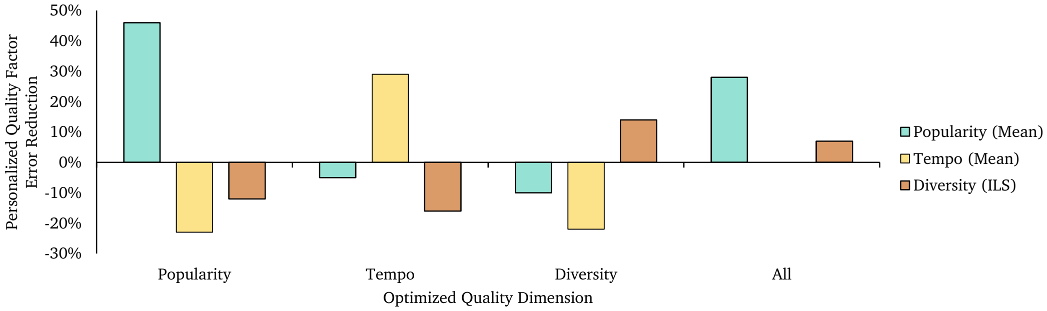


Figure 5.3: Selected results of the effectiveness analysis of the `PRA` algorithm for the last.fm data set with the `kNN` baseline method.

ratings/playlist occurrences per movie/song in the data set was taken as a reference, which is common practice in the literature [Jan+15]. For the music domain, the quality factors of tempo and loudness were chosen as additional optimization goals, as these are important quality factors in music playlists [SC12].

5.2.1 Effectiveness Analysis

Figures 5.2 and 5.3 show `PRA`'s effectiveness results for the MovieLens and last.fm data sets, respectively. For the figures, the algorithms `FunkSVD` and `kNN` were chosen as examples. However, with other baseline algorithms, similar results were obtained. Not all experimental results for all aforementioned combinations of quality factors are shown, as most of them followed a similar pattern.[21]

The figures show the algorithm's potential in reducing the difference (or *error*) between user tendencies and list characteristics. For example, the value of the left-most bar in Figure 5.2, 79 %, indicates that `PRA` was on average able to reduce the difference between each user's popularity tendency (according to their top-rated movies) and the item popularity in their recommendation list by 79 % compared to the original recommendation list produced by the baseline method `FunkSVD`. The

[21] The detailed results for experiments with all baselines and quality factors can be found in the original paper [JJL17].

group label “popularity” in the figure implies that for the bars displayed in this group, the optimization target of PRA was item popularity.

Continuing this line of thought, the other two bars in this group show that while optimizing for popularity, PRA also slightly improved the recommendation list’s diversity characteristics. However, in terms of the average release year of the recommended movies, the list characteristics actually diverted slightly from the user tendencies. This trend can also be observed for the other optimization goals and the music data set. That is, when optimizing for a specific quality factor, the match between user tendency and list characteristics with respect to this factor can be strongly improved, while the other factors either also marginally improve or slightly deteriorate.

However, when looking at the last group, in which all quality dimensions were optimized at once, all factors improved, although not as strongly as when optimizing them individually, which is to be expected. An exception is the tempo quality dimension, which did not improve when all factors were optimized together, indicating that there might be mutual interference between two or more factors in the music domain that prevents the algorithm from improving all factors simultaneously.

Surprisingly, PRA increased the Recall values in all cases for the already well-performing kNN method on the music data set (not shown in the figure). A possible interpretation is that the quality factors optimized by PRA were not yet captured by the underlying kNN baseline. In the movie domain, on the other hand, the PRA method led to virtually no change in accuracy, which confirms that the chosen exchange list size $|X_u|$ of 20 was well suited for maintaining accuracy.

5.2.2 Comparison With an Optimal Re-Ranking Strategy

As previously mentioned, the PRA algorithm cannot guarantee an optimal solution due to its greedy swapping strategy. Thus, an experiment was conducted to estimate how closely PRA’s performance can match an optimal solution. To this end, an optimal strategy was implemented that simply investigated each of the possible $(|T_U| + |X_u|)!$ permutations of the recommendation and exchange list item space.

Figure 5.4 shows the results of this experimental evaluation. For each group in the figure, a different optimization target was investigated. The first two bars show the results for PRA and the optimal strategy, respectively, with an exchange list size of 10. The right bar of each group shows PRA’s performance with an exchange list size of 20. Executing the optimal strategy with an exchange list size of 20 was not feasible due to the prohibitive runtime requirements.

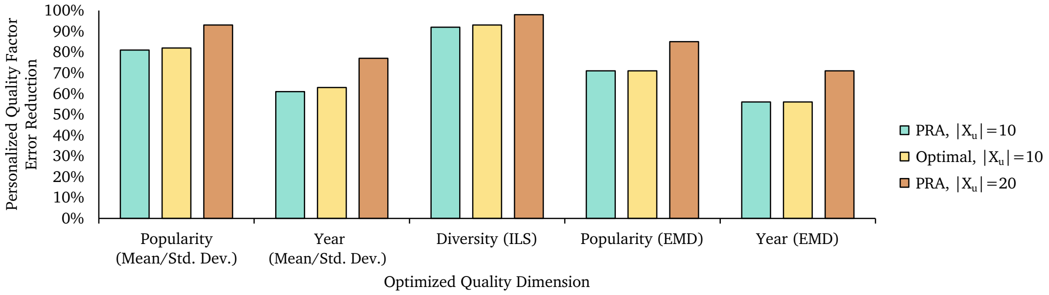


Figure 5.4: Comparison of PRA with an optimal re-ranking strategy for the MovieLens data set with BPR as a baseline strategy.

Based on the results, PRA was able to come very close to the performance of the optimal strategy in every quality dimension. At the same time, PRA was much more effective when using a larger exchange list size of 20, which was not even possible with the optimal strategy.

5.2.3 Comparison With an Alternative Method

Comparing PRA with other methods that try to optimize alternative quality factors of recommendation lists, reveals that nearly all previously proposed methods do not consider the user's individual tendencies. That is, they assume that, for example, *more* diversity is *always* desirable [AK12]. In contrast, the approach by Oh et al., called Personal Popularity Tendency Matching (PPTM), actually takes the user's popularity tendency into account when altering the recommendation list characteristics [Oh+11], similar to the PRA method. The approach by Oh et al. is built around the EMD, and as the name suggests, the primary goal of the algorithm is to match the popularity distribution of the recommendation list with the popularity distribution in the user's preferred set of items. However, the method can also be adapted to other quality factors. Due to its similarity with PRA, another set of empirical evaluations was conducted to compare PRA with PPTM in terms of their tendency matching abilities.

The results of these experiments are shown in Figure 5.5. In all cases displayed in the figure, the optimization goal was item popularity. On the left, the average EMD reduction rates for PRA are shown in groups of three, where each bar corresponds to the baseline strategy FunkSVD, BPR, or FM. For each group of three bars, a different exchange list size $|X_u|$ was tested. Similarly, for PPTM, the accuracy trade-off parameter c was different for each group.

As the figure shows, despite their different implementations, the algorithms performed quite similarly depending on how the trade-off parameters were chosen. However, for the BPR baseline, the PPTM re-ranking scheme performed much worse,

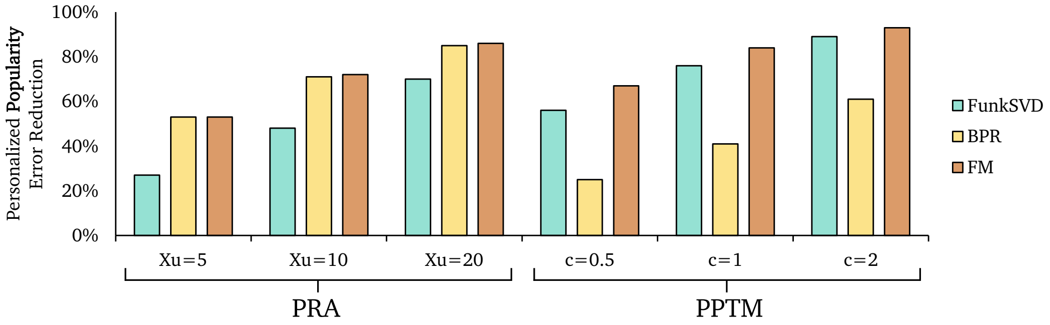


Figure 5.5: Comparison of PRA with the PPTM method by Oh et al. for the MovieLens data set based on all baseline strategies (FunkSVD, BPR, and FM) [Oh+11]. For both post-processing methods, a range of values for the respective trade-off parameters $|X_u|$ and c were tested (x-axis).

potentially suggesting that this approach is more sensitive to the specific baseline. In general, the results show that PRA can match or even outperform a comparable strategy from the literature. Additionally, PRA is more generic, since it is not reliant on the EMD measure and can optimize multiple quality factors at once.

5.3 Implications

The empirical evaluations presented in this chapter show that even a simple, yet generic post-processing strategy can improve an accuracy-optimized recommendation list by tailoring it more to the user's individual, multifaceted preferences. Designing RS algorithms only with accuracy in mind can lead to an uninspiring user experience, an outcome that can easily be avoided by taking additional quality factors into account. In fact, since PRA's inception, a number of further attempts have been made at tailoring recommendation lists to user tendencies (e.g., branded as *calibrated recommendations* [Ste18]) showing the research community's renewed interest in this important topic [NS17; Tin17; ZBP18].

Extracting user tendencies toward alternative quality factors from users' ratings or interaction histories is only one proposal to create more interesting recommendation lists that fit multifaceted user tastes better. As described in Section 2.3.1, self-reported user tendencies, as well as the user's mood, emotions, or even personality, could all be considered in future applications of the PRA scheme to generate recommendation lists that go beyond a one-size-fits-all approach.

6 Effects of Recommendations on Users' Decision-Making Behavior

As explained in the previous chapter, user preferences are not single-dimensional, and consequently, a range of quality factors need to be considered to design recommender systems that offer an appealing user experience. However, enhancing the user experience of a website, e.g., by providing engaging recommendations, is only one way how a recommender system can benefit users. Specifically, in e-commerce scenarios, in which users often take a long time to make purchase decisions, one main goal of a recommender system is to aid users in their decision-making processes [JA16; KPH15; Sch+06].

In such purchase situations, recommender systems can have a considerable impact on user decisions [JH09; KS94; Zan+06], which can be further amplified by using certain persuasive cues [GJG14; GL14; HKR00; TM07], such as persuasive explanations. However, a related topic that is still rather underexplored in the literature is how recommendations can *subconsciously* influence user decision making and whether individual users are susceptible to this manipulation to different degrees. This chapter presents two publications that investigate the effect of recommendations on user decision-making processes based on well-known concepts from the decision-making literature.

6.1 Anchoring in Recommendation Settings

Recommendations can influence user decision making in a variety of ways. On the one hand, this can happen transparently, for instance, when system providers display additional explanatory information about certain recommended items that highlights positive features or the items' specific match for the user [GL14; TM07]. On the other hand, marketers can use more subtle cues to influence user decisions via recommendations. To this end, they can exploit several phenomena known from the decision-making literature that apply to recommendation scenarios. For example, user decisions could be influenced by recommending less useful items (so-called decoy items) to make other recommended items seem more desirable in comparison [Fel+08; TFI11].

Similarly, the series of studies presented in this section [Köc+18] tries to ascertain whether the *anchoring effect*, a phenomenon known to occur in many decision scenarios, also applies to recommendation scenarios. Most famously, this effect was demonstrated in a series of studies by Tversky et al., one of which asked participants to estimate the percentage of African countries in the United Nations [TK74]. Before the participants made a decision, a number wheel—predetermined to land on 10 or 65—was spun in their presence. Even though the two events—the wheel's outcome and the estimate of countries—were unrelated, the subjects' estimates were biased toward the *anchor* value they saw on the wheel. Specifically, for subjects who received an initial stimulus of 10, the median estimated percentage of African countries was 25 %, but for participants with an anchor of 65, it was 45 %. Since these initial studies, the anchoring effect has been found to a apply to a range of scenarios, including decisions about how many items a consumer is willing to purchase and how much they are willing to spend [SD04; WKH98].

6.1.1 Research Question and Experimental Procedure

Based on the above-described insight from the decision-making and marketing literature, the aim of the series of studies described in this section [Köc+18] was to investigate whether the anchoring effect also applies to recommendations. More specifically, the assumption was that the *numerical features* of a recommended item (e.g., price) might influence consumer decisions toward an item with similar feature values, leading to an *attribute-level* anchoring effect. For example, if consumers were recommended a high-priced phone, they might be more likely to eventually purchase a similarly high-priced phone, because the price of the recommended phone subconsciously became an anchor in their decision-making process. The research question investigated was thus the following: *Can the numerical feature values of recommendations become anchors in users' subsequent decision-making processes and thus lead them to eventually choose items with similar features?*

To investigate this question, a set of experiments was conducted, both on historical click data and in a series of controlled user studies. First, the click log data of a large online fashion retailer that employs product recommendations was analyzed to identify whether the effect could be observed in a large-scale real-life setting. Initial evidence was found that the anchoring effect also occurs in recommendation scenarios. However, since the recommendations on this website were personalized, the relation between the eventually chosen items and the recommendations might have been based on the recommender system's ability to accurately predict the user's shopping intentions. Thus, a number of controlled follow-up studies were conducted on a fictitious online shopping site, specifically created for this purpose. To ensure that the observed effects did not originate from the fact that the recommended items matched participants' tastes, the recommendations were selected randomly in these experiments.

6.1.2 Offline Click Log Analysis

Goal and Setup. To investigate the above-mentioned research question (that is, whether recommendations can serve as anchors in purchasing scenarios), the browsing behavior of consumers on the website of a large European clothing retailer was analyzed. On this website, the detail page of each clothing item contains both descriptive information about the item itself and a set of recommendations. Based on the hypothesis that the displayed recommendations might influence the subsequent shopping behavior of the user, successive user clicks in the data were analyzed for any indication of anchoring.

In the anonymized data set, clicks were organized in browsing sessions, and for each click, basic information about the inspected item was given, such as the product ID, product category, and price category.[22] The analysis of the data was thus focused on three key attributes of each shopping session: the price level of the initially clicked item, the price level of the recommendations, and the price level of the subsequently clicked item. Based on these variables, three anchor values were possible. The recommendations could be more expensive, equally priced, or less expensive than the initially inspected item. Similarly, the user's reaction to this anchor could take three forms. That is, the user could inspect a second item that is more expensive, equally priced, or less expensive than the first item. In line with the anchoring hypothesis, the data was expected to show a relation between these two values. For example, when more expensive items were recommended compared to the initially viewed item, a disproportionate number of users was expected to subsequently visit more expensive items.

To control for environmental factors, the analysis was conducted on a single product category (T-shirts). Additionally, as mentioned, only the first two clicks of each session were analyzed, as both later clicks in each session and the final purchases might have been influenced by a variety of other effects not related to the research question.

Results. Figure 6.1 shows the results of the analysis. The bars in each group show how often an outcome was observed given a specific stimulus. For example, the bar on the very left shows that when customers were exposed to recommendations of a *lower* price category than the item they were initially looking at, they subse-

[22] Only the rough price category and not the exact sales price was included in the data set.

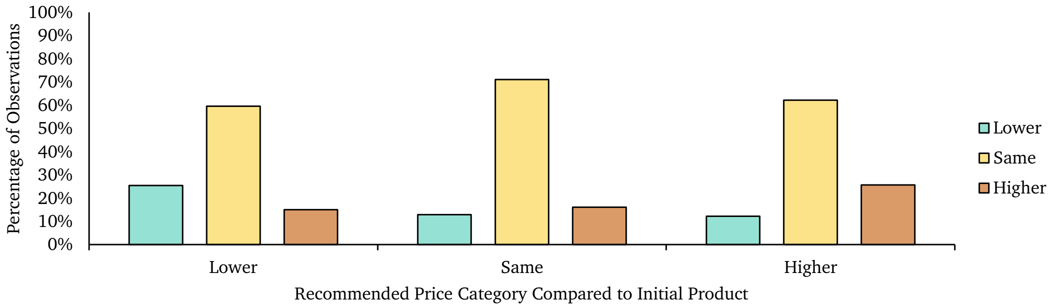


Figure 6.1: Results of the offline click log analysis.

quently also inspected a second item from a *lower* price category in 25 % of the cases. Note that, regardless of the given stimulus, most customers inspected a second item from the *same* price category as the first one, which can be attributed to normal browsing behavior. However, there is still a relation between the recommendations' attributes and the users' behavior. For example, when a lower-priced item was recommended, the percentage of customers that visited a lower-priced item afterward was 25 %, against only 13 % and 12 % when an equally or higher-priced item was recommended, respectively. A χ^2 test revealed that the price category of the recommendation and the price category of the second item were, indeed, not statistically independent.

One explanation might be that customers simply clicked on one of the recommended items. However, this was only the case in roughly 8 % of the analyzed sessions, and removing these cases from the analysis did not affect the significance of the observed anchoring effect. The analysis thus offers a first indication that the attributes—in this case, the price level—of recommendations might affect subsequent user decision making.

6.1.3 User Studies

While the results of the offline analysis suggest that recommendations might serve as anchors, it can still be argued that the recommender system on the website could have just been carefully tuned to anticipate the next click of the user, indirectly resulting in a measurable effect. Thus, more experiments were necessary to isolate the effect in a more controlled environment. To this end, three follow-up studies were conducted under controlled conditions, in which participants interacted with a fictitious online shopping site. The goal was to (i) decisively verify the existence of the anchoring effect of recommendations, (ii) identify how recommendations produce this effect, and (iii) rule out alternative explanations.

Isolation of the Effect

Goal and Setup. To isolate and quantify the effect observed in the previous analysis, a user study with 183 student participants was conducted in an online setting. In this study, participants were instructed to find a backpack for an upcoming hiking trip on a fictitious shopping website. The website designed for the experiment offered 18 choices with three distinguishing numerical features: weight, volume, and price. The options ranged from light, small, and inexpensive to heavy, large, and expensive. In addition, one of the available backpacks was *randomly chosen* by the system as a recommendation and placed at the top of the page with the label "you may like this backpack."

Results. A regression analysis of the collected data revealed that as suspected, the attributes of the recommended backpacks were related to the attributes of the backpacks participants chose, with regression coefficients of 0.157 for the price ($p < 0.01$), 0.185 for the volume ($p < 0.01$), and 0.093 for the weight ($p < 0.1$).[23] This confirms that even randomly generated recommendations can serve as anchors in decision-making processes. In addition, participants did not choose the randomly recommended item disproportionately often, which is unsurprising since it was not matched to their situation or taste. Additionally, excluding participants who chose the recommendation did not change the significance of the observed anchoring effect. Only for the weight attribute, the relation between recommendation and chosen item became insignificant ($p = 0.3$), indicating that anchoring does not affect every type of item feature equally strongly.

Search for a Possible Explanation

Goal and Setup. To search for an explanation as to *how* this attribute-level anchoring effect of recommendations influences consumers, a follow-up study was conducted with 72 test subjects. The study was devised to analyze how consumers distribute their *visual attention* among the alternatives. To this end, an eye-tracker was used to identify when participants looked at which items. The study followed the same procedure as the previous one, with participants choosing a backpack from a fictitious online shop. However, two changes were made to keep the experiment controlled. Subjects did not take part in the study from home but instead interacted with a defined hardware arrangement in a lab environment. The hardware con-

[23] The participants' domain expertise did not have a significant (direct or moderating) effect on the numerical feature values of the final choices.

sisted of a PC and an eye-tracking device mounted under the monitor. Additionally, due to the lower number of participants, only two of the available backpacks were randomly chosen as a recommendation: an expensive, large, heavy one and an inexpensive, small, light one—i.e., two options from opposite ends of the spectrum.

Results. In this study, the attribute-level anchoring effect was, again, observed. That is, subjects from each treatment group chose backpacks with significantly different features ($p < 0.05$). For example, participants who received a more expensive recommendation chose a backpack priced at €92.8, on average, while subjects who received the inexpensive recommendation, on average, selected a backpack with a price of €74.1. Additional insight into how this anchoring effect influenced participants was revealed by the collected eye-tracking data. The data indicated that participants' visual attention was directed toward products with features similar to those of the recommended product. That is, participants with a recommendation from the upper part of the spectrum visually inspected more expensive, heavier, and larger backpacks for a longer time.[24] One explanation for the anchoring effect induced by recommendations might thus be that consumers subconsciously focus more on items that are similar to the recommended alternative, which eventually influences their final choice. Interestingly, when asked whether they were aware of the effect, only half of the participants answered that they might have been affected by the recommendation. However, awareness of the bias did not influence how strongly the effect influenced the subjects.

Elimination of Alternative Explanations

Goal and Setup. For the first two user studies, in which recommendations were chosen at random, it would be unreasonable to assume that the personal fit of the recommended item had any kind of effect on the participants. Nonetheless, some participants might have assumed that the recommended option was in some way selected by the system provider for its superior features and that it would thus be wise to choose an option with similar features [CJ99]. To rule out this alternative explanation, a final set of user studies was conducted to determine whether users are consciously biased by recommendations because they consider them *informative*. For space reasons, only one of these studies is described here. In that study, 109 participants were again tasked with selecting a backpack from 18 alternatives. However, to test participants' reactions to a highly uninformative suggestion, the backpack recommendation at the top of the page was replaced with a rolling suitcase. The recommendation was randomly chosen from two rolling

[24] More detailed results of a regressions analysis are given the original paper [Köc+18].

suitcases that had identical features to the two backpack options from the previously described study in terms of weight, volume, and price.

Results. Even though the recommended product in this setup did not stem from the same product category as the items on offer, the results still showed that participants were biased toward the feature values of the recommended product. For example, participants who were recommended the more expensive rolling suitcase, chose, on average, a backpack priced at €95.4, while subjects who received the inexpensive rolling suitcase recommendation selected, on average, a backpack with a price of €83.1 ($p < 0.05$). The same significant effect was apparent in terms of the average chosen weight ($p < 0.1$) and volume ($p < 0.01$), confirming that the anchoring effect occurs *subconsciously* and not because users deliberately assume the recommendation's features to be in some way informative.

6.1.4 Implications

The results of the offline analysis and the user studies show that recommendations can become anchors that influence consumers' decision-making processes subconsciously. Even when recommendations are randomly chosen or stem from a different product category, the studies demonstrated that users are biased toward the feature values of the recommended products. These findings have implications not only for RS designers but also for online retailers and, possibly, consumer advocacy groups. On the one hand, the observed effects could become a valuable tool in the hands of retailers who want to (subconsciously) persuade consumers in a certain direction. Additionally, retailers might want to think about how they measure the success of their recommendations, as recommender systems might have positive sales effects even if consumers do not buy the recommended products. On the other hand, researchers should be aware of the anchoring effect when conducting user studies, since participants might be biased toward the feature values of recommendations. Depending on what is being investigated, study designers might not want participants to be influenced by the recommendation in such a way, and thus, steps need to be taken to prevent the effect's occurrence.

6.2 Maximizing and Satisficing in the Presence of Recommendations

Besides the anchoring effect, a range of other phenomena that can affect choice processes have been investigated in the decision-making literature. An often-studied phenomenon is the difference between so-called maximizers and satisficers. Originally introduced by Schwartz et al. [Sch+02], the general idea is that decision makers can be categorized based on how they generally approach choice situations [Dar+09; KFD12]. On the one hand, maximizers usually try to inspect as many of the available options as possible before they are ready to make a choice; yet, they are still rather unsatisfied with their chosen option. On the other hand, satisficers just scan the item space until they find an alternative that is "good enough." Thus, while maximizers try to find the best possible alternative, satisficers simply try to find an acceptable one.

In the context of recommender systems, only a few studies have investigated whether maximizers and satisficers react differently to recommendations. In the user study by Willemsen et al., which focused primarily on latent feature diversification, the participants' tendency to maximize was recorded among other variables [WGK16]. However, the participants' maximization tendency did not affect how they perceived the choice process, for example, in terms of choice difficulty and recommendation attractiveness. Similarly, Knijnenburg et al. investigated the effectiveness of an energy-saving recommender system and observed no differences between maximizers and satisficers in terms of, for example, their perceived level of control or RS effectiveness [KRW11]. However, maximizers were actually more satisfied with their choices, contradicting previous observations from the decision-making literature. The study presented in this section [JNJ18] continues this line of research by investigating the phenomenon in an alternative study setup that focuses specifically on the differences between maximizers and satisficers.

6.2.1 Research Questions and Experimental Procedure

Research Questions. One possible explanation for the above-mentioned results of previous studies that indicate no observable differences between maximizers and satisficers might be the presence of the recommender system itself. More specifically, the fact that the participants were aided by a recommender system might have somehow influenced their decision-making behavior so that the observable differences between maximizers and satisficers became unobservable. To further investigate this hypothesis, the study presented here asked participants to solve a decision-making task in the presence of a recommender system. However, in contrast to previous work, the study specifically focused on *objective* measurements related to maximizing and satisficing behavior, such as the length of time it took participants to make a decision or the number of items they inspected. Thus, the first research question investigated in the study was as follows: *Does the observable decision-making behavior of maximizers and satisficers differ in the presence of a recommender system?*

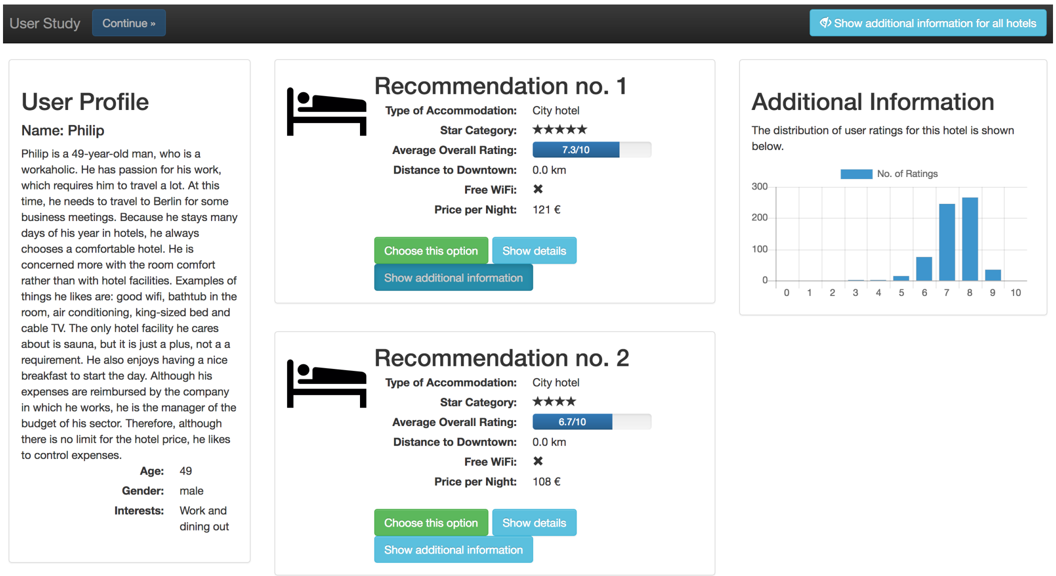
In addition, the decision-making literature suggests that maximizers and satisficers differ in their information needs when processing a choice situation. For example, maximizers tend to base their decisions more on relative comparisons than on absolute comparisons, and they refer more to external information sources, such as expert opinions or social comparisons [Wea+15; WGK16]. To analyze whether these types of individual needs also influence how users react to information supplied by recommender systems, the study also provided participants with different types of supplementary information that explained why certain items were recommended. The associated research questions were thus the following: *Do maximizers and satisficers react differently to explanatory information provided alongside recommendations? And, do they evaluate the usefulness of the different explanation styles differently?*

Experimental Setup. Participants were tasked with selecting a suitable hotel for a target person on a fictitious online hotel booking platform. The target person's preferences were described to the participants in text form (see Figure 6.2), and the preference profiles were created by the study designers based on real preference data from a historical hotel rating data set.[25] The list of recommended hotel options was then ordered via a `kNN` strategy based on the target user's preferences. For each hotel, basic features, such as the hotel's star rating, were listed, and by clicking a button next to the hotel, participants could open a pop-up with more detailed information about the hotel. Finally, next to each recommended hotel, an explanation was given as to why it was recommended.

Each participant had to make three decisions based on three target user profiles, and for each decision task, a different type of explanation was provided. The explanation styles were based on different knowledge sources (see Figure 6.3): the rating distribution of other hotel visitors (Style A) [Gre+10], a pros-and-cons comparison of the hotel's features (Style B) [Kni+12; Nun+14; RRS11], and an explanation of the recommendation mechanism itself in terms of selected neighbors' ratings (Style C) [BOH12; Che+13; Gre+10].

After finishing the three decision tasks, participants were asked to answer a final questionnaire, which included questions regarding the quality of the explanatory information and questions that captured the individual's maximization tendency based on the standard questionnaire items by Schwartz et al. [Sch+02]. Participants were recruited via e-mail invitations and social media. Overall, 243 people took part in the study, of which 109 completed the process. After the exclusion of 19 further individuals who completed the decision task in an unreasonably fast time, reliable results for 90 subjects were collected.

[25] Research suggests that maximizing and satisficing tendencies also transfer to situations in which a decision is made on behalf of someone else [CRM09].



User Profile

Name: Philip

Philip is a 49-year-old man, who is a workaholic. He has passion for his work, which requires him to travel a lot. At this time, he needs to travel to Berlin for some business meetings. Because he stays many days of his year in hotels, he always chooses a comfortable hotel. He is concerned more with the room comfort rather than with hotel facilities. Examples of things he likes are: good wifi, bathtub in the room, air conditioning, king-sized bed and cable TV. The only hotel facility he cares about is sauna, but it is just a plus, not a a requirement. He also enjoys having a nice breakfast to start the day. Although his expenses are reimbursed by the company in which he works, he is the manager of the budget of his sector. Therefore, although there is no limit for the hotel price, he likes to control expenses.

Age: 49
Gender: male
Interests: Work and dining out

Recommendation no. 1

Type of Accommodation: City hotel
Star Category: ★★★★★
Average Overall Rating: 7.3/10
Distance to Downtown: 0.0 km
Free WiFi: ✖
Price per Night: 121 €

Recommendation no. 2

Type of Accommodation: City hotel
Star Category: ★★★★
Average Overall Rating: 6.7/10
Distance to Downtown: 0.0 km
Free WiFi: ✖
Price per Night: 108 €

Additional Information

The distribution of user ratings for this hotel is shown below.

Figure 6.2: The user interface employed in the user study. On the left, a user profile description is given. In the middle, the recommendation list is shown. On the right, additional explanatory information is provided.

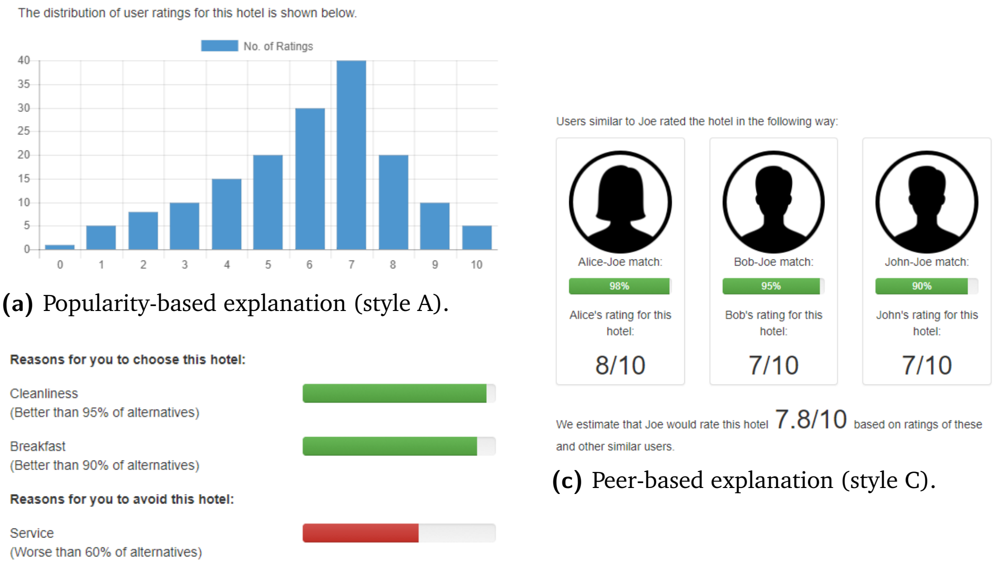
6.2.2 Results of the User Study

Decision-Making Behavior. In terms of the observed decision-making behavior, there were *no* significant[26] differences between maximizers and satisficers. That is, in contrast to what would be expected based on the decision-making literature, maximizers did not take longer to make a decision, did not open more item detail pages, and did not choose items from further down the list compared to satisficers.[27] In addition, the decision-making behavior of participants was not influenced by the explanation type provided to them. These results support the hypothesis that in the presence of a recommender system, the (normally expected) differences between maximizers and satisficers are not observable. One explanation could be that maximizers, who normally want to inspect most of the item space, experience a certain trust in the recommender system's reasoning, which leads them to accept one of the first few recommended items. In fact, the results show that about 25 % of the participants chose the first available option, corroborating this hypothesis.

Perceived Quality of Explanations. In the post-task questionnaire, the pros-and-cons explanation style (style B) received the best scores from both maximizers and satisficers in all quality dimensions expect trust. This outcome is not surprising, as feature-based explanations have already been shown to generally appeal to RS users [GJG14]. Looking at the preferences of maximizers and satisficers individually,

[26] Significance was determined by an analysis of variance (ANOVA) or, where the normality assumption did not hold, a Kruskal-Wallis test.

[27] Detailed results for all measurements can be found in the original paper [JNJ18].



The distribution of user ratings for this hotel is shown below.

(a) Popularity-based explanation (style A).

Reasons for you to choose this hotel:

Cleanliness
(Better than 95% of alternatives)

Breakfast
(Better than 90% of alternatives)

Reasons for you to avoid this hotel:

Service
(Worse than 60% of alternatives)

(b) Feature-based explanation (style B).

Users similar to Joe rated the hotel in the following way:

Alice-Joe match:	Bob-Joe match:	John-Joe match:
98%	95%	90%
Alice's rating for this hotel:	Bob's rating for this hotel:	John's rating for this hotel:
8/10	7/10	7/10

We estimate that Joe would rate this hotel 7.8/10 based on ratings of these and other similar users.

(c) Peer-based explanation (style C).

Figure 6.3: Explanation styles of the user study.

there were, mostly, no significant diversions in quality perceptions with respect to explanation styles for the two groups. Only in terms of transparency, maximizers significantly preferred style B over style C. An explanation could be that maximizers, as already mentioned, prefer relative information over absolute information, which corresponds to the pros-and-cons style. However, overall, the preferences toward different supplementary explanatory information mostly aligned for maximizers and satisficers. Consequently, at least regarding the tested explanation styles, no particular approach appears to be better suited for either group.

6.2.3 Implications

The results of the study show that not all phenomena from the decision-making literature also directly apply to scenarios in which recommender systems are present. In fact, the difference in decision-making behavior between maximizers and satisficers is a well-researched topic, with some studies that have included over 1,000 participants [Sch+02]. However, in line with the results from previous RS studies, the differences between maximizers and satisficers were not observable in the presented study, which could be due to the presence of a recommender system. To understand exactly what role recommender systems play in this context, more research is necessary. Continued research efforts could also help to clarify the effect of explanatory information, as investigated in the more recent study by [Cob+19]. Here, maximizers reacted differently to rating-style explanations, which also affected their final choices. Future studies should thus aim to uncover how recommendations and explanations can be used to specifically aid users with certain decision-making styles, for example, by reducing their decision times or increasing their choice satisfaction.

Conclusion 7

Recommender systems are an integral part of modern websites and applications. They are, for example, used on shopping websites to provide purchase suggestions or on music streaming services to help users explore music they might like. One key factor for the success of recommender systems is the research community's continued effort to find more accurate recommendation strategies, for example, based on state-of-the-art machine learning techniques. However, the degree to which recommendations match a user's preferences represents only one of the aspects that determines the practical success of a recommender system. Specifically, from a user perspective, a range of factors (e.g., related to the visual design of the presented recommendations), the diversity of the recommended items, or the user's current context, can affect how users perceive recommendations. This thesis has presented several publications that demonstrate how recommender systems can be designed and evaluated from a more user-focused perspective that takes such factors into account.

Chapter 2 served as an overview of the RS research landscape. It introduced major use cases of recommendations; commonly used algorithmic strategies; and the fundamentals of interactions between users and recommender systems, such as feedback and control mechanisms. Furthermore, the chapter highlighted the difference between industrial and academic evaluation practices, showcasing that specifically in academia, the success of recommender systems is often assessed in a manner that does not take the user fully into account.

As an example of a more holistic evaluation approach, the merits of several recommendation strategies in the novel application field of machine learning workflows were investigated in Chapter 3. The results highlighted the importance of assessing recommendation performance from different perspectives: in offline settings to obtain an initial accuracy estimate, in controlled user studies to gather qualitative insights, and in online tests to judge performance under real-life conditions.

However, when conducting evaluations, and more specifically user studies, it is also important to be aware of potential biases that can influence the observed results. Based on the example of a user study involving movie recommendations, Chapter 4 showed that participants react differently to recommendations depending on how familiar they are with them. Specifically, participants in this study assigned higher ratings to recommended movies when they had already watched them. Given that some algorithms tend to recommend more popular movies, this familiarity bias can, in turn, lead to incorrect assumptions about the algorithms' comparative performance. Experiments involving test subjects should thus keep the existence of such biases in mind and either address them in their analyses or avoid them altogether with the help of more sophisticated study designs.

Applying a user-focused perspective is important not only when *evaluating* recommender system but also when *designing* recommendation approaches. However, many of today's approaches simply try to rank items based on how well they fit the user's preferences, without considering alternative quality characteristics of the resulting recommendation list, such as its diversity. With this in mind, Chapter 5 presented a post-processing approach that aims to re-rank recommendation lists to reflect user tendencies with respect to alternative quality aspects, such as diversity or popularity. The results of a series of empirical evaluations of the proposed approach showed that it can efficiently balance multiple user-focused optimization goals while retaining overall accuracy.

Furthermore, while many academic proposals have focused on creating recommendation approaches that provide accurate suggestions or an engaging user experience, research has often overlooked the effect of recommendations on users' decision-making processes. Chapter 6 presented two publications that investigated how recommendations can subconsciously influence users' decisions. On the one hand, in Section 6.1, the results of a series of experiments demonstrated that recommendations can evoke an attribute-level anchoring effect. That is, users can become anchored to the numerical attributes of a recommended product, such as its price, and subsequently choose an item with similar characteristics.

On the other hand, in Section 6.2, a study on the maximizer-satisficer phenomenon showed no differences between maximizers and satisficers, an outcome that differs from previous insights documented in the decision-making literature. One reason for this unexpected user behavior might be connected to the presence of the recommender system, indicating another form of subconscious user manipulation. While these results provide a first account of how recommender systems can influence users' decision-making processes, more experiments are necessary to develop an exhaustive understanding of such phenomena. Further user studies in other domains or online settings could investigate, for example, how anchoring effects of recommendations could be exploited in practice and which other cognitive effects can be evoked by recommendations.

Bibliography

[Ado+12] Gediminas Adomavicius, Jesse C. Bockstedt, Shawn Curley, and Jingjing Zhang. “Effects of Online Recommendations on Consumers’ Willingness to Pay.” In: *Proceedings of the 2nd Workshop on Human Decision Making in Recommender Systems at RecSys ’12*. 2012, pp. 40–45 (cit. on pp. 3, 25).

[Ado+13] Gediminas Adomavicius, Jesse C. Bockstedt, Shawn Curley, and Jingjing Zhang. “Do Recommender Systems Manipulate Consumer Preferences? A Study of Anchoring Effects.” In: *Information Systems Research* 24.4 (2013), pp. 956–975 (cit. on pp. 3, 25).

[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 (cit. on p. 15).

[Aiz03] Akiko Aizawa. “An Information-Theoretic Perspective of TF–IDF Measures.” In: *Information Processing & Management* 39.1 (2003), pp. 45–65 (cit. on p. 50).

[AK07] Gediminas Adomavicius and YoungOk Kwon. “New Recommendation Techniques for Multicriteria Rating Systems.” In: *Intelligent Systems* 22.3 (2007), pp. 48–55 (cit. on p. 16).

[AK12] Gediminas Adomavicius and YoungOk Kwon. “Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques.” In: *IEEE Transactions on Knowledge and Data Engineering* 24.5 (2012), pp. 896–911 (cit. on pp. 45, 53).

[Ama+11] Xavier Amatriain, Alejandro Jaimes, Nuria Oliver, and Josep M. Pujol. “Data Mining Methods for Recommender Systems.” In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Springer, 2011, pp. 39–71 (cit. on p. 9).

[APO19] Ivana Andjelkovic, Denis Parra, and John O’Donovan. “Moodplay: Interactive Music Recommendation Based on Artists’ Mood Similarity.” In: *International Journal of Human-Computer Studies* 121 (2019), pp. 142–159 (cit. on p. 22).

[Arg+18] A. Argal, S. Gupta, A. Modi, P. Pandey, S. Shim, and C. Choo. “Intelligent Travel Chatbot for Predictive Recommendation in Echo Platform.” In: *Proceedings of the 8th Annual Computing and Communication Workshop and Conference (CCWC ’18)*. 2018, pp. 176–183 (cit. on p. 18).

[BGG11] Claudia Becerra, Fabio Gonzalez, and Alexander Gelbukh. “Visualizable and Explicable Recommendations Obtained from Price Estimation Functions.” In: *Joint Proceedings of the RecSys 2011 Workshop on Human Decision Making in Recommender Systems (Decisions@RecSys ’11) and User-Centric Evaluation of Recommender Systems and Their Interfaces-2 (UCERSTI ’11) at RecSys ’11)*. 2011, pp. 27–34 (cit. on p. 22).

[BGW12] Dirk G. F. M. Bollen, Mark P. Graus, and Martijn C. Willemsen. “Remembering the Stars?: Effect of Time on Preference Retrieval from Memory.” In: *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys ’12)*. 2012, pp. 217–220 (cit. on p. 20).

[BJ14] Geoffray Bonnin and Dietmar Jannach. “Automated Generation of Music Playlists: Survey and Experiments.” In: *ACM Computer Surveys* 47.2 (2014), pp. 1–35 (cit. on p. 50).

[BK14] Shay Ben-Elazar and Noam Koenigstein. “A Hybrid Explanations Framework for Collaborative Filtering Recommender Systems.” In: *Poster Proceedings of the 8th ACM Conference on Recommender Systems (RecSys ’14)*. 2014 (cit. on p. 23).

[BL15] Jöran Beel and Stefan Langer. “A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems.” In: *Proceedings of the 19th International Conference on Theory and Practice of Digital Libraries Research and Advanced Technology for Digital Libraries (TPDL ’15)*. 2015, pp. 153–168 (cit. on pp. 3, 12).

[Bob+13] Szymon Bobek, Mateusz Baran, Krzysztof Kluza, and Grzegorz J. Nalepa. “Application of Bayesian Networks to Recommendations in Business Process Modeling.” In: *Proceedings of the 2013 Workshop AI Meets Business Processes (AIBP ’13) at AI*IA ’13*. 2013, pp. 41–50 (cit. on p. 29).

[BOH12] Svetlin Bostandjiev, John O’Donovan, and Tobias Höllerer. “TasteWeights: A Visual Interactive Hybrid Recommender System.” In: *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys ’12)*. 2012, pp. 35–42 (cit. on p. 63).

[Bro+16] Yuri M. Brovman, Marie Jacob, Natraj Srinivasan, Stephen Neola, Daniel Galron, Ryan Snyder, and Paul Wang. “Optimizing Similar Item Recommendations in a Semi-structured Marketplace to Maximize Conversion.” In: *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys ’16)*. 2016, pp. 199–202 (cit. on p. 2).

[BS01] K. Bradley and B. Smyth. “Improving Recommendation Diversity.” In: *Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science (AICS ’01)*. 2001, pp. 75–84 (cit. on p. 45).

[BTG18] Michael D. Buhrmester, Sanaz Talaifar, and Samuel D. Gosling. “An Evaluation of Amazon’s Mechanical Turk, Its Rapid Rise, and Its Effective Use.” In: *Perspectives on Psychological Science* 13.2 (2018), pp. 149–154 (cit. on p. 41).

[Bur00] Robin Burke. “Knowledge-based Recommender Systems.” In: *Encyclopedia of Library and Information Science* 69.32 (2000), pp. 180–200 (cit. on pp. 2, 8, 10, 18).

[Bur02] Robin Burke. “Hybrid Recommender Systems: Survey and Experiments.” In: *User Modeling and User-Adapted Interaction* 12.4 (2002), pp. 331–370 (cit. on p. 10).

[CAS16] Paul Covington, Jay Adams, and Emre Sargin. “Deep Neural Networks for YouTube Recommendations.” In: *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys ’16)*. 2016, pp. 191–198 (cit. on pp. 2, 10).

[CGT13] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. “User-Centric vs. System-Centric Evaluation of Recommender Systems.” In: *Proceedings of the 2013 International Conference on Human-Computer Interaction (INTERACT ’13)*. 2013, pp. 334–351 (cit. on pp. 2, 40).

[Cha+11] Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. “Apolo: Interactive Large Graph Sensemaking by Combining Machine Learning and Visualization.” In: *Proceedings of the 17th SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’11)*. 2011, pp. 739–742 (cit. on p. 24).

[Che+13] Yu-Chih Chen, Yu-Shi Lin, Yu-Chun Shen, and Shou-De Lin. “A Modified Random Walk Framework for Handling Negative Ratings and Generating Explanations.” In: *Transactions on Intelligent Systems and Technology* 4.1 (2013), pp. 12:1–12:21 (cit. on p. 63).

[CJ99] Gretchen B. Chapman and Eric J. Johnson. “Anchoring, Activation, and the Construction of Values.” In: *Organizational Behavior and Human Decision Processes* 79.2 (1999), pp. 115–153 (cit. on p. 60).

[Cob+19] Ludovik Coba, Laurens Rook, Markus Zanker, and Panagiotis Symeonidis. “Decision Making Strategies Differ in the Presence of Collaborative Explanations: Two Conjoint Studies.” In: *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI ’19)*. 2019, pp. 291–302 (cit. on p. 65).

[Cos+03] Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. “Is Seeing Believing?: How Recommender System Interfaces Affect Users’ Opinions.” In: *Proceedings of the 2003 SIGCHI Conference on Human Factors in Computing Systems (CHI ’03)*. 2003, pp. 585–592 (cit. on pp. 15, 25).

[CP06] Li Chen and Pearl Pu. “Evaluating Critiquing-based Recommender Agents.” In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI ’06)*. 2006, pp. 157–162 (cit. on p. 18).

[CP08] Li Chen and Pearl Pu. “A Cross-Cultural User Evaluation of Product Recommender Interfaces.” In: *Proceedings of the 2nd ACM Conference on Recommender Systems (RecSys ’08)*. 2008, pp. 75–82 (cit. on p. 21).

[CRM09] Tilottama G. Chowdhury, S. Ratneshwar, and Praggyan Mohanty. “The Time-Harried Shopper: Exploring the Differences Between Maximizers and Satisficers.” In: *Marketing Letters* 20.2 (2009), pp. 155–167 (cit. on p. 63).

[Dal+14] Elizabeth M. Daly, Adi Botea, Akihiro Kishimoto, and Radu Marinescu. “Multi-Criteria Journey Aware Housing Recommender System.” In: *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys ’14)*. 2014, pp. 325–328 (cit. on p. 22).

[Dar+09] Ilan Dar-Nimrod, Catherine D. Rawn, Darrin R. Lehman, and Barry Schwartz. “The Maximization Paradox: The Costs of Seeking Alternatives.” In: *Personality and Individual Differences* 46.5 (2009), pp. 631–635 (cit. on p. 62).

[Dav+10] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. “The YouTube Video Recommendation System.” In: *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys ’10)*. 2010, pp. 293–296 (cit. on p. 11).

[DPM14] Simon Dooms, Toon De Pessemier, and Luc Martens. “Improving IMDb Movie Recommendations With Interactive Settings and Filters.” In: *Poster Proceedings of the 8th ACM Conference on Recommender Systems (RecSys ’14)*. 2014 (cit. on p. 24).

[Eks+14] Michael D. Ekstrand, F. Maxwell Harper, Martijn C. Willemsen, and Joseph A. Konstan. “User Perception of Differences in Recommender Algorithms.” In: *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys ’14)*. 2014, pp. 161–168 (cit. on pp. 3, 12, 40).

[Eks+15] Michael D. Ekstrand, Daniel Kluver, F. Maxwell Harper, and Joseph A. Konstan. “Letting Users Choose Recommender Algorithms: An Experimental Study.” In: *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15)*. 2015, pp. 11–18 (cit. on p. 24).

[Ela+13] Mehdi Elahi, Matthias Braunhofer, Francesco Ricci, and Marko Tkalcic. “Personality-Based Active Learning for Collaborative Filtering Recommender Systems.” In: *Proceedings of the 25th International Conference of the Italian Association for Artificial Intelligence (AI*IA ’13)*. 2013, pp. 360–371 (cit. on p. 19).

[Ela+15] Mehdi Elahi, Mouzhi Ge, Francesco Ricci, Ignacio Fernández-Tobías, Shlomo Berkovski, and David Massimo. “Interaction Design in a Mobile Food Recommender System.” In: *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS ’15) at RecSys ’15*. 2015, pp. 49–52 (cit. on p. 19).

[Fel+07] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. “An Integrated Environment for the Development of Knowledge-Based Recommender Applications.” In: *International Journal of Electronic Commerce* 11.2 (2007), pp. 11–34 (cit. on pp. 18, 23).

[Fel+08] Alexander Felfernig, Bartosz Gula, Gerhard Leitner, Marco Maier, Rudolf Melcher, and Erich Teppan. “Persuasion in Knowledge-Based Recommendation.” In: *Proceedings of the Third International Conference on Persuasive Technology (PERSUASIVE ’08)*. 2008 (cit. on pp. 25, 55).

[FZ12] Matthias Fuchs and Markus Zanker. “Multi-criteria Ratings for Recommender Systems: An Empirical Analysis in the Tourism Domain.” In: *Proceedings of the 13th International Conference on E-Commerce and Web Technologies (EC-Web ’12)*. 2012, pp. 100–111 (cit. on p. 16).

[Gar+14] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. “Offline and Online Evaluation of News Recommender Systems at swissinfo.ch.” In: *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys ’14)*. 2014, pp. 169–176 (cit. on pp. 2, 3, 11).

[GH15] Carlos A. Gomez-Uribe and Neil Hunt. “The Netflix Recommender System: Algorithms, Business Value, and Innovation.” In: *Transactions on Management Information Systems* 6.4 (2015), pp. 13:1–13:19 (cit. on pp. 2, 3, 7, 11, 12, 20).

[GJ13] Fatih Gedikli and Dietmar Jannach. “Improving Recommendation Accuracy Based on Item-specific Tag Preferences.” In: *Transactions on Intelligent Systems and Technology* 4.1 (2013), pp. 11:1–11:19 (cit. on p. 19).

[GJG14] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. “How Should I Explain? A Comparison of Different Explanation Types for Recommender Systems.” In: *International Journal of Human-Computer Studies* 72.4 (2014), pp. 367–382 (cit. on pp. 23, 25, 55, 64).

[GL14] Sofia Gkika and George Lekakos. “The Persuasive Role of Explanations in Recommender Systems.” In: *Proceedings of the 2nd International Workshop on Behavior Change Support Systems (BCSS ’14)*. 2014, pp. 59–68 (cit. on pp. 25, 55).

[Gol93] Lewis R. Goldberg. “The Structure of Phenotypic Personality Traits.” In: *American psychologist* 48.1 (1993), pp. 26 (cit. on p. 19).

[Gre+10] Brynjar Gretarsson, John O’Donovan, Svetlin Bostandjiev, Christopher Hall, and Tobias Hollerer. “SmallWorlds: Visualizing Social Recommendations.” In: *Computer Graphics Forum* 29.3 (2010), pp. 833–842 (cit. on p. 63).

[GT00] Mehmet Göker and Cynthia Thompson. “The Adaptive Place Advisor: A Conversational Recommendation System.” In: *Proceedings of the 8th German Workshop on Case Based Reasoning*. 2000, pp. 187–198 (cit. on p. 18).

[Her+04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. “Evaluating Collaborative Filtering Recommender Systems.” In: *Transactions on Information Systems* 22.1 (2004), pp. 5–53 (cit. on pp. 2, 20).

[HKN12] Yoshinori Hijikata, Yuki Kai, and Shogo Nishida. “The Relation Between User Intervention and User Satisfaction for Information Recommendation.” In: *Proceedings of the 27th Annual Symposium on Applied Computing (SAC ’12)*. 2012, pp. 2002–2007 (cit. on p. 24).

[HKR00] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. “Explaining Collaborative Filtering Recommendations.” In: *Proceedings of the 2000 Conference on Computer Supported Cooperative Work (CSCW ’00)*. 2000, pp. 241–250 (cit. on pp. 25, 55).

[Hol16] Carmen Holotescu. “MOOCBuddy: A Chatbot for Personalized Learning With MOOCs.” In: *Proceedings of the 13th International Conference on Human Computer Interaction (RoCHI ’16)*. 2016, pp. 91–94 (cit. on p. 18).

[Höp+10] Wolfram Höpken, Matthias Fuchs, Markus Zanker, and Thomas Beer. “Context-Based Adaptation of Mobile Applications in Tourism.” In: *Information Technology & Tourism* 12.2 (2010), pp. 175–195 (cit. on p. 25).

[IGÖ15] Jon Espen Ingvaldsen, Jon Atle Gulla, and Özlem Özgöbek. “User Controlled News Recommendations.” In: *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS ’15) at RecSys ’15*. 2015, pp. 45–48 (cit. on p. 19).

[IL00] Sheena S. Iyengar and Mark R. Lepper. "When Choice is Demotivating: Can One Desire Too Much of a Good Thing?" In: *Journal of Personality and Social Psychology* 79.6 (2000), pp. 995–1006 (cit. on p. 20).

[JA16] Dietmar Jannach and Gediminas Adomavicius. "Recommendations With a Purpose." In: *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. 2016, pp. 7–10 (cit. on pp. 8, 13, 55).

[Jan+] Dietmar Jannach, Paul Resnick, Alexander Tuzhilin, and Markus Zanker. "Recommender Systems—Beyond Matrix Completion." In: *Communications of the ACM* 59.11 (), pp. 94–102 (cit. on p. 13).

[Jan+15] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. "What Recommenders Recommend: An Analysis of Recommendation Biases and Possible Countermeasures." In: *User Modeling and User-Adapted Interaction* 25.5 (2015), pp. 427–491 (cit. on pp. 51, 87).

[JF14] Dietmar Jannach and Simon Fischer. "Recommendation-Based Modeling Support for Data Mining Processes." In: *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. 2014, pp. 337–340 (cit. on p. 5).

[JH09] Dietmar Jannach and Kolja Hegelich. "A Case Study on the Effectiveness of Recommendations in the Mobile Internet." In: *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. 2009, pp. 205–208 (cit. on pp. 3, 11, 12, 55).

[JJ17] Michael Jugovac and Dietmar Jannach. "Interacting With Recommenders—Overview and Research Directions." In: *Transactions on Interactive Intelligent Systems* 7.3 (2017), pp. 10:1–10:46 (cit. on pp. 4, 13, 87).

[JJK18] Michael Jugovac, Dietmar Jannach, and Mozhgan Karimi. "StreamingRec: A Framework for Benchmarking Stream-Based News Recommenders." In: *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 2018, pp. 269–273 (cit. on p. 88).

[JJL15] Dietmar Jannach, Michael Jugovac, and Lukas Lerche. "Adaptive Recommendation-Based Modeling Support for Data Analysis Workflows." In: *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15)*. 2015, pp. 252–262 (cit. on p. 88).

[JJL16] Dietmar Jannach, Michael Jugovac, and Lukas Lerche. "Supporting the Design of Machine Learning Workflows With a Recommendation System." In: *Transactions on Interactive Intelligent Systems* 6.1 (2016), pp. 8:1–8:35 (cit. on pp. 5, 27, 28, 33, 87).

[JJL17] Michael Jugovac, Dietmar Jannach, and Lukas Lerche. "Efficient Optimization of Multiple Recommendation Quality Factors According to Individual User Tendencies." In: *Expert Systems With Applications* 81 (2017), pp. 321–331 (cit. on pp. 5, 45, 48, 49, 51, 87).

[JK05] Dietmar Jannach and Gerold Kreutler. "Personalized User Preference Elicitation for e-Services." In: *Proceedings of the 2005 International Conference on e-Technology, e-Commerce, and e-Service (EEE '05)*. 2005, pp. 604–611 (cit. on p. 18).

[JK07] Dietmar Jannach and Gerold Kreutler. “Rapid Development of Knowledge-based Conversational Recommender Applications With Advisor Suite.” In: *Journal of Web Engineering* 6.2 (2007), pp. 165–192 (cit. on p. 18).

[JKL17] Dietmar Jannach, Iman Kamehkhosh, and Lukas Lerche. “Leveraging Multi-Dimensional User Models for Personalized Next-Track Music Recommendation.” In: *Proceedings of the Symposium on Applied Computing (SAC ’17)*. 2017, pp. 1635–1642 (cit. on p. 8).

[JL17] Dietmar Jannach and Malte Ludewig. “When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation.” In: *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys ’17)*. 2017, pp. 306–310 (cit. on p. 8).

[JLJ15a] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Adaptation and Evaluation of Recommendations for Short-Term Shopping Goals.” In: *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15)*. 2015, pp. 211–218 (cit. on p. 88).

[JLJ15b] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation.” In: *i-com* 14.1 (2015), pp. 29–39 (cit. on pp. 5, 40, 43, 87).

[JLJ15c] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Item Familiarity Effects in User-Centric Evaluations of Recommender Systems.” In: *Poster Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15)*. 2015 (cit. on p. 88).

[JLK15] Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. “Beyond ‘Hitting the Hits’: Generating Coherent Music Playlist Continuations With the Right Tracks.” In: *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15)*. 2015, pp. 187–194 (cit. on p. 8).

[JNJ16] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. “User Control in Recommender Systems: Overview and Interaction Challenges.” In: *Proceedings of the 17th International Conference on Electronic Commerce and Web Technologies (EC-Web ’16)*. 2016, pp. 21–33 (cit. on p. 88).

[JNJ17] Dietmar Jannach, Ingrid Nunes, and Michael Jugovac. “Interacting With Recommender Systems.” In: *Companion Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI ’17)*. 2017, pp. 25–27 (cit. on p. 88).

[JNJ18] Michael Jugovac, Ingrid Nunes, and Dietmar Jannach. “Investigating the Decision-Making Behavior of Maximizers and Satisficers in the Presence of Recommendations.” In: *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP ’18)*. 2018, pp. 279–283 (cit. on pp. 6, 62, 64, 87).

[JW10] Tamas Jambor and Jun Wang. “Optimizing Multiple Objectives in Collaborative Filtering.” In: *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys ’10)*. 2010, pp. 55–62 (cit. on p. 45).

[JWB05] Zhenhui Jiang, Weiquan Wang, and Izak Benbasat. “Multimedia-Based Interactive Advising Technology for Online Consumer Decision Support.” In: *Communications of the ACM* 48.9 (2005), pp. 92–98 (cit. on p. 18).

[Kap+15] Komal Kapoor, Vikas Kumar, Loren G. Terveen, Joseph A. Konstan, and Paul R. Schrater. "'I Like to Explore Sometimes': Adapting to Dynamic User Novelty Preferences." In: *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. 2015, pp. 19–26 (cit. on p. 45).

[Kar+10] Kristiina Karvonen, Sanna Shibasaki, Sofia Nunes, Puneet Kaur, and Olli Immonen. "Visual Nudges for Enhancing the Use and Produce of Reputation Information." In: *Proceedings of the Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI '10) at RecSys '10*. 2010, pp. 1–8 (cit. on p. 21).

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. "Matrix Factorization Techniques for Recommender Systems." In: *Computer* 42.8 (2009), pp. 30–37 (cit. on pp. 2, 10).

[KFD12] Evan Kirshenbaum, George Forman, and Michael Dugan. "A Live Comparison of Methods for Personalized Article Recommendation at Forbes.com." In: *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD '12)*. 2012, pp. 51–66 (cit. on p. 62).

[KHO11] Agnes Koschmider, Thomas Hornung, and Andreas Oberweis. "Recommendation-Based Editor for Business Process Modeling." In: *Data and Knowledge Engineering* 70.6 (2011), pp. 483–503 (cit. on p. 29).

[KJ17] Iman Kamehkhosh and Dietmar Jannach. "User Perception of Next-Track Music Recommendations." In: *Proceedings of the 25th Conference on User Modeling, Adaptation, and Personalization (UMAP '17)*. 2017, pp. 113–121 (cit. on p. 8).

[KJJ18] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. "News Recommender Systems—Survey and Roads Ahead." In: *Information Processing and Management* 54.6 (2018), pp. 1203–1227 (cit. on p. 88).

[KKP01] Alfred Kobsa, Jürgen Koenemann, and Wolfgang Pohl. "Personalised Hypermedia Presentation Techniques for Improving Online Customer Relationships." In: *The Knowledge Engineering Review* 16.2 (2001), pp. 111–155 (cit. on p. 18).

[Klu+12] Daniel Kluver, Tien T. Nguyen, Michael D. Ekstrand, Shilad Sen, and John Riedl. "How Many Bits per Rating?" In: *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys '12)*. 2012, pp. 99–106 (cit. on p. 16).

[KLZ15] Johannes Kunkel, Benedikt Loepp, and Jürgen Ziegler. "3D-Visualisierung zur Eingabe von Präferenzen in Empfehlungssystemen." In: *Proceedings of the 2015 Conference on Mensch und Computer 2015*. 2015, pp. 123–132 (cit. on pp. 22, 24).

[Kni+12] Bart P. Knijnenburg, Svetlin Bostandjiev, John O'Donovan, and Alfred Kobsa. "Inspectability and Control in Social Recommenders." In: *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys '12)*. 2012, pp. 43–50 (cit. on p. 63).

[Köc+16] Sören Köcher, Dietmar Jannach, Michael Jugovac, and Hartmut H. Holzmüller. "Investigating Mere-Presence Effects of Recommendations on the Consumer Choice Process." In: *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS' 16) at RecSys '16*. 2016, pp. 2–5 (cit. on pp. 3, 88).

[Köc+18] Sören Köcher, Michael Jugovac, Dietmar Jannach, and Hartmut H. Holzmüller. “New Hidden Persuaders: An Investigation of Attribute-Level Anchoring Effects of Product Recommendations.” In: *Journal of Retailing* 95.1 (2018), pp. 24–41 (cit. on pp. 3, 6, 56, 60, 87).

[KPH15] Sahar Karimi, K. Nadia Papamichail, and Christopher P. Holland. “The Effect of Prior Knowledge and Decision-Making Style on the Online Purchase Decision-Making Process: A Typology of Consumer Shopping Behaviour.” In: *Decision Support Systems* 77 (2015), pp. 137–147 (cit. on p. 55).

[KRW11] Bart P. Knijnenburg, Niels J. M. Reijmer, and Martijn C. Willemsen. “Each to His Own: How Different Users Call for Different Interaction Mthods in Recommender Systems.” In: *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys ’11)*. 2011, pp. 141–148 (cit. on pp. 17, 62).

[KS94] David A. Klein and Edward H. Shortliffe. “A Framework for Explaining Decision-theoretic Advice.” In: *Artificial Intelligence* 67.2 (1994), pp. 201–243 (cit. on p. 55).

[LAW14] Béatrice Lamche, Ugur Adigüzel, and Wolfgang Wörndl. “Interactive Explanations in Mobile Shopping Recommender Systems.” In: *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS ’14) at RecSys ’14*. 2014, pp. 14–21 (cit. on p. 23).

[Lee04] Wei-Po Lee. “Towards Agent-Based Decision Making in the Electronic Marketplace: Interactive Recommendation and Automated Negotiation.” In: *Expert Systems With Applications* 27.4 (2004), pp. 665–679 (cit. on p. 18).

[LHL97] Greg Linden, Steve Hanks, and Neal Lesh. “Interactive Assessment of User Preference Models: The Automated Travel Assistant.” In: *Proceedings of the 6th International Conference on User Modeling (UM ’97)*. Vol. 383. 1997, pp. 67–78 (cit. on p. 18).

[LHZ14] Benedikt Loepp, Tim Hussein, and Jürgen Ziegler. “Choice-Based Preference Elicitation for Collaborative Filtering Recommender Systems.” In: *Proceedings of the 2014 SIGCHI Conference on Human Factors in Computing Systems (CHI ’14)*. 2014, pp. 3085–3094 (cit. on p. 19).

[Li+14] Ying Li, Bin Cao, Lida Xu, Jianwei Yin, Shuiguang Deng, Yuyu Yin, and Zhaohui Wu. “An Efficient Recommendation Method for Improving Business Process Modeling.” In: *IEEE Transactions on Industrial Informatics* 10.1 (2014), pp. 502–513 (cit. on p. 29).

[LJJ17] Malte Ludewig, Michael Jugovac, and Dietmar Jannach. “A Light-Weight Approach to Recipient Determination When Recommending New Items.” In: *Proceedings of the RecSys Challenge Workshop at RecSys ’17*. 2017 (cit. on p. 88).

[LM05] Daniel Lemire and Anna Maclachlan. “Slope One Predictors for Online Rating-Based Collaborative Filtering.” In: *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM ’05)*. 2005, pp. 471–475 (cit. on p. 40).

[LMR11] Henrik Leopold, Jan Mendling, and Hajo A. Reijers. “On the Automatic Labeling of Process Models.” In: *Proceedings of the 23rd International Conference on Advanced Information Systems Engineering (CAiSE ’11)*. 2011, pp. 512–520 (cit. on p. 29).

[LO07] Haibin Ling and Kazunori Okada. “An Efficient Earth Mover’s Distance Algorithm for Robust Histogram Comparison.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.5 (2007), pp. 840–853 (cit. on p. 47).

[Loe+18] Benedikt Loepp, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. “Impact of Item Consumption on Assessment of Recommendations in User Studies.” In: *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18)*. 2018, pp. 49–53 (cit. on pp. 16, 43).

[LVZ13] Anísio Lacerda, Adriano Veloso, and Nivio Ziviani. “Exploratory and interactive daily deals recommendation.” In: *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys ’13)*. 2013, pp. 439–442 (cit. on p. 25).

[McC+04] Kevin McCarthy, James Reilly, Lorraine McGinty, and Barry Smyth. “On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems.” In: *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH ’04)*. 2004, pp. 176–184 (cit. on p. 18).

[MM09] Steffen Mazanek and Mark Minas. “Business Process Models as a Showcase for Syntax-Based Assistance in Diagram Editors.” In: *Proceedings of the 12th International on Model Driven Engineering Languages and Systems (MoDELS ’09)*. 2009, pp. 322–336 (cit. on p. 29).

[MRK06] Sean M. McNee, John Riedl, and Jospeh A. Konstan. “Being Accurate Is not Enough: How Accuracy Metrics Have Hurt Recommender Systems.” In: *Extended Abstracts of the 2006 Conference on Human Factors in Computing Systems (CHI ’06)*. 2006, pp. 1097–1101 (cit. on p. 20).

[Nei+15] Julia Neidhardt, Leonhard Seyfang, Rainer Schuster, and Hannes Werthner. “A Picture-Based Approach to Recommender Systems.” In: *Information Technology & Tourism* 15.1 (2015), pp. 49–69 (cit. on p. 19).

[NFT13] Dario De Nart, Felice Ferrara, and Carlo Tasso. “Personalized Access to Scientific Publications: from Recommendation to Explanation.” In: *Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization (UMAP ’13)*. 2013, pp. 296–301 (cit. on pp. 23, 24).

[NH12] Maria Augusta S.N. Nunes and Rong Hu. “Personality-Based Recommender Systems: An Overview.” In: *Proceedings of the 6th ACM Conference on Recommender Systems (Recsys ’12)*. 2012, pp. 5–6 (cit. on p. 19).

[NHA09] Amine Naak, Hicham Hage, and Esma Aïmeur. “A Multi-criteria Collaborative Filtering Approach for Research Paper Recommendation in Papyres.” In: *Proceedings of the 4th International Conference on E-Technologies: Innovation in an Open World, (MCETECH ’09)*. 2009, pp. 25–39 (cit. on p. 16).

[NJ17] Ingrid Nunes and Dietmar Jannach. “A Systematic Review and Taxonomy of Explanations in Decision Support and Recommender Systems.” In: *User-Modeling and User-Adapted Interaction* 27.3–5 (2017), pp. 393–444 (cit. on p. 22).

[NLF10] Theodora Nanou, George Lekakos, and Konstantinos G. Fouskas. “The Effects of Recommendations’ Presentation on Persuasion and Satisfaction in a Movie Recommender System.” In: *Multimedia Systems* 16.4 (2010), pp. 219–230 (cit. on pp. 3, 21).

[NR17] Thuy Ngoc Nguyen and Francesco Ricci. “Dynamic Elicitation of User Preferences in a Chat-Based Group Recommender System.” In: *Proceedings of the Symposium on Applied Computing (SAC ’17)*. 2017, pp. 1685–1692 (cit. on p. 18).

[NS17] Koki Nagatani and Masahiro Sato. “Accurate and Diverse Recommendation based on Users’ Tendencies toward Temporal Item Popularity.” In: *Proceedings of the 1st Workshop on Temporal Reasoning in Recommender Systems (RecTemp ’17) at RecSys ’17*. 2017, pp. 35–39 (cit. on p. 54).

[NT14] Dario De Nart and Carlo Tasso. “A Personalized Concept-Driven Recommender System for Scientific Libraries.” In: *Proceedings of the 10th Italian Research Conference on Digital Libraries (IRCDL ’14)*. 2014, pp. 84–91 (cit. on p. 24).

[Nun+14] Ingrid Nunes, Simon Miles, Michael Luck, Simone Barbosa, and Carlos Lucena. “Pattern-based Explanation for Automated Decisions.” In: *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI ’14)*. 2014, pp. 669–674 (cit. on p. 63).

[ODo+08] John O’Donovan, Barry Smyth, Brynjar Gretarsson, Svetlin Bostandjiev, and Tobias Höllerer. “PeerChooser: Visual Interactive Recommendation.” In: *Proceedings of the 2008 SIGCHI Conference on Human Factors in Computing Systems (CHI ’08)*. 2008, pp. 1085–1088 (cit. on p. 23).

[Oh+11] Jinoh Oh, Sun Park, Hwanjo Yu, Min Song, and Seung-Taek Park. “Novel Recommendation Based on Personal Popularity Tendency.” In: *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM ’11)*. 2011, pp. 507–516 (cit. on pp. 45, 53, 54).

[PB07] Michael J. Pazzani and Daniel Billsus. In: *The Adaptive Web*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Springer, 2007. Chap. Content–Based Recommendation Systems, pp. 325–341 (cit. on pp. 2, 9).

[PB15] Denis Parra and Peter Brusilovsky. “User-Controllable personalization: A Case Study With SetFusion.” In: *International Journal of Human-Computer Studies* 78 (2015), pp. 43–67 (cit. on p. 24).

[PBT14] Denis Parra, Peter Brusilovsky, and Christoph Trattner. “See What You Want to See: Visual User-Driven Approach for Hybrid Recommendation.” In: *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI ’14)*. 2014, pp. 235–240 (cit. on pp. 21, 22, 24).

[Per+04] Evelien Perik, Boris de Ruyter, Panos Markopoulos, and Berry Eggen. “The Sensitivities of User Profile Information in Music Recommender Systems.” In: *Proceedings of the 2nd Annual Conference on Privacy, Security, and Trust (PST ’04)*. 2004, pp. 137–141 (cit. on p. 19).

[Per+18] Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O’Donovan, eds. *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18)*. 2018 (cit. on p. 3).

[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 (2012), pp. 357–397 (cit. on p. 15).

[Rei+04] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. “Dynamic Critiquing.” In: *Proceedings of the 7th European Conference on Advances in Case-Based Reasoning (ECCBR ’04)*. 2004, pp. 763–777 (cit. on p. 18).

[Rei+05] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. “Incremental Critiquing.” In: *Knowledge-Based Systems* 18.4-5 (2005), pp. 143–151 (cit. on p. 18).

[Ren+09] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. “BPR: Bayesian Personalized Ranking from Implicit Feedback.” In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI ’09)*. 2009, pp. 452–461 (cit. on p. 40).

[Ren10] Steffen Rendle. “Factorization Machines.” In: *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM ’10)*. 2010, pp. 995–1000 (cit. on p. 50).

[RH09] Elena Reutskaja and Robin M Hogarth. “Satisfaction in choice as a function of the number of alternatives: When ‘goods satiate’.” In: *Psychology & Marketing* 26.3 (2009), pp. 197–203 (cit. on p. 20).

[Rib+14] Marco Tulio Ribeiro, Nivio Ziviani, Edleno Silva De Moura, Itamar Hata, Anisio Lacerda, and Adriano Veloso. “Multiobjective Pareto-Efficient Approaches for Recommender Systems.” In: *Transactions on Intelligent Systems and Technology* 5.4 (2014), pp. 1–20 (cit. on p. 45).

[RRS11] Francesco Ricci, Lior Rokach, and Bracha Shapira. “Introduction to Recommender Systems Handbook.” In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Springer, 2011 (cit. on p. 63).

[RSZ13] M. Rossetti, F. Stella, and M. Zanker. “Towards Explaining Latent Factors With Topic Models in Collaborative Recommender Systems.” In: *Proceedings of the 24th International Workshop on Database and Expert Systems Applications (DEXA ’13)*. 2013, pp. 162–167 (cit. on p. 23).

[Rut+08] Lloyd Rutledge, Natalia Stash, Yiwen Wang, and Lora Aroyo. “Accuracy in Rating and Recommending Item Features.” In: *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH ’08)*. 2008, pp. 163–172 (cit. on p. 18).

[Sai+13] Alan Said, Ben Fields, Brijnesh J. Jain, and Sahin Albayrak. “User-Centric Evaluation of a K-Furthest Neighbor Collaborative Filtering Recommender Algorithm.” In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW ’13)*. 2013, pp. 1399–1408 (cit. on pp. 40, 45).

[Sar+01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. “Item-Based Collaborative Filtering Recommendation Algorithms.” In: *Proceedings of the 10th International Conference on World Wide Web (WWW ’01)*. 2001, pp. 285–295 (cit. on p. 2).

[SC12] A. M. Sarroff and M. Casey. “Modeling and Predicting Song Adjacencies In Commercial Albums.” In: *Proceedings of the 9th Sound and Music Computing Conference (SMC ’12)*. 2012, pp. 364–371 (cit. on p. 51).

[Sch+02] Barry Schwartz, Andrew Ward, John Monterosso, Sonja Lyubomirsky, Katherine White, and Darrin R. Lehman. “Maximizing versus satisficing: happiness is a matter of choice.” In: *Journal of personality and social psychology* 83.5 (2002), pp. 1178 (cit. on pp. 20, 62, 63, 65).

[Sch+06] Tobias Schnabel, Paul N. Bennett, Susan T. Dumais, and Thorsten Joachims. “Using Shortlists to Support Decision Making and Improve Recommender System Performance.” In: *Proceedings of the 25th International Conference on World Wide Web*. WWW ’16. 2006, pp. 987–997 (cit. on p. 55).

[Sch+07] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. “Collaborative Filtering Recommender Systems.” In: *The Adaptive Web*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Springer, 2007, pp. 291–324 (cit. on p. 29).

[Sch04] Barry Schwartz. *The Paradox of Choice: Why More Is Less*. Harper Perennial, 2004 (cit. on p. 20).

[SD04] Itamar Simonson and Aimee Drolet. “Anchoring Effects on Consumers’ Willingness-to-Pay and Willingness-to-Accept.” In: *Journal of Consumer Research* 31.3 (2004), pp. 681–690 (cit. on p. 56).

[SG11] Guy Shani and Asela Gunawardana. “Evaluating recommendation systems.” In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Springer, 2011, pp. 257–297 (cit. on p. 11).

[SGT10] Benjamin Scheibehenne, Rainer Greifeneder, and Peter M. Todd. “Can There Ever Be Too Many Options? A Meta-Analytic Review of Choice Overload.” In: *Journal of Consumer Research* 37.3 (2010), pp. 409–425 (cit. on p. 1).

[Sha13] Amit Sharma. “PopCore: A System for Metwork-Centric Recommendation.” In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW ’13)*. 2013, pp. 31–34 (cit. on p. 24).

[Shi+12] Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larson, and Alan Hanjalic. “Adaptive Diversification of Recommendation Results via Latent Factor Portfolio.” In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’12)*. 2012, pp. 175–184 (cit. on p. 45).

[SHO15] James Schaffer, Tobias Höllerer, and John O’Donovan. “Hypothetical Recommendation: A Study of Interactive Profile Manipulation Behavior for Recommender Systems.” In: *Proceedings of the 28th International Florida Artificial Intelligence Research Society Conference (FLAIRS ’15)*. 2015, pp. 507–512 (cit. on p. 23).

[SKR01] J. Ben Schafer, Joseph A. Konstan, and John Riedl. “E-commerce recommendation applications.” In: *Data Mining and Knowledge Discovery* 5.1/2 (2001), pp. 115–153 (cit. on pp. 10, 21).

[SKR02] J. Ben Schafer, Joseph A. Konstan, and John Riedl. “Meta-Recommendation Systems: User-Controlled Integration of Diverse Recommendations.” In: *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM ’02)*. 2002, pp. 43–51 (cit. on p. 24).

[SNM09] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. “MoviExplain: A Recommender System With Explanations.” In: *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys ’11)*. 2009, pp. 317–320 (cit. on p. 23).

[SS11] E. Isaac Sparling and Shilad Sen. “Rating: How Difficult Is It?” In: *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys ’11)*. 2011, pp. 149–156 (cit. on p. 15).

[SSV15] Cecilia di Sciascio, Vedran Sabol, and Eduardo E. Veas. “*uRank*: Exploring Document Recommendations Through an Interactive User-Driven Approach.” In: *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS ’15) at RecSys ’15*. 2015, pp. 29–36 (cit. on pp. 19, 21, 24).

[Ste18] Harald Steck. “Calibrated Recommendations.” In: *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18)*. 2018, pp. 154–162 (cit. on p. 54).

[SVR09] Shilad Sen, Jesse Vig, and John Riedl. “Tagommenders: Connecting Users to Items Through Tags.” In: *Proceedings of the 18th International Conference on World Wide Web (WWW ’09)*. 2009, pp. 671–680 (cit. on p. 19).

[SW07] Avni M. Shah and George Wolford. “Buying Behavior as a Function of Parametric Variation of Number of Choices.” In: *Psychological Science* 18.5 (2007), pp. 369–370 (cit. on p. 20).

[SZ08] Börkur Sigurbjörnsson and Roelof van Zwol. “Flickr Tag Recommendation Based on Collective Knowledge.” In: *Proceedings of the 17th International Conference on World Wide Web*. WWW ’08. 2008, pp. 327–336 (cit. on p. 29).

[TFI11] Erich Teppan, Alexander Felfernig, and Klaus Isak. “Decoy Effects in Financial Service E-Sales Systems.” In: *Joint Proceedings of the RecSys 2011 Workshop on Human Decision Making in Recommender Systems (Decisions@RecSys ’11) and User-Centric Evaluation of Recommender Systems and Their Interfaces-2 (UCERSTI ’11) at RecSys ’11*. 2011, pp. 1–8 (cit. on pp. 25, 55).

[Tin17] Nava Tintarev. “Presenting Diversity Aware Recommendations: Making Challenging News Acceptable.” In: *Proceedings of the 2017 Workshop on Responsible Recommendation (FATREC ’17) at RecSys ’17*. 2017 (cit. on p. 54).

[TK74] Amos Tversky and Daniel Kahneman. “Judgment under Uncertainty: Heuristics and Biases.” In: *Science* 185.4157 (1974), pp. 1124–1131 (cit. on p. 56).

[TM07] Nava Tintarev and Judith Masthoff. “A Survey of Explanations in Recommender Systems.” In: *Proceedings of the 23rd International Conference on Data Engineering Workshops (ICDE ’07)*. 2007, pp. 801–810 (cit. on pp. 25, 55).

[VCV11] Saul Vargas, Pablo Castells, and David Vallet. “Intent-Oriented Diversity in Recommender Systems.” In: *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’11)*. 2011, pp. 1211–1212 (cit. on p. 45).

[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 Lifesytle Change (LIFESTYLE ’12) and the 1st International Workshop on Interfaces for Recommender Systems (InterfaceRS ’12) at RecSys ’12*. 2012, pp. 56–59 (cit. on p. 21).

[VSR09] Jesse Vig, Shilad Sen, and John Riedl. “Tagsplanations: Explaining Recommendations Using Tags.” In: *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI ’09)*. 2009, pp. 47–56 (cit. on p. 23).

[Wea+15] Kimberlee Weaver, Kim Daniloski, Norbert Schwarz, and Keenan Cottone. “The Role of Social Comparison for Maximizers and Satisficers: Wanting the Best or Wanting to Be the Best?” In: *Journal of Consumer Psychology* 25.3 (2015), pp. 372–388 (cit. on p. 63).

[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 (cit. on pp. 62, 63).

[WKH98] Brian Wansink, Robert J. Kent, and Stephen J. Hoch. “An Anchoring and Adjustment Model of Purchase Quantity Decisions.” In: *Journal of Marketing Research* 35.1 (1998), pp. 71–81 (cit. on p. 56).

[Wör+11] Wolfgang Wörndl, Johannes Huebner, Roland Bader, and Daniel Gallego-Vico. “A Model for Proactivity in Mobile, Context-Aware Recommender Systems.” In: *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys ’11)*. 2011, pp. 273–276 (cit. on p. 25).

[WV14] Wesley Waldner and Julita Vassileva. “Emphasize, Don’t Filter!: Displaying Recommendations in Twitter Timelines.” In: *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys ’14)*. 2014, pp. 313–316 (cit. on pp. 21, 22).

[WWL13] Wolfgang Wörndl, Jan Weicker, and Béatrice Lamche. “Selecting Gestural User Interaction Patterns for Recommender Applications on Smartphones.” In: *Proceedings of the 3rd Workshop on Human Decision Making in Recommender Systems (Decisions ’13) at RecSys ’13*. 2013, pp. 17–20 (cit. on p. 15).

[XB07] Bo Xiao and Izak Benbasat. “E-commerce Product Recommendation Agents: Use, Characteristics, and Impact.” In: *MIS Quarterly* 31.1 (Mar. 2007), pp. 137–209 (cit. on p. 13).

[Yah+15] Alexandre Yahi, Antoine Chassang, Louis Raynaud, Hugo Duthil, and Duen Horng (Polo) Chau. “Aurigo: an Interactive Tour Planner for Personalized Itineraries.” In: *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI ’15)*. 2015, pp. 275–285 (cit. on pp. 18, 22).

[YY10] Chun-Ya Yang and Soe-Tsyr Yuan. “Color Imagery for Destination Recommendation in Regional Tourism.” In: *Proceedings of the 14th Pacific Asia Conference on Information Systems (PACIS ’10)*. 2010 (cit. on p. 21).

[Zan+06] Markus Zanker, Marcel Bricman, Sergiu Gordea, Dietmar Jannach, and Markus Jessenitschnig. “Persuasive Online-Selling in Quality and Taste Domains.” In: *Proceedings of the Seventh International Conference on E-Commerce and Web Technologies (EC-Web ’06)*. 2006, pp. 51–60 (cit. on p. 55).

[ZBP18] Zainab Zolaktaf, Reza Babanezhad, and Rachel Pottinger. “A Generic Top-N Recommendation Framework for Trading-Off Accuracy, Novelty, and Coverage.” In: *Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE ’18)*. 2018, pp. 149–160 (cit. on p. 54).

[ZH08] Mi Zhang and Neil Hurley. “Avoiding Monotony: Improving the Diversity of Recommendation Lists.” In: *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys ’08)*. 2008, pp. 123–130 (cit. on p. 45).

[Zha+12] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. “Auralist: Introducing Serendipity into Music Recommendation.” In: *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM ’12)*. 2012, pp. 13–22 (cit. on pp. 8, 45).

[Zho+10] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. “Solving the Apparent Diversity-Accuracy Dilemma of Recommender Systems.” In: *Proceedings of the National Academy of Sciences* 107.10 (2010), pp. 4511–4515 (cit. on p. 45).

[Zie+05] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. “Improving Recommendation Lists Through Topic Diversification.” In: *Proceedings of the 14th International Conference on World Wide Web (WWW ’05)*. 2005, pp. 22–32 (cit. on pp. 45, 47, 50).

[ZJP08] Jiyong Zhang, Nicolas Jones, and Pearl Pu. “A Visual Interface for Critiquing-Based Recommender Systems.” In: *Proceedings of the 9th Conference on Electronic Commerce (EC ’08)*. 2008, pp. 230–239 (cit. on pp. 18, 19).

[ZP06] Jiyong Zhang and Pearl Pu. “A Comparative Study of Compound Critique Generation in Conversational Recommender Systems.” In: *Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH ’06)*. 2006, pp. 234–243 (cit. on p. 18).

Web pages

[Ber89] Tim Berners-Lee. *Information Management: A Proposal*. Report (CERN). Last accessed: May, 2019. 1989. URL: `http://w3.org/History/1989/proposal.html` (cit. on p. 1).

[Fun06] Simon Funk. *Netflix Update: Try This at Home*. Last accessed: May, 2019. 2006. URL: `https://sifter.org/simon/journal/20061211.html` (cit. on pp. 2, 40).

[Hid+16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. *Session-based Recommendations With Recurrent Neural Networks*. Last accessed: May, 2019. 2016. URL: `https://arxiv.org/abs/1511.06939` (cit. on pp. 2, 8, 10).

[SS02] Kirsten Swearingen and Rashmi Sinha. *Interaction Design for Recommender Systems*. Unpublished working notes. Last accessed: May, 2019. 2002. URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.7347&rep=rep1&type=pdf` (cit. on p. 15).

List of Figures

List of Tables

Publications

In this thesis by publication, the following six works of the author are included. The full texts of these works can be found after this list.

- Michael Jugovac and Dietmar Jannach. “Interacting With Recommenders—Overview and Research Directions.” In: *Transactions on Interactive Intelligent Systems* 7.3 (2017), pp. 10:1–10:46
- Dietmar Jannach, Michael Jugovac, and Lukas Lerche. “Supporting the Design of Machine Learning Workflows With a Recommendation System.” In: *Transactions on Interactive Intelligent Systems* 6.1 (2016), pp. 8:1–8:35
- Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation.” In: *i-com* 14.1 (2015), pp. 29–39
- Michael Jugovac, Dietmar Jannach, and Lukas Lerche. “Efficient Optimization of Multiple Recommendation Quality Factors According to Individual User Tendencies.” In: *Expert Systems With Applications* 81 (2017), pp. 321–331
- Sören Köcher, Michael Jugovac, Dietmar Jannach, and Hartmut H. Holzmüller. “New Hidden Persuaders: An Investigation of Attribute-Level Anchoring Effects of Product Recommendations.” In: *Journal of Retailing* 95.1 (2018), pp. 24–41
- Michael Jugovac, Ingrid Nunes, and Dietmar Jannach. “Investigating the Deci-sion-Making Behavior of Maximizers and Satisficers in the Presence of Recommendations.” In: *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP ’18)*. 2018, pp. 279–283

In addition to these six main publications, the author of this thesis contributed to the following other publications related to recommender systems that are not part of this thesis.

- Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. “What Recommenders Recommend: An Analysis of Recommendation Biases and Possible Countermeasures.” In: *User Modeling and User-Adapted Interaction* 25.5 (2015), pp. 427–491

- Dietmar Jannach, Michael Jugovac, and Lukas Lerche. “Adaptive Recommendation-Based Modeling Support for Data Analysis Workflows.” In: *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI ’15)*. 2015, pp. 252–262
- Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Adaptation and Evaluation of Recommendations for Short-Term Shopping Goals.” In: *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15)*. 2015, pp. 211–218
- Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Item Familiarity Effects in User-Centric Evaluations of Recommender Systems.” In: *Poster Proceedings of the 9th ACM Conference on Recommender Systems (RecSys ’15)*. 2015
- Sören Köcher, Dietmar Jannach, Michael Jugovac, and Hartmut H. Holzmüller. “Investigating Mere-Presence Effects of Recommendations on the Consumer Choice Process.” In: *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS’ 16) at RecSys ’16*. 2016, pp. 2–5
- Dietmar Jannach, Sidra Naveed, and Michael Jugovac. “User Control in Recommender Systems: Overview and Interaction Challenges.” In: *Proceedings of the 17th International Conference on Electronic Commerce and Web Technologies (EC-Web ’16)*. 2016, pp. 21–33
- Dietmar Jannach, Ingrid Nunes, and Michael Jugovac. “Interacting With Recommender Systems.” In: *Companion Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI ’17)*. 2017, pp. 25–27
- Malte Ludewig, Michael Jugovac, and Dietmar Jannach. “A Light-Weight Approach to Recipient Determination When Recommending New Items.” In: *Proceedings of the RecSys Challenge Workshop at RecSys ’17*. 2017
- Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. “News Recommender Systems—Survey and Roads Ahead.” In: *Information Processing and Management* 54.6 (2018), pp. 1203–1227
- Michael Jugovac, Dietmar Jannach, and Mozhgan Karimi. “StreamingRec: A Framework for Benchmarking Stream-Based News Recommenders.” In: *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18)*. 2018, pp. 269–273

Interacting with Recommenders—Overview and Research Directions

MICHAEL JUGOVAC, TU Dortmund, Germany
DIETMAR JANNACH, TU Dortmund, Germany

ACM Reference Format:
Michael Jugovac and Dietmar Jannach. 2017. Interacting with Recommenders—Overview and Research Directions. *ACM Trans. Interact. Intell. Syst.* 7, 3, Article 10 (September 2017), 1 page. https://doi.org/10.1145/3001837

Authors' addresses: Michael Jugovac, TU Dortmund, Germany; Dietmar Jannach, TU Dortmund, Germany.

2160-6455/2017/9-ART10 $15.00
https://doi.org/10.1145/3001837

Supporting the Design of Machine Learning Workflows with a Recommendation System

DIETMAR JANNACH, TU Dortmund, Germany
MICHAEL JUGOVAC, TU Dortmund, Germany
LUKAS LERCHE, TU Dortmund, Germany

ACM Reference Format:
Dietmar Jannach, Michael Jugovac, and Lukas Lerche. 2016. Supporting the Design of Machine Learning Workflows with a Recommendation System. *ACM Trans. Interact. Intell. Syst.* 6, 1, Article 8 (February 2016), 1 page. https://doi.org/10.1145/2852082

Authors' addresses: Dietmar Jannach, TU Dortmund, Germany; Michael Jugovac, TU Dortmund, Germany; Lukas Lerche, TU Dortmund, Germany.

2160-6455/2016/2-ART8 $15.00
https://doi.org/10.1145/2852082

Research article

Dietmar Jannach*, Lukas Lerche, and Michael Jugovac

Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation

https://doi.org/10.1515/icom-2015-0018

*Corresponding author: Dietmar Jannach, TU Dortmund
Lukas Lerche, TU Dortmund
Michael Jugovac, TU Dortmund

Expert Systems With Applications 81 (2017) 321–331

Efficient Optimization of Multiple Recommendation Quality Factors According to Individual User Tendencies

Michael Jugovac[a,*], Dietmar Jannach[a], Lukas Lerche[a]

[a] *TU Dortmund, Department of Computer Science, Otto-Hahn-Str. 12, 44227, Dortmund, Germany*

[*] Corresponding author.

`http://doi.org/10.1016/j.eswa.2017.03.055`

Journal of Retailing 95 (1, 2019) 24–41

New Hidden Persuaders: An Investigation of Attribute-Level Anchoring Effects of Product Recommendations

Sören Köcher[a,*], Michael Jugovac[b], Dietmar Jannach[c], Hartmut H. Holzmüller[a]

[a] *TU Dortmund University, Faculty of Business and Economics, Department of Marketing, Otto-Hahn-Straße 6, 44227 Dortmund, Germany*
[b] *TU Dortmund University, Department of Computer Science, Otto-Hahn-Straße 12, 44227, Dortmund, Germany*
[c] *Alpen-Adria-Universität Klagenfurt, Faculty of Technical Sciences, Department of Applied Informatics, Universitätsstraße 65-67, 9020 Klagenfurt, Austria*

[*] Corresponding author.

https://doi.org/10.1016/j.jretai.2018.10.004

Investigating the Decision-Making Behavior of Maximizers and Satisficers in the Presence of Recommendations

Michael Jugovac
TU Dortmund
Dortmund, Germany
michael.jugovac@tu-dortmund.de

Ingrid Nunes
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil
ingridnunes@inf.ufrgs.br

Dietmar Jannach
Alpen-Adria-Universität
Klagenfurt, Austria
dietmar.jannach@aau.at

ABSTRACT

Psychological theory distinguishes between maximizing and satisficing decision-making styles. Maximizers tend to explore more or all alternatives when making a choice, while satisficers evaluate options until they find one that is good enough. There is limited research that examines how the existence of a recommender influences the choice process and decisions of different types of decision-makers. We report the results of a controlled study, in which we monitored the choice process of participants when provided with automated recommendations and different types of additional information regarding available options.

Our analyses show that *none* of the differences that were expected based on the literature manifested itself in the experiment. Maximizers neither inspected more items, nor invested more time to study them. Instead, like satisficers, they mostly picked one of the top-ranked items recommended by the system, which emphasizes the value of recommenders in particular for maximizers, who would otherwise face a more challenging decision problem. The analysis of the preferences of participants over different types of additional information revealed that highlighting key pros and cons was perceived as particularly helpful for the maximizers, an insight that can be used for the design of explanation approaches for recommenders.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Decision support systems*; • **Human-centered computing** → **User studies**;

KEYWORDS

Explanations; decision making policies; maximizer; satisficer

ACM Reference Format:
Michael Jugovac, Ingrid Nunes, and Dietmar Jannach. 2018. Investigating the Decision-Making Behavior of Maximizers and Satisficers in the Presence of Recommendations. In *UMAP '18: 26th Conference on User Modeling, Adaptation and Personalization, July 8–11, 2018, Singapore, Singapore.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3209219.3209252

1 INTRODUCTION

Psychological theory suggests that people adopt different decision-making styles [14]. At one extreme, there are *maximizers*, who have a tendency to explore many or all available options before making a decision, whereas *satisficers* usually only scan the options until they find a satisfactory one [5, 24]. Interestingly, even though maximizers, as a result, tend to invest more time, they are often less happy with their final decisions [10].

Helping users in the decision process is one of the major goals of recommender systems (RS) [11, 13, 22], and a large amount of evidence exists on the impact of these systems on decisions made by their users [12, 15, 27]. While the recommendations of such systems are in many cases personalized according to the assumed preferences or personality of users, recommenders usually do not take possible differences in the users' decision making styles into account. There are only a few approaches in this context [6, 18]. They investigate, for example, the value of providing different user interfaces for different types of consumers or user interfaces that adapt themselves, e.g., according to the consumers' assumed expertise.

The long-term goal of our research is to customize recommenders to better support users with different decision-making styles. Examples of possible customizations are the presentation of larger choice sets for maximizers or the provision of certain types of complementary information to support users in the decision-making process according to their decision-making style. In this paper, we make a step towards achieving this goal and investigate foundational aspects by means of a user study. A first main question addressed in our study is to what extent the observations regarding the choice behavior of maximizers and satisficers manifest themselves in the context of a recommendation scenario. Differently from studies in the field of psychology, the decision process in our application scenario is supported by a recommender system and the choice set is presented to the study participants in a certain order determined by the system. Furthermore, to investigate if maximizers and satisficers have different information needs, we provided participants with different types of explanations [20] and measured their acceptance and perceived value for the different user groups. Differently from typical user studies on explanations, instead of only asking participants about their subjective experience regarding, e.g., the choice difficulty, we also rely on objective measures such as the time needed to make a decision, the number of inspected items, or the position of the selected item in the recommendation list.

The rest of the paper is organized as follows. After discussing previous work and our expectations in Section 2, we report the design of our user study in Section 3. The results of our analysis and our conclusions are discussed in Sections 4 and 5.

UMAP '18, July 8–11, 2018, Singapore, Singapore

ACM ISBN 978-1-4503-5589-6/18/07…$15.00
https://doi.org/10.1145/3209219.3209252

2 BACKGROUND AND EXPECTATIONS

As mentioned, a number of psychology studies have shown that maximizers generally spend more time on making decisions, yet ultimately experience more regret and less satisfaction with their choices than satisficers [5, 24]. For example, in a study about job seeking behavior [10], in which maximizers secured higher-paying jobs on average, they felt more negative affect during the decision process and lower satisfaction with their (objectively better) choices. In contrast to our study, in which we collected objective measurements about the participants decision behavior, such as time taken, many of the influential studies on this topic rely on self-reported experiences or thought experiments.

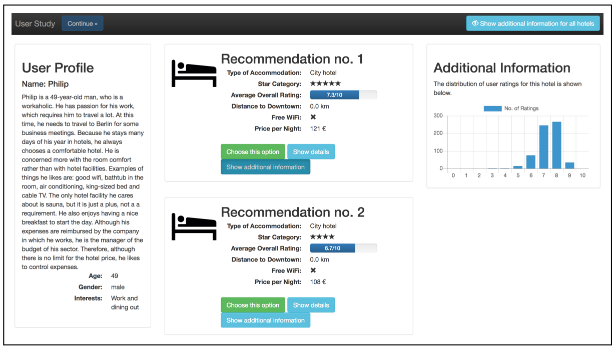
In the field of recommender systems, there is only limited research that takes the user's decision making style into account. For example, the users' decision making styles were considered in a study about diversification for recommender systems [26]. Based on the applied Structural Equation Model, the conclusion was that the maximization tendency of the participants did not affect any of the other measured variables, such as choice difficulty, recommendation attractiveness, or perceived diversity. Similarly, in a study about an energy-saving recommender system [18], maximizers and satisficers also showed no differences in terms of perceived control, interface satisfaction, recommender system effectiveness, etc. However, in contrast to the psychology literature, maximizers were actually *more* satisfied with their decisions.[1]

One hypothesis that could explain the above-mentioned observations is that the presence of a recommender system, which preranks the available options, influences the decision making behavior of users. In fact, changes in decision making policies have been previously observed, for example by Schnabel et al. [23], in which more users behaved like maximizers when they had access to a *shortlist* user interface element. In contrast, based on previous studies from the recommender systems literature, we expect that the presence of a recommender system can make more users behave like satisficers, i.e., engage less in search and comparison as would be expected for maximizers. Differently from previous studies, we also test this hypothesis based on *objective* measures, such as the decision time or the number of inspected items, which has not been done so far in recommender systems studies [18, 26].

Psychology literature, furthermore, suggests that maximizers and satisficers differ in their information needs. Maximizers, for example, tend to rely more on relative rather than absolute comparisons and consult external influences more frequently, such as expert opinions or social comparisons [24, 25]. Thus, in addition to examining the overall user decision behavior, we also investigate the effect of *additional explanatory information* provided during the decision making process. We compare three different explanation styles to find out whether they can be useful in the decision making process of maximizers or satisficers in different ways.

3 EXPERIMENTAL SETUP

In this section, we provide details about the study design and the recruited participants.



User Profile

Name: Philip

Philip is a 49-year-old man, who is a workaholic. He has passion for his work, which requires him to travel a lot. At this time, he needs to travel to Berlin for some business meetings. Because he stays many days of his year in hotels, he always chooses a comfortable hotel. He is concerned more with the room comfort rather than with hotel facilities. Examples of things he likes are: good wifi, bathtub in the room, air conditioning, king-sized bed and cable TV. The only hotel facility he cares about is sauna, but it is just a plus, not a a requirement. He also enjoys having a nice breakfast to start the day. Although his expenses are reimbursed by the company in which he works, he is the manager of the budget of his sector. Therefore, although there is no limit for the hotel price, he likes to control expenses.

Age: 49
Gender: male
Interests: Work and dining out

Recommendation no. 1

Type of Accommodation: City hotel
Star Category: ★★★★★
Average Overall Rating: 7.3/10
Distance to Downtown: 0.0 km
Free WiFi: ✖
Price per Night: 121 €

Recommendation no. 2

Type of Accommodation: City hotel
Star Category: ★★★★
Average Overall Rating: 6.7/10
Distance to Downtown: 0.0 km
Free WiFi: ✖
Price per Night: 108 €

Additional Information

The distribution of user ratings for this hotel is shown below.

Figure 1: User interface of the recommender system.

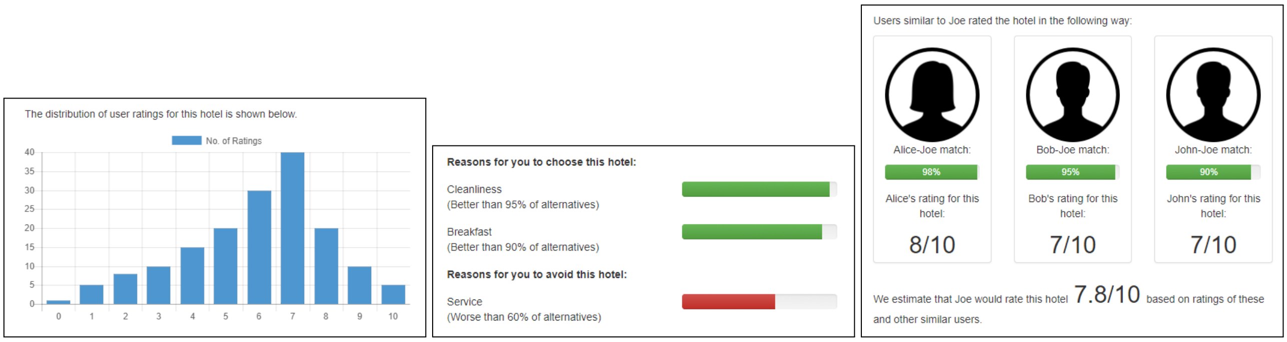
Study Tasks. Based on empirical evidence that maximizing and satisficing behavior also transfers to decisions that are made on behalf of others [4], we implemented a web-based system (Figure 1) through which participants were asked to choose a hotel for someone else. The fictitious profile of the target user was displayed along with a set of recommendations, which were computed using a user-based nearest neighbor algorithm on a historical dataset of hotel rating data. At the start of the experiment, an initial set of 10 recommendations was displayed, and users could request more (up to 40) recommendations, which allowed us to roughly track how many options users inspected before they made a decision.

To investigate if the different user groups have different information needs, participants could request "additional information" for each item. During the experiment, participants were asked to make decisions for three different profiles for three different cities. In each round, a different type of information (explanation) was presented (initially hidden), which could be used as a further basis for their decisions. The explanations (see Figure 2) were based on three fundamentally different knowledge sources: the quality perception of other consumers in the form of the rating distribution (Style A) [9], a pros-and-cons comparison with other hotels with respect to certain features (Style B) [16, 19, 21], and information about the recommendation process itself in terms of selected neighbor ratings (Style C) [1, 2, 8]. Before each of the three tasks, participants received a detailed interactive tutorial on the user interface, decision task, and additional information shown in the respective trial. The order of the profiles, cities, and types of additional information were randomized in a round-robin fashion during the trials.

After participants had made their three choices, they provided demographic data, after which they were shown three four-item questionnaires (using 7-point Likert scales) regarding the provided additional information types, namely, if they found the explanations transparent, useful, trustworthy, and if the information made them more confident in their choice. Finally, the participants answered the 13-item questionnaire proposed by Schwartz et al. [24], which we used to classify the participants into maximizers and satisficers. As usual in the field [10, 24], we used the median maximization scores to distinguish between maximizers and satisficers.

Study Variables. Overall, the *independent* variables are (i) the participants' decision making style (maximizer or satisficer) and (b) the investigated explanation styles, a *within-subjects* variable with three possible values.

[1] In another recommender systems study, the maximization tendency was measured, but in the final model the construct did not converge and was excluded [17].


The distribution of user ratings for this hotel is shown below.

Reasons for you to choose this hotel:

Cleanliness
(Better than 95% of alternatives)

Breakfast
(Better than 90% of alternatives)

Reasons for you to avoid this hotel:

Service
(Worse than 60% of alternatives)

Users similar to Joe rated the hotel in the following way:

Alice-Joe match:	Bob-Joe match:	John-Joe match:
98%	95%	90%
Alice's rating for this hotel:	Bob's rating for this hotel:	John's rating for this hotel:
8/10	7/10	7/10

We estimate that Joe would rate this hotel 7.8/10 based on ratings of these and other similar users.

(a) Popularity-based Explanation (Style A)

(b) Feature-based Explanation (Style B)

(c) Peer-based Explanation (Style C)

Figure 2: Examples of the three types of additional explanatory information evaluated in the user study.

As *dependent* variables, we used the following *objective* measures to assess differences in the decision-making behavior of the participants.

- The time needed to complete the decision-making task.
- The number of times the participant requested to see a page with detailed item information (detail requests).
- The number of requested recommendations. For this, we recorded how many of the four recommendation pages were loaded by the participant (recommendation requests).
- The number of times the participants requested to see additional explanatory information (explanation requests).
- The list position of the hotel chosen by the participants (choice index).[2]

As *subjective* measures, we analyzed the participants' responses to the four questionnaire items mentioned above regarding the value of the additionally displayed information.

Participants. We recruited participants via email lists and social networks. Between October and December 2017, 243 subjects from 5 different countries participated, and 109 completed the study. The majority of the participants were in their twenties. Most of them were from Brazil and Germany and had a computer science or information technology background. We excluded 19 subjects from the study, because they completed the process in an unreasonable short or long time. We used a threshold of 60 seconds for task completion, which we determined as the minimum time to read the target user profile and make a decision. Additionally, we excluded participants who took longer than 90 minutes for one of the tasks, assuming they likely focused on something else during the study.

4 OBSERVATIONS

Decision-Making Behavior of Maximizers and Satisficers. Table 1 shows the outcomes of our objective measurements. The results are provided in accumulated form as well as separated based on the participants' decision making styles and the provided explanation styles. To test if any of the observed differences across the different conditions were significant, we applied an ANOVA test or, in case its assumptions were not fulfilled, a Kruskal-Wallis test. The analyses showed that none of the differences were significant at a significance level of $p = 0.05$. In other words, independently of the explanation style, maximizers and satisficers did not differ significantly in their decision-making behavior in terms of the observed objective measures.

Thus, against our expectations and the existing research literature, maximizers did *not* take more time to make the decision, they did *not* look at more pages showing further alternatives, they did *not* inspect more item details or explanations, and they did *not* choose items further down the list than satisficers. In fact, from all participants, about 25% selected the first recommended hotel as their final choice, indicating that the recommendations were generally adopted well. The provision of different types of explanations also had no significant influence on their behavior. While we could not measure (e.g., through eye-tracking) how many alternatives the participants were looking at and for how long, the combination of measures (needed time, request for item details) suggest that there is no strong difference in the given sample.

As a result, our research leads to the hypothesis that the differences between maximizers and satisficers diminish or even disappear in the presence of recommendations, as was indicated in previous studies in the field of recommenders [18, 26]. One of the underlying reasons could be that maximizers (like satisficers) trust the recommendations and assume that there will be no better choices in the lower-ranked options. The recommendations in our study were, in fact, ordered by their assumed relevance for the given profiles, but there was no "objectively-best" ordering where one option strictly dominates another. Generally, the observed behavior might also be influenced by our everyday experiences, e.g., when using search engines, where users rarely inspect more than the first few pages [3]. Overall, to the best of our knowledge, no previous work in the field of psychology has examined the maximizer-satisficer theory for situations where the alternatives are pre-ordered according to some expected utility.

The Effect of Different Types of Additional Information. Table 2 shows the results obtained for the *subjective* measures regarding the different explanation styles. Combined with the objective results from Table 1, we can make the following observations.

Generally, we can observe that explanation style B (using pros and cons) received the highest absolute scores in all subgroups and all dimensions except trust. Considering the participants without distinguishing them based on their decision-making styles, the overall preference was for style B over styles A and C in terms

[2] The hotels in each city were presented in the same order for all participants.

Table 1: Mean (M), standard deviation (SD) and Median (Med) of scores obtained for objective variables. Choice=choice index, Detail=detail requests, Reco=recommendation requests, Expl=explanation requests. Results are shown separately by explanation style and combined in the last row of each group. Time is given in minutes.

Meas./	Expl.	Satisficer M±SD	Satisficer Med	Maximizer M±SD	Maximizer Med	All M±SD	All Med
Time	A	4.69± 3.50	3.55	4.69± 3.88	3.06	4.69± 3.67	3.18
	B	6.18± 9.29	4.12	5.08± 4.83	3.40	5.63± 7.38	3.87
	C	5.84± 6.29	3.79	4.85± 4.42	3.61	5.34± 5.43	3.64
	All	5.57± 6.77	3.84	4.87± 4.36	3.18	5.22± 5.69	3.58
Choice	A	5.47± 7.29	4	6.58± 6.31	6	6.02± 6.80	4
	B	7.87± 8.21	4	5.64± 5.78	4	6.76± 7.15	4
	C	5.60± 5.01	6	5.82± 8.37	2	5.71± 6.86	4
	All	6.31± 7.00	4	6.01± 6.87	4	6.16± 6.93	4
Detail	A	8.73±11.66	4	6.42± 6.00	5	7.58± 9.29	5
	B	7.04± 7.27	5	5.76± 7.53	3	6.40± 7.39	4
	C	6.73± 6.52	4	6.20± 7.81	4	6.47± 7.16	4
	All	7.50± 8.76	5	6.13± 7.11	4	6.81± 7.99	4
Reco.	A	0.76± 1.07	0	1.13± 1.32	1	0.9± 1.21	0.5
	B	0.89± 1.05	1	1.00± 1.43	0	0.94± 1.25	0
	C	0.69± 1.02	0	1.04± 1.45	0	0.87± 1.26	0
	All	0.78± 1.04	0	1.06± 1.39	0	0.92± 1.23	0
Expl.	A	7.82±10.39	3	9.56±12.87	4	8.69±11.66	3
	B	9.09± 9.49	6	10.56±10.91	10	9.82±10.19	7
	C	8.87±10.37	4	11.42±10.88	10	10.14±10.65	10
	All	8.59±10.03	4	10.51±11.53	8	9.55±10.83	5

Table 2: Mean (M), standard deviation (SD) and median (Med) scores obtained for each measurement across difference groups (explanation style and decision making policy).

Meas./	Expl.	Satisficer M±SD	Satisficer Med	Maximizer M±SD	Maximizer Med	All M±SD	All Med
Transp.	A	4.93±1.70	5	4.89±1.67	5	4.91±1.67	5
	B	5.56±1.14	6	5.73±1.07	6	5.64±1.10	6
	C	5.09±1.68	6	4.67±1.58	5	4.88±1.63	5
Useful.	A	4.80±1.60	5	5.11±1.35	5	4.96±1.48	5
	B	5.67±1.09	6	5.64±1.19	6	5.66±1.13	6
	C	5.13±1.34	5	5.00±1.12	5	5.11±1.23	5
Trust	A	4.93±1.44	5	5.07±1.36	5	5.00±1.39	5
	B	4.69±1.31	5	5.18±1.21	5	4.93±1.28	5
	C	4.22±1.43	4	4.22±1.28	4	4.22±1.35	4
Confid.	A	4.33±1.55	5	4.71±1.44	5	4.52±1.50	5
	B	5.11±1.27	5	5.33±1.24	5	5.22±1.25	5
	C	4.58±1.36	5	4.62±1.42	5	4.60±1.38	5

of transparency, usability, and confidence, with statistically significant differences (based on the corresponding statistical tests). Because style B is the only one that focuses on item features, this observation corroborates previous findings [7], in which different explanation styles were compared and "content-based" explanations were favored over, e.g., rating-based ones, in different dimensions. In contrast to this work, which used a slightly different visual representation, our pros-and-cons approach did not lead to a loss in decision efficiency, i.e., participants did not take more time when confronted with this type of explanations.

Nonetheless, even though the participants preferred explanations of style B (for example, in terms of usefulness), this did not lead to an increased actual use of the explanations. As mentioned earlier, the analysis of the results shown in Table 1 shows that users did not inspect significantly more explanations of style B than other explanation styles, leading to a gap between the participants' reported utility and their objectively observed behavior.

An interesting side-observation is that the "social" explanation style C received the lowest scores in terms of transparency, even though this style reveals the internal reasoning of the underlying recommender algorithm. This indicates that the participants had troubles understanding the meaning of what is presented in explanations of style C.

The differences between maximizers and satisficers in terms of their assessment of the different types of explanations were mostly small and not statistically significant. Statistical tests revealed only a significant preference of maximizers for style B over style C in terms of transparency. One reason for the maximizers' preference towards the feature-based comparison of the hotels could be their tendency to rely more on *relative* than absolute information, as previously observed [25]. Overall, except for this special case, in which maximizers seem to find feature-based explanations more transparent than peer-based information, maximizers and satisficers did not exhibit different information needs in the given scenario.

5 CONCLUSIONS AND IMPLICATIONS

In the presence of the recommender system that we employed in our user study, maximizers and satisficers did not exhibit significant differences in their observable decision-making behavior. This is in sharp contrast to existing psychology literature, but supports results of previous studies of this phenomenon in the context of recommender systems, which also observed no significant differences in terms of subjective measures. If these observations were generalizable, recommender systems could become a valuable assistive tool to mitigate the problems maximizers regularly face in decision-making tasks, such as negative affect and regret. However, further research is necessary to fully understand the effect that recommendations can have on users with different decision making styles, specifically in scenarios with different complexities, assortment sizes, and product domains.

Furthermore, the fact that participants from all groups preferred a pros-and-cons explanation style, which did not affect decision efficiency, could be a starting point for further detailed studies about the practical benefits of such explanations.

Finally, peer-based explanations received low scores overall, which is surprising, because maximizers are specifically known for engaging more in social comparison. Future work could focus on identifying the reason why maximizers did not prefer this social explanation type and which alternations should be made to better satisfy their information needs.

REFERENCES

[1] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. 2012. TasteWeights: A Visual Interactive Hybrid Recommender System. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. 35–42.

[2] Yu-Chih Chen, Yu-Shi Lin, Yu-Chun Shen, and Shou-De Lin. 2013. A Modified Random Walk Framework for Handling Negative Ratings and Generating Explanations. *Transactions on Intelligent Systems and Technology* 4, 1 (2013), 12:1–12:21.

[3] Chitika, Inc. 2013. The Value of Google Result Positioning. https://chitika.com/2013/06/07/the-value-of-google-result-positioning-2/. (2013). Retrieved February 20, 2018.

[4] Tilottama G. Chowdhury, S. Ratneshwar, and Praggyan Mohanty. 2009. The time-harried shopper: Exploring the differences between maximizers and satisficers. *Marketing Letters* 20, 2 (2009), 155–167.

[5] Ilan Dar-Nimrod, Catherine D. Rawn, Darrin R. Lehman, and Barry Schwartz. 2009. The Maximization Paradox: The costs of seeking alternatives. *Personality and Individual Differences* 46, 5 (2009), 631–635.

[6] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. 2006. An Integrated Environment for the Development of Knowledge-Based Recommender Applications. *International Journal of Electronic Commerce* 11, 2 (2006), 11–34.

[7] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How Should I Explain? A Comparison of Different Explanation Types for Recommender Systems. *International Journal of Human-Computer Studies* 72, 4 (2014), 367–382.

[8] Brynjar Gretarsson, John O'Donovan, Svetlin Bostandjiev, Christopher Hall, and Tobias Hollerer. 2010. SmallWorlds: Visualizing Social Recommendations. *Computer Graphics Forum* 29, 3 (2010), 833–842.

[9] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW '00)*. 241–250.

[10] Sheena S. Iyengar, Rachael E. Wells, and Barry Schwartz. 2006. Doing Better but Feeling Worse: Looking for the "Best" Job Undermines Satisfaction. *Psychological Science* 17, 2 (2006), 143–150.

[11] Dietmar Jannach and Gediminas Adomavicius. 2016. Recommendations with a Purpose. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. 7–10.

[12] Dietmar Jannach and Kolja Hegelich. 2009. A Case Study on the Effectiveness of Recommendations in the Mobile Internet. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. 205–208.

[13] Michael Jugovac and Dietmar Jannach. 2017. Interacting with Recommenders—Overview and Research Directions. *Transactions on Interactive Intelligent Systems* 7, 3 (2017), 10:1–10:46.

[14] Sahar Karimi, K. Nadia Papamichail, and Christopher P. Holland. 2015. The effect of prior knowledge and decision-making style on the online purchase decision-making process: A typology of consumer shopping behaviour. *Decision Support Systems* 77 (2015), 137 – 147.

[15] Evan Kirshenbaum, George Forman, and Michael Dugan. 2012. A Live Comparison of Methods for Personalized Article Recommendation at Forbes.com. In *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD '12)*. 51–66.

[16] David A. Klein and Edward H. Shortliffe. 1994. A Framework for Explaining Decision-theoretic Advice. *Artificial Intelligence* 67, 2 (1994), 201–243.

[17] Bart P. Knijnenburg, Svetlin Bostandjiev, John O'Donovan, and Alfred Kobsa. 2012. Inspectability and Control in Social Recommenders. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. 43–50.

[18] Bart P. Knijnenburg, Niels J.M. Reijmer, and Martijn C. Willemsen. 2011. Each to His Own: How Different Users Call for Different Interaction Methods in Recommender Systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys '11)*. 141–148.

[19] Khalil Muhammad, Aonghus Lawlor, Rachael Rafter, and Barry Smyth. 2015. Great Explanations: Opinionated Explanations for Recommendations. In *Proceedings of the 23rd International Conference on Case-Based Reasoning Research and Development (ICCBR '15)*. 244–258.

[20] Ingrid Nunes and Dietmar Jannach. 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction* 27, 3 (2017), 393–444.

[21] Ingrid Nunes, Simon Miles, Michael Luck, Simone Barbosa, and Carlos Lucena. 2014. Pattern-based Explanation for Automated Decisions. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI '14)*. 669–674.

[22] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer, Boston, MA, USA.

[23] Tobias Schnabel, Paul N. Bennett, Susan T. Dumais, and Thorsten Joachims. 2006. Using Shortlists to Support Decision Making and Improve Recommender System Performance. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. 987–997.

[24] Barry Schwartz, Andrew Ward, John Monterosso, Sonja Lyubomirsky, Katherine White, and Darrin R. Lehman. 2002. Maximizing versus satisficing: Happiness is a matter of choice. *Journal of Personality and Social Psychology* 83, 5 (2002), 1178–1197.

[25] Kimberlee Weaver, Kim Daniloski, Norbert Schwarz, and Keenan Cottone. 2015. The role of social comparison for maximizers and satisficers: Wanting the best or wanting to be the best? *Journal of Consumer Psychology* 25, 3 (2015), 372–388.

[26] Martijn C. Willemsen, Mark P. Graus, and Bart P. Knijnenburg. 2016. Understanding the role of latent feature diversification on choice difficulty and satisfaction. *User Modeling and User-Adapted Interaction* 26, 4 (2016), 347–389.

[27] Markus Zanker, Marcel Bricman, Sergiu Gordea, Dietmar Jannach, and Markus Jessenitschnig. 2006. Persuasive Online-Selling in Quality and Taste Domains. In *Proceedings of the Seventh International Conference on E-Commerce and Web Technologies (EC-Web '06)*. 51–60.