Department of Computer Science
Series of Publications A
Report A-2025-8

Methodological and Extrinsic Challenges in Offline Evaluation of Recommender Systems

Yang Liu

Doctoral dissertation, to be presented for public examination with the permission of the Faculty of Science of the University of Helsinki in Auditorium Tekla Hultin (F3003), Main Building, on 11th of August 2025 at 12 o'clock noon.

University of Helsinki
Finland

Supervisors
Dorota Głowacka, University of Helsinki, Finland
Alan Medlar, University of Helsinki, Finland

Pre-examiners
Shlomo Berkovsky, Macquarie University, Australia
Orland Hoeber, University of Regina, Canada

Opponent
Joseph Konstan, University of Minnesota, The United States

Custos
Dorota Głowacka, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: http://cs.helsinki.fi/
Telephone: +358 2941 911

ISSN 1238-8645 (print)
ISSN 2814-4031 (online)
ISBN 978-952-84-1349-3 (paperback)
ISBN 978-952-84-1350-9 (PDF)
Helsinki 2025
Unigrafia

Methodological and Extrinsic Challenges in Offline Evaluation of Recommender Systems

Yang Liu

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
yang.liu@helsinki.fi
https://researchportal.helsinki.fi/en/persons/yang-liu-2

PhD Thesis, Series of Publications A, Report A-2025-8
Helsinki, August 2025, 84+49 pages
ISSN 1238-8645 (print)
ISSN 2814-4031 (online)
ISBN 978-952-84-1349-3 (paperback)
ISBN 978-952-84-1350-9 (PDF)

Abstract

Recent studies in recommender systems have shown that performance improvements achieved by state-of-the-art methods could be attributed to the misapplication of offline evaluation protocols. These results highlight the potential for both methodological and extrinsic aspects of evaluation to impact results. Furthermore, specialised recommendation tasks, such as sequential and cross-domain recommendation, additionally pose unique and underexplored challenges for evaluation.

In this thesis, we present a series of studies examining different aspects of evaluation in top-n recommendation, sequential recommendation, and cross-domain recommendation. The first study examined sampled evaluation metrics in top-n recommendation. Our results demonstrate greater consistency between sampled and traditional (non-sampled) metrics compared to prior studies, as well as additional advantages, such as the potential for higher discriminative power and robustness against popularity bias in sampled metrics. The second study looked at how the most widely-used data splitting strategy in offline evaluation of sequential recommenders is deeply flawed due to data leakage, which results in performance being significantly overestimated. Our third study investigates user perceptions of cross-domain recommendations. The results show that simply telling users recommendations were based on information from a different domain

significantly altered their perceptions of recommendations, lowering both trust and interest. In the final study, we propose a novel meta-evaluation framework based on techniques from psychometric assessment that can be used to investigate various aspects of offline evaluation, including evaluation metrics, the information content of data sets and the role of item popularity and user engagement in recommendation.

This thesis contributes to our understanding of a diverse range of evaluation challenges in recommender systems. The findings raise critical concerns regarding the evaluation of recommender systems, showing that standard evaluation methods can result in questionable research findings. Additionally, this thesis contains insights into how evaluation can be improved for several recommendation tasks.

Computing Reviews (2012) Categories and Subject Descriptors:

Information systems → Information retrieval → Retrieval tasks and goals → Recommender systems
Information systems → Information retrieval → Evaluation of retrieval results → Retrieval effectiveness

General Terms:
Evaluation, Recommender Systems, Experiments

Additional Key Words and Phrases:
Offline Evaluation, Top-N Recommendation, Sequential Recommendation, Cross-domain Recommendation, Meta-evaluation, Item Response Theory, User Study

Acknowledgements

I was very lucky to be offered a PhD position right after finishing my Master's in the same lab, and for that, I deeply thank my supervisor, Associate Professor Dorota Głowacka, for her trust in me and for supporting my work through her research funding. I am especially grateful for the freedom she gave me to explore different ideas without ever making me feel pressured or rushed. She was with me at many conferences, and honestly, her presence made me feel much more comfortable and confident.

I am equally thankful to my co-supervisor, Dr. Alan Medlar, who has been both a thoughtful mentor and a close friend. He made me realize that research is not just about publishing papers, taught me how to be a responsible researcher, and offered unlimited help whenever I needed it. I truly appreciate his honesty and openness in our collaboration, which I deeply value and hope to carry with me in future work.

I would also like to thank the two pre-examiners, Professor Shlomo Berkovsky and Professor Orland Hoeber, for recognizing the value of this work and for providing valuable and constructive feedback. Thanks as well to all my lab mates and co-authors for their contributions and support, and especially Dr. Denis Kotkov for the many helpful discussions and encouragement along the way.

I would like to thank my family for their constant love and support, especially my parents, for their understanding and for supporting me in pursuing my dreams. I am also sincerely thankful to my friends for chatting with me and making me feel less alone in a foreign country. I especially want to thank Jing and Cokay, for still being my friends, even though I keep flaking on our plans because of deadlines.

Lastly, my beloved cat, MyMy, for the comfort and companionship that made the long Finnish winters less cold and dark.

Helsinki, June 2025
Yang Liu

List of Publications

This thesis is based on the four original articles listed below:

I. **Yang Liu**, Alan Medlar, and Dorota Glowacka. "On the consistency, discriminative power and robustness of sampled metrics in offline top-n recommender system evaluation". In *Proceedings of the 17th ACM Conference on Recommendation Systems (RecSys)*, pp. 1152-1157. 2023.

II. Huy Hoang Le, **Yang Liu**, Dorota Glowacka and Alan Medlar. "Flood Warning: Data Leakage in Offline Evaluation of Sequential Recommenders". UNDER REVIEW.

III. Denis Kotkov, Alan Medlar, **Yang Liu**, and Dorota Glowacka. "On the negative perception of cross-domain recommendations and explanations". In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 2102-2113. 2024.

IV. **Yang Liu**, Alan Medlar, and Dorota Glowacka. "What We Evaluate When We Evaluate recommendation systems: Understanding recommendation systems' Performance using Item Response Theory". In *Proceedings of the 17th ACM Conference on recommendation systems (RecSys)*, pp. 658-670. 2023.

In addition to the aforementioned articles, the author of this thesis has also published the following works:

1. **Yang Liu**, Alan Medlar, and Dorota Glowacka. "Can language models identify Wikipedia articles with readability and style issues?" In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR)*, pp. 113-117. 2021.

2. **Yang Liu**, Alan Medlar, and Dorota Glowacka. "Statistically significant detection of semantic shifts using contextual word embeddings". In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP)*, pp. 104-113. 2021.

3. **Yang Liu**, Alan Medlar, and Dorota Glowacka. "Lexical ambiguity detection in professional discourse". In *Information Processing & Management (IPM)*, 59(5):103000. 2022.

4. **Yang Liu**, Alan Medlar, and Dorota Glowacka. "ROGUE: A System for Exploratory Search of GANs". In *Proceedings of the 45th international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 3278-3282. 2022.

5. Alan Medlar, Jing Li, **Yang Liu**, and Dorota Glowacka. "Critiquing-based Modeling of Subjective Preferences". In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization (UMAP)*, pp. 234-242. 2022.

6. **Yang Liu**, Alan Medlar, and Dorota Głowacka. "Sample, Nudge and Rank: Exploiting Interpretable GAN Controls for Exploratory Search". In *Proceedings of the 29th International Conference on Intelligent User Interfaces (IUI)*, pp. 582-596. 2024.

Contents

Chapter 1

Introduction

Humans are informavores who are naturally driven to seek out information [149]. Advances in information technology have made it easier than ever to access vast amounts of information online. Yet, the key challenge lies in the fact that only a small fraction of this information is pertinent to our needs, leading to the problem of information overload. To address this issue, information filtering tools, such as recommender systems that leverage personalized user profiles to eliminate redundant information, have been developed [10].

Recommender systems have become an integral part of our daily lives, influencing many of the digital services and platforms we use. These systems are employed in various domains, such as e-commerce platforms like Amazon for product suggestion [88, 128], entertainment services like Netflix and Spotify for recommending movies and music [12, 59], and even social media sites like Facebook and Twitter for curating personalized feeds [7, 45]. In recent years, there has been a rapid advancement in the development of recommendation algorithms aimed at enhancing user satisfaction on these platforms [47, 76, 85, 128]. This progress highlights the importance of evaluation, indicating a clear need to identify the most effective recommendation algorithms for implementation in real-world systems.

Recommender system performance can be evaluated through either online or offline methods [62, 125]. While online evaluation captures actual user satisfaction in a live environment, offline evaluation that relies on historical interaction data, is more commonly used in empirical studies due to its effectiveness, reproducibility, and ease of implementation. It is important to note that the standard offline evaluation method in recommender systems is adapted from the field of information retrieval (IR), which incorporates the Cranfield paradigm [141] and rank-based metrics to assess the quality of top-N recommendations [11]. This evaluation method has been

widely adopted and plays a crucial role in benchmarking performance and tracking progress in recommender systems.

Offline evaluation in recommender systems presents numerous challenges. One significant issue is the lack of comparability in evaluation results across studies, given that arbitrary evaluation configurations are often applied. Despite the existence of the standard evaluation method mentioned above, many experimental design decisions, such as pre-processing approaches, evaluation metrics, and data splitting strategies, are frequently determined without clear justification [37, 55]. These different choices, however, can significantly influence evaluation outcomes, often leading to flawed conclusions about relative performance of recommender systems [17].

Second, there is a limited understanding of offline evaluation methods and the interpretation of their results. While the top-N evaluation method, borrowed from IR, appears to be a natural fit for recommender systems evaluation, it has been argued that the recommendation tasks and settings, particularly for offline evaluation, may not align with the assumptions commonly made in IR [11]. This raises a critical concern: can the standard evaluation method truly capture the effectiveness of recommender systems? Furthermore, interpreting the evaluation results can be challenging. For instance, different recommendation algorithms may perform very differently on various datasets [55]. While the "no free lunch theorem" offers a simple explanation [2], a deeper understanding would require further investigation into the influence of dataset characteristics on evaluation outcomes.

Third, offline evaluation may fail to accurately reflect the real-world effectiveness of a recommender system. Indeed, relying on pre-collected interaction data and static evaluation metrics, offline evaluation often struggles to capture the evolving dynamics of user preferences and satisfaction. This limitation has been demonstrated in many studies across different domains of recommender systems. For example, a study by Garcin et al. [43] reported significant discrepancies in accuracy between offline and online evaluations in a news recommendation platform. Similarly, Rehorek et al. [109] investigated this mismatch in the context of movie recommendations, further highlighting the challenges of relying solely on offline evaluation to assess recommender system performance.

With these challenges remaining unresolved, recent studies have raised significant concerns regarding the reliability of offline evaluation in recommender systems. An influential study by Dacrema et al. [38] published in 2019 revealed that the reported progress in recommender systems often diminishes upon closer inspection, attributing algorithmic improvements

to the misapplication of offline evaluation practices. Similarly, Bellogin et al. [11] highlighted the presence of inherent statistical biases within the offline evaluation methods, which undermines the accuracy and reliability of the evaluation results. These observations emphasize the pressing need for a comprehensive investigation into the effectiveness and robustness of offline evaluation methods in recommender systems.

Surprisingly, nearly all previous research addressing evaluation concerns has focused on the widely studied top-N recommendation task [11, 17, 23, 37, 38, 55]. Over the past decade, however, many specialised recommendation tasks, such as sequential recommendation and cross-domain recommendation, have gained significant attention in the recommender systems research community, which typically require distinct evaluation practices. For instance, in sequential recommendation, a modified data splitting strategy is necessary to preserve temporal information. It is, therefore, crucial to investigate the unique evaluation challenges associated with these tasks, in order to gain a clearer understanding of the progress made for each specific task and ensure that evaluation methods are effectively adapted.

The primary focus of this thesis is to identify and understand offline evaluation challenges in recommender systems. This can be further explored by examining the methodological and extrinsic challenges of offline evaluation for various recommendation tasks. While **methodological challenges** focus on the issues that arise from the improper design or flawed implementation of evaluation methods, **extrinsic challenges** examine the influence of external factors, primarily user behaviors and perceptions, on evaluation outcomes in the context of recommender systems. This thesis contributes a series of comprehensive studies to investigate different aspects and challenges of offline evaluation in top-N, sequential, and cross-domain recommendation. The practical implications of the thesis include: (i) providing a thorough examination of the effectiveness and robustness of offline evaluation, and (ii) offering a set of valuable insights into evaluation practices in recommender systems.

This chapter provides an overview of this thesis project, focusing on its motivations, objectives, and contributions. Section 1.1 presents the objectives and outlines the research questions addressed in this thesis. Section 1.2 then elaborates on the contributions of the thesis. In Sections 1.3 and 1.4, the author's specific contributions to each research work included are detailed and the organization of the thesis is described, respectively.

1.1 Objectives and Research Questions

This thesis presents a series of studies to build a comprehensive understanding of challenges and flaws associated with offline evaluation for diverse recommendation tasks. The primary objective is to answer the fundamental question: *Does offline evaluation in recommender systems truly measure what it claims to measure?* This can be further divided into three research questions.

RQ1: *How do methodological decisions in offline evaluation impact evaluation outcomes in recommender systems?*

Our investigation into this research question focuses on two recommendation tasks, namely top-N recommendation and sequential recommendation. The motivation stems from the recognition of two unrealistic offline evaluation practices corresponding to these tasks. First, to address the growing computational demands, sampled evaluation metrics have become the standard practice for offline evaluation for top-N recommendation. Nevertheless, this approach does not align with real-world implementation, where recommendations are generated based on the entire list of items [55] or, at the very least, a set of candidate items [22], rather than the randomly sampled or most popular items typically utilized in sampled metric-based evaluation. Second, when evaluating sequential recommender systems, a data splitting strategy known as temporal leave-one-out is widely employed to preserve temporal information during the evaluation [105, 132, 160]. This practice, however, does not reflect the real-world scenario, where only active users who return to the system after a specified timepoint are considered for evaluation. In contrast, offline evaluation with temporal leave-one-out considers both active and inactive users, which can lead to inflated performance estimates, as the recommendation model may leverage future information from active users to predict the current preferences of inactive ones [65, 131]. Given that numerous empirical studies in recommender systems have relied on these evaluation practices to claim their contributions, it is crucial to conduct a thorough analysis to fully understand their impacts on evaluation outcomes.

RQ2: *To what extent do extrinsic challenges contribute to discrepancies between offline evaluation results and real-world performance?*

One notorious issue with offline evaluation in recommender systems is its misalignment with online evaluation [9, 116]. Recommender systems

are inherently user-centric, making external factors related to users, such as their engagement and perceptions, essential in determining the actual effectiveness of a recommender system [76]. These user dynamics, however, are often challenging to be captured in offline evaluation that relies solely on historical interactions and static accuracy metrics. As a result, a system that performs well in offline tests may not necessarily deliver the same level of satisfaction for users in real-world environments. To investigate how these user-relevant factors influence the evaluation of recommender systems, we focus on the cross-domain recommendation task, where information from a relatively dense domain is leveraged to improve recommendation performance on a sparser domain [42]. The reason for selecting this specific recommendation task is two-fold. First, there is a lack of prior studies on understanding user perceptions of cross-domain recommendations, let alone their influence on system evaluation. Second, cross-domain recommendation settings may introduce changes in system design, including modifications to the interface and shifts in user interaction patterns, which are more likely to influence users and give rise to unique challenges distinct from those in other recommendation tasks. Since nearly all studies on cross-domain recommendation have relied on offline evaluation, there is a clear need to understand the impact of extrinsic evaluation challenges in this context to reveal the true progress in this domain.

RQ3: *Can we develop improved methodologies or frameworks to enhance our understanding of the methodological and extrinsic challenges in recommender systems evaluation?*

Our understanding of offline evaluation in recommender systems is somewhat limited. Despite the widespread use of top-N recommendation evaluation, it remains questionable whether this method can accurately capture user preferences. Indeed, recommender systems often face the inherent issue of missing information [11]. A rank-based metric with a value of 0 for a user can be attributed to either 1) the model fails to capture the user's preference, or 2) there is insufficient information to determine the relevance of recommendations. Moreover, it has been observed that the performance of different recommendation algorithms can vary considerably across different datasets [55]. Benchmark datasets in recommender systems are often collected from diverse platforms and domains, each with distinct characteristics. Despite this notable fact, the connections between dataset characteristics and recommender system performance remain unclear. Finally, the core of a recommender system lies in its ability to deliver

personalized experiences [76]. It is, therefore, critical in offline evaluation to capture individual user characteristics and understand their influences on system performance. To deepen our understanding of offline evaluation, we propose a novel methodology based on item response theory (IRT). As a psychometric framework, IRT was traditionally used to examine the relationship between test takers' ability and the characteristics of test items [15]. By applying this analogy to recommender systems, it enables the analysis of the recommender system's performance in relation to the characteristics of the datasets, as well as the influence of individual users and items simultaneously. Furthermore, the established framework of IRT ensures the reliability of the proposed approach, providing a solid solution for meta-evaluation and in-depth analysis of recommender systems evaluation.

1.2 Thesis Contributions

In tackling the three research questions described above, this thesis makes the following contributions, detailed in the four articles referred to hereafter as Articles I-IV.

Articles I and II address RQ1 and investigate methodological challenges in offline recommendation evaluation. Article I conducts a comprehensive study using 52 recommendation algorithms across three benchmark datasets to examine the consistency, discriminative power, and robustness of sampled evaluation metrics for top-N recommendation evaluation. It is shown that sampled metrics exhibit greater consistency with traditional (non-sampled) metrics than previously reported, while offering additional advantages, such as higher discriminative power and enhanced robustness to popularity bias when appropriate sampling configurations are applied. This study illustrates that the optimal sampling configurations are dependent on dataset characteristics as well as the goals of researchers and investigators, highlighting the need for greater transparency in experimental design decisions.

Article II contributes an in-depth investigation into the flaws of sequential recommender systems evaluation. We examined the issue of data leakage induced by the widely-adopted data splitting strategy, temporal leave-one-out, using nine state-of-the-art sequential recommenders and four datasets. The results raise significant concerns about this evaluation practice, showing that the performance of sequential recommenders dramatically drops after eliminating data leakage. More surprisingly, the reported progress in prior studies of sequential recommenders may be questionable,

given that they can even be outperformed by simple general recommenders that do not rely on sequential information. This study also proposes a data leakage prevention strategy for data splitting, namely split-by-timepoint leave-one-out, which is recommended for researchers to ensure the trustworthiness of evaluation results.

In addressing RQ2, Article III contributes a novel study design aimed at understanding how user-centric factors, such as user perceptions, influence the true performance of cross-domain recommendation. This study reveals an interesting fact: simply telling users that the recommendations are generated based on information from another domain significantly alter their perception of the recommender system, causing a decrease in user trust and interest in recommendations. The finding further highlights that ignoring user perceptions in offline evaluation can lead to inflated evaluation results, causing misalignment with online evaluation.

Moving on to RQ3, Article IV proposes a novel meta-evaluation framework by adapting item response theory, a well-established psychometric assessment technique, to recommender systems. This framework enables the examination of three key perspectives of offline evaluation, including recommender's ability, data set characteristics, and the influence of item popularity and user engagement. The results from a comprehensive evaluation study using 52 recommender systems and 3 benchmark datasets highlight that top-N evaluation metrics fail to reflect the high ability of recommender systems to model user preference. Meanwhile, it is found that the top-N recommendations with the most discriminatory power are biased towards lower difficulty items. The study further provides a series of qualitative analyses on how item popularity and user engagement impact system performance. This work demonstrates great potential to influence future research in recommender systems evaluation.

1.3 Author Contributions

Article I: The author of the thesis identified the need to investigate the impact of sampled metrics on offline evaluation in recommender systems, formulated the experimental design, conducted the experiments and analyzed the experimental results presented in the paper. Additionally, the author of the thesis was responsible for drafting and editing the manuscript.

Article II: The author of the thesis contributed to the conceptualization of the study and the experimental design. In addition, the author of the

thesis assisted in analyzing the experimental results as well as writing and editing the manuscript.

Article III: The author of the thesis was involved in the design of the user experiment and assisted in interpreting the results. The author of the thesis also contributed to drafting and refining the manuscript.

Article IV: The author of the thesis co-proposed the idea and addressed the challenges of adapting item response theory into the evaluation of recommender systems. The author of the thesis implemented and conducted the experiments to investigate key aspects of offline evaluation. Besides, the author of the thesis was responsible for drafting and editing the manuscript.

1.4 Structure of the Thesis

Chapter 1 outlines research backgrounds, objectives, and contributions. Chapter 2 provides an overview of recommender systems, with the primary focus on offline evaluation methods, to build the foundation for understanding the studies described in this thesis. In Chapter 3, we present two comprehensive evaluation studies investigating the methodological challenges in offline evaluation for top-N recommendation and sequential recommendation, which addresses RQ1. In relation to RQ2, Chapter 4 examines the impact of extrinsic evaluation challenges in cross-domain recommendation through a user experiment. The meta-evaluation framework developed to enhance the understanding of offline evaluation is presented in Chapter 5, addressing RQ3. At the end of the thesis, Chapter 6 revisits the key findings, offering a deeper discussion of their implications and suggesting directions for future research and exploration.

Chapter 2

Background

This chapter provides essential background on recommender systems necessary for understanding the research questions and studies discussed in this dissertation. We begin by introducing the foundational concepts and core algorithms of recommender systems in Section 2.1. Section 2.2 delves into evaluation methodologies in recommender systems, covering both offline and online approaches. We conclude the chapter with a discussion of the open challenges in recommender system evaluation in Section 2.3.

2.1 Recommender Systems

Recommender systems are information filtering tools invented in the early 1990s to address the growing challenge of information overload [76]. The first known recommender system, Tapestry [46], was introduced by Goldberg et al. to help users manage large volumes of electronic mail through collaborative filtering, a technique that leverages subjective evaluations from users to identify relevant contents. However, in this design, users were required to manually select whose evaluations to consider, which limited the practicality and usability of the system. This limitation was later addressed by researchers from MIT and the University of Minnesota with the introduction of GroupLens [112], a news article recommendation system that automated the process by aggregating collective evaluations/ratings from multiple users. This approach, now known as user-user collaborative filtering, became the foundation for many subsequent systems. Well-known examples include Ringo [127], a system for recommending music artists, and Video Recommender [57], designed for suggesting videos through e-mail and the web. Later, recommender systems began to be developed in the commercial sector alongside the rise of e-commerce in the late 1990s.

Several companies started marketing recommender engines, with one of the most successful being Net Perceptions [122], which partnered with major companies, such as Amazon, Best Buy, and JC Penney to deliver online shopping solutions.

Moving into the 2000s, recommender systems continued to develop significantly in both research and industry. There has been substantial development of recommendation algorithms, significantly expanding the scope of recommender system research. Prior to 2005, recommender systems were primarily based on collaborative filtering with approaches, such as item-item collaborative filtering [119], a dimensionality reduction-based method [120] and others being developed. Subsequently, other types of approaches, such as content-based [49, 111, 114] and knowledge-based methods [62], have been developed. Driven by the $1 million Netflix Prize challenge launched in 2006, many machine learning and data mining researchers entered the field, making substantial improvements in rating prediction accuracy [76]. During this time, matrix factorization [78] gained significant popularity due to its proven effectiveness [62]. It should be noted that, during this period, the focus of recommendation tasks shifted from rating prediction to top-N recommendation, which aims to provide a list of the most relevant items to users [26, 55]. This transformation was driven by the growing use of recommender systems in commercial applications [76], which showed that improving prediction accuracy did not necessarily guarantee better user experiences [11, 23, 55].

Around 2010, there was a growing awareness of the importance of user experience in recommender systems. Systems that offer greater interactivity to users, such as critique-based recommender systems that ask users to provide feedback on recommendations for improvement, were introduced [20]. Konstan and Riedl [76] discussed the evolution of recommender system research shifting the focus from algorthimic performance to user experience. They argued that measuring user experience reveals critical aspects of systems and requires a broader set of measures than those commonly used. In 2011, Pu et al. [106] proposed a unifying user-centric evaluation framework, ResQue, which focuses on various aspects of systems, including usability, usefulness, interface, interaction qualities and user satisfaction. Many user-focused studies in recommender systems have since based their experimental designs on ResQue [72, 107, 158].

Over the past decade, recommender system research has been largely driven by the advancements of deep learning. Around 2016, many neural network-based collaborative filtering approaches were developed to effectively model the latent features of users and items to enhance recommen-

dation quality [28, 159]. Graph neural networks [152] were employed to advance knowledge-based approaches. Subsequently, recurrent neural networks [98] and transformer-based models [140] were applied to sequential recommendation to capture the temporal dynamics of users' historical interactions [146]. Entering the era of large language models, conversational recommender systems introduced a new paradigm, enabling users to interactively engage with the system to support their decision-making process [61]. With the rapid advancement of new algorithms, numerous libraries specifically designed for benchmarking were introduced, such as daisyRec [133], Beta RecSys [100], RecBole [162], to name a few.

Alongside the rapid advancement of recommender systems, the research community has experienced significant expansion over the past two decades. In 2007, the first ACM Recommender Systems Conference (RecSys) was held in Minnesota, organized by Prof. Joseph Konstan, and has since become one of the most important events in the RS community [62]. Many established conferences focusing on data science, information systems, and web technologies have gradually introduced dedicated sections for recommender systems, such as ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), and The Web Conference (WWW). Additionally, two conferences, the ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR) and the International Conference on Intelligent User Interfaces (IUI) that emphasize user-centered design and personalization, have provided valuable insights into the human aspects of recommender systems. Recently, the first journal dedicated to recommender systems, ACM Transactions on Recommender Systems (TORS), was launched in 2023.

In the meantime, education on recommender systems has gained increasing popularity across universities and academic institutions [62]. With the emergence of online learning platforms, certain courses have been made available online, notably the Recommender System Specialization offered on Coursera by the University of Minnesota since 2018 [103]. Several handbooks and textbooks have been published since 2011, the most notable being those by Ricci et al. [113], Jannach et al. [62], and Aggarwal [4].

2.1.1 Recommendation Tasks

Early research in recommender systems focused on two primary tasks [26]: (i) predicting how much a user will like a particular item, referred to as rating prediction [78, 119, 122], and (ii) identifying a set of items that a user is likely to prefer, known as top-N recommendation [26, 121]. By the

late 2000s, it became widely recognized that top-N recommendation aligned more closely with the objectives of many commercial platforms [11, 23, 55], and since then it became the primary focus of the RS research community [11, 76]. More recently, researchers have recognized that users may have different needs at various stages or across different platforms [76], making specialised recommendation tasks essential to address these diverse requirements. For example, cross-domain recommendation have been introduced to improve the experience of new users, leveraging information from one platform or service to enhance their interactions with another [157]. This section provides a brief overview of top-N recommendation, and two specialized recommendation tasks, sequential recommendation and cross-domain recommendation, which are relevant to the studies presented in this thesis.

Top-N Recommendation

Top-N recommendation refers to the task in recommender systems of generating a list of the most relevant items for users. Numerous recommendation algorithms have been proposed over the past decades to address the top-N recommendation task [5, 23, 26, 121]. Formally, given a set of users $U = \{u_1, u_2, ..., u_n\}$ and items $I = \{i_1, i_2, ..., i_m\}$, user preferences are recorded by an interaction matrix R, where each element $r_{u,i} \in R$ denotes the rating of item i given by user u [130]. $r_{u,i}$ is usually measured on a 5-point Likert scale like in the MovieLens datasets [50]. For each user u, the set of items that have been rated is denoted as positive examples $\in I_u$, while the remaining non-rated items are considered to be negative examples $\in I \backslash I_u$. The goal of top-N recommendation is to identify a set of N items for each user that maximizes the likelihood of positive interactions to appear within that set.

Sequential Recommendation

Sequential recommendation, also known as next-item prediction, focuses on modeling the sequential patterns in users' historical interactions to predict the next item they are likely to prefer. This task is particularly effective in improving user experience on e-commerce platforms, such as Amazon, where the goal is to recommend the next item a user is highly likely to purchase based on their recent buying history [144]. Since its introduction in 2015, the number of studies on sequential recommendation has increased significantly every year [35]. Early approaches to sequential recommendation used Markov Chain (MC) to model sequential user behavior [111, 126]. Subsequently, many deep learning techniques have been applied to sequen-

tial recommender systems. Recurrent neural networks (RNNs) are used to capture long-term user interests [135], while transformer-based models, such as SASRec [66] and BERT4Rec [132], leverage the attention mechanism to capture relationships between items in the sequence.

Cross-domain Recommendation

Cross-domain recommendation was introduced in the early 2010s to overcome the limitations of single-domain recommender systems, such as data sparsity and the cold-start problem [42]. The core idea is to leverage information from a denser source domain to improve recommendations in a sparser target domain. Indeed, the development of cross-domain recommendation (CDR) algorithms largely depends on the overlap of users and items between the source and target domains. Collective matrix factorization leverages cross-domain knowledge derived from overlapping users and items to constrain the factorization process in both domains [108, 155, 157]. Another line of approaches, embedding and mapping, combine latent representation techniques with mapping functions to capture the transferable patterns across the two domains [41, 92, 165]. When there is no overlap in either users or items, clustering-based approaches are used to capture similarities in rating patterns [84, 101] or auxiliary information, such as tag correlations [33, 36] is utilized to infer relationships between domains. Despite this extensive exploration of CDR algorithms in recent years, surprisingly little effort has been made to understand user perceptions of CDR systems [70, 157, 164].

2.1.2 Recommendation Algorithms

Despite various categorizations of recommendation algorithms, this section adopts a more straightforward approach by dividing them into two categories: 1) traditional recommendation algorithms, and 2) neural-based algorithms, based on whether they incorporate neural networks [161]. Neural-based algorithms take the advantages of deep neural network architectures to enhance their capacity for modeling complex user preferences [159, 161]. However, recent studies have shown that, with careful hyperparameter tuning, traditional recommendation algorithms can outperform the state-of-the-art neural-based approaches [37, 38].

Traditional Recommendation Algorithms

Traditional recommendation algorithms typically include three types: collaborative filtering, content-based approaches, and hybrid approaches [3].

Collaborative filtering is based on the assumption that users with similar preferences tend to consume similar items [26, 76, 119]. This approach initially relied on explicit feedback from users to predict ratings. However, it has now shifted towards utilizing implicit feedback to rank items to generate recommendations [119]. Two primary collaborative filtering approaches have been developed: the nearest-neighbor approach and matrix factorization (MF) [11]. Within the nearest-neighbor approach, UserKNN [75] and ItemKNN [26] recommend items based on preferences of users with similar tastes or by finding items similar to those already liked by a given user [161]. Matrix factorization, proposed by Koren et al. [78], involves mapping users and items into a latent factor space. This method reconstructs the interaction matrix by multiplying latent factors to capture the underlying patterns in user-item interactions. Later, Rendle et al. [110] introduced Bayesian Personalized Ranking (BPR), which is based on pairwise user preferences and involves sampling negative items. This model provides a framework for incorporating implicit feedback into top-N recommendation algorithms, including MF.

Content-based approaches leverage auxiliary information, such as user profile and item attributes (e.g., genre and leading characters in movies), to generate recommendations [159]. More concretely, a content-based approach constructs a user profile as a weighted vector of attributes derived from the items the user interacts with [3]. The relevance score between users and unseen items can, therefore, be assessed by measuring the similarity between the user profile and item vectors, for example, using cosine similarity. To construct user and item vectors, Lang proposed using term frequency-inverse document frequency (TF-IDF) to compute the weights of each attribute extracted from news articles to be recommended [3, 82]. In contrast, Pazzani and Billsus [104] proposed a Bayesian model to predict the probability that a user would be interested in a web page given the attributes of that page.

Hybrid approaches combine the collaborative filtering and content-based methods, often leading to improved performance in recommendation. A simple method for combining these approaches involves rating aggregation or voting schemes, where predictions from two or more independently

trained models are integrated [62]. Furthermore, contextual information can be integrated with MF to enhance recommendation performance. Popescul and Pennock [124] developed a unifying probabilistic framework that combines both approaches, effectively addressing the cold-start problem with items.

Neural-based Algorithms

Incorporating deep learning (DL) architectures into traditional recommendation algorithms offers several benefits. Firstly, deep learning methods benefit content-based recommender systems by effectively leveraging auxiliary data. When various types of data, such as text, images and metadata, are available, DL techniques can utilize this information to enhance the recommendation process [159]. Moreover, the capacity of deep learning models to solve challenges across different domains facilitates the development of recommender systems tailored to specific recommendation tasks. For example, recurrent neural networks (RNNs) [98] are highly effective in modeling sequential data, making them a natural solution for sequential recommendation [56]. Similarly, Graph Neural Networks (GNNs) [152] offer a valid approach for incorporating additional contextual information into recommender systems through modeling relationships within, for example, social networks or knowledge graphs [151]. Below, we briefly introduce several notable prior works on the application of DL models in recommender systems. We refer the readers to a more detailed survey by Zhang et al. [159] for further information.

Many attempts have been made to integrate neural networks into collaborative filtering. Dziugaite et al. [29] introduced Neural Network Matrix Factorization (NNMF) by replacing the inner product in MF with a multilayer perceptron (MLP) for interaction reconstruction, enabling the model to learn arbitrary functions from the data. Another well-known model, Neural Collaborative Filtering (NCF), proposed by He et al. [53], follows a similar concept but differs from NNMF in its approach to modeling user-item interactions. More concretely, while Dziugaite et al. [29] focused on minimizing prediction error for explicit ratings, He et al. [53] concentrated on modeling implicit data. In their subsequent work, He et al. [52] proposed augmenting NCF with a convolutional layer to capture high-order correlations between user and item embedding dimensions, resulting in a model known as ConvNCF. Liang et al. [86] proposed MultiVAE, a framework that integrates variational autoencoders (VAEs) into collaborative filtering, enhancing the capabilities of traditional recommenders.

2.2 Evaluating Recommender Systems

Recent years have witnessed a rising awareness of the importance of evaluation within the RS community [55, 62, 125, 130, 158]. The goal of evaluation is typically to conduct comparative studies on a set of candidate recommendation algorithms and identify the most suitable one for incorporation into a real-world system [125]. Three key evaluation methodologies have been mostly used in recommender systems evaluation, namely offline evaluation, user studies, and online evaluation. **Offline evaluation** involves comparing recommendation approaches using pre-collected historical interaction data, allowing for controlled and reproducible assessments. **User studies** gather feedback from a small group of users who interact with the system and provide insights into user experiences and perceptions. **Online evaluation** involves testing systems with real users in a live environment, offering direct and practical insights into the system performance.

Compared to user studies and online evaluation, offline evaluation does not require the implementation of a real-world recommendation system, reduces the complexity of finding participants and maintaining users, and has been predominantly used in RS research for benchmarking algorithms [125]. However, studies over the past decade have raised significant concerns about offline evaluation [11, 17, 23, 37, 38], with one of the most notable issues being its misalignment with online evaluation [9, 43, 76]. This section reviews the three evaluation methodologies for recommender systems, with a primary focus on offline evaluation.

2.2.1 Offline Evaluation

Offline evaluation refers to the process of assessing a recommender system's performance using pre-collected datasets. This method has been widely adopted in recommender system studies due to its efficiency, ease of use, and reproducibility [76, 125].

Offline evaluation in RS initially centered on assessing how accurately a recommender system predicts user ratings for items by measuring prediction error [55, 76]. However, it has been shown that this measure is too stringent to assess the utility of recommender systems [23]. The RS community has, therefore, shifted towards adopting methodologies and metrics from the information retrieval field to improve their evaluation solutions [11]. This adaptation includes both the Cranfield paradigm [142], which involves conducting experiments with test collections to evaluate different retrieval approaches, and rank-based evaluation metrics, such as Precision, Recall, and Normalized Discounted Cumulative Gain (NDCG) [63].

Table 2.1: Summary of Notations

U, I	Set of users, set of items
R	Set of ratings
$I_u \subset I$	Set of items rated by u
$T_u \subset I_u$	Set of test items for u
$N_u = I \backslash I_u$	Set of negative items for u
$N_u^s \subset N_u$	Set of sampled negative items for u
$V_u, V_u^+ \subset I_u$	Original and ordered target set for u
$V_u^K \subset V_u^+$	Set of top K items from V_u for u
$V_{u,k}^+ \in V_u^+$	The k-th item recommended for u
$rel(V_{u,k}^+) \in \{0, 1\}$	Relevance of k-th item recommended for u

More concretely, the offline evaluation process for a recommender system can be outlined as follows. Following the definition in Section 2.1.1, given a user u, I_u represents the set of items rated by the user. Offline evaluation in RS typically involves constructing a target item set V_u for a given user u, based on which top-K items are obtained for calculating the evaluation metrics. Typically, the target item set can be composed by merging the test item set T_u and the negative item set N_u, resulting in $V_u = T_u \cup N_u$, where $N_u = I \backslash I_u$. A recommender system that is being evaluated returns scores that indicate the likelihood of the user liking each item in V_u. These scores are then used to rank items, producing an ordered list of recommendations V_u^+. With a pre-defined cutoff K, the ordered list V_u^+ is truncated to obtain V_u^K, which contains the top-K recommended items in ranked order. Finally, rank-based metrics are applied to the top-K recommendation list to assess its quality and the overall performance of the recommender system is typically obtained by taking the average across all users. The notations used in this thesis are summarized in Table 2.1.

Accordingly, the experimental design of offline evaluation involves three key aspects: (1) benchmark datasets, (2) data pre-processing, and (3) evaluation metrics. Despite the existence of the unified evaluation framework, there has been limited consistency in these experimental settings across studies. For instance, Sun et al. [134] highlighted that given the availability of many datasets for recommender system evaluation, their selection

is often arbitrary. In the study by Zhao et al. [160], four data splitting strategies commonly used in offline evaluation were summarized. Valcarce et al. [138] identified that the use of different metrics with varying cutoffs in evaluation has made results across studies incomparable. A more critical issue, frequently highlighted in several studies, is the insufficient reporting of experimental details in research papers [17, 37, 134, 161]. In this section, we examine the three aspects of the experimental design for recommender systems evaluation.

Benchmark Datasets

The effectiveness of recommendation algorithms has been greatly improved by the increased availability of benchmark datasets since the late 1990s. In 1998, the GroupLens research group released the first benchmark dataset, the MovieLens dataset, which contains 100K user ratings for movies, for recommender system evaluation [50]. Later, to meet the increasing demand for more complex recommendation models, larger versions of the MovieLens dataset (containing from 1M to the latest 32M ratings) have been gradually released from 2000 to 2024 and have gained great popularity in the RS research community [50]. Another popular dataset for movie recommendation is the Netflix dataset, which was launched in 2006 as part of the Netflix Prize Challenge [12].

In 2005, Ziegler [166] collected a large-scale dataset consisting of more than 1 million book ratings from Book-Crossing, facilitating studies on book recommender systems. Two datasets, Last.FM [18] and LFM-1b [123] were collected from Last.FM for music recommendation and were published by the GroupLens in 2011 and Schedl in 2016, respectively. With the increasing use of recommender systems in e-commerce, the Amazon datasets, containing a larger volume of product metadata and user reviews across various product categories, were released in 2014 [51]. A larger version of these datasets was subsequently introduced in 2018. The Amazon datasets were published and are maintained by the research group led by Julian McAuley at UC San Diego [51]. Regarding social media recommendation, one notable dataset is the Yelp dataset, which was released through the Yelp Challenge in 2018 [6]. Subsequently, four additional versions containing more user reviews have been released in recent years.

Indeed, the development of these datasets enables researchers to compare recommender systems on a standardized basis, which facilitates the tracking of progress in the field. Table 2.2 presents the statistics for the most frequently used datasets from 2017 to recent years [134]. These datasets span various domains including movies, books, e-commerce, and

Table 2.2: Statistics of Benchmark Datasets in recommender systems. Statistics presented for the Amazon and Yelp datasets specifically correspond to Amazon (2014) and Yelp (2018), respectively.

Dataset	Domain	#Users	#Items	#Interactions	Sparsity (%)
Last.fm [18]	Music	1,892	17,632	92,834	99.7217
Book-Crossing [166]	Book	105,283	340,556	1,149,780	99.9968
Netflix [12]	Movie	480,189	17,770	100,480,507	98.8224
MovieLens-1M [50]	Movie	6,040	3,952	1,000,209	95.8098
MovieLens-10M [50]	Movie	71,567	10,681	10,000,054	98.6918
MovieLens-20M [50]	Movie	138,493	27,278	20,000,263	99.4706
Amz-Music [51]	E-commerce	478,235	266,414	836,006	99.9993
Amz-Books [51]	E-commerce	8,026,324	2,330,066	22,507,155	99.9999
Amz-Electronic [51]	E-commerce	4,201,696	476,002	7,824,482	99.9996
Amz-Clothing [51]	E-commerce	3,117,268	1,136,004	5,748,920	99.9998
Amz-Sports [51]	E-commerce	1,990,521	478,898	3,268,695	99.9997
Yelp [6]	Social Media	1,326,101	174,567	5,261,669	99.9977
Epinions [136]	Social Media	22,166	296,277	922,267	99.9860

social media. The sparsity of a dataset is determined by the ratio of missing entries to the total number of entries in the user-item interaction matrix [62].

Among the datasets, MovieLens-1M is the most widely used, likely due to its lower sparsity and relatively small size [50, 134]. Most datasets provide explicit ratings on a five-point Likert scale, with the exceptions of Last.FM and Book-crossing. Book-crossing uses a 0-10 rating scale, while Last.FM records user listening counts for different artists [18, 166]. Due to the presence of temporal information (i.e. timestamps), MovieLens datasets, Amazon datasets, Yelp, and Last.FM are also commonly used for evaluating sequential recommenders [105].

Data Pre-processing

Data pre-processing in RS typically involves three steps: (1) data cleaning, (2) data transformation, and (3) data splitting [17]. Data cleaning aims to enhance data quality by removing users and items with very few

interactions. Common strategies include 5-core [153] and 10-core filtering [54, 147], which involves removing users and items with fewer than 5 or 10 interactions, respectively.

Data transformation in the context of RS involves converting explicit ratings (e.g., a user rating a movie as 5 stars) into implicit feedback (e.g., user clicks). A typical approach is to convert each entry in the interaction matrix as 0 or 1, indicating whether the user rated the item [53, 54, 110]. It should be noted that a substantial portion of recommender systems are developed based on implicit feedback given their high prevalence and availability in real-world scenarios [53, 60, 110, 159]. Additionally, the RS community has shifted its focus from rating prediction to top-N recommendation tasks [23], meaning that explicit ratings are no longer essential for modeling recommender systems.

Data splitting is employed in recommender systems evaluation to ensure the generalizability of the trained model. Following the established evaluation paradigm in machine learning, a common strategy involves dividing the dataset into three distinct subsets: training, validation and test sets [99, 160]. Specifically, the training set is used to build the recommender system, the validation set is used for hyperparameter tuning, and the test set is utilized for evaluating the system's performance [4]. As user-centered applications, data splitting in RS is typically performed on a per-user basis [17], meaning that the interaction records for each individual user are divided into the three subsets.

Recently, Zhao et al. [160] conducted a thorough investigation into alternative data splitting methods in top-N recommendation. They explored four data-splitting settings considering two aspects: (1) the ordering of interaction records, either temporal or random, and (2) the method of data splitting, either ratio-based or leave-one-out. For general recommender systems, particularly those based on time-insensitive algorithms, it is recommended to use random ordering and ratio-based splitting for more accurate evaluation. In contrast, for sequential recommendations, temporal ordering is preferred. Additionally, leave-one-out splitting is suggested for small datasets to maximize the size of the training set.

Evaluation Metrics

While other aspects, such as novelty, diversity and coverage are important, the primary focus of offline evaluation in recommender systems has been on effectiveness measures [4, 17]. Early studies in recommender systems measured the prediction errors of ratings using metrics, such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) to assess the accuracy

of recommender systems [55, 62]. These metrics, however, have been shown to be unsuitable for assessing rankings, which is a more critical concern for recommender system applications [23, 55, 76]. Consequently, many rank-based metrics widely used in IR have been applied to recommender systems [11, 138].

In 2000, Sarwar et al. [121] suggested to use Precision@K and Recall@K to evaluate top-N recommendation. While Precision@K measures the proportion of recommended items that are relevant, Recall@K assesses the proportion of relevant items that are actually recommended. Their formulas are defined as follows:

$$Precision@K = \frac{1}{|U|} \sum_{u \in U} \frac{|T_u \cap V_u^K|}{K}, \tag{2.1}$$

$$Recall@K = \frac{1}{|U|} \sum_{u \in U} \frac{|T_u \cap V_u^K|}{|T_u|}. \tag{2.2}$$

One problem with both Precision@K and Recall@K is that they did not take positions of items into account. To mitigate this issue, mean average precision (MAP) has gained great popularity for IR system evaluation, and has been shown to have a high discriminative power and stability [93]. MAP measures the quality of the entire ranking by examining how the relevant items are positioned across the target item list. To calculate MAP, we first define average precision for user u as follows:

$$AP_u = \frac{1}{|V_u|} \sum_{k=1}^{|V_u|} \frac{|T_u \cap V_u^k|}{k} \cdot rel(V_{u,k}^+), \tag{2.3}$$

where V_u^k denotes the set of top-k items extracted from V_u^+. Subsequently, MAP is calculated by taking the average of AP_u across all users, resulting in

$$MAP = \frac{1}{|U|} \sum_{u \in U} AP_u. \tag{2.4}$$

MAP can be considered an aggregation of precision at different Recall levels, providing a single-figure measure combining both precision and Recall [93].

Another metric that takes the ranking of items into account is Normalized Discounted Cumulative Gain (NDCG) proposed by Jarvelin and Kekalainen [63] for IR evaluation in 2002. NDCG@K rewards rankings where relevant items are positioned higher. This metric became popular in

recommender systems in the late 2000s [1, 89]. Discounted cumulative gain (DCG) for user u can be calculated as follows:

$$DCG_u@K = \sum_{k=1}^{K} \frac{rel(V_{u,k}^{+})}{\log_2(k+1)}, \tag{2.5}$$

where $rel(V_{u,k}^{+})$ represents the relevance (i.e. either positive or negative) of the k-th item in the ordered target item list V_u^{+} to u. To allow for a fair comparison between different users, NDCG@K is defined as the normalized ratio of DCG@K to its maximum possible value, IDCG@K (short for Ideal DCG), as follows:

$$NDCG@K = \frac{1}{|U|} \sum_{u \in U} \frac{DCG_u@K}{IDCG_u@K}, \tag{2.6}$$

$$IDCG_u@K = \sum_{k=1}^{\min(K,|T_u|)} \frac{1}{\log_2(k+1)}. \tag{2.7}$$

Another evaluation metric worth mentioning is HitRate@K, which is defined as the proportion of users who have at least one relevant item appearing in the recommendation list. The metric is calculated as follows:

$$HitRate@K = \frac{1}{|U|} \sum_{u \in U} \mathbb{I}(|T_u \cap I_u^K| > 0), \tag{2.8}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. The use of this metric in recommender systems was proposed by Karypis [67] in 2001 to evaluate item-based top-N recommendation.

Among the metrics introduced above, NDCG@K and Recall@K are the most popular ones in recent RS research [134]. Despite their widespread use, these metrics, which are directly adapted from IR evaluation, have been shown to be susceptible to statistical biases inherent in recommender systems due to the issue of missing information [11]. Furthermore, the choice of metrics and their corresponding cut-off values is often arbitrary [17, 37]. In 2018, Valcarce et al. [138] conducted a thorough investigation into the robustness and discriminative power of evaluation metrics. They observed that while NDCG offers better discriminative power, Precision demonstrates the highest robustness. Meanwhile, they recommended using deeper cut-offs (e.g. 100) to mitigate statistical bias and to gain greater discriminative power.

2.2.2 User Studies

Since the late 2000s, understanding and evaluating user experience has become an important topic in the RS research community [76]. In this context, user studies provide a valuable solution for researchers to gather feedback, enabling deeper insights into system design and user behavior.

User studies offer several benefits compared to offline and online evaluations. In a user study, user satisfaction can be directly measured, which provides a more accurate reflection of the true effectiveness of recommender systems than offline evaluation [106, 107]. In contrast to online evaluation, which focuses on overall evaluation criteria, such as revenue and click-through rates, user studies provide more detailed insights, such as user behavior, intentions and perceptions [125]. User studies, therefore, allow for the most comprehensive feedback compared to other types of evaluation methods [158].

In a user study, a small group of participants is recruited to perform predesigned tasks using the recommender systems being evaluated [125, 158]. During the experiment, participants' behaviors are observed and recorded to collect quantitative measurements, such as task completion time, click-through rates, user ratings, etc. Qualitative measurements (e.g. user satisfaction) that are often challenging to obtain, can be collected by asking participants to answer questionnaires before, during, and after the task. This section provides a brief overview of the user study design.

Study Design

Similar to offline evaluation, which is based on IR evaluation approaches, user studies in recommender systems also draw inspiration from interactive IR study design [68]. User study design involves multiple design decisions. Depending on the researchers' purpose, the study can take the form of a single-system usability test or involve a comparison between the developed system and a baseline. Meanwhile, a factorial design can be used when researchers aim to investigate multiple factors within a system, as it allows each combination of factors to be examined independently [68]. In user studies, candidate recommender systems can be compared using either a between-subjects or a within-subjects study design [48], where the difference lies in whether each participant engages with only one or all candidate systems. A within-subject design allows for direct comparison of candidate systems and is therefore considered more efficient. In contrast, between-subject design requires more participants to test candidate systems separately, making it less commonly used in practice. Another important

factor in user studies is to counterbalance the order of systems and tasks to control for order effects and avoid potential bias.

One significant challenge in conducting user studies is recruiting participants. Various methods, such as posting advertisements, inviting individuals within the same organization, and sending invitations through mailing lists, have been employed [68]. However, it can sometimes be difficult to access suitable participants due to factors such as limitations of the target demographic and availability of target users for specilaised systems. A modern solution adopted by many researchers is using Mechanical Turk to recruit participants [71]. In many fields conducting user studies, however, a prominent issue is that participants are often recruited through convenience sampling, leading to an overrepresentation of students and researchers [97]. As for the number of participants, a commonly cited reference is approximately 30, a number determined to meet the assumptions required for certain statistical tests [83]. However, the exact number required can vary depending on factors such as the effect size and desired statistical power. As a rule of thumb, smaller effect sizes and higher statistical power require more participants to identify significant differences between tested systems [74].

User studies often involve gathering information through questionnaires, making their design crucial for effective RS evaluation. On one hand, it is important to ensure that the questions are neutral and free from bias towards any of the systems being tested [125]. On the other hand, the questions should effectively measure various aspects of the recommender system, such as usability, usefulness and user satisfaction, to support a comprehensive evaluation [106]. One influential framework, ResQue, proposed by Pu et al. [106], links the characteristics of recommender systems with users' evaluation behaviors to provide guidelines for the development of questionnaires. In brief, ResQue consists of 15 constructs related to four high-level, independent dimensions for user evaluation: perceived system qualities, users' beliefs, subjective attitudes, and users' behavioural intentions. Perceived system qualities refer to users' perception of the objective characteristics of recommender systems, focusing on recommendation quality, interface design, and the system's ability to elicit user preferences. Users' beliefs represent higher-level user perceptions influenced by their perceived system quality. Users' attitudes reflect the overall feelings of users toward the system, while behavioural intentions measure how the system influences users' purchasing decisions. As stated by Pu et al. [106], researchers are encouraged to determine which constructs and questions to include, with the option to add extra questions based on their specific goals.

2.2.3 Online Evaluation

Despite involving real users, online evaluation differs from user studies by having users perform real-world tasks in live settings [125]. Consequently, online evaluations provide the most realistic assessment scenario, as users engage with the system in a natural and self-motivated manner [158].

Online evaluation is typically conducted through A/B testing, also known as online controlled experiments, which can be used to assess interface changes and algorithm enhancements [74]. In an online controlled experiment, users are randomly assigned to different RS configurations [74, 158]. This randomization ensures that user groups are evenly distributed across the test configurations, thereby eliminating bias. Similar to user studies, user behaviors are observed and recorded during the experiment. Hypothesis testing is then employed to identify statistically significant effects of different RS configurations on user behavior. Due to its focus on optimizing user experience and generating immediate business value, A/B testing is extensively applied in large-scale real-world recommender systems, such as Netflix [47] and Amazon [73].

Online evaluation can also be conducted via longitudinal analysis [40], which studies user behavior and metrics over time to understand the long-term effects of changes or interventions. For example, a recent study by Liang and Willemsen [87] investigated the long-term impact of digital nudging on users' exploration behavior and listening preferences over time in music recommendation in a 6-week experiment. However, longitudinal studies are relatively rare in RS research, likely because recommender systems typically focus on users' immediate interests or behaviors, rather than their long-term patterns [39, 87].

2.3 Summary and Open Challenges

This chapter provides an overview of recommender systems research aiming to 1) conceptualize recommender systems and 2) introduce evaluation methodologies for recommender systems. The following observations can be drawn from the review:

- Top-N recommendation has become the predominant approach in RS due to its close alignment with real-world recommendation scenarios.

- Specialised recommendation tasks, such as sequential recommendation and cross-domain recommendation, have been extensively studied in recent years to address diverse requirements of users.

- Deep learning methods have been integrated into recommender systems to enhance their capability to model complex user preferences.
- Recommender systems can be evaluated through three types of experiments: offline evaluation, user studies, and online evaluation.
- Offline evaluation in RS draws inspiration from IR evaluation, with rank-based metrics, such as Recall@K and NDCG@K, being widely adopted to assess the quality of the top recommended items.
- User studies are a powerful tool for collecting qualitative and subjective measurements from users, providing insights into various aspects of system performance.

Despite the valuable contributions of these studies to the robust evaluation of recommender systems, there remains several open challenges.

First, the evolving evaluation practices have not been closely examined. As datasets grow larger, evaluating recommender systems by ranking all items for each user becomes computationally expensive. Consequently, sampled evaluation metrics that use all positive items and a subset of sampled negative items for each user have become widely adopted in recommender systems [161]. Several recent studies, however, have highlighted a mismatch between sampled and traditional non-sampled metrics, raising concerns about the reliability of using sampled metrics for evaluation [16, 24, 80, 161]. Despite these efforts, two main issues still remain. First, these studies are based on a relatively small number of recommendation algorithms (i.e. 3-12), which limits their statistical power to assess the impact of sampled metrics in practice. Moreover, these studies primarily focus on measuring the consistency between sampled and non-sampled metrics, leaving other important properties, such as discriminative power and robustness, unexplored.

The second challenge lies in insufficient examination of adjustments made to evaluation practices for specialised recommender systems. Sequential recommendation has gained significant popularity in recent years [146]. The current evaluation practice for sequential recommendation is built upon the top-N evaluation framework, with an adjusted data-splitting strategy known as temporal leave-one-out, designed to preserve temporal information during the evaluation process. Surprisingly, this adjustment has not been thoroughly examined, despite recent reports highlighting its potential risk of causing data leakage [99, 161]. This raises a crucial question: Can we trust the reported progress in sequential recommender systems, given that the evaluation practice may inadvertently lead to additional vulnerability?

Third, to address various challenges in recommender systems, new system designs have been proposed over the past few decades [42, 61, 132]. However, little effort has been made to evaluate their practical utility with real users. In cross-domain recommendation, in particular, while numerous novel algorithms have been proposed since the 2010s [164], nearly all these studies rely on offline evaluation, which limits our understanding of their true effectiveness in real-world settings. Furthermore, cross-domain recommendation assumes that user information from one domain can be leveraged to enhance their experience in another [157]. To collect information across domains, modifications to interface design or user-system interaction patterns may be required, which could have a significant impact on user experience. Despite being suggested as a promising future direction in many survey papers [42, 157, 164], the influence of cross-domain settings on user perceptions remains underexplored.

In addition, there is a lack of understanding of the evaluation results in recommender systems. Current evaluation practices often output a single metric value as the proxy for a system's utility. While research papers rely on these metrics to highlight the superior performance of their approaches, the differences in metric values are typically small and insufficient to draw meaningful conclusions about relative system performance [16]. Moreover, an algorithm may demonstrate superior performance on certain datasets but not others [55], making it difficult to identify the factors contributing to the performance difference. More importantly, it is often challenging to understand whether improvements or declines in system performance are connected to specific phenomena or patterns in recommender systems.

In summary, there is a pressing need for a comprehensive examination of recommender system evaluation. This thesis conducts a series of evaluation studies to explore various methodological and extrinsic challenges in offline evaluation for top-N, sequential and cross-domain recommendation.

Chapter 3

Methodological Challenges of Offline Evaluation

Offline evaluation plays a crucial role in advancing the development of recommender systems. Early studies evaluated system performance by measuring the prediction error of user ratings; however, this approach has proven insufficient for assessing system utility [17, 55]. Subsequently, researchers sought solutions for evaluating rankings of recommendations and adapted a rank-based evaluation method from IR to measure the quality of top-N recommendation lists in recommender systems [11, 17]. Due to its better alignment with real-world recommendation scenarios [76, 138], where the goal is to identify a list of items to recommend to users, this method has become the standard practice over the past decade.

In recent years, many studies have raised concerns about offline evaluation practices in recommender systems [11, 17, 37, 38, 80]. These studies have primarily focused on identifying methodological challenges, such as the improper design or misapplication of evaluation methods. For instance, Bellogin et al. demonstrated that applying IR evaluation methods to recommender systems introduces significant biases in measuring effectiveness, due to the discrepancies between the evaluation assumptions for IR and recommender systems [11]. In the study by Dacrema et al. [38], it is pointed out that many recommender system studies involve improper evaluation practices, where state-of-the-art algorithms appear to outperform simple heuristic methods simply because the latter were not properly tuned.

Moreover, a well-known issue in recommender system evaluation is the application of arbitrary evaluation practices without proper justification, despite their potential to significantly impact the evaluation outcomes [17]. According to Herlocker et al. [55], "everyone seems to report with the protocol that generated better results". More surprisingly, some of those arbi-

trary practices become the standard after being adopted by many research papers, even though they have not been thoroughly validated. One example of this is sampled evaluation metrics, first introduced by Koren et al. [77] in 2008, which quickly became widespread in the RS research community due to its computational efficiency [16]. However, it was only recently that the practice was examined by Krichene et al. [80], who identified a significant flaw: the evaluation results using sampled metrics deviate substantially from those obtained with traditional non-sampled metrics.

Some offline evaluation practices may be improperly designed, as they, for instance, do not align with real-world conditions. For instance, in sequential recommendation, the data splitting strategy, temporal leave-one-out, was applied to preserve temporal information in evaluation [105, 132, 146]. This practice, which uses the last interaction of each user for evaluation, however, does not reflect the real-world scenario, where only active users who return to the system after a specified timepoint are considered for evaluation [99]. Applying improperly designed practices may introduce additional biases into the evaluation process, leading to inaccurate estimations of system utility.

Considering the issues discussed above, there is a critical need to thoroughly examine the methodological challenges in recommender system evaluation. In this chapter, we conduct two comprehensive evaluation studies examining the standard evaluation methods for top-N recommendation and sequential recommendation, respectively. Study I provides an in-depth analysis of sampled evaluation metrics in top-N recommendation (Article I [90]). It is noteworthy that, although the influence of sampled metrics has been explored in prior studies [16, 80], our investigation is more extensive, covering 52 recommendation algorithms and 3 benchmark datasets. Study II investigates the impact of the standard data splitting strategy, temporal leave-one-out, for the assessment of sequential recommender systems (Article II).

This chapter is organized as follows. Section 3.1 and 3.2 present the motivation, approach, and main findings of the two studies, respectively. We then conclude the chapter by discussing the implications of our results for recommender systems evaluation in Section 3.3.

3.1 Top-N Recommendation Evaluation

The problem of top-N recommendation, which involves identifying a list of N items most relevant to a user, has been a significant focus in recommender systems research for the past two decades [26]. Its evaluation typically

involves ranking a large collection of items for each individual user [80], which can result in increasing computational demands as the volume of datasets grows. Consequently, a more efficient evaluation method based on **sampled metrics**, where all positive items (i.e. rated[1]) and a limited set of sampled negative items (i.e. unrated) are used for the calculation of effectiveness metrics, has become widespread in offline recommendation evaluation [16, 53, 78, 80, 134]. Surprisingly, the sampling strategy and the sample size for negative items have been determined arbitrarily in the literature due to the absence of established guidelines in the RS community [37].

More recently, sampled metrics have been demonstrated to produce inconsistent evaluation results with traditional non-sampled metrics [17, 80]. For instance, the study by Krichene and Rendal [80] initially highlighted the shortcomings of sampled evaluation metrics, concluding that they should be avoided if possible. A subsequent investigation by Canamares and Castells [16], on the other hand, suggested the existence of an optimal sample size that makes trade-off between evaluation consistency and overall discriminative power. Despite interesting explorations, these studies were based on relatively few recommendation models, thereby lacking statistical power for the analysis.

This section presents a comprehensive study using 52 recommendation algorithms and 3 benchmark datasets to examine the sampled metrics from three important aspects, namely consistency, discriminative power, and robustness. We first introduce how sampled metrics are calculated in Section 3.1.1, and briefly describe the desired properties of evaluation metrics for top-N recommendation in Section 3.1.2. The main findings of our study are summarized in Section 3.1.3.

3.1.1 Sampled Evaluation Metrics

The key distinction between the sampled evaluation metrics and traditional rank-based metrics (i.e. non-sampled) introduced in Section 2.2.1 lies in how the target set is constructed. Recall the notations presented in Table 2.1, for a given user u, T_u and N_u denote the test item set and negative item set for that user. For traditional metrics, the target set is composed by merging the two item sets directly, as defined by $V_u = T_u \cup N_u$. In contrast, for sampled metrics, only a set of sampled negative items, denoted by N_u^s, is used to construct the target set, resulting in $V_u = T_u \cup N_u^s$. We hereafter

[1] In this study, we consider the case where recommendation algorithms are trained on implicit feedback from users. Explicit ratings are converted to implicit feedback as 1 or 0, indicating whether an item is rated by a user or not.

refer to the target set in these two settings as *full target* and *sampled target*, respectively.

A sampled target, based on which sampled metrics are calculated, can be parameterized by the *sample size* and the *sampling strategy*. In the literature, sample sizes typically range from 50 to 1,000 [30, 53, 78, 134, 148], with the most common being 100 [80]. Two sampling strategies including *uniform sampling* and *popularity-based sampling* have been employed [16, 80, 134]. While uniform sampling treats all negative items equally, popularity-based sampling weights each item based on the number of ratings it received (i.e. its popularity). For clarity, we hereafter use the term "configuration" to denote the combination of sample size and sampling strategy, employed to determine the sampled metrics.

3.1.2 Properties of Evaluation Metrics

An effective evaluation metric should measure a model's performance accurately. Serving as the proxy for traditional rank-based metrics in reducing computational cost, sampled metrics are expected to yield consistent evaluation results with those produced by non-sampled metrics. This property is referred to as **consistency**. Meanwhile, sampled metrics should have substantial ability to discriminate between the best-performing models, exhibiting strong **discriminative power**. For data-driven models, the evaluation metrics should be capable of delivering consistent results despite the challenge of missing data, i.e. maintain high **robustness**. In this section, we discuss briefly how to measure these three properties corresponding to sampled metrics.

Consistency. To measure the consistency, a common approach is to assess the Kendall'τ correlation coefficient [69] between model rankings obtained with sampled and traditional non-sampled metrics [80, 161]. One consensus in the RS research community is that two rankings yield a Kendall's τ correlation above 0.9 are deemed almost equivalent [138, 141].

Discriminative Power. In the context of RS, the discriminative power of a metric can be evaluated through tie analysis, which involves comparing the performance of a pair of recommenders for each individual user [17]. Given a group of recommenders, the overall discriminative power of a particular metric can be estimated by the average number of ties across all model pairs. In addition to the tie analysis, statistical significance tests on pairwise model performance have also been utilized to measure the dis-

criminative power of evaluation metrics [17, 19, 118, 138]. For instance, Canamares and Castells [17] measured the discriminative power of a given metric by summing P-values derived from statistical tests across all model pairs. This measure, however, lacks interpretability: even if the sum of P-values obtained remains the same, the number of model pairs with statistically significant differences in performance can vary.

Robustness. Prior studies have demonstrated that rank-based metrics are susceptible to the problem of missing relevance judgements in the test set, leading to biased evaluation [14, 138, 156]. Indeed, incompleteness in judgements has been demonstrated to cause two types of biases, *sparsity bias* and *popularity bias*, in recommendation evaluation [11]. While sparsity bias occurs when relevance judgments are missing at random, popularity bias arises because the distribution of missing judgments is non-uniform. To examine the robustness of sampled metrics, we progressively exacerbate the problem of missing data by removing the user-item ratings from the test set, either randomly or according to item popularity, to amplify sparsity bias and popularity bias, respectively. A robust evaluation metric is expected to rank recommenders consistently under these circumstances, suggesting the ranking consistency that is typically measured by Kendall's τ correlation coefficient to be a reasonable measure for robustness [11, 141].

3.1.3 Main Findings

The study covered 52 recommenders to ensure the statistical power of the analyses. Three benchmark datasets, MovieLens-100K, ML-1M, and Amazon-Books were selected due to their different characteristics. I examined the three most widely used rank-based metrics: precision, NDCG, and recall with cutoff values of 5, 10, and 20, respectively.

Consistency. Ranking consistency increases as negative sample sizes go up on all datasets, as shown in Figure 3.1[2]. However, the best performing sampling strategy that achieves the highest ranking correlation is dataset dependent. While uniform sampling yields higher ranking consistency on two MovieLens datasets, Amazon-Books performed best with popularity-based sampling. More interestingly, high ranking consistency can be achieved even with small sample sizes such as 50 and 100, as long as the appropriate sampling strategy is employed.

[2]The results from MovieLens-100K were not shown in the figures presented in this chapter because they were very similar to those of MovieLens-1M.

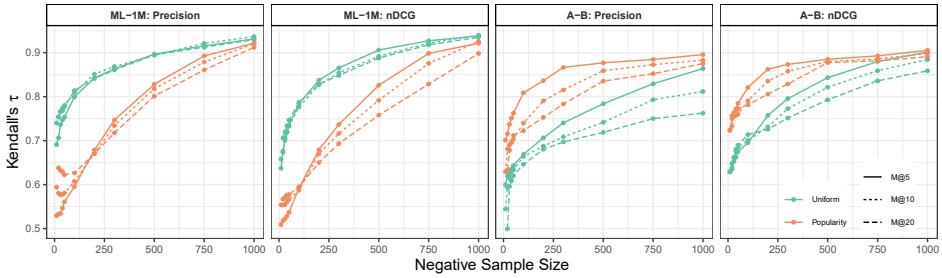

Figure 3.1: The consistency between model rankings for the traditional metrics and sampled metrics with varying negative sample sizes and sampling strategies on MovieLens-1M (denoted by ML-1M) and Amazon-Books (denoted by A-B).

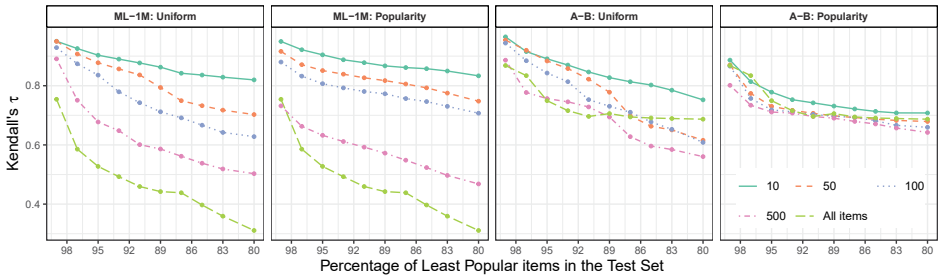

Figure 3.2: The impact of popularity bias on model ranking consistency with varying negative sample sizes and sampling strategies on MovieLens-1M and Amazon-Books (only Precision@10 shown).

Discriminative Power. Both very small and large sample sizes can result in a significant increase in ties, implying a decreased discriminative power (see Figure 2 in Article I). Indeed, while many models achieve comparable high performance with small negative sampling sizes, larger sample sizes introduce greater challenges to the recommendation task, resulting in similar poor performance across different models. A relatively small sample size can lead to a higher discriminative power, given that the lowest number of ties was found with a sample size ≤ 100 around 81% of the time.

Robustness. While larger sample sizes demonstrate greater robustness against sparsity bias, smaller sample sizes exhibit noticeable resilience to popularity bias (see Figure 3.2). However, popularity bias is a major concern compared to sparsity bias as the ranking correlation with high popularity bias can be as low as 0.3 in our experiment (e.g. in MovieLens-1M)

while the correlations with the sparsity bias consistently exceeded ∼0.8 across all the experimental settings.

It should be noted that the main findings related to the three aspects did not yield consistent suggestions regarding sampling configurations. To maintain high consistency with non-sampled metrics, it is suggested to use larger sample sizes; whereas relatively small sample sizes (e.g. 100) are shown to gain increased discriminative power and higher robustness against the popularity bias. These results also highlighted the importance of the selection of sampling strategies, demonstrating that high ranking consistency can be achieved with even small sample sizes like 100 when the best-performing sampling strategy is used on each dataset.

3.2 Sequential Recommendation Evaluation

In recent years, sequential recommendation that focuses on the next-item recommendation task has gained increased popularity in the RS research community [34, 66, 132, 146, 163]. Sequential recommender systems provide personalized recommendations by leveraging sequential dynamics of user's historical interactions, allowing for more accurate prediction of users' future preferences [146]. Furthermore, recent advances in deep learning have enabled the rapid development of sophisticated sequential recommendation models, such as SASRec [66] and BERT4Rec [132].

While much attention has been given to developing novel approaches, few studies have focused on the evaluation of sequential recommenders. Indeed, the evaluation of sequential recommenders largely relies on the traditional top-N recommendation evaluation scheme, while incorporating a temporal data splitting strategy named **temporal leave-one-out (LOO)** to preserve sequential information. Despite its prevalence, prior studies have demonstrated that temporal LOO cannot maintain the temporal consistency between training and test data, causing data leakage in evaluation [64, 65, 99, 134]. These studies were, however, primarily conducted in the context of non-sequential recommendation, for which temporal LOO has rarely been utilized. Moreover, the existing studies have narrowly focused on the impact of adopting temporal LOO on model ranking, neglecting its broader implications.

This section conducts a comprehensive evaluation study to examine the validity of using temporal LOO for the evaluation of sequential recommender systems. The study has two primary objectives. First, we aimed to investigate whether temporal LOO leads to data leakage in sequential

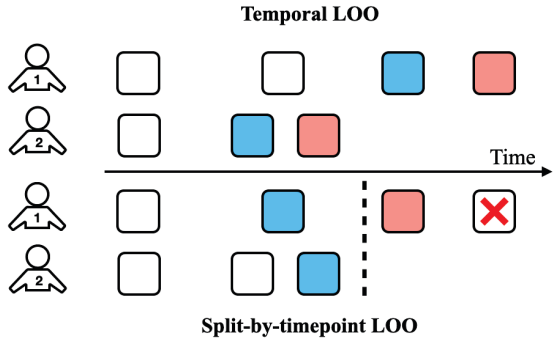

Figure 3.3: Examples of temporal LOO and split-by-timepoint LOO. White, blue, and red boxes represent training, validation, and test user-item interactions, respectively. In split-bytimepoint LOO (bottom), interactions are partitioned based on a global cut-off date across all users (black dashed line). Interactions that come after the test interactions are discarded (crossed box).

recommender systems evaluation. Second, we wanted to understand the influences of data leakage on the evaluation outcomes with temporal LOO. Section 3.2.1 briefly introduces temporal LOO and the issue of data leakage. The main findings of the study are summarized in Section 3.2.2.

3.2.1 Temporal Consistency and Data Leakage

Data leakage describes a situation in which some information used for training would not be available in real-life scenarios [65, 99]. In sequential recommendation, temporal data leakage can arise when the test set contains user-item interactions that occurred prior to some interactions in the training set, as a consequence of using a data splitting strategy that cannot maintain the temporal consistency. This section begins by introducing temporal LOO and illustrating its potential for causing temporal data leakage. We then present split-by-timepoint LOO, a novel temporal data splitting strategy that prevents data leakage by enforcing a global timeline between training and test sets. Figure 3.3 provides an overview of temporal LOO and split-by-timepoint LOO. By comparing the two data splitting strategies in our experiment, we can gain valuable insights into the impact of data leakage in sequential recommender systems.

Temporal LOO

As shown in Figure 3.3 (top), temporal LOO partitions user-item interactions based on a local timeline for each user. More concretely, temporal LOO uses the last interaction of each user for testing and the second to the last for validation. All remaining interactions are assigned to the training set.

Temporal LOO cannot maintain the temporal consistency between the training and test data. Indeed, when predicting the next item for a given user to consume at timepoint t, temporal LOO cannot prevent other users' interactions that occurred after timepoint t from being used in the training process. Those interactions after t are in fact unavailable in real-world recommender systems, which can lead to invalid or highly unlikely recommendation results.

Split-by-timepoint LOO

Split-by-timepoint LOO maintains the temporal consistency between training and test data by enforcing a global timeline, as can be seen in Figure 3.3 (bottom). Given a global cut-off date for all users, we refer to users who have at least one interaction after the timepoint as *active users*; otherwise, they are referred to as *inactive users*. For each active user (see Figure 3.3 (bottom, User 1)), the user-item interaction immediately preceding the cut-off date is used for validation, while the item immediately after it is assigned to the test set. All interactions prior to the cut-off date, but not in the validation set, are treated as training examples, whereas interactions after the cut-off date, but not in the test set, are discarded. For each inactive user (see Figure 3.3 (bottom, user 2)), the last interaction is assigned to the validation set and the remaining interactions are included in the training set. Consequently, inactive users are excluded from evaluation to prevent temporal data leakage into the training process. As for the selection of the global cut-off date, the strategy is to use the date that maximizes the test set size to (i) include a sufficient number of users for evaluation, and (ii) enhance the comparability with temporal LOO.

3.2.2 Main Findings

The study was conducted using 9 state-of-the-art sequential recommendation models and 4 benchmark datasets. We investigated the impact of data leakage in sequential recommender systems by analyzing the performance differences between temporal LOO and split-by-timepoint LOO. The analy-

Table 3.1: Relative difference between temporal LOO and split-by-timepoint LOO (subsamp.= ×), and temporal LOO with subsampled training data and split-by-timepoint LOO (subsamp.=✓).

		Popularity-sampled		Unsampled	
	Subsamp.	NDCG@10	recall@10	NDCG@10	recall@10
ML-1m	×	-73.4%	-64.5%	-90.0%	-87.7%
	✓	-72.2%	-63.7%	-88.9%	-86.5%
Yelp	×	-30.7%	-26.4%	-53.1%	-52.8%
	✓	-23.3%	-22.4%	-14.2%	-24.9%
Steam	×	-21.7%	-22.1%	-16.5%	-22.4%
	✓	-16.7%	-19.4%	-2.6%	-12.9%
Beauty	×	-62.7%	-55.9%	-78.3%	-75.2%
	✓	-41.6%	-39.1%	-69.8%	-67.4%

ses were based on both popularity-sampled and unsampled NDCG@10 and recall@10 due to their prevalence in the literature [105].

Existence of Data Leakage

The experimental results highlighted the existence of data leakage with temporal LOO. Dramatic performance drops in NDCG@10 and recall@10, ranging from 16.5% to 90.0% across sequential recommenders, were identified over four datasets when changing the data splitting strategy from temporal LOO to split-by-timepoint LOO (Table 3.1, subsamp.=×). Two potential factors account for the performance drops: i) the mitigation of data leakage using split-by-timepoint LOO, and ii) the reduced model quality resulting from the smaller training set with split-by-timepoint LOO. We further investigated the impact of training set sizes by randomly subsampling the training set until it was approximately the same size as split-by-timepoint LOO. As illustrated in Table 3.1 (subsamp.=✓), the reduced amount of training data is far from sufficient to explain the performance drops in most of the scenarios, thereby confirming the presence of data leakage. Moreover, we identified similar performance drops, with the active users (i.e. users less affected by data leakage) achieving considerably lower NDCG or recall than inactive users (i.e. users suffered from data leakage) with temporal LOO (see Table 3 in Article II).

Impacts of Data Leakage

Three significant consequences of data leakage by temporal LOO were identified. First, data leakage resulted in the overestimated evaluation results,

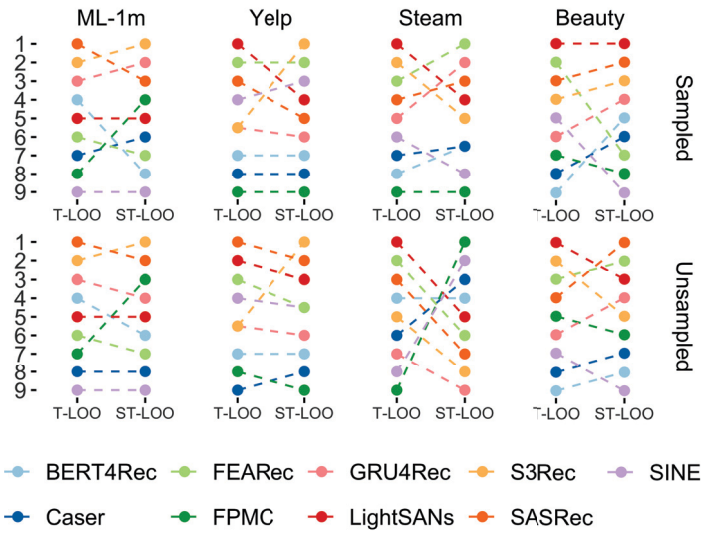

Figure 3.4: Inconsistent model rankings with temporal LOO (T-LOO) and split-by-timepoint LOO (ST-LOO) for popularity-sampled (top) and unsampled (bottom) nDCG@10.

with temporal LOO obtaining considerably higher NDCG and recall compared to split-by-timepoint LOO, irrespective of the model, the dataset, and whether the sampled metrics were used or not. Second, temporal LOO achieved inconsistent evaluation outcomes (i.e. the rankings of sequential recommendation models) with split-by-timepoint LOO, as can be seen in Figure 3.4. Indeed, even the worst-performing recommender with temporal LOO can be the best-performing model with split-by-timepoint LOO (see unsampled/Steam in Figure 3.4). Lastly, data leakage led us to mistakenly believe that sequential recommenders consistently achieved superior performance compared to general recommenders, given that there was a general recommender that outperformed the best-performing sequential recommenders in terms of unsampled NDCG on a majority of datasets with split-by-timepoint LOO (see Table 5 in Article II).

3.3 Discussion

This chapter presented two evaluation studies investigating the methodological challenges associated with offline evaluation of recommender sys-

tems (RQ1). We focused on two primary recommendation tasks, namely top-N recommendation and sequential recommendation.

For top-N recommendation, our study on sampled metrics yielded more positive results compared to earlier reports [80, 160]. These results were characterized by: (i) improved ranking consistency with traditional non-sampled metrics, and (ii) the potential to achieve greater discriminative power and enhanced robustness against popularity bias when appropriate sampling configurations are applied. As for sequential recommenders evaluation, our results showed empirically that temporal LOO was severely flawed due to data leakage. Indeed, the performance of sequential recommenders can decline significantly without data leakage, and can even be surpassed by simple general recommenders.

Based on these findings, the following recommendations are put forward to improve the offline evaluation practices in recommender systems. First, regarding the use of sampled metrics, given the significant impact of sampling configurations on ranking consistency, discriminative power, and robustness, it is crucial for researchers or investigators to make their own experimental decisions to fulfill their specific goals. For instance, they may use relatively small sample sizes to attain higher discriminative power or increased robustness against the popularity bias. Second, it is recommended to replace the problematic data splitting strategy, temporal LOO, with its more realistic variation, split-by-timepoint LOO, to avoid data leakage in sequential recommender systems. Finally, to ensure the robustness and reliability of evaluations, greater transparency in experimental design decisions is strongly encouraged.

Our findings also hold great potential to guide future research on recommender systems evaluation. The study on sampled metrics highlights the importance of identifying the optimal sampling configurations based on the characteristics of the dataset. In fact, our attitude toward sampled metrics is more positive than ever, not only due to the favorable results obtained in this study, but also because the use of sampled metrics appears increasingly unavoidable as datasets in recommender systems continue to grow in size. Therefore, putting more effort into optimizing evaluation configurations will help ensure the efficiency of the evaluation process while maintaining high trustworthiness in the results. Furthermore, the results indicating that sequential recommenders can be outperformed by general recommenders when data leakage is avoided raise important research questions for future exploration. It necessitates an in-depth investigation into the role that sequential information plays in enhancing recommendation performance.

Chapter 4

Extrinsic Challenges of Offline Evaluation

Offline evaluation faces extrinsic challenges due to its inability to account for user dynamics in recommender systems. As recommender systems gain widespread use in real-world applications, evaluating their effectiveness requires measuring user experience and satisfaction with recommendations [76]. Many exogenous factors, such as interface design or system settings, can influence user interactions with and perceptions of recommender systems, and therefore significantly impact the actual system performance. Given its static nature, offline evaluation often fails to account for these impacts, as it cannot effectively capture the evolving behaviors and interests of users. Consequently, approaches that perform well offline may not necessarily achieve equivalent user satisfaction in practice [9].

Indeed, many studies have shown that offline evaluation fails to reflect the true effectiveness of recommender systems. For instance, an early study by Cosley et al. [21] demonstrated that interface design can influence user opinions on items, which in turn, can affect user satisfaction. Ekstrand et al. [31] showed that user-centric aspects that are challenging to capture by offline evaluation, such as novelty, satisfaction, and diversity, can influence the effectiveness of recommender systems. A recent study by Nguyen et al. [102] examined how individual personality traits influence user satisfaction, indicating that overlooking such factors in offline evaluation can lead to inaccurate performance estimates. Moreover, several studies have revealed the misalignment between offline and online evaluation outcomes across various applications, including scientific article recommendation [9], news recommendation [43], and movie recommendation [109]. These findings highlight the importance of understanding extrinsic challenges in offline evaluation in order to gain more comprehensive insights into its reliability.

Recommender systems have advanced significantly in recent years, not only in terms of algorithm development but also through the emergence of specialised systems. For instance, cross-domain recommendation has attracted considerable attention due to its potential to address issues related to cold-start and data sparsity [157]. Cross-domain recommendation is based on the assumption that user interests across different domains are relevant. Therefore, it is possible to leverage information from a source domain to a relatively sparse target domain to generate more accurate recommendations [13]. Despite improvements in recommendation performance [42], cross-domain recommendation poses unique evaluation challenges. For instance, integrating information from different domains can influence interface design and user interaction patterns, potentially amplifying extrinsic challenges in offline evaluation. Unfortunately, there is a lack of research exploring user-relevant factors in cross-domain recommendation and their impacts on the evaluation [70, 157, 164].

More recently, cross-domain recommendations with explainability have been identified as a promising future research direction to improve user perceptions in terms of transparency, persuasiveness and trustworthiness [157]. Similarly, incorporating explanations can influence interface design and significantly impact user experience in the cross-domain setting. This topic, however, has been underexplored in empirical studies [143].

This chapter aims to address these research gaps by investigating how cross-domain settings and recommendation explanations impact user perceptions (Article III [79]). Our study confirms that offline evaluation of cross-domain recommendation faces extrinsic challenges, resulting in a potential overestimation of recommender systems' performance. This chapter is organized as follows: Section 4.1 provides an overview of our user study, highlighting the main findings. Section 4.2 offers a brief discussion of the results and their implications.

4.1 User Perceptions of Cross-Domain Recommendation

This section presents a user study designed to investigate how cross-domain recommendation setting and explanations influence user perceptions of recommendations. We briefly introduces the study design, implementation details, measures employed, and the primary findings of the study. In Section 4.1.4, the primary findings of the user study are outlined.

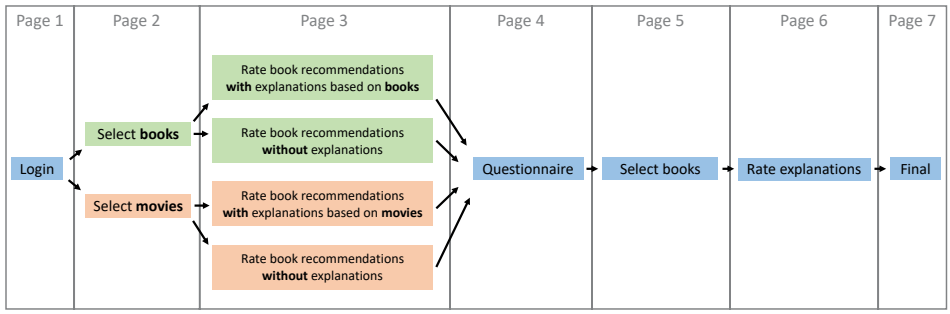

Figure 4.1: Overview of the between-subject study with four experimental conditions. The green boxes correspond to the single-domain recommendation setting with or without explanations; whereas the orange boxes are relevant to the cross-domain recommendation setting.

4.1.1 Study Design

A key consideration for the user study was to disentangle the effects of whether recommendations were single- or cross-domain from whether explanations were present or not. Therefore, the study included four experimental conditions for comparison. To simulate a cross-domain setting, we selected "movies" and "books" as the source and target domains.

We identified three primary experimental confounders that influence the study design. First, each participant must be clear about what information is available for generating recommendations. Our study, therefore, adopted a between-subject design to situate each participant in only one experimental condition. Second, it is essential that the quality of generated recommendations is consistent across all scenarios because otherwise variations in recommendation quality could affect user perceptions, leading to biased results. To achieve this, random recommendations for each participant in both single- and cross-domain settings were generated. A recommendation generation procedure was proposed to ensure that each participant received recommendations of fairly balanced quality (see Section 4.1.2). Finally, it is necessary to ensure that the generated explanations can justify the recommendations credibly. We additionally asked participants to rate the quality of generated explanations related to books that they had read, which served as a sanity check.

With all these considerations in mind, the overview of the user study with four experimental conditions is demonstrated in Figure 4.1. After a participant logged into the system, they were randomly assigned to one of the four experimental scenarios (Page 1-3). The user was then asked to

1) rate the book recommendations generated based on the items that the user liked (Page 3), 2) complete the questionnaire regarding seven aspects of user perceptions (Page 4), and 3) validate the quality of the generated explanations for books that they are familiar with (Page 5-6). As the study aimed to investigate the impact of the cross-domain setting itself (i.e. not specific cross-domain recommendation algorithms), the book recommendations provided to the user were actually random books and were not tailored to user preferences.

4.1.2 Implementation Details

This section briefly introduces the implementation details of the two major components of the study, namely, recommendation generation and explanation construction. The implementation was based on the tag genome datasets for books and movies, which contain relevance scores for various tags assigned to each item (i.e. book or movie).

Generating Recommendations

Random book recommendation lists were generated to ensure that the recommendations provided to the user have consistent quality across different experimental conditions. To create the recommendation lists, we first filtered out the top 25% of popular and unpopular books to avoid extreme cases. Subsequently, a topic diversification algorithm was employed to generate diversified recommendation lists [166]. However, this diversification led to the inclusion of some niche items, which were distinct from all the other books in the majority of the recommendation lists. We eliminated these niche items based on how frequently they appeared in the lists and then re-generated the recommendation lists. Following this procedure, 4,209 diversified recommendation lists were generated with a fairly balanced quality. The book recommendations for each user were generated by randomly sampling one of these pre-generated lists without replacement.

Generating Explanations

We applied tag-based explanations with a contrastive feature showing users both common and uncommon tags of the recommended books and items that they preferred. We first extracted a set of tags that appeared in both tag genome datasets. For each tag in the set, two scores, $score_c$ and $score_u$ were calculated, which measure how common or uncommon a tag is between a user's favourite items and the recommended books:

$$score_c(t, b, i) = rel(t, i) \cdot tfidf(t, b), \tag{4.1}$$

$$score_u(t, b, i) = rel(t, b) - rel(t, i), \tag{4.2}$$

where $rel(t, i)$ and $rel(t, b)$ represent the relevance scores of tag t in the user's preferred item i and book b, respectively. $tfidf(t, b)$ is calculated using the following formula:

$$tfidf(t, b) = rel(t, b) \cdot log\left(\frac{|B|}{\sum_{b \in B} rel(t, b)}\right), \tag{4.3}$$

which measures the importance of tag t to book b in a list of books B, following a similar concept of term frequency-inverse document frequency in natural language processing.

4.1.3 Measures

The study aims to assess how cross-domain recommendations and explanations affect perception and behavioural intentions of users. **Behavioural intentions**, namely users' interest in the recommendations they received, can be measured by the ratings given by each user directly. Seven aspects related to **user perceptions** were extracted from prior studies [8, 27, 137]:

- **Transparency**: the level of users' understanding of how recommendations are generated
- **Scrutability**: users' ability to tell if the system is wrong
- **Trust**: the level of users' confidence in the system being right
- **Persuasiveness**: to what degree users are convinced to consume items
- **Efficiency**: users' ability to make quick decisions
- **Effectiveness**: users' ability to make good decisions
- **Satisfaction**: the users' enjoyment from using the system

A questionnaire containing seven statements corresponding to each aspect was created based on the ResQue questionnaire[3] [106]. Participants were asked to rate their level of agreement with each statement on a 5-point Likert response scale.

[3]The statements for all the aspects can be found in Article III, Table 1.

Table 4.1: Odds ratios for four independent variables derived from seven ordinal logistic regression models corresponding to the seven aspects of user perceptions. Statistically significant results are indicated in bold, and significance codes used: $* < 0.05$, $** < 0.01$, $*** < 0.001$.

DV.	Familiar	Cross-domain	Explanations	CD: Exp.
Transparency	**1.155****	0.527	**2.257***	1.514
Scrutability	1.03	0.899	**2.604****	0.449
Trust	**1.156****	**0.505***	1.083	1.861
Persuasiveness	**1.219*****	0.706	0.817	1.249
Efficiency	**0.896***	0.651	1.259	1.642
Effectiveness	0.931	0.748	0.731	1.868
Satisfaction	1.039	1.275	1.454	0.853

4.1.4 Main Findings

This section presents the main findings of the user study with 237 valid participants. The study was conducted on Amazon Mechanical Turk. As part of quality control, a strategy known as catch-trials was applied to filter out participants who provided suspicious responses. The distribution of participants across the four experimental conditions is balanced, ranging from 57-63. Given that the majority of participants agreed with the explanations for both common and uncommon tags, the validity of the generated explanations can be confirmed. Ordinal logistic regression was applied to analyze the impact of cross-domain settings and explanations on user perceptions and behavioural intentions. It should be noted that an additional variable, referred to as "Familiar" in the following results, was incorporated to improve model fit. For further discussion related to this variable, readers are referred to Article III (Section 5).

User Perceptions. As shown in Table 4.1, the cross-domain setting decreased perceived trust among users, whereas explanations had positive impact on transparency and scrutability for all recommendations. In addition, cross-domain explanations influenced user perceptions the same as single-domain explanations, given that none of the interaction terms were statistically significant.

Behavioural Intentions. As can be seen in Table 4.2, cross-domain recommendations decreased users' interest, which may result from the de-

Table 4.2: Odds ratios of the ordinal logistic regression model, where the dependent variable corresponds to ratings that participants gave to express their interest in reading each book recommendation. Significance codes used: $* < 0.05$, $** < 0.01$, $*** < 0.001$.

DV.	Familiar (read)	Cross-domain	Explanations	CD: Exp.
Ratings	**5.929*** **	**0.701*** **	**0.783***	**1.662*** **

creased user trust discussed above. Moreover, it was found that explanations increased interest in cross-domain recommendations, but not to the same extent as single-domain recommendations without explanations.

4.2 Discussion

This chapter investigates the extrinsic challenges of offline evaluation, particularly in cross-domain recommendation, in a user study. We present a novel study design to investigate the impact of cross-domain settings and explanations on user perceptions and behavioural intentions while avoiding numerous confounding factors. These user dynamics, which are challenging to capture in offline evaluation, are important sources of extrinsic challenges, contributing to the discrepancy between the performance of recommenders estimated offline and in real-world environments.

The results reveal the existence of negative user perceptions and behavioural intentions toward cross-domain recommendations, regardless of recommendation quality. Although incorporating explanations partially mitigates these negative effects, its performance is nonetheless perceived as inferior to single-domain recommendations without explanations. Due to the lack of awareness regarding these negative effects, offline evaluation is highly probable to overstate the performance of cross-domain recommendation, yielding inconsistent results with online evaluation. Given that nearly all studies investigating cross-domain recommendation rely on offline evaluation, the reported progress should be interpreted with greater caution.

Moreover, our study highlights the importance of user experiments for recommender system evaluation. Indeed, various exogenous factors, such as interface design and users' original belief, can affect user perceptions of recommendations, which can further influence the utility of recommender systems. These impacts can be fully understood only through user studies.

Chapter 5

Meta-evaluation of Recommender System Evaluation

In Chapters 3 and 4, we examined the key challenges associated with offline evaluation in top-N, sequential and cross-domain recommendation. It is important to note that, even though adjustments are made to address the unique requirements of each recommendation task, current evaluation practices across all tasks follow a unified framework known as top-N recommendation evaluation, where rank-based metrics are used to assess the quality of top-N recommendation lists [76, 138]. Despite its widespread use, our understanding of top-N recommendation evaluation is still limited, and its application in recommender system evaluation raises several open questions.

First, it remains uncertain whether rank-based metrics are an effective proxy for assessing the utility of recommender systems. Recommender systems face the problem of incomplete user preference data, as their goal is to recommend previously unseen items to users rather than those already known to be preferred [16, 94, 129]. Consequently, even if a recommender system correctly identifies the relevant items, it may still underperform in terms of metric values due to the inaccessibility of relevant information. In other words, top-N recommendation evaluation that assesses the quality of recommendation lists may fail to reflect the true capabilities of recommendation algorithms [4].

Second, top-N recommendation evaluation has been shown to have a low level of discriminative power [16], which highlights the difficulty in distinguishing between the best-performing models. Indeed, a similar observation was made in our own evaluation study of top-N recommendation presented in Chapter 3. It was found that many users do not have even a single pos-

itive item in the top-10[4] across various algorithms and three benchmark datasets[5], leading to rank-based metrics with small magnitudes. The problem is more pronounced in the Amazon-Books dataset [51], likely due to its higher sparsity. Given the minimal difference in performance indicated by rank-based metrics, the reliability of top-N recommendation evaluation is questionable.

The primary objective of recommender systems is to deliver personalized content to each user [17, 76]. Rank-based metrics, however, treat all users equally and evaluate system effectiveness by averaging performance across them. As a result, top-N recommendation evaluation may fail to capture the variations in recommender system performance across users. For instance, some models may perform better for users with niche tastes, while others are more effective for average users.

Another limitation of top-N recommendation evaluation is the difficulty in explaining its results. A commonly observed issue in offline evaluation is that some algorithms perform well on certain datasets but not on others [55]. This discrepancy may arise from differences in dataset characteristics. For instance, models that more effectively capture item popularity distributions tend to perform better on datasets that exhibit higher popularity bias. This information is unfortunately hidden in top-N recommendation evaluation.

This chapter conducts a comprehensive meta-evaluation to gain a deeper understanding of top-N recommendation evaluation (Article IV [91]). We propose a novel methodology to adapt item response theory (IRT)[117], a family of latent variable models used in psychometric assessment and educational testing, to recommender systems evaluation. IRT allows us to jointly estimate the latent ability of recommender systems in modeling user preferences and the latent difficulty of ranking pairs of positive and negative items for a given user. We analyze whether rank-based metrics accurately reflects the utility of recommender systems by comparing them with the recommender's latent ability. Further analyses of the marginal distribution of recommendation difficulties across users and items facilitate a deeper understanding of how user engagement and item popularity influence the evaluation. This chapter begins with a brief introduction to item response theory and its adaptation to recommender systems in Section 5.1. Section 5.2 presents the main findings of the study, and Section 5.3 provides a brief discussion to conclude the chapter.

[4]The cutoff value of 10 is mostly adopted in the literature [37, 38, 134].

[5]The median proportions of such users across ~ 50 recommendation models are 54.8%, 39.8%, and 92% on MovieLens-100K, MovieLens-1M, and Amazon-Books, respectively.

5.1 Adapting Item Response Theory to RS

This section aims to establish a meta-evaluation framework based on item response theory (IRT). We begin by introducing the basic concepts of IRT, and then illustrate how to adapt IRT to recommender system settings.

5.1.1 Introduction to IRT

IRT is a family of statistical models in psychometrics used to analyze how individual test takers respond to test items [25, 117]. Unlike traditional testing, where each question is considered to have an equal contribution to the assessment, IRT is under the assumption that different characteristics (e.g difficulty) of test items can have distinct impacts on the evaluation. As an example, it is more reasonable to assign a higher score to a test taker who is able to answer more difficult questions. The core of IRT is, therefore, to jointly estimate the latent ability of test takers and characteristics of test items based on the pattern of responses via statistical modeling.

It is worth noting that Item Response Theory (IRT) has gained increasing popularity in recent years within AI. For example, it has been used to evaluate classification models in machine learning [95, 96], question answering in natural language processing (NLP) [115], and to facilitate test set creation and evaluation in NLP [81, 139]. Since the publication of our paper, IRT has also been applied to fairness evaluation in recommender systems [154]. To avoid confusion related to the meaning of "item" in recommender systems versus IRT, we use the term "question" when referring to test items in the context of IRT.

IRT Models. Common IRT models include 1PL (i.e. Rasch), 2PL, and 3PL models, depending on how many parameters are utilized to model each question. Taking the 3PL model as an example, it models the probability of a test taker j giving the correct response to a question i as a logistic function with three extra parameters for the question, namely, difficulty, b_i, discrimination, a_i, and guessability, c_i:

$$P(U_{ij} = 1|\theta_j, a_i, b_i, c_i) = c_i + \frac{1 - c_i}{1 + exp(-a_i \cdot (\theta_j - b_i))}, \tag{5.1}$$

where U_{ij} represents the response with 1 as the correct response and 0 otherwise and θ_j denotes the latent ability of the test taker. According to

the characteristics of the logistic function, three parameters related to the question can be explained as follows:

- **difficulty** ($b_i \in \mathbb{R}$): is measured on the same scale as θ_j, so that higher ability of a test taker with respect to b_i indicates a higher probability of the response being correct,
- **discrimination** ($a_i > 0$): is determined by the slope at b_i and a larger a_i illustrates that the question better distinguishes test takers with abilities above b_i from those with lower abilities,
- **guessability** ($c_i \in [0, 1]$): is the probability that a question can be guessed correctly, irrespective of θ_j.

In addition, the 3PL model can be simplified to a 2PL model with the guessability parameter $c_i = 0$, and to a 1PL model with $c_i = 0$ and $a_i = 1$.

5.1.2 IRT Adaptation in Recommender Systems

To evaluate the performance of a recommender system, a natural thought is to treat each recommender system as a "test taker" in the IRT formulation. However, the traditional IRT models can only be used to model dichotomous responses, while the goal is to evaluate the whole ranking of recommendations. To overcome this issue, we take inspiration from the Bayesian Personalized Ranking [110] to create "questions" as (user, positive item, negative item) triples. A recommender is considered to give the correct response if it can rank the positive item higher than the negative one; otherwise, the recommender is considered to provide an incorrect response. In this way, it is feasible to assess the recommender's performance in modeling user preference, which is defined as the ability to rank the preferred items over less preferred ones.

Formal definition. Given a set of N RS models and M questions (i.e. triples), each question can be represented by $\mathcal{T}_m = (u_k, i_l, i_t)$, where i_l and i_t represent positive and negative items for the user u_k, respectively. The latent ability θ_n of each RS model and parameters for each question, such as question difficulty b_m and discrimination a_m, can be estimated with various approaches, e.g. maximum likelihood estimation (MLE) [32], Markov Chain Monte Carlo (MCMC) [44] and variational inference (VI) [58]. We further aggregate the parameters for each individual item, user, and even user-item pair by averaging across all triples containing the target object. Taking

difficulty (b_m) as an example, the recommendation difficulties for item i_t, user u_k, and user-item pair (u_k, i_t) can be calculated as follows:

$$d_t^{(item)} = \frac{\sum_m b_m \cdot I(i_t \in \mathcal{T}_m)}{\sum_m I(i_t \in \mathcal{T}_m)}, \tag{5.2}$$

$$d_k^{(user)} = \frac{\sum_m b_m \cdot I(u_k \in \mathcal{T}_m)}{\sum_m I(u_k \in \mathcal{T}_m)}, \tag{5.3}$$

$$d_{kt}^{(pair)} = \frac{\sum_m b_m \cdot I[(u_k, i_t) \in \mathcal{T}_m]}{\sum_m I[(u_k, i_t) \in \mathcal{T}_m]}, \tag{5.4}$$

where $I(\cdot)$ denotes the indicator function.

Inference. Mean-field VI is the most appropriate approach for IRT modeling in the RS setting as other approaches such as MCMC are not suitable for the scale of the datasets involved [150]. Mean-field VI is under the assumption that the posterior distribution can be simplified by factorizing over each latent variable resulting in

$$q_\phi(\boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{\mu}, \boldsymbol{\tau}) = p(\boldsymbol{\mu})p(\boldsymbol{\tau}) \prod_{n,m} p(\theta_n)p(\boldsymbol{t_m}), \tag{5.5}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{t}$ denote the latent variables of RS models' ability and question parameters. The parameters of prior distributions of the latent variables are $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$. To estimate the latent variables, variational inference aims to minimize the KL-divergence between the simplified posterior $q_\phi(\cdot)$ and the true posterior, which is done by maximizing the evidence lower-bound (ELBO) given that the true posterior is unknown.

5.2 Main Findings

This study involves 52 recommendation models and three benchmark datasets, allowing for reliable inference of IRT parameters. Due to the extremely large number of triples that can be composed even with small datasets[6], a **sampling procedure** is proposed to make IRT training more scalable (Article IV, Section 4.5). We validate the sampling procedure by showing a high consistency between 5 replicates of IRT models trained on different sets of sampled triples. The reported results are obtained with 2PL models as they consistently outperform 3PL models in terms of classification accuracy of predicted responses.

[6]The number of triples grows quadratically with respect to the number of items in a dataset.

Table 5.1: The Spearman's rank correlation coefficients between the model ability and various rank-based evaluation metrics. All results are statistically significant ($P < 0.05$).

Metrics	MovieLens-1M			Amazon-Books		
	All	Bottom	Top	All	Bottom	Top
Precision@10	0.808	0.912	0.151	0.816	0.802	0.473
nDCG@10	0.82	0.91	0.226	0.805	0.795	0.381
Recall@10	0.831	0.93	0.301	0.828	0.798	0.531
HitRate@10	0.828	0.924	0.253	0.822	0.802	0.49
MRR@10	0.792	0.892	0.152	0.798	0.795	0.344
MAP@10	0.808	0.905	0.177	0.796	0.781	0.365

5.2.1 Recommender's Ability

We investigate whether top-N evaluation metrics can capture the recommender's ability to model user preferences by measuring the Spearman's rank correlation coefficient between their achieved model rankings, as shown in Table 5.1[7]. Overall, the correlations are higher than ~ 0.8 across the three datasets. However, a closer inspection discloses that higher scores in top-N evaluation fail to reflect a recommender's ability given that the correlations between top-performing models (i.e. models with above median scores for rank-based metrics) decrease dramatically. It should be noted that top-performing models are the primary focus in RS offline evaluation where the goal is to identify the optimal recommender to be applied in real-world applications.

5.2.2 Test Set Information and Characteristics

Test Set Information

IRT also allows for the analysis of how informative the test set is in assessing the abilities of recommenders (See details in Article IV, Section 2.3). It is noted that MovieLens-1M contains substantially more information than MovieLens-100K due to a larger size. Despite Amazon-Books containing 3×

[7]For clarity, the results from MovieLens-100K were omitted from the thesis, as they were very similar to those obtained with MovieLens-1M. Readers are referred to the original paper (Article IV) for further details.

the number of question triples compared to MovieLens-1M, their maximum Fisher information is approximately equal.

Test Set Characteristics

By averaging the difficulty and discrimination values across triples, we can further analyse the characteristics of user-item pairs in the test set. Figure 5.1 illustrates different patterns of difficulty and discrimination of test sets of MovieLens and Amazon-Books datasets. For clarity, we refer to the user-item pairs in the test set (i.e. blue dots) as "test pairs" and top-10 user-item pairs that appeared in the test set (i.e. red dots) as "top-10 pairs". It should be noted that the difficulty and discrimination can contextualize recommendation ability (see the distribution of red dots). A model that yields more red dots (i.e. top-10 recommendations), especially those of higher difficulty, typically has greater recommendation ability.

MovieLens datasets. The highest discriminatory power lies in lower difficulty test pairs (see blue dots in any subplot in the middle row), suggesting the inability of rank-based metrics to distinguish between top-performing models. Moreover, top-10 pairs are concentrated at low difficulties regardless of the models (see red dots in each subplot), which indicates the existence of challenging user-item pairs that are difficult for most recommenders to rank correctly.

Amazon-Books. Top-10 pairs are more evenly spread over the distribution of difficulty (see blue dots in the bottom row). Given that many top-ranking models, such as SGL and NCL, are able to identify top-10 pairs with high difficulty (see red dots), it is suggested that each of these models provides distinct top-10 recommendations that users prefer. Hence, model ensembling may be an effective method for improved recommendation performance on Amazon-Books.

5.2.3 Item Popularity and User Engagement

The qualitative analyses further reveal how item popularity and user engagement impact the recommendation difficulty of each item and user[8].

[8]However, these results are not shown with Amazon-Books. One conjecture is that the limited number of popular items (i.e. items with more than 500 ratings) in the dataset fails to reveal any significant patterns.

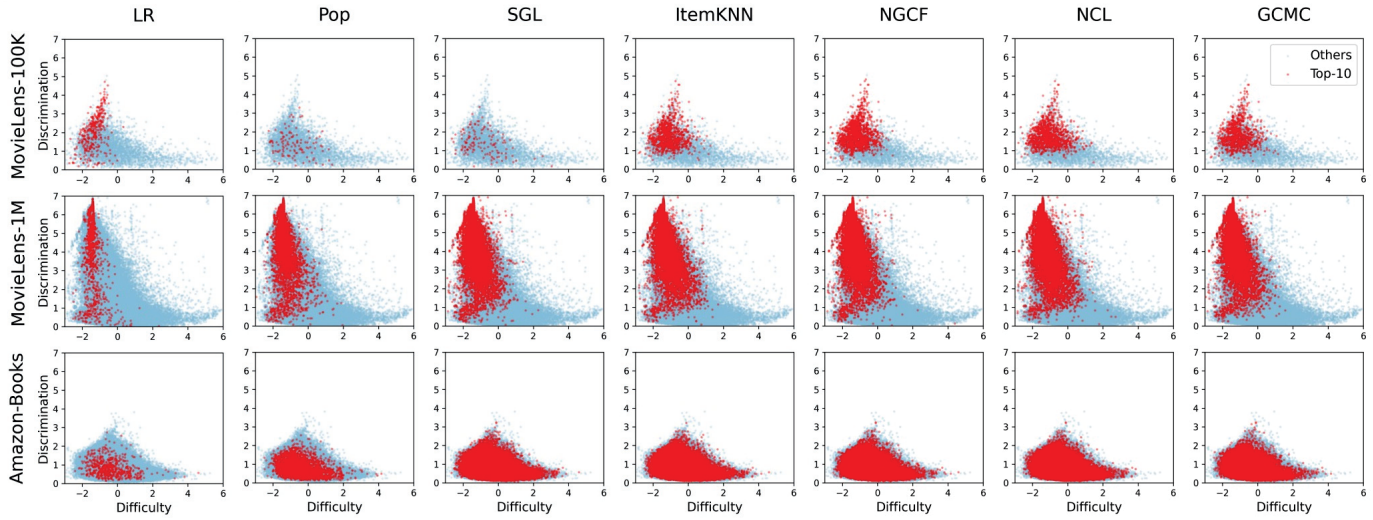

Figure 5.1: User-item pair difficulty (x-axis) and discrimination (y-axis) plots for selected methods across three datasets. The blue dots represent user-item pairs included in the test set, whereas the red dots denote test user-item pairs, where the item is featured in the top-10 recommendations for the corresponding user.

We summarize the main findings below and refer readers to the original publication for more details (Article IV, Figures 5 and 6).

Item Popularity. Two notable facts regarding the recommendation difficulty of items can be observed: 1) items with high popularity tend to have lower difficulty, and 2) items with low popularity can be of low difficulty if they were rated by highly engaged users.

User Engagement. It was found that higher engagement does not necessarily entail lower user difficulty, instead, user difficulty converges to the long running average as user engagement increases. Moreover, less engaged users can benefit from shared preferences from highly engaged users.

5.3 Discussion

This chapter proposes an IRT-based approach to facilitate meta-evaluation of recommender system evaluation. Our analyses demonstrate that IRT is a powerful tool for gaining a comprehensive understanding of offline recommendation evaluation from three aspects: 1) the recommender's ability to model user preferences, 2) information, difficulty and discrimination of a test set, and 3) the impact of user engagement and item popularity.

Our findings showed that top scores from rank-based metrics do not reflect improvements in recommendation algorithms' ability to model user preferences. We further analyzed different patterns of difficulty and discrimination in three benchmark datasets, showing how these parameters can be used to contextualize latent ability between methods and datasets. Finally, our qualitative analysis revealed valuable patterns in how recommendation difficulty interacts with user engagement and item popularity, and how these patterns relate to real-world recommendation scenarios.

The proposed IRT framework can also be regarded as an effective evaluation framework for recommender systems. First, IRT directly estimates the latent ability of recommender systems to model user preferences. Although rank-based evaluation aligns with the top-N recommendation task more closely, latent ability that is based on pairwise item ranking may better reflect real preference of users and, therefore, is more resilient to the problem of missing judgements in offline evaluation [11]. Meanwhile, IRT de-emphasizes item pairs that are easy to rank while attributing more importance to the difficult ones, intrinsically providing higher discriminative power. In addition, IRT may present a solution for managing popularity bias in evaluation. For instance, items with higher popularity are more

likely to have lower difficulty and, therefore, contribute less to the evaluation, as we discussed in Section 5.2.3.

Despite these benefits, IRT modeling in the context of recommender systems imposes a high computational burden, thereby preventing the approach from being widely used. It may be worthwhile, however, to construct and manage a unified leaderboard on each benchmark dataset using IRT to assist researchers in uncovering patterns hidden in the raw values of rank-based metrics.

Chapter 6

Discussion

Offline evaluation plays a crucial role in the development of recommender systems. Initially designed to assess the performance of IR systems, the top-N evaluation method has become the predominant approach for evaluating recommender systems over the past decade [17]. However, recent studies have identified numerous challenges with this method, revealing its vulnerability to biases in datasets [11], potential for misapplication [37, 38], and inconsistency with online evaluation [9, 76]. Furthermore, research on evaluation flaws has primarily focused on top-N recommendation, with limited attention given to other specialised recommendation tasks, such as sequential and cross-domain recommendation. These observations highlight an urgent need for a thorough investigation into the effectiveness and robustness of offline evaluation of recommender systems.

In this thesis, we conducted a series of evaluation studies exploring both methodological and extrinsic challenges related to various recommendation tasks. We examined the validity of the top-N evaluation method and identified, for the first time, distinct evaluation flaws specific to sequential and cross-domain recommendation. Additionally, this thesis contributes a novel meta-evaluation framework based on item response theory which allows an in-depth understanding of recommender system evaluation from various aspects. This chapter revisits the key findings of the studies, discusses their implications, and highlights potential future research directions.

6.1 Summary of the Main Findings

This section summarizes the key findings from the studies conducted in this thesis. We examine how these findings relate to the three research questions that guided this work.

RQ1: Methodological Challenges

Our investigation of the methodological challenges in Chapter 3 focused on the evaluation practices in top-N recommendation and sequential recommendation. In the case of top-N recommendation, we showed that previously reported levels of inconsistency between sampled and traditional non-sampled metrics had been overestimated. Nevertheless, sampled evaluation metrics have additional advantages, including higher discriminative power and improved resilience to popularity bias when appropriate sampling configurations are applied.

In sequential recommender systems, we observed that the commonly used data splitting strategy, temporal leave-one-out, caused data leakage during evaluation. Consequently, the performance of sequential recommender systems was significantly overestimated. Another important finding of the study is that sequential recommenders could even be outperformed by many non-sequential models after mitigating data leakage.

RQ2: Extrinsic Challenges

Extrinsic challenges often arise from insufficient consideration of user-relevant factors in offline evaluation. Given the complexities involved in conducting field studies with real users, there has been limited investigation of extrinsic evaluation challenges in different recommendation tasks. Cross-domain recommendation, in particular, may face potential changes in interface design or interaction patterns, presenting unique evaluation challenges that remain underexplored. Indeed, the results of our user study presented in Chapter 4 demonstrate that by simply telling users that the recommendations were generated based on information from another domain can lead to decreased user trust and interest. Although incorporating recommendation explanations can partially alleviate negative user perceptions, cross-domain recommendation would still perform worse than single-domain recommendation without explanations.

RQ3: Improved Methodology for Meta-evaluation

To enhance our understanding of recommender systems evaluation, we proposed a powerful meta-evaluation framework based on IRT in Chapter 5 to examine the top-N evaluation method. The study addressed both methodological and extrinsic factors in offline evaluation, focusing on both the va-

lidity of top-N evaluation and how the characteristics of datasets, individual users, and individual items influence recommender systems' performance. Surprisingly, we found that higher model rankings in top-N evaluation did not necessarily reflect higher ability of recommender systems to model user preferences. We further investigated the relationship between system performance and dataset characteristics, such as difficulty and discrimination. The qualitative analysis of the study also provided valuable insights into the impact of user engagement and item popularity on recommender systems' performance.

6.2 Implications and Future Directions

This thesis identified key challenges in offline evaluation across multiple recommendation paradigms, highlighting critical concerns about the reliability of the reported progress in the RS research community. This section further discusses how these observations and the findings can contribute to driving practical impact in top-N recommendation, sequential recommendation, and cross-domain recommendation, respectively.

Top-N Recommendation

Our observations from the studies draw attention to improved evaluation practices for top-N recommendation. Despite traditional rank-based evaluation being widely adopted, it has been shown to suffer from low discriminative power. Moreover, as datasets used in recommender system continue to grow in size, computationally efficient alternative methods for evaluation are inevitably required. Fortunately, based on our findings, sampled evaluation metrics show great potential to address these limitations. One promising research direction is, therefore, to investigate how to determine the optimal negative sampling strategy, given the characteristics of each dataset. This would enable us to overcome the limitations of sampled metrics, achieving higher discriminative power and robustness without compromising our confidence in the evaluation results. Another promising direction involves employing the IRT framework to construct optimal test sets. This would provide a more efficient solution that bridges the gap between traditional and sampled evaluation metrics, allowing test sets to be created by balancing difficulty and discrimination, thus achieving higher accuracy while maintaining strong discriminative power.

Sequential Recommendation

Our findings suggest that performance gains of sequential recommenders over non-sequential models are potentially misleading, highlighting a pressing need for re-evaluation of their effectiveness with more reliable evaluation practices. To mitigate data leakage, we recommend replacing temporal leave-one-out using alternative data-splitting strategies, such as split-by-timepoint leave-one-out. Moreover, given our limited understanding of sequential recommender systems, further exploration is needed to determine whether sequential dynamics can genuinely enhance recommendations, and if so, in what contexts and to what extent. Indeed, each dataset lies on a spectrum of how much sequential information there is to take advantage of by sequential recommenders. It is, therefore, important to develop unbiased methods for estimating the level of sequential information in the datasets. This will not only deepen our understanding of its influence on recommendation performance, but facilitate the selection of more appropriate datasets for evaluation.

Cross-domain Recommendation

Our work highlights the potential negative impact of the cross-domain setting on user perceptions of cross-domain recommendations. One future research direction is to quantify the impact of user perceptions, investigating whether top-performing cross-domain recommendation algorithms can mitigate this issue and how much improvement in offline performance is necessary to produce meaningful real-world impact. Moreover, our findings suggest that future research could focus on exploring novel system designs, such as incorporating better recommendation explanations, to reduce the impact of negative user perceptions.

Our results also emphasize the importance of user experiments in understanding the true effectiveness of recommender systems. However, a hidden fact is that many researchers avoid user studies due to the complexities of system implementation and participant recruitment [9, 116]. A "half-way house" solution may be to utilize Mechanical Turk in recommender system evaluation, allowing for more efficient collection of human feedback [71]. As an alternative, future exploration could focus on the development of user simulation techniques, such as leveraging advances in large language models [145]. Incorporating simulated users into evaluations can potentially provide valuable insights into user-relevant factors like perception and satisfaction, leading to more informative conclusions.

6.3 Limitations

The primary limitation of the thesis is that it only focuses on the utility of recommender systems, while other relevant evaluation criteria, such as diversity and novelty, are not explored. Furthermore, the findings of the experimental studies in Chapters 3 and 5 may be constrained to specific experimental conditions, such as the algorithms and datasets. We attempted to mitigate these limitations by including a variety of algorithms and using datasets with diverse characteristics to enhance the generalisability of the results. However, given that these studies are data-driven, the results may be influenced by inherent biases in the datasets used.

Similarly, the user study presented in Chapter 4 was conducted using Amazon Mechanical Turk. Several limitations relevant in this context include 1) lack of control over the experimental environment, and 2) potential response biases. To address these limitations, our study incorporated additional verification steps to filter out susceptible participants.

References

[1] Zeinab Abbassi, Sihem Amer-Yahia, Laks V. S. Lakshmanan, Sergei Vassilvitskii, and Cong Yu. Getting recommender systems to think outside the box. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys'09, pages 285–288. Association for Computing Machinery, 2009.

[2] Stavros P. Adam, Stamatios-Aggelos N. Alexandropoulos, Panos M. Pardalos, and Michael N. Vrahatis. No free lunch theorem: A review. In *Approximation and Optimization: Algorithms, Complexity and Applications*, pages 57–82. Springer International Publishing, 2019.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[4] Charu C. Aggarwal. Evaluating recommender systems. In *Recommender Systems: The Textbook*, pages 225–254. Springer International Publishing, 2016.

[5] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, Dietmar Jannach, and Claudio Pomo. Top-n recommendation algorithms: A quest for the state-of-the-art. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP'22, pages 121–131. Association for Computing Machinery, 2022.

[6] Nabiha Asghar. Yelp dataset challenge: Review rating prediction. arXiv:1605.05362, 2016.

[7] Enkh-Amgalan Baatarjav, Santi Phithakkitnukoon, and Ram Dantu. Group recommendation system for Facebook. In *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, pages 211–219. Springer Berlin Heidelberg, 2008.

[8] Krisztian Balog and Filip Radlinski. Measuring recommendation explanation quality: The conflicting goals of explanations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'20, pages 329–338. Association for Computing Machinery, 2020.

[9] Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. Research paper recommender system evaluation: a quantitative literature survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, RepSys'13, pages 15–22. Association for Computing Machinery, 2013.

[10] Nicholas J. Belkin and W. Bruce Croft. Information filtering and nformation retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[11] Alejandro Bellogín, Pablo Castells, and Iván Cantador. Statistical biases in information retrieval metrics for recommender systems. *Information Retrieval Journal*, 20:606–634, 2017.

[12] James Bennett and Stan Lanning. The Netflix prize. In *Proceedings of the KDD Cup Workshop 2007*, pages 3–6. Association for Computing Machinery, 2007.

[13] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Cross-domain mediation in collaborative filtering. In *User Modeling 2007*, pages 355–359. Springer Berlin Heidelberg, 2007.

[14] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'04, pages 25–32. Association for Computing Machinery, 2004.

[15] Li Cai, Kilchan Choi, Mark Hansen, and Lauren Harrell. Item response theory. *Annual Review of Statistics and Its Application*, 3(1):297–321, 2016.

[16] Rocío Cañamares and Pablo Castells. On target item sampling in offline recommender system evaluation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys'20, pages 259–268. Association for Computing Machinery, 2020.

[17] Rocío Cañamares, Pablo Castells, and Alistair Moffat. Offline evaluation options for recommender systems. *Information Retrieval Journal*, 23(4):387–410, 2020.

[18] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys'11, pages 387–388. Association for Computing Machinery, 2011.

[19] Benjamin A. Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Transactions on Information Systems (TOIS)*, 30(1):4, 2012.

[20] Li Chen and Pearl Pu. Interaction design guidelines on critiquing-based recommender systems. *User Modeling and User-Adapted Interaction*, 19:167–206, 2009.

[21] Dan Cosley, Shyong K. Lam, Istvan Albert, Joseph A. Konstan, and John Riedl. Is seeing believing? How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'03, pages 585–592. Association for Computing Machinery, 2003.

[22] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys'16, pages 191–198. Association for Computing Machinery, 2016.

[23] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys'10, pages 39–46. Association for Computing Machinery, 2010.

[24] Alexander Dallmann, Daniel Zoller, and Andreas Hotho. A case study on sampling strategies for evaluating neural sequential item recommendation models. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 505–514. Association for Computing Machinery, 2021.

[25] Rafael Jaime De Ayala. *The Theory and Practice of Item Response Theory*. Guilford Publications, 2013.

[26] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[27] Vicente Dominguez, Pablo Messina, Ivania Donoso-Guzmán, and Denis Parra. The effect of explanations and algorithmic accuracy on visual recommender systems of artistic images. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 408–416. Association for Computing Machinery, 2019.

[28] Zhenhua Dong, Zhe Wang, Jun Xu, Ruiming Tang, and Jirong Wen. A brief history of recommender systems. In *Proceedings of the 4th International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, DLP-KDD'22. Association for Computing Machinery, 2022.

[29] Gintare K. Dziugaite and Daniel M. Roy. Neural network matrix factorization. arXiv:1511.06443, 2015.

[30] Travis Ebesu, Bin Shen, and Yi Fang. Collaborative memory network for recommendation systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'18, pages 515–524. Association for Computing Machinery, 2018.

[31] Michael D. Ekstrand, F. Maxwell Harper, Martijn C. Willemsen, and Joseph A. Konstan. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys'14, pages 161–168. Association for Computing Machinery, 2014.

[32] Scott R. Eliason. *Maximum Likelihood Estimation: Logic and Practice*. Sage Publications, 1993.

[33] Manuel Enrich, Matthias Braunhofer, and Francesco Ricci. Cold-start management with cross-domain collaborative filtering and tags. In *E-Commerce and Web Technologies*, pages 101–112. Springer Berlin Heidelberg, 2013.

[34] Xinyan Fan, Zheng Liu, Jianxun Lian, Wayne Xin Zhao, Xing Xie, and Ji-Rong Wen. Lighter and better: Low-rank decomposed self-attention networks for next-item recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and*

Development in Information Retrieval, SIGIR'21, pages 1733–1737. Association for Computing Machinery, 2021.

[35] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39(1):10, 2020.

[36] Ignacio Fernández-Tobías and Iván Cantador. Exploiting social tags in matrix factorization models for cross-domain collaborative filtering. In *CBRecSys 2014 - Workshop on New Trends in Content-Based Recommender Systems*, pages 34–41. Association for Computing Machinery, 2014.

[37] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)*, 39(2):20, 2021.

[38] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys'19, pages 101–109. Association for Computing Machinery, 2019.

[39] Andres Ferraro, Dietmar Jannach, and Xavier Serra. Exploring longitudinal effects of session-based recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys'20, pages 474–479. Association for Computing Machinery, 2020.

[40] Garrett M. Fitzmaurice, Nan M. Laird, and James H. Ware. *Applied Longitudinal Analysis*. John Wiley & Sons, 2012.

[41] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI'19, pages 94–101. AAAI Press, 2019.

[42] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. Cross-domain recommendation without sharing user-relevant data. In *Proceedings of the 2019 World Wide Web Conference*, WWW'19, pages 491–502. Association for Computing Machinery, 2019.

[43] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. Offline and online evaluation of news recommender systems at swissinfo.ch. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys'14, pages 169–176. Association for Computing Machinery, 2014.

[44] Charles J. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7(4):473–483, 1992.

[45] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for Twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW'13 Companion, pages 593–596. Association for Computing Machinery, 2013.

[46] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[47] Carlos A Gomez-Uribe and Neil Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, 2015.

[48] Anthony G Greenwald. Within-subjects designs: To use or not to use? *Psychological Bulletin*, 83(2):314–320, 1976.

[49] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 1725–1731. AAAI Press, 2017.

[50] F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2015.

[51] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW'16, pages 507–517. International World Wide Web Conferences Steering Committee, 2016.

[52] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. Outer product-based neural collaborative filtering. In

Proceedings of the 27th International Joint Conference on Artificial Intelligence, pages 2227–2233, 2018.

[53] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW'17, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[54] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'16, pages 549–558. Association for Computing Machinery, 2016.

[55] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[56] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. arXiv:1511.06939, 2016.

[57] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'95, pages 194–201. ACM Press/Addison-Wesley Publishing Co., 1995.

[58] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[59] Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. Music personalization at Spotify. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys'16, pages 373–373. Association for Computing Machinery, 2016.

[60] Dietmar Jannach, Lukas Lerche, and Markus Zanker. Recommending based on implicit feedback. In *Social Information Access: Systems and Technologies*, pages 510–569. Springer Berlin Heidelberg, 2018.

[61] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54(5):105, 2021.

[62] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: an Introduction*. Cambridge University Press, 2010.

[63] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[64] Olivier Jeunen. Revisiting offline evaluation for implicit-feedback recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys'19, pages 596–600. Association for Computing Machinery, 2019.

[65] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. A critical study on data leakage in recommender system offline evaluation. *ACM Transactions on Information Systems (TOIS)*, 41(3):75, 2023.

[66] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of 2018 IEEE International Conference on Data Mining*, pages 197–206. IEEE, 2018.

[67] George Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, CIKM'01, pages 247–254. Association for Computing Machinery, 2001.

[68] Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1–2):1–224, 2009.

[69] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[70] Muhammad M. Khan, Roliana Ibrahim, and Imran Ghani. Cross domain recommender systems: A systematic literature review. *ACM Computing Surveys (CSUR)*, 50(3):36, 2017.

[71] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'08, pages 453–456. Association for Computing Machinery, 2008.

[72] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems. *User Modeling and User-adapted Interaction*, 22:441–504, 2012.

[73] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'13, pages 1168–1176. Association for Computing Machinery, 2013.

[74] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18:140–181, 2009.

[75] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Grouplens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[76] Joseph A. Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-adapted Interaction*, 22:101–123, 2012.

[77] 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*, pages 426–434. Association for Computing Machinery, 2008.

[78] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[79] Denis Kotkov, Alan Medlar, Yang Liu, and Dorota Glowacka. On the negative perception of cross-domain recommendations and explanations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'24, pages 2102–2113. Association for Computing Machinery, 2024.

[80] Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge discovery and Data Mining*,

KDD'20, pages 1748–1757. Association for Computing Machinery, 2020.

[81] John P. Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 648–657. Association for Computational Linguistics, 2016.

[82] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, pages 331–339. Elsevier, 1995.

[83] James R. Lewis and Jeff Sauro. The factor structure of the system usability scale. In *Human Centered Design*, pages 94–103. Springer Berlin Heidelberg, 2009.

[84] Bin Li, Xingquan Zhu, Ruijiang Li, and Chengqi Zhang. Rating knowledge sharing in cross-domain collaborative filtering. *IEEE Transactions on Cybernetics*, 45(5):1068–1082, 2014.

[85] Yang Li, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. Recent developments in recommender systems: A survey. *IEEE Computational Intelligence Magazine*, 19(2):78–95, 2024.

[86] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, WWW'18, pages 689–698. International World Wide Web Conferences Steering Committee, 2018.

[87] Yu Liang and Martijn C. Willemsen. Exploring the longitudinal effects of nudging on users' music genre exploration behavior and listening preferences. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys'22, pages 3–13. Association for Computing Machinery, 2022.

[88] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[89] Nathan N. Liu and Qiang Yang. Eigenrank: A ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'08, pages 83–90. Association for Computing Machinery, 2008.

[90] Yang Liu, Alan Medlar, and Dorota Glowacka. On the consistency, discriminative power and robustness of sampled metrics in offline top-n recommender system evaluation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys'23, pages 1152–1157. Association for Computing Machinery, 2023.

[91] Yang Liu, Alan Medlar, and Dorota Glowacka. What we evaluate when we evaluate recommender systems: Understanding recommender systems' performance using item response theory. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys'23, pages 658–670. Association for Computing Machinery, 2023.

[92] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 2464–2470. AAAI Press, 2017.

[93] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press Cambridge, 2008.

[94] Benjamin M. Marlin and Richard S. Zemel. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys'09, pages 5–12. Association for Computing Machinery, 2009.

[95] Fernando Martínez-Plumed, Ricardo B.C. Prudêncio, Adolfo Martínez-Usó, and José Hernández-Orallo. Making sense of item response theory in machine learning. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, pages 1140–1148. IOS Press, 2016.

[96] Fernando Martínez-Plumed, Ricardo B.C. Prudêncio, Adolfo Martínez-Usó, and José Hernández-Orallo. Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artificial Intelligence*, 271:18–42, 2019.

[97] Alan Medlar, Denis Kotkov, and Dorota Głowacka. Unexplored frontiers: A review of empirical studies of exploratory search. *SIGIR Forum*, 58(1), 2024.

[98] Larry R. Medsker and Lakhmi C. Jain. *Recurrent Neural Networks: Design and Applications*. CRC Press, 1999.

[99] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Exploring data splitting strategies for the evaluation of recommendation models. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys'20, pages 681–686. Association for Computing Machinery, 2020.

[100] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, Iadh Ounis, Siwei Liu, Yaxiong Wu, Xi Wang, Shangsong Liang, Yucheng Liang, Guangtao Zeng, Junhua Liang, and Qiang Zhang. BETA-Rec: Build, evaluate and tune automated recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys'20, pages 588–590. Association for Computing Machinery, 2020.

[101] Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. TALMUD: transfer learning for multiple domains. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM'12, pages 425–434. Association for Computing Machinery, 2012.

[102] Tien T. Nguyen, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. User personality and user satisfaction with recommender systems. *Information Systems Frontiers*, 20:1173–1189, 2018.

[103] University of Minnesota, Recommender systems specialization. `https://coursera.org/specializations/recommender-systems`, 2018, Last accessed: 1 January 2025.

[104] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27:313–331, 1997.

[105] Aleksandr Petrov and Craig Macdonald. A systematic review and replicability study of BERT4Rec for sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys'22, pages 436–447. Association for Computing Machinery, 2022.

[106] Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys'11, pages 157–164. Association for Computing Machinery, 2011.

[107] Pearl Pu, Li Chen, and Rong Hu. Evaluating recommender systems from the user's perspective: Survey of the state of the art. *User Modeling and User-Adapted Interaction*, 22:317–355, 2012.

[108] Dimitrios Rafailidis and Fabio Crestani. A collaborative ranking model for cross-domain recommendations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM'17, pages 2263–2266. Association for Computing Machinery, 2017.

[109] Tomas Rehorek, Ondrej Biza, Radek Bartyzal, Pavel Kordik, Ivan Povalyev, and Ondrej Podstavek. Comparing offline and online evaluation results of recommender systems. In *Proceedings of the REVEAL workshop at RecSys Conference*, RecSys'18. Association for Computing Machinery, 2018.

[110] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI'09, pages 452–461. AUAI Press, 2009.

[111] 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*, WWW'10, pages 811–820. Association for Computing Machinery, 2010.

[112] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW'94, pages 175–186. Association for Computing Machinery, 1994.

[113] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, pages 1–35. Springer Berlin Heidelberg, 2010.

[114] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*, WWW'07, pages 521–530. Association for Computing Machinery, 2007.

[115] Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4486–4503, 2021.

[116] Marco Rossetti, Fabio Stella, and Markus Zanker. Contrasting offline and online results when evaluating recommendation algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys'16, pages 31–34. Association for Computing Machinery, 2016.

[117] John Rust and Susan Golombok. *Modern Psychometrics: The Science of Psychological Assessment*. Routledge, 2014.

[118] Tetsuya Sakai. Laboratory experiments in information retrieval. *The Information Retrieval Series*, 40, 2018.

[119] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW'01, pages 285–295. Association for Computing Machinery, 2001.

[120] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth International Conference on Computer and Information Science*, volume 1, 2002.

[121] Badrul Sarwar, George Karypis, Joseph Konstan, and John T. Riedl. Application of dimensionality reduction in recommender system-A case study. In *ACM WebKDD workshop on Web Mining for E-commerce*. Association for Computing Machinery, 2000.

[122] Badrul M Sarwar, Joseph A Konstan, Al Borchers, Jon Herlocker, Brad Miller, and John Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW'98, pages 345–354. Association for Computing Machinery, 1998.

[123] Markus Schedl. The LFM-1b dataset for music retrieval and recommendation. In *Proceedings of the 2016 ACM on International Confer-*

ence on Multimedia Retrieval, ICMR'16, pages 103–110. Association for Computing Machinery, 2016.

[124] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'02, pages 253–260. Association for Computing Machinery, 2002.

[125] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297. Springer Berlin Heidelberg, 2011.

[126] Guy Shani, David Heckerman, Ronen I. Brafman, and Craig Boutilier. An MDP-based recommender system. *Journal of Machine Learning Research*, 6(9):1265–1295, 2005.

[127] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'95, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[128] Brent Smith and Greg Linden. Two decades of recommender systems at Amazon.com. *IEEE Internet Computing*, 21(3):12–18, 2017.

[129] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'10, pages 713–722. Association for Computing Machinery, 2010.

[130] Harald Steck. Evaluation of recommendations: Rating-prediction and ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys'13, pages 213–220. Association for Computing Machinery, 2013.

[131] Aixin Sun. Take a fresh look at recommender systems from an evaluation standpoint. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'23, pages 2629–2638. Association for Computing Machinery, 2023.

[132] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the*

28th ACM International Conference on Information and Knowledge Management, CIKM'19, pages 1441–1450. Association for Computing Machinery, 2019.

[133] Zhu Sun, Hui Fang, Jie Yang, Xinghua Qu, Hongyang Liu, Di Yu, Yew-Soon Ong, and Jie Zhang. DaisyRec 2.0: Benchmarking recommendation for rigorous evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8206–8226, 2023.

[134] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. Are we evaluating rigorously? Benchmarking recommendation for reproducible evaluation and fair comparison. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys'20, pages 23–32. Association for Computing Machinery, 2020.

[135] Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS'16, pages 17–22. Association for Computing Machinery, 2016.

[136] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. eTrust: Understanding trust evolution in an online world. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'12, pages 253–261, 2012.

[137] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 801–810. IEEE, 2007.

[138] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. On the robustness and discriminative power of information retrieval metrics for top-n recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys'18, pages 260–268. Association for Computing Machinery, 2018.

[139] Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. Comparing test sets with item response theory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 1141–1158. Association for Computational Linguistics, 2021.

[140] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010. Curran Associates, Inc., 2017.

[141] Ellen M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of the 24th Annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'01, pages 74–82. Association for Computing Machinery, 2001.

[142] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[143] Alexandra Vultureanu-Albişi and Costin Bădică. A survey on effects of adding explanations to recommender systems. *Concurrency and Computation: Practice and Experience*, 34(20):e6834, 2022.

[144] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'20, pages 1101–1110. Association for Computing Machinery, 2020.

[145] Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, et al. User behavior simulation with large language model-based agents for recommender systems. *ACM Transactions on Information Systems (TOIS)*, 43(2):55, 2024.

[146] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6332–6338. International Joint Conferences on Artificial Intelligence Organization, 2019.

[147] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 165–174. Association for Computing Machinery, 2019.

[148] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI'19, pages 5329–5336. AAAI Press, 2019.

[149] Ryen W. White and Resa A. Roth. *Exploratory Search: Beyond the Query-response Paradigm*. Number 3 in Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2009.

[150] Mike Wu, Richard L. Davis, Benjamin W. Domingue, Chris Piech, and Noah Goodman. Variational item response theory: Fast, accurate, and expressive. In *Proceedings of The 13th International Conference on Educational Data Mining*, pages 257–268. ERIC, 2020.

[151] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: A survey. *ACM Computing Surveys*, 55(5):97, 2022.

[152] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[153] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM'19, pages 529–538. Association for Computing Machinery, 2019.

[154] Ziqi Xu, Sevvandi Kandanaarachchi, Cheng Soon Ong, and Eirini Ntoutsi. Fairness evaluation with item response theory. In *Proceedings of the ACM on Web Conference 2025*, WWW'25, pages 2276–2288. Association for Computing Machinery, 2025.

[155] Deqing Yang, Jingrui He, Huazheng Qin, Yanghua Xiao, and Wei Wang. A graph-based recommendation across heterogeneous domains. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM'15, pages 463–472. Association for Computing Machinery, 2015.

[156] Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM'06, pages 102–111. Association for Computing Machinery, 2006.

[157] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. A survey on cross-domain recommendation: taxonomies, methods, and future directions. *ACM Transactions on Information Systems (TOIS)*, 41(2):42, 2022.

[158] Eva Zangerle and Christine Bauer. Evaluating recommender systems: Survey and framework. *ACM Computing Surveys (CSUR)*, 55(8):170, 2022.

[159] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):5, 2019.

[160] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. Revisiting alternative experimental settings for evaluating top-n item recommendation algorithms. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM'20, pages 2329–2332. Association for Computing Machinery, 2020.

[161] Wayne Xin Zhao, Zihan Lin, Zhichao Feng, Pengfei Wang, and Ji-Rong Wen. A revisiting study of appropriate offline evaluation for top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 41(2):32, 2022.

[162] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM'21, pages 4653–4664. Association for Computing Machinery, 2021.

[163] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International*

Conference on Information and Knowledge Management, CIKM'20, pages 1893–1902. Association for Computing Machinery, 2020.

[164] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. Cross-domain recommendation: Challenges, progress, and prospects. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4721–4728. International Joint Conferences on Artificial Intelligence Organization, 2021.

[165] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. Transfer-meta framework for cross-domain recommendation to cold-start users. In *Proceedings of the 44th International ACM SIGIR conference on research and development in Information Retrieval*, SIGIR'21, pages 1813–1817. Association for Computing Machinery, 2021.

[166] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW'05, pages 22–32. Association for Computing Machinery, 2005.