Using Implicit Feedback for Recommender Systems: Characteristics, Applications, and Challenges

Dissertation

zur Erlangung des Grades eines

D o k t o r s   d e r   N a t u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Lukas Lerche

Dortmund

2016

Tag der mündlichen Prüfung: 13.12.2016

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

Gutachter:
Prof. Dr. Dietmar Jannach
Prof. Dr. Markus Zanker

Abstract

Recommender systems are software tools to tackle the problem of *information overload* by helping *users* to find *items* that are most relevant for them within an often unmanageable set of choices. To create these *personalized* recommendations for a user, the algorithmic task of a recommender system is usually to quantify the user's interest in each item by predicting a relevance score, e.g., from the user's current situation or personal preferences in the past. Nowadays, recommender systems are used in various domains to recommend items such as products on e-commerce sites, movies and music on media portals, or people in social networks.

To assess the user's preferences, recommender systems proposed in past research often utilized *explicit feedback*, i.e., deliberately given ratings or like/dislike statements for items. In practice, however, in many of today's application domains of recommender systems this kind of information is not existent. Therefore, recommender systems have to rely on *implicit feedback* that is derived from the users' behavior and interactions with the system. This information can be extracted from navigation or transaction logs. Using implicit feedback leads to new challenges and open questions regarding, for example, the huge amount of signals to process, the ambiguity of the feedback, and the inevitable noise in the data. This *thesis by publication* explores some of these challenges and questions that have not been covered in previous research. The thesis is divided into two parts.

In the first part, the thesis reviews existing works on implicit feedback and recommender systems that exploit these signals, especially in the *Social Information Access* domain, which utilizes the "community wisdom" of the social web for recommendations. Common application scenarios for implicit feedback are discussed and a categorization scheme that classifies different types of observable user behavior is established. In addition, state-of-the-art algorithmic approaches for implicit feedback are examined that, e.g., interpret implicit signals directly or convert them to explicit ratings to be able to use "classic" recommendation approaches that were designed for explicit feedback.

The second part of the thesis comprises some of the author's publications that deal with selected challenges of implicit feedback based recommendations. These contain (i) a specialized learning-to-rank algorithm that can differentiate different levels of interest indicator strength in implicit signals, (ii) contextualized recommendation techniques for the e-commerce domain that adapt product suggestions to customers' current short-term goals as well as their long-term preferences, and (iii) intelligent reminding approaches that aim at the re-discovery of relevant items in a customer's browsing history. Furthermore, the last paper of the thesis provides an in-depth analysis of different biases of various recommendation algorithms. Especially the popularity bias, the tendency to recommend mostly popular items, can be problematic in practical settings and countermeasures to reduce this bias are proposed.

Contents

1 Introduction

Personalization has become ubiquitous in most parts of our digital life. Today, information like news, search results, and advertisements are not static content any more, but tailored to the preferences of each user. This kind of personalization is also a core feature of *recommender systems (RS)*. The algorithmic task of a recommender system is often considered to be to predict the relevance of certain *items* (products, documents, ...) for a *user* or to create an ordered list of recommended items of interest. The recommendations are based on various kinds of available information, e.g., a user's personal preferences or current situation. Nowadays, personalized recommendations are a key functionality on many modern websites and for mobile applications in domains like e-commerce, media repositories, social networks, or document-based information in general.

Historically, recommender systems can be classified into the two groups of *collaborative filtering* RS and *content-based filtering* RS, although more fine-grained classifications exist [Ric+11]. Collaborative filtering RS utilize the past interactions of all users to find those that share similar behavior (their *neighbors*). Recommendations are subsequently created by recommending items to a user that their most similar neighbors also liked in the past. Recommender systems that use content-based filtering, on the other hand, use certain characteristics of the items, e.g., their classification or metadata, to recommend items to a user that share similar characteristics with items the user already prefers. Often, a recommender system utilizes both collaborative and content-based filtering approaches and is therefore considered to be a *hybrid* technique [Jan+11].

To be able create recommendation in the first place, some information about the user has to be known to the recommender system. This *user profile*, which is necessary for personalized recommendations, usually contains a user's past interactions with the system, e.g., transaction histories or ratings given to items. In the past, research on recommender systems often focused on recommendation tasks that used so-called *explicit feedback*, e.g., deliberately given ratings or like/dislike statements for items, as the only form of user input to create the user profile [Jaw+14]. Based on these ratings, a recommender system could then predict the rating or relevance score a user might have given to an unseen item. Over time, many different approaches have emerged that were able to successfully leverage this explicit feedback to create useful recommendations that accurately reflect the user's interests. In practice, however, many domains do not provide the means for this kind of explicit user interaction. Also, in domains that support explicit feedback, often there are only few users that provide ratings and only few items that receive (a considerable amount of) ratings, so user profiles tend to be sparse [Jan+09; Jan+12a].

Instead of (only) relying on explicit user feedback, arbitrary user behavior and interactions with the system can be utilized to indirectly gain knowledge about the users' interests and preferences. This *implicit feedback* is available in many different domains, e.g., online shopping, social networks, and media services, as it generally can be extracted from navigation or transaction logs. However, other challenges arise. For example, implicit feedback can be ambiguous, i.e., it may not always be clear if the interaction of a user with an item should be interpreted as positive or negative feedback. Various approaches to use implicit feedback for recommender systems have emerged in recent years and some of the classic techniques that were proposed for explicit feedback and rating prediction can be modified and applied to work for implicit feedback domains such as e-commerce.

This *thesis by publication* explores various open questions and challenges that have not been covered in previous research. The thesis is based on some of the author's publications that are briefly summarized and discussed in the context of related work in the rest of this first chapter of the thesis. The topics include recommendations based on implicit feedback (Section 1.1), recommender systems for the e-commerce domain (Section 1.2), and the popularity bias of recommender algorithms (Section 1.3). The full texts of the publications can be found in the appendix. The Chapters 2 to 4 of this thesis, which are also based on a publication by the author, provide an overview of the current state of research on implicit feedback RS, discuss implicit feedback in comparison with explicit feedback, show recent algorithmic approaches, and propose a classification of implicit feedback signal types based on various usage scenarios and domains.

1.1 Recommendations Based on Implicit Feedback

To give a short overview of these different topics regarding implicit feedback, the content of the chapters 2 to 4 is briefly summarized here. The chapters are based on a section of the forthcoming book *Social Information Access* by Springer [Bru+17].

As mentioned in the introduction, the use of implicit feedback in recommender systems faces many different challenges. Especially in the *Social Information Access* domain, which focuses on information search and retrieval using the "community wisdom" of the social web [Bru08], recommender systems for implicit feedback have become important tools. In this domain, explicit feedback signals, e.g., ratings, are less common compared to implicit feedback that is often based on user actions like sharing, tagging, commenting on social networks, or checking-in with location-based applications. The following topics will be discussed later on in Chapters 2 to 4.

Chapter 2: Implicit Feedback – An Introduction

This chapter characterizes the differences between explicit and implicit feedback signals. It shows that explicit and implicit feedback cannot be expressed in terms of Boolean categories but rather as a continuum, since explicit feedback can also to some degree be used to infer implicit preferences. A historical categorization of implicit feedback types is presented and challenges when using implicit signals for recommendations are discussed, e.g., regarding the ambiguity of implicit feedback as positive or negative signals.

Chapter 3: Categories of Observable Implicit Feedback Signals

Based on an overview of recent literature, this chapter reviews application domains in which implicit feedback is the primary form of user feedback. The domains are discussed in the context of recent trends, e.g., the rise of the social web, e-commerce applications, and the development of ubiquitous services and devices. An extension of the previously examined categorization scheme for implicit feedback is proposed that contains additional categories to reflect these technological advancements.

Chapter 4: Algorithms for Implicit Feedback Situations

This chapter covers algorithmic approaches that can be used to generate recommendations based on implicit feedback signals. Apart from basic strategies to interpret implicit feedback or convert it to explicit signals, *association rule mining* and specialized *one-class collaborative filtering* techniques from the recent literature are discussed. In addition, hybrid approaches that combine explicit and implicit feedback signals are examined.

Among other techniques, the chapter also includes a discussion of the author's BPR^{++} extension to the *Bayesian Personalized Ranking* (BPR) algorithm [Ren+09] proposed in [Ler+14]. BPR is a recent recommendation technique for implicit feedback. Compared to BPR, BPR^{++} also supports the use of *graded implicit feedback* as opposed to only unary signals and therefore allows distinguishing between signals of different strengths, e.g., when an interaction occurred more often or more recently than another. The interpretation of the implicit feedback becomes more fine-grained and the additional knowledge can lead to more accurate recommendations of BPR^{++} compared to BPR.

1.2 E-Commerce Recommender Systems

A common application domain where implicit feedback signals are prevalent is *e-commerce*. When companies started to offer their goods online, they began to use recommender systems to personalize the customer experience. Their goal was to recreate the guidance that would normally be offered by a salesperson in a store. Today, recommender systems have become a common and successful feature on most shopping sites and can create additional value, e.g., by helping consumers to discover new products matching their shopping interests or by giving the retailers the means to promote certain goods. In practice, what should be recommended largely depends on the companies' business goals [Sai+13; Sai+14b]. For example, a recommender system might be used to help users discover new products and promote new item categories [Dia+08]. It can also be used to provide homogeneous suggestions that match the user's current shopping goals [Jan+15a] or even to remind users of items that they already know [Ler+16]. These last two publications [Jan+15a] and [Ler+16] are also included in this thesis. In the following, they are briefly discussed in the context of related works.

1.2.1 Short-Term Recommenders And Reminders

In the first paper [Jan+15a], a number of hybrid strategies for the e-commerce domain are proposed that adapt recommendations that are generated by different baseline RS to the user *context*, e.g., the user's current navigation history within the online store. The underlying assumption is that while customers have a general preference for certain items, brands, or product characteristics, they usually arrive at a store with a specific *short-term* shopping intent in mind. Therefore, the presented techniques combine baseline recommender algorithms that model the users' long-term preferences with short-term techniques to account for the users' current goals in a shopping session. The strategies are benchmarked on two real-world e-commerce datasets using a novel evaluation scheme designed to simulate the temporal aspects of the recommendation process. The results show that the choice of the long-term baseline strategy is particularly important at the beginning of new shopping sessions. In addition, the proposed recency- and content-based short-term adaptation strategies have a high predictive accuracy in the tested domains.

The second publication [Ler+16] continues the previous work from [Jan+15a] discussed above and focuses on using *reminders* as recommendations in the e-commerce domain. Typical recommender systems in research recommend only items that are new to the users. However, this might not always be the preferred strategy in practice. The intended goal of the proposed reminders is not to show new items for discovery or catalog exploration, but to present users with already known items that were of (recent) interest to them but might be forgotten. In contrast to the comparably simple reminding strategies introduced in the previous work [Jan+15a], more elaborate reminding techniques are proposed that utilize, e.g., the similarity between items and sessions in the user's transaction history, as well as the intensity of item interactions to determine suitable reminders. In addition, the previously proposed evaluation scheme is further extended to decrease the number of "too obvious" recommendations. The results indicate that reminders work well in terms of predicting purchases, at least in offline evaluation settings. Since reminding strategies by design do not lead to the discovery of new items, measuring their true value is challenging. Therefore, the results are backed by a field study and the analysis of navigation logs from two e-commerce sites.

1.2.2 Related Works on Contextualization and E-Commerce

The role of recommender systems in e-commerce and how the recommendations can be personalized and contextualized for the users has been discussed in the past. The works in [Sch+99] and [Sch+01a] are examples of early overviews on e-commerce recommender systems. The authors review a number of recommender systems used in practice, e.g., on Amazon or eBay, and discuss application models as well as open research problems. They also propose a taxonomy for e-commerce recommendations with the two key dimensions *(1) degree of automation* (Are the recommendation automatically displayed or does the user explicitly have to request them?) and *(2) degree of persistence* (Are the recommendations only based on the current user session or do they take previous interests and other customers into account?). In terms of the *degree of automation*, most of today's recommender systems on e-commerce websites tend to be fairly automatic, integrated into the user interface, and do not require the user to specifically request recommendations. On the other hand, the *degree of persistence* describes the way how contextualization is used in a recommender systems and this still remains a key aspect when employing recommendations in an e-commerce system. This relationship between short-term contextualization and long-term personalization was discussed in the author's paper [Jan+15a] described above. The proposed hybrid approach tries to adapt the recommendations to the current shopping goal of the user and might identify a recent interest drift, while at the same time it provides suggestions that match the general preferences of the user. The reminding strategies presented in [Ler+16] are also means of contextualization, as only items that are relevant for the current situation of the user can make good reminders.

According to [Ado+11], the techniques proposed in the two publications discussed above classify as "contextual post-filtering" strategies. Research on short-term contextualization for collaborative filtering approaches is, however, comparably scarce, although some similar works exist, e.g., [Har+15a], [Ren+10a], and [Tav+14]. These also propose different strategies to understand the user's short-term shopping intents, but they do not take reminders into consideration. One of the few examples from research that discusses reminders is [Sch+15] where the value of user-controlled "shortlists" is evaluated. These shortlists can help users to organize shopping sessions and remind them of forgotten products.

1.3 Evaluation and Biases of Recommendation Algorithms

Success in the e-commerce domain is usually quantified in terms of some business metrics, like the revenue or a click-through rate. However, the evaluation of recommender systems in research strongly focuses on offline experimental setups and optimizing predictive accuracy. For example, over 85% of 63 works of the full paper proceedings of the ACM RecSys conference in 2014 and 2015 employ offline evaluation exclusively [Jan+16b]. Only five papers [Bas+14; Per+14; Eks+14; Har+15b; Zha+15] conducted a laboratory study or employed a crowdsourcing platform to evaluate their approaches and only three works [Liu+14; Gar+14; Eks+15] include an online A/B test in a real world system. This focus on offline accuracy optimization originated from the fact that recommender systems research emerged from the field of Information Retrieval [Her+04].

While the predictive accuracy of an algorithm can be an indicator for its performance in practice [GU+15], it essentially only reflects how well it can predict – or rather "postdict" – the hidden signals or ratings of items that were already in a dataset, i.e., the status quo of the data at the time of recording [Zhe+10; Cre+10; Ste11; Jan+16c]. By design, the relevance of item new to a user cannot be assessed. In addition, if the data originated from a domain that is very popularity- or recency-oriented, like movies, news [Kir+12; Gar+14], or the blog-like web pages [Jan+15a], the algorithms that have a *bias* to recommend popular or recent items usually show very good predictive performance, as they recreate trends that were present in the original data. In the following, this bias will be discussed briefly and put the context of related works on the real-world performance of recommender algorithms.

1.3.1 Popularity Bias and Countermeasures

The *popularity bias* of different recommendation algorithms and possible countermeasures were examined in [Jan+15f], which is a publication included in this thesis. This paper includes a multidimensional evaluation of common and popular recommendation techniques from research regarding, e.g., their accuracy, catalog coverage, and bias to recommend popular items. The analyses show, among others, that although the predictive accuracy of many techniques is similar, their actual recommendations can differ. Additional simulations and parameter tuning experiments indicate that some of the most accurate algorithms have a strong popularity bias, which leads to the recommendation of mostly popular items. Since this bias might be undesirable in practice, the article also proposes two novel techniques as countermeasures. One of these approaches is designed for the BPR algorithm mentioned earlier and it is able to reduce the bias with only a small trade-off in predictive accuracy. The approach is discussed in more detail later on in Chapter 4 of this thesis.

1.3.2 Related Works on Real-World Recommendation Performance

Although algorithms with a popularity bias generally work well for offline settings in terms of predictive accuracy, this bias might not be desired in practice, especially in the e-commerce domain. The promotion of highly popular items and top-sellers can contradict the vendor's intention to show diverse recommendation and promote niche items, as well as prevent customers from exploring the product catalog and finding new and serendipitous items. As discussed earlier, the suggested products should also sometimes match the user's short-terms goals, for example, by employing context-sensitive recommendations and reminders.

A number of works exist that point out the discrepancy between offline results and online performance. In [Gar+14], different algorithms for news recommendations are compared in an offline experiment and a live A/B test. It was shown that, in the offline setup, popularity-based recommendations performed best in terms of predictive accuracy. However, in the online scenario, a context-based technique led to the best click-through rates and visit durations on the news site. The work in [Kir+12] also discusses a large-scale news recommender systems and, similar to [Gar+14] and other works [Zhe+10; Cre+10; Ste11], the authors report that results of offline experiments may not correlate with the actual performance of recommendations techniques in productive use because the requirements for good recommendations in practice are very domain dependent. A case-study on the recommendation of mobile games in a digital store was carried out in [Jan+09]. In a live experiment, a content-based strategy worked best to increase the revenue, while a CF-based method led to the highest increase in the click-through rate and both strategies were able to beat the manually selected recommendations that were used before. Again, the authors state that the online results are not representative of the preliminary offline experiments.

In [Cre+11; Cre+12], an offline/online comparison of several recommendation algorithms shows that none of the algorithms performs significantly better or worse than any of the others in terms of perceived satisfaction in the online experiment. On the other hand, in terms of offline accuracy there are significant differences. The work in [GU+15] discusses the employed recommendation and evaluation techniques on the Netflix video-on-demand platform. The authors state that offline experiments can help to prune the number of recommendation algorithms and configurations that should be tested in practice, but they do not help to accurately predict which algorithm will perform best in productive use. Therefore, employing large-scale A/B tests is essential. However, even A/B tests can have certain drawbacks because there are usually external factors involved that can lead to noise in the results of the tests.

1.4 Publications

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

1.4.1 What Recommenders Recommend

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 (Dec. 2015), pp. 427–491

The research is a joint effort with Dietmar Jannach, Iman Kamehkhosh, and Michael Jugovac. The author of this thesis wrote parts of the text and his specific contributions are the adaptable sampling strategy for BPR and most of the experimentation, analysis, and result interpretation. Some of the results have also appeared in a previous publication [Jan+13c].

At the UMAP '16 conference the article was awarded with the *2015 James Chen Award for UMUAI Best Paper*.

1.4.2 Using Graded Implicit Feedback for Bayesian Personalized Ranking

Lukas Lerche and Dietmar Jannach. "Using Graded Implicit Feedback for Bayesian Personalized Ranking". In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, CA, USA). RecSys '14. 2014, pp. 353–356

The paper was written together with Dietmar Jannach. The BPR++ algorithm was designed, implemented, and evaluated by the author of this thesis.

1.4.3 Adaptation and Evaluation of Recommendations for Short-Term Shopping Goals

Dietmar Jannach, Lukas Lerche, and Michael Jugovac. "Adaptation and Evaluation of Recommendations for Short-term Shopping Goals". In: *Proceedings of the 2015 ACM Conference on Recommender Systems*. (Vienna, Austria). RecSys '15. 2015, pp. 211–218

The paper is the result of joint work with Dietmar Jannach and Michael Jugovac. The hybrid recommendation techniques were designed and evaluated by the author of this thesis who also contributed to the design of the evaluation scheme. First results of this research were also presented in a workshop paper [Jan+13b].

1.4.4 On the Value of Reminders within E-Commerce Recommendations

Lukas Lerche, Dietmar Jannach, and Malte Ludewig. "On the Value of Reminders within E-Commerce Recommendations". In: *Proceedings of the 24st International Conference on User Modeling, Adaptation and Personalization*. (Halifax, NS, Canada). UMAP '16. 2016

The research is a joint work with Dietmar Jannach, Malte Ludewig. Aaron Larisch contributed to the live experiments in the context of a Bachelor thesis project. The author of this thesis contributed to all parts of the paper and wrote major fractions of the text.

The paper was awarded with the UMAP '16 *James Chen Best Student Paper Award*.

1.4.5 Recommending Based on Implicit Feedback

Dietmar Jannach, Lukas Lerche, and Markus Zanker. "Recommending Based on Implicit Feedback". In: *Social Information Access*. Ed. by Peter Brusilovsky and Daqing He. Vol. 10100. LNCS. Heidelberg: Springer, 2017. Chap. 14

The publication was a joint effort with Dietmar Jannach and Markus Zanker. The author of this thesis contributed to Sections 2 and 3. Section 4 was mainly written by the author of this thesis as well as the discussion of the *case studies on e-commerce* in Section 6.1 and the *application-specific requirements* in Section 6.3. As mentioned before, the introductory Chapters 2 to 4 of this thesis are based on Sections 2 to 5 of the publication.

2 Implicit Feedback – An Introduction

As mentioned in the introduction, recommendations are a key functionality on many modern websites and mobile applications. Typically, the task of recommendation components within applications is to point users to additional items of interest by ranking or filtering them according to the past preferences and the current contextual situation of these users. The recommendations are created based on user feedback, which can be available as *explicit* or *implicit feedback*. In the following, these two signal types, which were briefly explained in the beginning of this thesis, will be explored in more detail.

Mainstream research in the field of recommender systems was historically fueled by applications scenarios in which preference statements of users in the form of *explicit item ratings* are available [Jaw+14]. This led to the development of sophisticated algorithms that are able to very accurately predict which rating a user would probably give to a certain item. Much of the power of these algorithms is based on the existence of large datasets of historical ratings, in which for each user dozens of explicit ratings exist. Since the evaluation is often only done on this historical offline data, the algorithms are optimized to accurately "postdict" recommendations rather than to predict, which may or may not overlap with real-world performance. While there exist a number of dedicated social web platforms on which users can rate movies, books, restaurants, or other businesses, there are also many real-world application domains in which rating matrices of explicit feedback are very sparse or even non-existent today [Jan+09; Jan+12a]. For example, while some popular items on Amazon.com receive many ratings, most of the items in the catalog do not have any ratings. In domains like friend discovery for social networks no explicit rating matrices exist at all, since people usually cannot be rated directly by other users.

When building personalized recommenders in such application domains, *indirect ways* of assessing the interests and preferences of users by monitoring and interpreting their actions and behavior have to be employed. In the research literature, these observations of a user's actions are called "implicit feedback". They are interpreted as *statements on the relevance of a particular item*. Sometimes also the term "nonintrusive" feedback is used because users are not explicitly stating their preferences, but these are derived from their observed actions. In a classic e-commerce setting, an example of a user action that might indicate a preference for an item occurs when the user views the detailed product description of an item or puts the item on a wish list. On media streaming platforms, the repeated consumption of a track or music video can be interpreted as an interest or preference of the user toward the track itself, the track's artist, or the genre. On a social network, sharing a certain news story in a post might express the user's general interest in the topic.

Implicit and explicit feedback are, however, not a set of Boolean categories, but rather a continuum. Consider the case of a user playing a music track or sharing a news article. These actions can be interpreted as implicit feedback, i.e., the user might have a preference towards the track or the article contents. It might also be inferred from the user actions that the user is interested in the track's artists or the topic of the news story. However, if the user (explicitly) gave the track a rating or "liked" the news article, it could also (implicitly) be inferred that there might be an interest in the artist or topic. Therefore, when using the term *implicit feedback*, this includes all kinds of interactions with the systems from which user preferences can be *inferred indirectly*.

Open Problems

In reality, the amount of available implicit preference signals can be huge. Today, every mouse move of a user can in theory be tracked in an online application. In the future, with the continuing development of the *Internet of Things* and users being "always-on" by means of mobile or wearable devices, even larger amounts of information about the users' behavior and about the objects with which they interact will be available.

Besides the technical challenge of efficiently processing such a constant stream of possibly large amounts of data, a number of further questions has to be addressed. These questions include, for example, which of the many types of signals should be used to build a preference profile and how to combine these signals with possibly existing explicit rating information. Furthermore, different types of signals might indicate a different strength of a preference, i.e., a purchase may count more than an item view action in an online store. Finally, implicit feedback signals are often positive-only and in addition interpreting the signals correctly is challenging as, e.g., an online shopper can be disappointed later on with a purchase or was purchasing something for a friend.

Overall, recommendation based on implicit feedback in real-world applications is much more common than relying (solely) on explicit ratings, e.g., because the acquisition of ratings requires certain efforts from the user's side. A general problem of explicit ratings is that many users see ratings as a means to assess the *quality* of an item, e.g., a movie, rather than to express their *enjoyment*, which is probably more relevant in a recommendation scenario. Thus, recommendations based on the true user behavior might in fact be more reliable than predictions that are based on explicit ratings in reality.

Explicit vs. Implicit Feedback

Explicit feedback can be seen as a quality assessment by a user that is deliberate and unambiguous. The interpretation of the assessment is always dependent on the domain and the context in which it is given. For recommender systems, explicit feedback is often a (numerical) rating assigned by a user for one specific item. This rating is often given in the context of the corresponding recommendation task, e.g., to indicate the relevance of some items in a specific situation.

Implicit feedback, on the other hand, contains different kinds of user interactions that are not necessarily intended to provide a deliberate assessment of the system but can nevertheless be exploited to infer the user's positive or negative opinion. In recommender systems, for instance, this feedback can be used to determine if a user has a positive or negative attitude towards an item or multiple similar items. However, implicit feedback can often only be interpreted with a degree of uncertainty, as it might not be universally clear whether the recorded interaction signals are positive, negative, or somewhere in between. For example, the viewing duration of an item can be a signal of increased interest but, at the same time, it can be interpreted as the level of difficulty to understand the content. The signal might be not even meaningful at all when a user was distracted by something else. In addition, even if an implicit feedback signal can be interpreted accurately, there is still the question how to quantify and how to combine it with other types of implicit and explicit feedback.

2.1 Explicit Feedback Signals

The most prominent form of explicit feedback in the literature are user-provided *ratings*, e.g., on a 1-to-5 scale often displayed as "stars". In most settings, only one overall rating per item is available. In multi-criteria recommendation approaches, more fine-grained rating feedback regarding different quality dimensions of the items is employed.

Apart from star ratings, other common forms of explicit feedback are unary "like" or "recommend to a friend" statements as well as binary "thumbs up/down" selections. In certain applications, there are also explicit negative user actions such as "banning" a track on a music streaming platform or blocking or hiding certain messages on a Social Web platform. Although the latter signals are unary or binary, they are not *implicit* feedback. Sometimes these aspects are confused as most implicit feedback algorithms only rely on unary or binary signals and can therefore be applied for these feedback types as well, as mentioned in the discussion of implicit feedback algorithms later on in Section 4.2.

Besides these directly processable preference expressions, there are other forms of explicit feedback. However, these require further analysis or are application-specific. On the Social Web, users can for example express their opinions through reviews in natural language or by annotating items with tags that have a (known) positive or negative connotation. An example for application-specific explicit feedback would be that a user of an online bookstore puts a book on a "recommended reading" list. Also, adding a browser bookmark for a website *can* be an explicit statement in case the bookmark is put into a folder with a clear positive or negative connotation, e.g., "My Favorites".

The distinction between the different feedback types for these latter cases can however be a continuum and it might be possible to infer further implicit preferences from explicit statements. For example, any bookmarking action is never an explicit feedback signal, independent of the fact that the user's quality assessment for the item can be potentially unambiguously derived. The users' intention is *not* to inform the system about their preferences in the first place. Such an argument could also be raised for explicit star ratings, where the user's main intention might be to use the rating as a personal reminder for themselves or to share their experiences with other users and not state their opinion in the first place.

In addition, explicit rating information may be sparse as such ratings require extra work by the users, who might not immediately see the benefit of specifying their preferences. Furthermore, providing an explicit rating requires a considerable amount of cognitive effort by the users and some might be challenged in expressing their preferences using a single rating on a pre-defined and often coarse scale, as reported, e.g., in [Whi+05]. The work discusses different factors that impact the utility of implicit feedback in search systems. One result was that the user preference towards giving explicit or implicit feedback is highly influenced by the complexity of the interaction task, i.e., for complex search tasks users preferred to provide implicit feedback. For these search tasks, e.g., browsing an online store, the search task itself was the focus of the user and individual items were not all evaluated by them. On the other hand, in domains like digital media & entertainment, where the search task is less demanding, users might be willing to give explicit feedback more often.

2.2 Classification of Implicit Feedback

As mentioned before, implicit feedback subsumes all sorts of user actions and behavior that were not intentionally executed in order to provide feedback on specific items or the system performance in general. These implicit signals can be observed either directly or indirectly and are worthwhile to exploit in order to infer a positive or negative user bias towards a specific item, towards items with specific characteristics, or towards a specific action taken by the system. Usually, one of the tasks when using implicit feedback is to find a suitable way of interpreting the feedback, for example, by mapping it onto a rating scale or by learning relative (pair-wise) preference models, which will be discussed later on in Chapter 4.

2.2.1 Directly Observable User Actions

These interpretable signals typically are observable user actions, e.g., when users view or purchase something at an online store, when they select news articles of certain topics, when they listen to a track on a music streaming portal, when they tag or bookmark a resource, or join a group on a social network. The user's navigation behavior – from category browsing to mouse and eye movements – represents another typical category of implicit feedback.

An early categorization of possible types of observable implicit feedback signals – focusing on information filtering and recommendation – can be found in [Nic97]. This classification was later extended by Oard and Kim in [Oar+98], who identified three types of observable behavior: ***Examination***, ***Retention***, and ***Reference***. Later on, in [Oar01], a fourth category – ***Annotation*** – was added, which in some sense unifies implicit and explicit feedback based on the types of observable behavior [Jaw+14]. In the bibliographical review presented in [Kel+03], the authors introduce a fifth dimension called ***Create***, which relates to the user activity of writing or to editing an original piece of information, e.g., as seen in [Hil+92] or [Bud+99]. The five categories of implicit feedback are summarized in Table 2.1.

The development of recommendation technology over the last two decades suggests that this classification should be extended. Therefore, these five types of observable behavior will be discussed again later on in Section 3.2 after a review of recent research on implicit feedback in recommender systems and related fields. A suitable extension is presented later on in Table 3.1.

Tab. 2.1: Summary of the five types of observable behavior, adapted from Oard and Kim in [Oar01], [Oar+98], and [Kel+03].

Category	Examples of Observable Behavior
Examination	Duration of viewing time, repeated consumption, selection of text parts, dwell time at specific locations in a document, purchase, or subscription.
Retention	Preparation for future use by bookmarking or saving a named or annotated reference, printing, deleting.
Reference	Establishing a link between objects. Forwarding a document and replying, creating hyperlinks between documents, referencing documents.
Annotation	Annotate, rate or publish an object (includes explicit feedback).
Create	Write or edit a document.

2.2.2 Feature-Related Indirect Preference Signals

Implicit feedback for an item can also be inferred from indirect preference signals that are based on explicit feedback (ratings) on related objects or from other user actions that are not directly related to a specific item. The term "preference signals" is used here as the user's actions usually cannot be directly considered as feedback on a specific item. An explicit "like" expression for an artist on a social music platform can, for instance, be used as a positive signal for the artist's musical pieces in a music recommender system. Such types of information are usually exploited by content-based filtering recommender systems, which often rely on these forms of "indirect" preference signals.

2.2.3 User-Action-Related Indirect Preference Signals

Item-independent information like user demographics, the user's *current* location, or the user's embedding in a social network are usually *not* considered to be implicit feedback. Depending on the application scenario, some of these user features *can* however represent indirect preference indicators, i.e., a form of implicit feedback, if the characteristics are the results of user actions that are at least indirectly related with the recommendation targets.

For example, in a restaurant recommender, information about the user's *past* geographic location and movement profile *can* be considered as implicit preference signals in case the movement profile allows to infer a restaurant preference of a specific user without having the user explicitly "checked in" to the restaurant. Also, the user's connections in a social network *can* be considered as implicit preference signals in particular when the goal is to recommend people or groups.

2.2.4 Discussion

As discussed in this section, apart from implicit feedback based on observable user actions, a variety of additional preference signals can be used in the user profiling and recommendation process including in particular the users' demographics or other user characteristics that are independent of an individually recommended item. In the categorization of different feedback types, these signals are usually not considered as implicit feedback. Furthermore, user-independent, additional information about items – including information about item features or to which other items they are connected – is also not considered to be implicit feedback per se, but it might be useful in correctly interpreting implicit feedback signals such as listening or viewing actions. Similarly, contextual information about the users like the location or time when a specific explicit rating was issued, do not fall into this category of implicit feedback, but help to contextualize the collected feedback.

Overall, the distinction between explicit and implicit feedback and other types of information is not always consistent in the research literature and, as discussed, cannot be seen as a Boolean categorization. However, all kinds of feedback that is not explicitly meant to provide an opinion or a relevance assessment shares a set of specific challenges that will be discussed next.

2.3 Challenges of Using Implicit Feedback

When relying on implicit feedback, a number of challenges has to be addressed. This list is far from being complete and an in-depth discussion on the topic can be found in [Jaw+14].

Interpretation of Signal Strength. In many situations, several types of user actions have to be considered in parallel and the question on how to aggregate them turns up. Usually a uniform weighting strategy is not appropriate. For example, in an e-commerce scenario a purchase action might be a stronger preference indicator than a repeated item visit. Apart from labeling different feedback types in advance, these graded signals could also be identified with post-processing techniques. In [Rot+10], for example, communication patterns of users are analyzed to determine the degree of friendship between the users in a social network.

Interpretation in Relation to Explicit Signals. Sometimes, both explicit and implicit feedback signals are available, but with different degrees of coverage of the item space. Therefore suitable ways of combining them are needed. A simple and often employed approach is to interpret all implicit actions as, e.g., "four-star" ratings on a five-star item rating scale and subsequently transform them into explicit rating signals. However, this is often inappropriate as the rating database becomes "dominated" by the large amounts of implicit signals. Also, the implicit feedback "scales", e.g., visit duration, track play counts etc., are also incompatible with the five-point scales used for explicit feedback.

Transparency. When explicit feedback is available, it might be easier for the user to understand the rationale of the provided recommendations as they, e.g., can be used in system-generated explanations more easily. Recommendations that result from implicit feedback signals might not be that obvious or plausible for the user. For example, showing a recommendation to a user with the explanation "because you rated [movie A] with 5 stars" might be more plausible than the explanation "because you watched [movie A]", as in the latter case the user might not have liked movie A after all.

Lack of Negative Signals. Implicit feedback is often "positive-only", i.e., algorithms can only learn positive biases from a user's interaction with an item. This lack of negative signals often means that special types of algorithms (one-class collaborative filtering) have to be applied. This also leads to challenges when applying standard evaluation measures as no ground-truth about non-relevant items is available.

Data Not Missing at Random. In most domains, implicit feedback signals for the few very popular items are prevalent while feedback for niche items can be very sparse [Mar+07]. Therefore, the distribution of feedback is skewed in a long-tail shape. Building recommendation models based on such data can easily lead to a strong popularity bias ("blockbuster effect") and a "starvation" of the niche items.

Abundance of Data. The computation of sophisticated machine learning models can be challenging on large platforms even when only explicit ratings are considered. The amount of data points to be processed, if for example every single navigation action of a user is logged, makes this problem even worse. Furthermore, given the variety of available types of data points, it is not always clear which of the many signals are the most promising ones to retain and consider in the recommendation process.

On the other hand, while implicit feedback signals have some disadvantages when compared to explicit ratings, one advantage of implicit signals is that they can be collected from all users, while (sufficient amounts of) explicit rating information might in many domains only be available from a few "heavy" users. As a result, the models that are learned solely from explicit ratings might overrepresent some user groups.

3 Categories of Observable Implicit Feedback Signals

In this section typical examples of applications from the research literature will be reviewed that use implicit feedback. Then, an extension of the previously discussed categorization scheme for implicit feedback by Oard and Kim [Oar+98] will be proposed. The extension has additional types of user actions which became observable due to technological advancements during the last years.

3.1 Types of Observed Behavior in Applications

Historically, one of the various roots of today's recommender systems lies in the field of Information Filtering, an area that dates back to the 1960s under the term "Selective Dissemination of Information" [Hen63]. The main tasks of information filtering systems are to identify and rank documents within larger collections based on their presumed degree of relevance given the user's search query or profile information. Recommender systems nowadays are used in various applications domains, e.g., e-commerce, media consumption and social networks.

In the following, examples of research works from the recommender systems literature will be given to illustrate the various (new) ways of how user actions and observable behavior can be interpreted and used in different application scenarios. The review of existing works will serve as a basis of the proposal to extend the categorization scheme of [Oar+98] in Section 3.2.

3.1.1 Navigation and Browsing Behavior

Monitoring how users navigate a website or how they use a (web-based) application is a very general type of observable user actions. Several early works that focused on implicit feedback aimed at the *dynamic content adaptation*, e.g., by generating links to possibly additionally relevant content or filtering the available content according to the user's preferences.

Analyzing Dwelling Times for Information Filtering. As mentioned in Chapter 2, interpreting dwelling time as implicit feedback is a challenging task. One of the earlier works in the area of personalized information filtering that tries to rely on the observation of the users' behavior, e.g., dwelling times, to infer their interests is found in [Mor+94]. The authors' specific assumption was that users of their NetNews system will spend more time on interesting items than on non-interesting ones. To verify their hypothesis, they designed a study in which users had to read news articles during a period of several weeks and provide explicit ratings for the articles. The collected data indeed showed that reading times are good indicators for the relevance of an article and that both the length and the readability of an article (typographical denseness of the text) only had a limited impact on reading time. The news filtering systems discussed later in [Kon+97] and [Sak+97] had similar goals and the studies confirm that relying on reading times alone can help to generate accurate recommendations in many situations. Furthermore, as mentioned in [Kob+01], too short viewing times can also be interpreted as *negative* implicit feedback and not only as *missing positive feedback*. The complexity of the interpretation of dwelling time as positive or negative feedback will be further discussed in Section 4.1.3.

Monitoring Navigation Actions. Before the large success of WWW search engines, a number of proposals were made to help users with finding relevant websites based on the observation of their browsing behavior. The "Letizia" system [Lie95] is an early example for an approach that relies on the user's browsing behavior to infer the user's interest. To find additional relevant web pages, the system analyzes the links that a user clicked, initiated searches, or bookmarking activities and applies content-based heuristics. Other early tools that are similar to the basic idea of customizing recommendations based on the user's joint navigation behavior (e.g., link selection) combined with document content similarities are described in [Arm+95] or, with a focus on personalized recommendations, [Mla96].

Browsing Actions. In [Kob+01], a number of additional browsing-based interest indicators besides the following of hyperlinks are mentioned, including micro-level actions like scrolling, highlighting, or the visual enlargements of objects. Depending on the installed equipment on the client side, one can also try to capture the eye gaze of the user [Cas+10] or approximate them by tracking the user's mouse movements [Rod+08]. From a technical perspective, server-side logging of client-side actions can nowadays be implemented very efficiently using AJAX-based micro requests. Further user interface level actions that can occur while browsing include requesting help or explanations for an object.

Web Usage Mining. In contrast to approaches that only rely on navigation or browsing logs of individual users, web usage mining systems aim to detect usage patterns in the logs of a larger user community using, e.g., clustering or association rule mining techniques. Personalization systems like the WebPersonalizer system presented in [Mob+00] try to match the current user's most recent navigation activities with "aggregated profiles" based on clusters of accessed URLs to generate personalized website recommendations.

Discussion. The "Social Information Access" aspect is most obvious in the last category (Web Usage Mining) where the behavior of other users in the community is directly exploited to make suggestions for the current user. Nonetheless, also the other presented techniques, which were partially designed for individual-user settings, can in principle be extended to consider the behavior of the community, e.g., by adding collaborative features within the server-side components.

3.1.2 Shopping Behavior

Implicit feedback signals in e-commerce applications – and also others as mentioned below – could in principle be considered as a subclass of the *navigation and browsing behavior*. However, in a commercial context, specific *semantic meanings* can be attached to some navigation actions such as viewing an item or adding it to a wish list or to the shopping basket, while usually not all navigation actions are considered to be relevant for exploitation.

Shopping Basket Analysis. Amazon.com's "Users who bought ... also bought ..." denotation of one of their recommendation list types characterizes the main idea of such approaches quite well. The general underlying concept is to find patterns in the shopping baskets of users [Lin+03]. Often, these patterns are identified using more general techniques like classic Association Rule Mining [Agr+93] or variations thereof, which can then be applied to generate recommendations for the current user [Lin+02].

Shop Visitor Navigation Logs. Other types of user actions can be employed for building user profiles on shopping sites. For example, another category of recommendations on Amazon.com's site is named "Users who viewed ..." and shows products that were also inspected by users when looked at the current item. One difference to the above-mentioned general approaches based on navigation logs is, as said, that a purchase is a very distinctive action and one of the main business metrics to be optimized. Recent examples of works that aim to exploit the user's recent navigation behavior to predict the next shopping action include [Jan+15a; Ren+10a;

Sha+05; Tav+14] and are often based on approaches that model sequential decision processes.

Discussion. In the past, academic researchers often converted explicit rating datasets into "purchase transactions", e.g., by considering five-star ratings as purchases, because not many public datasets were available. In recent years, an increased rate of works that are based on real-world shop navigation logs can be observed. Academic competitions like the 2015 ACM RecSys Challenge[1] help to fuel these types of research as they are based on publicly available real-world datasets. With the emergence of the Social Web, more and more shopping platforms allow their users to comment, review, and share their experiences on the site, and a variety of other user-related data becomes available for specific tasks like next-basket predictions.

3.1.3 Media Consumption Behavior

Reading news online is, as described above, a classic information filtering scenario in which implicit feedback is prevalent for recommendations. Other types of electronic media consumption in which implicit feedback recommendation systems were employed include the recommendation of (IP) TV programs based on viewing times, video recommendations using the watching behavior, or music recommendation based on listening logs.

Using implicit feedback signals related to media consumption often creates additional challenges. Both for music and TV shows it is not always clear who – maybe even multiple people – in the household is currently watching or listening. In addition, user actions like skipping to the next track can be context dependent and interpreting it as a general negative assessment of the previous track might be misleading.

TV-Related Recommendations. Recommending based on implicit feedback in the context of TV programs was for instance explored in [Gad+07], where the viewing duration, as in [Hu+08], was considered as an indicator for the signal strength and methods were proposed to deal with the uncertainty of the signal. The case of *linear* programs in contrast to video-on-demand services was, e.g., discussed in [Zib+12] where the authors also consider various information signals related to noise in the data and the new-item problem. In the deployed recommendations of the TiVo system [Ali+04], the fact that someone recorded a show is treated as an implicit feedback signal and combined with explicit binary feedback. According to the recent literature review in [Vér+15], implicit profiling is therefore the most common approach in this domain.

[1] `http://recsys.acm.org/recsys15/challenge/`

Music Recommendation. For music recommendation, consider the work presented in [Pal+10; Pal+11] where the authors develop a multi-criteria music recommendation approach which utilizes both explicit as well as implicit feedback. Implicit feedback signals are inferred both for the overall rating of the track as well as for the criteria preferences (i.e., on music, lyrics, and voice). As feedback signals the authors use the total time spent by users listening to a track, the number of accesses to a track, and the actual play duration for each individual listening event of the track.

Another music-related approach is presented in [Lee+10], where the authors, as in [Che+14], rely on listening logs of users obtained from the Last.fm music platform as a basis for music recommendation. A specific aspect of their work is that their algorithms exploit additional (time-related) context information which is derived automatically from logs.

A special form of implicit (and sometimes explicit) feedback for music recommendation are social actions on several media platforms like Last.fm or YouTube, i.e., users can share so-called *playlists* (ordered sequences of music tracks or videos) with others. The particularity of this recommendation problem is that shared playlists represent a form of feedback which is not related to one single item but to the whole recommendation list. The recommendation outcome is usually not a list of items on which the user should find at least one relevant element but rather a list of items which should be sequentially consumed by a user and therefore all relevant [Bon+14].

3.1.4 Social Behavior

With the development of the "Participatory Web", social networks, and Web 2.0 technologies, users transformed from being pure information consumers to also being active content contributors. They can now explore the information space of the Web not only by accessing the structures provided by (classic) information providers, but also by using the behavior or content of other peers in their social networks as guidance. Typical interactions of this "social navigation" are commenting or posting on a social network or microblogging platform, tagging or bookmarking content on the Web, or establishing social connections with other people [Höö+12].

Given these novel types of interactions, a number of additional preference signals can be used in recommendation processes. Some of these types of signals were anticipated in the *Annotation* and *Create* categories of observable behavior in [Oar01] and [Kel+03]. Since the observable user actions on the Social Web are not necessarily directly related to a target object (such as "annotate" or "publish") but can signify also indirect preference indications, “*Social & Public Action*” is introduced here as an additional category.

Tags and Bookmarks. Bookmarking or tagging items with keywords is a classic implicit feedback signal in Information Filtering. In the Social Web sphere, tags and bookmarks are now shared with others and can serve as a basis, e.g., to build tag-based recommender systems [Guy+10; Zha+08; Sen+09; Ged+13].

Posts and Comments. Publishing information on social media in terms of a post or comment about an opinion or the own current activity is another type of implicit preference signal on the Social Web. Such often very short posts can be analyzed to build user profiles that reflect the user’s interests [Abe+11]. The contents of posts was for example analyzed in [Pen+11] by means of a topic modeling technique with the goal to recommend other users on the social network that might be worth following. Finally, the problem of filtering interesting items in a social “feed” often corresponds to a classic collaborative information filtering problem with some additional challenges, e.g., that the content to be analyzed can be very short [Sri+10].

Structuring Objects. The organization of objects for later use is another observable user action mentioned in [Oar01]. A typical example in the recommendation domain is when users share music playlists, i.e., lists of tracks in a fixed order, which can serve as a basis for next-track music recommendation [Bon+14].

Connecting with Others. A final category of implicit feedback signals is the user’s embedding within a social network. One can analyze the user’s social neighborhood, explicit or implicit trust signals, or the network topology [Arm+12] to recommend additional friends or people to follow, or inspect existing group memberships or channel subscriptions and their topics to recommend further groups or other items [Guy+09].[2] In [Lin+13], for instance, the followers of Twitter accounts are used to generate interest profiles to counter the cold-start problem for app recommendations. The work in [Car+09] is an example for the application domain of personalized social search. There, query results are enhanced based on the user’s relation in a social network.

3.1.5 Ubiquitous User Modeling

With the availability of modern smartphone devices and their various sensors as well as the emerging trend of the “Internet of Things”, more and more information

[2] As indicated in Chapter 2, such information is considered in this work only as implicit feedback if the signal is related to some target recommendation object.

about the user’s current location and environment becomes available. To summarize these types of observable user actions under, the umbrella term “*Physical Action*” is proposed in the extended classification scheme.

Location and Movement Profiles. The past and current movement profile can be a valuable indicator of the user’s interests. In [Boh+14], for example, the movement profiles and dwelling times in a museum are interpreted as indicators for the visitor’s interest in the individual exhibit objects. Other application domains in which the past locations of the users can be exploited for user profiling include in particular the tourism domain – think, e.g., of previously visited places or GPS trajectories [Zhe+12] as interest indicators – or leisure activities. For mobile domains, proactive recommendations have been proposed to enhance and expand the mobile experience by “*providing the right information, at the right time, and in the right form for the current context*” [Soh+08]. To avoid being obtrusive, these proactive approaches rely heavily on implicit user signals and have to interpret user behavior carefully [Gil+12]. In [Ler+12], for example, user activity is tracked with a movement profile of GPS locations. Based in this, a proactive recommendation model is proposed that automatically determines idle phases where it might be appropriate to notify the user with suggestions.

Note that in contrast to context-aware recommendations (CARS) this work does not necessarily focus on the user’s *current* location to make suitable recommendations, but rather relies on the observed user behavior and relationships between past user actions to determine the appropriate next steps.

Smart Homes. In the Internet of Things, all sorts of electronic devices, e.g., in a smart home, will be connected with the network and can represent additional sources of information about the environment of a user or with which devices the user has interacted. One typical task in such a context is called “activity recognition”, i.e., to estimate based on the available sensor data, e.g., from a mobile phone [DP+14; Par+09], which activities the users currently pursue and where they are located. Quite an amount of research on knowledge-based or learning-based activity recognition has been conducted in the area of *smart homes*, see, e.g., [Tap+04] for an early work. While these types of sensor information have been largely ignored in the mainstream recommender systems literature, adapting light or music in smart homes based on the user preferences [Kha+09] are prime examples for personalized systems that often use implicit signals. Some more recent examples include the automatic identification of users while watching TV in order to learn their interests [Lin+14] or the use of gaze tracking in combination with explicit ratings to derive content-based interest profiles [Kli+14].

3.2 An Extended Categorization of Observable User Behavior

Oard and Kim's early categorization scheme – *Examination*, *Retention*, and *Reference* – mainly focused on document-centric applications and is in particular suitable when the goal is to recommend news messages, text documents, or web pages (see Table 3.1). This also holds for the additions in [Oar01] and [Jaw+14]: *Annotation* and *Create*. With the wide-spread application of recommendation technology in all sorts of domains that can be observed in the last two decades, a variety of other types of implicit feedback signals have been successfully exploited since Oard's and Kim's early work.

Based on the review of application scenarios for implicit feedback in recommender systems, the existing classification scheme should be extended with additional observable user actions. They are related to (a) the user's social behavior and (b) the increased availability of data for "ubiquitous" user modeling. The new items are shown in Table 3.1 with a detailed description. They meet the requirements of new behavioral patterns that emerged with the widespread availability of connected mobile devices and social functionalities on the Web. Keep in mind that in practice the types of observable behavior can overlap. For example, the *Social & Public Actions* "posting" and "rating" articles on a social network can also be seen as *Create* and *Annotation* actions.

Tab. 3.1: Extension of the five types of observable behavior (see Table 2.1 and [Oar01; Oar+98; Kel+03]) by two now categories: *Social & Public Action* and *Physical Action*.

Category	Examples of Observable Behavior
Examination	Duration of viewing time, repeated consumption, selection of text parts, dwell time at specific locations in a document, purchase, or subscription.
Retention	Preparation for future use by bookmarking or saving a named or annotated reference, printing, deleting.
Reference	Establishing a link between objects. Forwarding a document and replying, creating hyperlinks between documents, referencing documents.
Annotation	Annotate, rate or publish an object (includes explicit feedback).
Create	Write or edit a document.
Social & Public Action	Public posting, commenting and communicating, activity posts, following and connecting with people, joining groups, expressing trust.
Physical Action	Observed user actions that can be interpreted as feedback towards objects of the physical world. Being at a location, roaming profiles and dwelling time, other recognizable activities in the physical world (e.g., smart homes, Internet of Things).

4 Algorithms for Implicit Feedback Situations

In this section, algorithmic approaches to generate recommendations based on implicit feedback will be discussed. As mentioned earlier, interpreting implicit feedback can be difficult. First, techniques to transform and encode preference signals as explicit feedback to be able to use standardized recommender system algorithms for rating prediction will be examined. After that, selected examples of collaborative filtering algorithms that are especially designed to deal with "one-class" only feedback signals will be briefly presented. Then, methods to find frequent patters in implicit feedback will be examined in more detail and examples of hybrid algorithms that try to combine explicit ratings with implicit feedback will be shown. Finally, recommendation techniques for activity logs in e-commerce will be discussed and application-specific requirements and variants of the BPR algorithm will be considered in context of the included papers of this thesis [Ler+14; Jan+15a; Jan+15f].

4.1 Converting Implicit Signals to Ratings

As discussed in Section 2.3, implicit feedback is often "positive-only", i.e., no or only minimal information is given about items that are disliked by the users. Also, there are often multiple signals and different kinds of feedback, e.g., when a user visits an item detail page in an online store multiple times, bought some items, and placed other items on a "wish list". In addition, the "rating matrix" of implicit feedback is very sparse most of the time. Therefore, the available data consists of only a few positive signals that are sometimes hard to interpret and numerous unlabeled examples.

To deal with such situations, many so-called "One-Class Collaborative Filtering" techniques were proposed in the literature, some of which are discussed later in Section 4.2. In this section, an alternative to this approach is shown, which is the transformation of the given data into two-class or multilevel numerical "ratings" and the creation of a user-item rating matrix. Such a transformation allows to apply standard recommendation techniques which were originally designed for

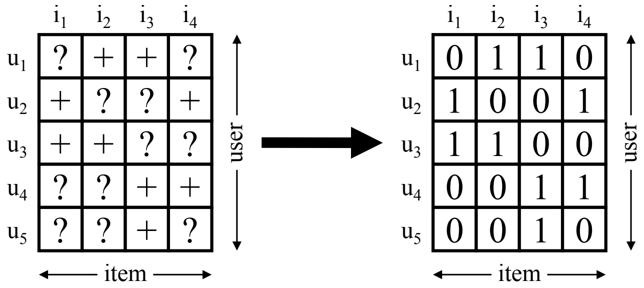

Fig. 4.1: AMAN: Transformation of implicit feedback to explicit “0” and “1” ratings [Pan+08; Ren+09].

explicit ratings. In the following, ways of transforming implicit feedback signals into numerical rating values are discussed.

4.1.1 Problems of Basic Transformation Strategies

The first step of a basic implicit-to-numerical transformation is to add a virtual rating of “1” to the user-item rating matrix for each observed user-item interaction. Different options exist to deal with the unknown data points and the missing negative feedback, each of them having certain drawbacks as discussed in [Pan+08].

All Missing as Negative (AMAN). In the AMAN approach, all non-observed items are treated as a “0” rating and thus as negative feedback. The resulting user-item-matrix contains only ones and zeros, see Figure 4.1. When a machine learning model is fitted to this data, the distribution between the two classes – 0 and 1 – is strongly biased toward the negative feedback, since there are only few positive entries in the matrix. Any rating prediction technique for explicit feedback might tend to always predict 0. Usually, regularization methods are used to prevent this kind of overfitting but the ratio of negative to positive feedback in the data is still problematic [Ren+09].

All Missing as Unknown (AMAU). In the AMAU approach, the missing data points are treated as unknowns, see Figure 4.2. Therefore, a rating prediction algorithm only operates on the positive ratings, i.e., ignores all missing data. Since the whole dataset only consists of “1” ratings, a basic rating prediction algorithm would always come to a trivial solution and predict “1”. Even with proper regularization based on the unknown data points, typical explicit feedback algorithms can tend to always predict 1 [Sre+03].

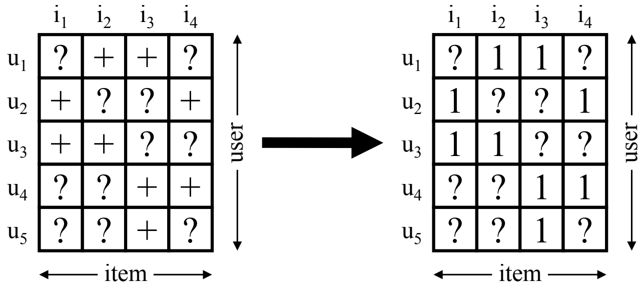

Fig. 4.2: AMAU: Transformation of implicit feedback to explicit unknown and "1" ratings [Pan+08; Ren+09].

4.1.2 Discerning Negative from Unknown Signals

To avoid the drawbacks of the *extreme* approaches to deal with unknown examples – labeling them as negative or ignoring them – more advanced approaches assume that there might be *some* negative examples in the unknown data. If these could be labeled properly, existing explicit feedback approaches could be employed.

Several ways to guess which of the unknown entries could be negative feedback have been introduced in the past, some of them as part of a one-class collaborative filtering techniques, which will be discussed in Section 4.2.

A simple approach is to randomly sample negative examples from the unknowns, as done, e.g., in [Ren+09]. To learn the ranking of items, their approach uses positive-negative item-pairs for each user. However, since there is no negative feedback in the data, they select a random unknown item for each (positive) feedback to create the pairs. More elaborate schemes use statistical [Paq+13] or weighting-based [Pan+08] approaches to choose negative samples in a way that the distribution of the resulting set of negative ratings resembles the set of the positive ratings.

As an alternative to inferring negative ratings, users could be asked to give some negative (and positive) feedback, e.g., in an initial interaction phase with the system. However, this would be considered explicit feedback and might be perceived as a burden by users [Pan+08; Par+11a].

4.1.3 Converting Graded Implicit Feedback to Ratings

Instead of converting implicit feedback signals into explicit 0/1 ratings, some proposals in the literature adopt more fine-grained strategies. Since in many application settings different types of user behavior can be observed, the idea is to assign a different "strength" to each type of signal, i.e., to encode the different levels of *graded* relevance feedback as ratings.

In a study on recommendations in an online mobile games store [Jan+09] the authors used explicit ratings – which were only sparsely available – and in addition considered view and purchase actions, which were transformed into explicit rating values. On a scale from -2 to $+2$, view events were interpreted as 0 and purchase events as $+1$. Explicit positive (negative) ratings where considered as a $+2$ (-2) rating. The choice of this encoding was done somewhat arbitrarily and led to a very skewed distribution of the rating values, as there are many more view events than purchase events.

A time-based approach of assigning numerical values to implicit feedback signals was proposed by Lee et al. [Lee+08] in the context of a recommender system for wallpaper images. Purchase information was used as an implicit signal and the strength of the signal was determined based on the release date of an item and the point in time when the user made the purchase. The authors then used a time-based decay function to promote more recent events that received higher scores.

An approach of combining different feedback types was presented by Parra and Amatriain [Par+11b] in the context of music recommendation. The authors propose to use a linear regression model to combine three different aspects of implicit feedback signals – personal feedback, global feedback, and recentness – into a rating score. They conclude that the former two interaction types have the strongest impact on the recommendation accuracy. In [Par+11a], this model is extended to a logistic regression model that includes a number of additional variables related to consumption behavior as well as demographic data.

In another work in the field of music recommendation [Kor+10], items that have both explicit ratings and observed user actions are exploited to learn which types of implicit feedback can be mapped onto which ratings. The user actions that were interpreted as implicit feedback consist of play counts and play percentages, listening date and time, number of skips, and next-track statistics. Subsequently, the system rates items that did not receive any explicit feedback with a naive Bayesian classification based on the implicit signals that the item received.

The examples above show that transforming signals of different types of user behavior into one single (rating) score largely depends on the respective domain and cannot be generalized easily. Sometimes, it may not be possible to map different kinds of feedback, e.g., viewing and buying an item, to a linear rating scale. Other signals may be difficult to interpret. For example, a short dwelling time for an item detail page could be interpreted as negative feedback because the user seems to be not interested in the item. However, it could also mean that the user already knows the item and does not need to look at the page again. On the other hand, a long dwelling time does not necessarily correspond to positive feedback. A user might have lost interest and abandoned the page without actively leaving it because the item was not relevant anymore. In the following section, the correlation between implicit signals and explicit ratings will therefore be discussed in more detail.

4.1.4 Correlating Implicit Feedback with Explicit Ratings

The question of how to encode different types of feedback into numerical scores can, as discussed, be challenging and is most of the time done in an arbitrary manner. Several researchers have therefore investigated the relationship between explicit ratings and implicit feedback actions, including [Cla+01; Par+11b; Par+11a; Piz+10; Sha+06]. Depending on the domain and experimental setup, the obtained results are however not always consistent.

In [Cla+01], the results of a laboratory study are reported in which users were first asked to freely browse the Web for 30 minutes and subsequently had to rate each visited page with respect to how interesting its contents were. The recorded user actions, such as mouse movements and dwelling times, were then compared with the collected explicit ratings. The analysis revealed that the time spent and the scrolling activity on a web page correlates with explicit ratings. Other indicators, however, such as mouse movement and clicks, had no clear relation to the participant's interest.

Zhang and Callan [Zha+05] report the results of a user study on a web-based news filtering system. The participants had to read personalized news for one hour per day over a period of 4 weeks and assess the articles according to multiple dimensions, such as relevance, novelty, and readability. After the study, each participant also completed a questionnaire about the topics of the articles that they read. In addition, the same user actions as in [Cla+01] were recorded and the authors similarly concluded that dwelling time and scrolling activity correlate the most with the explicit ratings. However, they also state that the answers of the final questionnaire about topic interests are much more correlated to the explicit feedback on the news articles recorded before than to the implicit signals. They therefore advise that in real-world settings the users should initially be asked about their topics of interest.

Building on the insights and log data from this study and the work from [Cla+01], the authors of [Zig+06] propose a Bayesian modeling technique to combine implicit and explicit feedback signals. Their results, however, indicate that compared to the explicit feedback the implicit feedback possessed only limited predictive value. Thus, the combination of both feedback types was only marginally better than when using explicit feedback alone.

The results of a similar study on the relationship between various types of browsing actions and explicit interest statements are reported in [Sha+06]. The strongest correlations with the explicit ratings were found for the indicator "time of mouse movement relative to reading time" (time spent on the page) and the number of visited links on the page. Note that mouse movements were not considered to be a good indicator according to the study [Cla+01] discussed above. However, the authors did not put the mouse movements in relation to the dwelling time.

More recently, Parra et al. [Par+11b; Par+11a] report on their attempt to correlate implicit and explicit preferences in the music domain. The authors first carried out a survey in which the participants rated tracks from Last.fm. This information was used to derive preference patterns and biases, e.g., whether users generally prefer recent or popular tracks. The insights of the survey were then used to design a linear regression model to predict ratings from *what they call* implicit feedback signals. In fact, the authors rather adopt an approach based on meta-data to learn which features of the liked items are particularly relevant to the users.

Finally, in some domains implicit feedback seems to be more meaningful than explicit preference information. In [Piz+10], Pizzato et al. use the data of 21.000 users of an online dating platform and compare the predictive accuracy of different input types. In contrast to most of the other works reviewed so far, their results show that explicit preference statements are often incomplete or imprecise and recommending based on implicit feedback can be more accurate. This emphasizes once more that the interpretation of implicit feedback can be highly dependent on the respective domain.

4.2 One-Class Collaborative Filtering Techniques

The naive conversion strategies to generate (binary) numerical scores from implicit feedback have their drawbacks, e.g., converting all unobserved data points into zeros (*All Missing as Negative*, AMAN) or leaving them as unknowns (*All Missing as Unknown*, AMAU) both result in a class imbalance problems and standard rating prediction techniques tend to always predict 0 or 1, respectively. Therefore, more sophisticated techniques were proposed to deal with positive-only feedback in the literature.

These so-called *one-class collaborative filtering* (OCCF) techniques [Pan+08] are algorithms that only need one single type of signals. They usually interpolate which of the missing data points could be negative feedback or try to guess if a user prefers one item over a different, unknown one. Typically, OCCF techniques are used in domains where only unary implicit feedback is available. As discussed earlier in

Section 2.1, some of them are also applicable when dealing with unary explicit feedback, such as “likes” on a social network. In the following, examples of selected OCCF techniques will be presented in more detail – wALS, Random Graphs, BPR, CLiMF – and briefly review other related approaches.

4.2.1 Weighted and Sampling Alternating Least Squares

In [Pan+08], the authors introduce the *Weighted Alternating Least Squares* (wALS) and *Sampling Alternating Least Squares Ensemble* (sALS-ENS) algorithms to handle the missing feedback in a way that is somewhere in between the two extremes of AMAU and AMAN. In the first strategy, a low-rank approximation $\vec{X}$ of the “rating” matrix R is calculated and in the objective function, a confidence weight is used to express the probability that a signal is (correctly) interpreted as positive or negative. A weight of 1 is assigned to the positive data points, since they are known beforehand. The unknown, missing values, on the other hand, have a confidence value lower than 1, because some of them have the chance to be negative samples. The following equation shows the objective function.

$$\mathcal{L}(\vec{U}, \vec{V}) = \sum_{ij} W_{ij}(R_{ij} - \vec{U}_i \vec{V}_j^T)^2 + \lambda(||\vec{U}||_F^2 + ||\vec{V}||_F^2) \quad (4.1)$$

The low-rank approximation $\vec{X}$ of R is decomposed to $\vec{X} = \vec{U}\vec{V}^T$ and can be used for prediction. To prevent overfitting, the objective function features a regularization term. The matrix W is the non-negative weight matrix that assigns confidence values to the observations and the optimization problem is solved by an Alternating Least Squares (ALS) algorithm.

The characteristics of this algorithm are influenced by the choice of W. For $W = 1$, the confidence for all data points would be 1 and therefore the strategy would be equivalent to AMAN, where all unknown are treated as negatives. The authors propose three different weighting schemes W for the unknown entries W_{ij} in the user-item interaction matrix, which are described in Table 4.1.

Calculating a large approximative low-rank matrix is however computationally intensive and, in addition, the class imbalance problem is still present, because there are still many more negative than positive samples.

Therefore, with sALS-ENS the authors propose a more advanced way to consider all (known) positive examples from the data and add a subsample of negative feedback based on a sampling probability matrix. They propose three sampling strategies that behave similar to the ones used for the weighting matrix of wALS. As a result, a smaller rating matrix is generated that can be used as a basis for calculating the low-rank approximation of R via ALS. The experiments show that the wALS approach is

Tab. 4.1: Weighting schemes for OCCF.

Weighting scheme	Confidence matrix	Description
Uniform	$W_{ij} = \delta$	All missing entries are assigned a fixed confidence weight δ.
User-Oriented	$W_{ij} \propto \sum_j R_{ij}$	Higher confidence is set for "heavy" users as they are assumed to know the item catalog better and therefore discard unknown items with a higher probability.
Item-Oriented	$W_{ij} \propto m - \sum_i R_{ij}$	Higher confidence is set for items that received more interactions, i.e., unpopular items have a higher probability to be discarded as negative.

slightly superior in terms of accuracy but considerably slower than sALS-ENS. An approach similar to wALS has been proposed in [Hu+08] that also uses a weighting term for the implicit observations.

4.2.2 Random Graphs

In a similar spirit, Paquet and Koenigstein [Paq+13] model the unknown negative feedback using a random graph. The approach is based on a bipartite graph G that contains edges $g_{mn} = 1$ between users m and items n when observed implicit feedback is available.

The additional assumption is however that, although a user m has accepted (interacted with, clicked, purchased) an item n, there should be some other (hidden) items that the user considered but discarded as not relevant. The authors therefore model a second *hidden* graph H that is also bipartite and contains edges $h_{mn} = 1$ whenever a user n considered an item m. In addition, $g_{mn} = 1 \Rightarrow h_{mn} = 1$ holds, i.e., if a user m accepted an item n, then it was considered before.

Therefore, G is a subgraph of H and all the other edges of H are the considered items that were discarded as not relevant, i.e., the negative feedback. Since the negative feedback is unobserved, the authors use the following popularity-based sampling strategy to generate the edges in H that represent the negative feedback.

For each user m with d_m observations of positive feedback, additional d_m edges of negative feedback are randomly sampled from a distribution $M(\pi)$ based on the popularity of all items. Instead of using a popularity distribution with $\pi_n = d_n$, where d_n is the number of times that there was positive feedback for an item n, the authors assume that popular items are generally more liked, i.e., have less

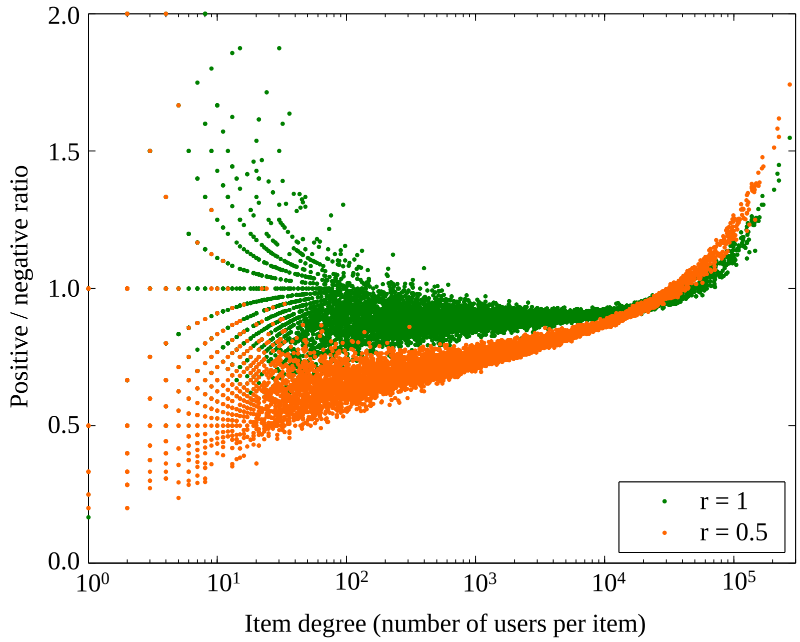

Fig. 4.3: Ratio of positive and negative edges in H [Paq+13].

negative feedback. Therefore, for the popular items, less negative examples should be sampled for H and the distribution is modified in the following way: $\pi_n = d_n^{\gamma}$ with $\gamma = 1 - \log d_{max} / \log r$.

The parameter r controls the ratio between sampled negative and known positive samples in H, as can be seen in Figure 4.3. In addition, the sampling procedure is done "without replacement", i.e., if the sampling draws an already known positive example $h_{mn} = 1 \wedge g_{mn} = 1$, no negative sample $h_{mn} = 0$ is added. Therefore, the ratio is skewed for the most popular items, as there is a higher chance to draw a positive example for them, which results in a lower amount of generated negative feedback signals.

An advantage of this approach is that it is easily extensible with richer feedback signals. For example, the hidden graph H could also be populated with implicit negative examples gathered from other sources, e.g., a visited detail page of an item without a subsequent purchase or a purchase of an equivalent item could indicate $h_{mn} = 1 \wedge g_{mn} = 0$. Similarly, information about items that could have never been considered ($h_{mn} = 0$) could be included in the graph, e.g., because the item was not listed in the shop at the time the user was active.

To generate recommendations with the two graphs G and H the authors propose a bilinear collaborative filtering model with matrix factorization which can estimate the probability of accepting an item as relevant after considering it under $p(g_{mn} = 1 | h_{mn} = 1)$. The model is designed to be largely agnostic of the popularity of the items. Therefore, the popularity bias of the recommendations can be reduced. More

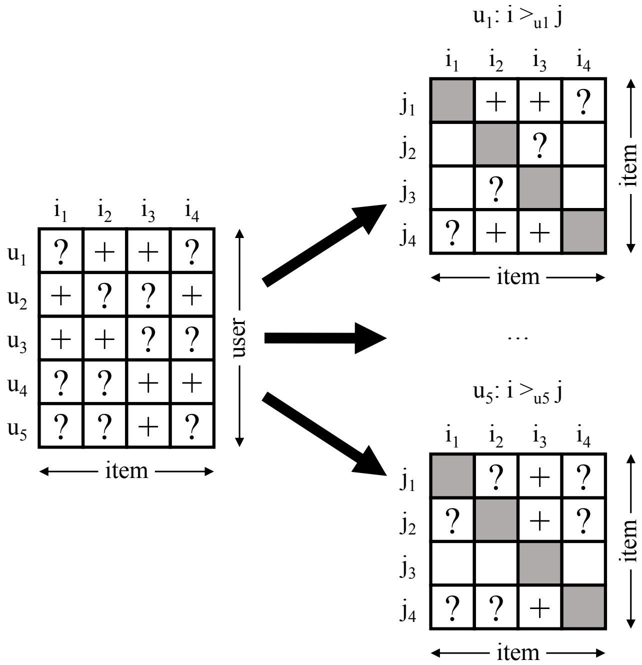

Fig. 4.4: Transformation of implicit feedback to pairwise preferences for each user [Ren+09].

details on the model have been discussed in [Paq+13]. The proposed approach was developed and deployed in the context of the Microsoft Xbox Live environment, in which one particular challenge lies in the large amounts of data that have to be processed.

4.2.3 Bayesian Personalized Ranking

Bayesian Personalized Ranking (BPR) [Ren+09] deals with the one-class CF problem by turning it into a ranking task and implicitly assuming that users prefer items they have interacted with over other unknown items.

In some sense, BPR therefore creates artificial negative feedback in a similar spirit as the approaches discussed so far. However, instead of applying rating-prediction techniques using the implicit feedback data, BPR ranks the all candidate items for each user.

The overall goal of the algorithm is to find a personalized total ranking $>_u \subset I^2$ for all users $u \in U$ and pairs of items $(i, j) \in I^2$ that has to satisfy the properties of a total order (totality, antisymmetry, transitivity).

To model the negative feedback, Rendle et al. [Ren+09] use a pair-wise interpretation of the positive-only feedback. The general idea is that a user's positive feedback for an item is interpreted as the user's preference of this item over all other items that the user did not give feedback for. As shown in Figure 4.4, the positive-only feedback is thus transformed to positive and negative feedback in pairs of items (i, j) where the user preferred i over j (positive), or j over i (negative). If the user interacted with both items or none of them, no additional information can be deduced for the pair (i, j). The different pairs of items form the training data D_S for the BPR algorithm and can be formalized as triples of a user and an item pair:

$$
\begin{aligned}
D_S &:= \{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\} \\
&\text{with} \\
I_u^+ &: \text{items with implicit feedback from } u
\end{aligned}
\tag{4.2}
$$

To create a personalized ranking of items, the authors introduce a general optimization criterion called BPR-OPT, which is derived through a Bayesian analysis of the problem and which aims to maximize the posterior probability $p(\Theta| >_u) \propto p(>_u |\Theta)p(\Theta)$, where Θ is the parameter vector of the underlying algorithmic model. The optimization criterion, including substitutions for smoothing, is formulated as follows:

$$
\begin{aligned}
\text{BPR-OPT} &:= \ln p(\Theta| >_u) \\
&= \ln p(>_u |\Theta)p(\Theta) \\
&= \sum_{(u,i,j)\in D_S} \ln \sigma(\hat{x}_{uij}) - \lambda_\Theta ||\Theta||^2 \\
&\text{with} \\
\sigma(x) &= 1/(1 + e^{-x}) \\
\hat{x}_{uij} &:= \hat{x}_{ui} - \hat{x}_{uj} : \text{a model-specific relationship function} \\
\lambda_\Theta &: \text{model specific parameters}
\end{aligned}
\tag{4.3}
$$

The BPR-OPT criterion is related to the AUC metric and optimizes it indirectly. To solve the optimization problem, a gradient descent on the model parameters Θ can be used. Since it is computationally expensive to take all triples $(u, i, j) \in D_S$ into account, a stochastic gradient descent approach randomly chooses the triples uniformly from D_S.

By decomposing the model specific function $\hat{x}_{uij}$ – which is a real-valued function for the relationship of the items i and j for user u – into $\hat{x}_{ui}$ and $\hat{x}_{uj}$, existing techniques for rating prediction can be applied to calculate the two terms. In [Ren+09], both a matrix factorization model and a kNN approach are presented as the underlying model for the BPR algorithm. Compared to stand-alone Matrix Factorization (MF) or kNN models, which minimize the rating prediction error, the BPR-OPT criterion instead ensures that the item ranking is optimized.

4.2.4 Collaborative Less-is-More Filtering

Collaborative Less-is-More Filtering (CLiMF) is another approach for ranking optimization in one-class CF settings [Shi+12]. CLiMF aims to directly optimize a smoothed version of the Mean Reciprocal Rank (MRR) metric to achieve an optimal ranking of the top-n items. In comparison to BPR, both algorithm optimize a smoothed version of a ranking metric. BPR, however, implicitly assumes negative feedback in the data, while CLiMF only uses the positive feedback signals.

The optimization target of CLiMF, the reciprocal rank RR_i of a recommendation list for a user i, represents the position of the earliest occurrence of a relevant item for the user. For N items it can be defined as:

$$\begin{aligned} RR_i &= \sum_{j=1}^{N} \frac{Y_{ij}}{R_{ij}} \prod_{k=1}^{N} (1 - Y_{ik}\mathbb{I}(R_{ik} < R_{ij})) \\ &\text{with} \\ Y_{ij} &= 1 \text{ if } i \text{ interacted with } j\text{, else } 0 \\ R_{ij} &= \text{rank of item } j \text{ in list of user } i \\ \mathbb{I}(x) &= 1 \text{ if } x = true\text{, else } 0 \end{aligned} \tag{4.4}$$

In essence, the formula only calculates $1/R_{ij}$ for the first relevant item j for user i. However, directly optimizing the reciprocal rank with standard optimization functions – like gradient descent – is not possible, since it is a non-smooth function. Therefore, the authors introduce a smoothed approximation of RR_i which can be optimized. To that end, the indicator function $\mathbb{I}(x)$ and the rank $1/R_{ij}$ are substituted by the following approximations:

$$\begin{aligned} \mathbb{I}(R_{ik} < R_{ij}) &\approx g(f_{ik} - f_{ij}) \\ 1/R_{ij} &\approx g(f_{ij}) \\ &\text{with} \\ g(x) &= 1/(1 + e^{-x}) \\ f_{ij} &= \langle U_i, V_j \rangle \end{aligned} \tag{4.5}$$

Here, the predictor function for the relevance score f_{ij} is based on a factor model of the latent user and item factor vectors U_i and V_j. Although inserting the substitutions of Equation 4.5 in Equation 4.4 creates a smooth approximation of the reciprocal rank and could in theory be optimized, the optimization task has a complexity of $O(N^2)$, i.e., is quadratic with the number of items, which is not practically feasible in most domains. It is, however, possible to derive a lower bound of the reciprocal rank which can be optimized with a lower complexity [Shi+12]:

$$L(U_i, V) = \sum_{j=1}^{N} Y_{ij}[\ln g(f_{ij}) + \sum_{k=1}^{N} \ln(1 - Y_{ik}g(f_{ik} - f_{ij}))] \tag{4.6}$$

The optimization function of CLiMF (Equation 4.6) has two terms that are maximized: (1) Y_{ij} and (2) the rest of the equation in square brackets. Maximizing the first term promotes the relevant items. Maximizing the second term optimizes the ranking by learning latent factors. As discussed in [Wan+10], this can also lead to a diversification of the recommendation results. Equation 4.7 shows the final regularized optimization function for all users. It can be optimized with a stochastic gradient descent approach and a complexity of $O(dS)$ with S being the number of observed positive feedback examples and d being a scalar.

$$\begin{aligned} F(U, V) = &\sum_{i=1}^{M}\sum_{j=1}^{N} Y_{ij}[\ln g(U_i^T V_j) + \sum_{k=1}^{N} \ln(1 - Y_{ik}g(U_i^T V_k - U_i^T V_j))] \\ &- \frac{\lambda}{2}(||U||^2 + ||V||^2) \end{aligned} \tag{4.7}$$

Later on, the authors proposed a generalized version of CLiMF called xCLiMF which is able to deal with situations where a relevance level for the feedback is available [Shi+13]. A similar generalization to deal with graded relevance feedback was also proposed for BPR in [Ler+14], which will be discussed later in Section 4.6.2.

4.2.5 Other One-Class Approaches

The problem of positive-only data can also be found in the field of classification when there are only positively labeled training data. Support Vector Machines (SVM) are a typical method that was originally designed for two-class classification tasks and requires labeled input data. In [Sch+01b], Schölkopf et al. develop a theoretical foundation to apply support vector machines (SVM) to unlabeled, one-class data.

These one-class SVM (1-SVM) are able to identify a region in the item-space where most of the "positive" examples are located. Likewise, the other regions of the item-space can be labeled as "negative". A practical implementation of 1-SVM for recommender systems and a benchmark on the MovieLens data is presented in [Yaj06]. Similar examples for classification tasks without the need for negative training samples are [Yu+02] and [Ke+12], where the goal is to classify websites and text based on positive and unlabeled data only.

A density estimation that is similar to the one by Schölkopf et al. [Sch+01b] is presented in [BD+97]. The authors introduce a model to estimate high-density areas of the data points in the item-space. Additionally, the model assumes that not all the data points are positive feedback but could also be negative examples. If there is positive feedback as well as unlabeled examples in the data, it is possible to solve the one-class classification problem by applying the expectation-maximization (EM) algorithm (see, e.g., [War+09] and [Den98]). In [Li+05], finally, unlabeled data points are treated as negative examples. This transforms the task into a problem of "learning with noise" and is solved with a regression approach to model a linear function as a classifier.

4.3 Frequent Patterns in Implicit Feedback

One of the most prominent examples of a recommendation system is Amazon.com's list of shopping proposals labeled "Customers who bought ... also bought ...". The label suggests that the contents of the non-personalized but item-dependent list are based on the analysis of the buying behavior of Amazon's customer base and the detection of item co-occurrence patterns.

In these classic *Shopping Basket Analysis* settings, the goal is to find sets of items that are frequently purchased together. The input to the analysis is a set of purchase *transactions* where each transaction contains a set of items that were bought together, e.g., in one shopping session.

4.3.1 Association Rule Mining

Technically, the identification of such patterns can be accomplished with the help of *Association Rule Mining* (ARM) techniques [Agr+93]. An association rule has the form $A \Rightarrow B$, where A and B are sets of items and the arrow can express something like "whenever A was purchased, also B was purchased". Typically, the strength of a rule is usually expressed with the help of the measures *support* and *confidence* defined later on.

Following the description of [Sar+00], *Association Rule Mining* can be formally defined as follows. Let $T = \{t_1, \ldots, t_m\}$ be the set of all transactions and $I = \{i_1, \ldots, i_n\}$ the set of all available items. Each transaction t consists of a subset of items $t \subseteq I$. A transaction t could therefore represent a shopping basket of items that were bought by a customer. Let $A, B \subseteq I$ and $A \cap B = \emptyset$, i.e., A and B are also subsets of I but have no items in common. An *association rule* is defined as the implication $A \Rightarrow B$. It expresses that whenever the items contained in A are included in the transaction t, then items contained in B will also be included in t. The left side of a rule $A \Rightarrow B$ is often called the *rule body* or *antecedent* while the right side is the *rule head* or *consequent*.

As can be seen from the definition above, each co-occurrence of two or more items in a transaction can be expressed as an association rule. However, not all association rules are helpful, e.g., two items could only have occurred together once in a single transaction, and the goal of *Association Rule Mining* is to find only those rules in a set of transactions T that are meaningful. To quantify the significance of association rules, various measures have been introduced in the past[1] but the most widely-used measures are *support* and *confidence*.

The *support* of a set of items A is the proportion of transactions $t \in T$ that contains A, i.e., the transactions where $A \subseteq t$. It can also interpreted as the probability of the co-occurrence of all items in A in a transaction.

$$supp(A) = \frac{|\{t \in T; A \subseteq t\}|}{|T|} \tag{4.8}$$

The *confidence* of an association rule is then defined as the ratio between the number of transactions that contain $A \cup B$ in relation to the number of transactions that only contain A. Therefore, the *confidence* is the conditional probability of B given A. It can be expressed with the support of A and $A \cup B$.

$$conf(A \Rightarrow B) = \frac{supp(A \cup B)}{supp(A)} \tag{4.9}$$

Usually a minimum support and a minimum confidence have to be satisfied by an association rule to be considered meaningful. Therefore, when generating the association rules, first a threshold for the support is applied to find the most frequent sets of items in the transactions. However, when there are n items in the set of all items I, the number of possible subsets that have to be considered is $2^n - 1$, excluding the empty set. For a large number of items n, considering all combinations individually is infeasible. An efficient calculation of the support is however possible by exploiting the *downward-closure property* [Agr+93]: If an itemset A is frequent according to some support threshold, then all of its subsets $A' \subseteq A$ are also frequent for that threshold. Likewise, if an itemset A is not frequent according to some support threshold, then all of its supersets $A' \supseteq A$ are also not frequent for that threshold. After the most frequent itemsets have been found, a confidence threshold is used to determine the most important association rules.

To detect these rules automatically and for large amounts of data, a variety of algorithms was developed over the last decades to find frequent patterns and derive association rules. Starting with the Apriori algorithm [Agr+93] that uses the *downward-closure property*, more efficient schemes like FP-Growth [Han+00] and techniques that find patterns in parallel [Li+08] or are able to identify rules for

[1] http://michael.hahsler.net/research/association_rules/measures.html

niche items [Lin+02] were proposed. For cases in which the sequence of the item interactions is relevant, *Sequential Pattern Mining* [Agr+95] can be applied.

4.3.2 Recommending with Association Rules

Once the rules are determined, recommendations based on association rules can be created, e.g., in the e-commerce domain, as follows. First, the set of the current user's (recently) purchased or viewed items is determined and then rules that contain these items in the antecedent (A) are detected. The elements appearing in the consequent (B) of the corresponding rules then form the set of possible recommendations. Items considered for recommendation can then be ranked with a $score_{ui}$ based on different heuristics, e.g., by using the confidence of the rules that are applicable to the subsets A of the items I_u that a user u purchased and that lead to the inclusion of an individual item i.

$$score_{ui} = \sum_{A \subseteq I_u} conf(A \Rightarrow i) \tag{4.10}$$

A specific aspect to consider in the recommendation domain is that one is not necessarily interested in the strongest rules, i.e., the rules with the highest confidence, as they might lead to obvious recommendations, but rather in rules for unexpected patterns or niche items. Also, depending on the domain, different kinds of association rules can be mined. In [Lin+02], for example, the score used to rank the items was calculated using both user associations ("user u_1 likes an item" $\Rightarrow$ "user u_2 likes an item") and items associations ("item i_1 is liked" $\Rightarrow$ "item i_2 is liked"). When recommending, their approach ranks the items by a $score^{User}_{ui}$ for users that are above a fixed support threshold, i.e., users that have already left some feedback in the system. The score is calculated both based on the support and confidence of the user associations. However, A is now a subset of the users U_i that liked the item i.

$$score^{User}_{ui} = \sum_{A \subseteq U_i} supp(A \cup u) \cdot conf(A \Rightarrow u) \tag{4.11}$$

For users below the support threshold, i.e., cold-start users that only left little feedback, item association rules are used to calculate the item ranking similar to Equation 4.10. To find niche items, the item association rules are however mined for each item as a fixed *consequent* and only the subset of transactions T' that contains the *consequent* is used to calculate the support of each item. Since T' is usually small compared to T, the resulting support for the items is higher. Otherwise, rules for new or niche items would be filtered out, since their support would often be below the general support threshold over all transactions T.

In a similar way, Amazon.com's "Customers who bought ..." recommendations can be generated with Association Rule Mining. The particularity of such an approach is that only frequent itemsets of size two are required – which means that simple co-occurrence counts can be sufficient – and that these recommendations can already be provided in the context of an item when the customer views the item's detail page for the first time.

In the literature, Association Rule Mining techniques have been applied in different recommendation scenarios. Examples include the identification of navigation patterns in the context of Web Usage Mining [Mob+00] or e-learning [Gar+09], the detection of rules to exploit item characteristics in e-commerce [Hua+04; Pit+11], or the recommendation of next tracks in music playlist generation [Bon+14; Har+12]. Furthermore, Association Rules and "co-visitation counts" also serve as a basis for the YouTube video recommendation system [Dav+10].

4.4 Hybrid Implicit-Explicit Techniques

In some domains both explicit and implicit feedback signals are available. For example, in most online stores users can rate products and at the same time their navigation behavior is logged by the system. In the following sections, some approaches will be discussed that combine explicit and implicit feedback or use the explicit rating of an item as an additional implicit input signal. Many ways to hybridize explicit and implicit feedback have been proposed in the literature. Some focus on the specifics of certain domains, e.g., the music domain [Jaw+10; Kor+10], TV programs [Ali+04; Yu+04], or web pages [Zha+05]. Others propose new techniques to combine the different types of feedback, for example, when using matrix factorization [Liu+10; Pil+10].

4.4.1 Hybrid Neighborhood and MF Models

In application domains where explicit ratings are available, matrix factorization (MF) techniques can nowadays be seen as the state-of-the art for efficient and accurate rating prediction. In [Kor08] the author proposes to combine classic neighborhood models and MF for explicit ratings with implicit feedback. To that end, four hybridization strategies are introduced that build on each other: (1) a neighborhood model, (2) Asymmetric-SVD, (3) SVD++, and (4) an integrated model.

The first model is based on the classic way of predicting a rating for a user, e.g., by aggregating the ratings of similar items weighted by their similarity $\hat{r}_{ui} = \sum r_{uj} \cdot sim_{ij}$. The complete neighborhood model is defined as follows:

$$\begin{aligned}\hat{r}_{ui} = \mu + b_u + b_i \\ + |R^k(i;u)|^{-\frac{1}{2}} \sum_{j \in R^k(i;u)} (r_{uj} - b_u - b_j) w_{ij} \\ + |N^k(i;u)|^{-\frac{1}{2}} \sum_{j \in N^k(i;u)} c_{ij}\end{aligned} \tag{4.12}$$

Besides the overall average rating μ and the user and item biases b_u and b_i, the *neighborhood model* includes all explicit $R^k(i;u)$ and implicit ratings $N^k(i;u)$ of the user u for the k nearest neighbors of item i (see Formula 4.12). For each item-item combination of i with its neighbors j, the sum is weighted with the factors w_{ij} and c_{ij} which model the strength of the relationship between i and j and are not given by a similarity function but learned in an alternating least squares learning phase discussed in [Kor08].

The second model is *Asymmetric-SVD* in which the computationally expensive neighborhood calculation of Formula 4.12 is substituted by an MF approach and the rating prediction is therefore changed to:

$$\begin{aligned}\hat{r}_{ui} = \mu + b_u + b_i \\ + q_i^T \Bigg(|R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} (r_{uj} - b_u - b_j) x_j \\ + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \Bigg)\end{aligned} \tag{4.13}$$

Instead of directly looking at all neighbors of item i to calculate a prediction, an "SVD-like" lower rank decomposition of the rating matrix is introduced. Compared to traditional SVD, e.g., $\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$, there are no user-wise latent factors p_u in this model. Instead, p_u is approximated and replaced with a term over all explicit $R(u)$ and implicit $N(u)$ ratings of user u (between the large parentheses in Formula 4.13). The parameters x_j and y_j are latent item weights that are learned in the optimization process. As a side note, compared to p_u in the classic SVD approach, the three model parameters q_i, x_j, and y_j are not user-dependent. Therefore, the model can directly predict ratings for a user without being completely re-trained for all users again.

The third model, *SVD++*, simplifies the *Asymmetric-SVD* model by reintroducing the latent factors p_u for each user u, but only for the explicit feedback. *SVD++* is defined as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \tag{4.14}$$

The final model combines both the *SVD++* and the *neighborhood model* into an *integrated model*. The underlying reason is that neighborhood models perform well when detecting localized relationships between a few specific items but fail to capture the overall structure in a large set of ratings [Kor08]. MF techniques, on the other hand, behave complementary. The hybrid approach is defined as:

$$\begin{aligned}\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) \\ + |R^k(i;u)|^{-\frac{1}{2}} \sum_{j \in R^k(i;u)} (r_{uj} - b_{uj}) w_{ij} \\ + |N^k(i;u)|^{-\frac{1}{2}} \sum_{j \in N^k(i;u)} c_{ij}\end{aligned} \tag{4.15}$$

In the evaluation Koren uses the Netflix dataset and generates implicit feedback by transforming the explicit ratings. He compares their methods against the classic neighborhood model $\hat{r}_{ui} = \sum r_{uj} \cdot sim_{ij}$ and SVD, and concludes that by adding implicit feedback, the recommendation accuracy can be significantly improved compared to the baselines. Also, *SVD++* performs better than *Asymmetric-SVD* when the implicit feedback is generated from explicit feedback. However, it is stated that for domains where implicit feedback is available, *Asymmetric-SVD* should in theory be more accurate.

4.4.2 Collaborative Feature-Combination

In [Zan+09] an approach to combine multiple (explicit and implicit) aspects of the user model was proposed. Classic CF approaches only take one type of rating data (explicit ratings) into account and consequently pose a challenge in cold-start situations. The proposed *collaborative feature-combination* recommender can help to deal with these challenges by considering existing implicit feedback – e.g. the navigation history of a user – if explicit feedback is not available. The general idea is to extend the single-category neighborhood calculation to multiple relevance-ordered feature dimensions. In this case, the ranking score for the recommendation can be calculated as follows.

$$
\begin{aligned}
rec_{fch^*}(i, u, d_t, d_{rec}) &= \frac{\sum_{v \in N_u} score_{i,v}}{|N_u|} \\
&\text{with} \\
score_{i,v} &= sim_{fch^*}(u, v, d_t) \text{ if } i \in R_{d_{rec},v} \preceq R_{d_t} \text{ else } 0 \\
sim_{fch^*}(u, v, d_t) &= \sum_{d \preceq d_t} w_d \times cos(\overrightarrow{R_{d,u}}, \overrightarrow{R_{d,v}}) \\
cos(\overrightarrow{a}, \overrightarrow{b}) &: \text{cosine similarity} \\
R_{d,u} &: \text{rating vector for feature dimension } d \text{ and user } u \\
R_{d_t} &: \text{threshold feature dimension} \\
w_d &: \text{feature dimension weight}
\end{aligned}
\tag{4.16}
$$

In this equation, the recommendation score consists of the average of the weighted cosine similarity $score_{i,v}$ over all users N_u. The similarity $sim_{fch^*}(u, v, d_t)$ is calculated as a weighed combination over the feature dimensions, e.g., the implicit feedback of observed *buy*, *context*, *view*, or *navigation* actions. In addition, the feature dimensions are ordered, for example, by their predictive performance, i.e., *buy* $\prec$ *context* $\prec$ *view* $\prec$ *navigation*. When creating recommendations, a *threshold feature dimension* R_{d_t} has to be specified, and the algorithm only uses feature dimensions that have a higher predictive accuracy than the threshold dimension. For example, by using the dimension *context* as the threshold, only implicit feedback of *buy* and *context* actions is included and the other (less meaningful) feature dimensions *view* and *navigation* are excluded in the calculation. The approach is therefore capable of gradually including different types of implicit feedback signals in the prediction model.

4.4.3 Bayesian Adaptive User Profiling

Similar to the collaborative feature combination approach, the authors of [Zig+06] propose a method to avoid the cold-start problem by simultaneously taking explicit and implicit feedback into account to model the user profile in a hierarchical Bayesian approach. Initially, there is only little explicit feedback available. Therefore, for new users, the model automatically focuses on the "cheap" implicit feedback and the collaborative information gathered from other users. The authors use a general Bayesian model to formalize this as follows:

$$
\begin{aligned}
f^u &\sim P(f|\theta) \\
y &= f^u(x) \\
&\text{with} \\
f^u &: \text{model of user } u \\
x &: \text{item} \\
y &: \text{rating}
\end{aligned}
\tag{4.17}
$$

From a general perspective, the user model is a function f^u for each user u that estimates a rating y for each item x and is modeled as a prior distribution on some parameters θ. In addition, the user model is personalized by learning from a sample dataset D_u for each user that consists of item/rating-pairs. With Bayes' Rule, the general model can be extended to:

$$
\begin{aligned}
P(f^u|\theta, D_u) &= \frac{P(D_u|f^u,\theta)P(f^u|\theta)}{P(D_u|\theta)} \\
&= P(f^u|\theta)\prod_{i=1}^{N_u}\frac{P(f^u(x_i^u)=y_i^u|f^u)}{P(f^u(x_i^u)=y_i^u|\theta)} \\
&\text{with} \\
D_u &= \{(x_i^u, y_i^u)|i = 1 \dots N_u\} \\
N_u &: \text{number of training samples for } u
\end{aligned}
\tag{4.18}
$$

For each user u, the belief about the user model is also based on the training data D_u. Equation 4.18 shows that the user model depends on both the model's prior probability $P(f^u|\theta)$ and the data likelihood given the user model f^u. If the number of training samples N_u for a user is small, i.e., there is little explicit feedback available, the prior probability based on the observed behavior and other users is the major contributor to the final model. For the prior, the authors use a hierarchical Gaussian network, which is further discussed in [Zig+06].

4.4.4 Reciprocal Compatibility

In [Piz+10], explicit and implicit feedback is used in the domain of online dating as a two-step approach. The explicit feedback, which consists of features like age and body type that the user prefers, is used to filter the possible recommendation results. The ranking of user profiles, on the other hand, is based on implicit feedback –

viewing user profiles, sending and replying to messages – by calculating a “reciprocal compatibility score”. This similarity measure is calculated as follows:

$$
\begin{gathered}
recip\_compat(u,v) = \frac{2}{compat(u,v)^{-1} + compat(v,u)^{-1}} \\
\text{with} \\
compat(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{k_i} \frac{f_{u,i,j}}{k_i} \times P(v,i,j)
\end{gathered}
\tag{4.19}
$$

Here, $P(v,i,j)$ indicates that some feature A_i (e.g., *body type*) has a certain value a_{ij} (e.g., *slim*) in the profile of user v. The factor $f_{u,i,j}$ is the implicit preference of user u for that features value a_{ij}, e.g., the number of times the user viewed the profile of a *slim* user. Therefore, this approach uses the observed user behavior to weight the preference of explicitly given features.

4.5 Recommending Based on Activity Logs in E-Commerce

Two recent approaches will be focused here that use the activity logs of Zalando, a larger European online retailer for fashion products, as described in [Tav+14] and [Jan+15a]. These two works were chosen as they are based on a real-world dataset[2] that contains information that is (a) typically available for many online shops and (b) corresponds to what companies might share with researchers as no sensitive information is contained in the logs. Furthermore, the log contains *all* user interactions for a given time period[3] and is not limited to a particular user group, e.g., *heavy* users. The social aspect when generating recommendations in this setting is the collective behavior of the website users which is analyzed to identify patterns in the navigation and buying behavior.

4.5.1 Data Aspects

The user activity log contains more than 24 million recorded user actions of different types (item views, purchases, cart actions). Most of the actions – about 20 million – are item views and about 1 million actions correspond to purchases. The user actions are related to more than 1.5 million *sessions*, which comprise sequential actions within a certain time frame. Each log entry contains a limited amount of information about the item itself like the category, price range, or color. The actions

[2] The data is not publicly available.
[3] The data was sampled in a way that no conclusions about visitor or sales numbers can be drawn.

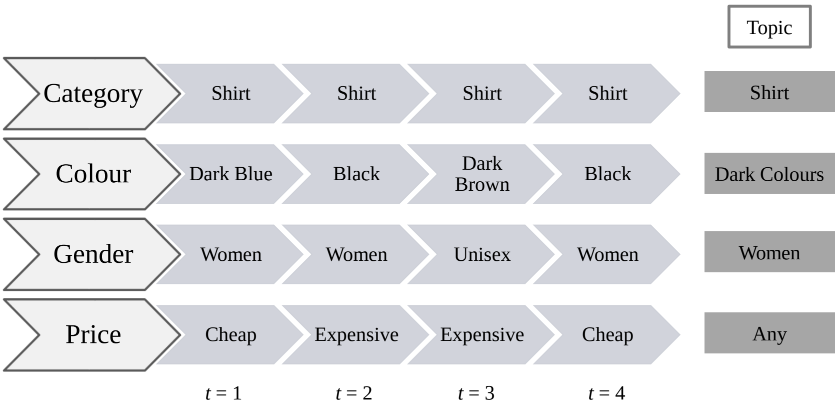

Fig. 4.5: Viewing a session as a sequence of attributes [Tav+14].

were performed by about 800.000 users. The catalog of products (including product variants) appearing in the log is huge and consists of around 150.000 items.

The dataset exemplifies several of the challenges mentioned in Section 2.3, including the abundance of data[4], the general sparseness with respect to the available purchase data as the majority of users has never made any purchase, and the problem of the interpretation of the strength of the different signals.

On the other hand, such datasets allow to perform analyses and design algorithms closer to the demands of real-world recommendation systems than the non-contextualized ex-post prediction of missing entries in a user-item rating matrix, which is the most common evaluation setup in research [Jan+12b].

4.5.2 Topic Detection for User Sessions

In [Tav+14], the authors present an approach to automatically infer the "topic" or short-term shopping goal for the individual user sessions. The proposed approach for topic detection is based on Markov Decision Processes (MDP) and can be easily transformed to serve as a topic-driven recommendation technique or MDP-based recommender system [Sha+05].

The basic idea of their approach is to view each session as a sequence of item attributes as shown in Figure 4.5. The example shows that the user has only viewed items from the category "shirt". However, the shirts had different colors and different price ranges. The general topic (shopping goal) to be inferred is shown on the right-hand side of the figure, i.e., the user looked for dark-colored shirts for women in any price range.

[4]The data sample was taken within a limited period of time.

Approach. Technically, the idea is to model the topic detection problem as a reinforcement-learning problem based on Markov Decision Processes. The observed sequences of actions – in this case sequences of item features – are therefore used to train a model to predict the most likely next observation (state). The distribution of the item attribute values in the user session are considered the *topic* of the session, which can then be leveraged in the recommendation process. The learned models are strictly session-dependent, i.e., no long-term profile of the individual user is built in this approach.

Relying on MDPs for the recommendation task was done, although in a different form, for example in [Sha+05]. The particular challenge however lies in the computational complexity of such an approach given the huge amounts of items and possible states. In [Tav+14], this scalability problem is addressed by using *factorized* MDPs. Instead of sequences of observed interactions with items, they model sequences of item attribute values and learn such fMDPs independently for each attribute. Furthermore, an approximation technique is used in the optimization phase to avoid scalability problems in terms of memory requirements.

As a result of the approximation process the probabilities that express the most likely next observed attribute values are obtained. This information can be used to extract the topic of the session as well as to rank items based on their particular item features.

Results. In their empirical evaluation, the authors first compare different strategies for topic detection. The results show that their method is highly accurate in predicting the topic (around 90%, depending on the length of the observed history) and much better than the compared baselines, among them a simple Markov process based on the frequencies of item clicks.

Second, a comparison was made for the recommendation tasks where the baselines include (a) models that rely on the long-term user profile and (b) latent factor techniques. They were compared with more simple baselines that recommend popular items or items that are feature-wise similar to the last viewed item. As an evaluation measure, the *average rank* (position) of the correct item in the recommendations was used. The results show that the proposed MDP-based method is better than the collaborative filtering (CF) methods, which rely only on longer-term models. In addition, also the simple contextualized baseline methods are better than the CF methods.

Discussion. From a general perspective, the experiments in [Tav+14] show that the consideration of short-term shopping goals in conjunction with the sequence of the observed user actions can be crucial for the success of recommendation systems in real-world environments. Assessing the true value of the final recommendations unfortunately remains challenging as even the best performing method only lead to an average rank of 15,000 (due to the large item assortment in the shop).

The results also indicate that optimizing for long-term goals alone as done in the state-of-the-art baseline methods can be insufficient. Overall, at least in the e-commerce domain, using implicit feedback data with time information might help to develop models closer to real-world requirements than models that generate time- and situation-agnostic predictions for missing items in the rating matrix. Furthermore, the work highlights scalability limitations of existing approaches when it comes to real-world datasets. For the first set of experiments, the authors merely used a few percent of the available data to be able to perform the optimization. For the larger dataset, unfortunately no information is provided on computation times for model building and generating recommendations.

The work in [PC+14] is a related case study for context-aware recommendations of shopping places. The authors employ a post-filtering approach based on the user's short-term goals to create an intention-based ranking of relevant nearby locations.

4.5.3 Evaluating a Combination of Short-Term and Long-Term Models

The discussion in the previous section indicated the importance of generating recommendations that consider the recent short-term user intent while solely exploiting long-term preference models might be insufficient. In fact, many of the recommendations of popular e-commerce sites like Amazon.com are either simply reminders of recently viewed items or recommendations that are connected to the currently viewed item ("Users who viewed ... also viewed ...").

In one of the works included in this thesis [Jan+15a], the authors aim to quantify the effectiveness of such comparably simple recommendation strategies and furthermore analyze the possible benefits of combining them with optimized long-term models. One further goal is to assess how quickly the different strategies are able to adapt their recommendations after the most recent user action in a session.

Evaluation Protocol. Since standard evaluation setups in the research literature do not cover situations in which time-ordered session logs are available, the authors propose a parameterizable and domain-independent evaluation protocol as shown in Figure 4.6.

The general idea is to split the data as usual into training and test data while maintaining the order of the log entries. The task in the test phase is then to predict

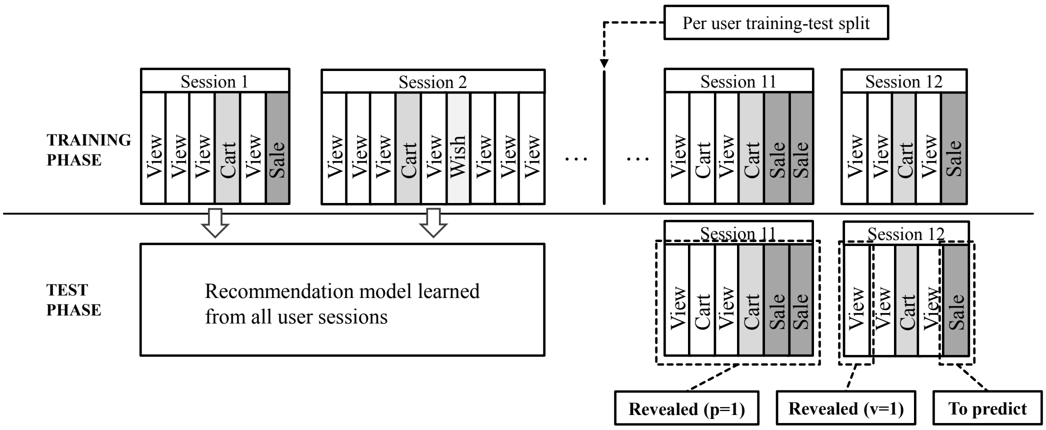

Fig. 4.6: Proposed Evaluation Protocol [Jan+15a].

for each buy event of a session in the test data the item that was purchased. In contrast to similar protocols, e.g., the one used in the ACM RecSys 2015 Challenge, the idea is to vary the amount of information that a recommender is allowed to see from the *current* and *previous* session. In one setup, for example, the first 2 item views of the current session and all user actions of the preceding session could be revealed. Using this extra information, it is for example possible to assess the effectiveness of a strategy that recommends the most recently viewed items. As a success measure, the Recall can be used which indicates if the purchased item was in the top-k list.

Algorithms and Results. A number of different algorithms were used, including the one-class CF method BPR described in Section 4.2.3 as well as the more recent Factorization Machines approach of [Ren12]. These long-term preference modeling approaches were then combined with a number of short-term adaptation strategies, including approaches that recommend (a) the most recently viewed items, (b) items that are similar to those viewed in the current context regarding their content features, (c) generally popular items, or (d) items that co-occurred with the currently viewed ones in past transactions. Combinations of the different short-term strategies were tested as well.

Similar to the findings reported in [Tav+14], the results show that standard CF methods like Factorization Machines do not perform well at all in this evaluation setup and only the BPR method, which has a comparably strong popularity bias, outperforms the popularity-based baseline when no context information is available.

All short-term adaptation strategies, on the other hand, immediately led to a strong increase in terms of the Recall even when a weak baseline strategy was used and only the first two item views in a session were revealed. Although the comparison of context-agnostic long-term models and the short-term strategies is in some sense "unfair" as a few more user actions are known to the short-term strategy, the strong increase in accuracy helps to quantify the importance of the adaptation process.

In the end, the best-performing method was a hybrid technique which used BPR as a baseline and adapted the recommendation lists by favoring both recently viewed items as well as items whose features are similar to those that were viewed in the current session.[5] In absolute numbers, the Recall of the baseline method of 0.40 was increased to 0.66 through the hybrid method for a configuration in which only the first two item views of the current session and the last two preceding sessions were revealed.

Discussion. Although the short-term adaptations in the experimental analysis were effective, the results also show that the choice of a strong baseline and the capability of understanding the user's long-term preferences are important. On the other hand, while the results of the log-based analysis emphasize the importance of considering short-term interests, it is not fully clear whether the "winning" models fulfill the business goals of the shop owner in the best possible way. The BPR method, for example, can exhibit a comparably strong tendency of recommending popular items and is probably not very helpful when the goal of the recommendation component in a shop is to guide the customers to long-tail items or to help them discover additional or new items in the catalog.

Reminding users of recently viewed items shows to be very effective, e.g., because users might have a tendency to postpone their buying decisions for at least another day in order to sleep on them. However, while the strategy leads to good results in terms of the Recall, it is unclear if the recommendations generate any additional revenue for the shop owner.

In the work in [Jan+15a], user actions like "add-to-wish-list" or "put-in-cart" were not considered and more work is required to understand (a) how to weight these user actions in comparison to, e.g., view actions and (b) whether or not it is reasonable from an application perspective to remind users of the items in their carts or wish lists.

4.6 Considering Application-Specific Requirements in BPR

The e-commerce domain discussed in the section before is a typical example for a domain where implicit feedback is prevalent and BPR is a state-of-the-art ranking

[5] The importance of feature-based similarities was also the basis in [Tav+14].

algorithm for implicit feedback one-class collaborative filtering situations (Section 4.2). Since its original presentation in [Ren+09], several variations and extensions were proposed for BPR in the literature, e.g., to make the algorithm better suited for certain application requirements like the aforementioned weighting of different kinds of user actions. In this section, some of these proposals will be examined in more detail. Among others, an approach to counteract the popularity bias of the algorithm will be discussed and algorithm extensions to deal with graded relevance feedback will be presented.

Besides the discussed methods, multiple other enhancements to BPR were proposed in the literature. The improvements are for example related to the inclusion of the temporal information, the social connection, or the item taxonomy [Du+11; Kan+12; Ren+10a]. Some approaches also extend the two-dimensional user-item perspective of BPR towards additional dimensions [KG+12; Liu+15; Ren+10b]. In terms of the pair-wise item-item relations, there are some approaches that introduce the concept of group-wise relations [Pan+13a; Pan+13b].

4.6.1 Dealing with the Popularity Bias

Some one-class collaborative filtering algorithms discussed in Section 4.2 use different strategies to create artificial negative feedback signals. In most cases, some kind of weighting or sampling scheme is used to derive negative feedback from the structure of the observed interactions. In the wALS, sALS-ENS, Random Graph, and BPR approaches, the created negative examples were chosen in a way that was inversely proportional to the popularity of the items, i.e., the algorithms assume that popular items are more acceptable.

While in general this assumption seems plausible, it can lead to a popularity bias in the recommendations, i.e., the algorithms tend to recommend popular items to everyone. In the journal paper [Jan+15f] that is included in this thesis, the authors compared a number of recommendation techniques regarding popularity and concentration criteria and showed that BPR tends to focus strongly on the most popular items. Although popularity-biased recommendations can lead to high values in terms of Precision and Recall [Jan+15f], the bias might be undesired in specific application settings.

In BPR, the popularity bias emerges from the specific way the algorithm takes samples to learn the preference relations. As discussed in 4.2, the BPR algorithm optimizes the set of model parameters Θ with a stochastic gradient descent procedure by randomly sampling triples (u, i, j) from D_S. The distribution of the observed (positive feedback) signals is typically non-uniform, i.e., the popularity of the items has a long-tail shape. As a result, sampling the triples randomly from all observed

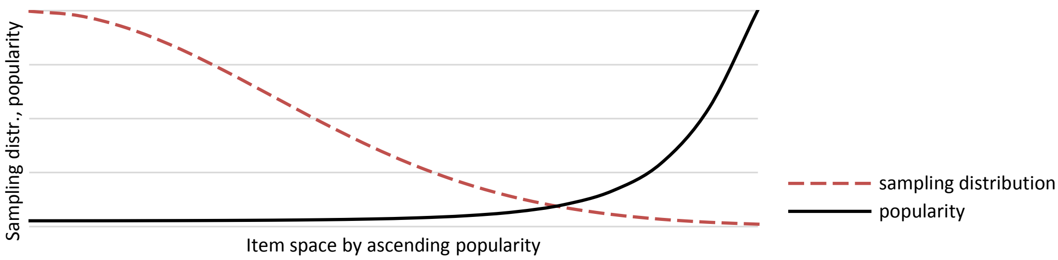

Fig. 4.7: Sampling distribution for items i with positive feedback compared to their popularity. The x-axis represents the items space by ascending popularity. The y-axis contains the distribution of the item popularity (solid line) and sampling probability of function ϕ (dashed line). Popular items are sampled less frequently.

interactions leads to a large amount of sampled triples (u, i, j) where the item i is a popular item. The item j, however, is randomly sampled over all items and thus more likely to be part of the long-tail of unpopular items. As a result, the gradient descent algorithm updates the model parameters with many triples (u, i, j) that contain a (popular, unpopular) item pair, and therefore BPR favors popular items in the recommendation step. In [Jan+15f], an adapted sampling strategy was introduced that counters this popularity bias of BPR.

Approach. Instead of applying random uniform sampling, a modified distribution function ϕ is used to sample the triples (u, i, j). The non-uniform sampling with a distribution ϕ biases the sampling probability of the items i in a way that more unpopular items are sampled. As a result, the model is updated more often with tuples (u, i, j) where i is less popular. Figure 4.7 shows the shape of a sampling function ϕ that can be used to sample the positive feedback signals of items i (dashed line) in comparison to the popularity distribution of the item space (solid line). The function ϕ is a monotonously decreasing distribution function that samples more popular items with a lower probability.

Different distribution functions for ϕ are possible and in [Jan+15f] a normal distribution $\phi(\omega)$ with a mean of $\mu = 0$ and a standard deviation $\sigma = \frac{|L_u|}{\omega}$ is used. Here, $|L_u|$ is the number of rated items of user u.

$$\phi(\omega) = \mathcal{N}\left(0, \left(\frac{|L_u|}{\omega}\right)^2\right) \tag{4.20}$$

The strength of the counter-bias in the new sampling process can be chosen by varying the breadth of the function with the parameter ω. For example, increasing ω leads to a narrower distribution $\phi(\omega)$ and less popular items are sampled for i. On the other hand, setting $\omega < 1$ leads to a more uniform selection of items.

Results and Discussion. The proposed sampling method [Jan+15f] was compared with the original implementation by Rendle et al. [Ren+09] on the MovieLens and

Yahoo movie datasets. The results show that there is an (expected) trade-off between recommendation accuracy and the popularity bias countermeasures. When increasing the breadth ω of the sampling distribution, thus focusing on more unpopular items, the *average popularity* and the *Gini index* (to measure the concentration of recommendations) decrease.At the same time, however, Precision and Recall also decrease, but at a much lower rate. A 5% decrease in accuracy can for example be traded in for a 10% reduction of the overall popularity of the recommended items. The actual size of the desired effect can be determined based on the specific application requirements.

4.6.2 Supporting Graded Relevance Feedback

As mentioned before, in many domains implicit feedback occurs not only as a binary indicator but in a graded form. In the e-commerce domain, for example, different observable user actions like item views or purchases are interest indicators of different strength. The repeated consumption of items in an online media service is another example for an indicator that should be interpreted as a stronger signal than a single consumption event.

In its original form, BPR only supports binary feedback but there are some extensions that allow different graded signal strengths to be considered when learning the preference relations.

BPRC. To that end, Wang et al. [Wan+12] introduced a confidence weight in the objective function of the BPR-OPT criterion. In their *BPR with confidence* approach, the confidence score originates from the problem setting of recommending social network messages and is calculated based on the difference of reception times of two messages. Thus, the optimization criterion is extended as follows to BPRC-OPT:

$$
\begin{aligned}
\text{BPRC-OPT} &= \sum_{(u,i,j)\in D_S} \ln \sigma(c_{uij}\hat{x}_{uij}) - \lambda_\Theta ||\Theta||^2 \\
&\text{with} \\
c_{uij} &= \frac{1}{t_i - t_j} : \text{confidence weight}
\end{aligned}
\tag{4.21}
$$

The confidence weight c_{uij} is the inverse of the difference between the reception times t_i and t_j of two messages. If the time between two messages is long, the confidence weight c_{uij} lowers their impact $\hat{x}_{uij}$ in the training phase of the model. Since the confidence values are given by the application setting, the optimization is analogous to BPR.

The approach was benchmarked against classic kNN, MF, and BPR on a dataset from Sina Weibo[6], which is a microblog system similar to Twitter. The BPRC approach has the same recommendation characteristics as BPR but has a higher accuracy in terms of Precision and Recall. In addition, the authors report that the confidence-based method greatly outperforms many other algorithms in a cold-start scenario.

ABPR. In [Pan+15], a similar generalization of BPR for so-called "heterogeneous implicit feedback" has been proposed. This so-called *Adaptive Bayesian Personalized Ranking* (ABPR) has the ability to model and reduce uncertainty for different types of observed feedback. In their work, the authors discuss a problem setting with two types of implicit feedback: transactions and examinations, i.e., item purchase events and item click events in an online store. A naive approach would be to assume that both types of user actions are equivalent positive implicit feedback. However, viewing a product page is not necessarily positive feedback, and using only the transactions would result in sparse training data. Therefore, like the confidence-extension for BPR [Wan+12], the ABPR approach assumes confidence weights for both types of feedback. The optimization criterion thus extended to:

$$\begin{aligned} \text{ABPR-O}\textsc{pt} = \sum_{(u,i,j)\in D_S} & f^T_{uij}(c_{ui},\Theta) + \lambda_E f^E_{uij}(c_{ui},\Theta) - \lambda_\Theta ||\Theta||^2 \\ & \text{with} \\ f^T_{uij}, f^E_{uij} &: \text{estimation function for transaction, examination} \\ c_{ui} &: \text{individual confidence weight} \\ \lambda_E &: \text{global confidence weight} \end{aligned} \tag{4.22}$$

The optimization criterion now depends on both the estimation of the transactions f^T_{uij} and examinations f^E_{uij}. Furthermore, the impact of the examinations is controlled by a global confidence weight parameter λ_E and the individual confidence weights c_{ui} determine the impact for each transaction or examination triple (u, i, j). Compared to the BPRC approach, where the messages had a time stamp that determined the confidence, in this setting the weights are not deduced directly from some form of meta-data and are instead determined in the optimization process. When there is a transaction for user u and item i in the training data, the confidence weight is assumed to be 1. Otherwise it is initially unknown and learned in the extended stochastic gradient descent algorithm discussed in [Pan+15]. This is in some sense similar to the graph-based approach for one-class implicit feedback discussed in Section 4.2.2.

ABPR was benchmarked on the MovieLens and Netflix datasets. In terms of accuracy and ranking metrics, it performs significantly better than classic BPR that uses both types of implicit feedback in a naive way.

[6] `http://www.weibo.com`

BPR++. In the paper [Ler+14] included in this thesis, a graded preference relation scheme is introduced to extend the set of triples D_S used for training the model in the gradient descent phase. This new set D_S^{++} includes additional triples based on the graded observed feedback. Similar to the BPRC and ABPR approaches discussed before, the goal of the BPR++ technique is to adapt BPR to non-binary feedback, e.g., the confidence in the interaction, the number of times an interaction was observed, the recency of the interaction or the type of an interaction. The enlarged training set is defined as follows:

$$D_S^{++} := \{(u,i,j)|pweight(u,i) > pweight(u,j), i \in I, j \in I\} \tag{4.23}$$

The function $pweight(u,i)$ is a real-valued preference weight function that models the strength of the interaction between user u and item i, e.g., confidence, time, or rating. If there is no interaction between u and i, the preference weight is 0. Compared to the original set of training triples as shown before in Equation 4.2, the extended set D_S^{++} contains all triples of D_S. In addition, triples that would have been ignored in BPR because both items had observed feedback can now appear in D_S^{++} if they have a different preference weight.

The number of these additional triples $D_S^{++} \setminus D_S$ is however comparably small, which means that these triples will not be often considered when using the random sampling strategy of BPR. The authors therefore introduce a weighted sampling approach, similar to the one presented in [Jan+15f], which increases the sampling probability for the new triples in $D_S^{++} \setminus D_S$.

The BPR++ approach was benchmarked against the original BPR technique on two e-commerce datasets with implicit feedback and a MovieLens dataset with ratings. Different preference weight functions where used in the experiments to model the strength of the relevance including the time when an interaction occurred, the number of interactions, and – for the MovieLens data – the ratings. The results show that on the implicit feedback datasets the use of the interaction time with BPR++ significantly increased the accuracy (Precision@10 and Recall@10) when compared to BPR. On the MovieLens dataset this is also true when the preference strength is modeled by ratings.

Besides being able to improve the prediction accuracy, the adapted sampling strategy helps to reduce the time needed for the gradient descent procedure to converge.

5 Conclusion

For a long time, research in the field of recommender systems was mainly focused on rating prediction tasks based on datasets of explicit user feedback. This was caused by competitions like the Netflix Prize and the availability of publicly available data. These "Machine Learning"-like problem settings, however, ignored some practical challenges that are unique to the field of recommender systems.

In recent years this trend has changed. With the rise of social networks and e-commerce, recommendation techniques that can deal with new practical requirements and challenges have gained importance. In these domains, implicit feedback is prevalent.

5.1 Summary

This thesis by publication provided a general review on the use of implicit feedback in recommender systems, especially in the *Social Information Access* domain, which utilizes the "community wisdom" of the social web for information search and retrieval [Bru08]. With the other included publications, it further examined topics like contextualized recommendations and reminders in the e-commerce domain, the popularity bias of recommender algorithms, and specialized implicit feedback techniques. This should provide the reader with an overview of the complex challenges of implicit feedback recommendations.

In the introductory chapters of this thesis, examples of common application scenarios and challenges for implicit feedback were discussed and a categorization scheme was established that classifies different types of observable behavior. In addition, several state-of-the-art algorithmic approaches that employ implicit feedback signals were examined. Among them were techniques that allow the use of "classic" recommendation approaches for explicit feedback by intelligently converting implicit signals to explicit ratings, as well as one-class algorithms that can directly handle implicit feedback. The discussion also included common techniques used in e-commerce ("Because you bought ...") and elaborated on learning-to-rank algorithms, especially Bayesian Personalized Ranking and its extensions.

One of these extensions, BPR++, was proposed in the included publications. It enables the underlying BPR algorithm to use more fine-grained information from the implicit signals, like the interaction intensity, duration, or strength, instead of just unary indicators. In the evaluation it was shown that, depending of the type of information taken from the signals, the predictive accuracy compared the original BPR can be increased.

A domain where different types of feedback signals are very frequent is the e-commerce domain. For example, viewing an item in an online store might be interpreted differently than purchasing this item. To recommend products that match the customer's *current* interest, it is often essential to utilize implicit signals, since the previous explicit ratings of a customer cannot reflect this short-term interest. Two of the included publications proposed novel approaches to create recommendations that are contextualized to the customer's current short-term goals and long-term preferences, as well as reminding techniques that present relevant items that were previously browsed by a customer but then abandoned. The works showed that contextualization is a key to "good" recommendations in e-commerce and that reminders seem to be well accepted by the customers, which was validated with an experiment on a real online store.

The final publication included in this thesis discussed the characteristics of different recommender algorithms. The evaluation was executed on multiple datasets from different domains and included – besides predictive accuracy – properties like the coverage, concentration, diversity, and popularity of the recommendations. The tendency to recommend popular items, the *popularity bias*, can be problematic in practical settings. Offline experiments often show that algorithms that have this popularity bias often achieve very good predictive performance. In practice, this might not be the case since recommending mostly popular items might not be the desired intention and other criteria could be more important to create "good" recommendations. Different approaches to counter this popularity bias were then proposed in the publication.

5.2 Perspectives

The work in [Kon+12] identified three key challenges for future developments in the field of recommender systems. These challenges – *scalability*, *better exploitation of user-contributed content* and *research infrastructure* – are especially applicable for domains where implicit preference signals are prevalent. As we have seen in this thesis, the last challenge is particularly crucial. It stresses the need for more effective evaluation techniques and metrics to assess the value that theoretical concepts and research contributions can have in practical applications when real users are involved.

Even if such evaluation measures are found, offline experiments can never fully replace the real-live evaluations like user studies and A/B tests. Although an offline multi-metric evaluation can help to understand the characteristics of recommendation techniques, there are far too many external factors involved that influence the true value of recommendations [GU+15]. Lastly, the recommendation techniques that should be employed in practice always have to be selected by taking into account the specific goals of the application and the domain.

Finally, there are ongoing debates whether some research results can be reproduced or are even valid at all [Sai+14a]. Many works on implicit feedback recommender systems are based on non-public data, often due to privacy constraints. Furthermore, there are no standardized datasets and evaluation frameworks available that are broadly accepted and used by the research community.

In summary, research on recommender systems and more specifically implicit feedback has come a long way. Still, a lot of issues and open challenges remain for the future and this thesis – hopefully – will help to tackle at least some of them.

Bibliography

[Abe+11] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. "Semantic Enrichment of Twitter Posts for User Profile Construction on the Social Web". In: *Proceedings of the 8th Extended Semantic Web Conference on The Semanic Web: Research and Applications*. (Heraklion, Crete, Greece). Vol. Part II. ESWC '11. 2011, pp. 375–389 (cit. on p. 24).

[Ado+11] Gediminas Adomavicius and Alexander Tuzhilin. "Context-Aware Recommender Systems". In: *Recommender Systems Handbook*. Springer, 2011, pp. 217–253 (cit. on p. 5).

[Agr+93] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. "Mining Association Rules Between Sets of Items in Large Databases". In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. (Washington, DC, USA). SIGMOD '93. 1993, pp. 207–216 (cit. on pp. 21, 42, 43).

[Agr+95] Rakesh Agrawal and Ramakrishnan Srikant. "Mining Sequential Patterns". In: *Proceedings of the Eleventh International Conference on Data Engineering*. (Washington, DC, USA). ICDE '95. 1995, pp. 3–14 (cit. on p. 44).

[Ali+04] Kamal Ali and Wijnand van Stam. "TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (Seattle, WA, USA). KDD '04. 2004, pp. 394–401 (cit. on pp. 22, 45).

[Arm+12] Marcelo G. Armentano, Daniela Godoy, and Analía Amandi. "Topology-based Recommendation of Users in Micro-Blogging Communities". English. In: *Journal of Computer Science and Technology* 27.3 (2012), pp. 624–634 (cit. on p. 24).

[Arm+95] Robert Armstrong, Dayne Freitag, Thorsten Joachims, and Tom Mitchell. "WebWatcher: A Learning Apprentice for the World Wide Web". In: *AAAI Technical Report SS-95-08*. 1995, pp. 6–12 (cit. on p. 20).

[Bas+14] Mathieu Bastian, Matthew Hayes, William Vaughan, Sam Shah, Peter Skomoroch, Hyungjin Kim, et al. "LinkedIn Skills: Large-scale Topic Extraction and Inference". In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, California, USA). RecSys '14. 2014, pp. 1–8 (cit. on p. 6).

[BD+97] Shai Ben-David and Michael Lindenbaum. “Learning Distributions by their Density Levels: A Paradigm for Learning without a Teacher”. In: *Journal of Computer and System Sciences* (1997) (cit. on p. 41).

[Boh+14] Fabian Bohnert and Ingrid Zukerman. “Personalised Viewing-time Prediction in Museums”. English. In: *User Modeling and User-Adapted Interaction* 24.4 (2014), pp. 263–314 (cit. on p. 25).

[Bon+14] Geoffray Bonnin and Dietmar Jannach. “Automated Generation of Music Playlists: Survey and Experiments”. In: *ACM Computing Surveys* 47.2 (2014), 26:1–26:35 (cit. on pp. 23, 24, 45).

[Bru+17] Peter Brusilovsky and Daqing He. *Social Information Access*. Vol. 10100. LNCS. Heidelberg: Springer, 2017 (cit. on p. 2).

[Bru08] Peter Brusilovsky. “Social information access: the other side of the social web”. In: *International Conference on Current Trends in Theory and Practice of Computer Science*. 2008, pp. 5–22 (cit. on pp. 3, 61).

[Bud+99] Jay Budzik and Kristian Hammond. “Watson: Anticipating and Contextualizing Information Needs”. In: *62nd Annual Meeting Of The American Society For Information Science*. 1999, pp. 727–740 (cit. on p. 15).

[Car+09] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har’El, Inbal Ronen, et al. “Personalized Social Search Based on the User’s Social Network”. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM ’09. 2009, pp. 1227–1236 (cit. on p. 24).

[Cas+10] Sylvain Castagnos, Nicolas Jones, and Pearl Pu. “Eye-tracking Product Recommenders’ Usage”. In: *Proceedings of the 2010 ACM Conference on Recommender Systems*. (Barcelona, Spain). RecSys ’10. 2010, pp. 29–36 (cit. on p. 20).

[Che+14] Zhiyong Cheng and Jialie Shen. “Just-for-Me: An Adaptive Personalization System for Location-Aware Social Music Recommendation”. In: *Proceedings of International Conference on Multimedia Retrieval*. ICMR ’14. 2014, 185:185–185:192 (cit. on p. 23).

[Cla+01] Mark Claypool, Phong Le, Makoto Wased, and David Brown. “Implicit Interest Indicators”. In: *Proceedings of the 6th International Conference on Intelligent User Interfaces*. (Santa Fe, NM, USA). IUI ’01. 2001, pp. 33–40 (cit. on pp. 33, 34).

[Cre+10] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. “Performance of Recommender Algorithms on Top-n Recommendation Tasks”. In: *Proceedings of the 2010 ACM Conference on Recommender Systems*. (Barcelona, Spain). RecSys ’10. 2010, pp. 39–46 (cit. on pp. 6, 7).

[Cre+11] Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Papadopoulos, and Roberto Turrin. “Comparative Evaluation of Recommender System Quality”. In: *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems*. (Vancouver, BC, Canada). CHI EA ’11. 2011, pp. 1927–1932 (cit. on p. 8).

[Cre+12] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. “Investigating the Persuasion Potential of Recommender Systems from a Quality Perspective: An Empirical Study”. In: *ACM Transactions on Interactive Intelligent Systems* 2.2 (June 2012), 11:1–11:41 (cit. on p. 8).

[Dav+10] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, et al. “The YouTube Video Recommendation System”. In: *Proceedings of the 2010 ACM Conference on Recommender Systems*. (Barcelona, Spain). RecSys ’10. 2010, pp. 293–296 (cit. on p. 45).

[Den98] François Denis. “PAC Learning from Positive Statistical Queries”. English. In: *Algorithmic Learning Theory*. Vol. 1501. Lecture Notes in Computer Science. Springer, 1998, pp. 112–126 (cit. on p. 42).

[Dia+08] M. Benjamin Dias, Dominique Locher, Ming Li, Wael El-Deredy, and Paulo J.G. Lisboa. “The Value of Personalised Recommender Systems to E-business: A Case Study”. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*. (Lausanne, Switzerland). RecSys ’08. 2008, pp. 291–294 (cit. on p. 4).

[DP+14] Toon De Pessemier, Simon Dooms, and Luc Martens. “Context-aware Recommendations Through Context and Activity Recognition in a Mobile Environment”. English. In: *Multimedia Tools and Applications* 72.3 (2014), pp. 2925–2948 (cit. on p. 25).

[Du+11] Liang Du, Xuan Li, and Yi-Dong Shen. “User Graph Regularized Pairwise Matrix Factorization for Item Recommendation”. English. In: *Advanced Data Mining and Applications*. Ed. by Jie Tang, Irwin King, Ling Chen, and Jianyong Wang. Vol. 7121. Lecture Notes in Computer Science. Springer, 2011, pp. 372–385 (cit. on p. 56).

[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 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, CA, USA). RecSys ’14. 2014, pp. 161–168 (cit. on p. 6).

[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. 6).

[Gad+07] Sandra Clara Gadanho and Nicolas Lhuillier. “Addressing Uncertainty in Implicit Preferences”. In: *Proceedings of the 2007 ACM Conference on Recommender Systems*. (Minneapolis, MN, USA). RecSys ’07. 2007, pp. 97–104 (cit. on p. 22).

[Gar+09] Enrique Garcia, Cristobal Romero, Sebastian Ventura, and Carlosde Castro. “An Architecture for Making Recommendations to Courseware Authors Using Association Rule Mining and Collaborative Filtering”. In: *User Modeling and User-Adapted Interaction* 19.1-2 (2009), pp. 99–132 (cit. on p. 45).

[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 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, CA, USA). RecSys ’14. 2014, pp. 169–176 (cit. on pp. 6, 7).

[Ged+13] Fatih Gedikli and Dietmar Jannach. “Improving Recommendation Accuracy Based on Item-specific Tag Preferences”. In: *ACM Transactions on Intelligent Systems and Technology* 4.1 (2013), 11:1–11:19 (cit. on p. 24).

[Gil+12] Miriam Gil and Vicente Pelechano. “Exploiting User Feedback for Adapting Mobile Interaction Obtrusiveness”. In: *Proceedings of the 6th International Conference on Ubiquitous Computing and Ambient Intelligence*. (Vitoria-Gasteiz, Spain). UCAmI ’12. 2012, pp. 274–281 (cit. on p. 25).

[GU+15] Carlos A Gomez-Uribe and Neil Hunt. “The Netflix Recommender System: Algorithms, Business Value, and Innovation”. In: *ACM Transactions on Management Information Systems (TMIS)* 6.4 (2015), p. 13 (cit. on pp. 6, 8, 63).

[Guy+09] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, et al. “Personalized Recommendation of Social Software Items Based on Social Relations”. In: *Proceedings of the 2009 ACM Conference on Recommender Systems*. (New York, NY, USA). RecSys ’09. 2009, pp. 53–60 (cit. on p. 24).

[Guy+10] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. “Social Media Recommendation Based on People and Tags”. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’10. 2010, pp. 194–201 (cit. on p. 24).

[Han+00] Jiawei Han, Jian Pei, and Yiwen Yin. “Mining Frequent Patterns Without Candidate Generation”. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. (Dallas, TX, USA). SIGMOD ’00. 2000, pp. 1–12 (cit. on p. 43).

[Har+12] Negar Hariri, Bamshad Mobasher, and Robin Burke. “Context-aware Music Recommendation Based on Latent Topic Sequential Patterns”. In: *Proceedings of the 2012 ACM Conference on Recommender Systems*. (Dublin, Ireland). RecSys ’12. 2012, pp. 131–138 (cit. on p. 45).

[Har+15a] Negar Hariri, Bamshad Mobasher, and Robin Burke. “Adapting to User Preference Changes in Interactive Recommendation”. In: *Proc. IJCAI ’15*. 2015, pp. 4268–4274 (cit. on p. 6).

[Har+15b] F. Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren Terveen. “Putting Users in Control of Their Recommendations”. In: *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys ’15. 2015, pp. 3–10 (cit. on p. 6).

[Hen63] C. B. Hensley. “Selective Dissemination of Information (SDI)”. In: *Proceedings of the 1963 Spring Joint Computer Conference*. (Detroit, MI, USA). AFIPS ’63. 1963, pp. 257–262 (cit. on p. 19).

[Her+04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. “Evaluating Collaborative Filtering Recommender Systems”. In: *ACM Transactions on Information Systems* 22.1 (Jan. 2004), pp. 5–53 (cit. on p. 6).

[Hil+92] William C. Hill, James D. Hollan, Dave Wroblewski, and Tim McCandless. “Edit Wear and Read Wear”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’92. 1992, pp. 3–9 (cit. on p. 15).

[Hu+08] Yifan Hu, Yehuda Koren, and Chris Volinsky. “Collaborative Filtering for Implicit Feedback Datasets”. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. ICDM ’08. 2008, pp. 263–272 (cit. on pp. 22, 36).

[Hua+04] Zan Huang, Wingyan Chung, and Hsinchun Chen. “A Graph Model for E-commerce Recommender Systems”. In: *Journal of the American Society for Information Science and Technology* 55.3 (2004), pp. 259–274 (cit. on p. 45).

[Höö+12] Kristina Höök, David Benyon, and Alan J. Munro. *Designing Information Spaces: The Social Navigation Approach*. Springer Science & Business Media, 2012 (cit. on p. 23).

[Jan+09] Dietmar Jannach and Kolja Hegelich. “A Case study on the Effectiveness of Recommendations in the Mobile Internet”. In: *Proceedings of the 2009 ACM Conference on Recommender Systems*. (New York, NY, USA). RecSys ’09. 2009, pp. 205–208 (cit. on pp. 2, 7, 11, 32).

[Jan+11] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems – An Introduction*. Cambridge University Press, 2011 (cit. on p. 1).

[Jan+12a] Dietmar Jannach, Zeynep Karakaya, and Fatih Gedikli. “Accuracy Improvements for Multi-criteria Recommender Systems”. In: *Proceedings of the 13th ACM Conference on Electronic Commerce*. (Valencia, Spain). EC ’12. 2012, pp. 674–689 (cit. on pp. 2, 11).

[Jan+12b] Dietmar Jannach, Markus Zanker, Mouzhi Ge, and Marian Gröning. “Recommender Systems in Computer Science and Information Systems – A Landscape of Research”. In: *Proceedings of the 13th International Conference on E-Commerce and Web Technologies*. (Vienna, Switzerland). EC-WEB ’12. 2012, pp. 76–87 (cit. on p. 51).

[Jan+13a] Dietmar Jannach and Lukas Lerche. “Perspektiven in der Offline-Evaluation von Empfehlungsalgorithmen”. In: *HMD Praxis der Wirtschaftsinformatik* 50.5 (2013), pp. 34–44 (cit. on p. 86).

[Jan+13b] Dietmar Jannach, Lukas Lerche, and Matthäus Gdaniec. “Re-ranking Recommendations Based on Predicted Short-term Interests–A Protocol and First Experiment”. In: *Proceedings of the Workshop Intelligent Techniques for Web Personalization and Recommender Systems at AAAI 2013*. ITWP ’13. 2013 (cit. on pp. 9, 86).

[Jan+13c] Dietmar Jannach, Lukas Lerche, Fatih Gedikli, and Geoffray Bonnin. “What Recommenders Recommend – An Analysis of Accuracy, Popularity, and Sales Diversity Effects”. In: *Proceedings of the 21st International Conference on User Modeling, Adaptation and Personalization*. UMAP ’13. 2013, pp. 25–37 (cit. on pp. 8, 86).

[Jan+15a] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Adaptation and Evaluation of Recommendations for Short-term Shopping Goals”. In: *Proceedings of the 2015 ACM Conference on Recommender Systems*. (Vienna, Austria). RecSys ’15. 2015, pp. 211–218 (cit. on pp. 4–6, 9, 21, 29, 50, 53–55, 85).

[Jan+15b] 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*. (Atlanta, GA, USA). IUI ’15. 2015, pp. 252–262 (cit. on p. 86).

[Jan+15c] Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. "Beyond "Hitting the Hits": Generating Coherent Music Playlist Continuations with the Right Tracks". In: *Proceedings of the 2015 ACM Conference on Recommender Systems*. (Vienna, Austria). RecSys '15. 2015, pp. 187–194 (cit. on p. 86).

[Jan+15d] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. "Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation". In: *i-com*. Vol. 14. 1. 2015, pp. 29–39 (cit. on p. 86).

[Jan+15e] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. "Item Familiarity Effects in User-Centric Evaluations of Recommender Systems". In: *Proceedings of the 2015 ACM Conference on Recommender Systems (Demos and Posters Track)*. (Vienna, Austria). RecSys '15. 2015 (cit. on p. 86).

[Jan+15f] 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 (Dec. 2015), pp. 427–491 (cit. on pp. 6, 8, 29, 56, 57, 60, 85).

[Jan+16a] Dietmar Jannach, Lukas Lerche, and Geoffray Bonnin. "Empfehlungssysteme, automatische Erzeugung von Wiedergabelisten und Musikdatenbanken". In: *Handbuch der Funktionalen Musik*. Ed. by Günther Rötter. Springer (forthcoming), 2016 (cit. on p. 87).

[Jan+16b] Dietmar Jannach and Gediminas Adomavicius. "Recommendations With A Purpose". In: *RecSys 2016*. 2016 (cit. on p. 6).

[Jan+16c] Dietmar Jannach, Paul Resnick, Alexander Tuzhilin, and Markus Zanker. "Recommender Systems – Beyond Matrix Completion". In: *Communications of the ACM* (2016) (cit. on p. 6).

[Jan+16d] Dietmar Jannach, Michael Jugovac, and Lukas Lerche. "Supporting the Design of Machine Learning Workflows with a Recommendation System". In: *ACM Transactions on Interactive Intelligent Systems*. TiiS 6.1 (Feb. 2016), 8:1–8:35 (cit. on p. 87).

[Jan+17a] Dietmar Jannach, Iman Kamehkhosh, and Lukas Lerche. "Leveraging Multi-Dimensional User Models for Personalized Next-Track Music Recommendation". In: *Proceedings of the 32nd ACM Symposium on Applied Computing*. (Morocco, Marrakesh). SAC '17. 2017 (cit. on p. 87).

[Jan+17b] Dietmar Jannach and Lukas Lerche. "Offline Performance vs. Subjective Quality Experience: A Case Study in Video Game Recommendation". In: *Proceedings of the 32nd ACM Symposium on Applied Computing*. (Morocco, Marrakesh). SAC '17. 2017 (cit. on p. 87).

[Jan+17c] Dietmar Jannach, Lukas Lerche, and Markus Zanker. "Recommending Based on Implicit Feedback". In: *Social Information Access*. Ed. by Peter Brusilovsky and Daqing He. Vol. 10100. LNCS. Heidelberg: Springer, 2017. Chap. 14 (cit. on pp. 10, 85).

[Jaw+10] Gawesh Jawaheer, Martin Szomszor, and Patty Kostkova. "Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service". In: *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. (Barcelona, Spain). HetRec '10. 2010, pp. 47–51 (cit. on p. 45).

[Jaw+14] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. "Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback". In: *ACM Transactions on Interactive Intelligent Systems* 4.2 (2014), 8:1–8:26 (cit. on pp. 1, 11, 15, 17, 26).

[Kam+16] Iman Kamehkhosh, Dietmar Jannach, and Lukas Lerche. "Personalized Next-Track Music Recommendation with Multi-dimensional Long-Term Preference Signals". In: *Workshop on Multi-dimensional Information Fusion for User Modeling and Personalization*. (Halifax, NS, Canada). IFUP '16. 2016 (cit. on p. 87).

[Kan+12] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluis Garcia-Pueyo. "Supercharging Recommender Systems Using Taxonomies for Learning User Purchase Behavior". In: *Proceedings of the VLDB Endowment* 5.10 (June 2012), pp. 956–967 (cit. on p. 56).

[Ke+12] Ting Ke, Bing Yang, Ling Zhen, Junyan Tan, Yi Li, and Ling Jing. "Building High-Performance Classifiers Using Positive and Unlabeled Examples for Text Classification". English. In: *Advances in Neural Networks (ISNN '12)*. Vol. 7368. Lecture Notes in Computer Science. Springer, 2012, pp. 187–195 (cit. on p. 41).

[Kel+03] Diane Kelly and Jaime Teevan. "Implicit Feedback for Inferring User Preference: A Bibliography". In: *SIGIR Forum* 37.2 (2003), pp. 18–28 (cit. on pp. 15, 16, 23, 27).

[KG+12] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Multi-relational Matrix Factorization Using Bayesian Personalized Ranking for Social Network Data". In: *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*. (Seattle, Washington, USA). WSDM '12. 2012, pp. 173–182 (cit. on p. 56).

[Kha+09] A Khalili, Chen Wu, and Hamid Aghajan. "Autonomous Learning of User's Preference of Music and Light Services in Smart Home Applications". In: *Behavior Monitoring and Interpretation Workshop at German AI Conference*. 2009 (cit. on p. 25).

[Kir+12] Evan Kirshenbaum, George Forman, and Michael Dugan. "A Live Comparison of Methods for Personalized Article Recommendation at Forbes.com". In: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. (Bristol, UK). ECML/PKDD '12. 2012, pp. 51–66 (cit. on pp. 6, 7).

[Kli+14] Tomás Kliegr and Jaroslav Kuchar. "Orwellian Eye: Video Recommendation with Microsoft Kinect". In: *Proceedings 21st European Conference on Artificial Intelligence*. ECAI/PAIS '14. 2014, pp. 1227–1228 (cit. on p. 25).

[Kob+01] Alfred Kobsa, Jürgen Koenemann, and Wolfgang Pohl. "Personalised Hypermedia Presentation Techniques for Improving Online Customer Relationships". In: *Knowledge Engineering Review* 16.2 (2001), pp. 111–155 (cit. on p. 20).

[Kon+12] Joseph A. Konstan and John Riedl. "Recommender Systems: From Algorithms to User Experience". English. In: *User Modeling and User-Adapted Interaction* 22.1-2 (2012), pp. 101–123 (cit. on p. 62).

[Kon+97] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. "GroupLens: Applying Collaborative Filtering to Usenet News". In: *Communications of the ACM* 40.3 (1997), pp. 77–87 (cit. on p. 20).

[Kor+10] Suzana Kordumova, Ivana Kostadinovska, Mauro Barbieri, Verus Pronk, and Jan Korst. "Personalized Implicit Learning in a Music Recommender System". English. In: *User Modeling, Adaptation, and Personalization*. Vol. 6075. Lecture Notes in Computer Science. Springer, 2010, pp. 351–362 (cit. on pp. 32, 45).

[Kor08] Yehuda Koren. "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (Las Vegas, NV, USA). KDD '08. 2008, pp. 426–434 (cit. on pp. 45–47).

[Lee+08] Tong Queue Lee, Young Park, and Yong-Tae Park. "A Time-based Approach to Effective Recommender Systems Using Implicit Feedback". In: *Expert Systems with Applications* 34.4 (May 2008), pp. 3055–3062 (cit. on p. 32).

[Lee+10] Dongjoo Lee, SungEun Park, Minsuk Kahng, Sangkeun Lee, and Sang-goo Lee. "Exploiting Contextual Information from Event Logs for Personalized Recommendation". In: *Computer and Information Science 2010*. Vol. 317. Studies in Computational Intelligence. Springer, 2010, pp. 121–139 (cit. on p. 23).

[Ler+12] Benjamin Lerchenmüller and Wolfgang Wörndl. "Inference of User Context from GPS Logs for Proactive Recommender Systems". In: *AAAI Workshop Activity Context Representation: Techniques and Languages, AAAI Technical Report WS-12-05*. 2012 (cit. on p. 25).

[Ler+14] Lukas Lerche and Dietmar Jannach. "Using Graded Implicit Feedback for Bayesian Personalized Ranking". In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, CA, USA). RecSys '14. 2014, pp. 353–356 (cit. on pp. 3, 9, 29, 41, 60, 85).

[Ler+16] Lukas Lerche, Dietmar Jannach, and Malte Ludewig. "On the Value of Reminders within E-Commerce Recommendations". In: *Proceedings of the 24st International Conference on User Modeling, Adaptation and Personalization*. (Halifax, NS, Canada). UMAP '16. 2016 (cit. on pp. 4, 5, 9, 85).

[Ler12] Lukas Lerche. "Entwurf und Umsetzung von hybriden Empfehlungssystemen". MA thesis. TU Dortmund University, Department of Computer Science, Chair XIII, 2012 (cit. on p. 86).

[Li+05] Xiao-Li Li and Bing Liu. "Learning from Positive and Unlabeled Examples with Different Data Distributions". English. In: *Machine Learning: ECML '05*. Vol. 3720. Lecture Notes in Computer Science. Springer, 2005, pp. 218–229 (cit. on p. 42).

[Li+08] Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, and Edward Y. Chang. "PFP: Parallel FP-growth for Query Recommendation". In: *Proceedings of the 2008 ACM Conference on Recommender Systems*. (Lausanne, Switzerland). RecSys '08. 2008, pp. 107–114 (cit. on p. 43).

[Lie95] Henry Lieberman. "Letizia: An Agent That Assists Web Browsing". In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. Vol. 1. IJCAI '95. 1995, pp. 924–929 (cit. on p. 20).

[Lin+02] Weiyang Lin, Sergio A. Alvarez, and Carolina Ruiz. "Efficient Adaptive-support Association Rule Mining for Recommender Systems". In: *Data Mining and Knowledge Discovery* 6.1 (2002), pp. 83–105 (cit. on pp. 21, 44).

[Lin+03] Greg Linden, Brent Smith, and Jeremy York. “Amazon.com Recommendations: Item-to-Item Collaborative Filtering”. In: *IEEE Internet Computing* 7.1 (Jan. 2003), pp. 76–80 (cit. on p. 21).

[Lin+13] Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, and Tat-Seng Chua. “Addressing Cold-start in App Recommendation: Latent User Models Constructed from Twitter Followers”. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’13. 2013, pp. 283–292 (cit. on p. 24).

[Lin+14] Kuei-Hong Lin, Kuo-Huang Chung, Kai-Shun Lin, and Jia-Sin Chen. “Face Recognition-aided IPTV Group Recommender with Consideration of Serendipity”. In: *International Journal of Future Computer and Communication* 3.2 (2014), pp. 141–147 (cit. on p. 25).

[Liu+10] Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. “Unifying Explicit and Implicit Feedback for Collaborative Filtering”. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. (Toronto, ON, Canada). CIKM ’10. 2010, pp. 1445–1448 (cit. on p. 45).

[Liu+14] Haishan Liu, Anuj Goyal, Trevor Walker, and Anmol Bhasin. “Improving the Discriminative Power of Inferred Content Information Using Segmented Virtual Profile”. In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, California, USA). RecSys ’14. 2014, pp. 97–104 (cit. on p. 6).

[Liu+15] Yong Liu, Peilin Zhao, Aixin Sun, and Chunyan Miao. “AdaBPR: A Boosting Algorithm for Item Recommendation with Implicit Feedback”. In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. (Buenos Aires, Argentina). IJCAI ’15. 2015 (cit. on p. 56).

[Mar+07] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. “Collaborative Filtering and the Missing at Random Assumption”. In: *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*. (Vancouver, BC, Canada). UAI ’07. 2007, pp. 267–275 (cit. on p. 18).

[Mla96] Dunja Mladenic. *Personal WebWatcher: Design and Implementation, J. Stefan Institute, Slovenia, Technical report IJS-DP-7472*. 1996 (cit. on p. 20).

[Mob+00] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. “Automatic Personalization Based on Web Usage Mining”. In: *Communications of the ACM* 43.8 (2000), pp. 142–151 (cit. on pp. 21, 45).

[Mor+94] Masahiro Morita and Yoichi Shinoda. “Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval”. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’94. 1994, pp. 272–281 (cit. on p. 20).

[Nic97] David M. Nichols. “Implicit Rating and Filtering”. In: *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*. 1997 (cit. on p. 15).

[Oar+98] Douglas W. Oard and Jinmook Kim. “Implicit Feedback for Recommender Systems”. In: *Proceedings of the 1998 AAAI Workshop on Recommender Systems*. (Madison, WI, USA). 1998 (cit. on pp. 15, 16, 19, 27).

[Oar01] Douglas W. Oard. “Modeling Information Content Using Observable Behavior”. In: *Proceedings of the 64th Annual Conference of the American Society for Information Science and Technology*. 2001 (cit. on pp. 15, 16, 23, 24, 26, 27).

[Pal+10] K. Palanivel and R. Sivakumar. “A Study on Implicit Feedback in Multicriteria E-commerce Recommender System”. In: *Journal of Electronic Commerce Research* 11.2 (2010), pp. 140–156 (cit. on p. 23).

[Pal+11] K. Palanivel and R. Sivakumar. “A Study on Multicriteria Recommender System Using Implicit Feedback and Fuzzy Linguistic Approaches”. In: *Recent Trends in Information Technology*. ICRTIT. 2011, pp. 1087–1092 (cit. on p. 23).

[Pan+08] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, et al. “One-class Collaborative Filtering”. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. ICDM ’08. 2008, pp. 502–511 (cit. on pp. 30, 31, 34, 35).

[Pan+13a] Weike Pan and Li Chen. “CoFiSet: Collaborative Filtering via Learning Pairwise Preferences over Item-sets”. In: *Proceedings of the 2013 SIAM International Conference on Data Mining*. 2013. Chap. 19, pp. 180–188. eprint: `http://epubs.siam.org/doi/pdf/10.1137/1.9781611972832.20` (cit. on p. 56).

[Pan+13b] Weike Pan and Li Chen. “GBPR: Group Preference Based Bayesian Personalized Ranking for One-class Collaborative Filtering”. In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. (Beijing, China). IJCAI ’13. 2013, pp. 2691–2697 (cit. on p. 56).

[Pan+15] Weike Pan, Hao Zhong, Congfu Xu, and Zhong Ming. “Adaptive Bayesian Personalized Ranking for Heterogeneous Implicit Feedbacks”. In: *Knowledge-Based Systems* 73.0 (2015), pp. 173–180 (cit. on p. 59).

[Paq+13] Ulrich Paquet and Noam Koenigstein. “One-class Collaborative Filtering with Random Graphs”. In: *Proceedings of the 22nd International Conference on World Wide Web*. (Rio de Janeiro, Brazil). WWW ’13. 2013, pp. 999–1008 (cit. on pp. 31, 36–38).

[Par+09] Kurt Partridge and Bob Price. “Enhancing Mobile Recommender Systems with Activity Inference”. In: *User Modeling, Adaptation, and Personalization*. Vol. 5535. Lecture Notes in Computer Science. Springer, 2009, pp. 307–318 (cit. on p. 25).

[Par+11a] Denis Parra, Alexandros Karatzoglou, Idil Yavuz, and Xavier Amatriain. “Implicit Feedback Recommendation via Implicit-to-Explicit Ordinal Logistic Regression Mapping”. In: *Proceedings of the CARS ’11*. 2011 (cit. on pp. 31–34).

[Par+11b] Denis Parra and Xavier Amatriain. “Walk the Talk: Analyzing the Relation Between Implicit and Explicit Feedback for Preference Elicitation”. In: *Proceedings of the 19th International Conference on User Modeling, Adaptation, and Personalization*. (Girona, Spain). UMAP ’11. 2011, pp. 255–268 (cit. on pp. 32–34).

[PC+14] Romain Picot Clemente, Cécile Bothorel, and Philippe Lenca. “Contextual Recommender System on a Location-Based Social Network for shopping places recommendation using association rules mining”. In: *ACIIDS 2014 : the 6th Asian Conference on Intelligent Information and Database Systems*. Vol. 551 - SCI (Studies in Computational Intelligence). Apr. 2014, pp. 3–13 (cit. on p. 53).

[Pen+11] Marco Pennacchiotti and Siva Gurumurthy. "Investigating Topic Models for Social Media User Recommendation". In: *Proceedings of the 20th International Conference Companion on World Wide Web*. (Hyderabad, India). WWW '11. 2011, pp. 101–102 (cit. on p. 24).

[Per+14] Maria Soledad Pera and Yiu-Kai Ng. "Automating Readers' Advisory to Make Book Recommendations for K-12 Readers". In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, California, USA). RecSys '14. 2014, pp. 9–16 (cit. on p. 6).

[Pil+10] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. "Fast ALS-based Matrix Factorization for Explicit and Implicit Feedback Datasets". In: *Proceedings of the 2010 ACM Conference on Recommender Systems*. (Barcelona, Spain). RecSys '10. 2010, pp. 71–78 (cit. on p. 45).

[Pit+11] Arthur Pitman and Markus Zanker. "An Empirical Study of Extracting Multidimensional Sequential Rules for Personalization and Recommendation in Online Commerce". In: *10. Internationale Tagung Wirtschaftinformatik*. 2011, pp. 180–189 (cit. on p. 45).

[Piz+10] Luiz Pizzato, Thomas Chung, Tomasz Rej, Irena Koprinska, Kalina Yecef, and Judy Kay. "Learning User Preferences in Online Dating". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. ECML-PKDD, Preference Learning Workshop. 2010 (cit. on pp. 33, 34, 49).

[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*. (Montreal, Canada). UAI '09. 2009, pp. 452–461 (cit. on pp. 3, 30, 31, 38, 39, 56, 57).

[Ren+10a] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing Personalized Markov Chains for Next-basket Recommendation". In: *Proceedings of the 19th International Conference on World Wide Web*. (Raleigh, NC, USA). WWW '10. 2010, pp. 811–820 (cit. on pp. 6, 21, 56).

[Ren+10b] Steffen Rendle and Lars Schmidt-Thieme. "Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation". In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*. (New York, NY, USA). WSDM '10. 2010, pp. 81–90 (cit. on p. 56).

[Ren12] Steffen Rendle. "Factorization Machines with libFM". In: *ACM Transactions on Intelligent Systems and Technology* 3.3 (May 2012), 57:1–57:22 (cit. on p. 54).

[Ric+11] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Springer, 2011 (cit. on p. 1).

[Rod+08] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. "Eye-mouse Coordination Patterns on Web Search Results Pages". In: *Extended Abstracts on Human Factors in Computing Systems*. CHI EA '08. 2008, pp. 2997–3002 (cit. on p. 20).

[Rot+10] Maayan Roth, Assaf Ben-David, David Deutscher, Guy Flysher, Ilan Horn, Ari Leichtberg, et al. “Suggesting Friends Using the Implicit Social Graph”. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’10. 2010, pp. 233–242 (cit. on p. 17).

[Sai+13] Alan Said, Jimmy Lin, Alejandro Bellogín, and Arjen de Vries. “A Month in the Life of a Production News Recommender System”. In: *Proceedings of the 2013 Workshop on Living Labs for Information Retrieval Evaluation*. LivingLab ’13. 2013, pp. 7–10 (cit. on p. 4).

[Sai+14a] Alan Said and Alejandro Bellogín. “Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks”. In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, CA, USA). RecSys ’14. 2014, pp. 129–136 (cit. on p. 63).

[Sai+14b] Alan Said, Alejandro Bellogín, Jimmy Lin, and Arjen de Vries. “Do Recommendations Matter?: News Recommendation in Real Life”. In: *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*. CSCW Companion ’14. 2014, pp. 237–240 (cit. on p. 4).

[Sak+97] Hidekazu Sakagami and Tomonari Kamba. “Learning Personal Preferences on Online Newspaper Articles from User Behaviors”. In: *Computer Networks and ISDN Systems* 29.8-13 (1997), pp. 1447–1455 (cit. on p. 20).

[Sar+00] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. “Analysis of Recommendation Algorithms for E-commerce”. In: *Proceedings of the 2nd ACM Conference on Electronic Commerce*. (Minneapolis, MN, USA). EC ’00. 2000, pp. 158–167 (cit. on p. 42).

[Sch+01a] J. Ben Schafer, Joseph A. Konstan, and John Riedl. “E-Commerce Recommendation Applications”. In: *Applications of Data Mining to Electronic Commerce*. Ed. by Ron Kohavi and Foster Provost. Boston, MA: Springer US, 2001, pp. 115–153 (cit. on p. 5).

[Sch+01b] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. “Estimating the Support of a High-Dimensional Distribution”. In: *Neural Computation* 13.7 (July 2001), pp. 1443–1471 (cit. on p. 41).

[Sch+15] Tobias Schnabel, Paul N. Bennett, Susan T. Dumais, and Thorsten Joachims. “Using Shortlists to Support Decision Making and Improve Recommender System Performance”. In: *CoRR* abs/1510.07545 (2015) (cit. on p. 6).

[Sch+99] J. Ben Schafer, Joseph Konstan, and John Riedl. “Recommender Systems in e-Commerce”. In: *Proceedings of the 1st ACM Conference on Electronic Commerce*. EC ’99. 1999, pp. 158–166 (cit. on p. 5).

[Sen+09] Shilad Sen, Jesse Vig, and John Riedl. “Tagommenders: Connecting Users to Items Through Tags”. In: *Proceedings of the 18th International World Wide Web Conference*. (Madrid, Spain). WWW ’09. 2009, pp. 671–680 (cit. on p. 24).

[Sha+05] Guy Shani, David Heckerman, and Ronen I. Brafman. “An MDP-based Recommender System”. In: *Journal of Machine Learning Research* 6 (2005), pp. 1265–1295 (cit. on pp. 22, 51, 52).

[Sha+06] Bracha Shapira, Meirav Taieb-Maimon, and Anny Moskowitz. “Study of the Usefulness of Known and New Implicit Indicators and Their Optimal Combination for Accurate Inference of Users Interests”. In: *Proceedings of the 2006 ACM Symposium on Applied Computing*. (Dijon, France). SAC ’06. 2006, pp. 1118–1119 (cit. on pp. 33, 34).

[Shi+12] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. “CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-more Filtering”. In: *Proceedings of the 2012 ACM Conference on Recommender Systems*. (Dublin, Ireland). RecSys ’12. 2012, pp. 139–146 (cit. on pp. 40, 41).

[Shi+13] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, and Alan Hanjalic. “xCLiMF: Optimizing Expected Reciprocal Rank for Data with Multiple Levels of Relevance”. In: *Proceedings of the 2013 ACM Conference on Recommender Systems*. (Hong Kong, China). RecSys ’13. 2013, pp. 431–434 (cit. on p. 41).

[Soh+08] Timothy Sohn, Kevin A. Li, William G. Griswold, and James D. Hollan. “A Diary Study of Mobile Information Needs”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. (Florence, Italy). CHI ’08. 2008, pp. 433–442 (cit. on p. 25).

[Sre+03] Nathan Srebro, Tommi Jaakkola, et al. “Weighted Low-rank Approximations”. In: *Proceedings of the Twentieth International Conference on Machine Learning*. ICML ’03. 2003, pp. 720–727 (cit. on p. 30).

[Sri+10] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. “Short Text Classification in Twitter to Improve Information Filtering”. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. (Geneva, Switzerland). SIGIR ’10. 2010, pp. 841–842 (cit. on p. 24).

[Ste11] Harald Steck. “Item Popularity and Recommendation Accuracy”. In: *Proceedings of the 2011 ACM Conference on Recommender Systems*. (Chicago, IL, USA). RecSys ’11. 2011, pp. 125–132 (cit. on pp. 6, 7).

[Tap+04] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. “Activity Recognition in the Home Using Simple and Ubiquitous Sensors”. In: *Pervasive Computing*. Vol. 3001. Lecture Notes in Computer Science. Springer, 2004, pp. 158–175 (cit. on p. 25).

[Tav+14] Maryam Tavakol and Ulf Brefeld. “Factored MDPs for Detecting Topics of User Sessions”. In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, CA, USA). RecSys ’14. 2014, pp. 33–40 (cit. on pp. 6, 22, 50–52, 54, 55).

[Vér+15] Douglas Véras, Thiago Prota, Alysson Bispo, Ricardo Prudêncio, and Carlos Ferraz. “A Literature Review of Recommender Systems in the Television Domain”. In: *Expert Systems with Applications* 42.22 (2015), pp. 9046–9076 (cit. on p. 22).

[Wan+10] Jun Wang and Jianhan Zhu. “On Statistical Analysis and Optimization of Information Retrieval Effectiveness Metrics”. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. (Geneva, Switzerland). SIGIR ’10. 2010, pp. 226–233 (cit. on p. 41).

[Wan+12] Sheng Wang, Xiaobo Zhou, Ziqi Wang, and Ming Zhang. "Please Spread: Recommending Tweets for Retweeting with Implicit Feedback". In: *Proceedings of the 2012 Workshop on Data-driven User Behavioral Modelling and Mining from Social Media*. (Maui, Hawaii, USA). DUBMMSM '12. 2012, pp. 19–22 (cit. on pp. 58, 59).

[War+09] Gill Ward, Trevor Hastie, Simon Barry, Jane Elith, and John R. Leathwick. "Presence-only Data and the EM Algorithm". In: *Biometrics* 65.2 (2009), pp. 554–563 (cit. on p. 42).

[Whi+05] Ryen W. White, Ian Ruthven, and Joemon M. Jose. "A Study of Factors Affecting the Utility of Implicit Relevance Feedback". In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 2005, pp. 35–42 (cit. on p. 14).

[Yaj06] Yasutoshi Yajima. "One-class Support Vector Machines for Recommendation Tasks". In: *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. (Singapore). PAKDD '06. 2006, pp. 230–239 (cit. on p. 41).

[Yu+02] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. "PEBL: Positive Example Based Learning for Web Page Classification Using SVM". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (Edmonton, AB, Canada). KDD '02. 2002, pp. 239–248 (cit. on p. 41).

[Yu+04] Zhiwen Yu and Xingshe Zhou. "TV3P: An Adaptive Assistant for Personalized TV". In: *IEEE Transactions on Consumer Electronics* 50.1 (2004), pp. 393–399 (cit. on p. 45).

[Zan+09] Markus Zanker and Markus Jessenitschnig. "Collaborative Feature-Combination Recommender Exploiting Explicit and Implicit User Feedback". In: *Proceedings of the IEEE Conference on Commerce and Enterprise Computing*. CEC '09. July 2009, pp. 49–56 (cit. on p. 47).

[Zha+05] Yi Zhang and Jamie Callan. "Combining Multiple Forms of Evidence While Filtering". In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. (Vancouver, BC, Canada). HLT '05. 2005, pp. 587–595 (cit. on pp. 33, 45).

[Zha+08] Shiwan Zhao, Nan Du, Andreas Nauerz, Xiatian Zhang, Quan Yuan, and Rongyao Fu. "Improved Recommendation Based on Collaborative Tagging Behaviors". In: *Proceedings of the 13th International Conference on Intelligent User Interfaces*. IUI '08. 2008, pp. 413–416 (cit. on p. 24).

[Zha+15] Qi Zhao, Yi Zhang, Daniel Friedman, and Fangfang Tan. "E-commerce Recommendation with Personalized Promotion". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys '15. 2015, pp. 219–226 (cit. on p. 6).

[Zhe+10] Hua Zheng, Dong Wang, Qi Zhang, Hang Li, and Tinghao Yang. "Do Clicks Measure Recommendation Relevancy?: An Empirical User Study". In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10. 2010, pp. 249–252 (cit. on pp. 6, 7).

[Zhe+12] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. “Towards Mobile Intelligence: Learning from GPS History Data for Collaborative Recommendation”. In: *Artificial Intelligence* 184/185 (2012), pp. 17–37 (cit. on p. 25).

[Zib+12] Dávid Zibriczky, Balázs Hidasi, Zoltán Petres, and Domonkos Tikk. “Personalized Recommendation of Linear Content on Interactive TV Platforms: Beating the Cold Start and Noisy Implicit User Feedback”. In: *UMAP ’12 Workshop on TV and Multimedia Personalization*. UMAP ’12. 2012 (cit. on p. 22).

[Zig+06] Philip Zigoris and Yi Zhang. “Bayesian Adaptive User Profiling with Explicit & Implicit Feedback”. In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. (Arlington, VA, USA). CIKM ’06. 2006, pp. 397–404 (cit. on pp. 33, 48, 49).

List of Figures

List of Tables

Publications

This thesis by publication includes the following four works of author that are closely related to the evaluation of recommender systems, implicit feedback and e-commerce. The full texts of these publications are appended after this list.

- 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 (Dec. 2015), pp. 427–491

- Lukas Lerche and Dietmar Jannach. “Using Graded Implicit Feedback for Bayesian Personalized Ranking”. In: *Proceedings of the 2014 ACM Conference on Recommender Systems*. (Foster City, Silicon Valley, CA, USA). RecSys ’14. 2014, pp. 353–356

- Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Adaptation and Evaluation of Recommendations for Short-term Shopping Goals”. In: *Proceedings of the 2015 ACM Conference on Recommender Systems*. (Vienna, Austria). RecSys ’15. 2015, pp. 211–218

- Lukas Lerche, Dietmar Jannach, and Malte Ludewig. “On the Value of Reminders within E-Commerce Recommendations”. In: *Proceedings of the 24st International Conference on User Modeling, Adaptation and Personalization*. (Halifax, NS, Canada). UMAP ’16. 2016

In addition, this thesis is also based on parts of the following publication that has yet to appear.

- Dietmar Jannach, Lukas Lerche, and Markus Zanker. “Recommending Based on Implicit Feedback”. In: *Social Information Access*. Ed. by Peter Brusilovsky and Daqing He. Vol. 10100. LNCS. Heidelberg: Springer, 2017. Chap. 14

Apart from these five main publications, the author has worked on related topics in the field of recommender systems that are not in the focus of this thesis. These publications are listed below.

- Lukas Lerche. “Entwurf und Umsetzung von hybriden Empfehlungssystemen”. MA thesis. TU Dortmund University, Department of Computer Science, Chair XIII, 2012

- Dietmar Jannach and Lukas Lerche. “Perspektiven in der Offline-Evaluation von Empfehlungsalgorithmen”. In: *HMD Praxis der Wirtschaftsinformatik* 50.5 (2013), pp. 34–44

- Dietmar Jannach, Lukas Lerche, Fatih Gedikli, and Geoffray Bonnin. “What Recommenders Recommend – An Analysis of Accuracy, Popularity, and Sales Diversity Effects”. In: *Proceedings of the 21st International Conference on User Modeling, Adaptation and Personalization*. UMAP ’13. 2013, pp. 25–37

- Dietmar Jannach, Lukas Lerche, and Matthäus Gdaniec. “Re-ranking Recommendations Based on Predicted Short-term Interests–A Protocol and First Experiment”. In: *Proceedings of the Workshop Intelligent Techniques for Web Personalization and Recommender Systems at AAAI 2013*. ITWP ’13. 2013

- 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*. (Atlanta, GA, USA). IUI ’15. 2015, pp. 252–262

- Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. “Beyond “Hitting the Hits”: Generating Coherent Music Playlist Continuations with the Right Tracks”. In: *Proceedings of the 2015 ACM Conference on Recommender Systems*. (Vienna, Austria). RecSys ’15. 2015, pp. 187–194

- Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Item Familiarity Effects in User-Centric Evaluations of Recommender Systems”. In: *Proceedings of the 2015 ACM Conference on Recommender Systems (Demos and Posters Track)*. (Vienna, Austria). RecSys ’15. 2015

- Dietmar Jannach, Lukas Lerche, and Michael Jugovac. “Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation”. In: *i-com*. Vol. 14. 1. 2015, pp. 29–39

- Dietmar Jannach, Michael Jugovac, and Lukas Lerche. “Supporting the Design of Machine Learning Workflows with a Recommendation System”. In: *ACM Transactions on Interactive Intelligent Systems*. TiiS 6.1 (Feb. 2016), 8:1–8:35

- Iman Kamehkhosh, Dietmar Jannach, and Lukas Lerche. “Personalized Next-Track Music Recommendation with Multi-dimensional Long-Term Preference Signals”. In: *Workshop on Multi-dimensional Information Fusion for User Modeling and Personalization*. (Halifax, NS, Canada). IFUP ’16. 2016

- Dietmar Jannach, Lukas Lerche, and Geoffray Bonnin. “Empfehlungssysteme, automatische Erzeugung von Wiedergabelisten und Musikdatenbanken”. In: *Handbuch der Funktionalen Musik*. Ed. by Günther Rötter. Springer (forthcoming), 2016

- Dietmar Jannach, Iman Kamehkhosh, and Lukas Lerche. “Leveraging Multi-Dimensional User Models for Personalized Next-Track Music Recommendation”. In: *Proceedings of the 32nd ACM Symposium on Applied Computing*. (Morocco, Marrakesh). SAC ’17. 2017

- Dietmar Jannach and Lukas Lerche. “Offline Performance vs. Subjective Quality Experience: A Case Study in Video Game Recommendation”. In: *Proceedings of the 32nd ACM Symposium on Applied Computing*. (Morocco, Marrakesh). SAC ’17. 2017

What recommenders recommend: An analysis of recommendation biases and possible countermeasures

[Placeholder]

Dietmar Jannach
TU Dortmund, Germany
dietmar.jannach@tu-dortmund.de

Lukas Lerche
TU Dortmund, Germany
lukas.lerche@tu-dortmund.de

Iman Kamehkhosh
TU Dortmund, Germany
iman.kamehkhosh@tu-dortmund.de

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

User Modeling and User-Adapted Interaction, 25.5, December 2015

DOI: http://dx.doi.org/10.1007/s11257-015-9165-3

Using Graded Implicit Feedback for Bayesian Personalized Ranking

[Placeholder]

Lukas Lerche
TU Dortmund, Germany
lukas.lerche@tu-dortmund.de

Dietmar Jannach
TU Dortmund, Germany
dietmar.jannach@tu-dortmund.de

RecSys '14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA
DOI: http://dx.doi.org/10.1145/2645710.2645759

Adaptation and Evaluation of Recommendations for Short-term Shopping Goals

[Placeholder]

Dietmar Jannach
TU Dortmund, Germany
dietmar.jannach@
tu-dortmund.de

Lukas Lerche
TU Dortmund, Germany
lukas.lerche@
tu-dortmund.de

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

RecSys '15, September 16–20, 2015, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2792838.2800176

On the Value of Reminders within E-Commerce Recommendations

[Placeholder]

Lukas Lerche
TU Dortmund, Germany
lukas.lerche@tu-dortmund.de

Dietmar Jannach
TU Dortmund, Germany
dietmar.jannach@tu-dortmund.de

Malte Ludewig
TU Dortmund, Germany
malte.ludewig@tu-dortmund.de

UMAP '16, July 13–17, 2016, Halifax, NS, Canada

DOI: http://dx.doi.org/10.1145/2930238.2930244