Beyond accuracy: enhancing multiple perspectives of recommendation through multi-objective optimization and evaluation

Politecnico
di Bari

Department of Electrical and Information Engineering
Electrical and Information Engineering Ph.D. Program
SSD: ING-INF/05 - Information Processing Systems

Final Dissertation

Beyond Accuracy: Enhancing Multiple Perspectives of Recommendation through Multi-Objective Optimization and Evaluation

by

Vincenzo Paparella

Supervisor
Prof. Tommaso Di Noia

Coordinator of the Ph.D. Program
Prof. Mario Carpentieri

Course n° 37, 01/01/2022 - 31/12/2024

Politecnico di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il sottoscritto PAPARELLA VINCENZO nato a TERLIZZI il14/09/1996

residente a RUVO DI PUGLIA in via ALDO MORO 130/A e-mail vincenzo.paparella@outlook.it

iscritto al 3° anno di Corso di Dottorato di Ricerca in INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo 37°

ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

BEYOND-ACCURACY: ENHANCING MULTIPLE PERSPECTIVES OF RECOMMENDATION THROUGH MULTI-OBJECTIVE OPTIMIZATION AND EVALUATION

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca in INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo 37°, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari*", emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data BARI, 20/02/2025 Firma Vincenzo Paparella

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data BARI, 20/02/2025 Firma Vincenzo Paparella

Politecnico
di Bari

Department of Electrical and Information Engineering
Electrical and Information Engineering Ph.D. Program
SSD: ING-INF/05 - Information Processing Systems

Final Dissertation

Beyond Accuracy: Enhancing Multiple Perspectives of Recommendation through Multi-Objective Optimization and Evaluation

by

Vincenzo Paparella

Referees
Prof. Raffaele Perego
Prof. Marco de Gemmis

Supervisor
Prof. Tommaso Di Noia

Coordinator of the Ph.D. Program
Prof. Mario Carpentieri

Course n° 37, 01/01/2022 - 31/12/2024

To Melody and Jack

“Make me small
So I can fit in your pockets
When you go
‘Cause when you’re gone
All of the colors, turn to gray”

Abstract

Recommender Systems (RSs) have become essential tools for alleviating information overload by providing personalized suggestions across various domains, including e-commerce, streaming platforms, and social networks. Traditionally, the evaluation and optimization of RSs have centered on accuracy as the primary success metric. While accuracy is critical for predicting user preferences, it fails to address broader dimensions crucial for enhancing user satisfaction, ensuring stakeholder fairness, and addressing societal impacts. Moreover, when multiple objectives are considered, conflicts often arise, i.e., improving one objective can detrimentally affect others, leading to a spectrum of possible optima. These challenges give rise to several critical questions: How can RSs evolve to balance accuracy with other objectives, such as diversity, novelty, and fairness, while meeting the needs of multiple stakeholders, including users, content providers, and platforms? How can we simultaneously evaluate RS effectiveness across diverse criteria? How can a single optimal solution be selected from a set of trade-offs? Finally, can we design a generic framework for optimizing RSs that accounts for multiple, often conflicting objectives? These questions highlight key open challenges in the field of RS research.

This dissertation addresses these gaps by focusing on two main areas: methodologies for multi-objective evaluation of RSs and the challenges associated with designing Multi-Objective Recommender Systems (MORSs). After an in-depth exploration of the background of RSs and multi-objective optimization, the thesis makes significant contributions in the following areas: (i) the application of Pareto frontiers to conduct a multi-objective evaluation of graph-based RSs, focusing on fairness aspects; (ii) the introduction of quality indicators for Pareto frontiers to uncover the potential of RSs beyond traditional accuracy metrics; (iii) the development of an analytical framework to assess the sensitivity of RSs to hyper-parameter tuning in multi-objective scenarios; (iv) a reproducibility study that identifies key challenges and ambiguities in the design and evaluation of MORSs; (v) the proposal of a novel, post-hoc Pareto-optimal solution selection strategy tailored explicitly for RS tasks; (vi) designing a flexible MORS framework incorporating objective-agnostic and scale-aware loss functions to achieve optimization across diverse recommendation objectives.

Publications

Certain ideas and figures presented in this thesis have been previously published in other works. Below is a comprehensive list of the Ph.D. candidate's publications. In this list, the Ph.D. candidate's name is highlighted in boldface wherever he is the corresponding author or one of the corresponding authors.

[1] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, Francesco Maria Donini, **Vincenzo Paparella**, and Claudio Pomo. "An Analysis of Local Explanation with LIME-RS". In: *Proceedings of the 12th Italian Information Retrieval Workshop 2022, Milan, Italy, June 29-30, 2022*. Ed. by Gabriella Pasi, Paolo Cremonesi, Salvatore Orlando, Markus Zanker, David Massimo, and Gloria Turati. Vol. 3177. CEUR Workshop Proceedings. CEUR-WS.org, 2022.

[2] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Daniele Malitesta, Vincenzo Paparella, and Claudio Pomo. "Auditing Consumer- and Producer-Fairness in Graph Collaborative Filtering". In: *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part I*. Ed. by Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo. Vol. 13980. Lecture Notes in Computer Science. Springer, 2023, pp. 33–48. DOI: 10.1007/978-3-031-28244-7_3.

[3] Dario Di Palma, Vito Walter Anelli, Daniele Malitesta, Vincenzo Paparella, Claudio Pomo, Yashar Deldjoo, and Tommaso Di Noia. "Examining Fairness in Graph-Based Collaborative Filtering: A Consumer and Producer Perspective". In: *Proceedings of the 13th Italian Information Retrieval Workshop (IIR 2023), Pisa, Italy, June 8-9, 2023*. Ed. by Franco Maria Nardini, Nicola Tonellotto, Guglielmo Faggioli, and Antonio Ferrara. Vol. 3448. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 79–84.

[4] Paolo Sorino, **Vincenzo Paparella**, Domenico Lofù, Tommaso Colafiglio, Eugenio Di Sciascio, Fedelucio Narducci, Rodolfo Sardone, and Tommaso Di Noia. "A Pareto-Optimality-Based Approach for Selecting the Best Machine Learning Models in Mild Cognitive Impairment Prediction". In: *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2023, Honolulu, Oahu, HI, USA, October 1-4, 2023*. IEEE, 2023, pp. 3822–3827. DOI: 10.1109/SMC53992.2023.10394057.

[5] **Vincenzo Paparella**. “Pursuing Optimal Trade-Off Solutions in Multi-Objective Recommender Systems”. In: *RecSys ’22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*. Ed. by Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge. ACM, 2022, pp. 727–729. doi: 10.1145/3523227.3547425.

[6] **Vincenzo Paparella**, Vito Walter Anelli, Ludovico Boratto, and Tommaso Di Noia. “Reproducibility of Multi-Objective Reinforcement Learning Recommendation: Interplay between Effectiveness and Beyond-Accuracy Perspectives”. In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, 2023, pp. 467–478. doi: 10.1145/3604915.3609493.

[7] **Vincenzo Paparella**, Vito Walter Anelli, Franco Maria Nardini, Raffaele Perego, and Tommaso Di Noia. “Post-hoc Selection of Pareto-Optimal Solutions in Search and Recommendation”. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*. Ed. by Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos. ACM, 2023, pp. 2013–2023. doi: 10.1145/3583780.3615010.

[8] **Vincenzo Paparella**, Vito Walter Anelli, Franco Maria Nardini, Raffaele Perego, and Tommaso Di Noia. “Flex-MORe: A Flexible Multi-Objective Recommendation Framework”. In: *Under Review*. 2024.

[9] **Vincenzo Paparella**, Dario Di Palma, Vito Walter Anelli, Alessandro De Bellis, and Tommaso Di Noia. “Unveiling the Potential of Recommender Systems through Multi-Objective Metrics”. In: *Proceedings of the 14th Italian Information Retrieval Workshop, Udine, Italy, September 5-6, 2024*. Ed. by Kevin Roitero, Marco Viviani, Eddy Maddalena, and Stefano Mizzaro. Vol. 3802. CEUR Workshop Proceedings. CEUR-WS.org, 2024, pp. 119–122.

[10] **Vincenzo Paparella**, Dario Di Palma, Vito Walter Anelli, and Tommaso Di Noia. “Broadening the Scope: Evaluating the Potential of Recommender Systems beyond prioritizing Accuracy”. In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, 2023, pp. 1139–1145. doi: 10.1145/3604915.3610649.

[11] **Vincenzo Paparella**, Alberto Carlo Maria Mancino, Antonio Ferrara, Claudio Pomo, Vito Walter Anelli, and Tommaso Di Noia. “Knowledge Graph Datasets for Recommendation”. In: *Proceedings of the Fifth Knowledge-aware and Conversational Recommender Systems Workshop co-located with 17th ACM Conference on Recommender Systems (RecSys 2023), Singapore, September 19th, 2023*. Ed. by Vito Walter Anelli, Pierpaolo Basile, Gerard de Melo, Francesco Maria Donini, Antonio Ferrara, Cataldo Musto, Fedelucio Narducci, Azzurra Ragone, and Markus Zanker. Vol. 3560. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 109–117.

[12] **Vincenzo Paparella**, Claudio Pomo, Vito Walter Anelli, Ludovico Boratto, and Tommaso Di Noia. “A Framework for Hyper-parameter Tuning Sensitivity Analysis in Recommender Systems Considering Multiple Objectives”. In: *To submit to the Information Processing and Management (IPM) journal*. 2024.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recommender Systems [153] (RSs) are ubiquitous in our daily digital lives. Ordinary people engage with RSs unconsciously while shopping on platforms like Amazon [115], streaming movies on Netflix [23], or listening to music on Spotify [78]. Indeed, RSs alleviate users from the *information overload* problem, in which people are overwhelmed by the massive amount of data they are exposed to on the World Wide Web. For instance, a user may prefer shopping on Zalando over Asos—or vice versa—based on how effectively each platform helps her discover desired items. RSs help online platforms to reach this situation by learning the user profile and suggesting a ranked list of products or services tailored to individual interests. Consequently, numerous research efforts have been made to develop adequate recommendation algorithms, from similarity-based solutions [151, 156] to approaches that rely on machine and deep learning techniques [84, 105, 122].

Despite the diversity of paradigms available in the realm of RSs, they commonly share the same predominant purpose: to guide users to relevant items in the context of information overload [38, 86]. Hence, the core way of operationalizing this purpose is to learn from data a function that accurately predicts the relevance of an item for each user. Indeed, traditional RSs aim to learn users' preferences by minimizing the rating prediction errors or maximizing the users' recommendation accuracy. Consequently, the accuracy of recommendations is considered the gold standard for measuring the effectiveness of RSs. Generally, an RS is deemed better than another if it consistently suggests more relevant items to users, achieving overall superior accuracy performance.

Although providing accurate suggestions to users is crucial, it has been argued that "being accurate is not enough" for an RS success [126]. For instance, recommending only Swedish House Mafia tracks to their fans might achieve high accuracy. However, the same recommendation could be perceived as obvious to these users, thus diminishing its value for them. Potential adverse results of this issue are user satisfaction erosion and the creation of filter bubbles [181].

Additionally, the end-user is not the only stakeholder involved in the recommendation process [2]. While end-users desire personalized and relevant recommendations, online platforms aim to achieve broader business objectives such as increasing sales, user retention, or content consumption. Simultaneously, item providers, such as artists or content creators, are directly affected by the exposure and ranking of their offerings [164]. If a recommendation algorithm predominantly emphasizes connections to popular items, this approach might unintentionally neglect individuals pursuing careers in niche fields [4]. Consequently, a limited equal visibility for these items could negatively impact society. An analogous issue can occur also on the customer side. Unfairness in RSs can lead to unequal utility distribution, where certain consumers/users are privileged by receiving recommendations with more quality [61]. This situation could also lead to discrimination when the algorithm tends to disadvantage user groups based on demographic characteristics such as ethnicity, gender, age, or socioeconomic status [10].

These observations have led researchers to focus on the beyond-accuracy perspectives of the recommendation problem. On the one hand, diversity and novelty of recommendations have been identified as crucial to shape better the user experience [98, 184]. On the other hand, extensive research has been conducted on how RSs may harm consumer and provider fairness [36, 55, 59]. Chapter 2 provides a brief but comprehensive overview of multiple perspectives of recommendations treated within this dissertation.

Overall, while predicting the relevance of individual items for users remains a central and significant challenge, focusing solely on a single objective, i.e., prediction accuracy and its associated metrics, may overly simplify the problem, thereby limiting the practical impact of academic research. Despite registering a growing interest in beyond-accuracy perspectives, many works focus exclusively on accuracy and a limited number of beyond-accuracy dimensions. This limitation applies to both the evaluation and optimization of RSs.

On the evaluation side, accuracy is consistently prioritized over the other facets of recommendation. RSs are mainly ranked according to their comparison of accuracy metrics, albeit some computed beyond-accuracy metrics sometimes accompany this evaluation. This singular emphasis on accuracy not only constrains the understanding of the full potential of RSs on multiple perspectives but also shapes other facets of evaluation within the field, such as hyper-parameter tuning and selecting the best model. In this regard, the *multi-objective evaluation* paradigm can effectively audit several objectives simultaneously without prioritizing one. However, this paradigm is currently overlooked in the recommendation research, lacking rigorous methodologies to measure RS performance quantitatively.

On the optimization side, a growing interest has emerged in *Multi-Objective Recommender Systems* (MORSs) [89, 218, 224]. These systems are built by blending multiple objectives through multi-objective optimization [224]. Therefore, research efforts focus on designing (differentiable) loss functions for specific objectives or optimization problems where several objectives are considered through constraints. However, the published works about MORSs are less than scientific papers about traditional RSs that improve accuracy performance. Indeed, many challenges and limitations in developing MORSs remain unsolved [224].

1.1 Thesis Statement

This dissertation advances the field of Recommender Systems (RSs) by proposing novel methodologies for their multi-objective evaluation and optimization, moving beyond the traditional focus on accuracy to enhance beyond-accuracy perspectives such as diversity, novelty, and fairness. The contents of this thesis are organized into four parts. The second and third parts include the main research contributions of this work. They are constituted of chapters reporting notions, analyses, and proposals devised from the scientific papers to which they refer.

The first part includes two chapters. These chapters provide helpful background notions and definitions regarding RSs and Multi-Objective Optimization (MOO). MOO-related notions lay the foundational concepts for multi-objective evaluation and optimization of RSs, e.g., the formal definition of MOO problems and Pareto optimality.

The second part focuses on methodologies for multi-objective evaluation of RSs, spanning into three chapters. Chapter 4 uses the lack of analysis in the literature on fairness aspects of graph-based collaborative filtering approaches as a showcase to conduct a qualitative evaluation exploiting Pareto frontiers. In this regard, Chapter 5 makes a step forward by employing quality indicators of Pareto frontiers from MOO theory to quantitatively unveil the beyond-accuracy perspectives of RSs without prioritizing accuracy. Finally, Chapter 6 provides a rigorous analytical framework to assess the sensitivity to hyper-parameter tuning of RSs in a multi-objective scenario.

The third part addresses some challenges and limitations of the current development of Multi-Objective RSs (MORSs). Chapter 7 highlights some practical challenges in the design and reproducibility of MORSs through a reproducibility study. From these observations, given the frequent detail omission about the selection strategy of the best model in MORS works, Chapter 8 proposes a theoretically justified technique to select Pareto optimal solutions specifically tailored to the recommendation task. Finally, Chapter 9 proposes a flexible multi-objective recommendation framework that injects an objective-agnostic and objective scale-aware additional loss function term in the training of an RS.

To end, Chapter 10 concludes this thesis by synthesizing the insights and advancements discussed within this dissertation.

1.2 Research Contributions

This section provides a concise yet comprehensive overview of the research contributions presented in this thesis, organized by thematic chapters. For each chapter, a brief summary of the content is provided, along with details about the related publications and the role of the Ph.D. candidate, Vincenzo Paparella, in the contributions to these works.

1.2.1 Ch. 4: Assessing Consumer and Provider Fairness in Graph Collaborative Filtering through Pareto frontiers

Contributions. Chapter 4 explores the evaluation of graph-based recommendation models, addressing a critical gap in the literature regarding the analysis of fairness in graph-based Collaborative Filtering (CF) approaches, which are predominantly assessed on accuracy metrics. We compare the performance of graph-based CF models against two classical CF baselines using consumer and provider fairness metrics within a single-objective evaluation framework. Our findings reveal that the superior accuracy of graph-based CF models often comes at the expense of user fairness, item exposure, and the equilibrium between these dimensions. To provide a more comprehensive perspective, we introduce a novel taxonomy for graph-based CF, identifying node representation and neighborhood exploration as the two primary dimensions influencing fairness and accuracy. Their individual and combined impacts are systematically analyzed. Shifting to a multi-objective evaluation paradigm, we leverage Pareto frontiers to visualize trade-offs between competing objectives across various hyper-parameter configurations. Specifically, we examine three 2-dimensional spaces: accuracy vs. item exposure, accuracy vs. user fairness, and item exposure vs. user fairness. The use of Pareto frontiers facilitates a qualitative analysis that simultaneously considers beyond-accuracy objectives and their interplay with recommendation relevance, offering deeper insights into the complex dynamics of multi-objective evaluation in recommender systems.

Publications. The chapter covers the topic explored in “Auditing Consumer- and Producer-Fairness in Graph Collaborative Filtering” [15], published at the 45th European Conference on Information Retrieval (ECIR 2023). A condensed version of the work has been published at the 13th edition of the Italian Information Retrieval Workshop (IIR) 2023 in the discussion paper titled “Examining Fairness in Graph-Based Collaborative Filtering: A Consumer and Producer Perspective” [58].

Ph.D. Candidate’s role. The Ph.D. candidate was responsible for designing and conducting the experiments and leading the analysis of trade-offs among accuracy, consumer fairness, and item exposure. Additionally, the candidate contributed to the writing and development of the paper.

1.2.2 Ch. 5: Quality Indicators of Pareto frontiers for Multi-Objective Evaluation of Recommendations

Contributions. In Chapter 5, we highlight that traditional evaluation of Recommeder Systems (RSs) consistently prioritizes their accuracy. Indeed, the accuracy is the primary metric for selecting the best-performing model. With a motivating example, we show that this procedure undermines a comprehensive understanding of the potential of RSs across beyond-accuracy perspectives. Hence, we broad RS evaluation by introducing a multi-objective evaluation that leverages Pareto frontiers formed by different hyper-parameter model configurations assessed under multiple dimensions. In contrast to the previous chapter, we enable a quantitative approach by employing the Quality Indicators from the multi-objective optimization theory to evaluate the Pareto frontiers. The experiments reveal that this multi-objective evaluation overturns the ranking of performance among RSs.

Publications. The chapter covers the topic explored in "Broadening the Scope: Evaluating the Potential of Recommender Systems beyond prioritizing Accuracy" [191], published and presented at the 17th ACM Conference on Recommender Systems (RecSys 2023). A condensed version of the work has been published at the 14th edition of the Italian Information Retrieval Workshop (IIR) 2024 in the discussion paper titled "Unveiling the Potential of Recommender Systems through Multi-Objective Metrics" [190].

Ph.D. Candidate's role. The Ph.D. candidate is the corresponding author of all the referenced papers [190, 191].

1.2.3 Ch. 6: Sensitivity of Recommender System to Hyper-parameter Tuning in Multi-objective Scenarios

Contributions. In Chapter 6, we address the challenge of understanding how Recommender Systems (RSs) performance can be affected by hyper-parameters tuning, particularly when considering multiple objectives. As RSs are increasingly required to balance accuracy with other important factors such as fairness, diversity, and novelty, it becomes crucial to understand the sensitivity of these systems to hyper-parameters adjustments. While the development of Multi-Objective Recommender Systems (MORSs) offers a practical framework for balancing these diverse objectives, online platforms cannot afford to completely overhaul their existing systems by implementing MORSs from scratch. Instead, they must find ways to incorporate beyond-accuracy objectives into their current models with minimal disruption. To address this challenge, we investigate the sensitivity of existing RS models to hyper-parameter tuning under different trade-offs between accuracy and beyond-accuracy objectives, such as novelty, diversity, and bias mitigation. We propose a novel evaluation framework that utilizes Pareto optimality to assess the robustness of model performance under varying hyper-parameters configurations. Through this framework, we identify how different hyper-parameters influence the consistency of achieving Pareto optimal solutions and examine the level of precision required in tuning to maintain performance. This contribution provides a deeper understanding of the stability of RS models in multi-objective contexts. It offers insights into how hyper-parameters tuning can be more effectively managed to balance multiple objectives without incurring excessive computational costs.

Publications. The chapter covers the topic explored in "A Framework for Hyper-parameter Tuning Sensitivity Analysis in Recommender Systems Considering Multiple Objectives", a paper to sumbit to the Information Processing and Management

(IPM) journal.

Ph.D. Candidate's role. The Ph.D. candidate is the corresponding author of the paper under review.

1.2.4 Ch. 7: A Reproducibility Study of Multi-Objective Recommendation

Contributions. In Chapter 7, we examine the reproducibility landscape of Multi-Objective Recommender Systems (MORSs) and find that most published studies in this domain are not accompanied by accessible source code or datasets, rendering replication and validation of their findings challenging. To deal with this issue, we select and reproduce a state-of-the-art MORS study, aiming to investigate the barriers to reproducibility and identify shortcomings in existing research practices. Our analysis reveals critical limitations in the experimental design of the reproduced study. Notably, the criteria for selecting the best-performing models are not mentioned, and performance evaluations are confined to the objectives explicitly optimized within the MORS framework. These limitations highlight the need for more comprehensive and transparent methodologies in MORS research. Our experiments led to new insights into the challenges of MORSs, including issues related to perspective trade-offs, the inherent difficulty of managing multiple conflicting objectives, and the recognition that recommendations are inherently multi-sided, affecting various stakeholders differently.

Publications. The chapter covers the topic explored in "Reproducibility of Multi-Objective Reinforcement Learning Recommendation: Interplay between Effectiveness and Beyond-Accuracy Perspectives" [188], published and presented at the 17th ACM Conference on Recommender Systems (RecSys 2023).

Ph.D. Candidate's role. The Ph.D. candidate is the corresponding author of the referenced paper [188].

1.2.5 Ch. 8: A novel strategy to select Pareto Optimal Solutions in Multi-Objective Recommendation Problems

Contributions. In Chapter 8, we recognize the evolution of the recommendation task from optimizing a single objective to addressing multi-objective problems, which yield a set of Pareto optimal solutions. Although the need to identify a single Pareto optimal solution for deployment is critical, no strategies specifically tailored to the unique requirements of Recommender Systems (RSs) have been proposed. This chapter bridges this gap by introducing "Population Distance from Utopia" (PDU), a novel, post-hoc, and theoretically grounded strategy for selecting a single Pareto optimal solution from the Pareto frontier in the context of RSs. Unlike conventional methods derived from multi-objective optimization theory, which rely solely on mean performance values across objectives, PDU establishes a utopia point for each individual sample in the dataset. This fine-grained approach allows for a "calibrated" selection process that considers not only the overall ("global") performance of an RS model but also the distribution and consistency of its performance at the sample level across multiple quality criteria. We conduct a comprehensive qualitative and empirical evaluation of PDU, comparing it to state-of-the-art selection strategies. Our results demonstrate that the innovative formulation and calibration feature of PDU significantly influence the selection process.

Publications. The chapter covers the topic explored in "Post-hoc Selection of Pareto-Optimal Solutions in Search and Recommendation" [189], published and presented at the 32nd ACM Conference on Information and Knowledge Management (CIKM 2023).

Ph.D. Candidate's role. The Ph.D. candidate is the corresponding author of the referenced paper [189].

1.2.6 Ch. 9: A Flexible Framework for Multi-Objective Recommendation

Contributions. Chapter 9 introduces Flex-MORe, a Flexible multi-objective recommendation framework that extends the training of Recommender Systems (RSs). While state-of-the-art multi-objective RSs employing scalarization approaches can address specific recommendation scenarios effectively, they often lack generalization. A key limitation lies in their tendency to overlook the scale of the loss functions, leading to potential dominance by the objective with the largest scale of values. Flex-MORe overcomes these limitations by incorporating an objective-agnostic and scale-aware loss function in the training procedure of a recommendation baseline. To achieve objective agnosticism, Flex-MORe introduces a novel smoothing approach that renders ranking-based metrics differentiable, enabling their seamless integration into the training process of RS models. Through extensive experimental analysis, we demonstrate that Flex-MORe achieves state-of-the-art performance while effectively balancing diverse objectives. Additionally, the framework offers the flexibility to adjust the weights of objectives in the loss function, allowing fine-grained control over their influence while maintaining competitive accuracy.

Publications. The chapter covers the topic explored in "Flex-MORe: A Flexible Multi-Objective Recommendation Framework", a paper currently under review.

Ph.D. Candidate's role. The Ph.D. candidate is the corresponding author of the paper under review.

1.3 Bibliographical Notes

This section outlines the research articles published during the Ph.D. that are not extensively discussed in the dissertation. These works emerged as parallel investigations, addressing research questions identified while exploring the broader literature.

Closely aligned with the topic of this dissertation, the Ph.D. candidate participated in the Doctoral Symposium at the 16th ACM Conference on Recommender Systems (RecSys 2022), presenting the paper *"Pursuing Optimal Trade-Off Solutions in Multi-Objective Recommender Systems"* [187]. This paper encapsulates the initial Ph.D. proposal developed during the first year of study, with its stated research questions subsequently examined in Chapters 6, 8, and 9 of this dissertation.

Two additional papers were co-authored in the broader domain of recommender systems. The first paper, *"An Analysis of Local Explanation with LIME-RS"* [12], was presented at the 12th Italian Information Retrieval (IIR) Workshop in 2022. This work focuses on explanations of recommendations, specifically analyzing the post-hoc approach of LIME-RS. The study reveales that explanations generated by local surrogate models often lack consistency with user and item characteristics, potentially leading to explanations that are ineffective or misaligned with user needs.

The second, *"Knowledge Graph Datasets for Recommendation"* [192], was published in the 5th Knowledge-aware and Conversational Recommender Systems Workshop, co-located with the 17th ACM Conference on Recommender Systems (RecSys 2023). This paper introduces two enriched versions of the Movielens25M and LibraryThing datasets. A novel linking methodology is developed to map items in these datasets to entities in the DBpedia, Wikidata, and Freebase Knowledge Graphs (KGs). Then, we retrieve the triples having these entities as subjects up to two hops. The linking methodology is then rigorously evaluated against a state-of-the-art entity linker, highlighting its effectiveness.

Finally, the candidate applied multi-objective optimization theory expertise to the Mild Cognitive Impairment (MCI) prediction task. In the paper *"A Pareto-Optimality-Based Approach for Selecting the Best Machine Learning Models in Mild Cognitive Impairment Prediction"* [166], published in the IEEE International Conference on Systems, Man, and Cybernetics (SMC 2023), the authors propose a novel method for selecting classifiers. The approach evaluates classifiers based on accuracy and their ability to correctly identify MCI subjects, demonstrating the value of Pareto optimality in medical predictive modeling.

Part I
Background

Chapter 2

Recommender Systems

Recommender Systems (RSs) are a class of software tools and algorithms designed to assist *users* in identifying *items* of interest by predicting their preferences [153]. By analyzing past user *interactions*, demographic data, or contextual information, these systems aim to deliver personalized recommendations across a wide range of domains, such as e-commerce, streaming services, and social networks. This definition reveals that RSs can exploit different kinds of data to produce their recommendations. However, three fundamental components devise any recommendation data:

- **Users**: the set of individuals or entities that receive and consume the recommendations. Users may have associated metadata, such as demographic information, preferences, or historical behavior.
- **Items**: the objects or services available for recommendation. An item can represent products, movies, songs, or any entity relevant to the application domain. Each item may have associated features or descriptive attributes (e.g., price, genre, or keywords).
- **Interactions**: the set of observable relationships between users and items, representing user behavior or preferences. The interactions, also called *transactions*, can be expressed by *explicit* and *implicit* feedback. On the one hand, explicit feedback is in the form of binary like/dislike relation or discrete values in a defined range (e.g., $\{1, \ldots, 5\}$). On the other hand, implicit feedback usually catches only positive feedback (e.g., purchases, plays, clicks) without any information about what the user dislikes. For this reason, this kind of rating is usually also called unary rating. Interactions can be represented as a *matrix*, where each entry denotes the user's observed preference or interaction value with the item.

2.1 Definition of the Recommendation Problem

Formally, let $\mathcal{U}$ be the set of users in the system and $\mathcal{I}$ the set of items in the catalog. Then, $\mathrm{R} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ is the user-item preference matrix, where $r_{u,i} \in \mathrm{R}$ contains either

$$\begin{bmatrix} 2 & 4 & \dots & 3 & 5 \\ 3 & 3 & \dots & 5 & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 3 & 4 & \dots & 5 & 2 \\ 2 & 3 & \dots & 2 & 5 \end{bmatrix}$$

(a) Full user-item matrix.

$$\begin{bmatrix} 2 & ? & \dots & 3 & 5 \\ 3 & ? & \dots & ? & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 3 & ? & \dots & 5 & 2 \\ 2 & 3 & \dots & ? & ? \end{bmatrix}$$

(b) Missing ratings.

Figure 2.1. An example of rating prediction. On the left, we have a full user-item matrix R that holds the rating for each user *u* and each item *i* at $R_{u,i}$. On the right, we have a real case in which some ratings are missing, and the recommendation task is defined as the prediction of these missing ratings.

$$\begin{bmatrix} 1 & 0 & \dots & 1 & 1 \\ 1 & 0 & \dots & 0 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & \dots & 1 & 1 \\ 1 & 1 & \dots & 0 & 0 \end{bmatrix}$$

(a) Full user-item matrix.

$$\begin{bmatrix} 1 & ? & \dots & 1 & 1 \\ 1 & ? & \dots & ? & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & ? & \dots & 1 & 1 \\ 1 & 1 & \dots & ? & ? \end{bmatrix}$$

(b) Missing interactions.

Figure 2.2. An example of interaction prediction. On the left, we have a user-item matrix R that holds the interaction for user *u* and item *i* at $R_{u,i}$. On the right, some interactions are missing, and the recommendation task is defined as the prediction of top-*k* items.

explicit or implicit feedback of user *u* for the item *i*. Figure 2.1a shows an example of a user-item matrix with explicit feedback, while Figure 2.2a visualizes the case of an implicit user-item matrix. In real-world scenarios, obtaining a complete user-item matrix is not feasible, especially as the size of the item catalog grows [86]. Therefore, a recommendation algorithm's primary objective is to find a utility function to predict which items a user might prefer without directly asking the user for this information [156].

Definition 2.1 (Recommendation Problem)**.** *Given a utility function* $f : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$, *the recommendation problem is defined as:*

$$\forall u \in \mathcal{U}, i' = \arg\max_{i \in \mathcal{I}} f(u, i), \tag{2.1}$$

where $i' \in \mathcal{I} / \mathcal{I}_u^+$ *is not in the list of (positive) already consumed items* $\mathcal{I}_u^+$ *by the user u.*

The formal definition above can vary according to the type of utility function that underlines the RS. In the case of explicit feedback, the utility function typically generates predicted ratings for each user-item pair, aiming to reconstruct the rating matrix (see Figure 2.1b). Here, the primary goal of the RS is to estimate the rating a user would give to an item, making the recommendation task essentially a *rating prediction* task. In contrast, for implicit feedback, the utility function focuses on predicting missing interactions rather than ratings (see Figure 2.2b) and sorting these interactions in descending order of assigned scores. Therefore, the recommendation task becomes a *top-k prediction* task, where an ordered list of k un-interacted items is generated. However, it becomes evident that these tasks may overlap. Indeed, the top-k recommendation task can be seen as a natural progression after the rating prediction task, where items are reordered for each user based on their predicted scores.

In conclusion, RSs, in their most basic form, can be described as tools designed to generate *ranked* lists of items to users. These systems utilize explicit or implicit data reflecting user preferences, often augmented with supplementary information about items, other users, contextual factors, and the user's historical interactions to produce personalized recommendations.

Historically, recommendation tasks were initially focused on explicit feedback settings. Solving the rating prediction task proved highly successful and dominated the field for many years. The research community retained this setting up to the Netflix Prize competition [23], where the video streaming service offered a one million USD prize to the team that could minimize the Root Mean Square Error by the end of the competition. However, obtaining a suitable dataset with explicit ratings can be challenging, as repeatedly asking users to rate items may negatively impact the user experience. Moreover, research has indicated that ratings obtained from online systems do not always reliably reflect users' actual preferences for items [126]. These findings have led to adopting the implicit feedback setting and the top-k prediction task [49], the predominant objective in RSs today.

2.2 Recommendation Techniques

The Recommender Systems (RSs) field has witnessed the development of various methodologies, many of which have emerged as state-of-the-art solutions within the research community. These methodologies address the recommendation problem, each grounded in distinct assumptions about user behavior. For example, some approaches hypothesize that users evaluate new items based on their similarity to previously consumed items. In contrast, others suggest that the preferences or behaviors of similar users influence user decisions. Broadly, these techniques are categorized into three main paradigms: (i) *collaborative filtering*, (ii) *content-based filtering*, and (iii) *hybrid methods* [153]. The following sections provide a concise yet comprehensive overview of these recommendation approaches and their classification.

2.2.1 Collaborative Filtering Approaches

Collaborative Filtering (CF) remains one of the most widely adopted paradigms in RSs, leveraging the principle that "users who have agreed in the past are likely to agree in the future" [151]. This principle implies that a user's rating for a new item will likely align closely with the ratings of other users with similar interaction patterns. A key advantage of CF is its independence from auxiliary data sources, such as item attributes, relying solely on user-item interaction data. However, its performance is highly dependent on the availability and density of these interactions, making it susceptible to sparsity and cold-start challenges.

CF methods are broadly categorized into *memory-based* and *model-based* approaches [153]. Memory-based methods use historical user-item interaction data to directly predict unknown ratings, while model-based methods aim to learn latent representations of users and items to make predictions.

Memory-based Collaborative Filtering

Memory-based CF algorithms operate directly on the user-item interaction matrix, typically using similarity measures to identify relationships between users or items. The most prominent example is the k-Nearest Neighbors (kNN) algorithm, which computes a similarity matrix to determine the most similar entities (users or items). The specific schema of the kNN algorithm depends on whether the similarity is computed across users or items:

- **User kNN** [151]: the utility function f estimates the interaction score for a given user-item pair (u, i) based on the similarity between user u and other users who have interacted with item i:

$$f^{\text{User}k\text{NN}}(u,i) = \sum_{v \in \mathcal{U}_u^+} sim(u,v), \tag{2.2}$$

  where $sim(\cdot)$ is a similarity function and $\mathcal{U}_u^+$ is the set of users that interacted with the item i. This approach is particularly effective in capturing shared preferences among similar users. However, as the number of users in the system grows, the computational cost of similarity calculations increases, leading to scalability challenges.

- **Item kNN** [156]: to address the scalability limitations of user-based methods, the utility function f of item kNN focuses on item-item similarities. Here, the algorithm predicts the likelihood of interaction between user u and item i based on the similarity between item i and other items previously interacted with by u:

$$f^{\text{Item}k\text{NN}}(u,i) = \sum_{j \in \mathcal{I}_u^+} sim(i,j), \tag{2.3}$$

  where $sim(\cdot)$ is a similarity function and $\mathcal{I}_u^+$ is the set of items interacted by the user u. By shifting the computational focus from users to items, item kNN achieves greater scalability in systems with many users.

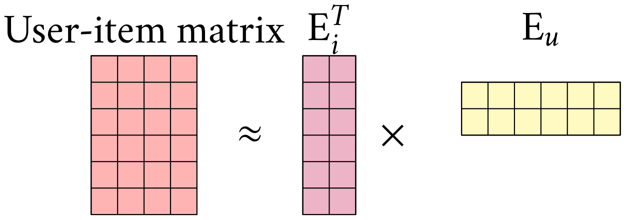


Figure 2.3. Exemplification of latent factor models. These model learn latent representations (embeddings) of users and items. Then, the user-item matrix is estimated by computing the dot product of each user-item embeddings pair.

While memory-based methods are intuitive and interpretable, their reliance on explicit similarity measures makes them sensitive to data sparsity and limits their ability to generalize in complex scenarios. These limitations have driven the development of more sophisticated, model-based approaches that can better capture latent structures in the data.

Model-based Collaborative Filtering

The advent of machine learning and deep learning has significantly advanced model-based Collaborative Filtering (CF), enabling these methods to dominate personalized recommendation tasks. This shift gained momentum following the Netflix Prize competition in 2006, where model-based approaches, particularly matrix factorization methods, demonstrated superior accuracy compared to memory-based counterparts [104]. Model-based CF approaches leverage sophisticated algorithms to uncover latent relationships within user-item interaction data, leading to more nuanced and effective recommendations.

Among the diverse techniques under the model-based approach, **Matrix Factorization** (MF) methods [105], often referred to as latent factor models, have emerged as the most prominent. These models analyze the user-item interaction matrix to identify latent factors, i.e., abstract features representing user preferences and item attributes. By learning these latent representations, MF captures complex user-item relationships that are not directly observable. Figure 2.3 exemplifies how latent factor models work.

In MF, users and items are represented as embeddings in a shared latent space, with their interactions modeled through these embeddings. Formally, let $\mathrm{e}_u \in \mathrm{R}^d$ and $\mathrm{e}_i \in \mathrm{R}^d$ represent the d-dimensional latent embeddings of user u and item i, respectively, with $d << |\mathcal{U}|$ and $d << |\mathcal{I}|$. The matrices $\mathrm{E}_u \in \mathbb{R}^{|\mathcal{U}| \times d}$ and $\mathrm{E}_i \in \mathbb{R}^{|\mathcal{I}| \times d}$ aggregate these embeddings for all users $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, respectively. These embeddings constitute the learnable parameters Θ. Hence, the utility function f to estimate the interaction score for a given user-item pair (u, i) is:

$$f^{\mathrm{MF}}(u, i) = \mu + b_u + b_i + \mathrm{e}_i^T \mathrm{e}_u, \tag{2.4}$$

where μ represents the global average score, b_u, and b_i are the biases associated with user u and item i, respectively, and $\mathrm{e}_i^T \mathrm{e}_u$ is the dot product of the user and item embeddings.

Building on the success of MF, numerous extensions have been proposed to enhance its flexibility and predictive power. For example, Neural Collaborative Filtering (NCF) [85] replaces the dot product operation with a multi-layer perceptron, enabling the model to learn more complex interaction patterns. Conversely, SimpleX [122] introduces a simplified framework that focuses on explicit disentanglement of user preferences and item characteristics, improving interpretability and efficiency. These extensions share the fundamental principle of learning latent user and item representations but differ in how the interaction score is modeled through the utility function f.

Despite their success, model-based CF methods face several challenges. Firstly, training and inference become computationally expensive as the number of users and items grows, thus showing scalability issues. Secondly, these methods struggle with new users and items lacking sufficient interaction data, facing the cold-start problem. Finally, complex models risk overfitting sparse data, necessitating robust regularization techniques.

2.2.2 Content-based Approaches

Content-Based Filtering (CBF) methods leverage item and user attributes to generate personalized recommendations by aligning the characteristics of items with the preferences of target users [140]. The central premise is that recommendations can be derived by analyzing content information that describes the items and their properties. However, effectively characterizing items with rich, structured content is often non-trivial. Traditionally, item attributes have been represented as simple *keywords* extracted from metadata or textual descriptions. In recent years, the research community has increasingly adopted *concept-based* approaches, where items are characterized through semantic representations derived from structured knowledge sources such as Wikipedia, DBpedia, and Freebase. These methods enable a deeper understanding of the items' meaning beyond surface-level keywords.

The process of content-based recommendation typically involves three key components [153]:

- **Content Analyzer**: this module processes item-related information from diverse sources and represents the items within a specific description space (e.g., a vector space model or semantic embeddings).
- **Profile Learner**: this component gathers user preference data (e.g., ratings, interaction history) and uses probabilistic techniques, relevance feedback mechanisms, or similarity-based methods to construct and generalize user profiles. The profiles encapsulate the users' preferences in the same feature space as the items.
- **Filtering Component**: given the user profile, this module matches the user-learned representations with item descriptions to suggest the most relevant items, thereby producing the lists of recommendations.

Thanks to their design, CBF offers several compelling advantages. CBF systems can inherently explain recommendations by highlighting item attributes that align with the user's preferences, thereby increasing trust and interpretability. In addition, unlike collaborative filtering approaches, CBF systems do not require historical user-item interactions to make recommendations, allowing them to handle new items effectively and solve cold-start problems. However, CBF methods tend to overfit user preferences, leading to a lack of diversity or serendipity, as the suggested items are too similar to those previously consumed. This problem is often named *overspecialization*. Moreover, the quality of recommendations heavily depends on the availability and richness of structured knowledge about the domain. In scenarios where semantic or content-rich data is unavailable, the performance of CBF methods can degrade. In this regard, we have provided the augmentation of two well-known datasets, i.e., Movielens25M and LibraryThing [192]. The items in the dataset have been linked with their corresponding entities in the Wikidata and DBpedia knowledge graphs. Then, we retrieved triples from DBpedia and Wikidata up to two hops, enabling the collection of structured information linked to these resources.

2.2.3 Hybrid Approaches

The limitations inherent in Collaborative Filtering (CF) and Content-Based Filtering (CBF) methods have motivated the development of hybrid recommendation approaches, combining the strengths of both techniques to address their weaknesses. Hybrid recommenders aim to enhance recommendation accuracy, robustness, and diversity by synergistically leveraging collaborative signals and content information.

Several strategies can be employed to design hybrid recommender systems. These include [9]:

- **Parallel integration**: CF and CBF methods are implemented independently, and their results are combined using an aggregation function (e.g., weighted sum, ranking fusion, or ensemble techniques) to produce the final recommendations.
- **Sequential integration**: one approach is used to pre-filter or post-filter the candidate recommendations generated by the other. In other words, some CBF characteristics are incorporated into CF approaches or vice versa.
- **Unified models**: a single model is designed to exploit collaborative and content information simultaneously.

By effectively merging these information sources, hybrid approaches can address key challenges of both CF and CBF approaches, such as the cold-start problem, the overspecialization issue, and scalability drawbacks. Overall, hybrid recommendation systems represent a powerful paradigm that balances the advantages of multiple techniques, offering flexibility and improved performance across diverse domains and scenarios.

2.3 Recommendation Pipeline

Practical Recommender Systems (RSs) development relies on a well-defined workflow that organizes and streamlines the key stages of model creation, training, and evaluation. Given the diversity of recommendation strategies and the growing complexity of modern approaches, a standardized pipeline is critical to ensure reproducibility and fair comparison across models [51, 176].

A typical recommendation pipeline, especially for model-based approaches, encompasses three primary phases: (i) *input pre-processing*, (ii) *model optimization*, and (iii) *model evaluation*. In the following, we briefly describe each stage.

2.3.1 *Input Pre-processing*

The input pre-processing phase is a critical foundation for any RS, as input data quality, consistency, and structure directly influence the model's accuracy and performance. This stage primarily involves collecting raw data comprising user-item interactions, transforming them into a structured format (e.g., user-item interaction matrices), and preparing additional metadata when applicable. Metadata, such as item attributes or user profiles, are essential for content-based and hybrid recommendation systems. Additionally, some pre-processing operations may be applied to the user-item interaction data.

Converting Explicit Feedback into Implicit Signals

As highlighted in section 2.1, the interactions can be categorized into two types: *explicit* feedback (e.g., numerical ratings) and *implicit* feedback (e.g., clicks, purchases, or views). Hence, when the utility function f is designed for a top-k recommendation task, the pre-processing step often involves converting eventual explicit ratings into implicit signals. From a visual perspective, this operation reshapes a user-item matrix as illustrated in Figure 2.1b into a user-item matrix exemplified in Figure 2.2b. For example, explicit ratings are converted into binary signals based on a predefined threshold:

$$y_{u,i} = \begin{cases} 1 & \text{if } r_{u,i} \geq \tau, \\ 0 & \text{otherwise,} \end{cases}$$

where $r_{u,i}$ represents the explicit rating provided by user u for item i, and τ is the threshold. Ratings equal to or higher than τ indicate a positive interaction, while lower ratings are treated as non-interactions. It is worth noticing that setting $\tau = 1$ retains all the interactions.

P-core Filtering

To ensure data sparsity does not hinder model training and evaluation, p-core pre-processing is often applied. The p-core filtering technique retains only those users and/or items with a minimum number p of interactions. Formally, for a given interaction matrix, the pre-processed data satisfies the following condition:

$$\deg(u) \geq p \quad \text{and/or} \quad \deg(i) \geq p, \quad \forall\, u \in \mathcal{U},\ i \in \mathcal{I},$$

where $\deg(u)$ and $\deg(i)$ represent the number of interactions (degree) of user u and item i, respectively, and p is the minimum interaction threshold. This pre-processing strategy helps to avoid the cold start issue during model training, especially when dealing with this problem is out of the scope of the research. Indeed, users and items can be classified as *warm* or *cold* based on the predefined interaction threshold p. Specifically, warm users are those with more than p interactions, indicating they are the most active users on the platform. In contrast, cold users have fewer than p interactions, making them the least active. A similar definition applies to items: warm items are those with interactions exceeding p, reflecting their popularity, while cold items fall below this threshold, indicating limited engagement.

2.3.2 Model Optimization

Once pre-processed data is available, the focus shifts to building and optimizing the recommendation model. Specifically, this phase is involved solely in the case of model-based RSs, in which latent representations of users and items encompassing the parameters Θ of the model are learned (see section 2.2.1). The parameters Θ are learned by optimizing a loss function that reflects the recommendation task and the underlying assumptions.

For explicit feedback scenarios, the *Mean Squared Error* (MSE) loss [195] is commonly used to minimize the difference between predicted and true ratings:

$$\mathcal{L}_{\text{MSE}} = \sum_{(u,i)\in\mathcal{D}} (r_{u,i} - f(u,i))^2, \tag{2.5}$$

where $r_{u,i}$ denotes the actual rating for user u and item i, $\mathcal{D}$ is the set of observed interactions, and $f(u,i)$ is the predicted rating for the pair (u,i).

The Bayesian Personalized Ranking (BPR) loss function [149] is widely adopted for implicit feedback scenarios. BPR optimizes the ranking of items by encouraging that, for each user $u \in \mathcal{U}$, an observed (positive) item $i^+ \in \mathcal{S}^+$ is ranked higher than an unobserved (negative) item $i^- \in \mathcal{S}^- := \mathcal{I} \setminus \mathcal{S}^+$:

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i^+,i^-)\in\mathcal{D}} -\ln \sigma(f(u,i^+) - f(u,i^-)), \tag{2.6}$$

where $\sigma(\cdot)$ is the sigmoid function, and $f(u,i^+)$ and $f(u,i^-)$ are the predicted scores for the pairs of user u with the positive item i^+ and with the negative item i^-, respectively. It is evident that, when adopting a loss function like BPR, *negative sampling* strategies are needed, i.e., strategies to pair an observed item with an unobserved item. The common approach is to sample a subset of negative items for each user. Indeed, Rendle et al. [149] suggest sampling only one negative item for each pair of user-positive items for that user.

The availability of loss functions to minimize in the RS literature is not limited to the abovementioned criteria. Other examples are the Pairwise Hinge (PH), Binary Cross-Entropy (BCE), Softmax Cross-Entropy (SCE), and contrastive losses.

2.3.3 Model Evaluation

Once the recommendation algorithm computes scores for user-item pairs, the next critical step is inference. Inference involves generating recommendation lists tailored to individual users based on these scores. These lists are subsequently evaluated to assess the algorithm's effectiveness. Recommendation evaluation can be broadly categorized into two approaches: (i) offline evaluation, predominantly utilized in academic research, and (ii) online evaluation, primarily employed in industry settings. In this section, we focus on offline evaluation, as it is the framework used to validate the methodologies discussed in this dissertation. Specifically, we examine key aspects of offline evaluation, including (i) dataset splitting, (ii) hyper-parameter tuning, and (iii) evaluation metrics.

Data Splitting

Following data pre-processing, the next step is to partition the dataset of user-item interactions into subsets designated for training, validation, and testing. Each subset serves a distinct purpose: the training set is used to fit the model, the validation set is employed for hyper-parameter tuning and early stopping, and the test set is used to assess model performance. Best practices in machine learning emphasize the importance of maintaining these separate datasets to ensure the internal validity of experimental results. However, studies [50, 172, 175] have shown that some offline evaluations bypass the validation set, directly tuning hyper-parameters or applying early stopping on the test set. Such practices risk inflating performance metrics and reducing reproducibility.

Below, we summarize the literature's most commonly used dataset splitting strategies.

Random Splitting. The majority of works adopts random splitting methods [95, 175], where a user's interactions are distributed across training, validation, and test sets in a randomized manner, often using predefined proportions (e.g., 80% for training, 10% for validation, and 10% for testing). While simple and computationally efficient, random splitting has a significant drawback. It disregards the temporal order of interactions, possibly resulting in scenarios where future interactions are used to predict past behaviors. Such an approach constitutes an unrealistic assumption in real-world settings. The risk is undermining the evaluation's internal validity unless we can assume that user preferences are entirely static and independent of time. These assumptions, however, rarely hold in real-world recommender systems.

Time-Aware Splitting. Time-aware splitting considers the temporal sequence of user interactions to address the limitations of random splitting. Under this approach, interactions occurring before a specific timestamp *t* are assigned to the training and validation sets, while interactions after *t* are reserved for testing. This method ensures that the test set reflects future interactions relative to the training data, aligning more closely with real-world scenarios.

Despite its advantages, time-aware splitting introduces its challenges. For instance, it may result in small sample sizes for specific users, as many users exhibit transient engagement with platforms, often remaining active for only short periods. This issue is particularly pronounced in long-period datasets, where users may contribute only a few interactions within the recorded time frame. Additionally, time-aware splitting requires precise timestamp information for each interaction, often unavailable in publicly accessible datasets, especially those used in academic research.

Hyper-parameter Tuning

Hyper-parameter tuning is a crucial aspect of developing and evaluating recommendation algorithms. However, it has been frequently observed that many offline evaluation experiments unfairly favor newly proposed algorithms by meticulously optimizing their hyper-parameters while neglecting to apply the same rigor to baseline models [48, 50, 162]. Furthermore, unlike other experimental details, such as datasets and data splits, the range of hyper-parameters explored and the methods used for tuning are rarely documented [162, 223]. This lack of transparency undermines the reproducibility and fairness of such comparisons.

Three hyper-parameter optimization strategies are utilized for offline recommendation experimentation: (i) *grid search*, (ii) *Bayesian optimization* using *Tree-structured Parzen Estimators*, and (iii) *random search*. Among these, grid search is the most widely reported in the literature [177]. Indeed, throughout this dissertation, we will mainly use this approach.

Grid search involves defining a finite set of candidate values for each hyper-parameter, combined to create a Cartesian product representing all possible configurations [65]. Each configuration corresponds to a unique set of hyper-parameters, and models are trained for all configurations. This exhaustive exploration ensures comprehensive coverage of the hyper-parameter space.

Once all models have been trained, the best-performing configuration should be chosen for at least two reasons. Firstly, only one optimized recommendation model is ultimately deployed in real-world applications. Secondly, the performance of the best models is reported within the research papers to be compared with the baselines.

It is critical that hyper-parameter tuning and the subsequent selection of the best-performing model are conducted using the validation set, which must remain separate from the test set used for reporting final experimental results. This separation ensures that the evaluation remains unbiased and prevents overfitting to the test data.

To identify the optimal configuration, a target metric must be defined. Typically, this metric reflects the recommendation algorithm's accuracy in generating relevant suggestions, such as nDCG@10. The best-performing model is thus the one that achieves the highest value for the chosen target metric on the validation set.

While offline hyper-parameter tuning is essential for identifying promising configurations, it is important to acknowledge its limitations. There is no guarantee that the hyper-parameters optimized for offline evaluations will also yield the best performance in online environments [220]. Offline experiments are, by nature, static and may not fully capture the dynamics of real-world interactions. As such, practitioners must remain cognizant of a recommendation algorithm's sensitivity to small changes in hyper-parameter values. This robustness can significantly influence the practical applicability of the algorithm.

Measuring the Relevance of Recommendations

The final step in designing an offline evaluation experiment in Recommender Systems (RSs) is selecting appropriate metrics to evaluate the generated recommendations using the test set. Metrics provide a quantitative means to assess the outputs of a recommendation algorithm concerning the stated objective. In the context of RSs, the common goal of helping users discover relevant items is typically operationalized as the ability to accurately rank items from most to least relevant for a user. Various accuracy-oriented metrics are employed to evaluate whether this objective has been achieved, depending on the specific recommendation task (e.g., rating prediction or top-k recommendation).

For rating prediction tasks, metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Square Error (RMSE) are commonly used. However, this section focuses on metrics designed to evaluate the top-k ranking prediction task, as these are more widely adopted in the literature and this dissertation. Top-k metrics are particularly well-suited to measuring the accuracy of ranking predictions, aligning closely with the practical goals of most RSs.

Accuracy metrics for top-k recommendation evaluate the presence of relevant items within the top-k entries of a recommended list, where k is typically chosen from the set $k \in \{1, 5, 10, 20, 50\}$. Then, these metrics apply a cutoff, focusing only on the first k items of the list when it contains more than k elements.

Given the set of users $\mathcal{U}$, the recommendation list $\mathcal{L}_u$ for the user u, its top-k items $\mathcal{L}_u^{(1,\dots,k)}$, and the relevant items in the test set $\mathcal{I}_u^{+,\text{test}}$, we outline the most commonly used accuracy-oriented metrics for evaluating top-k recommendation tasks below.

Definition 2.2 (Precision@k). *The precision (Precision@k) is defined as the average, over all the users, of the proportion of items in the top-k that are relevant to each user:*

$$Precision@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}_u^{(1,\dots,k)} \cap \mathcal{I}_u^{+,test}|}{k}. \tag{2.7}$$

Definition 2.3 (Recall@k)**.** *The recall (Recall@k) is the average of the proportion of relevant items retrieved in the top-k of each user:*

$$Recall@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\mathcal{L}_u^{(1,\ldots,k)} \cap \mathcal{I}_u^{+,test}|}{|\mathcal{I}_u^{+,test}|}. \tag{2.8}$$

Definition 2.4 (normalized Discount Cumulative Gain@k)**.** *The normalized Discount Cumulative Gain (nDCG@k) supposes that the items appearing earlier in the list are more valuable to users than those ranked lower:*

$$nDCG@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{IDCG@k} \sum_{i \in \mathcal{L}_u^{(1,\ldots,k)}} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{2.9}$$

where $rel_i = 1$ *if the item at rank* $i \in 1, \ldots, k$ *is relevant for the user, zero otherwise.* $IDCG@K = \sum_{i=1}^{k} \frac{1}{\log_2(i+1)}$ *is the ideal DCG, representing the best possible ranking of the k most relevant items.*

Definition 2.5 (Hit Ratio)**.** *The Hit Ratio (HR@k) measures the number of times that at least a relevant item is within a user's recommendation list:*

$$HR@k = \frac{|\mathcal{U}_{hits}@k|}{|\mathcal{U}|}, \tag{2.10}$$

where $|\mathcal{U}_{hits}@k|$ *is the number of users for whom a relevant item is included in* $\mathcal{L}_u^{(1,\ldots,k)}$. *The higher the value is, the more accurate the recommendations are.*

2.4 Multiple Perspectives of Recommendation

Recommender Systems (RSs) have traditionally focused on optimizing a single objective, i.e., providing relevant content to users. However, while accuracy is crucial for fostering user trust and engagement, overemphasizing relevance can lead to unintended consequences. On the one hand, adverse side effects entail users confined into filter bubbles [181], with limited novel or diverse suggested items. On the other hand, adverse results overlook the needs of other stakeholders, such as businesses and content creators. For instance, biases in item exposure can lead to unfair outcomes for content providers, promoting disparities in visibility. Hence, while predicting the relevance of individual items for users remains a central objective, this approach risks oversimplifying the problem and neglecting the multiple perspectives of recommendation. For this reason, many researchers have focused on enhancing these multiple perspectives, both from the optimization and evaluation points of view.

This section is devoted to briefly describing beyond accuracy aspects of recommendations that are treated within this dissertation and how to evaluate them. We categorize these aspects in (i) *user-centric* and (ii) *multi-stakeholder* objectives.

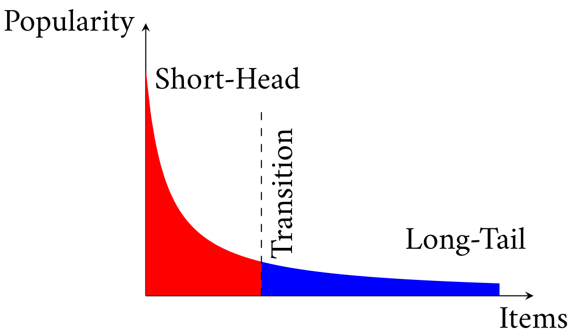


Figure 2.4. Illustration of a long-tail distribution. The curve demonstrates a small number of high-frequency items (the head) transitioning into a large number of low-frequency items (the tail). The dashed line marks the boundary between these regions.

2.4.1 *User-Centric Objectives*

User-centric objectives are quality metrics historically introduced to improve recommendations for the users who consume the suggestions. Indeed, designing algorithms that surface more relevant items in the recommendation list could not be valuable enough for the users. Evident and monotonous suggestions are tedious from the user's perspective, who needs to experience a broader and less widely known offer of items. In other words, the user should be presented with items that are not only relevant but also *diverse* and *novel*, respectively.

Novelty

The **novelty** of an item for a user refers to how different it is compared to what the user has previously seen or known. Some works distinguish novelty and *serendipity*, where an item is serendipitous if it is both novel and surprising [86, 126]. To better differentiate these two objectives, novelty is often defined in a user-independent manner. Indeed, item novelty is quantified as the inverse of its popularity, such as the number of ratings it has received [98]. Then, novel items belong to the *long-tail*, in contrast to popular items in the *short-head* (Figure 2.4).

Given the set of users $\mathcal{U}$, the recommendation list $\mathcal{L}_u$ for the user u, and its top-k items $\mathcal{L}_u^{(1,\dots,k)}$, in the following, we define two metrics to measure the novelty of recommendations, i.e., Expected Popularity Complement (EPC) [42] and Expected Free Discovery (EFD) [42].

Definition 2.6 (Expected Popularity Complement)**.** *Expected Popularity Complement (EPC) measures the expected number of relevant items belonging to the long-tail:*

$$EPC@k = \frac{c}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{i_k \in \mathcal{L}_u^{(1,\dots,k)}} p(seen \mid k, u, \mathcal{L}_u^{(1,\dots,k)})\, p(rel \mid i_k, u)\,(1 - p(seen \mid i_k)), \quad (2.11)$$

where c is a normalizing constant, and $p(seen \mid \cdot)$ and $p(rel \mid \cdot)$ are the probability of an *item to be seen and relevant, respectively. Higher values demonstrate a higher presence of long-tail relevant items.*

Definition 2.7 (Expected Free Discovery). *Expected Free Discovery (EFD) is a measure based on the expected inverse collection frequency that expresses the ability of an algorithm to recommend relevant long-tail items:*

$$EFD@k = \frac{c}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{i_k \in \mathcal{L}_u^{(1,\dots,k)}} p(seen \mid k, u, \mathcal{L}_u^{(1,\dots,k)})\, p(rel \mid i_k, u)\, (-log_2(p(i \mid seen)), \tag{2.12}$$

where c is a normalizing constant, and $p(seen \mid \cdot)$ and $p(rel \mid \cdot)$ are the probability of an item to be seen and relevant, respectively. $p(i \mid seen)$ reflects a factor of item popularity, whereby high novelty values correspond to long-tail items few users have interacted with, and low novelty values correspond to popular head items. Higher values demonstrate a higher presence of long-tail relevant items.

Diversity

In contrast to novelty, diversity generally applies to a set of items. The **diversity** of a set of items refers to how different the items are compared to each other. Generally, diversity is categorized into (i) *individual diversity* and (ii) *aggregate diversity*. Individual diversity accounts for how different the items in the recommendation list of a single user are. Conversely, aggregate diversity represents the total amount of different items a recommendation algorithm can provide to the whole set of users. Given the set of users $\mathcal{U}$, the recommendation list $\mathcal{L}_u$ for the user u, and its top-k items $\mathcal{L}_u^{(1,\dots,k)}$, we define two metrics to assess the aggregate diversity of recommendations, i.e., the Gini Index (Gini) [42] and the Item Coverage (IC).

Definition 2.8 (Gini Index). *Gini Index ($\hat{Gini}@k$) is a measure of aggregate diversity used to measure the distributional inequality, i.e., how unequally different items are chosen by users when a particular recommendation algorithm is used:*

$$\hat{Gini}@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{k-1} \frac{\sum_{i=1}^{|\mathcal{L}_u^{(1,\dots,k)}|} (2i - |\mathcal{L}_u^{(1,\dots,k)}|\ times(i))}{\sum_{i=1}^{|\mathcal{L}_u^{(1,\dots,k)}|} times(i)}, \tag{2.13}$$

where $times(i)$ is a function returning the number of times the item i appears in the recommendation lists. $\hat{Gini}@K$ is 0 when all items are equally chosen, while $\hat{Gini}@K$ is 1 when the recommender always selects the same item.

To adhere to the principle that higher is better, in the remainder of this dissertation, we will refer to the version Gini@$K = 1 - \hat{\text{Gini}}@K$.

Definition 2.9 (Item Coverage). *The Item Coverage (IC@k) computes the number of items that are shown to at least one user in the recommendation lists:*

$$IC@k = \left| \bigcup_{u \in \mathcal{U}} \mathcal{L}_u^{(1,\dots,k)} \right|. \tag{2.14}$$

A higher value of item coverage implies a higher diversity of the items within the recommendation lists.

2.4.2 Multi-Stakeholder Objectives

While end-user perspectives remain central in Recommender Systems (RSs), broadening the scope of evaluation to include additional stakeholders that can affect or be affected by the delivery of recommendations is crucial. Diverse types of stakeholders involved in the recommendation process are categorized in [2] as follows:

- **Consumers** represent the end-users who interact with the platform and receive recommendations. These individuals engage with the system to address their needs by expecting the recommendations to provide relevant and satisfactory solutions.
- **Providers** are entities associated with the recommended items. Their definition can vary depending on the domain and the focus of analysis. For instance, in the context of movie recommendations, providers could include production studios, directors, actors, or even the countries of production. Providers play a critical role in shaping the pool of recommended items and their perceived quality.
- **System** refers to the organization responsible for developing and maintaining the recommender platform. This entity serves as the intermediary that connects consumers with items, such as a retailer, e-commerce platform, broker, or another type of venue.

RSs ideally aim to "create value in parallel for all involved stakeholders," [93] with a value reflecting the "goodness" of recommendations from the perspective of each stakeholder. However, the nature of this value depends heavily on the context and domain in which the system operates, often aligning with either economic and business-related values or societal and human-centric values [22].

This discussion focuses on societal and human-centric values relevant to the stakeholders outlined earlier. These values emphasize the *fairness* of recommendations, which can manifest differently depending on the specific stakeholder being considered during evaluation. Closely tied to the notion of fairness is the critical issue of *algorithmic bias*, which plays a central role in assessing and ensuring equitable outcomes for all stakeholders.

Fairness

Understanding fairness in RSs is intricate as it requires considering a complex ecosystem of various entities and interconnected concepts. Nonetheless, we seek to provide a brief but comprehensive overview of fairness in RSs and how to evaluate it.

Fairness is crucial when deploying RSs, particularly when the risk of harmful discrimination arises. Fairness can be categorized according to the stakeholder it concerns [61]:

- **Consumer-Side Fairness**: this dimension focuses on ensuring that consumers are treated equally in both the quantitative and qualitative aspects of their interaction with the system.
- **Provider-Side Fairness**: this perspective addresses fairness for item providers, ensuring that the entities represented by the recommended items receive equitable treatment.

When consumer and provider fairness are considered simultaneously, we talk about *multi-sided fairness* [34]. Fairness is frequently conceptualized along the dimensions of *individual fairness* and *group fairness* [60]. On the one hand, *individual fairness* seeks to ensure that similar individuals are treated similarly, emphasizing fairness at the granular, individual level, focusing on consistency in treatment based on relevant characteristics. On the other hand, *group fairness* aims to achieve equitable treatment across groups, ensuring that no systemic disparities exist between them.

In the following, we provide some metrics to assess the recommendation performance of an algorithm under the lens of consumer and provider fairness.

Consumer Fairness. Here, we are interested in measuring the unfair distribution of the utility of recommendations among users. The idea is that the users on the platform should receive recommendations having the same quality in terms of relevance. This assessment could be performed in terms of individual fairness and group fairness. We start by measuring the individual consumer fairness with the variance of accuracy metric values over a set of users [210].

Definition 2.10 (Variance)**.** *Let Q_u be the quality of the recommendation provided to the user u measured using a suitable accuracy metric (e.g., nDCG) and let* Q *be the set containing Q_u $\forall u \in \mathcal{U}$. The variance of users' recommendation quality $\sigma^2(\mathrm{Q})$ measures the consistency of recommendation quality across different customers:*

$$\sigma^2(\mathrm{Q}) = \frac{1}{|\mathcal{U}|} \sum (Q_u - \overline{\mathrm{Q}}), \tag{2.15}$$

where $\overline{\mathrm{Q}}$ is the mean of the set Q*. A low variance indicates that recommendation quality is consistent across users, suggesting individual customer-level fairness. Conversely, a high variance reveals disparities in the quality of recommendations, potentially indicating unfair treatment of certain users.*

Then, we move to measure group-based consumer fairness. Group-based customer fairness requires dividing users into two groups or more. The group definition often involves comparing outcomes for a protected group, typically representing individuals who are vulnerable or disadvantaged, with an unprotected group (i.e., the dominant group). The objective is to ensure that protected group members receive treatment comparable to their unprotected counterparts, thereby mitigating structural biases and promoting equity. In this regard, we define the Mean Absolute Deviation (MAD) [55].

Definition 2.11 (Mean Absolute Deviation). *Let Q_u be the quality of the recommendation provided to the user u measured using a suitable accuracy metric (e.g., nDCG). Let $\mathcal{U}_x$ and $\mathcal{U}_y$ be two distinct groups of users. The mean absolute deviation (MAD) of the recommendation qualities $\mathrm{Q}^{(\mathcal{U}_x)}$ and $\mathrm{Q}^{(\mathcal{U}_y)}$ between the groups $\mathcal{U}_x$ and $\mathcal{U}_y$ measures the disparity in recommendation quality between different user groups:*

$$MAD\left(\mathrm{Q}^{(\mathcal{U}_x)}, \mathrm{Q}^{(\mathcal{U}_y)}\right) = \left| \frac{\sum_{x\in\mathcal{U}_x} Q_x}{|\mathcal{U}_x|} - \frac{\sum_{y\in\mathcal{U}_y} Q_y}{|\mathcal{U}_y|} \right|. \tag{2.16}$$

A smaller MAD value indicates that the average recommendation quality across user groups is closer to the overall mean, suggesting equitable treatment of groups. A higher MAD value highlights disparities in recommendation quality across groups, pointing to potential group-level unfairness.

When more than two groups are defined, the MAD values of all possible group pair combinations are computed. Then, the final metric value is the average of all the MAD values.

Provider Fairness. Here, we are interested in measuring the disparity of item exposure. The position in the list significantly influences the exposure of items in a recommendation list. Indeed, the position determines the item probability of being noticed and interacted with by the users [26, 164]. Consequently, the position of items in the ranked list impacts the visibility and consumption of individual items or groups of them. This dynamic has profound implications for equitable representation, as disparities in exposure can lead to unequal opportunities for certain items or groups to be consumed or appreciated by users. We provide two metrics to assess the exposure disparity of item groups: Ranking-based Statistical Parity (RSP) [228] and Ranking-based Equal Opportunity (REO) [228].

Definition 2.12 (Ranking-based Statistical Parity). *This metric is based on statistical parity, which forces the ranking probability distributions of different item categories c_i, with $i \in \{1, \ldots, n\}$, to be the same in a ranking task. Therefore, RSP formally encourages $P(R@k \mid c_1) = P(R@k \mid c_2) = \ldots = P(R@k \mid c_n)$, where R@k represents "the item being ranked in top-k", and $P(R@k \mid c_i)$ is the probability of items belonging to the category c_i being ranked in top-k. In the end, $RSP@k$ is computed as:*

$$RSP@k = \frac{std\left(P\left(R@k \mid c_1\right), \ldots, P\left(R@k \mid c_n\right)\right)}{mean\left(P\left(R@k \mid c_1\right), \ldots, P\left(R@k \mid c_n\right)\right)}, \tag{2.17}$$

where std(·) and mean(·) are the standard deviation and mean operator, respectively. Lower values of RSP indicate a fairer exposure of items in terms of statistical parity.

Given a generic category C_A of items, the probability $P(R@k|C_A)$ is computed as follows:

$$P\left(R@k \mid C = C_A\right) = \frac{\sum_{u=1}^{m} \sum_{i=1}^{k} \eta_{C_A}\left(R_{u,i}\right)}{\sum_{u=1}^{m} \sum_{i\in\mathcal{I}\setminus\mathcal{I}_u^+} \eta_{C_A}(i)},$$

where $\sum_{i=1}^{k} \eta_{C_A}\left(R_{u,i}\right)$ calculates how many un-interacted items from group C_A are ranked in top-k for user u, while $\sum_{i \in \mathcal{I} \setminus \mathcal{I}_u^+} \eta_{C_A}(i)$ calculates how many un-interacted items belong to group C_A for user u.

Definition 2.13 (Ranking-based Equal Opportunity)**.** *This metric is based on equal opportunity. In a ranking task, this concept is defined as the need for the ranking-based true positive rate (TPR) to be the same for different item categories c_i, with $i \in \{1, \ldots, n\}$. In this case, TPR is defined as the probability of an item belonging to a category to be ranked in the top-k given the ground truth that the user likes, i.e., $P\left(R@K | c_i, y = 1\right)$, where $y = 1$ represents users like items. In the end, REO@k is computed as:*

$$REO@k = \frac{std\left(P\left(R@k \mid c_1, y = 1\right), \ldots, P\left(R@k \mid c_n, y = 1\right)\right)}{mean\left(P\left(R@k \mid c_1, y = 1\right), \ldots, P\left(R@k \mid c_n, y = 1\right)\right)}, \tag{2.18}$$

where $std(\cdot)$ and $mean(\cdot)$ are the standard deviation and mean operator, respectively. Lower values of REO indicate a fairer exposure of items in terms of equal opportunity.

Given a generic category C_A of items, the probability $P(R@k|C = C_A, y = 1)$ is computed as follows:

$$P\left(R@k \mid C = C_a, y = 1\right) = \frac{\sum_{u=1}^{m} \sum_{i=1}^{k} \eta_{C_A}\left(R_{u,i}\right) Y\left(u, R_{u,i}\right)}{\sum_{u=1}^{m} \sum_{i \in \mathcal{I} \setminus \mathcal{I}_u^+} \eta_{C_A}(i) Y(u, i)}, \tag{2.19}$$

where $Y(u, R_{u,i})$ returns 1 if the item $R_{u,i}$ ranked in position i in the recommendation list of user u is liked by the user, 0 otherwise. The quantity $\sum_{i=1}^{k} \eta_{C_A}\left(R_{u,i}\right) Y\left(u, R_{u,i}\right)$ counts how many items in test set from category C_A are ranked in top-k for user u, while $\sum_{i \in \mathcal{I} \setminus \mathcal{I}_u^+} \eta_{C_A}(i) Y(u, i)$ counts the total number of items from category C_A in test set for user u.

Algorithmic Bias

Algorithmic Bias in RSs refers to systematic and unfair patterns in the behavior of recommendation algorithms that disadvantage specific individuals, groups, or items [121]. This bias can arise at various stages of the recommendation process, from data collection and model training to the system's deployment. Addressing algorithmic bias is crucial to ensuring fairness and trustworthiness in RSs.

A well-known form of bias in the recommendation domain is the **popularity bias**. This phenomenon generates a situation in which RSs over-suggest popular items, leading to the under-representation and reduced visibility of long-tail items [24]. Given the set of users $\mathcal{U}$, the recommendation list $\mathcal{L}_u$ for the user u, and its top-k items $\mathcal{L}_u^{(1,\ldots,k)}$, we define two metrics to evaluate to what extent a recommendation algorithm suffers from popularity bias: Average Percentage of items in the Long-Tail (APLT) [3] and Average Recommendation Popularity (ARP) [214].

Definition 2.14 (Average Percentage of items in the Long-Tail)**.** *The Average Percentage of items in the Long-Tail (APLT@k) measures precisely in what proportion unpopular* *items are recommended in users' recommendation lists:*

$$APLT@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|\{i, i \in (\mathcal{L}_u^{(1,\dots,k)} \cap \Gamma)\}|}{|\mathcal{L}_u^{(1,\dots,k)}|} \tag{2.20}$$

where Γ is the set of long-tail items. Higher values indicate a higher presence of long-tail items in the recommendation lists.

Definition 2.15 (Average Recommendation Popularity)**.** *The Average Recommendation Popularity (ARP@k) measures the average popularity of the recommended items:*

$$ARP@k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\sum_{i \in \mathcal{L}_u^{(1,\dots,k)}} \varphi(i)}{|\mathcal{L}_u^{(1,\dots,k)}|} \tag{2.21}$$

where $\varphi(i)$ is the number of interactions recorded for item i. Lower values indicate a higher presence of long-tail items in the recommendation lists.

Although popularity bias is a central issue faced in the literature of RSs, we may be interested in evaluating more generic phenomena concerning bias. For instance, sometimes, a user group's preferences on various item categories are not fairly reflected in the recommendations the group receives. Consequently, the recommendation algorithms are leading to a situation in which existing biases in the data source are amplified in the output of recommendations. **Bias Disparity** [182] helps to measuring this phenomenon. To formally define the bias disparity, we consider a set of users $\mathcal{U}$, a set of items $\mathcal{I}$, and the implicit feedback user-item matrix S, where $S(u, i) = 1$ if user u has selected item i, and zero otherwise. We assume that the users are partitioned into a set G of *groups*, while the items are divided into a set C of *categories*. We define the *input preference ratio* $PR_S(g, c)$ of a group $g \in G$ for category $c \in C$ as the fraction of interactions between users from group g with items belonging to category c:

$$PR_S(g, c) = \frac{\sum_{u \in g} \sum_{i \in c} S(u, i)}{\sum_{u \in g} \sum_{i \in I} S(u, i)}. \tag{2.22}$$

Then, we define the *source bias* $B_S(g, c)$ of group g for category c as:

$$B_S(g, c) = \frac{PR_S(g, c)}{P(c)}, \tag{2.23}$$

where $P(c) = |c|/m$ is the probability of selecting an item from category c uniformly at random. Values of $B_S(g, c)$ less than 1 denote *negative bias*, i.e., the group g on average tends to select less often from category c. Values of $B_S(g, c)$ greater than 1 denote *positive bias*, i.e., the group g prefers category c disproportionately to its size. By assuming the recommendation algorithm's outputs for each user u as a ranked list of r items, we define the binary matrix R, where $R(u, i) = 1$ if item i is recommended for user u, and zero otherwise. Given the matrix R, we compute the *output preference ratio* of the recommendations, $PR_R(g, c)$, of group g for category c with Equation 2.22, and the *recommendation bias* $B_R(g, c)$ of group g for category c with Equation 2.23. Then, we can define the bias disparity metric.

Definition 2.16 (Bias Disparity)**.** *The Bias Disparity (BD) of a group g for a category c between the source data S and the output recommendations R, that is, the relative change of the bias value, is:*

$$BD(g, c) = \frac{B_R(g, c) - B_S(g, c)}{B_S(g, c)}.$$

The closer the BD values are to 0, the more the output recommendation bias does not deviate from the source bias.

Chapter 3

Background of Multi-Objective Evaluation and Optimization of Recommender Systems

Accuracy has traditionally been the primary objective in the development of Recommender Systems (RSs). This focus is particularly evident during both the optimization and evaluation phases of the recommendation pipeline (see Section 2.3). However, in real-world applications, recommendations are influenced by factors beyond accuracy alone, which may include user-centric considerations, multi-stakeholder dynamics, or the need to address multiple tasks (see Section 2.4). These considerations highlight the necessity for a more comprehensive perspective on RS operations.

To foster a holistic understanding of a RS, it is essential to account for the diverse perspectives that influence recommendations during both model optimization and performance evaluation. This approach underpins two key concepts explored in this dissertation: (i) the *multi-objective evaluation* of RSs and (ii) the design and implementation of *multi-objective RSs*.

3.1 Multi-Objective Evaluation of RSs

In the evaluation of Recommender Systems (RSs), a wide array of metrics has been proposed to assess the performance of recommendation algorithms across multiple perspectives. Section 2.4 defines several widely used beyond-accuracy metrics, reflecting a growing recognition of the need for broader evaluation criteria in the literature. While recent studies often complement accuracy evaluations with metrics that capture additional dimensions, these are typically presented in tables or graphs where accuracy remains disproportionately emphasized. A significant limitation in current evaluation practices is the tendency to assess each metric independently, rather than viewing them simultaneously. This fragmented approach undermines the development of a holistic understanding of RS performance and perpetuates the overemphasis on accuracy at the expense of other critical aspects.

Multi-objective evaluation of RSs can be a pivotal approach to addressing the limitations of traditional, single-metric evaluations. *A multi-objective evaluation simultaneously assesses the quality of recommendations across multiple perspectives*. These perspectives are often context-dependent, as different domains, recommendation tasks, and scenarios necessitate specific metrics and tailored evaluation setups. However, the complexity of multi-objective evaluation increases with the number of perspectives considered.

The first step in performing a multi-objective evaluation is defining a set of appropriate metrics, denoted as $(m_1, \ldots, m_n)$, which collectively capture the key considerations across the relevant perspectives. Then, how determine the overall (i.e., multi-objective) performance of the algorithm poses an important challenge. In addition, how to rank several algorithms performance is not straightforward. For instance, consider two metrics, m_1 and m_2 and two RSs R_A and R_B. If R_A outperforms R_B on m_1, but R_B excels on m_2, it becomes non-trivial to determine which system performs better overall. This scenario necessitates strategies for aggregating performance across metrics and redefining the notion of optimality.

According to Bauer et al. [22], several strategies can be employed to develop multi-objective evaluation mechanisms, including:

- Weighted (typically linear) aggregation of individual metrics [30] into a single numeric score, facilitating a more straightforward comparison of candidate systems by consolidating multiple performance dimensions into one measure of overall performance.
- Dimensionality reduction of metrics by transforming certain individual metrics into constraints [217], thereby simplifying the evaluation while still capturing essential performance aspects.
- Identification of the Pareto frontier of multidimensional performance vectors across different candidate systems, providing insights into the trade-offs between competing objectives and helping to identify optimal solutions in a multi-objective context.

Among them, the visualization of Pareto frontiers is the most widely used method [15, 19, 74]. However, these visualizations are typically assessed from a qualitative perspective, lacking a rigorous and quantitative evaluation framework.

Overall, we believe that the multi-objective evaluation paradigm is often overlooked in the recommendation domain, where the literature predominantly emphasizes the relevance of recommendations as the gold standard. This narrow focus on accuracy not only limits the understanding of RSs' full potential but also influences other aspects of evaluation in this field (see Section 2.3.3). For instance, the selection of the best model is typically based on accuracy performance within a validation set, which may hinder a comprehensive understanding of RS capabilities beyond accuracy. Furthermore, while numerous studies have explored the sensitivity of algorithms to hyper-parameter tuning [20, 162], these investigations primarily center on accuracy performance, leaving the models' behavior under a multi-objective evaluation largely unexplored.

In the chapters 4, 5, and 6 of this dissertation, we delve into the multi-objective evaluation of RSs, aiming to uncover its unexamined potential to provide a holistic understanding of RS performance and to disentangle the complex interplay of multiple facets within these systems.

3.2 Multi-Objective Recommender Systems

While predicting the relevance of individual items for users remains a central problem in RSs, focusing solely on a single objective, i.e., prediction accuracy, and its corresponding metrics may oversimplify the complexity of real-world scenarios.

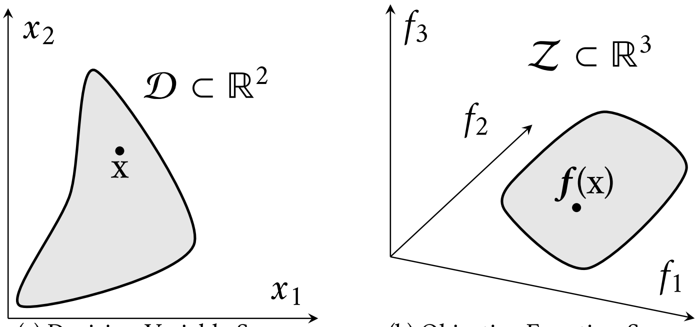
Hence, it is crucial to embrace a more comprehensive perspective that incorporates multiple optimization goals, diverse stakeholder objectives, and their inherent trade-offs [91]. This shift has catalyzed the rising interest in Multi-Objective Recommender Systems (MORSs) [5]. *MORSs are designed to balance multiple objectives by employing multi-objective optimization techniques* [224]. However, handling multiple objectives introduces significant challenges, as conflicts between objectives often arise. For example, improving one objective may adversely affect another due to the trade-offs intrinsic to real-world applications. Such trade-offs are particularly evident when incorporating beyond-accuracy aspects in the optimization process. For instance, the RSs that achieves the highest accuracy may perform poorly in terms of novelty and diversity, and vice versa [152]. The presence of trade-offs implies that a multi-objective optimization process may yield more than one optimal solution, i.e., a solution in which no objective can be further improved without hurting the other ones [114].

Zheng et al. [224] highlight several limitations in the current development and evaluation practices of MORSs. Although the next sections of this chapter provide the basis for understanding the technical background of MORSs, we deal with some of these challenges in Chapters 7, 8, and 9.

3.3 Multi-Objective Optimization Problem

Multi-Objective Recommender Systems (MORSs) are fundamentally grounded in the principles of Multi-Objective Optimization (MOO). Additionally, many core concepts from MOO can be effectively applied to the multi-objective evaluation of generic RSs. To lay the foundation for the discussions in this dissertation, we provide a comprehensive overview of MOO problems, introducing the essential notions that underpin our works. We begin with a formal definition of a MOO problem.

A MOO problem involves multiple objective functions, each of which needs to be minimized or maximized. The general form of a MOO problem can be expressed as follows:


x_2
$\mathcal{D} \subset \mathbb{R}^2$
x
x_1
f_3
$\mathcal{Z} \subset \mathbb{R}^3$
f_2
$\boldsymbol{f}$(x)
f_1

(a) Decision Variable Space. (b) Objective Function Space.

Figure 3.1. Search spaces in multi-objective optimization problems. For a solution x in the decision variable space $\mathcal{D}$, there exists a point $\boldsymbol{f}(\mathrm{x})$ in the objective function space $\mathcal{Z}$.

Definition 3.1 (Multi-Objective Optimization Problem). *A Multi-Objective Optimization (MOO) problem is defined as:*

$$\begin{array}{rll} \min\limits_{\mathrm{x}} & f_m(\mathrm{x}) & m = 1, 2, \ldots, M \\ \textit{subject to} & \mathrm{g}_j(\mathrm{x}) \leq 0 & j = 1, 2, \ldots, J \\ & \mathrm{h}_k(\mathrm{x}) = 0 & k = 1, 2, \ldots, K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, 2, \ldots, N \end{array} \tag{3.1}$$

The vector $\mathrm{x} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{N}\}^\mathrm{T}$ is formed by n independent variables called *decison variables*. The last constraint restricts these decision variables to take a value within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bound, that constitute the *decision variable space $\mathcal{D} \subseteq \mathbb{R}^N$*.

The terms $g_j(\mathrm{x})$ and $h_k(\mathrm{x})$ define the set of J equality and K inequality constraints, respectively. On the one hand, a solution x that does not satisfy all of the $J + K$ constraints and the $2N$ variable bounds stated above is called an *infeasible solution*. On the other hand, if any solution x satisfies all constraints and variable bounds, it is known as a *feasible solution*.

There are M objective functions $\boldsymbol{f}(\mathrm{x}) = \{f_1(\mathrm{x}), f_2(\mathrm{x}), \ldots, f_M(\mathrm{x})\}^T$ considered in the above definition. For each solution $\mathrm{x} \in \mathcal{D}$, there exist a point $\boldsymbol{f}(\mathrm{x}) = \mathrm{z} = \{z_1, z_2, \ldots, z_M\}^T$, composing the set $\mathcal{Z} \subseteq \mathbb{R}^M$ called *objective function space*. Figure 3.1 illustrates the decision variable and the objective function spaces.

It is worth mentioning that the definition of a MOO problem in Eq. (3.1) supposes that all the M objective functions $\boldsymbol{f}(\mathrm{x})$ are to be minimized. However, each objective function can be generally either minimized or maximized. In the optimization context, the duality principle suggests that we can convert a minimization problem into a maximization one by multiplying the objective function by -1. Without loss of generality, in the remainder of this chapter, we will refer to optimization problems in which all the objective functions are to be minimized. Indeed, when an objective is required to be maximized by using such an algorithm, the duality principle can be used to transform the original objective for maximization into an objective for minimization.

3.4 Notions of Optimality in MOO problems

To precisely define the Multi-Objective Optimization (MOO) problem, it is essential to establish the meaning of minimization in $\mathbb{R}^M$. This requires determining how the objective vectors $\boldsymbol{f}(\mathrm{x}) \in \mathbb{R}^M$ are compared for different solutions $\mathrm{x} \in \mathbb{R}^N$.

In single-objective optimization, the "less than or equal to" ($\leq$) relation is used to compare the scalar values of objective functions. Indeed, in these problems, the goal is to optimize a single scalar-valued objective function $f : \mathbb{R}^N \longrightarrow \mathbb{R}$. The relation $\leq$ leads to a unique optimal solution $\mathrm{x}^\star \in \mathcal{D}$, such that: $f(\mathrm{x}^\star) = \min_{\mathrm{x}\in\mathcal{D}} f(\mathrm{x})$. Therefore, this relation induces a total order in $\mathbb{R}$, guaranteeing that all solutions can be ranked according to their objective values.

In contrast, MOO problems involve the simultaneous optimization of multiple—often conflicting—objectives $\boldsymbol{f}(\mathrm{x})$, where $\boldsymbol{f} : \mathbb{R}^N \longrightarrow \mathbb{R}^M$. There is no natural way to rank all solutions using a single scalar criterion in $\mathbb{R}^M$, as the objective functions may often conflict. Hence, weaker order definitions are required to compare vectors in $\mathbb{R}^M$. Typically, one way to compare the various solutions in a MOO problem is to use the concept of Pareto dominance, generalized by the French-Italian economist Vilfredo Pareto in 1896.

Definition 3.2 (Pareto dominance relation). *A solution* x_1 *Pareto-dominates a solution* x_2*, denoted by* $\mathrm{x}_1 \prec \mathrm{x}_2$*, if and only if the following conditions are true:*

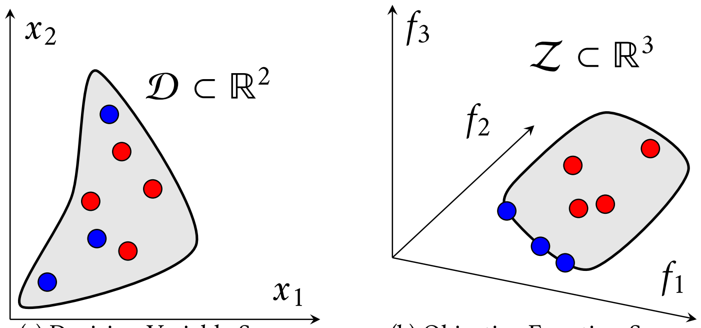
1. $f_i(\mathrm{x}_1) \leq f_i(\mathrm{x}_2)\, \forall i \in \{1, \dots, M\}$;
2. $\exists\, i \in \{1, \dots, M\} \mid f_i(\mathrm{x}_1) < f_i(\mathrm{x}_2)$.

The above definition practically says that a solution x_1 dominates a solution x_2 if x_1 is no worse than x_2 in all objectives and x_1 is strictly better than x_2 in at least one objective. The Pareto dominance relation is *transitive* because if $\mathrm{x}_1 \prec \mathrm{x}_2$ and $\mathrm{x}_2 \prec \mathrm{x}_3$, then $\mathrm{x}_1 \prec \mathrm{x}_3$. A binary relation must at least satisfy transitivity to qualify as an ordering relation. Thus, the dominance relation qualifies as an ordering relation. However, since the Pareto dominance relation is not reflexive and antisymmetric, it is only a *strict partial-order* relation.

Given a finite set of solutions $\{x_1, x_2, \dots, x_N\}$, we can perform all possible pairwise comparisons to determine which solution dominates which and identify the non-dominated solutions relative to each other. Ultimately, we aim to have a set of solutions, any two of which dominate each other. This set is called *Pareto optimal set*, which contains the *Pareto optimal* solutions, formally defined as follows.

Definition 3.3 (Pareto Optimality). *A solution* $\mathrm{x}^\star \in \mathcal{D}$ *is Pareto optimal if:*

$$\nexists\, \mathrm{x} \in \mathcal{D} \mid \mathrm{x} \prec \mathrm{x}^\star. \tag{3.2}$$


x_2
$\mathcal{D} \subset \mathbb{R}^2$
x_1
f_3
$\mathcal{Z} \subset \mathbb{R}^3$
f_2
f_1

(a) Decision Variable Space. (b) Objective Function Space.

Figure 3.2. Illustration of the Pareto optimal set (blue points on the left) and its image, i.e., the Pareto frontier (blue points on the left). The red points are the dominated solutions.

Definition 3.4 (Pareto Optimal Set)**.** *The Pareto optimal set $\mathcal{P}^\star$ is defined as:*

$$\mathcal{P}^\star = \{\mathrm{x}^\star \in \mathcal{D} \,|\, \nexists \mathrm{x} \in \mathcal{D} \,:\, \mathrm{x} \prec \mathrm{x}^\star\}. \tag{3.3}$$

Algorithms often produce solutions that, while not strictly Pareto optimal, fulfill other criteria, making them valuable for practical applications. For example, a weakly Pareto optimal solution is defined as follows.

Definition 3.5 (Weak Pareto Optimality)**.** *A solution $\mathrm{x}^\star \in \mathcal{D}$ is weakly Pareto optimal if:*

$$\nexists \mathrm{x} \in \mathcal{D} \,|\, \boldsymbol{f}(\mathrm{x}) < \boldsymbol{f}(\mathrm{x}^\star) \, \forall i \in \{1, 2, \ldots, M\}. \tag{3.4}$$

A point is considered weakly Pareto optimal if no other point exists that improves *all* objective functions simultaneously. In contrast, a point is Pareto optimal if no other point can improve *at least one* objective function without causing a deterioration in another. While all Pareto optimal points are also weakly Pareto optimal, the reverse is not necessarily true, i.e., weakly Pareto optimal points are not always Pareto optimal.

The image in the objective function space of the Pareto optimal set $\mathcal{P}^\star$ is called *Pareto frontier.*

Definition 3.6 (Pareto Frontier)**.** *For a Pareto optimal set $\mathcal{P}^\star$, the Pareto frontier $\mathcal{PF}^\star$ is defined as:*

$$\mathcal{PF}^\star = \{\boldsymbol{f}(\mathrm{x}^\star) \in \mathcal{Z} \,|\, \mathrm{x}^\star \in \mathcal{P}^\star\}. \tag{3.5}$$

Figure 3.2 illustrates the concept of Pareto optimal set and its corresponding Pareto frontier.

3.5 Special points

We now define some special points often exploited in multi-objective optimization algorithms, i.e., the *utopia point* and the *nadir point.*

Once a Pareto optimal set $\mathcal{P}^\star$ for the problem in Equation (3.1) is obtained, most real-world applications require selecting a single optimal solution. Generally, the *utopia point* helps to implement this process.

Definition 3.7 (Utopia Point)**.** *A point* $\boldsymbol{f}^\diamond \in \mathbb{R}^M$ *is a utopia point if and only if* $f_i^\diamond = \min_{\mathrm{x}} f_i(\mathrm{x}) \mid \mathrm{x} \in \mathcal{D}\ \forall i \in \{1, 2, \ldots, M\}$.

Generally, the utopia point is the *ideal* point in $\mathbb{R}^M$ that is unattainable. Hence, a common approach consists of reaching the *closest* solution to the utopia point as the best one, where, in most of the cases, the term *closest* refers to the solution which minimizes the Euclidean distance to the utopia point.

Along with the utopia point, the *nadir point* also helps select a solution from the Pareto frontier. Dually to the utopia point, the nadir point represents the point in the objective function space having the worst possible values for each objective.

Definition 3.8 (Nadir Point)**.** *A point* $\boldsymbol{f}^\triangle \in \mathbb{R}^M$ *is a nadir point if and only if* $f_i^\triangle = \max_{\mathrm{x}} f_i(\mathrm{x}) \mid \mathrm{x} \in \mathcal{D}\ \forall i \in \{1, 2, \ldots, M\}$.

Compared to the utopia point, determining the nadir point can be challenging, even for simple problems [109].

3.6 Two Approaches to MOO Problems

Single-Objective Optimization (SOO) focuses on optimizing a single criterion, where the goal is to identify the solution that minimizes the objective function. In contrast, Multi-Objective Optimization (MOO) involves simultaneously optimizing multiple objectives. From this distinction, it is evident that SOO and MOO differ in several key aspects:

- The cardinality of optimal solutions in MOO is typically greater than one, as multiple trade-offs often exist among conflicting objectives.
- MOO involves multiple, distinct goals that require simultaneous consideration.
- MOO operates across two search spaces: the decision space (possible solutions) and the objective space (resulting trade-offs).

Although the primary distinction between SOO and MOO lies in the cardinality of the optimal set, a system designer requires only one actionable solution in practice. However, in the context of MOO, this operational need presents a challenge: *which solution should a system designer choose among the Pareto optimal set?*

To illustrate, consider the problem of recommending movies on a streaming platform. The recommendation system must balance multiple objectives, such as (i) matching user preferences, (ii) promoting diverse genres of movies, (iii) evenly exposing the movies of different producers, and (iv) maximizing the revenue of the platform. These objectives often conflict, leading to different trade-offs. For instance, maximizing the platform's revenue could negatively affect the relevance of the recommendation. Conversely, optimizing for engagement could come at the expense of immediate revenue. Hence, some questions emerge: should the system favor short-term revenue at the risk of diminishing user retention? Or should it prioritize user engagement even if it reduces profitability?

Selecting a single solution in such scenarios becomes complex and depends on whether precise preferences among the objectives are known. If a preference factor among objectives is unavailable or unclear, the solution selection process may involve many other considerations, often non-technical, qualitative, or experience-driven.

In this case, the priority shifts to finding diverse trade-off solutions by treating all objectives equally important. Once this Pareto optimal set is obtained, system designers can apply higher-level qualitative assessments or mathematical methods to select a single solution. This procedure leads to a foundational principle for an ideal multi-objective optimization procedure:

1. Identify multiple trade-off solutions that capture a wide range of objective values.
2. Use higher-level information or mathematical techniques to select one solution from the trade-off set.

This approach ensures a methodical and practical framework, reducing subjectivity in decision-making. However, it can be computationally expensive, as it may require training multiple algorithms to generate the Pareto frontier or developing a single algorithm capable of efficiently discovering multiple solutions.

Then, when a reliable relative preference vector is available, there is no reason to find other trade-off solutions, and the selection process can be significantly streamlined. A *preference-based* approach becomes sufficient in such a case. Indeed, each trade-off solution corresponds to a specific order of importance of the objectives. A simple method would be to design a composite objective function as the weighted sum of the objectives, where a weight for an objective is proportional to the preference factor assigned to that objective. This scalarization converts the MOO problem into an SOO problem, enabling the system to directly optimize the composite objective and find a single trade-off solution aligned with the provided preferences. Furthermore, multiple trade-off solutions can still be obtained by varying the preference vector and repeating the optimization. It is intuitive to realize that determining a reliable preference vector is inherently subjective and challenging, especially when there is minimal or no prior knowledge about the trade-off space.

3.7 MOO Methods

The previous section outlined two widely adopted approaches to addressing Multi-Objective Optimization (MOO) problems. The choice between these approaches depends on the availability of explicit preferences among objectives. On the one hand, when no clear hierarchy or preference factor exists among the objectives, generating a Pareto optimal set is systematic. This set provides diverse trade-off solutions from which a single actionable solution can later be selected based on higher-level considerations. On the other hand, if a system designer knows an explicit preference factor for each objective, the MOO problem can be reformulated into a Single-Objective Optimization (SOO) problem. This procedure is typically achieved by creating a composite objective function through a weighted summation of the individual objectives, where the weights reflect their relative importance.

These two approaches directly lead to a classification of MOO methods based on their underlying optimization strategies:

1. **Scalarization methods**: these methods transform a MOO problem into an SOO problem by assigning preference weights to objectives and combining them into a single composite objective function, typically using a weighted sum.
2. **Population-Based Heuristic methods**: these methods employ heuristic search strategies to directly explore the Pareto frontier without requiring predefined preference weights. Among these, Multi-Objective Evolutionary Algorithms (MOEAs) are the most prominent. MOEAs leverage population-based search techniques to simultaneously optimize multiple objectives, offering a diverse set of Pareto-optimal solutions.

3.7.1 Scalarization methods

As the name implies, the scalarization method transforms a MOO problem into a SOO problem by scalarizing the objectives. The scalarization is achieved by multiplying each objective function with a predefined weight that reflects its relative importance and then summing them into a single composite objective. This specific method of combining each objective function is called **weighted sum method**. Formally, the MOO problem defined in Eq. (3.1) becomes:

$$\begin{array}{llr} \min\limits_{\mathrm{x}} & F(\mathrm{x}) = \sum\limits_{m} w_m f_m(\mathrm{x}) & m = 1, 2, \dots, M \\ \text{subject to} & \mathrm{g}_j(\mathrm{x}) \leq 0 & j = 1, 2, \dots, J \\ & \mathrm{h}_k(\mathrm{x}) = 0 & k = 1, 2, \dots, K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, 2, \dots, N \end{array} \tag{3.6}$$

Here, w_m, with $m \in \{1, 2, \dots, M\}$, is the weight of the m-th objective function. It is a usual practice to set the weights such that their sum is one, i.e., $\sum_{m=1}^{M} w_m = 1$. The following theorem indicates how to mathematically define these weights [127].

Theorem 3.1. *The solution to the problem represented by equation (3.6) is Pareto-optimal if the weight is positive for all objectives.*

This theorem holds for any MOO problem. When a positive weight vector is used, the optimal solution to the scalarized problem in Eq.(3.6) is guaranteed to be a Pareto-optimal solution. However, this does not imply that all Pareto-optimal solutions can be obtained using positive weight vectors. The following theorem confirms this result for convex problems [127].

Theorem 3.2. *Suppose* $\mathrm{x}^\star$ *is a Pareto-optimal solution of a convex multi-objective optimization problem. In that case, there exists a non-zero positive weight vector* w *such that* $\mathrm{x}^\star$ *is a solution to the problem given by equation (3.6).*

The theorem above states that multiple Pareto-optimal solutions can be obtained by solving the scalarized problem in Eq. (3.6) with various positive preference vectors, each used independently. Each solution corresponds to a distinct Pareto-optimal point. This result can be extended to non-convex MOO problems, especially in cases where the Pareto front exhibits convexity.

Moreover, determining an appropriate weight vector also depends on the scaling of each objective function. Objectives often differ in their orders of magnitude, which can disproportionately influence the optimization process. Therefore, it is advisable to normalize the objectives to ensure comparability and balanced contributions to the composite objective function.

Finally, in addition to the weighted sum method, there are many variations for the scalarization approach [224], such as the weighted exponential sum method, weighted product, weighted metric method, weighted Chebyshev method, and exponential weighted criterion. These variations retain the practice of multiplying the objective functions with their corresponding preference factors. However, they mainly differ in the aggregation function of the objectives.

3.7.2 Population-based heuristic methods

The previous section emphasized two key characteristics of scalarization methods. Firstly, these methods typically require multiple executions to identify various points on the Pareto optimal set. Secondly, many of them depend on preference information among the objectives.

In contrast, alternative approaches to directly generate Pareto frontiers and implement the ideal multi-objective optimization procedure rely on Multi-Objective Evolutionary Algorithms (MOEAs). MOEAs extend the framework of Evolutionary Algorithms (EAs), which are stochastic search and optimization techniques inspired by the natural process of evolution.

EAs begin with a population of candidate solutions, often generated randomly within predefined bounds for each variable. However, when prior knowledge about desirable solution characteristics is available, it can be advantageous to incorporate this information into the initial population generation. Following initialization, EAs proceed through an iterative process of population updating to refine the solutions over successive generations.

The population update in EAs involves three main phases:

- **Selection**: this phase identifies parent individuals from the current population based on their *fitness* value, determined using a *fitness function*. Selection typically involves a stochastic process where higher fitness values increase the likelihood of selection.

- **Variation**: it is achieved through two operators, i.e., crossover and mutation, which generate a modified population. The crossover operator combines information from selected parent solutions to create offspring, with a specified crossover probability dictating the proportion of individuals participating in this process. The remaining fraction of the population is carried over unmodified. Offspring produced by crossover are further perturbed using a mutation operator, where each variable is modified with a mutation probability.
- **Elite Preservation**: the elitism operator combines the current and newly generated populations, retaining only the most promising solutions. This operation ensures that the algorithm maintains a non-decreasing performance trend across generations.

Fitness value evaluation is pivotal in the evolutionary process, assigning a scalar fitness value to each individual. The fitness assignment typically involves ranking individuals according to a preference relation and assigning fitness values based on their rank, enabling straightforward sorting of solutions from best to worst.

MOEAs differ from traditional EAs in their fitness assignment and population ranking handling. While retaining the fundamental phases of EAs, MOEAs incorporate multiple fitness functions, each corresponding to a specific objective. Consequently, population ranking in MOEAs is based on a dominance principle, like the Pareto dominance relation, which accounts for the simultaneous optimization of multiple objectives. Most known evolutionary algorithms are MOGA [67], NSGA [167] and NGSA-II [52], SPEA [229] and SPEA2 [232], PAES [103], and PESA [47] and PESA-II [46]. Although these approaches can deal with convex and non-convex Pareto frontiers, they do not always guarantee a Pareto optimal set, but only a good approximation. Furthermore, the computation cost may increase depending on the data size and the number of parameters to be learned.

Part II
Methodologies for Multi-Objective Evaluation of Recommender Systems

Chapter 4

Auditing Consumer- and Producer-Fairness in Graph Collaborative Filtering

To date, *graph* collaborative filtering (CF) strategies have been shown to outperform pure CF models in generating accurate recommendations. Nevertheless, recent works have raised concerns about fairness and potential biases in the recommendation landscape since unfair recommendations may harm the interests of Consumers and Producers (CP). Acknowledging that the literature lacks a careful evaluation of graph CF on CP-aware fairness measures, we initially evaluated the effects on CP-aware fairness measures of eight state-of-the-art graph models with four pure CF recommenders. Unexpectedly, the observed trends show that graph CF solutions do not ensure a large item exposure and user fairness. To disentangle this performance puzzle, we formalize a taxonomy for graph CF based on the mathematical foundations of the different approaches. The proposed taxonomy shows differences in node representation and neighbourhood exploration as dimensions characterizing graph CF. Under this lens, the experimental outcomes become clear and open the doors to a multi-objective CP-fairness analysis. To perform this multi-objective analysis, we employ Pareto frontiers to qualitatively audit to what extent the graph CF can balance the trade-off among accuracy, item exposure, and user fairness. Hence, in this chapter we show how Pareto frontiers can be exploited to perform a multi-objective evaluation of RSs. Codes are available at: `https://github.com/sisinflab/ECIR2023-Graph-CF`.[1]

[1] This chapter is based on the work published in the *Proceedings of the 45th European Conference on Information Retrieval* (ECIR 2023) "Auditing Consumer- and Producer-Fairness in Graph Collaborative Filtering".

4.1 Introduction and Motivations

Recommender systems (RSs) are ubiquitous and utilized in a wide range of domains from e-commerce and retail to media streaming and online advertising. Personalization, or the system's ability to suggest relevant and engaging products to users, has long served as a key indicator for gauging the success of RSs. In recent decades, collaborative filtering (CF) [62], the predominant modeling paradigm in RSs, has shifted from neighborhood techniques [62, 151, 156] to frameworks based on the learning of users' and items' latent factors [105, 150, 211]. More recently, deep learning (DL) models have been proposed to overcome the linearity of traditional latent factors approaches.

Among these DL algorithms, graph-based methods view the data in RSs from the perspective of graphs. By modeling users and items as nodes with latent representations and their interactions as edges, the data can be naturally represented as a user-item bipartite graph. By iteratively aggregating contributions from near- and long-distance neighborhoods, the so-called message-passing schema updates nodes' initial representations and effectively distills the collaborative signal [200]. Early works [25, 215] adopted the vanilla graph convolutional network (GCN) [101] architecture and paved the way to advanced algorithms lightening the message-passing schema [45, 84] and exploring different graph sampling strategies [206]. Recent approaches propose simplified formulations [123, 141] that optionally transfer the graph CF paradigm to different spaces [163, 173]. As some graph edges may provide noisy contributions to the message-passing schema [186], a research line focuses on meaningful user-item interactions [180, 199, 203]. In this context, explainability is the natural next step [117] towards the disentanglement of user-item connections into a set of user intents [201, 207].

On the other side, the adoption of DL (and, often, black-box) approaches to the recommendation task has raised issues regarding the fairness of RSs. The concept of fairness in recommendation is multifaceted. Specifically, the two core aspects to categorize recommendation fairness may be summarized as (1) the primary parties engaged (consumers vs. producers) and (2) the type of benefit provided (exposure vs. relevance). Item suppliers are more concerned about exposure fairness than customers because they want to make their products better known and visible (**P**roducer fairness). However, from the customer's perspective, relevance fairness is of utmost importance, and hence system designers must ensure that exposure of items is equally effective across user groups (**C**onsumer fairness). A recent study highlights that nine out of ten publications on recommendation fairness concentrated on either C-fairness or P-fairness [131], disregarding the joint evaluation between C-fairness, P-fairness, and the accuracy.

The various graph CF *strategies* described above have historically centered on the enhancement of system accuracy, but, actually, never focused on the recommendation fairness dimensions. Despite some recent graph-based approaches have specifically been designed to address C-fairness [69, 108, 146, 193, 197, 208] and P-fairness [28, 119, 120, 174, 222, 225], there is a notable *knowledge gap* in the literature about the effects of the state-of-the-art graph *strategies* on the three objectives of C-fairness, P-fairness, and system accuracy. This work intends to complement the previous research and provide answers to pending research problems such as how different graph models perform for the three evaluation objectives. By measuring these dimensions in terms of **overall accuracy**, **user fairness**, and **item exposure**, we observe these aspects in detail[2].

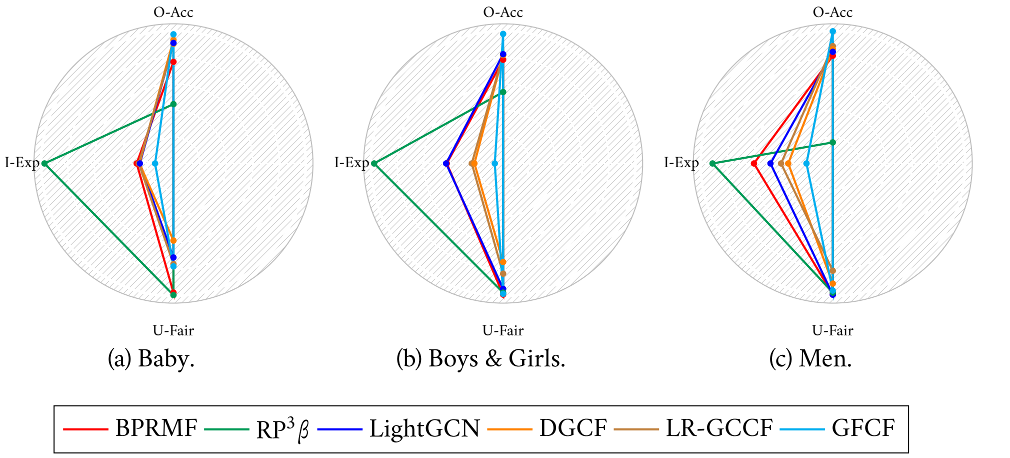
Motivating example. A preliminary comparison of the leading graph and classical CF models is carried out to provide context for our study. The graph-based models include LightGCN [84], DGCF [201], LR-GCCF [45], and GFCF [163], which are tested against two classical CF baselines, namely BPRMF [149] and $\text{RP}^3\beta$ [139], on the Baby, Boys & Girls, and Men datasets from the Amazon catalog [134]. We train each baseline using a total of 48 unique hyper-parameter settings and select the optimal configuration for each baseline as the one achieving the highest accuracy on the validation set (as in the original papers). Overall accuracy, user fairness, and item exposure (as introduced above) are evaluated. Figure 4.1 displays the performance of the selected baselines on the three considered recommendation objectives. For better visualization, all values are scaled between 0 and 1 using min-max normalization, and, when needed, they are replaced by their 1's complement to adhere to the "higher numbers are better" semantics. As a result, in each of the three dimensions, the values lay in [0, 1] with higher values indicating the better. Please, note that such an experimental evaluation is not the main focus of this work but it is the motivating example for the more extensive analysis we present later. The interested reader may refer to Section 4.3 for a presentation of the full experimental settings to reproduce these results and the ones reported in the following sections of the chapter.

First, according to Figure 4.1, graph CF models are significantly more accurate than the classical CF ones, even if the latter perform far better in terms of item exposure. Moreover, the displayed trends suggest there is no clear winner on the user fairness dimension: classical CF models show promising performance, while some graph CF models do not achieve remarkable results. As a final observation, an underlying trade-off between the three evaluation goals seems to exist, and it might be worth investigating it in-depth. Such outcomes open to a more complete study on how **different strategy patterns** recognized in graph CF may affect the three recommendation objectives, which is the scope of this work.

Research questions and contributions. In the remainder of this chapter, we therefore attempt to answer the following two research questions (RQs):

RQ1. Given the different graph CF strategies, the raising question is: *"Can we explain the variations observed when testing several graph models on overall accuracy, item exposure, and user fairness separately?"* According to a recent benchmark that identifies some state-of-the-art graph techniques [227], the suggested graph CF taxonomy (Ta-

[2] In the rest of the chapter, when no confusion arises, we will refer to C-fairness with user fairness, to P-fairness with item exposure, and to their combination as CP-fairness.



(a) Baby.

(b) Boys & Girls.

(c) Men.

Figure 4.1. Kiviat diagrams indicating the performance of selected pure and graph CF recommenders on overall accuracy (i.e., O-Acc, calculated with the *nDCG@20*), item exposure (i.e., I-Exp, calculated with the *APLT@20* [3]), and user fairness (U-Fair, calculated with the *UMADrat@20* [55]). Higher means better.

ble 4.1) extends the set of graph-based models introduced in the motivating example by examining eight state-of-the-art graph CF baselines through their strategies for *nodes representation* and *neighborhood exploration*. We present a more nuanced view of prior findings by analyzing the impact of each taxonomy dimension on overall accuracy and CP-fairness.

RQ2. The demonstrated performance prompts the questions: *"How and why nodes representation and neighborhood exploration algorithms can strike a trade-off between overall accuracy, item exposure, and user fairness?"* We employ the Pareto optimality to determine the influence of such dimensions in two-objective scenarios, where the objectives include overall accuracy, item exposure, and user fairness. The Pareto frontier is computed for three 2-dimensional spaces: accuracy/item exposure, accuracy/user fairness, and item exposure/user fairness.

4.2 Nodes Representation and Neighborhood Exploration in Graph Collaborative Filtering: A Formal Taxonomy

4.2.1 *Preliminaries*

Let $\mathcal{U}$ be the set of $|\mathcal{U}|$ users, and $\mathcal{I}$ the set of $|\mathcal{I}|$ items in the system, respectively. We represent the observed interactions between users and items in a binary format (i.e., implicit feedback). Specifically, let $\mathrm{R} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ be the user-item feedback matrix, where $r_{u,i} = 1$ if user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$ have a recorded interaction,

Table 4.1. Categorization of the chosen graph baselines according to the proposed taxonomy. For each model, we refer to the technical description reported in the original paper and try to match it with our taxonomy.

Models	**Nodes Representation**				**Neighborhood Exploration**			
	Latent representation		**Weighting**		**Explored nodes**		**Message passing**	
	low	*high*	*weighted*	*unweighted*	*same*	*different*	*implicit*	*explicit*
GCN-CF* [101]		✓		✓	✓			✓
GAT-CF* [186]		✓	✓		✓			✓
NGCF [200]	✓			✓		✓		✓
LightGCN [84]	✓			✓		✓		✓
DGCF [201]	✓		✓			✓		✓
LR-GCCF [45]	✓			✓	✓	✓		✓
UltraGCN [123]	✓				✓	✓	✓	
GFCF [163]						✓	✓	

*The postfix -CF indicates that we re-adapted the original implementations (tailored for the task of node classification) to the task of personalized recommendation.

$r_{u,i} = 0$ otherwise. Following the above preliminaries, we introduce $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathrm{R})$ as the bipartite and undirected graph connecting users and items (the graph nodes) when there exists a recorded bi-directional interaction among them (the graph edges). Nodes features for user $u \in \mathcal{U}$ and $i \in \mathcal{I}$ are suitably encoded as the embeddings $\mathrm{e}_u \in \mathbb{R}^d$ and $\mathrm{e}_i \in \mathbb{R}^d$, with $d << N, M$. Given the dual nature of user and item derivations, we only report user-side formulas.

4.2.2 Updating node representation through message-passing

The representation of users' and items' nodes are updated by leveraging the graph topology from $\mathcal{G}$. In this respect, the message-passing schema has recently gained attention in the literature. The algorithm works by aggregating the information (i.e., the *messages*) from the *neighbor* nodes into the *ego* node, and the process is recursively performed for multiple hops thus exploring wider neighborhood portions. In general, the message-passing for l hops is:

$$\mathrm{e}_u^{(l)} = \omega\left(\left\{\mathrm{e}_{i'}^{(l-1)}, \forall i' \in \mathcal{N}(u)\right\}\right), \tag{4.1}$$

where $\omega(\cdot)$ and $\mathcal{N}(\cdot)$ are the aggregation function and neighborhood node set, respectively, while l is in $1 \leq l \leq L$, where L is a hyper-parameter. Note that the following statements hold: $\mathrm{e}_u^{(0)} = \mathrm{e}_u$ and $\mathrm{e}_i^{(0)} = \mathrm{e}_i$. A reworking of Equation (4.1) for $l \in \{2, 3\}$

allows *same*- and *different*-type node representation emerge [16]:

$$
\begin{array}{ll}
\begin{array}{l}\textbf{\textit{Same-type}}\\ \textbf{\textit{node}}\\ \textbf{\textit{representation}}\end{array} & \begin{cases} \underbrace{\mathrm{e}_u^{(2)}}_{\text{(user)}} = \omega\Big(\Big\{\omega\Big(\Big\{\underbrace{\mathrm{e}_{u''}^{(0)}}_{\text{(user)}}, \forall u'' \in \mathcal{N}(i') \setminus \{u\}\Big\}\Big), \forall i' \in \mathcal{N}(u)\Big\}\Big) \end{cases} \\[2ex]
\begin{array}{l}\textbf{\textit{Different-type}}\\ \textbf{\textit{node}}\\ \textbf{\textit{representation}}\end{array} & \begin{cases} \underbrace{\mathrm{e}_u^{(3)}}_{\text{(user)}} = \omega\Big(\Big\{\omega\Big(\Big\{\omega\Big(\Big\{\underbrace{\mathrm{e}_{i'''}^{(0)}}_{\text{(item)}}, \forall i''' \in \mathcal{N}(u'') \setminus \{i''\}\Big\}\Big), \\ \qquad \forall u'' \in \mathcal{N}(i') \setminus \{u''\}\Big\}\Big), \forall i' \in \mathcal{N}(u)\Big\}\Big). \end{cases}
\end{array}
\tag{4.2}
$$

To better clarify the extent of Equation (4.2), after an **even** and an **odd** number of explored hops, *ego* node updates leverage by design *same*- and *different*-type node connections, i.e., user-user/item-item and user-item/item-user as evident from Equation (4.2). While the existing literature does not always consider the two scenarios as distinct, we underline the importance of investigating the influence of different node-node connections explored during the message-passing. In light of the above, we will count the number of explored hops as follows: $\mathrm{e}_*^{(2l)}, \forall l \in \{1, 2, \ldots, \frac{L}{2}\}$ as obtained through l **same**-type node connections (denoted as *same-l*), and $\mathrm{e}_*^{(2l-1)}, \forall l \in \{1, 2, \ldots, \frac{L}{2}\}$ as obtained through l **different**-type node connections (denoted as *different-l*). In the following, we introduce the graph convolutional network (GCN) and its recent CF applications.

The baseline: graph convolutional network (GCN). The standard graph convolutional network from Kipf et al. [101] performs feature transformation, message aggregation, application of a one-layer neural network, element-wise addition, and ReLU activation, respectively. Let us consider $\mathrm{W}^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}$ and $\mathrm{b}^{(l)} \in \mathbb{R}^{d_l}$ as the weight matrix and the bias for the l-th explored hop. The message-passing for user u is:

$$
\mathrm{e}_u^{(l)} = \mathrm{ReLU}\left(\sum_{i' \in \mathcal{N}(u)} \left(\mathrm{W}^{(l)} \mathrm{e}_{i'}^{(l-1)} + \mathrm{b}^{(l)}\right)\right). \tag{4.3}
$$

GCN for collaborative filtering. Inspired by the GCN message-passing approach, the authors from Wang et al. [200] propose neural graph collaborative filtering (NGCF). At each hop exploration, the model aggregates the neighborhood information and the inter-dependencies among the *ego* and the neighborhood nodes. Formally, the aggregation could be formulated as follows:

$$
\mathrm{e}_u^{(l)} = \mathrm{LeakyReLU}\left(\sum_{i' \in \mathcal{N}(u)} \left(\mathrm{W}_{\mathrm{neigh}}^{(l)} \mathrm{e}_{i'}^{(l-1)} + \mathrm{W}_{\mathrm{inter}}^{(l)} \left(\mathrm{e}_{i'}^{(l-1)} \odot \mathrm{e}_u^{(l-1)}\right) + \mathrm{b}^{(l)}\right)\right), \tag{4.4}
$$

where LeakyReLU is the activation function, $\mathrm{W}_{\mathrm{neigh}}^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}$ and $\mathrm{W}_{\mathrm{inter}}^{(l)} \in \mathbb{R}^{d_{l-1} \times d_l}$ are the neighborhood and inter-dependencies weight matrices, respectively, while $\odot$ is the Hadamard product.

He et al. [84] propose a light convolutional network, namely LightGCN, with the rationale to simplify the message-passing schema from GCN and NGCF by dropping feature transformations (i.e., the weight matrices and biases) and the non-linearity applied after the message aggregation. Specifically, they implement:

$$\mathrm{e}_u^{(l)} = \sum_{i' \in \mathcal{N}(u)} \mathrm{e}_{i'}^{(l-1)}. \tag{4.5}$$

The variation shows superior accuracy to the state-of-the-art. A slightly different solution [45] can outperform LightGCN regarding the accuracy level.

4.2.3 Weighting the importance of graph edges

The message-passing schema is inherently designed to aggregate into the *ego* node all messages coming from its neighborhood. Nevertheless, the *binary* nature of the user-item feedback (i.e., 0/1) would suggest that not all recorded user-item interactions necessarily hide the same importance to the nodes they involve.

In general, let $a_{y \longrightarrow x}^{(l)}$ be the importance of the neighbor node y on its ego node x after l explored hops. We re-write the formulation of the message-passing after l explored hops (presented in Equation (4.1)) as:

$$\mathrm{e}_u^{(l)} = \omega\left(\left\{a_{i' \longrightarrow u}^{(l)} \mathrm{e}_{i'}^{(l-1)}, \forall i' \in \mathcal{N}(u)\right\}\right). \tag{4.6}$$

The baseline: graph attention network (GAT). Attention mechanisms have reached considerable success in the GCN-related literature to weight the contribution of neighbor messages before aggregation. The original study [186] proposes the following message-passing formulation:

$$\begin{aligned}\mathrm{e}_u^{(l)} &= \sum_{i' \in \mathcal{N}(u)} \left(a_{i' \longrightarrow u}^{(l)} \mathrm{W}_{\mathrm{neigh}}^{(l)} \mathrm{e}_{i'}^{(l-1)} + \mathrm{b}^{(l)}\right) \\ &= \sum_{i' \in \mathcal{N}(u)} \left(\alpha\left(\mathrm{e}_{i'}^{(l-1)}, \mathrm{e}_u^{(l-1)}\right) \mathrm{W}_{\mathrm{neigh}}^{(l)} \mathrm{e}_{i'}^{(l-1)} + \mathrm{b}^{(l)}\right),\end{aligned} \tag{4.7}$$

where $\alpha(\cdot)$ is the importance function depending on the lastly-calculated embeddings of the neighbor and the ego nodes, e.g., $a_{i' \longrightarrow u}^{(l)} = \alpha\left(\mathrm{e}_{i'}^{(l-1)}, \mathrm{e}_u^{(l-1)}\right)$.

GAT for collaborative filtering. The authors from Wang et al. [201] design a message-passing schema that calculates the importance of neighborhood nodes for *ego* nodes by disentangling the intents underlying each user-item interaction. Similarly to He et al. [84] and Chen et al. [45], they therefore propose the following embedding update formulation:

$$\begin{aligned}\mathrm{e}_u^{(l)} &= \sum_{i' \in \mathcal{N}(u)} a_{i' \longrightarrow u}^{(l)} \mathrm{e}_{i'}^{(l-1)} \\ &= \sum_{i' \in \mathcal{N}(u)} \alpha\left(\mathrm{e}_{i'}^{(l-1)}, \mathrm{e}_u^{(l-1)}, K, T\right) \mathrm{e}_{i'}^{(l-1)},\end{aligned} \tag{4.8}$$

where $\alpha\left(\cdot, K, T\right)$ is the importance function of the lastly-calculated embeddings from the neighbor and the *ego* nodes, e.g., $a_{i' \longrightarrow u}^{(l)} = \alpha\left(\mathrm{e}_{i'}^{(l-1)}, \mathrm{e}_u^{(l-1)}, K, T\right)$, K is the total number of intents, and T is the total number of routing iterations to repeat the disentangling procedure.

4.2.4 *Going beyond message-passing*

The recent graph learning literature [44, 226] has outlined the phenomenon of *over-smoothing*, that leads node representations to become more similar as more hops are explored. The issue is generally tackled by limiting the neighborhood exploration to (maximum) three hops, and to two hops when attention mechanisms are introduced. However, the idea of improving accuracy by restricting the number of explored neighborhoods is counter-intuitive and "conflicts" with the rationale behind collaborative filtering [17]. This awareness led works such as Mao et al. [123] and Shen et al. [163] to surpass and simplify the traditional concept of message-passing. UltraGCN [123] adopts negative sampling to contrast over-smoothing and additional objective terms to (i) approximate the infinite neighborhood exploration and (ii) mine relevant "unexpected" node-node interactions such as the item-item ones. Conversely, GFCF [163] translates the graph-based recommendation task into the graph signal processing domain to obtain a closed-form formulation for approximating the infinite neighborhood exploration. Given that such recent strategies do not *explicitly* perform the message-passing schema as presented above, in the remaining sections of this chapter, we will adopt the terms *explicit* and *implicit* message-passing as shorthands to denote the two model families, respectively.

4.2.5 *A taxonomy of graph CF approaches*

We propose (see Table 4.1) a taxonomy to classify the state-of-the-art graph models. The taxonomy considers the recurrent **strategy patterns** as emerged by conducting an in-depth review and analyzing the different graph CF approaches.

- **Node representation** indicates the representation strategy to model users' and items' nodes. It involves the *dimensionality* of node embeddings, and the possibility of *weighting* the neighbor node contributions.
- **Neighborhood exploration** refers to the procedure for exploring the multi-hop neighborhoods of each node to update the node latent representation. It involves the type of *node-node connections* which are explored, and the *message-passing* schema (i.e., *explicit* or *implicit* as previously defined).

In the next two sections, we will assess the performance of the graph CF models from the taxonomy in Table 4.1. Thus, we consider GCN-CF [101], GAT-CF [186], NGCF [200], LightGCN [84], DGCF [201], LR-GCCF [45], UltraGCN [123], and GFCF [163] for a total of eight graph CF solutions.

4.3 Experimental Settings and Protocols

In this section, we present the experimental details to conduct our analysis.

Datasets

As a pre-processing stage, for each dataset, we randomly sample 60k interactions and drop users and items with less than five interactions to avoid the cold-start effect [82, 83]. The final dataset statistics are: (1) Baby has 5,842 users, 7,925 items, 35,475 interactions; (2) Boys & Girls has 3,042 users, 12,912 items, 35,762 interactions; (3) Men has 3,909 users, 27,656 items, 51,519 interactions.

Reproducibility

Datasets are split using the 70/10/20 train/validation/test hold-out strategy. Baselines are trained through grid search (48 explored configurations), with a batch size of 256 and 400 epochs. Datasets and codes (implemented with Elliot [11]) are available at this link: `https://github.com/sisinflab/ECIR2023-Graph-CF`.

Evaluation

As for the *overall accuracy*, we use the recall (*Recall@k*) and the normalized discounted cumulative gain (*nDCG@k*). Concerning the *item exposure*, we focus on: (1) item novelty [184, 185] through the expected free discovery (*EFD@k*) measuring the expected portion of relevantly-recommended items that have already been seen by the users; (2) item diversity [160] with the 1's complement of the Gini index (*Gini@k*), a statistical dispersion measure which estimates how a model suggests heterogeneous items to users; (3) the average percentage of items from the long-tail (*APLT@k*) which are recommended in users' lists [3] to calculate recommendation's bias towards popular items. *User fairness* indicates how equally each user group receives accurate recommendations. Users are split into quartiles based on the number of items they interacted with. We then measure *UMADrat@k* and the *UMADrank@k* [55], where the former stands for the average deviation in the predicted ratings among users groups, while the latter represents the average deviation in the recommendation accuracy (calculated in terms of *nDCG@k*) among users groups. The best hyper-parameter configurations are found by considering *Recall@20* on the validation.

4.4 Taxonomy-aware evaluation

This section aims to answer RQ1 (*"Can we explain the variations observed when testing several graph models on overall accuracy, item exposure, and user fairness separately?"*) by showing how the proposed taxonomy of graph strategies can explain the recommendation evaluation on CP-Fairness and overall accuracy. We experiment with 48 hyper-parameter configurations to investigate various combinations of graph CF techniques for *message-passing, explored nodes, edge weighting,* and *latent representations*. Results refer to the Amazon Men dataset and top-20 lists (Table 4.2). Please note that we report the **best** metric result for each <dimension, value> pair (the

Table 4.2. Best metric results (and corresponding graph CF model) for each <dimension, value> pair, on the Amazon Men dataset for top-20 lists. **Bold** is used to indicate the best result in the pairs having a two-valued dimension, while † is used only for the "explored nodes" dimension to indicate also the best results on *same* and *different*. The symbols ↑ and ↓ indicate whether better stands for high or low values. We use *"rank"* and *"rat"* as the *UMADrank@k* and *UMADrat@k*.

Dimensions	**Values**	**Overall Accuracy**		**Item Exposure**			**User Fairness**	
		Recall↑	*nDCG*↑	*EFD*↑	*Gini*↑	*APLT*↑	*rank*↓	*rat*↓
Message passing	*implicit*	0.1222 (GFCF)	**0.0911** **(GFCF)**	**0.2615** **(GFCF)**	0.2871 (UltraGCN)	0.1808 (UltraGCN)	0.0123 (UltraGCN)	**0.0022** **(UltraGCN)**
	explicit	**0.1223** **(LR-GCCF)**	0.0884 (LR-GCCF)	0.2536 (LR-GCCF)	**0.5090** **(LR-GCCF)**	**0.3823** **(GAT-CF)**	**0.0002** **(DGCF)**	0.0169 (LightGCN)
Explored nodes	*same-1*	0.1221† (LR-GCCF)	0.0884† (LR-GCCF)	0.2500† (LR-GCCF)	0.4377 (LR-GCCF)	0.3433 (GAT-CF)	**0.0002**† **(DGCF)**	**0.0022**† **(UltraGCN)**
	same-2	0.1184 (LightGCN)	0.0841 (LightGCN)	0.2380 (LightGCN)	**0.5090**† **(LR-GCCF)**	**0.3823**† **(GAT-CF)**	**0.0002**† **(DGCF)**	0.0209 (NGCF)
	different-1	**0.1222**† **(GFCF)**	**0.0911**† **(GFCF)**	**0.2615**† **(GFCF)**	0.4093 (NGCF)	0.3424 (GAT-CF)	**0.0002**† **(DGCF)**	**0.0022**† **(UltraGCN)**
	different-2	0.1210 (DGCF)	0.0850 (DGCF)	0.2407 (LightGCN)	0.4934† (LR-GCCF)	0.3438† (LR-GCCF)	**0.0002**† **(DGCF)**	0.0388 (LightGCN)
Weighting	*weighted*	0.1210 (DGCF)	0.0857 (DGCF)	0.2428 (DGCF)	0.3240 (DGCF)	**0.3823** **(GAT-CF)**	**0.0002** **(DGCF)**	0.0301 (DGCF)
	unweighted	**0.1223** **(LR-GCCF)**	**0.0884** **(LR-GCCF)**	**0.2536** **(LR-GCCF)**	**0.5090** **(LR-GCCF)**	0.3438 (LR-GCCF)	0.0101 (GCN-CF)	**0.0169** **(LightGCN)**
Latent representations	*emb-64*	0.1193 (LR-GCCF)	0.0871 (LR-GCCF)	0.2479 (LR-GCCF)	**0.5090** **(LR-GCCF)**	0.3627 (GAT-CF)	**0.0002** **(DGCF)**	0.0054 (UltraGCN)
	emb-128	0.1221 (LR-GCCF)	0.0883 (LR-GCCF)	**0.2536** **(LR-GCCF)**	**0.5090** **(LR-GCCF)**	0.3644 (GAT-CF)	**0.0002** **(DGCF)**	0.0111 (UltraGCN)
	emb-256	**0.1223** **(LR-GCCF)**	**0.0884** **(LR-GCCF)**	0.2532 (LR-GCCF)	0.5038 (LR-GCCF)	**0.3823** **(GAT-CF)**	**0.0002** **(DGCF)**	**0.0022** **(UltraGCN)**

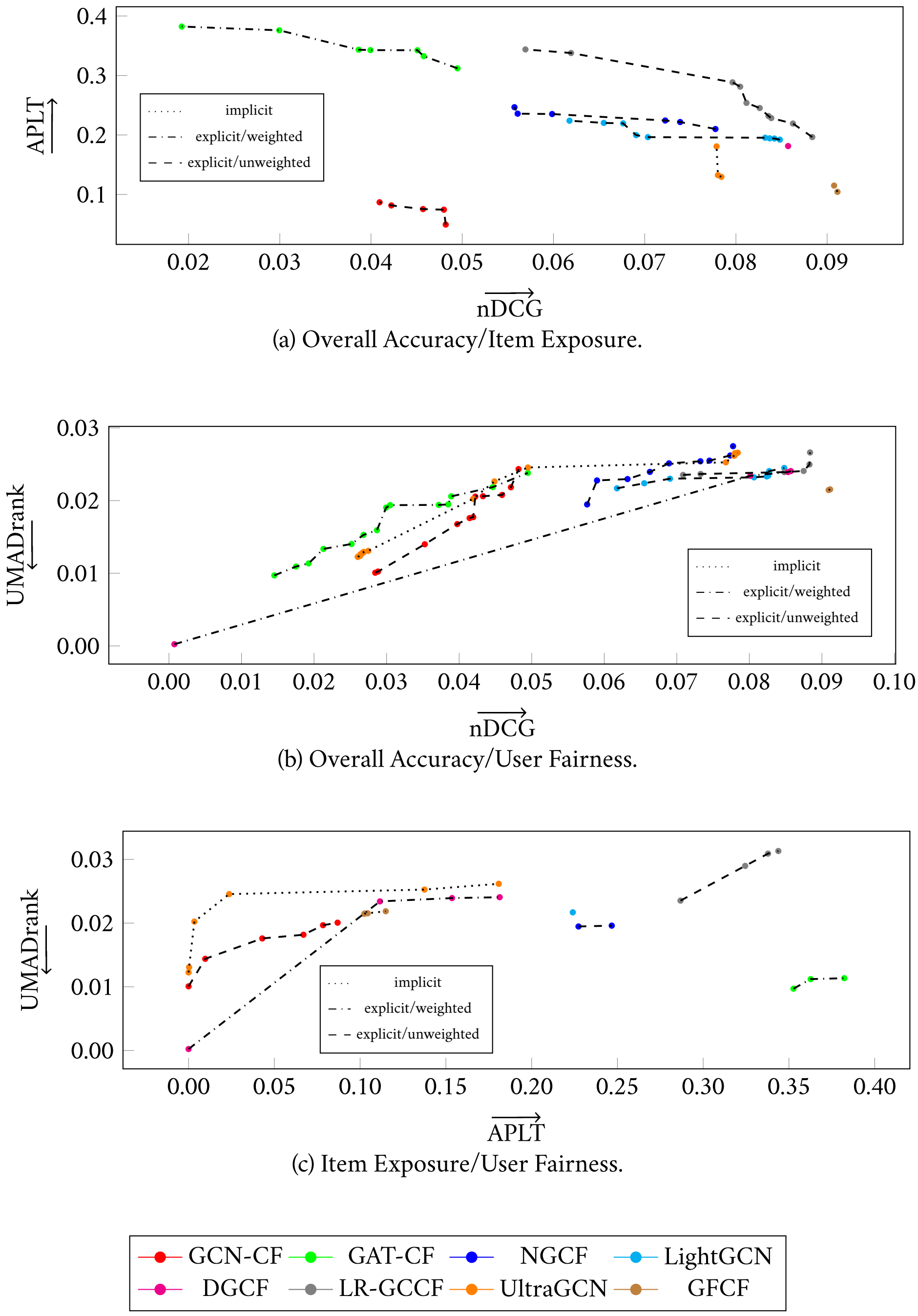
corresponding best graph recommendation model is displayed below each metric result) to ease the interpretation of results and provide meaningful insights.

- **Message-passing.** We investigate the two widely-recognized message-passing strategies: *implicit* and *explicit*. The most obvious pattern indicates that both sets have almost the same number of top-performing models in each of the evaluation criteria. *Explicit* graph approaches perform better on item exposure, where they outperform *implicit* techniques (i.e., on *Gini* and *APLT*) two out of three times by a significant margin. On the one hand, this tendency may be due to the absence of a direct message (information) propagating along the user-item graph in *implicit* techniques, which prevents the user node from exploring vast item segments. On the other hand, it appears that models from both families perform similarly on accuracy and user fairness, indicating that there is no obvious reason to favor *implicit* over *explicit* or vice versa.

- **Explored nodes.** Here, we examine four methods to explore nodes (adopting the message-passing re-formulation from Equation (4.2)): *same* and *different*, with 1 and 2 hops. Similarly to the trend found for the message-passing dimension, the results demonstrate that the two primary categories (*same* and *different*) are nearly equally performing across all measurements, with *same-2* and *different-1* being the prominent ones. In detail, the *different-1* exploration outperforms the *same-2* on the overall accuracy level (GFCF is the leading model here). Conversely, *same-2* is the best strategy for item exposure (with LR-GCCF and GAT-CF leading). As observed for the message-passing, user fairness does not give a reason to choose between *same* and *different*. The exploration of 1 hop in *same* and *different* settings is the preferable technique, even if 2 hops connections lead to a better item exposure.
- **Weighted.** This study examines *weighted* and *unweighted* graph CF techniques. Differently from above, we observe that *unweighted* solutions provide the best performance on almost all CP-fairness metrics, with LR-GCCF steadily being the superior approach. The only trend deviation refers to GAT-CF (i.e., a *weighted* method) surpassing *unweighted* solutions on the *APLT* level, that is, recommending items from the long-tail. The behavior is likely attributable to the design of *weighted* techniques, which can investigate farther neighbors of the *ego* node (observe the performance of GAT-CF on the *same-2* dimension), leading user profiles to match distant (and possibly niche) products in the catalog. On the contrary, it is interesting to notice how the other two metrics accounting for item exposure (i.e., *EFD* as item novelty measure and *Gini* as item diversity measure) seem to privilege *unweighted* graph techniques (i.e., LR-GCCF). The observed behaviors differ as the three metrics provide completely different perspectives of the *item exposure*, and thus they are uncorrelated.
- **Latent representations.** We compare the performance of graph CF techniques adopting latent representations with *64*, *128*, and *256* features, respectively. It is worth noticing that higher latent representations (i.e., *128* and *256*) result in better performance on all measurements. Specifically, it appears that the *128* dimension is the turning point after which the trend becomes stable (i.e., the metric values for *128* and *256* are frequently comparable). This may be an important insight since the majority of research works in recent literature tend to employ *64*-embedded representations of nodes without exploring further dimensionalities (see Table 4.1 as a reference).

4.5 Trade-off Analysis

This section analyses how the graph CF baselines balance the trade-off among accuracy, item exposure, and user fairness, and aims to answer RQ2 (*"How and why nodes representation and neighborhood exploration algorithms can strike a trade-off between overall accuracy, item exposure, and user fairness?"*). We report the results only for the Amazon Men dataset. The negative Pearson correlation values for accuracy/item exposure (*nDCG/APLT*) and accuracy/user fairness (*nDCG/UMADrank*)



(a) Overall Accuracy/Item Exposure.

(b) Overall Accuracy/User Fairness.

(c) Item Exposure/User Fairness.

Figure 4.2. Overall Accuracy/Item Exposure, Overall Accuracy/User Fairness, and Item Exposure/User Fairness trade-offs on Amazon Men, assessed through *nDCG/APLT*, *nDCG/UMADrank*, and *APLT/UMADrank*, respectively. Each point depicts a model hyper-parameter configuration set belonging to the corresponding Pareto frontier. Colors refer to a particular baseline, while lines styles discern their technical strategies based on the proposed taxonomy. Arrows indicates the optimization direction for each metric on x and y axes.

suggest that a trade-off may be necessary, and desirable. In addition, the same correlation metric indicates the necessity of a trade-off for item exposure/user fairness (*APLT*/*UMADrank*). Among the strategy patterns identified in the proposed taxonomy (see Table 4.1), we select the most important architectural dimensions, **message-passing** and **weighting** of graph edges, to conduct this study. In detail, the analysis studies three combined categories: (1) models with implicit message-passing (denoted as *implicit*); (2) models with explicit message-passing and neighborhood weighting (denoted as *explicit/weighted*); (3) models with explicit message-passing without neighborhood weighting (denoted as *explicit/unweighted*). For each analyzed trade-off, we select the Pareto optimal solutions of the baselines laying on the model-specific Pareto frontier [187]. Figure 4.2 plots graph models Pareto frontiers in the common *objective function spaces* related to the considered trade-offs. The careful reader may notice the different axis' scales across the graphics due to the metric values. The colors of Pareto optimal solutions are model-specific, while the line style is used to distinguish the categories: dotted lines for *implicit*, dash-dot lines for *explicit/weighted*, and dashed lines for *explicit/unweighted*.

- **Accuracy/Item Exposure.** Figure 4.2a shows that the *explicit/weighted* models exhibit a trade-off, as they maximize either *nDCG* (i.e., DGCF) or *APLT* (i.e., GAT-CF), but not both. This is expected since DGCF is designed as a version of GAT-CF with improved accuracy. It is worth mentioning that DGCF's trade-off is reached at the expense of item exposure. In contrast to these models, *explicit/unweighted* baselines show a balanced trade-off because they do not prioritize accuracy or item exposure exclusively. In detail, LR-GCCF provides the best performance in terms of *nDCG* and *APLT* simultaneously. From a visual inspection, LR-GCCF's Pareto frontier dominates those of the other *explicit/unweighted* models. Conversely, GCN-CF exhibits the worst trade-off because it is neither ideal for *nDCG* nor *APLT*. As for the *implicit* models, they appear to prioritize precision over the provision of long-tail items.

  Under this lens, the latest (i.e., implicit) approaches seem to increase accuracy, even if this is to the detriment of the niche items exposure.
- **Accuracy/User Fairness.** To ease the interpretation of Figure 4.2b, we recall that *UMADrank* (used to measure User Fairness) measures to what extent the model ranking performance differs among the user groups (partitioned based on their activity on the platform). Figure 4.2b shows that, for GAT-CF and GCN-CF, the poor performance in terms of *nDCG* is associated with high variability in terms of user fairness. In fact, for these two models, the *UMADrank* value indicates high variability across user groups. Something different emerges for models such as NGCF, LightGCN, LR-GCF, and GFCF. These models, GFCF in particular, exhibit valuable recommendation accuracy with better stability in terms of ranking performance across the different user groups. As a consequence, the Pareto frontiers associated with these models dominate the others. In detail, GFCF is the best-performing one regarding this trade-off. Conversely, UltraGCN and DCGF do not show consistent behavior demonstrating a strong sensitivity to

the chosen hyper-parameters set.

In this setting, no graph CF strategy emerges as the absolute winner. Specifically, every graph CF strategy is not enough to guarantee adequate fairness among different user groups. Then, the positive results are associated with particular configurations of some models and are lost when the hyper-parameter set changes.

- **Item Exposure/User Fairness.** The trade-off indicates to what extent graph CF models can treat final users fairly and recommend items from the long tail. In Figure 4.2c, it is possible to identify two groups of baselines: the models that show poor performance in terms of item exposure (UltraGCN, DGCF, GCN-CF, and GFCF) and the models that exhibit an acceptable exposure for long-tail items (LightGCN, NGCF, LR-GCCF, and GAT-CF). In detail, a cluster of models that belong to the *explicit/unweighted* category stands out in this second group. Not only are these models able to recommend niche items, but also they are stable (among the user groups) in terms of accuracy. On the contrary, although GAT-CF lies close to the *utopia point*, it exhibits greater variability regarding the accuracy metric. Indeed, comparing Figure 4.2c with Figure 4.2a, GAT-CF demonstrates to achieve adequate user fairness, but its performance is still very poor in terms of accuracy.

  To summarize, even if a system designer could be more interested in promoting models solely guaranteeing the best value for APLT (Producer Fairness), the explicit/unweighted strategies can generally ensure a satisfactory (for Consumers and Producers) trade-off between user fairness and item exposure.

4.6 Summary

In this chapter, we assess the performance of graph CF models on Consumer and Producer (CP)-fairness metrics showing that their superior accuracy capabilities is reached at the expense of user fairness, item exposure, and their combination. By recognizing nodes representation and neighborhood exploration as the two main dimensions of a novel graph CF taxonomy, we study their influence on CP-fairness and overall accuracy separately and simultaneously. The outcomes raise concerns about the effective application of recent approaches in graph CF (e.g., implicit message-passing techniques). On such basis, we are performing further investigations on other datasets and algorithms, and we are working on new graph models balancing accuracy and CP-Fairness.

Chapter 5

Unveiling the Potential of Recommender Systems without Prioritizing Accuracy

Although beyond-accuracy metrics have gained attention in the last decade, the accuracy of recommendations is still considered the gold standard to evaluate Recommender Systems (RSs). This approach prioritizes the accuracy of recommendations, neglecting the quality of suggestions to enhance user needs, such as diversity and novelty, as well as trustworthiness regulations in RSs for user and provider fairness. As a result, single metrics determine the success of RSs, but this approach fails to consider other criteria simultaneously. A downside of this method is that the most accurate model configuration may not excel in addressing the remaining criteria. This study seeks to broaden RS evaluation by introducing a multi-objective evaluation that considers all model configurations simultaneously under several perspectives. To achieve this, several hyper-parameter configurations of an RS model are trained, and the Pareto-optimal ones are retrieved. While in the previous chapter we have utilized the Pareto frontiers to qualitatively discuss about multi-objective evaluation of graph-based methods, we know employ the ***Quality Indicators*** **(QI)** of Pareto frontiers. QIs enable quantitatively evaluating the model's performance by considering various configurations and giving the same importance to each metric. The experiments show that this multi-objective evaluation overturns the ranking of performance among RSs, paving the way to revisit the evaluation approaches of the RecSys research community. We release codes and datasets in the following GitHub repository: `https://github.com/sisinflab/RecMOE`.[1]

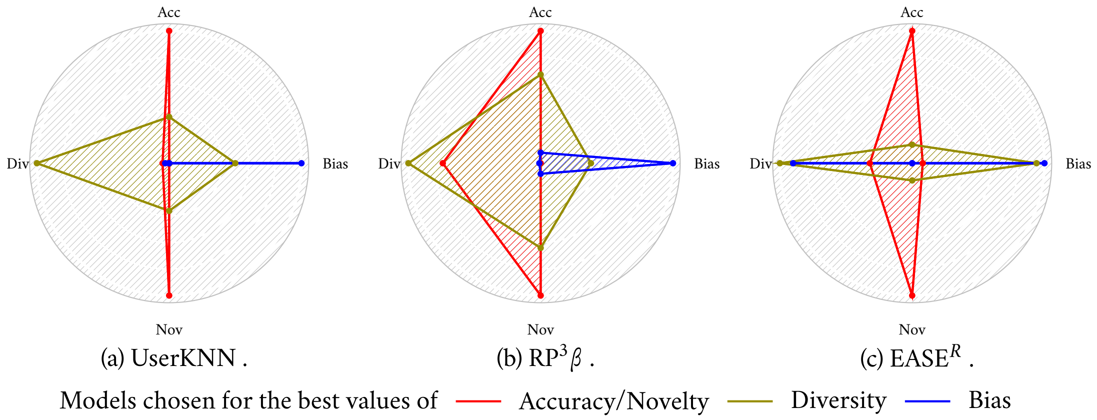
[1] This chapter is based on the work published in the *Proceedings of the 17th ACM Conference on Recommender Systems* (RecSys 2023) "Broadening the Scope: Evaluating the Potential of Recommender Systems beyond prioritizing Accuracy".

5.1 Introduction and Motivation

The success of Recommender Systems (RSs) is often measured by its ability to accurately predict a user's preferences and suggest relevant items. However, other beyond-accuracy metrics have been proposed to capture different aspects of recommendation quality, such as diversity and novelty of suggestions [152, 168, 185], and fairness issues [29, 111, 228]. While beyond-accuracy metrics have gained momentum in the RecSys research community, accuracy of suggestions is still consistently prioritized over the other facets of recommendation [14, 20]. The common practice is to select the best model solely based on the accuracy metrics (e.g., nDCG, Recall, or Precision), which limits the consideration of performance on beyond-accuracy metrics. Consequently, the best model in terms of accuracy may not guarantee the best performance in terms of diversity, novelty, or fairness, and vice versa. This limitation in choosing the best models may result in a lack of information on the actual behavior of RS models across multiple perspectives of recommendation. In this regard, we provide a motivating example by training 32 hyper-parameter settings of three baselines (i.e., EASE^R [171], RP$^3\beta$ [139], and UserKNN [151]) on the Goodreads dataset[2]. Figure 5.1 shows the min-max normalized values of recommendation algorithm performance by selecting the best hyper-parameter settings for each baseline. We do this based on the best values of various metrics representing accuracy (nDCG), novelty (EPC) [185], diversity (1 - Gini coefficient) [92], and algorithmic bias (APLT) [4] evaluation perspectives. When selecting the model based on the highest value of a given metric, a larger shape area on the resulting graph indicates reasonably high values of the other metrics. As expected, we find that the selection strategy for the best model tremendously impact the other metrics. Namely, selecting the best hyper-parameter setting according to accuracy guarantees the best value of novelty, but leads to sub-optimal value of diversity and worse value of algorithmic bias, and vice versa. Then, assessing a model's performance for each metric, for example after selecting it based solely on accuracy, results in a lack of knowledge about the potential of the model on beyond-accuracy metrics. Hence, the need of a *multi-objective evaluation* emerges to *simultaneously* assess the models' performance on several criteria, even though the training of such models could still aim to maximize the accuracy of recommendation (e.g., to choose the best iteration, or trigger a stopping condition in the training phase).

To address this problem of multi-objective evaluation, we exploit the definition of Pareto optimality from the Multi-Objective Optimization (MOO) theory [124]. Given a set of objectives to maximize, we define a specific hyper-parameter setting of a model as a Pareto-optimal solution if there is no other setting that improves at least one objective function without hurting another one. The set of such Pareto-optimal configurations composes the so-called Pareto frontier [204]. An approach to consider simultaneously more metrics in the evaluation would be to select a solution from the Pareto frontier through well-known methods (e.g., hypervolume [224]). However,

[2] More details on the experimental settings will be provided in Section 5.3.



(a) UserKNN . (b) $\mathrm{RP}^3\beta$. (c) EASE^R .

Figure 5.1. Kiviat diagrams indicating the performance of the models on the Goodreads dataset. The models are selected according to different metrics for each objective (i.e., Accuracy/Novelty, Diversity, and Bias). Higher means better.

evaluating a specific configuration of a model only provides information on that particular setting and fails to provide insights into the overall potential of the model. Therefore, to enhance the multi-objective evaluation of RSs, we need to assess the entire set of Pareto-optimal configurations of a model. Simply visualizing the Pareto frontier only enables qualitative analysis, being challenging when multiple objectives are involved. We propose to introduce in RSs research the ***Quality Indicators***, previously adopted in the literature of MOO [109], which are designed to evaluate Pareto frontiers by providing a real number to quantify and rank the performance of a model corresponding to a Pareto frontier under different perspectives. To the best of our knowledge, QIs have already been exploited to evaluate Pareto frontiers — mostly their relative dominance — generated by evolutionary algorithm [43, 71, 76] applied in the context of Multi-Objective RSs [218, 224]. In contrast, we aim to use them to offer insights into unexplored aspects of traditional RSs. In detail, the contributions of our work are:

- We experimentally show the negative impact of prioritizing recommendation accuracy over other important metrics and motivate the need of a multi-objective evaluation of RSs models. The results emphasize the importance of a more comprehensive evaluation approach to ensure a thorough understanding of RS behavior across multiple dimensions.
- We train 32 hyper-parameter settings of 5 state-of-the-art recommendation models using 3 public datasets. We compute the Pareto frontier in two multi-objective scenarios to provide a exhaustive evaluation of the recommendation models.
- To enhance the multi-objective evaluation of RSs, we evaluate various models under different scenarios simultaneously by utilizing the ***Quality Indicators*** of Pareto frontiers to enable an even more comprehensive analysis of RSs.

5.2 Quality Indicators

In this section, we present the Quality Indicators (QIs) to assess the Pareto frontiers corresponding to an RS model. Indeed, we aim to perform a multi-objective evaluation of RSs. QIs are devised to measure multiple quality facets of a Pareto frontier. Hence, they can be classified according to the quality they assess. Among the selected QIs for this work, they can be divided as follows [110]: (i) QIs for spread, (ii) QIs for uniformity, (iii) QIs for cardinality, and (iv) QIs that consider all these quality aspects.

Spread QIs

The QIs for Spread indicate the range of the Pareto-optimal solutions on the Pareto frontier. For our study, we use the Maximum Spread ($\mathcal{MS}$) [231]. Specifically, this spread indicator measures the range of a Pareto frontier by considering the maximum extent of each objective.

Definition 5.1 (Maximum Spread)**.** *Given the Pareto-optimal solutions set A and the number of objectives m, $\mathcal{MS}$ is defined as:*

$$\mathcal{MS}(A) = \sqrt{\sum_{j=1}^{m} \max_{a,a' \in A} (a_j - a'_j)^2}, \tag{5.1}$$

where a and a′ are solutions belonging to A. The higher the value, the better the extensiveness of the curve.

Uniformity QIs

The uniformity of a Pareto frontier provides information about the distribution of the solutions. A higher uniformity of the curve denotes that the solutions are less dispersed, while a low uniformity indicates more diversity within the set. In the case of RSs, having low uniformity leads to a wide range of options for decision-makers. Specifically, we employ the Spacing metric ($\mathcal{SP}$) [157] that measures the variation in the Manhattan distances between the Pareto-optimal solutions.

Definition 5.2 (Spacing)**.** *Given the N Pareto-optimal solutions $a_i \in A$ and the number of objectives m, $\mathcal{SP}$ is defined as:*

$$\mathcal{SP}(A) = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (\bar{d} - d_1(a_i, A/a_i))^2}, \tag{5.2}$$

with $d_1(a_i, A/a_i) = \min_{a \in A/a_i} \sum_{j=1}^{m} \left|a_{ij} - a_j\right|$, where $\bar{d}$ is the mean of all the Manhattan distances $d_1(a_1, A/a_1)), \dots, d_1(a_N, A/a_N))$ and a_{ij} represents the j-th objective of the solution a_i. The lower the value, the more concentrated the solutions are on the Pareto frontier. However, an $\mathcal{SP} = 0$ indicates that all the solutions could be equidistant.

The interpretation of $\mathcal{SP}$ is strictly related to $\mathcal{MS}$.

Cardinality QIs

Given a set of generic solutions, the QIs for cardinality determine the proportion of Pareto-optimal solutions in this set. A well-known QI for cardinality is the Specifically, Error Ratio ($\mathcal{ER}$) [183].

Definition 5.3 (Error Ratio)**.** *Given K generic solutions belonging to the set B, $\mathcal{ER}$ is defined as:*

$$\mathcal{ER}(B) = \frac{\sum_{b \in B} e(b)}{K}, \tag{5.3}$$

with $e(b) = 1$ if b is a Pareto-optimal solution, 0 otherwise. A higher $\mathcal{ER}$ value indicates greater Pareto-optimal solutions in the set B.

All quality aspects QI.

The QIs included in this category provide insights into the spread, uniformity, and cardinality of the Pareto frontiers simultaneously. Among them, the Hypervolume ($\mathcal{HV}$) [233] is a volume-based QI that measures the volume of the objective function space dominated by the Pareto frontier.

Definition 5.4 (Hypervolume)**.** *Given the Pareto-optimal solutions $a \in A$ and a reference point r, $\mathcal{HV}$ is defined as:*

$$\mathcal{HV}(A) = \lambda \left(\bigcup_{a \in A} \{x \mid a \prec x \prec r\} \right), \tag{5.4}$$

where λ denotes the Lebesgue measure. The larger the hypervolume, the better the solution set is.

5.3 Experiments

Given a set of multiple metrics to assess simultaneously, we aim to answer the following research questions:

RQ1: To what extent can the models provide Pareto-optimal configurations? Are these configurations uniformly distributed, or are they dispersed enhancing diverse solutions to the trade-off?

RQ2: Which model has the Pareto frontier that simultaneously offers better solutions on multiple metrics?

5.3.1 Experimental Setup

We now provide details about the experimental setup to conduct the experiments of this work.

Datasets

We select three different datasets to cover several domains. Specifically, we use *Amazon Music* (music domain), *Goodreads* [196] (book domain), and *Movielens1M* [81] (movie domain). Regarding Goodreads (18892 users, 25475 items, 1378033 interactions, 0.99 sparsity) and Movielens1M (6040 users, 3706 items, 1000209 interactions, 0.95 sparsity), we do not apply any pre-processing step, while we obtain a pre-processed version of the Amazon Music dataset from work by Anelli et al. [13] (14354 users, 10027 items, 145523 interactions, 0.99 sparsity).

Baselines and Hyper-parameters Settings Exploration

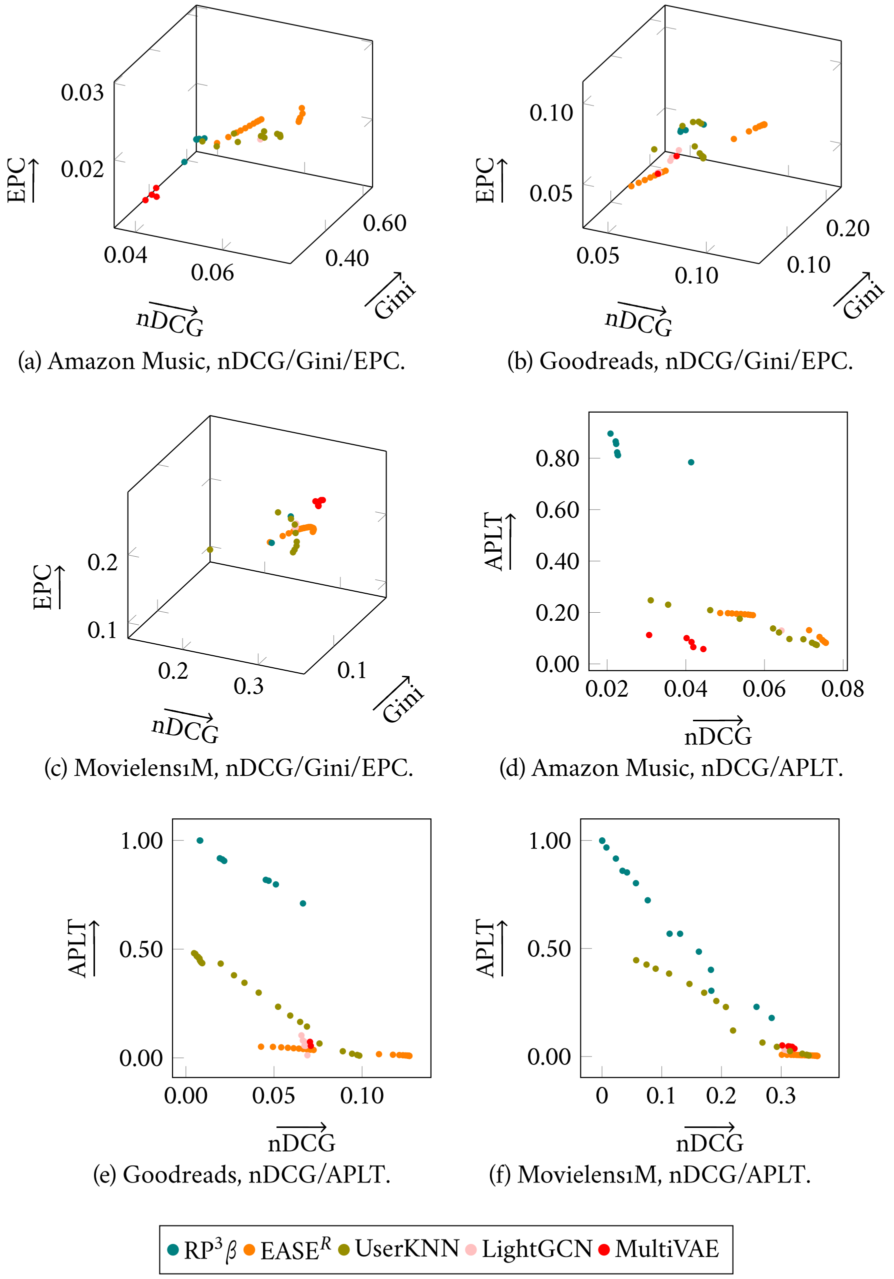
We train five recommendation algorithms, i.e., EASE^R [171], MultiVAE [112], LightGCN [84], RP$^3\beta$ [139], and UserKNN [151]. Specifically, we train 32 hyper-parameter values combinations of each model by exploiting the Elliot framework [11]. We define the set of hyper-parameters values for these baselines from previous works [14, 15]. We provide complete information on the explored values in the GitHub repository. We set nDCG@10 as the optimization target. MultiVAE and LightGCN are trained with a batch size of 256 and 300 epochs by applying the early stopping strategy with patience of 10.

Metrics

We assess the baselines' performance under several perspectives. We compute nDCG, Precision, and Recall for the accuracy of recommendations. From the final user point of view, we evaluate the diversity (with Gini index [92] and Item Coverage) and novelty (with EPC and EFD [185]). Finally, we measure the popularity bias of the recommendations with APLT [4] – the greater, the better – and ARP [92] – the less, the better. All these metrics refer to cutoff 10.

Multi-Objective Evaluation Methodology

We clarify how we obtain the Pareto frontiers corresponding to each baseline to evaluate them through the quality indicators described in Section 5.2. Given the experimental setup described above, we can identify a subset of the computed metrics to compose a multi-dimensional objective function space. Each single hyper-parameters configuration of a model represents a solution in this space since we have computed their performance values regarding such metrics. As a result, we obtain 32 points in the objective function space for each baseline. Among these points, we can identify the Pareto-optimal configurations, which lay on the Pareto frontier. Consequently, given an objective function space designated by a set of metrics, we gather five Pareto frontiers, each corresponding to one trained baseline. Once the Pareto-optimal solutions composing the Pareto frontiers are identified, we can exploit the QIs to evaluate the Pareto frontiers of the models.



(a) Amazon Music, nDCG/Gini/EPC.

(b) Goodreads, nDCG/Gini/EPC.

(c) Movielens1M, nDCG/Gini/EPC.

(d) Amazon Music, nDCG/APLT.

(e) Goodreads, nDCG/APLT.

(f) Movielens1M, nDCG/APLT.

Figure 5.2. Pareto optimal solutions plots for Amazon Music, Goodreads, and MovieLens1M. The first row refers to the nDCG/Gini/EPC scenario, and the second row refers to the nDCG/APLT scenario. The arrows indicate the optimal directions.

Table 5.1. Classical analysis of the baselines' results in terms of Accuracy, Diversity, Novelty, and Bias of recommendations. The arrows indicates the descending or ascending order for the best solution. Best values are in bold. Second best values are underlined.

Model	**nDCG↑**	**Recall↑**	**Precision↑**	**Gini↑**	**IC↑**	**EPC↑**	**EFD↑**	**APLT↑**	**ARP↓**
Amazon Music									
EASE^R	**0.07560**	**0.09481**	**0.02049**	0.25846	8891	**0.02863**	**0.34370**	0.08196	37.6760
UserKNN	0.07329	0.09424	0.02004	0.21426	8361	0.02741	0.32669	0.07363	42.7840
MultiVAE	0.04446	0.06264	0.01269	0.22379	6556	0.01606	0.19478	0.05773	28.4834
LightGCN	0.06433	0.08632	0.01797	0.33387	**9121**	0.02355	0.28666	0.12980	28.1607
RP$^3\beta$	0.04136	0.05070	0.01071	**0.44327**	8973	0.01521	0.20087	**0.78420**	**4.46494**
Goodreads									
EASE^R	**0.12685**	**0.08278**	**0.09680**	0.04144	6842	**0.10599**	**1.23522**	0.00882	475.874
UserKNN	0.09842	0.06533	0.07416	0.02873	6434	0.08117	0.92929	0.01021	587.527
MultiVAE	0.07090	0.04812	0.05718	0.05126	7387	0.05974	0.69948	0.05533	443.142
LightGCN	0.06896	0.04835	0.05352	0.06434	7729	0.05722	0.68752	0.01176	356.040
RP$^3\beta$	0.06645	0.04177	0.05066	**0.19076**	**14941**	0.05759	0.78194	**0.71016**	**64.3545**
Movielens1M									
EASE^R	**0.36075**	**0.15574**	**0.32462**	0.06152	980	**0.27472**	**3.22977**	0.00260	1198.44
UserKNN	0.34603	0.14980	0.31189	0.04556	920	0.25320	3.01901	0.00462	1305.30
MultiVAE	0.32223	0.14189	0.29147	**0.12550**	**1836**	0.25631	3.00231	0.03657	1002.73
LightGCN	0.31087	0.13204	0.28113	0.09899	1481	0.24170	2.84602	0.02806	1046.17
RP$^3\beta$	0.28403	0.12287	0.27017	0.09266	1588	0.21789	2.58115	**0.17851**	**961.877**

We carry out the multi-objective evaluation by identifying two different evaluation scenarios. On the one hand, we focus on user-centered objectives (accuracy, diversity, and novelty of recommendations). This scenario leads to a three-dimensional space in which the axes are nDCG, Gini index, and EPC. On the other hand, we compare the accuracy of recommendations against the algorithmic bias, by obtaining a two-dimensional objective function space (nDCG vs. APLT). Figure 5.2 depicts the Pareto frontiers of the models trained on each datasets for the two evaluation scenarios.

5.3.2 Results and Discussion

To commence the experimental assessment, we establish a benchmark for the upcoming investigation. In detail, a preliminary analysis of the baselines' performance is conducted by reporting the results of the best configurations according to the values of nDCG@10 in Table 5.1. This analysis serves as context and motivates the subsequent exploration where QIs of the Pareto frontiers are utilized to answer the research questions (Table 5.2).

A "traditional" analysis of recommendation performance

The results in Table 5.1 corroborate the recent literature findings [13, 50]. For the three datasets, EASE^R and UserKNN are the models providing the most accurate recommendations. Observing the novelty metrics, the accuracy and novelty of recommendations exhibit a positive correlation. However, we arrive at very different conclusions by examining the other beyond-accuracy metrics. On the one hand, concerning the diversity of recommendations, the remaining models (LightGCN, MultiVAE, RP$^3\beta$) generally perform better than EASE^R and UserKNN across all datasets. On the other hand, RP$^3\beta$ consistently outperforms its competitors in addressing the popularity bias. This peculiar performance puzzle does not offer insight into the general behaviour of the model or whether other instances of it follow a similar performance trend. To unravel this puzzle, we shift to a multi-objective evaluation-based analysis aimed at assessing the recommendation performance under several criteria simultaneously.

Distribution of Pareto-optimal configurations

To answer RQ1, we examine the values of **Error Ratio** ($\mathcal{ER}$), **Maximum Spread** ($\mathcal{MS}$), and **Spacing metric** ($\mathcal{SP}$). Different scenarios may arise when examining the behaviour of a model. Firstly, when the model yields higher $\mathcal{ER}$, $\mathcal{MS}$, and $\mathcal{SP}$ values, it suggests that the model's configurations are widely spread and varied, implying that it can provide multiple solutions on the Pareto frontier. Secondly, suppose the model exhibits higher $\mathcal{ER}$ and $\mathcal{MS}$ values but lower $\mathcal{SP}$ values. In that case, it indicates that the model's settings are dispersed but concentrated in certain areas of the objective function space. This behaviour could result in fewer solutions on the Pareto frontier. Thirdly, if the model has higher values of $\mathcal{ER}$ and lower values of $\mathcal{MS}$ and $\mathcal{SP}$, it implies that the model can offer various Pareto-optimal settings, which are all concentrated in the same area of the objective function space. Finally, a low number of Pareto-optimal configurations can indicate some issues with the solutions' characteristics, regardless of the $\mathcal{MS}$ and $\mathcal{SP}$ values.

Our investigation begins with the nDCG/APLT metrics for the Movielens1M dataset (as shown in Table 5.2), with Figure 5.2f illustrating the results for a better understanding. Within this context, RP$^3\beta$ provides a broad range of acceptable solutions ($\mathcal{ER}$=0.47) with a wide dispersion (highest value of $\mathcal{MS}$), and the solutions are dispersed along the entire Pareto frontier (highest value of $\mathcal{SP}$). Therefore, RP$^3\beta$ offers various solutions for an optimal trade-off between recommendation accuracy and algorithmic bias. UserKNN exhibits similar behaviour, with the second highest values for $\mathcal{ER}$, $\mathcal{MS}$, and $\mathcal{SP}$ (0.5, 0.53, and 0.02, respectively). In contrast, EASE^R offers a limited choice, featuring a not extensive and highly concentrated frontier (low values of $\mathcal{MS}$ and $\mathcal{SP}$), despite having numerous solutions on the frontier (highest value of $\mathcal{ER}$). Finally, MultiVAE and LightGCN present a limited number of Pareto-optimal configurations (lowest $\mathcal{ER}$ values), which influence the quality of their Pareto frontiers regarding range and spacing. As illustrated in Figure 5.2f, QIs provide an adequate and quantitative depiction of the models' behaviour. We can then extend our scrutiny to the remaining datasets. UserKNN, RP$^3\beta$, LightGCN, and MultiVAE maintain their respective performance across the Amazon Music (Figure 5.2d) and Goodreads (Figure 5.2e) datasets. Upon examination of Table 5.2, for these datasets, EASE^R demonstrates higher $\mathcal{MS}$ values than the one

Table 5.2. Quality Indicators of the Pareto frontiers results for the identified scenarios. The arrow indicates the descending or ascending order for the best solution. $\mathcal{SP}$ has no specific order of solutions, since its interpretation is strictly connected with the MS indicator. C counts how many solutions lay on the Pareto frontier.

Model	**Objectives**									
	Accuracy / Novelty / Diversity					**Accuracy / Bias**				
	$\mathcal{HV}\uparrow$	$\mathcal{ER}\uparrow$	$\mathcal{MS}\uparrow$	$\mathcal{SP}$	$C\uparrow$	$\mathcal{HV}\uparrow$	$\mathcal{ER}\uparrow$	$\mathcal{MS}\uparrow$	$\mathcal{SP}$	$C\uparrow$
					Amazon Music					
EASE^R	**0.00095**	**0.46875**	0.24986	0.01476	**15**	0.01355	**0.43750**	0.11886	0.00669	**14**
UserKNN	0.00082	0.34375	**0.29452**	0.00496	11	0.01448	0.34375	**0.17871**	0.00980	11
LightGCN	0.00051	0.06250	0.01335	0.00000	2	0.00835	0.03125	0.00000	0.00000	1
MultiVAE	0.00022	0.12500	0.09656	0.01738	4	0.00468	0.15625	0.05629	0.00351	5
RP$^3\beta$	0.00039	0.18750	0.20753	0.05888	6	**0.03489**	0.21875	0.11336	0.01173	7
					Goodreads					
EASE^R	0.00074	**0.59375**	0.09910	0.00227	**19**	0.00439	0.65625	0.09433	0.00214	21
UserKNN	**0.00110**	0.31250	**0.19889**	0.01287	10	0.02267	**0.71875**	**0.48042**	0.01471	**23**
LightGCN	0.00051	0.18750	0.06743	0.00783	6	0.00696	0.18750	0.09180	0.01536	6
MultiVAE	0.00043	0.06250	0.05022	0.00000	2	0.00521	0.06250	0.01827	0.00000	2
RP$^3\beta$	0.00083	0.12500	0.05584	0.01213	4	**0.05544**	0.28125	0.29529	0.02657	9
					Movielens1M					
EASE^R	0.00865	**0.68750**	0.09833	0.00446	**22**	0.00281	**0.65625**	0.06001	0.00196	**21**
UserKNN	**0.01296**	0.28125	**0.30929**	0.03641	9	0.08191	0.50000	0.52723	0.01810	16
LightGCN	0.00807	0.18750	0.01012	0.00287	6	0.00974	0.15625	0.00617	0.00181	5
MultiVAE	0.01216	0.21875	0.03419	0.00427	7	0.01639	0.18750	0.02528	0.00293	6
RP$^3\beta$	0.00839	0.06250	0.03796	0.00000	2	**0.14014**	0.46875	**0.86913**	0.03228	15

for Movielens1M. The corresponding Pareto frontiers are broader (higher $\mathcal{MS}$), but the solutions are concentrated into two well-separated clusters (lower $\mathcal{SP}$). This outcome emphasizes that EASE^R leaves the intermediate area between these clusters uncovered, being incapable of offering a balanced optimal trade-off between the two objectives. Let us focus on the user-centric scenario, where our objectives include nDCG/Gini/EPC, as shown in Figures 5.2a, 5.2b, and 5.2c. It is worth noting that UserKNN has proven its proficiency in generating several well-diversified hyper-parameter configurations across all datasets. This model boasts the best or second-best values of $\mathcal{ER}$ and $\mathcal{MS}$, along with high $\mathcal{SP}$ values, particularly for the Goodreads and Movielens1M datasets. However, LightGCN and MultiVAE exhibit subpar performance considering the number of Pareto-optimal configurations and their distribution, while EASE^R boasts a wide Pareto frontier but is confined to specific regions, failing to cover the central (and more balanced) area. In contrast, RP$^3\beta$ behaves differently from the previous scenario, providing fewer solutions on the Pareto frontier for the accuracy/diversity/novelty trade-off.

In summary, in response to RQ1, we can assert that UserKNN provides several diversified optimal solutions that effectively balance the two scenarios. Conversely, EASE^R, while offering numerous optimal solutions, tends to provide solutions that are concentrated and clustered. RP$^3\beta$ is effective in balancing accuracy and bias but struggles in disentangling *user-centred metrics. Finally, it is worth noting that LightGCN and MultiVAE yield inferior performance in this regard.*

Performance on all quality metrics

In response to RQ2, we can utilize the Hypervolume ($\mathcal{HV}$) measure. $\mathcal{HV}$ evaluates the performance of models from multiple objectives simultaneously, as shown in Table 5.2. By considering the cardinality and dispersion of the Pareto-optimal solutions and the dominance among the Pareto frontiers, $\mathcal{HV}$ provides us with valuable insights. The higher the volume or area under the frontier, the greater the $\mathcal{HV}$. The results show that UserKNN outperforms the other models by achieving the best or second-best values of $\mathcal{HV}$ for all datasets and scenarios. This result indicates that UserKNN generates an extensive and diversified Pareto frontier while performing well across all metrics. While EASE^R has the highest value of $\mathcal{HV}$ for the Amazon Music dataset in the user-centred scenario, it does not dominate or get dominated in the remaining cases. This result highlights the model's limited reliance on accounting for multiple metrics. LightGCN shows no distinctive trends, while MultiVAE's $\mathcal{HV}$ decreases when dealing with sparser datasets. RP$^3\beta$ confirms its capability in managing the nDCG/APLT trade-off by achieving the highest values of $\mathcal{HV}$ and visual dominance of its Pareto frontiers against the others in Figures 5.2d, 5.2e, and 5.2f.

In summary, to answer RQ2, our findings indicate that in terms of multi-objective evaluation, UserKNN is the superior model overall. However, when considering the accuracy/bias trade-off, RP$^3\beta$ emerges as a noteworthy contender.

Final observations

In evaluating recommendation systems, accuracy is typically given top priority. Thus, in our initial analysis, EASE^R emerged as the frontrunner due to its impressive accuracy. However, when subjected to our multi-objective evaluation, EASE^R was often outperformed by other models. UserKNN, on the other hand, demonstrated superior performance across diverse metrics. Surprisingly, RP$^3\beta$ ranked the lowest in terms of accuracy but proved to be particularly effective in finding a balance between nDCG and APLT (bias) performance. These findings challenge the traditional ranking of recommendation systems, paving the way for new research in model evaluation.

5.4 Summary

In our study, we utilize Quality Indicators of Pareto frontiers to conduct a multi-objective evaluation of Recommender Systems (RSs). Our experiments aim to assess RSs with three (Accuracy / Novelty / Diversity) and two (Accuracy / Bias) conflicting objectives. While EASE^R exhibits superior accuracy, our evaluation has unveiled a new ranking of the baselines. UserKNN stands out as it provides several diverse solutions which perform well in both multi-objective scenarios. Additionally, RP$^3\beta$ proved to be highly effective in the accuracy/algorithmic bias scenario. Moving forward, we plan to extend this evaluation to other baselines. Furthermore, we intend to leverage the Pareto frontiers' quality indicators to evaluate the impact of the models' hyper-parameters in a multi-objective scenario.

Chapter 6

Hyper-parameter Tuning Sensitivity in Recommender Systems with Multiple Objectives

Recommender systems (RSs) are integral to digital platforms, delivering personalized experiences that drive user engagement across various domains. Traditionally, RSs have prioritized optimizing accuracy, yielding significant business advantages. However, this singular focus overlooks critical beyond-accuracy objectives, such as fairness, diversity, novelty, and bias mitigation, which are essential for addressing multi-stakeholder interests and promoting ethical, inclusive recommendations. Multi-Objective Recommender Systems (MORSs) provide a promising framework for balancing competing objectives but pose challenges in adapting traditional accuracy-based RSs without extensive redesign.

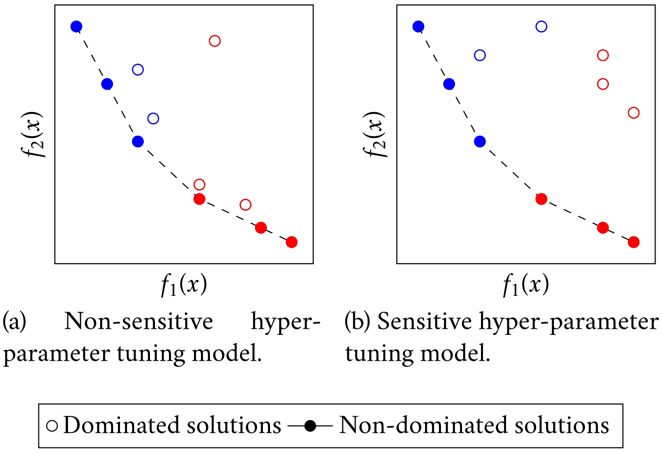
This chapter investigates the sensitivity of traditional RS models to hyper-parameter tuning in multi-objective scenarios. We propose a novel evaluation framework leveraging Pareto optimality to assess the impact of hyper-parameter configurations on balancing accuracy with beyond-accuracy goals. Through comprehensive experiments on six diverse RS models, spanning neighborhood-based, factorization-based, and graph-based methods, across 32 hyper-parameter configurations, we analyze the sensitivity of these models under two scenarios: (i) balancing accuracy, novelty, and diversity, and (ii) mitigating popularity bias alongside accuracy. We also provide insights into the role of individual hyper-parameters, offering practical guidance for minimizing tuning effort while balancing competing objectives. These contributions bridge the gap between traditional RS models and the demands of modern, multi-objective environments. [1]

[1] This chapter is based on the work "A Framework for Hyper-parameter Tuning Sensitivity Analysis in Recommender Systems Considering Multiple Objectives", to submit to the Information Processing and Management (IPM) journal.

6.1 Introduction

Recommender systems (RSs) are pivotal to modern digital platforms, delivering personalized experiences that drive user engagement across domains like e-commerce, entertainment, and travel. Traditionally, the primary focus of RSs has been optimizing accuracy, ensuring that recommendations closely align with users' preferences. This focus on relevance has yielded significant business advantages, including increased sales, enhanced user satisfaction, and improved retention rates, thereby establishing accuracy as the dominant evaluation metric in the field [93]. However, the singular emphasis on accuracy has revealed critical shortcomings since RSs are increasingly deployed in complex, multi-stakeholder environments. Emerging research highlights the need to move "beyond accuracy" by addressing objectives such as fairness, diversity, novelty, and bias mitigation. For instance, racial minority hosts on Airbnb earn less and attract fewer customers compared to white hosts[2]. In the music industry, legitimate artists often struggle for fair compensation due to manipulative practices by distributors, bots, and streaming platforms[3]. Gender biases in hiring algorithms, exemplified by Amazon's recruiting tool, further illustrate how algorithmic decisions can reinforce societal inequities[4]. These examples underscore how ostensibly neutral systems can inadvertently perpetuate inequities, revealing the urgent need to consider fairness and inclusivity in algorithmic design and evaluation. Incorporating beyond-accuracy objectives is increasingly essential for platforms that reflect diverse stakeholder interests and uphold ethical values. Properties like novelty and diversity not only enhance user satisfaction by promoting discovery but also mitigate issues of over-specialization. Similarly, fairness ensures equitable exposure for underrepresented items or providers, aligning recommendations with societal values and ethical standards. Efforts such as the NORMalize workshops [169, 194] emphasize the growing recognition of normative approaches in RS research. Additionally, regulatory frameworks such as the General Data Protection Regulation (GDPR) in Europe [63] and the California Consumer Privacy Act (CCPA) in the United States [39] further necessitate that recommendation algorithms be transparent, fair, and unbiased. To address these challenges, Multi-Objective Recommender Systems (MORSs) have emerged as a framework for balancing accuracy with other desirable objectives [90, 219]. MORSs aim to optimize multiple criteria simultaneously, ensuring that recommendations are relevant but accomplish also other objectives. While MORSs offer a promising solution for platforms designing RSs from scratch, integrating beyond-accuracy objectives into existing relevance-based RSs presents unique challenges. Established platforms must navigate the tension between meeting stakeholder values and minimizing the costs associated with redesigning, testing, and deploying entirely new solutions. A practical and scalable alternative involves leveraging hyper-parameters tuning to adjust trade-

[2] `https://shorturl.at/ePMm6`
[3] `https://shorturl.at/csQTT`
[4] `https://shorturl.at/a2AVy`



(a) Non-sensitive hyper-parameter tuning model.

(b) Sensitive hyper-parameter tuning model.

Figure 6.1. Dispersion of the solutions in the multi-objective space. Let $f_1(x)$ and $f_2(x)$ be two metrics for which the lower is the better. Each point represents a model hyper-parameter configuration set. Red and blue dots refer to two different values for a given hyper-parameter. In contrast to the model on Figure 6.1a, the model on Figure 6.1b needs precise tuning since most of the dominated configuration sets (filled dots) are far from the Pareto frontier (empty dots).

offs between accuracy and beyond-accuracy objectives. By systematically exploring hyper-parameters configurations, platforms can identify Pareto-optimal solutions that balance competing objectives without requiring substantial modifications to their underlying models. This approach raises a critical question: How can we assess the sensitivity of existing RS models to hyper-parameters tuning concerning trade-offs among multiple objectives? Specifically, we address the following research questions:

- **RQ1.** *How can we evaluate the sensitivity to the hyper-parameters tuning of existing RS models given a particular trade-off?* To address this research question, we propose to depict several hyper-parameters configurations of the same model as a point in the objective function space when considering multiple objectives. Then, the configurations ensuring the best trade-offs will lie on the Pareto frontier, while the others will be dominated solutions. As intuitively depicted in Figure 6.1, a model employing many solutions near the Pareto frontier is more constant in providing optimal or close to optimal solutions (Figure 6.1a). In contrast, the model is sensitive to hyper-parameter tuning if numerous solutions are distant from the non-dominated solutions (Figure 6.1b). Therefore, we offer two metrics for a novel evaluation framework to assess the model performance's sensitivity to hyper-parameter tuning based on the distances of solutions to the Pareto frontier in an objective function space.
- **RQ2.** *To what extent are traditional accuracy-based RSs sensitive to hyper-parameter tuning when considering multiple objectives?* Expanding on RQ1, we utilize our framework to investigate the sensitivity of traditional RSs to hyper-parameter adjustments under two scenarios: (i) a user-centric scenario that balances accuracy,

novelty, and diversity, and (ii) a scenario focused on the trade-off between accuracy and the mitigation of popularity bias in recommendations. Specifically, we train 32 distinct hyper-parameter configurations across six models from diverse families, namely, neighborhood-based, factorization-based, and graph-based methods, to evaluate the degree of hyper-parameter precision required to achieve Pareto optimal solutions.

- **RQ3.** *What is the impact of individual hyper-parameters on the tuning of the aforementioned RSs, and are there specific hyper-parameter values that consistently yield Pareto optimal (or near-optimal) solutions?* Building on the findings from RQ2, we leverage our framework to identify the hyper-parameters and their corresponding values that enable less precise tuning while still achieving Pareto optimal solutions for the studied RSs. This analysis is performed within the same experimental scenarios outlined in RQ2.

To summarize, the contributions of our work are the following:

- We introduce a novel evaluation framework that leverages Pareto optimality to assess the sensitivity of RS models to hyper-parameter configurations. By representing each configuration as a point in the objective function space, our framework quantifies the distances between each solution and the Pareto frontier, enabling a robust analysis of model performance consistency across trade-offs.
- Using our framework, we systematically evaluate six diverse RS models, spanning neighborhood-based, factorization-based, and graph-based methods across 32 distinct hyper-parameter configurations. This analysis reveals how traditional accuracy-focused RSs can accommodate beyond-accuracy objectives, such as novelty, diversity, and bias mitigation, without requiring extensive tuning.
- We provide a detailed investigation into the role of individual hyper-parameters, identifying specific configurations that consistently achieve Pareto optimal or near-optimal solutions. These insights offer practical guidance for practitioners seeking to balance accuracy with other objectives while minimizing the need for extensive hyper-parameter optimization.

6.2 Related Work

The development and evaluation of Recommender Systems (RSs) have long been central to academic and industrial research. However, recent works highlight the need for a more analytical approach to research in RSs. Among other points, Jannach et al. [91] emphasize the importance of understanding how stable algorithms are in their performance across various metrics when hyper-parameter values are slightly altered. Such stability is vital for reproducibility and real-world deployment, where frequent retraining or model updates are often required. Few works concerning offline evaluation experiments focus on the sensitivity to hyper-parameter tuning of the recommendation baselines.

Mostly, these works are reproducibility studies that aim to reveal methodological flaws that compromise the validity of experimental results [21, 50]. For instance, some works have reported that the hyper-parameters of baselines are not tuned to the same extent as those of the new proposed recommendation algorithm, or even at all [48, 162]. In addition, several experiments improperly tuned hyper-parameters on test datasets rather than validation datasets, violating standard practices and potentially inflating performance metrics. In their survey, Sun et al. [175] observed that more than 33% of the offline evaluations tuned hyper-parameters on the test dataset. At the same time, the remaining 67% papers did not mention any information about hyper-parameters tuning. Similarly, Dacrema et al. [50] reported that several offline studies they attempted to reproduce evaluated the model's performance on the test dataset after each training epoch, and reported the best metric value. Unlike other elements of experimental setups, such as datasets or data splits, the ranges of hyper-parameters explored and the methods used for tuning are rarely disclosed [162, 223]. In response to these issues, some works have proposed guidelines for fair comparison in hyper-parameter tuning. For example, Shehzad et al. [162] suggest authors to report the hyper-parameter ranges explored and the tuning method used. Fang et al. [64] explore multiple hyper-parameter search algorithms to ensure robust evaluation methodologies.

In contrast, other works have explicitly focused on the hyper-parameter optimization process in RSs. For instance, Anelli et al. [20] explore the discriminative power of accuracy and novelty metrics in hyper-parameter tuning, analyzing which specific hyper-parameters most significantly impact the accuracy of the BPR-MF algorithm. Similarly, Matuszyk et al. [125] conduct a comparative analysis of various optimization strategies for hyper-parameter tuning of RSs.

While these studies provide valuable insights into hyper-parameters impact on accuracy, they are primarily limited to single-objective perspectives, focusing exclusively on accuracy metrics. These works highlight suboptimal or incorrect hyper-parameter tuning procedures [220]. Still, they do not offer a comprehensive framework for assessing the sensitivity of RS performance to hyper-parameter changes across multiple objectives. This narrow focus on accuracy overlooks the broader implications of hyper-parameter tuning in multi-objective scenarios, a critical consideration for real-world RSs deployed on online platforms. Such systems must balance competing objectives, including user satisfaction, fairness, and diversity while mitigating potential harms.

Some works have begun addressing this gap by integrating multi-objective perspectives into hyper-parameter tuning. For example, Quadrana et al. [145] propose a multi-objective optimization framework for hyper-parameter tuning in the next-song recommendation task, demonstrating its effectiveness in balancing competing objectives. Additionally, Moscati et al. [129] investigate the interplay between accuracy and beyond-accuracy metrics by identifying Pareto-optimal hyper-parameter configurations for a recommendation baseline.

To the best of our knowledge, no previous studies have proposed an analytical framework to evaluate the sensitivity of RSs to hyper-parameter tuning in a multi-objective context. This work introduces the first comprehensive framework to assess the extent to which RS performance is influenced by hyper-parameter variations, considering the relevance of recommendations and beyond-accuracy objectives simultaneously.

6.3 The framework (RQ1)

This section provides a formal definition of the evaluation framework proposed in this work. This framework aims to quantitatively measure the sensitivity to the hyper-parameter tuning of an RS model in a multi-objective evaluation scenario. Hence, this section theoretically answers to RQ1.

6.3.1 Protocol

Let us train a set $\mathcal{K}$ of different hyper-parameter configurations of the same recommendation model whose performance is represented by m metrics $\{n_1, \dots, n_m\}$. Then, a point $\boldsymbol{k} \in \mathcal{K}$ is defined as $\boldsymbol{k} = \{\phi(n_1), \dots, \phi(n_m)\}$, where $\phi(\cdot)$ is the min-max normalization function.[5] As illustrated in Figure 6.1, each configuration may thus be represented as a point in a m-dimensional objective function space.

Let $\mathcal{K} = \{\boldsymbol{k}_1, \boldsymbol{k}_2, \dots, \boldsymbol{k}_{|\mathcal{K}|}\} \in \mathbb{R}^{|\mathcal{K}| \times m}$ be the set of $|\mathcal{K}|$ points in the objective function space, with $\boldsymbol{k}_t \in \mathbb{R}^m$. Among the set $\mathcal{K}$, let us suppose to have the set $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_{|\mathcal{P}|}\}$ of $|\mathcal{P}|> 1$ sequentially ordered Pareto optimal points with $\mathcal{P} \subseteq \mathcal{K}$ and $\boldsymbol{p}_t \in \mathbb{R}^m$. The Pareto frontier can be shaped as a polyline consisting of $|\mathcal{P}|-1$ segments $\overline{\boldsymbol{p}_i\boldsymbol{p}_j}$, each having as vertices a pair of Pareto optimal points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, with $i \in \{1, \dots, |\mathcal{P}|-1\}$ and $j = i + 1$. Then, we introduce the formulation for the distance between a point $\boldsymbol{k}_t$ and the Pareto frontier $\mathcal{P}$ as:

$$\Delta\left(\mathcal{P}, \boldsymbol{k}_t\right) = \min(\{\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right)\}), \tag{6.1}$$

where $\{\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right)\}$ is the set of the distances $\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right)$ computed between the point $\boldsymbol{k}_t \in \mathcal{K}$ and each segment $\overline{\boldsymbol{p}_i\boldsymbol{p}_j} \in \mathcal{P}$ such that $i \in \{1, \dots, |\mathcal{P}|-1\}, j = i + 1$. The computation of $\Delta\left(\mathcal{P}, \boldsymbol{k}_t\right)$ is then performed for each point $\boldsymbol{k}_t \in \mathcal{K}$. In this way, we obtain the set of distances $\{\Delta\left(\mathcal{P}, \boldsymbol{k}_t\right)\}$ of each point $\boldsymbol{k}_t$ from the Pareto frontier $\mathcal{P}$.

The scenario described above refers to the case in which more solutions in $\mathcal{K}$ are Pareto optimal, i.e., $|\mathcal{P}|> 1$. However, some recommendation models may provide a single Pareto optimal solution, thus formally resulting in a Pareto frontier $\mathcal{P} = \{\boldsymbol{p}_1\}$ with cardinality $|\mathcal{P}|= 1$. In such cases, it is not feasible to shape the Pareto frontier as a polyline consisting of $|\mathcal{P}|-1$ segments $\overline{\boldsymbol{p}_i\boldsymbol{p}_j}$. Consequently, the computation of the distance $\Delta\left(\mathcal{P}, \boldsymbol{k}_t\right)$ between the Pareto frontier $\mathcal{P} = \{\boldsymbol{p}_1\}$ and a point $\boldsymbol{k}_t$ reduces to the Euclidean distance between $\boldsymbol{p}_1$ and $\boldsymbol{k}_t$. Formally:

$$\Delta\left(\mathcal{P}, \boldsymbol{k}_t\right) = ||\boldsymbol{p}_1 - \boldsymbol{k}_t||, \tag{6.2}$$

[5] We provide an empirical justification for the normalization of the metrics values in Section 6.3.4.

where $||\cdot||$ represents the Euclidean norm.

In the following, we delve into how calculating each distance $\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right)$.

Distance computation

Now, we focus on how to compute the distance $\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right)$ from a segment $\overline{\boldsymbol{p}_i\boldsymbol{p}_j}$ belonging to the Pareto frontier to a model configuration represented by $\boldsymbol{k}_t \in \mathcal{K}$.

Conceptually, computing the distance between the segment $\overline{\boldsymbol{p}_i\boldsymbol{p}_j}$ and the point $\boldsymbol{k}_t$ is equivalent to calculating the Euclidean distance between the point $\boldsymbol{k}_t$ and its closest point lying on the segment $\overline{\boldsymbol{p}_i\boldsymbol{p}_j}$. Then, the first step is finding the closest point $\boldsymbol{g}_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} \in \mathbb{R}^m$ belonging to $\overline{\boldsymbol{p}_i\boldsymbol{p}_j}$ from $\boldsymbol{k}_t$.

We define the segment $\overline{\boldsymbol{p}_i\boldsymbol{p}_j} = \boldsymbol{p}_i - \boldsymbol{p}_j$ and the segment $\overline{\boldsymbol{p}_i\boldsymbol{k}_t} = \boldsymbol{p}_i - \boldsymbol{k}_t$. To find where the point $\boldsymbol{k}_t$ projects onto the infinite line defined by $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, we compute a scalar value $w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t}$, which is given by:

$$w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} = \frac{\overline{\boldsymbol{p}_i\boldsymbol{k}_t} \cdot \overline{\boldsymbol{p}_i\boldsymbol{p}_j}}{\overline{\boldsymbol{p}_i\boldsymbol{p}_j} \cdot \overline{\boldsymbol{p}_i\boldsymbol{p}_j}},$$

where $w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t}$ tells us how far along the segment $\overline{\boldsymbol{p}_i\boldsymbol{p}_j}$ the projection of $\boldsymbol{k}_t$ lies. Specifically: (i) if $0 < w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} < 1$, the projection lies between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ on the segment; (ii) if $w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} = 0$, the projection lies exactly at $\boldsymbol{p}_i$; (iii) if $w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} = 1$, the projection lies exactly at $\boldsymbol{p}_j$; (iv) if $w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} < 0$, the projection lies before $\boldsymbol{p}_i$; (v) if $w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} > 1$, the projection lies beyond $\boldsymbol{p}_j$. By clamping $w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t}$ to the range [0, 1], we ensure the closest point lies on the segment. Then, we can calculate the closest point $\boldsymbol{g}_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t}$ as:

$$\boldsymbol{g}_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} = \boldsymbol{p}_i + w_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t} \times \overline{\boldsymbol{p}_i\boldsymbol{p}_j}.$$

Finally, we can compute the distance $\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right)$ as the Euclidean distance between the point $\boldsymbol{k}_t$ and $\boldsymbol{g}_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t}$. Formally:

$$\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right) = ||\boldsymbol{k}_t - \boldsymbol{g}_{\boldsymbol{p}_i\boldsymbol{p}_j\boldsymbol{k}_t}||, \tag{6.3}$$

where $||\cdot||$ represents the Euclidean norm.

By computing $\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right) \; \forall \, \overline{\boldsymbol{p}_i\boldsymbol{p}_j} \in \mathcal{P}$ such that $i \in \{1, \ldots, P-1\}, j = i+1$, we obtain the set $\{\Delta\left(\overline{\boldsymbol{p}_i\boldsymbol{p}_j}, \boldsymbol{k}_t\right)\}$. The minimum value of this set is the distance $\Delta\left(\mathcal{P}, \boldsymbol{k}_t\right)$ of the point $\boldsymbol{k}_t$ from the Pareto frontier $\mathcal{P}$ as formalized in Eq. (6.1).

6.3.2 Metrics

Section 6.3.1 describes the protocol to follow to apply the multi-objective evaluation approach proposed in this work. The framework aims to provide a methodology to understand to what extent a recommendation model needs accurate fine-tuning to reach Pareto optimal performance given various assessed perspectives. For this reason, we now define two metrics that rely on the distances between each configuration model and its Pareto frontier. The core idea is to exploit such distances to quantitatively measure the probability that, by varying the hyper-parameters values of a model, such model offers Pareto optimal solutions or solutions close to them. The higher this probability, the less sensitive to hyper-parameter tuning is the model from a multi-dimensional perspective.

A straightforward metric is the mean of the distances computed as in section 6.3.1. Given the set of distances $\{\Delta(\mathcal{P}, \boldsymbol{k}_t)\}$ of each point $\boldsymbol{k}_t \in \mathcal{K}$ from the Pareto frontier $\mathcal{P}$, we define the mean μ of the distances in $\{\Delta(\mathcal{P}, \boldsymbol{k}_t)\}$ as follows:

$$\mu = \frac{\sum_{\boldsymbol{k}_t \in \mathcal{K}} \Delta(\mathcal{P}, \boldsymbol{k}_t)}{|\mathcal{K}|}. \tag{6.4}$$

The metric μ indicates how distant the model configurations are from the Pareto optimal configurations on average. Then, the lower the value of μ, the shorter is the average distance between the dominated model configurations and the model configurations that ensure a Pareto optimal trade-off of the m metrics. This metric captures a scenario in which many configurations of a model are Pareto optimal since the higher the number of solutions lying on the Pareto frontier, the lower the value of μ. From the hyper-parameter tuning point of view, it is evident that a lower value of μ suggests that the considered model needs a less precise hyper-parameter tuning to provide a Pareto optimal (or close to) solution as it offers many configurations on the Pareto frontier or near to it. Conversely, a higher value of μ reveals that the model needs several hyper-parameter adjustments to gather a Pareto optimal configuration (or close to it).

After computing the mean μ of the distances in $\{\Delta(\mathcal{P}, \boldsymbol{k}_t)\}$ of each point $\boldsymbol{k}_t \in \mathcal{K}$ from the Pareto frontier $\mathcal{P}$, we can calculate the standard deviation of these distances. Given the set of distances $\{\Delta(\mathcal{P}, \boldsymbol{k}_t)\}$ of each point $\boldsymbol{k}_t \in \mathcal{K}$ from the Pareto frontier $\mathcal{P}$ and their mean μ, we define the standard deviation σ of the distances in $\{\Delta(\mathcal{P}, \boldsymbol{k}_t)\}$ as follows:

$$\sigma = \sqrt{\frac{\sum_{\boldsymbol{k}_t \in \mathcal{K}} (\Delta(\mathcal{P}, \boldsymbol{k}_t) - \mu)^2}{|\mathcal{K}|-1}}. \tag{6.5}$$

The metric σ indicates the dispersion of the distance values of the model configurations from the Pareto frontier. Then, the lower the value of σ, the less dispersed are the model configurations in terms of distance from the Pareto frontier. In other words, a small value of σ catches a situation where the model configurations are equally distant from the Pareto frontier. Then, from the hyper-parameter tuning perspective, a lower value of σ implies that modifications in the hyper-parameter values do not strongly impact the achievement of a solution closer to the Pareto frontier. Conversely, a higher value of σ means that a precise adjustment of the hyper-parameters can help reach a solution closer to the Pareto frontier than others. It is worth mentioning that σ computed as in Eq. (6.5) does not indicate to what extent the model can provide different trade-off solutions given the examined objectives. Indeed, this dispersion metric is computed on the distances of several solutions from the Pareto frontier and not on the metric performance.

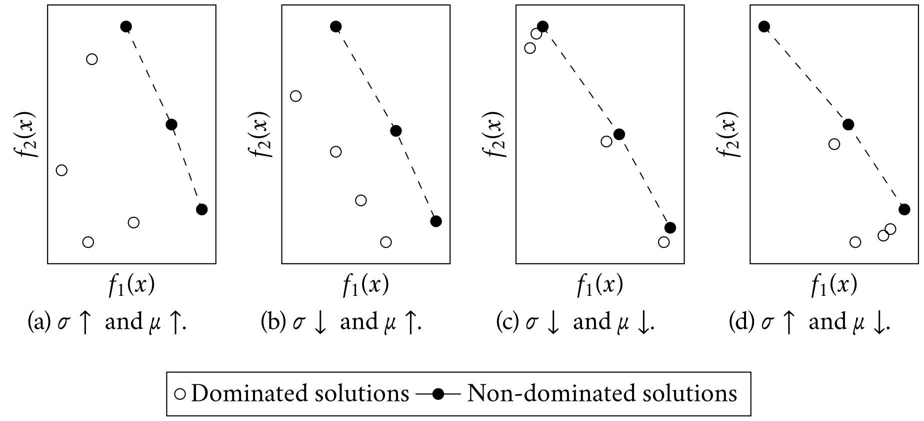


Figure 6.2. Different cases for joint interpretation of σ and μ in terms of sensitivity to hyper-parameters tuninf of recommendation models in a multi-objective scenario. $f_1(x)$ and $f_2(x)$ be two metrics for which the higher is the better.

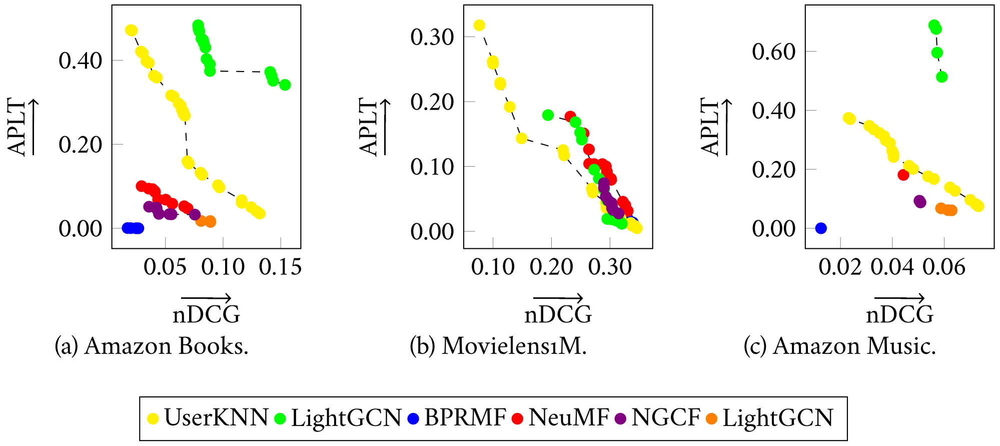
6.3.3 Interpretation of the metrics

In the previous section, we have introduced the metrics μ and σ to assess the sensitivity to hyper-parameters tuning of Recommender Systems (RSs) in a multi-objective scenario. To summarize, these metrics capture different situations in the objective function space:

- μ measures to what extent the solutions in the objective function space are close to or lie on the Pareto frontier;
- σ measures to what extent the solutions in the objective function space are dispersed in terms of distance to the Pareto frontier.

Since these metrics catch different properties, their simultaneous assessment provides a comprehensive lens for interpreting RS sensitivity to hyper-parameter tuning through the Pareto frontiers. We can provide a comprehensive simultaneous interpretation of μ and σ by identifying four distinct illustrative cases:

1. **High Mean and Standard Deviation**: the model provides solutions with high and varied distances from the Pareto frontier, indicating that the hyper-parameters tuning can lead to highly different outcomes in a multi-faceted assessment. Hence, the model is significantly susceptible to hyper-parameter tuning (Figure 6.2a).
2. **High Mean and Low Standard Deviation**: the model provides solutions uniformly distanced from the Pareto frontier but with consistent displacement. Hence, the model is moderately sensitive to hyper-parameter adjustments (Figure 6.2b).
3. **Low Mean and Standard Deviation**: the model provides solutions that are concentrated near or on the Pareto frontier, indicating that the model achieves robust performance across a range of hyper-parameter settings. Hence, the model shows insensitivity to hyper-parameter tuning (Figure 6.2c).



(a) Amazon Books.

(b) Movielens1M.

(c) Amazon Music.

Figure 6.3. Pareto Frontiers of six recommendation baselines evaluated with nDCG and APLT. The Pareto frontiers of better performing models dominate the others.

4. **Low Mean and High Standard Deviation**: the model provides many solutions near or on the Pareto frontier. However, a few outlier solutions are farther from the frontier. Hence, the model exhibits minimal sensitivity to hyper-parameter tuning (Figure 6.2d).

The fourth scenario is particularly intriguing. Delving deeper into this case can uncover valuable insights, such as identifying the specific hyper-parameter configurations responsible for the outlier solutions that deviate significantly from the frontier. Therefore, a hyper-parameter-level fine-grained analysis of μ and σ could let us discover the hyper-parameters (and their values) that make the recommender system less sensitive to the hyper-parameter tuning. Such an analysis could inform the refinement of hyper-parameter search spaces to minimize suboptimal outcomes. Ultimately, this taxonomy provides a structured framework for understanding the interplay between model performance and hyper-parameter tuning in a multi-objective scenario, offering a basis for targeted optimization and deeper exploration of model behavior.

6.3.4 Notes on the metrics values normalization

When adhering to the evaluation protocol outlined in Section 6.3.1, a model's hyper-parameter configuration is characterized by a suite of metric values which are normalized using min-max normalization ($\phi(\cdot)$). This normalization serves two critical purposes.

First, it ensures that metrics with different scales are treated equitably. Without normalization, a metric with a larger scale could disproportionately influence the computation of the distance $\Delta(\mathcal{P}, \boldsymbol{k}_t)$ in Eq. (6.1), leading to biased results.

Second, normalization facilitates a fair comparison among models by mitigating performance biases and enabling a consistent evaluation of sensitivity to hyper-parameter tuning in multi-objective scenarios. In this regard, Figure 6.3 depicts the

Pareto frontiers of six recommendation baselines evaluated using nDCG and APLT (see Section 6.4 for experimental details). A poorly performing model is likely to have its Pareto frontier dominated by those of other models, resulting in smaller distances between the dominated points and the model's Pareto frontier. However, smaller distances do not indicate reduced sensitivity to hyper-parameter tuning. Instead, they reflect the model's inferior overall performance. Normalization of metric values is indispensable to eliminate this performance bias and accurately assess hyper-parameter sensitivity.

6.4 Experiments

This section provides an overview of the experimental settings used to validate the proposed framework for measuring the sensitivity of recommender systems to hyper-parameters tuning. We begin by outlining the evaluation methodology and the multi-objective evaluation scenarios. Next, we describe the recommendation baselines and their associated hyper-parameter search spaces. Following this, we present the datasets utilized in the experiments. Lastly, we detail the evaluation protocol to ensure reproducibility and transparency.

6.4.1 Experimental Setup

Evaluation Scenarios and Methodology

This experimental setup aims to evaluate the proposed framework's ability to measure the sensitivity of recommendation algorithms to hyper-parameter tuning in a multi-objective scenario (Section 6.3.1). To achieve this, we train $|\mathcal{K}| = 32$ distinct hyper-parameter configurations for each recommendation baseline outlined in Table 6.1, resulting in a total of $32 \times 6 = 192$ trained models. We then employ a suite of metrics to evaluate the recommendation lists generated by each model, capturing performance from multiple perspectives. Then, the experiments are structured into two scenarios: (i) a user-centric scenario that focuses on metrics that evaluate the quality of recommendations from the end user's perspective (i.e., accuracy, diversity, and novelty); (ii) a popularity bias scenario that deals with the issue of popularity bias in recommendations, assessing the trade-off between the relevance of recommendations and the ability of the models to suggest less popular items.

For each scenario, we define m metrics to evaluate model performance. For a given recommendation baseline, the configurations of the trained models form solutions in an m-dimensional objective function space. This results in 32 points in the objective function space for each baseline. Among these points, the Pareto optimal configurations, which constitute the Pareto frontier $\mathcal{P}$, are identified, while the remaining points represent dominated solutions.

The following outlines the scenarios employed to utilize the proposed framework. Contextually, we also define the metrics computed that compose the m-dimensional objective function spaces.

User-centered scenario. The evaluation of Recommender Systems (RSs) has traditionally focused on accuracy, often measured through metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and ranking-based metrics such as Normalized Discounted Cumulative Gain (nDCG). These metrics assess how well a system predicts or ranks items that align with user preferences, ensuring that relevant items are prioritized. However, while accuracy remains a foundational goal, this narrow focus overlooks critical aspects of recommendation quality, particularly the ability to foster discovery and engagement through novelty and diversity [126, 181]. These elements are necessary to prevent accurate recommendations from becoming predictable, uninspiring, and ultimately less valuable to the user [181].

Novelty focuses on providing recommendations that introduce unfamiliar and previously unseen items [126, 185]. For instance, in a movie recommendation context, suggesting niche films or those outside mainstream popularity enhances novelty, encouraging users to explore beyond their typical preferences.

Unlike novelty, which evaluates the relationship between a recommended item and the user's prior interactions, diversity measures the relationships among the recommended items [8, 185]. For example, a recommendation list that includes movies spanning multiple genres, themes, or styles exhibits high diversity, whereas one focused solely on a single genre might seem monotonous. Diversity operates on two levels: individual diversity, which captures the variety within a single user's recommendation list, and aggregate diversity, which evaluates the breadth of unique items recommended across all users in the system. High aggregate diversity is crucial for mitigating concentration on a small subset of items.

Balancing accuracy, novelty, and diversity presents inherent trade-offs [191]. Highly accurate systems tend to prioritize popular or predictable items, often at the expense of novelty and diversity. Conversely, enhancing novelty by recommending obscure or long-tail items can reduce perceived relevance. Similarly, improving diversity by including a wider range of items might introduce irrelevant or less desirable recommendations, impacting user satisfaction. These trade-offs underscore the importance of adopting multi-objective evaluation strategies that account for the complex interplay between these dimensions.

Given this context, we employ our multi-objective evaluation framework in a recommendation scenario that simultaneously focuses on recommendations' accuracy, novelty, and diversity. Therefore, we employ $m = 3$ metrics to evaluate these recommendation objectives, obtaining a three-dimensional objective function space:

- Normalized Discounted Cumulative Gain (nDCG) is a ranking-based evaluation metric widely used to measure the accuracy of recommendations. It accounts for the recommended items' relevance and position in the recommendation list (see Section 2.3.3).
- Gini index is a measure of aggregate diversity used to measure the distributional inequality, i.e., how unequally different items are chosen by users when a particular RS is used [42] (see Section 2.4.1).

- Expected Popularity Complement (EPC) measures the novelty of recommendations [42]. It measures the expected number of relevant items belonging to the long-tail (see Section 2.4.1).

Popularity bias scenario. Popularity bias is a well-documented issue in recommender systems, where algorithms tend to favor previously widely popular items, often at the expense of less known or niche content [24]. This bias occurs because many recommendation algorithms rely on user interactions with items to drive recommendations. Items that are more frequently interacted with tend to be ranked higher, resulting in recommendations that are overly dominated by popular content. While these recommendations appear accurate for users who have aligned preferences with the mainstream, they fail to introduce novelty or encourage users to explore new and diverse options, limiting the overall usefulness of the system. One significant consequence of popularity bias is the unequal exposure of items, which raises important concerns about provider fairness [26, 164]. By their widespread engagement, popular items dominate the recommendation landscape, making it difficult for less popular or emerging items to gain visibility. This situation creates a feedback loop where already-popular items are continually recommended while newer or niche content, which may be equally relevant to the user, remains underrepresented. This imbalance limits the diversity of items recommended to users and perpetuates content dominance from well-established providers or creators. For instance, in an online marketplace or streaming service, popular items from major brands or established creators receive far more exposure than smaller, independent creators, reducing opportunities for them to reach new audiences.

Achieving an optimal balance remains a challenge. Prioritizing accuracy based on historical interactions can exacerbate popularity bias, reinforcing the dominance of popular content. Conversely, emphasizing niche items can reduce relevance by introducing items that may not resonate with the user.

We consider $m = 2$ metrics to evaluate the sensitivity to hyper-parameter tuning of recommendation baselines in a scenario, including accuracy and popularity bias of recommendations. We evaluate the relevance through the nDCG (see Eq. (**??**)). Furthermore, we consider two metrics that deal with how much the popularity of items in the catalog influences those suggested to users. In particular, we consider the average percentage of items in the long-tail (*APLT*) [3], which measures in what proportion unpopular items (i.e., niche) are recommended in users' recommendation lists (see Section 2.4.2).

Baselines and Hyper-parameter tuning

We train six recommendation models from three different families of algorithms: (i) **Neighborhood-based** models; (ii) **Matrix factorization** models; (iii) **Graph-based** models.

Regarding neighborhood-based models, we select the following baselines:

- **UserKNN** [151]: a collaborative filtering algorithm that computes the similarity between users based on their historical preferences. It predicts a user's preference by aggregating the ratings of the most similar users;

- **ItemKNN** [156]: a collaborative filtering algorithm that measures item similarity based on user interaction patterns. It predicts a user's preference by analyzing similar items with which the user has previously interacted.

The matrix factorization trained models are:

- **BPRMF** [149]: Bayesian Personalized Ranking Matrix Factorization optimizes the recommendation process by directly modeling pairwise comparisons. It focuses on ranking items such that relevant ones are placed above irrelevant ones for a specific user;
- **NeuMF** [85]: Neural Matrix Factorization combines traditional matrix factorization with deep learning. It uses a multi-layer perceptron (MLP) to model complex and non-linear interactions between user and item embeddings.

Finally, the graph-based models considered are:

- **NGCF** [200]: Neural Graph Collaborative Filtering extends collaborative filtering by propagating user and item embeddings through a user-item bipartite graph. It captures higher-order connectivity and feature interactions in the graph structure;
- **LightGCN** [84]: Light Graph Convolutional Network simplifies graph-based recommendation models by removing unnecessary operations like feature transformation and activation. It aggregates embeddings over multiple graph layers to model user-item interactions effectively.

We choose these algorithms to generalize the findings of the obtained results and find common patterns among different algorithms belonging to the same family. We explore 32 distinct hyper-parameter configurations of each model through a grid search. Table 6.1 overviews the hyper-parameters tuned for each baseline.

Datasets

We adopt three public datasets to train the baseline and use the proposed evaluation framework, i.e., Amazon Books [134], Movielens1M [81], and Amazon Music [13, 134]. We choose these datasets to ensure the generalization of the results by varying the data domains and characteristics. Amazon Books is a dataset from the book domain. It contains 771099 interactions among 30839 users and 30548 items (0.99 sparsity). Movielens1M is a movie dataset with 6040 users, 3706 items, and 1000209 interactions, resulting in a lower sparsity than the other datasets. Finally, Amazon Music is a dataset of the music domain containing 10027 interacted items by 14354 users, totalizing 145523 interactions (0.99 sparsity).

Evaluation Protocol

The datasets are processed in an implicit feedback setting and split using a 70-10-20 hold-out strategy. During training, model performance is evaluated on the validation

Table 6.1. Overview of the hyper-parameters tuned for the recommendation baselines adopted in this study.

Family	Algorithm	Hyper-parameter	Values
Neighborhood-based	UserKNN	Neighbors Similarity	$\{10, 20, 30, 50, 100, 150, 200, 250\}$ {cosine, jaccard, euclidean, pearson}
	ItemKNN	Neighbors Similarity	$\{10, 20, 30, 50, 100, 150, 200, 250\}$ {cosine, jaccard, euclidean, pearson}
Factorization-based	BPRMF	Factors Learning Rate Regularization	$\{8, 16, 32, 64\}$ $\{0.001, 0.0005, 0.005, 0.0001\}$ $\{0.1, 0.05\}$
	NeuMF	Factors Learning Rate Negative Samples	$\{8, 16, 32, 64\}$ $\{0.001, 0.0005, 0.005, 0.0001\}$ $\{4, 8\}$
Graph-based	NGCF	Factors Layers Learning Rate	$\{8, 16, 32, 64\}$ $\{1, 2, 3, 4\}$ $\{0.001, 0.0005\}$
	LightGCN	Factors Layers Learning Rate	$\{8, 16, 32, 64\}$ $\{1, 2, 3, 4\}$ $\{0.001, 0.0005\}$

set every 20 epochs. Following standard evaluation practices in the recommender systems community [20], we select the best iteration based on the nDCG@10 score on the validation set. The model from this iteration is then used to report performance on the test set. We employ early stopping with a patience value of 5 epochs to prevent overfitting.

6.5 Results and Discussion

This section delves into the outcomes of our experimental evaluation, providing a detailed analysis of the results obtained, and directly addressing the research questions posed in the introduction. The plots of the objective function spaces gathered for each combination of model family, dataset, and experimental scenario are reported in Appendix A.

6.5.1 *Sensitivity to Hyper-parameter Tuning in Multi-Objective Scenarios (RQ2)*

In this section, we aim to understand to what extent traditional accuracy-based RSs are sensitive to hyper-parameter tuning when considering multiple objectives. Table 6.2 reports the value of μ and σ computed apply our proposed framework outlined in Section 6.3 for the six traditional recommendation baselines consid-

Table 6.2. Mean (i.e., μ) and standard deviation (i.e., σ) of the distances among each hyper-parameter configuration and the Pareto frontier for each model. The results are categorized into the studied scenarios. For each scenario, bold and underline stand for best and second-to-best values, respectively.

Metrics	**Models**	**Amazon Books**		**Amazon Music**		**Movielens1M**	
		μ	σ	μ	σ	μ	σ
Accuracy – Bias	UserKNN	**0.0117**	**0.0384**	**0.0157**	**0.0292**	**0.0232**	**0.0484**
	ItemKNN	0.1221	0.1910	0.6187	0.4509	0.0633	0.0825
	NGCF	0.1398	0.1109	0.6559	0.4038	0.1509	0.1771
	LightGCN	0.6807	0.4511	0.6718	0.4185	0.7008	0.3918
	BPRMF	0.6711	0.3216	0.7794	0.1789	0.7917	0.3748
	NeuMF	0.1711	0.2172	0.7340	0.2957	0.0875	0.1049
Accuracy – Novelty – Diversity	UserKNN	0.3918	0.4011	**0.4209**	0.4092	**0.0723**	**0.1038**
	ItemKNN	0.7633	0.4856	1.0015	0.6552	0.0809	0.1311
	NGCF	0.4281	0.4135	0.8201	0.5270	0.2819	0.3171
	LightGCN	0.8316	0.5604	0.7509	0.5135	0.8424	0.5195
	BPRMF	1.0089	0.4543	0.9587	**0.2599**	0.9314	0.5469
	NeuMF	**0.2581**	**0.3045**	0.8261	0.4299	0.1091	0.1292

ered. The values are reported for each dataset and for both the accuracy/bias and accuracy/novelty/diversity scenarios.

Accuracy/Bias Scenario

In the accuracy/bias scenario, UserKNN emerges as the model least sensitive to hyper-parameter tuning. As shown in Table 6.2, this model consistently achieves the lowest values of μ and σ across all three datasets. This outcome aligns with the interpretation outlined in Section 6.3.3, where low mean and standard deviation values indicate that the model does not require precise tuning to reach Pareto optimal solutions. Figures A.2a and A.2b further corroborate this finding, illustrating that the majority of UserKNN's configurations produce Pareto optimal solutions across the Amazon Books dataset, offering a diverse range of trade-offs. Similar patterns are observed for the Amazon Music and Movielens1M datasets. ItemKNN also demonstrates relatively low sensitivity to hyper-parameter tuning in the accuracy/bias scenario, ranking as the second-best performer in terms of μ. However, its higher σ values, particularly for the Amazon Books and Amazon Music datasets, suggest that some configurations fail to approach the Pareto frontier. These findings correspond to the fourth case described in Section 6.3.3, where variability in performance arises. For example, Figure A.2c shows that dominated configurations are primarily associated with solutions using "Jaccard" as the similarity metric. This observation highlights the potential of our framework to investigate the impact of individual hyper-parameter choices systematically (Section 6.5.2). In contrast, NeuMF and NGCF exhibit fluctuating behavior across datasets. While these models perform relatively well on the Amazon Books and Movielens1M datasets, their performance deteriorates on the Amazon Music dataset (Figures A.8c and A.7d). Conversely, BPRMF and LightGCN generally rank among the worst-performing models in this analysis, frequently displaying high μ and σ values. However, the Amazon Music dataset provides an intriguing exception, where BPRMF demonstrates low σ but high μ, indicating consistent yet suboptimal solutions that remain distant from the Pareto frontier. This scenario aligns with the second case described in Section 6.3.3, where low standard deviation does not imply insensitivity to hyper-parameter tuning. Supporting this, Figures A.8a and A.8b reveal that many BPRMF configurations with suboptimal performance share near-zero or zero values for APLT, further confirming their limitations. Finally, our analysis reveals that models exhibiting lower sensitivity to hyper-parameter tuning tend to demonstrate a negative correlation between nDCG and APLT (i,e., UserKNN and ItemKNN). These models typically produce a greater number of Pareto optimal solutions, significantly influencing both μ and σ. However, certain models are capable of achieving specific configurations that excel simultaneously in both nDCG and APLT. For instance, as illustrated in Figure A.8c, two NeuMF configurations achieve superior performance on both objectives, while the remaining dominated solutions distribute across various trade-offs.

This observation highlights a critical dilemma in model selection: whether to prioritize a model capable of achieving high simultaneous multi-objective performance at the cost of requiring precise hyper-parameter tuning, or to opt for a model that, while easier to train, offers a broader spectrum of trade-offs. This trade-off underscores the importance of balancing practical considerations, such as computational cost and tuning complexity, with the ability to meet specific application requirements effectively.

Accuracy/Novelty/Diversity Scenario

In the accuracy/novelty/diversity scenario, achieving Pareto optimal solutions proves to be more challenging due to the reduced cardinality of the Pareto frontiers, as observed in Appendix A. This lower cardinality indicates fewer configurations capable of simultaneously balancing accuracy, novelty, and diversity. Despite this increased complexity, some patterns identified in the accuracy/bias scenario persist.

UserKNN continues to demonstrate robustness as a recommendation model that requires less precise hyper-parameter tuning when balancing multiple objectives. In this scenario, it consistently achieves the best or second-best values for μ and σ, further validating its stability and adaptability across diverse trade-offs.

NeuMF and NGCF display similar behaviors to those observed in the accuracy/bias scenario. They require more precise hyper-parameter tuning solely for the Amazon Music dataset, which consistently poses challenges in achieving optimal configurations. In contrast, LightGCN and BPRMF remain the weakest performers in this analysis, frequently exhibiting high μ and σ values, indicating a pronounced sensitivity to hyper-parameter tuning and a tendency to provide suboptimal solutions.

ItemKNN maintains a consistent performance across most datasets, except for the Amazon Books dataset, where its effectiveness slightly diminishes.

6.5.2 Impact of Hyper-parameters on the Tuning of Recommender Systems (RQ3)

In this section, we delve into the impact of individual hyper-parameter values on model sensitivity to hyper-parameter tuning. While the previous section focused on analyzing model sensitivity to hyper-parameter tuning considering all the 32 trained configurations, our goal here is to identify specific hyper-parameters and their respective values that are most likely to result in a Pareto optimal or near-optimal configuration. To conduct this investigation, we isolate the dominated configurations in the objective function space that share a specific hyper-parameter value alongside the entire Pareto frontier. Subsequently, we calculate the metrics σ and μ based on the distances between all the configurations (having that hyper-parameter and value) and the Pareto frontier. This analysis enables us to assess how much a particular hyper-parameter value contributes to achieving Pareto optimal trade-offs concerning multiple objectives. Tables 6.3, 6.4, and 6.5 report the results of this analysis.

Factorization Models

Table 6.3 illustrates the impact of the factors and learning rate hyper-parameters for the factorization models, namely BPRMF and NeuMF. For NeuMF, the hyper-parameter setting *factors=64* emerges as the most influential, consistently producing solutions closer to the Pareto frontier across all datasets and scenarios. This observation is evidenced by the lowest (or near-lowest) values of μ. Except for the Amazon Music dataset, this configuration also minimizes σ in both scenarios, indicating that most solutions are tightly clustered near the frontier. Figures A.8c and A.17c depict these situations for both scenarios using the Amazon Music dataset. Thus, when *factors=64* are used, fine-tuning other hyper-parameters becomes less critical. Furthermore, the observed reductions in μ and σ compared to the global case suggest that constraining a single hyper-parameter simplifies the overall tuning process.

For BPRMF, *learning rate=0.005* is the most impactful hyper-parameter overall. Higher learning rates tend to enhance multi-objective performance, potentially due to the sequential nature of BPRMF training. However, this behavior is not replicated in NeuMF. Instead, a pattern similar to that observed in NeuMF is seen in BPRMF, where *factors=64* ranks as the second most impactful hyper-parameter. This may stem from the larger embedding size enabling more effective modeling of user and item characteristics, addressing accuracy-related and beyond-accuracy objectives.

Notably, reducing the number of factors generally leads to an increase in μ for both models, particularly for NeuMF. These findings underscore the central role of the factors hyper-parameter in shaping the performance of factorization-based models, making it the most impactful parameter in this category.

Graph-based Models

Table 6.4 examines the influence of the factors, layers, and learning rate hyper-parameters for graph-based models, specifically NGCF and LightGCN. In most dataset-scenario combinations, *factors=64* emerges as the most impactful hyper-parameter, consistently producing Pareto optimal or near-optimal solutions. This configuration generally yields the lowest values of μ and acceptable levels of σ. These results align with those observed for factorization-based models, further confirming that increasing the embedding size enhances the model's capacity to capture latent user preferences and item characteristics better, ultimately leading to superior multi-objective performance.

In contrast to factorization models, a clear hierarchy of hyper-parameter importance is evident in the graph-based models. Notably, *layers=4* consistently ranks as the second most influential hyper-parameter, significantly contributing to Pareto optimal configurations. Conversely, the *learning rate* appears to be the least impactful hyper-parameter in this analysis, exhibiting minimal influence on the model's ability to achieve optimal trade-offs.

These findings underscore a recurring theme across different model families: the factors hyper-parameter is critical for achieving Pareto optimality, emphasizing the importance of embedding size in capturing multifaceted relationships in the data. Moreover, the structured impact of layers in graph-based models highlights the necessity of tailoring hyper-parameter choices to the unique architectural properties of each model type.

Neighborhood-based Models

Table 6.5 analyzes the influence of the neighbors and similarity hyper-parameters for the UserKNN and ItemKNN models. These models, particularly UserKNN, inherently exhibit lower sensitivity to precise hyper-parameter adjustments to achieve Pareto optimality, as previously demonstrated in Table 6.2. Consequently, this fine-grained analysis reveals fewer notable patterns than other model families. Indeed, numerous hyper-parameter configurations yield μ and σ values equal to zero, indicating that these configurations consistently guarantee Pareto optimal trade-offs without requiring extensive tuning.

Among the similarity metrics, cosine similarity emerges as the most effective, consistently achieving the best results in terms of μ and σ across various datasets and scenarios. This finding highlights the robustness of cosine similarity in facilitating optimal trade-offs between multiple objectives. Conversely, other metrics, such as Jaccard or Manhattan, exhibit more variability in their ability to produce Pareto optimal configurations.

These results confirm the inherent advantage of UserKNN and, to a lesser extent, ItemKNN in requiring minimal tuning effort, making them suitable choices for scenarios where simplicity and stability in hyper-parameter tuning are prioritized. Moreover, the consistent performance of cosine similarity suggests its suitability as a default choice when deploying neighborhood-based models.

6.6 Summary

This chapter addresses the challenge of hyper-parameter tuning sensitivity in Recommender Systems (RSs) when evaluating them under multiple objectives. By introducing a novel evaluation framework, we provide a systematic approach to understanding to what extent traditional RSs require precise hyper-parameter tuning to achieve Pareto optimal solutions in multi-objective scenarios. Using metrics like the mean distance to the Pareto frontier) and the standard deviation of such distances, the framework quantifies both the proximity and consistency of solutions relative to the Pareto frontier. This enables a nuanced analysis of hyper-parameter sensitivity, guiding researchers and practitioners toward configurations that balance accuracy with beyond-accuracy goals. We conduct experiments on six RS models spanning neighborhood-based, factorization-based, and graph-based approaches. These experiments are carried out under two scenarios: balancing accuracy with novelty and diversity and mitigating popularity bias alongside accuracy. The results reveal notable patterns. Neighborhood-based models, such as UserKNN, demonstrated low sensitivity to hyper-parameter tuning, consistently achieving Pareto-optimal solutions. In contrast, models like LightGCN and BPRMF exhibited higher sensitivity, requiring precise tuning to approach optimal performance. The analysis further identified the embedding size (factors) as the most impactful hyper-parameter across multiple models, with larger embeddings generally enhancing the capacity to effectively model user preferences and item characteristics. The findings underscore the practical trade-offs between flexibility and performance. While some models achieve robust multi-objective performance with minimal tuning, others demand precise hyper-parameter adjustments to deliver similar results. This insight is crucial for practitioners optimizing traditional RSs in diverse, real-world scenarios.

Table 6.3. Mean (i.e., μ) and standard deviation (i.e., σ) of the distances among specific hyper-parameter configurations of matrix factorization models and the Pareto frontier. Then, the μ and σ values are inspected for specific values of the hyper-parameters, categorized into the studied scenarios. For each model, the best values of μ and σ are in bold for each hyper-parameter type (i.e., factors and learning rate). Among them, the absolute best values are also underlined.

Trade-off	**Models**	**Factors**								**Learning Rate**							
		8		**16**		**32**		**64**		**0.0001**		**0.0005**		**0.001**		**0.005**	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Amazon Books																	
Acc	BPRMF	0.6901	**0.3180**	0.6736	0.3307	0.6637	0.3507	**0.6571**	0.3518	0.8498	**0.0028**	0.8489	0.0036	0.8464	0.0068	**0.1393**	0.1642
Bias	NeuMF	0.3751	0.2498	0.1562	0.2268	0.1248	0.1855	**0.054**	**0.0541**	0.396	0.2634	0.1167	0.2035	**0.0581**	**0.0668**	0.1055	0.0896
Acc	BPRMF	1.0815	**0.3386**	0.9942	0.5003	**0.9796**	0.526	0.9804	0.514	1.261	**0.0028**	1.2598	0.0035	1.2559	0.0087	**0.259**	0.2387
Nov-Div	NeuMF	0.6456	0.1911	0.2157	0.295	0.1734	0.2506	**0.0462**	**0.0674**	0.5283	0.3294	0.1933	0.3069	**0.1284**	**0.2066**	0.1717	0.2079
Amazon Music																	
Acc	BPRMF	**0.7095**	0.0441	0.772	0.0425	0.8855	**0.0407**	0.7504	0.3417	0.8674	0.1075	0.8301	0.0896	0.8047	**0.0709**	**0.6152**	0.2715
Bias	NeuMF	1.0849	0.2058	0.8015	**0.0704**	0.6007	0.0759	**0.4487**	0.2739	**0.6629**	0.5094	0.8262	0.2475	0.6941	0.1799	0.7527	**0.1283**
Acc	BPRMF	0.9584	0.0364	0.9856	0.0339	1.0793	**0.0217**	**0.8116**	0.5041	1.0644	0.0724	1.0338	0.0575	1.0065	**0.0529**	**0.7302**	0.4543
Nov-Div	NeuMF	1.3715	0.1701	0.9945	**0.0827**	0.6426	0.1122	**0.296**	0.1953	**0.7391**	0.6159	0.8904	0.4386	0.8162	0.3786	0.8589	**0.298**
Movielens1M																	
Acc	BPRMF	0.8813	**0.3083**	0.7901	0.3883	0.758	0.4293	**0.7376**	0.4222	1.2334	**0.0018**	0.9393	0.0935	0.7309	0.0502	**0.2633**	0.1994
Bias	NeuMF	0.1856	0.0881	0.0566	0.1163	**0.0496**	0.094	0.0581	**0.0622**	0.1992	0.1206	0.0678	0.0841	**0.0344**	0.0635	0.0485	**0.0584**
Acc	BPRMF	1.0818	**0.4665**	0.9367	0.5643	0.8762	0.6191	**0.8309**	0.6037	1.6441	**0.0021**	1.163	0.1832	0.6276	0.1511	**0.2909**	0.2284
Nov-Div	NeuMF	0.2731	0.0678	0.0861	0.1349	0.0464	0.0871	**0.0309**	**0.0269**	0.1926	0.1736	0.087	0.1153	**0.0674**	**0.096**	0.0895	0.1016

Table 6.4. Mean (i.e., μ) and standard deviation (i.e., σ) of the distances among specific hyper-parameter configurations of graph-based models and the Pareto frontier. Then, the μ and σ values are inspected for specific values of the hyper-parameters, categorized into the studied scenarios. For each model, the best values of μ and σ are in bold for each hyper-parameter type (i.e., factors, layers, and learning rate). Among them, the absolute best values are also underlined.

Trade-off	Models	Factors								Layers								Learning Rate			
		8		16		32		64		1		2		3		4		0.0005		0.001	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Amazon Books																					
Acc	NGCF	0.2090	0.1167	0.6234	**0.0676**	**0.1082**	0.0907	0.1796	0.1130	0.2422	0.0877	0.1811	0.0906	0.1064	0.0878	**0.0295**	**0.0415**	**0.1395**	0.1215	0.1401	**0.1031**
Bias	LightGCN	1.2323	0.042	0.8866	0.1153	0.5526	0.1205	**0.0511**	**0.0605**	0.7839	0.5092	**0.6246**	0.4807	0.637	0.4653	0.6772	**0.4199**	0.6847	**0.4549**	**0.6767**	0.4621
Acc	NGCF	1.0341	0.1969	0.5032	0.1072	0.1303	0.096	**0.0447**	**0.0457**	0.5668	0.4759	0.4605	0.4416	0.3753	0.4225	**0.3098**	**0.3387**	**0.4183**	**0.4157**	0.4379	0.4246
Nov-Div	LightGCN	1.4871	0.054	1.1374	0.1203	0.6536	0.1866	**0.0481**	**0.0526**	0.966	0.6299	0.7968	0.5866	**0.776**	0.5806	0.7873	**0.5339**	0.8422	**0.559**	**0.8209**	0.5799
Amazon Music																					
Acc	NGCF	1.1751	0.1325	0.8137	0.1454	0.4801	**0.0687**	**0.1545**	0.1396	0.7591	0.415	0.7384	0.4442	0.5939	**0.3925**	**0.5322**	0.3958	0.6664	**0.3854**	**0.6453**	0.4338
Bias	LightGCN	1.1668	0.1035	0.8868	**0.0973**	0.5401	0.1023	**0.0932**	0.0978	0.796	0.438	0.6807	0.4476	0.6389	0.434	**0.5713**	**0.4056**	0.6933	**0.4079**	**0.6502**	0.441
Acc	NGCF	1.4694	0.1591	1.057	0.179	0.6372	**0.0866**	**0.1168**	0.1244	0.9487	0.5557	0.9109	0.5856	0.7377	0.5189	**0.6831**	**0.5013**	**0.8137**	**0.511**	0.8265	0.5592
Nov-Div	LightGCN	1.3891	0.106	1.0137	0.0978	0.5266	0.1193	**0.0742**	**0.0855**	0.884	0.5362	0.7567	0.5527	0.7128	0.5379	**0.6502**	**0.5017**	0.7761	**0.5069**	**0.7257**	0.5353
Movielens1M																					
Acc	NGCF	0.2868	0.2784	0.1791	0.0965	0.1032	0.083	**0.0344**	**0.0708**	0.2538	0.2585	0.2122	0.1734	0.084	0.0736	**0.0535**	**0.0647**	0.183	0.1933	**0.1188**	**0.1589**
Bias	LightGCN	1.2258	**0.0957**	0.7302	0.1317	0.5883	0.2067	**0.2591**	0.2389	0.8637	**0.2231**	0.693	0.4447	0.6515	0.3998	**0.5951**	0.4757	**0.6815**	0.4002	0.7202	**0.3953**
Acc	NGCF	0.6154	0.3875	0.3198	0.2318	0.158	0.1427	**0.0344**	**0.0683**	0.4942	0.4703	0.346	0.2903	0.1841	0.1601	**0.1033**	**0.1027**	0.3157	0.3466	**0.2481**	**0.2919**
Nov-Div	LightGCN	1.5965	**0.0485**	0.8728	0.1071	0.6442	0.2083	**0.2561**	0.2215	1.0203	**0.3997**	0.8225	0.5607	0.7962	0.5462	**0.7306**	0.6076	**0.8224**	0.5333	0.8624	**0.5219**

Table 6.5. Mean (i.e., μ) and standard deviation (i.e., σ) of the distances among specific hyper-parameter configurations of neighborhood-based models and the Pareto frontier. Then, the μ and σ values are inspected for specific values of the hyper-parameters, categorized into the studied scenarios. For each model, the best values of μ and σ are in bold for each hyper-parameter type (i.e., number of neighbors and the similarity metric). Among them, the absolute best values are also underlined.

Trade-off	**Models**	**Neighbors**																**Similarity**							
		10		**20**		**30**		**50**		**100**		**150**		**200**		**250**		**cos**		**jac**		**euc**		**man**	
		μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Amazon Books																									
Acc	UserKNN	0.0781	0.0902	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.0039	0.0079	0.0056	0.0111	0.006	0.012	0.0196	0.0555	0.0272	0.0529	**0.0**	**0.0**	**0.0**	**0.0**
Bias	ItemKNN	0.0987	**0.0950**	0.1432	0.1863	0.1499	0.2596	0.1636	0.2962	0.1384	0.2528	0.1003	0.2008	0.0916	0.1798	**0.0912**	0.1824	**0.0167**	**0.0199**	0.4297	0.1201	0.0213	0.0379	0.0208	0.0319
Acc	UserKNN	0.5166	0.5864	0.4574	0.5221	0.4283	0.4946	0.389	0.4491	0.3476	0.4014	0.3345	0.3706	0.3315	0.3599	**0.3298**	**0.3557**	**0.0035**	**0.0068**	0.014	0.0199	0.7758	0.1431	0.774	0.1439
Nov-Div	ItemKNN	**0.6528**	0.7538	0.6819	0.6045	0.7015	0.537	0.7272	0.4834	0.7825	**0.4616**	0.8263	0.4713	0.8571	0.484	0.8774	0.4941	**0.0965**	**0.0455**	0.669	0.4235	1.1192	0.0866	1.1687	0.0652
Amazon Music																									
Acc	UserKNN	0.0522	0.0603	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.0142	0.0165	0.0162	0.0187	0.0206	0.0258	0.0228	0.0319	0.0135	0.0382	0.0355	0.0369	**0.0002**	**0.0007**	0.0138	0.0149
Bias	ItemKNN	**0.5369**	0.62	0.5457	0.6101	0.5762	0.5756	0.6058	0.5029	0.6491	0.4346	0.6614	**0.4182**	0.6853	0.4235	0.6894	0.4265	**0.0351**	**0.0312**	0.4221	0.3331	1.0085	0.0612	1.0091	0.0584
Acc	UserKNN	0.4358	**0.4483**	**0.3894**	0.4496	0.4031	0.4655	0.4144	0.4785	0.4242	0.4899	0.4259	0.4736	0.4342	0.4606	0.4405	0.4497	**0.0123**	0.0347	0.0264	0.0415	0.8317	0.0267	0.8134	**0.0189**
Nov-Div	ItemKNN	**0.8929**	0.9627	0.9257	0.9096	0.9814	0.8277	1.0211	0.7149	1.0472	0.6269	1.0488	0.6036	1.0527	0.5946	1.0421	**0.5827**	**0.2273**	**0.0654**	0.5624	0.3638	1.605	0.0994	1.6113	0.0906
Movielens1M																									
Acc	UserKNN	0.0001	0.0002	**0.0**	**0.0**	**0.0**	**0.0**	0.0006	0.0013	0.0011	0.0023	0.0405	0.0415	0.0628	0.0678	0.0804	0.087	**0.0**	**0.0**	0.0036	0.005	0.0464	0.0653	0.0428	0.0631
Bias	ItemKNN	0.0989	0.093	0.0504	0.0685	0.0523	**0.0608**	0.0531	0.0634	**0.0467**	0.0825	0.0558	0.0989	0.069	0.113	0.0809	0.1298	**0.018**	0.051	0.1839	0.0553	0.0196	**0.0229**	0.032	0.0414
Acc	UserKNN	0.1144	0.1322	0.0353	0.0408	**0.0043**	**0.0087**	0.0054	0.0107	0.0431	0.051	0.0858	0.0963	0.1272	0.1398	0.163	0.1669	**0.0**	**0.0**	0.0072	0.0135	0.1507	0.1141	0.1314	0.1126
Nov-Div	ItemKNN	0.1037	0.1009	0.0219	0.0438	**0.0**	**0.0**	0.03	0.0513	0.0655	0.131	0.1217	0.1704	0.1381	0.1924	0.1662	0.2132	0.0055	0.0154	**0.0017**	**0.0048**	0.2274	0.1655	0.0889	0.1006

Part III
Facing Challenges in Multi-Objective Recommender Systems

Chapter 7

Exposing the Challenges of Multi-Objective Recommendation through a Reproducibility Study

Providing effective suggestions is of predominant importance for successful Recommender Systems (RSs). Nonetheless, the need of accounting for additional multiple objectives has become prominent, from both the final users' and the item providers' points of view. This need has led to a new class of RSs, called *Multi-Objective Recommender Systems* (MORSs). These systems are designed to provide suggestions by considering multiple (conflicting) objectives simultaneously, such as diverse, novel, and fairness-aware recommendations. In this work, we reproduce a state-of-the-art study on MORSs that exploits a reinforcement learning agent to satisfy three objectives, i.e., accuracy, diversity, and novelty of recommendations. The selected study is one of the few MORSs where the source code and datasets are released to ensure the reproducibility of the proposed approach. Interestingly, we find that some challenges arise when replicating the results of the original work, due to the nature of multiple-objective problems. We also extend the evaluation of the approach to analyze the impact of improving user-centered objectives of recommendations (i.e., diversity and novelty) in terms of algorithmic bias. To this end, we take into consideration both popularity and category of the items. We discover some interesting trends in the recommendation performance according to different evaluation metrics. In addition, we see that the multi-objective reinforcement learning approach is responsible for increasing the bias disparity in the output of the recommendation algorithm for those items belonging to positively/negatively biased categories. We publicly release datasets and codes in the following GitHub repository: `https://github.com/sisinflab/MORS_reproducibility`.[1]

[1] This chapter is based on the work published in the *Proceedings of the 17th ACM Conference on Recommender Systems* (RecSys 2023) "Reproducibility of Multi-Objective Reinforcement Learning Recommendation: Interplay between Effectiveness and Beyond-Accuracy Perspectives".

7.1 Introduction

The primary focus of the recommender systems research community thus far has been to construct algorithms that can detect and propose accurate content to users. However, mainstream metrics – such as accuracy – have frequently been prioritized during this process, neglecting other high-quality content-derived elements. The last few years have highlighted the diversity and novelty of recommendations as indispensable factors for sparking user interest, as encouraging a heterogeneous array of relevant items is more likely to fulfill a user's diverse needs [75, 152]. In addition, the fairness in one- and two-sided marketplaces [35, 111, 147] and bias towards certain groups of items and users of recommendations [6, 228] have been classified as crucial due to the recent regulations on trustworthy AI [136].

To this extent, several works have been proposed that evaluate the recommendation techniques under a beyond-accuracy perspective or re-rank the recommendation lists [14, 111, 131]. However, it is worth noting that training a model accounting for the sole accuracy can lead to less than optimal recommendations. This narrowly focused training ignores other crucial goals, like encouraging long-term engagement, promoting a diverse range of user-item interactions, and ultimately leading to purchases. Nevertheless, achieving these objectives requires a differentiable function that considers both objectives, which is arduous. Consequently, Multi-Objective Optimization (MOO) [124] is constrained when the objectives are only represented by non-differentiable metrics or functions in specific domains.

Upon examining the current state of the recommendation research, a growing trend has emerged, with researchers sharing the source code used in their experiments and algorithms [51]. This trend is likely due to the mounting importance placed on reproducibility as a significant criterion in academic peer reviewing. However, for the majority of Multi-Objective Recommender Systems (MORSs) [7, 224], the source code was either not supplied or lacked necessary details, making it difficult to reproduce the findings reliably. The reasons behind this omission or lack of clarity remain obscure, as a proper scientific paper must include all pertinent information for others to replicate the research accurately.

Given the importance of the reproducibility aspects in the research community and the lack of reproducible works in MORSs, in this work, we reproduce the results of a state-of-the-art Multi-Objective Reinforcement Learning-based framework for recommendation. In particular, we choose the work of Stamenkovic et al. [168], who are among the few to release the source codes and datasets of their MORS work. This framework aims to promote relevant, diverse, and novel recommendations simultaneously. However, some limitations arise when reading the paper. First, the authors do not explicitly specify the criteria for choosing the best models when reporting their results. This aspect is critical because their framework simultaneously deals with multiple objectives. Moreover, the performance analysis is limited to the accuracy, diversity, and novelty of recommendations. Therefore, the contributions of our work are the following:

- We briefly survey the reproducibility context of MORSs, discovering that most MORSs studies are published without being accompanied by source codes and datasets, making reproducing these works difficult;
- We reproduce the work by Stamenkovic et al. [168] with their source codes and datasets. We analyze whether their results are reproducible even if some important details are missing. We further analyze if their framework allows controlling the influence of each objective, given the intrinsic nature of diversity and novelty of recommendation;
- We extend the evaluation of their framework to other dimensions, by assessing the algorithmic bias of their framework on several item categories. Firstly, we see if their framework amplifies the source bias in the recommendation output. Secondly, we analyze the equality and equity of the exposure of popular and unpopular items.

7.2 Context of Multi-Objective Recommender Systems

This section provides the context of the current state-of-the-art MORSs, briefly recapitulating the most relevant and recent works in the field. We survey these works from a reproducibility perspective. Indeed, we report Table 7.1 to summarize the papers which are considered *reproducible* and *non-reproducible*. We consider a paper to be reproducible according to two criteria [51]: the authors release i) a working version of the source code, ii) the dataset they use (possibly pre-processed). This review justifies our choice to reproduce the work by Stamenkovic et al. [168].

Many approaches have been adopted to address multiple objectives in the recommendation scenario. The most intuitive method is re-ranking the recommendations produced by an algorithm trained on accuracy metrics (e.g., nDCG, Recall) to fit other objectives. In this regard, Li et al. [111] propose a user-fairness oriented re-ranking strategy to make recommendations fair for advantaged and disadvantaged groups according to their level of activity. Rahmani et al. [147] reproduce this work discovering that the user-oriented re-ranking strategy does not mitigate popularity bias among users with different degrees of interest toward popular items. Therefore, they highlight the need to look at several dimensions of analysis when blending multiple objectives. Conversely, Naghiaei et al. [131] implement a re-ranking strategy to meet objectives both from user and provider fairness. Another family of works [114, 204] leverage on the Karush–Kuhn–Tucker (KKT) conditions [158] to blend the objectives in a scalarization function to gather a single Pareto optimal solution through a Multi-Gradient Descent Algorithm (MGDA) [57]. Last, some recent papers employ Multi-Objective Reinforcement Learning (MORL) to consider several objectives simultaneously. Ge et al. [70] propose MoFIR by considering CTR for relevance and item exposure for provider fairness as objectives. They modify a commonly used RL algorithm (DDPG) by introducing a conditioned network. Stamenkovic et al. [168]

Table 7.1. *Reproducible* and *non-reproducible* state-of-the-art works regarding MORSs..

Work	Venue/Journal	Year	Source Code	Datasets
Lin et al. [114]	RecSys	2019	✗	✓
Li et al. [111]	WWW	2021	✓	✓
Xie et al. [212]	WWW	2021	✗	✗
Naghiaei et al. [131]	SIGIR	2022	✓	✓
Wu et al. [204]	TOIS	2022	✗	✗
Ge et al. [70]	WSDM	2022	✗	✗
Stamenkovic et al. [168]	WSDM	2022	✓	✓

present SMORL, a Scalarized MORL framework to simultaneously satisfy accuracy, diversity, and novelty in session-based RSs.

Although most of MORSs works present prominent algorithms, some concerns about their reproducibility arise. As shown in Table 7.1, a limited number of papers are accompanied with source code and datasets, and then classified as reproducible. Among the reproducible papers, Rahmani et al. [147] already reproduced the work by Li et al. [111]. Therefore, we preferred extensively studying the work by Stamenkovic et al. [168]. The rationale behind this choice is twofold. On the one hand, the authors adopt a novel MORL approach (instead of a re-ranking strategy). On the other hand, they focus only on user-centered objectives, opening the way to extend the experiments to other evaluation dimensions. In this regard, we do not compare it with the work by Naghiaei et al. [131] who already performed an extensive evaluation of their work. Moreover, these two papers have different targets, making a comparison unreasonable.

7.3 Background

In their work, Stamenkovic et al. [168] introduce the *Scalarized Multi-Objective Reinforcement Learning* (SMORL) approach in session-based recommendations. This state-of-the-art algorithm exploits a single RL agent to solve the next item recommendation problem to produce suggestions that are simultaneously relevant, diverse, and novel. Therefore, the authors solely focus on user-centered metrics, excluding other stakeholders or dimensions of analysis.

Problem formulation

The next item recommendation problem is formulated as a Multi-Objective Markov Decision Process (MOMDP). Given a set of items $\mathcal{I}$, a user-item interaction sequence is represented as $\mathrm{x}_{1:t} = \{x_1, x_2, \ldots, x_t\}$, where $x_i \in \mathcal{I}$ and $t \in (0, t)$. The problem consists in recommending the next item, $\mathrm{x}_{\mathrm{t+1}}$. The MOMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathrm{P}, \mathrm{R}, \rho_0, \gamma)$. $\mathcal{S}$ is a continuous state space corresponding to the user state, which is defined at timestamp t as $\mathrm{s}_t = G(\mathrm{x}_{1:t}) \in \mathcal{S}(t > 0)$, with G a sequential model. $\mathcal{A}$ is a discrete action space, where an action a of the agent consists in suggesting a selected item. The utility score for the state-action pair (s_t, a_t) is defined by the multi-objective Q-value function $\mathrm{Q}(\mathrm{s}_t, a_t)$. $\mathrm{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow \mathbb{R}$ is a probability function of moving from state s_t to the next state s_{t+1} when the agent takes action a_t. $\mathrm{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^m$ is the reward function that returns a vector of rewards $\mathrm{r}(\mathrm{s}, a)$ by taking action a at state s, each corresponding to each objective. ρ_0 is the initial state. $\gamma \in [0, 1]$ is the discount factor. The solution for the MOMDP is to find the target policy $\pi_\theta(a|\mathrm{s})$ to maximize the expected cumulative reward:

$$\max_{\pi_\theta} \mathbb{E}_{\tau\sim\pi_\theta}[f(\mathrm{R}(\tau))], \text{ with } \mathrm{R}(\tau) = \sum_{t=0}^{|\tau|} \gamma^t \mathrm{r}\left(\mathrm{s}_t, a_t\right) \tag{7.1}$$

where $\theta \in \mathbb{R}^d$ denotes the policy parameters. The expectation is taken over trajectories $\tau = (\mathrm{s}_0, a_0, \mathrm{s}_1, a_1 \ldots)$, obtained by performing actions that follow a target policy: $\mathrm{s}_0 \sim \rho_0, a_t \sim \pi_\theta\left(\cdot \mid \mathrm{s}_t\right), \mathrm{s}_{t+1} \sim \mathrm{P}\left(\cdot \mid \mathrm{s}_t, a_t\right)$. The scalarization function $f : \mathbb{R}^m \mapsto \mathbb{R}$ is $f_{\mathrm{w}}(\mathrm{x}) = \mathrm{w}^T\mathrm{x}$, where $\mathrm{w} = [w_1, \ldots, w_m]$ are the weights to control the importance of the objectives.

The model

We now summarize the algorithm proposed by Stamenkovic et al. [168]. The authors cast this task as a self-supervised multi-class classification problem. They use a generative sequence model $G(\cdot)$ to map the user-item interaction sequence into a hidden state $\mathrm{s}_t = G(\mathrm{x}_{1:t})$. On the self-supervised head side, they define a fully connected layer to map s_t into classification logits $\mathrm{y}_{\mathrm{t+1}}$. Based on these logits, they train the model exploiting the cross entropy loss L_s. On the SMORL head side, it can be seen as a regularizer to make recommendations more diverse and novel. Here, the authors stack additional fully connected layers to calculate one-dimensional Q-values on top of G for each objective (i.e., accuracy, diversity, and novelty). Therefore, they obtain a vector-valued Q-value function $\mathrm{Q}(\mathrm{s}_t, a_t) = [Q_{acc}(\mathrm{s}_t, a_t), Q_{div}(\mathrm{s}_t, a_t), Q_{nov}(\mathrm{s}_t, a_t)]$, which they learn exploiting the *Scalarized Deep Q-learning* (SDQL) algorithm [130]. Therefore, the Q-network is optimized through the loss function L_{SDQL}, which is defined as follows:

$$L_{SDQL} = \left(\mathrm{w}^T(\mathrm{y}_t^{SDQL}(\mathrm{s}_t, a_t) - \gamma\mathrm{Q}(\mathrm{s}_{t+1}, a_t))\right)^2, \tag{7.2}$$

where $\mathrm{y}_t^{SDQL}(\mathrm{s}_t, a_t) = \mathrm{r}_t + \gamma\mathrm{Q}'(\mathrm{s}_{t+1}, \mathrm{argmax}_{a'}[\mathrm{w}^T\mathrm{Q}'(\mathrm{s}_{t+1}, a')])$, with Q' the target network. It is worth mentioning that $\mathrm{w} = [w_{acc}, w_{div}, w_{nov}]$, since accuracy, diversity, and novelty of recommendations are considered as objectives. In conclusion, the final loss optimized by Stamenkovic et al. [168] is:

$$L_{SMORL} = L_S + \alpha L_{SDQL}, \tag{7.3}$$

with α a hyperparameter that controls the influence of the SMORL part.

Rewards Definition

Finally, the authors define the rewards in the following ways. They set the accuracy reward as done by Xin et al. [213] when the algorithm matches the next clicked item in the sequence, i.e., $r_{acc}(\mathrm{s}_t, a_t) = 1$, where a_t is the action of clicking an item. Then, they define the diversity reward as:

$$r_{div} = r_{div}(\mathrm{s}_t, p_t) = 1 - \cos(l_t, p_t) = 1 - \frac{\mathrm{e}_{l_t}^T \mathrm{e}_{p_t}}{||\mathrm{e}_{l_t}||||\mathrm{e}_{p_t}||}, \tag{7.4}$$

with l_t is the last clicked item in the session, p_t is a top prediction obtained from self-supervised layer, and e_x is the embedding of the item x obtained from a pre-trained GRU4Rec model. Finally, they define the novelty reward as 0 if p_t is in the top 10% of most popular items, one otherwise.

7.4 Experiments Replication Methodology

In this section, we describe our methodology to reproduce the work by Stamenkovic et al. [168]. We provide details on the datasets and the recommendation algorithms used. Furthermore, we discuss the evaluation protocol adopted, highlighting the challenges when replicating this work. We aim to answer the following research questions:

RQ1: Are we able to replicate the results reported in the paper by Stamenkovic et al. [168]?

RQ2: Given the weights setting of SMORL in Equation (7.2), does the nature of the objectives make it difficult to control them by varying the values of the weights?

7.4.1 Experimental Settings

In our study, we do not aim to barely verify the reproducibility of the work by Stamenkovic et al. [168]. On the contrary, we aim to highlight challenges and critical features when dealing with multiple objectives in RSs. For this reason, we do not re-implement the codes and use the ones shared by the authors.

Datasets

We use session-based datasets from the original work, i.e., *RC 15*[2] and *Retailrocket*[3]. The sessions contain sequences of clicked items. The authors share both datasets in their pre-processed versions. Precisely, the authors discard sessions in RC 15 having

[2] `https://www.kaggle.com/datasets/chadgostopp/recsys-challenge-2015`

[3] `https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset`

a sequence length of less than three and then sample a subset of 200K sessions. In addition, we report what the authors write in the original paper about Retailrocket: "*We remove the items which are interacted less than three times (3) and the sequences whose length is smaller than three (3)*". However, by examining the code released for the pre-processing step, we noticed that the authors implement this step for the Retailrocket dataset as follows: (*i*) they firstly remove the sequences, and (*ii*) then they remove the items. This specific order of actions makes sessions of lengths 1 and 2 still exist in the final dataset. Indeed, some sessions whose length is greater than 2 (i.e., initially maintained) are characterized by items that are later removed, reducing the sequences. On the one hand, this aspect does not affect the replication of the experiments in this work. On the other hand, we deal with this issue when extending such experiments, as explained later in Section 7.5. After the pre-processing step, RC 15 has 200,000 sequences, 26,702 items, and 1,110,965 clicks, and Retailrocket has 195,523 sequences, 70,852 items, and 1,176,680 clicks.

Baselines

We reproduce the baselines implemented in the original work. Specifically, we perform experiments on the vanilla baselines and their versions integrated with SMORL. The baselines are the following:

- **GRU4Rec** [87]: This model uses a GRU-based Recurrent Neural Network (RNN) for session-based recommendations. The network's input is the session's actual state, while the output is the item on the next event in the session.
- **Caser** [179]: This model abandoned RNN structures by leveraging the convolution filters of Convolutional Neural Network (CNN) to capture sequential patterns on the embedding matrix of previous items within a session.
- **NextItNet** [216]: This model uses a convolutional generative network for session-based top-k item recommendations. The main characteristics are the dilated convolutional layer to maintain a wide receptive field and the residual connections to ease the network's training.
- **SASRec** [99]: This model uses a self-attention mechanism to capture the sequential patterns of user interactions within a session and generate embeddings for the items in the session.

We adopt the hyper-parameter settings of the original paper to ensure a fair reproducibility of the experiments. Moreover, we set $\alpha = 1$ in the loss function of Eq. (7.3), since 1 is generally the optimal value according to Stamenkovic et al. [168].

Evaluation protocol

In the original work, the authors state that they use 5-fold cross-validation to evaluate their algorithm by splitting the dataset into training, validation, and testing with a ratio of 8:1:1. We notice that such splitting is performed at the session level. Therefore, each session is uniquely entered into the training, validation, or testing set. In order to report the results in their paper, they average the performance across all folds. However, the authors share just one fold out of five. Consequently, their evaluation protocol is not entirely replicable with their original splitting. For this reason, we utilize the fold they share. Specifically, we train the models on the training set. Then, we choose the best iteration of the training process according to the value of nDCG@10 on the validation set. As done in the paper we replicate, we evaluate the validation set every 5,000 batches of updates on RC 15 and every 10,000 batches of updates on Retailrocket. Finally, we report the results of the testing set on that iteration. It is worth mentioning that Stamenkovic et al. [168] do not provide any information in the paper on the strategy for selecting the best iteration.

Each session in the validation/test set comprises a sequence of n items according to a timestamp t_i, with $i = 1, \dots, n$. The algorithm lists recommended items for each t_i in the session, amounting to n recommendations per session. Therefore, each list generated at t_i is compared with the actual item associated at timestamp t_i.

Evaluation metrics

In Section 7.3, we summarized the approach proposed by Stamenkovic et al. [168]. We have seen that their approach aims to provide accurate, diverse, and novel recommendations simultaneously. Indeed, they evaluate the model along these dimensions with the following metrics that we also use in this work. To measure to what extent the recommendations are accurate, we use Hit Ratio (HR@k) on clicks [168] and the normalized Discount Cumulative Gain [106] (nDCG@k), with $k \in \{10, 20\}$. Instead, both diversity and novelty of recommendations are measured in an aggregated manner through the Item Coverage [8] (CV@k), with $k \in \{1, 5, 10, 20\}$, of all top-k recommendations of the validation/test sequences. Specifically, we consider the coverage of all the items for diversity and the set of less popular items for novelty. Moreover, Stamenkovic et al. [168] introduce a novel metric to evaluate the repetitiveness of recommendations (R@k), with $k \in \{5, 10, 20\}$, that we report in the results.

7.4.2 Results and Discussion

Replication of the Experiments (RQ1)

We start the results discussion by answering RQ1. To this end, we report the results obtained when replicating the work by Stamenkovic et al. [168] in Table 7.2 and in Table 7.3 for RC 15 and Retailrocket datasets, respectively. In these tables, the results labeled with “Orig.” are retrieved from the original work, while the ones achieved in our reproducibility study are labeled with “Repr.”. In particular, for these results, we also report the best iteration (*Epoch* and *step* of validation) according to nDCG@10, as explained in Section 7.4.1.

Table 7.2. Replicated recommendation performance on RC 15 dataset. The results labeled with “Orig.” are retrieved from the original work, while the ones labeled with “Repr.” are replicated in this work. *Epoch* and *Step* refer to the best iteration on the validation set on nDCG@10. The best results are highlighted in bold. Arrows indicate whether better stands for low ↓ or high ↑ values.

Model	Type	Iteration		Accuracy↑				Diversity↑				Novelty↑				Repetitivness↓		
		Epoch	Step	HR@10	NG@10	HR@20	NG@20	CV@1	CV@5	CV@10	CV@20	CV@1	CV@5	CV@10	CV@20	R@5	R@10	R@20
GRU	Orig.			0.3793	0.2279	0.4581	0.2478	0.2481	0.4330	0.5188	0.5942	0.1777	0.3707	0.4654	0.5492	12.11	25.63	53.24
GRU-SMORL	Orig.			**0.4007**	**0.2433**	**0.4793**	**0.2632**	**0.2825**	**0.4758**	**0.5577**	**0.6334**	**0.2086**	**0.4176**	**0.5086**	**0.5927**	**11.29**	**23.81**	**48.88**
GRU	Repr.	2	10000	0.4089	0.2454	0.4891	0.2658	0.2519	**0.4398**	**0.5234**	**0.5993**	0.1824	**0.378**	**0.4705**	**0.5548**	**12.2515**	**25.9609**	**53.6975**
GRU-SMORL	Repr.	11	40000	**0.4123**	**0.2503**	**0.4916**	**0.2704**	**0.2613**	0.437	0.5091	0.5761	**0.1859**	0.3746	0.4546	0.529	12.3178	25.9971	53.7227
Caser	Orig.			0.3593	0.2177	0.4371	0.2372	0.2631	0.4349	0.5019	0.5608	0.1912	0.3724	0.4466	0.5120	14.38	29.65	60.73
Caser-SMORL	Orig.			**0.3664**	**0.2224**	**0.4425**	**0.2417**	**0.3174**	**0.5157**	**0.5944**	**0.6685**	**0.2476**	**0.4621**	**0.5495**	**0.6316**	**13.77**	**28.56**	**58.52**
Caser	Repr.	2	10000	**0.3699**	**0.2235**	**0.4481**	**0.2433**	0.2226	0.3621	0.4195	0.4672	0.1487	0.2917	0.3551	0.4081	14.5989	30.1426	61.5313
Caser-SMORL	Repr.	20	75000	0.3679	0.223	0.4444	0.2424	**0.299**	**0.5047**	**0.587**	**0.6619**	**0.2269**	**0.4499**	**0.5411**	**0.6244**	**14.4391**	**29.8228**	**60.9314**
NextItNet	Orig.			0.3885	0.2332	0.4684	0.2535	0.2950	0.4914	0.5705	0.6427	0.2313	0.4354	0.5228	0.6030	10.03	22.02	46.84
NextItNet-SMORL	Orig.			**0.4116**	**0.2505**	**0.4898**	**0.2703**	**0.3385**	**0.5639**	**0.6518**	**0.7283**	**0.2720**	**0.5156**	**0.6131**	**0.6981**	**9.97**	**21.73**	**45.49**
NextItNet	Repr.	2	10000	0.4137	0.2483	0.4937	0.2686	**0.296**	**0.4916**	**0.5673**	**0.6382**	**0.2328**	**0.4356**	**0.5192**	**0.598**	**10.3075**	**22.5104**	**47.816**
NextItNet-SMORL	Repr.	6	25000	**0.4208**	**0.2563**	**0.4987**	**0.2761**	0.2763	0.4605	0.524	0.5818	0.208	0.4009	0.4712	0.5354	10.516	22.9839	48.5665
SASRec	Orig.			0.4257	0.2599	0.5053	0.2801	0.2971	0.5208	0.6046	0.6792	0.2298	0.4679	0.5607	0.6436	10.62	23.24	49.28
SASRec-SMORL	Orig.			**0.4315**	**0.2651**	**0.5104**	**0.2851**	**0.3380**	**0.5755**	**0.6508**	**0.7158**	**0.2698**	**0.5285**	**0.6120**	**0.6842**	**10.38**	**22.79**	**48.48**
SASRec	Repr.	6	25000	0.4313	0.2625	0.5109	0.2827	0.304	0.5345	0.6214	0.6949	0.2375	0.4829	0.5793	0.661	10.7326	23.4566	49.7044
SASRec-SMORL	Repr.	29	105000	**0.4388**	**0.2691**	**0.5167**	**0.2888**	**0.3257**	**0.5576**	**0.6352**	**0.6994**	**0.2579**	**0.5084**	**0.5947**	**0.666**	**10.6158**	**23.2888**	**49.5189**

Table 7.3. Replicated recommendation performance on Retailrocket dataset. The results labeled with “Orig.” are retrieved from the original work, while the ones labeled with “Repr.” are replicated in this work. *Epoch* and *Step* refer to the best iteration on the validation set on nDCG@10. The best results are highlighted in bold. Arrows indicate whether better stands for low ↓ or high ↑ values.

Model	Type	Iteration		Accuracy↑				Diversity↑				Novelty↑				Repetitivness↓		
		Epoch	Step	HR@10	NG@10	HR@20	NG@20	CV@1	CV@5	CV@10	CV@20	CV@1	CV@5	CV@10	CV@20	R@5	R@10	R@20
GRU	Orig.			0.2673	0.1878	0.3082	0.1981	0.2439	0.4695	0.5699	0.6632	0.1837	0.4139	0.5238	0.6267	14.25	29.44	60.59
GRU-SMORL	Orig.			**0.3060**	**0.2103**	**0.3535**	**0.2224**	**0.2796**	**0.5369**	**0.6419**	**0.7353**	**0.2154**	**0.4871**	**0.6029**	**0.7064**	**13.53**	**28.02**	**57.89**
GRU	Repr.	2	10000	0.3154	0.2196	0.3653	0.2323	**0.2401**	**0.4712**	**0.5728**	**0.6649**	**0.1809**	**0.416**	**0.5267**	**0.6285**	**14.0767**	**28.9063**	**59.3497**
GRU-SMORL	Repr.	7	30000	**0.3338**	**0.2365**	**0.3853**	**0.2496**	0.2386	0.4643	0.5598	0.643	0.1716	0.4068	0.5119	0.6039	14.6226	30.1449	61.617
Caser	Orig.			0.2302	0.1675	0.2628	0.1758	0.2327	0.4379	0.5133	0.5718	0.1643	0.3773	0.4605	0.5252	16.16	33.24	68.39
Caser-SMORL	Orig.			**0.2657**	**0.1898**	**0.3052**	**0.1998**	**0.2855**	**0.5411**	**0.6324**	**0.7138**	**0.2224**	**0.4917**	**0.5925**	**0.6827**	**15.90**	**32.47**	**66.76**
Caser	Repr.	5	20000	0.2563	0.188	0.2908	0.1967	0.2255	0.4329	0.5079	0.5678	0.1581	0.3718	0.4545	0.5208	**16.0098**	**32.6635**	**66.8796**
Caser-SMORL	Repr.	12	50000	**0.2824**	**0.2056**	**0.323**	**0.2159**	**0.2414**	**0.4876**	**0.584**	**0.6658**	**0.1775**	**0.4328**	**0.5388**	**0.6292**	16.5034	33.673	68.6404
NtItNet	Orig.			0.3007	0.2060	0.3506	0.2186	0.2867	0.5113	0.6033	0.6837	0.2305	0.4595	0.5605	0.6495	12.25	25.76	54.00
NextItNet-SMORL	Orig.			**0.3183**	**0.2222**	**0.3659**	**0.2342**	**0.3429**	**0.6335**	**0.7351**	**0.8129**	**0.2800**	**0.5938**	**0.7062**	**0.7924**	**10.92**	**22.89**	**47.73**
NextItNet	Repr.	2	10000	0.3294	0.2276	0.3797	0.2403	0.2824	0.5481	0.6436	0.723	0.2231	0.4999	0.6051	0.693	12.0508	25.4494	53.0647
NextItNet-SMORL	Repr.	12	50000	**0.3371**	**0.236**	**0.3858**	**0.2483**	**0.3095**	**0.5918**	**0.6889**	**0.7677**	**0.246**	**0.5477**	**0.6549**	**0.7423**	**11.7523**	**24.627**	**51.1179**
SASRec	Orig.			0.3085	0.2107	0.3572	0.2227	0.2767	0.5305	0.6300	0.7149	0.2171	0.4806	0.5899	0.6838	15.67	32.27	66.07
SASRec-SMORL	Orig.			**0.3521**	**0.2477**	**0.4028**	**0.2605**	**0.3037**	**0.5724**	**0.6672**	**0.7476**	**0.2366**	**0.5261**	**0.6311**	**0.7202**	**12.58**	**26.69**	**56.14**
SASRec	Repr.	2	10000	0.335	0.2311	0.3869	0.2442	**0.2413**	**0.4917**	**0.5969**	**0.6881**	**0.1823**	**0.4386**	**0.5537**	**0.6543**	**15.3474**	**31.5314**	**64.4512**
SASRec-SMORL	Repr.	7	30000	**0.3493**	**0.2463**	**0.4034**	**0.26**	0.2368	0.4609	0.5532	0.632	0.1707	0.403	0.5045	0.5918	15.5551	32.0816	65.7265

For both datasets, in the original paper, the baselines integrated with SMORL consistently perform better than the vanilla versions for all metrics. This phenomenon is still observed in our replicated results but less frequently. More in-depth, SMORL versions outperform the vanilla baselines from the accuracy perspective (except for Caser in RC 15), confirming that SMORL enhances the accuracy power of the generative models. This trend is not borne out as frequently as before from the beyond-accuracy metrics. Specifically, GRU vanilla achieves higher beyond-accuracy values for both datasets. In addition, NextItNet and SASRec vanillas also follow this tendency in RC 15 and Retailrocket, respectively. We notice that our replication experiments reach even higher values of relevance metrics than in the paper of Stamenkovic et al. [168]. This observation indicates the success of the training process of our replication experiments and the goodness of the approach proposed by Stamenkovic et al. [168]. We conjecture that the observed differences on the beyond-accuracy side are due to the criterion of choosing the best iteration of the models based on nDCG@10, i.e., an accuracy metric, following the best practices in the RecSys research community [20]. In this regard, we reiterate that the authors give no indication of their criterion for choosing the best model iteration[4]. Here are other instances of these models, chosen by Stamenkovic et al. [168], that perform slightly worse on the accuracy side but consistently better on diversity and novelty, thus more clearly reflecting the original trends. However, we do not have enough information to identify them. This consideration opens a perspective issue in MORSs. Since we are dealing with multiple objectives, it is not clear how to choose the best model. Indeed, such objectives are often characterized by a trade-off among them. The best solution for this trade-off can be subjective, thus jeopardizing the comparison and reproducibility of the models. Therefore, as a research community, we should define and declare the criterion used to select a model when dealing with multiple objectives for a fair comparison of the algorithms.

To conclude, we answer RQ1 by saying that we can consistently replicate the trends of SMORL from the accuracy side. As for the beyond accuracy metrics, such trends are not as equal as in the original paper. Therefore, a perspective issue in MORSs is opened about the selection strategy of the best model when dealing with multiple objectives. We observe to what extent it is crucial to define and explain the criteria followed to report the results in the MORSs field.

Controlling the objectives (RQ2)

We now answer RQ2 by investigating whether we can control the influence of each objective in SMORL through the weights mechanism available in Equation (7.2). We summarize the results gathered by varying the weights configurations of SMORL in Table 7.4 and Table 7.5 for RC 15 and Retailrocket datasets, respectively. We do not report the results of the NextItNet-SMORL model since one configuration

[4] We rule out that they have reported the values at the last iteration. Indeed, for instance, we observed that GRU-based models achieve significantly worse performance than those reported when selected by this criterion.

Table 7.4. Recommendation performance on RC 15 dataset obtained by varying the weights configurations of SMORL. The best results are highlighted in bold, the second best results are underlined.

Iteration		Weights			Accuracy↑				Diversity↑				Novelty↑				Repetitivness↓		
Epoch	Step	w_{acc}	w_{div}	w_{nov}	HR@10	NG@10	HR@20	NG@20	CV@1	CV@5	CV@10	CV@20	CV@1	CV@5	CV@10	CV@20	R@5	R@10	R@20
GRU-SMORL																			
11	40000	1	1	1	0.4123	0.2503	0.4916	0.2704	**0.2613**	**0.437**	**0.5091**	**0.5761**	**0.1859**	**0.3746**	**0.4546**	**0.529**	**12.3178**	**25.9971**	**53.7227**
6	25000	0	1	0	0.4096	0.2478	0.4889	0.2679	0.2477	0.4248	0.4985	0.5658	0.1744	0.3612	0.4428	0.5176	12.3357	26.0906	53.9247
6	25000	0	0	1	0.4086	0.2468	0.4886	0.2671	0.2425	0.4168	0.4893	0.5592	0.169	0.3523	0.4325	0.5103	12.6314	26.638	54.9408
8	30000	0	1	1	0.4119	**0.2504**	0.4921	**0.2708**	0.252	0.4248	0.4945	0.5586	0.1766	0.361	0.4384	0.5096	12.4518	26.3624	54.4028
6	25000	1	1	0	0.4112	0.2496	0.4928	0.2703	0.2387	0.4052	0.478	0.5402	0.1638	0.3393	0.42	0.4892	12.5256	26.5378	54.8614
6	25000	1	0	1	**0.4134**	0.2503	**0.493**	0.2705	0.2376	0.4018	0.4694	0.5331	0.1629	0.3356	0.4105	0.4812	12.5936	26.634	54.9122
Caser-SMORL																			
20	75000	1	1	1	0.3679	0.223	0.4444	0.2424	0.299	0.5047	0.587	0.6619	0.2269	0.4499	0.5411	0.6244	14.4391	29.8228	60.9314
11	40000	0	1	0	0.3668	0.2217	0.4475	0.2422	0.2766	0.4659	0.5392	0.6033	0.2057	0.4068	0.488	0.5593	14.7028	30.2958	61.8642
24	90000	0	0	1	0.3683	0.2227	0.445	0.2422	0.3188	0.5206	0.5986	0.6723	0.2493	0.4674	0.5541	0.636	14.1055	29.1983	59.78
8	30000	0	1	1	0.3693	0.2233	**0.4492**	0.2436	0.2399	0.4063	0.4721	0.5289	0.1642	0.3407	0.4135	0.4766	14.8185	30.5242	62.3802
30	110000	1	1	0	**0.3718**	**0.2253**	0.4487	**0.2448**	**0.3284**	**0.5328**	**0.6156**	**0.6945**	**0.2591**	**0.481**	**0.5729**	**0.6606**	**13.7896**	**28.6924**	**58.8673**
35	130000	1	0	1	0.368	0.2234	0.4443	0.2428	0.3226	0.5243	0.6062	0.6883	0.2524	0.4715	0.5624	0.6537	13.8622	28.776	58.9865
SASRec-SMORL																			
29	105000	1	1	1	**0.4388**	**0.2691**	**0.5167**	**0.2888**	**0.3257**	0.5576	0.6352	0.6994	0.2579	0.5084	0.5947	0.666	10.6158	23.2888	49.5189
35	130000	0	1	0	0.4332	0.265	0.5124	0.2851	0.3332	0.5691	0.6493	0.7144	0.2665	0.5213	0.6103	0.6827	10.6494	23.3472	49.6007
31	115000	0	0	1	0.4332	0.2653	0.5113	0.2851	0.3347	**0.5726**	**0.6512**	**0.7179**	0.2679	**0.5253**	**0.6124**	**0.6866**	10.4927	**23.0172**	48.7668
31	115000	0	1	1	0.4346	0.2677	0.5134	0.2877	0.3262	0.5556	0.6318	0.6935	0.2579	0.5063	0.5909	0.6594	10.4846	23.0519	49.0306
45	165000	1	1	0	0.4353	0.2675	0.5127	0.2871	**0.3353**	0.5669	0.6457	0.7074	**0.2683**	0.5188	0.6064	0.6749	10.5834	23.213	49.1603
31	115000	1	0	1	0.4359	0.2675	0.5139	0.2873	0.3291	0.5598	0.6377	0.7	0.2621	0.5109	0.5975	0.6667	**10.4753**	23.0578	**49.0174**

Table 7.5. Recommendation performance on Retailrocket dataset obtained by varying the weights configurations of SMORL. The best results are highlighted in bold, and the second-best results are underlined.

Iteration		Weights			Accuracy↑				Diversity↑				Novelty↑				Repetitivness↓		
Epoch	Step	w_{acc}	w_{div}	w_{nov}	HR@10	NG@10	HR@20	NG@20	CV@1	CV@5	CV@10	CV@20	CV@1	CV@5	CV@10	CV@20	R@5	R@10	R@20
									GRU-SMORL										
7	30000	1	1	1	**0.3338**	**0.2365**	**0.3853**	**0.2496**	0.2386	0.4643	0.5598	0.643	0.1716	0.4068	0.5119	0.6039	14.6226	30.1449	61.617
5	20000	0	1	0	0.3289	0.2325	0.3781	0.2449	0.2396	0.4562	0.5438	0.622	0.1756	0.3985	0.4942	0.5807	**14.2112**	**29.401**	**60.2192**
5	20000	0	0	1	0.3296	0.2315	0.3792	0.2441	0.2346	0.4471	0.5348	0.6132	0.1707	0.3884	0.4846	0.571	14.4944	29.8744	61.3454
5	20000	0	1	1	0.3335	0.2359	0.3824	0.2483	0.2302	0.4338	0.5156	0.5875	0.1638	0.3732	0.463	0.5424	14.4541	29.8884	61.1587
7	30000	1	1	0	0.332	0.2332	0.3822	0.2459	**0.2479**	**0.4814**	**0.5798**	**0.6666**	**0.1817**	**0.4257**	**0.5342**	**0.6302**	14.2563	29.4297	60.3808
5	20000	1	0	1	0.3319	0.2349	0.3798	0.247	0.2281	0.4232	0.5028	0.5749	0.1622	0.3618	0.4488	0.5284	14.6393	30.1216	61.8217
									Caser-SMORL										
12	50000	1	1	1	0.2824	0.2056	0.323	0.2159	0.2414	0.4876	0.584	0.6658	0.1775	0.4328	0.5388	0.6292	16.5034	33.673	68.6404
12	50000	0	1	0	0.2827	0.2052	0.3229	0.2154	0.2433	0.4883	0.5836	0.6694	0.1784	0.4334	0.5384	0.6335	16.3698	33.4589	68.3446
10	40000	0	0	1	0.2561	0.1875	0.2918	0.1965	0.2189	0.4205	0.493	0.5493	0.1509	0.3582	0.4379	0.5002	16.1661	33.0279	67.384
18	70000	0	1	1	**0.2862**	**0.2076**	0.3263	**0.2178**	**0.2498**	**0.4941**	**0.5928**	**0.681**	**0.1839**	**0.44**	**0.5488**	**0.6463**	16.2249	33.175	67.8257
10	40000	1	1	0	0.2855	0.2074	**0.3267**	**0.2178**	0.2386	0.4691	0.561	0.6417	0.1731	0.4123	0.5133	0.6026	16.3606	33.3982	68.1704
12	50000	1	0	1	0.259	0.1908	0.2948	0.1998	0.2294	0.4387	0.5134	0.5745	0.1607	0.3782	0.4605	0.5281	**16.0087**	**32.6664**	**66.9529**
									SASRec-SMORL										
7	30000	1	1	1	**0.3493**	**0.2463**	**0.4034**	**0.26**	0.2368	0.4609	0.5532	0.632	0.1707	0.403	0.5045	0.5918	15.5551	32.0816	65.7265
7	30000	0	1	0	0.344	0.238	0.3976	0.2516	**0.2477**	**0.4967**	0.5969	0.6835	**0.1848**	**0.4431**	0.5534	0.6491	15.4283	**31.7588**	**65.0142**
7	30000	0	0	1	0.3431	0.2388	0.3965	0.2523	0.2471	0.4958	**0.5974**	**0.6862**	0.1841	0.442	**0.5538**	**0.6521**	**15.4162**	31.761	65.1188
7	30000	0	1	1	0.3488	0.2442	0.4028	0.2579	0.2467	0.4879	0.5829	0.6655	0.182	0.4333	0.5378	0.6291	15.4677	31.8648	65.3941
7	30000	1	1	0	0.3482	0.2425	0.4013	0.256	0.2427	0.4808	0.5768	0.6613	0.1781	0.4253	0.531	0.6244	15.4782	31.8257	65.2962
7	30000	1	0	1	0.347	0.2436	0.4017	0.2575	0.2435	0.4794	0.5748	0.6588	0.1787	0.4237	0.5287	0.6217	15.4631	31.907	65.3999

of this model took more than ten days to train for only one dataset, making it impractical to explore six different configurations for two datasets in a feasible time. By looking at Tables 7.4 and 7.5, apparently no single trend occurs. None of the weights configurations seems to perform better concerning any single analyzed objective generally. However, some considerations can be made. By reinforcing toward only either diversity or novelty (i.e., $w = [0, 1, 0]$ and $w = [0, 0, 1]$), we notice a reduced capability of the model to suggest relevant items. In addition, we do expect to power diverse or novel suggestions with these settings separately. Conversely, frequently these configurations outperform the others in terms of diversity and novelty, making explicit the difficulty in controlling these two objectives separately. This behavior is because these objectives are likely to be positively correlated; namely, the set of diverse items contains many novel items, and vice versa. Therefore, a general question arises on to what extent it is convenient to consider multiple positively correlated objectives in the same MORS. The risk is that we can train RSs to accomplish several objectives without having control over them. By adding the control on the accuracy side (i.e., $w = [1, 1, 1]$, $w = [1, 1, 0]$, and $w = [1, 0, 1]$), we notice a general improvement on the relevance of the item, as further confirmation of SMORL's ability to control the accuracy objective. Concerning the repetitiveness of recommendations, we do not note any particular phenomenon.

We answer RQ2 by saying that, with SMORL, we can control the reinforcing of the accuracy objectives while having more difficulties when controlling diversity and novelty individually due to their positive correlation. This opens a quest in MORSs. Is it appropriate to consider correlated objectives in the same system, or is it counterproductive to their control?

7.5 Bias Experiments

In the previous Section, we have replicated the work by Stamenkovic et al. [168]. In contrast, we extend our investigation on their MORL-based recommendation algorithm here. This investigation shares the same experimental setup used to train and produce the recommendations with the vanilla baselines and their versions integrated with SMORL in Section 7.4. Indeed, we provide a broader view of the experiments considering other evaluation dimensions. Given that the SMORL approach simultaneously focuses on user-centered objectives of recommendations (i.e., diversity and novelty), we analyze how enhancing user-oriented goals with such an approach can affect algorithmic bias in recommendations. Mainly, we assess the algorithmic bias from the popularity and the categories of the items. Therefore, we drive the analysis in order to answer the following research questions:

RQ3: Is SMORL responsible for bias disparity for certain categories of items in the recommendations output given the source bias?

RQ4: Does SMORL affect the equality and equity of items' exposure concerning their popularity?

7.5.1 Algorithmic Bias investigation settings

In this section, we present how we broaden the experimental evaluation of the multi-objective recommendation algorithm proposed by Stamenkovic et al. [168] that we have reproduced. To our knowledge, most algorithmic bias metrics available in the literature are not time-aware. For this reason, such metrics do not fit with the task of sequential-based recommendations. In addition, amid the proliferation of scholarly works on session-based recommendation in recent times, regrettably, there are no universally recognized benchmark datasets or evaluation standards that have gained consensus within the research community. To have a reliable setting, we must cast the sequential-based recommendations to a classic task. In this regard, in Section 7.4.1, we have observed that the baselines considered in this work generate n lists of suggested items for each session associated by timestamp, t_i. We treat sessions as users interacting with an ordered list of n items. We consider the last item of these lists as belonging to the test set, while the remaining items are in the training set. For this reason, we consider only the last recommendation lists generated by the baselines, i.e., the ones generated at t_n, for the evaluation in terms of algorithmic bias. Indeed, the baselines provide these last recommendation lists having in memory the entire previous sequences of items the users have interacted with. In addition, we previously discarded the sessions of length 1 in the test set of the Retailrocket dataset resulting from the pre-processing step described in Section 7.4.1. This operation is necessary to compute the algorithmic bias metrics presented later on, but it should not impact the evaluation since the discarded sessions are only 514.

Evaluation metrics

This section describes the metrics exploited to show the performance on the algorithmic bias of the work by Stamenkovic et al. [168]. Specifically, we consider the following metrics (see Section 2.4 for details):

- **Bias Disparity (BD)** [182]. This metric evaluates the discrepancy between input bias and recommendation bias. It captures instances where recommendation algorithms amplify existing biases in the data source, resulting in more pronounced biases in the generated recommendations.
- **Ranking-based Statistical Parity (RSP)** [228]. This metric is based on the concept of statistical parity, which involves ensuring that the ranking probability distributions for different item categories are identical in a ranking task.
- **Ranking-based Equal Opportunity (REO)** [228]. In contrast to RSP, this metric is founded on the principle of equal opportunity. Within a ranking task, this means ensuring that the ranking-based true positive rate (TPR) is consistent across different item categories. The TPR is defined as the probability that an item from a specific category is ranked within the top-k, given that the user likes the item according to the ground truth.

To provide a complete reproducibility environment for these metrics, we use the Elliot recommendation framework [11].

Item categories

In the previous section, we have presented the metrics we use to evaluate the algorithmic bias which affects the approach proposed by Stamenkovic et al. [168]. In all these metrics, the items are associated with a category. Here, we describe how and in which categories we assign each item.

We compute BD by considering a unique group of sessions corresponding to the entire dataset, i.e., $|G| = 1$, while we assess the item's category by exploiting additional information provided with the datasets. As for the RC 15 dataset, it provides the context of the item click, which can be a special offer, brand, or category identifier. Specifically, we take into account this last identifier for the assignment of the items to a category. In addition, since the context of a click varies in time, an item could be associated with more categories. However, we desire to have a unique category for each item. To this end, we set the most recent category for each item, being able to divide 19,288 items into $|C| = 12$ categories uniquely having the following sizes: $|c_1| = 5749$, $|c_2| = 4776$, $|c_3| = 1252$, $|c_4| = 1388$, $|c_5| = 1032$, $|c_6| = 1174$, $|c_7| = 881$, $|c_8| = 644$, $|c_9| = 674$, $|c_{10}| = 360$, $|c_{11}| = 1267$, $|c_{12}| = 91$. The Retailrocket dataset yields some properties for each item. Among them, a category identifier is available. In addition, the category identifiers are associated with a parent identifier according to a tree of categories. For this reason, we consider these parent identifiers as the categories of the item. Similarly to what was done for the RC 15 dataset, we uniquely assign the most recent category to each item. However, more than 250 categories were recognized in this way, making reporting and analyzing the results unattainable. To solve this issue, we split these categories into quartiles based on the number of interactions they were involved in. Hence, we can divide 65,663 items into $|C| = 4$ categories containing the following number of items: $|c_1| = 868$, $|c_2| = 4875$, $|c_3| = 13444$, $|c_4| = 46476$.

We calculate RSP and REO by dividing the items according to their popularity. In particular, we identify a category that comprises the 20% most popular items and a category including the remaining items. We will refer to RSP and REO when dealing with these categories as PopRSP and PopREO, respectively.

7.5.2 Results and Discussion

Bias Disparity (RQ3)

We start our investigation of the algorithmic bias in SMORL by answering RQ3. Given the existing bias in the datasets regarding the item categories, we aim to assess whether SMORL amplifies this in the output of the recommendations. Consequently, since it is not a de-biasing algorithm, we desire BD values close to 0, which means

Table 7.6. Results on Source Bias/Recommendation Bias (Bias Disparity) for different categories of items on RC 15 dataset. Values of Bias Disparity closer to 0 are in bold, assessed for different cutoff@k, with $k \in \{5, 10, 20\}$.

Model	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
Cutoff@5												
GRU	0.82/0.79 (-0.03)	0.83/0.86 (**0.03**)	4.66/4.51 (**-0.03**)	0.82/0.84 (**0.02**)	1.07/1.26 (0.18)	0.66/0.71 (0.07)	1.25/1.21 (**-0.03**)	0.23/0.2 (**-0.13**)	0.43/0.43 (**-0.01**)	0.27/0.25 (**-0.07**)	0.09/0.06 (**-0.3**)	0.31/0.28 (**-0.1**)
GRU-SMORL	0.82/0.8 (**-0.02**)	0.83/0.88 (0.05)	4.66/4.52 (**-0.03**)	0.82/0.86 (0.05)	1.07/1.21 (**0.13**)	0.66/0.66 (**0.0**)	1.25/1.2 (-0.04)	0.23/0.17 (-0.23)	0.43/0.43 (-0.02)	0.27/0.19 (-0.27)	0.09/0.05 (-0.4)	0.31/0.26 (-0.18)
Caser	0.82/0.81 (**-0.01**)	0.83/0.86 (**0.03**)	4.66/4.67 (**0.0**)	0.82/0.83 (**0.01**)	1.07/1.17 (**0.09**)	0.66/0.64 (-0.04)	1.25/1.25 (**0.0**)	0.23/0.15 (**-0.32**)	0.43/0.38 (-0.12)	0.27/0.18 (-0.31)	0.09/0.04 (-0.53)	0.31/0.24 (**-0.21**)
Caser-SMORL	0.82/0.8 (-0.02)	0.83/0.87 (0.05)	4.66/4.6 (-0.01)	0.82/0.84 (0.02)	1.07/1.18 (0.1)	0.66/0.65 (**-0.02**)	1.25/1.26 (0.01)	0.23/0.14 (-0.36)	0.43/0.41 (**-0.06**)	0.27/0.19 (**-0.28**)	0.09/0.05 (**-0.44**)	0.31/0.19 (-0.39)
NextItNet	0.82/0.8 (-0.02)	0.83/0.86 (**0.03**)	4.66/4.61 (**-0.01**)	0.82/0.84 (**0.02**)	1.07/1.2 (**0.12**)	0.66/0.68 (0.03)	1.25/1.17 (-0.06)	0.23/0.18 (-0.19)	0.43/0.39 (**-0.11**)	0.27/0.24 (**-0.09**)	0.09/0.08 (**-0.1**)	0.31/0.27 (**-0.12**)
NextItNet-SMORL	0.82/0.82 (**0.0**)	0.83/0.88 (0.06)	4.66/4.42 (-0.05)	0.82/0.86 (0.05)	1.07/1.21 (0.13)	0.66/0.66 (**0.0**)	1.25/1.25 (**0.0**)	0.23/0.19 (**-0.17**)	0.43/0.38 (-0.12)	0.27/0.21 (-0.21)	0.09/0.04 (-0.55)	0.31/0.2 (-0.37)
SASRec	0.82/0.81 (**-0.01**)	0.83/0.86 (**0.04**)	4.66/4.49 (**-0.04**)	0.82/0.83 (**0.01**)	1.07/1.22 (0.14)	0.66/0.68 (0.03)	1.25/1.27 (0.02)	0.23/0.19 (**-0.15**)	0.43/0.38 (-0.13)	0.27/0.25 (**-0.05**)	0.09/0.06 (**-0.3**)	0.31/0.25 (-0.18)
SASRec-SMORL	0.82/0.83 (0.02)	0.83/0.88 (0.06)	4.66/4.39 (-0.06)	0.82/0.84 (0.02)	1.07/1.13 (**0.05**)	0.66/0.66 (**0.0**)	1.25/1.26 (**0.01**)	0.23/0.19 (-0.18)	0.43/0.42 (**-0.03**)	0.27/0.19 (-0.28)	0.09/0.07 (-0.17)	0.31/0.29 (**-0.07**)
Cutoff@10												
GRU	0.82/0.79 (-0.03)	0.83/0.86 (**0.04**)	4.66/4.5 (**-0.03**)	0.82/0.85 (**0.03**)	1.07/1.27 (0.18)	0.66/0.7 (0.06)	1.25/1.22 (**-0.02**)	0.23/0.19 (**-0.16**)	0.43/0.44 (**0.0**)	0.27/0.24 (**-0.1**)	0.09/0.06 (**-0.31**)	0.31/0.27 (-0.13)
GRU-SMORL	0.82/0.8 (**-0.02**)	0.83/0.87 (0.05)	4.66/4.53 (**-0.03**)	0.82/0.87 (0.06)	1.07/1.22 (**0.14**)	0.66/0.66 (**0.0**)	1.25/1.21 (-0.03)	0.23/0.17 (-0.26)	0.43/0.43 (**0.0**)	0.27/0.19 (-0.27)	0.09/0.05 (-0.41)	0.31/0.27 (**-0.12**)
Caser	0.82/0.8 (**-0.01**)	0.83/0.86 (**0.03**)	4.66/4.66 (**0.0**)	0.82/0.83 (**0.01**)	1.07/1.18 (**0.1**)	0.66/0.63 (-0.05)	1.25/1.27 (**0.02**)	0.23/0.14 (**-0.37**)	0.43/0.38 (-0.12)	0.27/0.2 (**-0.26**)	0.09/0.04 (-0.51)	0.31/0.25 (**-0.18**)
Caser-SMORL	0.82/0.79 (-0.03)	0.83/0.88 (0.05)	4.66/4.6 (-0.01)	0.82/0.85 (0.03)	1.07/1.18 (**0.1**)	0.66/0.64 (**-0.03**)	1.25/1.27 (**0.02**)	0.23/0.14 (**-0.37**)	0.43/0.41 (**-0.06**)	0.27/0.19 (-0.29)	0.09/0.05 (**-0.45**)	0.31/0.19 (-0.37)
NextItNet	0.82/0.8 (-0.02)	0.83/0.86 (**0.03**)	4.66/4.62 (**-0.01**)	0.82/0.84 (**0.02**)	1.07/1.19 (**0.11**)	0.66/0.69 (0.04)	1.25/1.17 (-0.06)	0.23/0.18 (**-0.19**)	0.43/0.39 (-0.11)	0.27/0.24 (**-0.08**)	0.09/0.08 (**-0.12**)	0.31/0.26 (**-0.16**)
NextItNet-SMORL	0.82/0.82 (**0.0**)	0.83/0.88 (0.05)	4.66/4.44 (-0.05)	0.82/0.85 (0.04)	1.07/1.21 (0.13)	0.66/0.67 (**0.01**)	1.25/1.25 (**0.0**)	0.23/0.18 (**-0.19**)	0.43/0.39 (**-0.1**)	0.27/0.21 (-0.22)	0.09/0.04 (-0.56)	0.31/0.19 (-0.39)
SASRec	0.82/0.81 (**-0.01**)	0.83/0.86 (**0.04**)	4.66/4.48 (**-0.04**)	0.82/0.83 (**0.02**)	1.07/1.21 (0.13)	0.66/0.68 (0.03)	1.25/1.27 (0.02)	0.23/0.2 (**-0.12**)	0.43/0.38 (-0.13)	0.27/0.25 (**-0.06**)	0.09/0.06 (-0.3)	0.31/0.25 (-0.2)
SASRec-SMORL	0.82/0.83 (0.02)	0.83/0.88 (0.06)	4.66/4.39 (-0.06)	0.82/0.84 (0.03)	1.07/1.13 (**0.06**)	0.66/0.67 (**0.01**)	1.25/1.26 (**0.01**)	0.23/0.18 (-0.2)	0.43/0.43 (**-0.01**)	0.27/0.2 (-0.24)	0.09/0.07 (**-0.2**)	0.31/0.29 (**-0.05**)
Cutoff@20												
GRU	0.82/0.8 (**-0.02**)	0.83/0.87 (**0.05**)	4.66/4.45 (-0.05)	0.82/0.85 (**0.03**)	1.07/1.26 (0.18)	0.66/0.7 (0.06)	1.25/1.21 (**-0.03**)	0.23/0.2 (**-0.1**)	0.43/0.43 (**0.0**)	0.27/0.25 (**-0.06**)	0.09/0.06 (**-0.31**)	0.31/0.25 (-0.18)
GRU-SMORL	0.82/0.8 (**-0.02**)	0.83/0.88 (0.06)	4.66/4.48 (**-0.04**)	0.82/0.86 (0.05)	1.07/1.23 (**0.15**)	0.66/0.67 (**0.01**)	1.25/1.19 (-0.05)	0.23/0.17 (-0.24)	0.43/0.43 (-0.01)	0.27/0.2 (-0.23)	0.09/0.05 (-0.44)	0.31/0.26 (**-0.17**)
Caser	0.82/0.81 (**-0.01**)	0.83/0.87 (**0.05**)	4.66/4.6 (**-0.01**)	0.82/0.83 (**0.02**)	1.07/1.18 (**0.1**)	0.66/0.64 (-0.03)	1.25/1.25 (**0.0**)	0.23/0.15 (**-0.32**)	0.43/0.37 (-0.14)	0.27/0.21 (**-0.23**)	0.09/0.04 (-0.54)	0.31/0.23 (**-0.26**)
Caser-SMORL	0.82/0.8 (-0.02)	0.83/0.89 (0.07)	4.66/4.53 (-0.03)	0.82/0.84 (0.03)	1.07/1.18 (**0.1**)	0.66/0.65 (**-0.02**)	1.25/1.26 (0.01)	0.23/0.15 (-0.34)	0.43/0.4 (**-0.07**)	0.27/0.19 (-0.28)	0.09/0.05 (**-0.47**)	0.31/0.19 (-0.39)
NextItNet	0.82/0.8 (-0.01)	0.83/0.87 (**0.04**)	4.66/4.56 (**-0.02**)	0.82/0.84 (**0.02**)	1.07/1.19 (**0.11**)	0.66/0.69 (0.04)	1.25/1.16 (-0.07)	0.23/0.19 (-0.16)	0.43/0.39 (-0.09)	0.27/0.27 (**0.0**)	0.09/0.07 (**-0.16**)	0.31/0.25 (**-0.21**)
NextItNet-SMORL	0.82/0.82 (**0.0**)	0.83/0.89 (0.06)	4.66/4.41 (-0.05)	0.82/0.85 (0.03)	1.07/1.22 (0.13)	0.66/0.68 (**0.02**)	1.25/1.22 (**-0.02**)	0.23/0.19 (**-0.14**)	0.43/0.4 (**-0.08**)	0.27/0.22 (-0.18)	0.09/0.04 (-0.56)	0.31/0.18 (-0.41)
SASRec	0.82/0.81 (**0.0**)	0.83/0.87 (**0.05**)	4.66/4.44 (**-0.05**)	0.82/0.84 (**0.02**)	1.07/1.21 (0.13)	0.66/0.7 (0.05)	1.25/1.24 (**0.0**)	0.23/0.2 (**-0.1**)	0.43/0.38 (-0.12)	0.27/0.26 (**-0.01**)	0.09/0.06 (-0.31)	0.31/0.24 (-0.23)
SASRec-SMORL	0.82/0.83 (0.02)	0.83/0.89 (0.07)	4.66/4.35 (-0.07)	0.82/0.84 (0.03)	1.07/1.13 (**0.06**)	0.66/0.68 (**0.02**)	1.25/1.24 (-0.01)	0.23/0.19 (-0.15)	0.43/0.43 (**0.0**)	0.27/0.22 (-0.18)	0.09/0.07 (**-0.23**)	0.31/0.28 (**-0.1**)

Table 7.7. Results on Source Bias/Recommendation Bias (Bias Disparity) for different categories of items on Retailrocket dataset. Values of Bias Disparity closer to 0 are in bold, assessed for different cutoff@k, with $k \in \{5, 10, 20\}$.

Model	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4
	Cutoff@5				Cutoff@10				Cutoff@20			
GRU	0.43/0.34 (**-0.22**)	0.65/0.54 (-0.17)	0.76/0.73 (**-0.04**)	1.12/1.14 (**0.02**)	0.43/0.35 (**-0.18**)	0.65/0.53 (-0.18)	0.76/0.74 (**-0.03**)	1.12/1.14 (**0.02**)	0.43/0.37 (**-0.14**)	0.65/0.54 (-0.18)	0.76/0.75 (**-0.01**)	1.12/1.13 (**0.01**)
GRU-SMORL	0.43/0.3 (-0.29)	0.65/0.55 (**-0.15**)	0.76/0.7 (-0.08)	1.12/1.15 (0.03)	0.43/0.3 (-0.31)	0.65/0.55 (**-0.15**)	0.76/0.7 (-0.08)	1.12/1.15 (0.03)	0.43/0.31 (-0.27)	0.65/0.56 (**-0.14**)	0.76/0.71 (-0.07)	1.12/1.14 (0.02)
Caser	0.43/0.22 (**-0.48**)	0.65/0.6 (**-0.08**)	0.76/0.69 (**-0.1**)	1.12/1.15 (**0.03**)	0.43/0.23 (-0.46)	0.65/0.58 (**-0.1**)	0.76/0.71 (**-0.08**)	1.12/1.14 (**0.02**)	0.43/0.23 (-0.46)	0.65/0.58 (**-0.11**)	0.76/0.72 (**-0.06**)	1.12/1.14 (**0.02**)
Caser-SMORL	0.43/0.22 (-0.49)	0.65/0.5 (-0.23)	0.76/0.68 (-0.12)	1.12/1.16 (0.04)	0.43/0.23 (**-0.45**)	0.65/0.5 (-0.24)	0.76/0.67 (-0.12)	1.12/1.16 (0.04)	0.43/0.25 (**-0.42**)	0.65/0.5 (-0.23)	0.76/0.68 (-0.11)	1.12/1.16 (0.04)
NextItNet	0.43/0.32 (-0.25)	0.65/0.54 (**-0.18**)	0.76/0.69 (**-0.09**)	1.12/1.15 (**0.03**)	0.43/0.31 (-0.27)	0.65/0.53 (**-0.18**)	0.76/0.7 (**-0.09**)	1.12/1.15 (**0.03**)	0.43/0.31 (-0.29)	0.65/0.54 (**-0.17**)	0.76/0.7 (**-0.08**)	1.12/1.15 (**0.03**)
NextItNet-SMORL	0.43/0.34 (**-0.21**)	0.65/0.51 (-0.22)	0.76/0.69 (-0.1)	1.12/1.15 (**0.03**)	0.43/0.32 (**-0.25**)	0.65/0.51 (-0.22)	0.76/0.69 (**-0.09**)	1.12/1.15 (**0.03**)	0.43/0.32 (**-0.24**)	0.65/0.52 (-0.2)	0.76/0.7 (**-0.08**)	1.12/1.15 (**0.03**)
SASRec	0.43/0.41 (**-0.05**)	0.65/0.55 (-0.16)	0.76/0.72 (-0.06)	1.12/1.14 (**0.02**)	0.43/0.42 (**-0.03**)	0.65/0.55 (-0.16)	0.76/0.72 (-0.06)	1.12/1.14 (**0.02**)	0.43/0.43 (**0.01**)	0.65/0.56 (**-0.14**)	0.76/0.73 (-0.04)	1.12/1.14 (**0.02**)
SASRec-SMORL	0.43/0.35 (-0.19)	0.65/0.55 (**-0.15**)	0.76/0.73 (**-0.04**)	1.12/1.14 (**0.02**)	0.43/0.36 (-0.17)	0.65/0.55 (**-0.15**)	0.76/0.73 (**-0.05**)	1.12/1.14 (**0.02**)	0.43/0.36 (-0.15)	0.65/0.56 (**-0.14**)	0.76/0.74 (**-0.03**)	1.12/1.13 (**0.02**)

that the algorithm does not deviate from the source bias when providing the recommendations. In Table 7.6 and Table 7.7, we show the results regarding the bias disparity. We follow the representation of Tsintzou et al. [182], in which we report the input/output bias and in parentheses the bias disparity (i.e., BS/BR (BD)).

Starting from the RC 15 dataset (Table 7.6), we detect that the items in the category c_3 are characterized by a strong positive source bias (values much greater than one), indicating that the items from this category are enjoyed in the sessions disproportionately to the category size. Indeed, we observe that c_3 is not the most populated category (1252 items). Therefore, it contains trendy items. Paying attention to this category, we notice that SMORL-integrated models have absolute values of bias disparity greater than their respective vanilla versions. We explain this phenomenon by affirming that SMORL promotes diverse and novel items, thus reducing the number of recommended mainstream items (which populate c_3). Indeed, the sign of the bias disparity values is negative. Consequently, although SMORL is useful in diversifying recommendations, it is responsible for a higher bias disparity in output. This observation highlights the need for MORSs to focus on several lines of analysis and not only on the objectives that the system is optimizing. Indeed, we are not claiming that such bias introduces negative effects in the recommendation, but we should be aware of what is happening in other aspects. Other concerns occur for the items in category c_{10} that exhibit a strong negative bias in the source, i.e., they are barely relished in the sessions (the same following considerations hold for categories c_{11} and c_{12}). In addition, we observe that category c_{10} is populated by only 360 items. We notice that SMORL-based models convey greater values of bias disparity for this category than vanilla baselines, further reducing the exposure of these items. We conjecture that this is because, in SMORL, the agent is rewarded if it recommends relevant items. Therefore, it promotes diverse and novel items, but still relevant (i.e., items belonging to the mid-tail [3]), at the expense of highly niche items that are relevant to very few sessions. Consequently, by rewarding the agent for item relevance (we remind that the generative models are trained to provide relevant recommendations), we risk affecting the exposure of highly niche items, resulting unfair from the provider fairness point of view [1, 29, 55]. Finally, we do not remark on phenomenons for the remaining categories. Indeed, they are less biased at the source (values close to 1), and all the baselines maintain a roughly equal level of bias in the output.

A similar discussion emerges for the Retailrocket dataset (Table 7.7). On the one hand, both vanillas and SMORL-integrated models do not notably increase the bias in the output for the categories less biased in the source (c_3 and c_4). On the other hand, SMORL still increases the bias of those categories negatively biased at the source (generally more than vanilla baselines), as in the case of c_1 and c_2.

To end, we answer RQ3 by claiming that SMORL is responsible for injecting a bias disparity in the recommendations output for those items laying on positively/negatively biased categories in the source. Relevant items-based rewards can potentially reduce the exposure of highly niche items, leading to a provider-side unfair situation. Conversely, reinforcing diversity and novelty objectives decreases the bias of categories containing mainly interacted items.

Popularity-based Equality and Equity of Items Exposure (RQ4)

Finally, we answer RQ4 by analyzing the algorithmic bias affecting the items' exposure, dividing the items into popular and unpopular categories. We look at this from two perspectives. With PopRSP, we assess the equality of the ranking probability of these items. At the same time, with PopREO, we evaluate the equity of the ranking probability of these items given the items enjoyed in the sessions [80, 221]. Figure 7.1 shows the results for both datasets. From the equality side, all models seem strongly biased, having high values of PopRSP for both datasets. GRU and NextItNet vanillas produce more biased recommendations when integrated with SMORL. In contrast, blending SMORL in Caser makes the suggestions less biased, while SASRec does not show a specific trend. Therefore, the success of SMORL in terms of equality of the ranking probability of popular and unpopular items is related to the generative model of recommendations. From the equity side, SMORL models generally have higher values of PopREO, except for NextItNet and Caser with Retailrocket, which means that they produce more biased recommendations than vanilla baselines.

In conclusion, we answer RQ4 by stating that we do not observe particular trends on the positive or negative effect of SMORL on the equality of the popular and unpopular item's exposure. Regarding equity, SMORL tends to worsen the recommendation bias more than vanilla baselines.

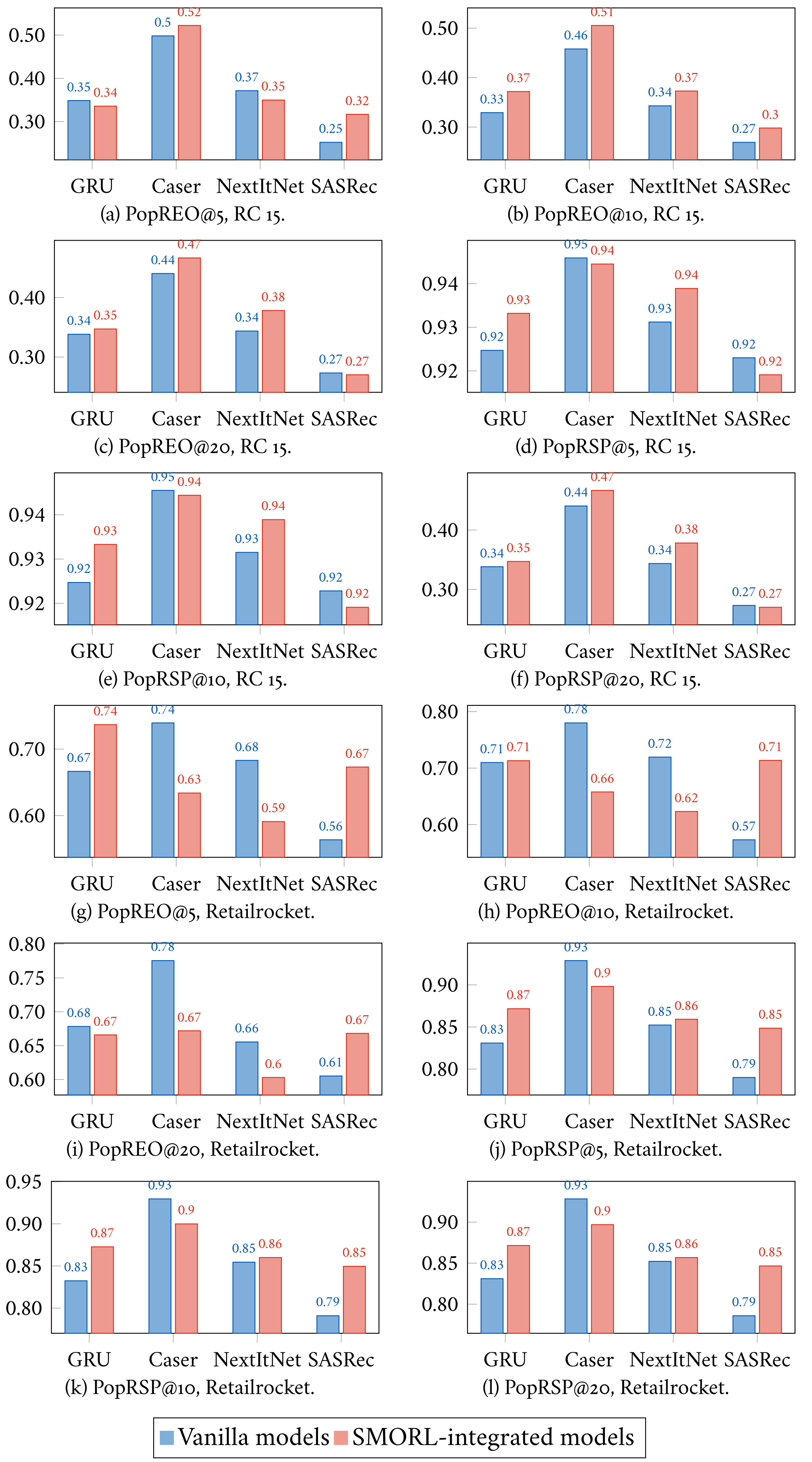
7.6 Summary

In this chapter, we briefly survey the state-of-the-art MORSs, showing that few papers release source codes and datasets to reproduce the work. We reproduce the work by Stamenkovic et al. [168], which proposes SMORL, a Multi-Objective Reinforcement Learning-based algorithm, to produce relevant, diverse, and novel recommendations. Firstly, we replicate the original experiments to highlight challenges in replicating MORSs papers when crucial details are missing. Furthermore, we assess the SMORL's ability to control the importance of each objective. Then, since SMORL is focused only on user-centered objectives, we extend the analysis of the algorithm to shed light on the recommendation biases and the items' exposure of SMORL. Our experiments in several directions led to new observations, summarized below.

Perspective issues in MORSs. The very nature of MORSs leads to perspective issues in selecting the best models. Since we deal with several objectives, more solutions are potential candidates to represent the best model selected with respect to the combination of two or more metrics. Consequently, as a research community, we must explicit the criteria adopted to select the models whose performance is reported in MORSs papers. This improvement is crucial to enhance the reproducibility of the works and a fair comparison among them.

Controlling the objectives is not trivial. MORSs' ability to control the influence of several objectives deserves much attention. Indeed, the positive correlation among some objectives makes their precise control complicated. Therefore, a new research direction is opened regarding the preliminary study on the nature of the objectives we consider. Indeed, the risk is to provide recommendations according to the optimization of several metrics without having control of them.

Recommendations are multi-sided. Despite focusing on user-centered objectives in MORSs, we should be aware of what is happening on the other sides of recommendations. While improving several user-related metrics in MORSs, we may affect other evaluation dimensions, such as algorithmic biases and provider fairness. As in the SMORL case, we may introduce biases in the recommendations or trouble the equality of items' exposure. Therefore, the analysis of dimensions beyond the objectives we are blending in our MORSs is needed.



(a) PopREO@5, RC 15.

(b) PopREO@10, RC 15.

(c) PopREO@20, RC 15.

(d) PopRSP@5, RC 15.

(e) PopRSP@10, RC 15.

(f) PopRSP@20, RC 15.

(g) PopREO@5, Retailrocket.

(h) PopREO@10, Retailrocket.

(i) PopREO@20, Retailrocket.

(j) PopRSP@5, Retailrocket.

(k) PopRSP@10, Retailrocket.

(l) PopRSP@20, Retailrocket.

Figure 7.1. Results on PopREO and PopRSP on RC 15 and Retail Rocket datasets. Lower values are better for both metrics.

Chapter 8

Post-hoc Selection of Pareto-Optimal Solutions in Search and Recommendation

Information Retrieval (IR) and Recommender Systems (RSs) tasks are moving from computing a ranking of final results based on a single metric to multi-objective problems. Solving these problems leads to a set of Pareto-optimal solutions, known as Pareto frontier, in which no objective can be further improved without hurting the others. In principle, all the points on the Pareto frontier are potential candidates to represent the best model selected with respect to the combination of two, or more, metrics. To our knowledge, there are no well-recognized strategies to decide which point should be selected on the frontier in IR and RSs. As shown in the previous chapter, this gap in the literature could lead to perspectives and reproducibility issues on the works proposing information systems with multiple objectives. In this chapter, we propose a novel, post-hoc, theoretically-justified technique, named "Population Distance from Utopia" (PDU), to identify and select the one-best Pareto-optimal solution for search and recommendation systems. PDU considers fine-grained utopia points, and measures how far each point is from its utopia point, allowing to select solutions tailored to user preferences, a novel feature we call "calibration". We compare PDU against state-of-the-art strategies through extensive experiments on tasks from both IR and RS, showing that PDU combined with calibration notably impacts the solution selection. We release codes and datasets at: `https://github.com/sisinflab/Selection-Pareto-Optimal-Solutions-IR-RS`.[1]

[1] This chapter is based on the work published in the *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (CIKM 2023) "Post-hoc Selection of Pareto-Optimal Solutions in Search and Recommendation".

8.1 Introduction

Many tasks in Information Retrieval (IR) and Recommender Systems (RSs) involve the optimization of multiple objective functions. As an example, consider the IR task of *diversifying search results* where, given a user query, we require the IR system to return a list of results that are both *relevant* for the user and *diverse* concerning the possible "facets" of the query [155]. Addressing this task asks for designing a two-objective ranking function comprehensively maximizing both the relevance and the diversity of the result list. The same considerations can be made in RSs. Despite the accuracy of recommendation being considered the gold measure to assess the quality of suggestions, over the last years, RSs have been required to meet other *beyond-accuracy* metrics to avoid obvious [185] and unfair [204] recommendations. Therefore, the choice of a recommendation model and its setting entail several criteria leading to a trade-off among them, resulting in a non-trivial challenge.

Multi-Objective Optimization (MOO) recently attracted several interesting IR and RS contributions [70, 168, 204]. MOO deals with *Pareto optimality,* i.e., the identification of solutions where no objective can be further improved without damaging the others. Pareto-optimal solutions are in turn collected in the so-called *Pareto Frontier,* a set of (possibly infinite) non-dominated solutions.

Existing approaches for MOO can be classified into two categories: i) *heuristic search* and, ii) *scalarization*. In the first category, multi-objective evolutionary algorithms are used to ensure that the emerging solutions are not dominated by each other, even if they can still be dominated by Pareto-optimal solutions not visited by the algorithm [31, 152]. In the second category, scalarization methods aggregate multiple objectives into one objective, possibly guaranteeing Pareto optimality. Scalarization approaches can exploit *model aggregation* techniques combining the output of different models trained on the single objectives. Alternatively, *label aggregation* techniques combine the labels of the training samples a priori, and the optimization is performed using the aggregated labels. Aggregation techniques may involve the setting of the importance or priority of the different objectives by weighting each objective through a scalar function (e.g., Linear Scalarization [124], Weighted Chebyshev [113]). Conversely, some techniques work by constraining the objectives of the problem, e.g., ϵ–Constraint [77] leading to a unique non-dominated solution.

Pareto optimality is commonly achieved by many different Pareto-optimal solutions. However, IR and RS MOO tasks generally require identifying a single Pareto-optimal solution to be deployed in the system. To the best of our knowledge, no strategies specifically tailored to IR and RS tasks have been previously proposed [204]. The state-of-the-art techniques from MOO theory are in fact aimed at identifying a set of Pareto-optimal solutions, without addressing the problem of *post-hoc* choosing one among the—possibly many—solutions identified for the IR and RS tasks. Indeed, many works in the IR and RS literature, although exploiting the techniques discussed above, do not either: (i) consider the problem of selecting a single best solution to the multi-objective problem or, (ii), discuss the criteria adopted to select a single Pareto-optimal solution [224].

In this chapter, we fill this gap by introducing "Population Distance from Utopia" (PDU), a novel post-hoc flexible strategy for selecting **one—best**—Pareto-optimal solution among the ones lying in the Pareto frontier for IR and RS tasks. PDU relies on the observation that the Pareto-optimal point coordinates are an aggregation—usually the mean—of the model performance for each sample, i.e., queries in IR and users in RS, on multiple objectives. PDU exploits the notion of "Utopia point" as the ideal optimization target. Differently from the methods from MOO theory, which are devised to solely consider the mean performance values when selecting a single Pareto-optimal solution, PDU is designed to set a utopia point for each sample of the dataset. This feature allows choosing a solution not only based on the "global" performance achieved by the IR/RS model, but also in a more fine-grained resolution that now considers multiple quality criteria that are expressed on a sample level. We call this feature "calibrated" selection. In detail, the contributions of this work are:

- We formally introduce PDU as a novel technique that allows one to select, in a principled way, the best Pareto-optimal solution previously identified by a state-of-the-art MOO technique.
- We provide a thorough comparison of PDU against state-of-the-art selection strategies. The analysis shows that PDU is the only selection method that allows identifying a "calibrated" solution, i.e., based on ideal targets expressed on a sample level.
- We experimentally compare PDU against state-of-the-art strategies on well-known IR and RS tasks by exploiting public data. The results show that, unlike other methods, PDU can identify Pareto-optimal solutions regardless of their position on the frontier. Moreover, PDU calibration can lead to the selection of significantly different trade-offs.
- We release a GitHub repository for our implementation of PDU and the state-of-the-art competitors as well as the data used in the experiments to allow a full reproducibility of the results.

8.2 Background

8.2.1 Selection Strategies

The Pareto frontier consists of a set of equally optimal solutions. Some methods to select a single Pareto-optimal solution assume the existence of a decision maker [107]. These methods are known as *Multi-Criteria Decision Making* (MCDM) strategies, where a decision-maker has knowledge of the preferences (hierarchy) among the objectives. However, decision-makers do not always know how to weigh the different objectives [30]. Moreover, in some cases, the complexity of the problem makes it difficult for a human decision-maker to evaluate and compare different options comprehensively. Conversely, mathematical methods can provide consistent, objective, and impartial decision-making approaches. In this work, we focus and outline mathematical strategies for selecting a solution from the Pareto frontier, i.e., strategies applicable in the absence of “a priori knowledge” that can feed an MCDM strategy.

Knee Point

The *Knee Point* [30] strategy aims to identify a knee of the Pareto frontier. The rationale is that solutions different from the knee point would exhibit limited improvement for one objective and a substantial deterioration for the others. As described by Branke et al. [30], these strategies were born as a variation of multi-objective evolutionary algorithms to find the knee regions on the Pareto frontier. Consequently, when other algorithms compute the Pareto Frontier, the extracted knee region may not have a knee-featured shape, thus making this strategy less convenient. Several methods to identify the knee point are proposed in the literature, mainly differing for the number of objectives.

Angle-based method (A-KP). When dealing with two objectives, the reflex angle between the slopes of the two vectors through a point $B = (x_i, y_i)$ and its two neighbors, i.e., $A = (x_{i-1}, y_{i-1})$ and $C = (x_{i+1}, y_{i+1})$, on the Pareto Frontier can be considered as an efficient indication of whether the point can be classified as a knee [30]. *The Pareto-optimal point having the maximum reflex angle computed from its neighbors is considered the knee* [54]. If no neighbor to the left (right) is found, a vertical (horizontal) line is used to calculate the angle. Even though this method is efficient in a two-dimensional scenario, it becomes impractical for more than two objectives, especially for the choice of neighbors.

Utility-based method (U-KP). A valid alternative to overcome the limitation of the angle-based method is adopting a marginal utility function. Let us consider a set of n objective functions $f(\cdot)$ and m sets of n uniformly distributed weights w, with $w_i \in [0, 1]$ such that $\sum_i w_i = 1$ [30]. The resulting utility function is then $U(\mathrm{x}, \mathrm{w}) = \sum_i w_i \cdot f_i(x)$. The Pareto-optimal solution having the minimum utility value for most weight configurations is the knee point.

Hypervolume

The *Hypervolume* [230] strategy was first introduced to compare the quality of different Pareto frontiers [66]. However, by computing the hypervolume of each solution on the Pareto frontier, this strategy can be straightforwardly exploited to select the best solution from the set [224]. Given a Pareto-optimal solution $\mathrm{x}^\star \in \mathbb{R}^k$, a reference point $\mathrm{r} \in \mathbb{R}^k$, and the Lebesgue measure λ, the hypervolume $\mathcal{HV}$ of $\mathrm{x}^\star$ with respect to r is:

$$\mathcal{HV} = \lambda(\{\mathrm{x} \in \mathbb{R}^k \mid \mathrm{x}^\star \prec \mathrm{x} \prec \mathrm{r}\}). \tag{8.1}$$

The $\mathcal{HV}$ value is the volume of the hypercube determined by the solution $\mathrm{x}^\star$ and the reference point r. *The Pareto-optimal point having the maximum hypervolume is the selected one.*

Other Techniques

Other simpler techniques that have been used for selecting a solution from the Pareto frontier are the *Euclidean Distance* and the *Weighted Mean* [135, 198]. The Euclidean Distance (*ED*) is computed between each solution on the Pareto frontier and the utopia point: $ED(\mathrm{x}^\star) = |\boldsymbol{f}(\mathrm{x}^\star) - \boldsymbol{f}^\diamond|$. *The Pareto-optimal point having the minimum Euclidean distance is the selected solution.* Instead, the Weighted Mean (*WM*) requires setting the importance of each objective through a set of weights. *Among all the Pareto-optimal points, the point maximizing the weighted mean corresponds to the selected solution.*

8.3 Population Distance from Utopia

Driven by the goal of overcoming the limitations of the other methods in a principled way for IR and RSs, we propose PDU (Population Distance from Utopia), a selection strategy taking into account the distance of the query/user metrics from the utopia point.

Our intuition starts from the observation that in a search and/or recommendation scenario, the Pareto frontier is populated by points representing aggregated results (usually, they represent the average value) on metrics referring to a set of experiments. For instance, in a RS setting, we could have a frontier representing the values of two metrics: *nDCG*, to measure the accuracy of the model, and *Intralist Diversity* (*ID*), to measure the diversity in the list of recommended items. Each point on the frontier may represent the corresponding values of *nDCG* and *ID* for a specific configuration of the hyperparameters. It is worth noticing that these values are computed as the value of the given metric averaged on all the system users. Suppose we focus instead on the point representing the single user. In that case, we may also reconsider the notion of utopia point in this more fine-grained view and adapt it to generalize with respect to the single user. The same observations hold in a search setting where we have queries instead of users. The questions leading our proposal are then:

- *What happens if we focus our analysis on the original points instead of their aggregated representation?*
- *Can we characterize each of these fine-grained points and exploit a generalized definition of utopia point that considers even the single user/query?*

We start by defining a generalized version of the utopia point.

A point $\boldsymbol{f}^\circ$ in the *objective function space* $\mathbb{R}^k$ is a ***generalized utopia point*** if and only if $f_i^\circ = h_i(\mathrm{x}) \,|\, \mathrm{x} \in \mathcal{X} \; \forall i \in \{1, 2, \ldots, k\}$. In our definition, h_i is a function that considers the characteristics of the original data and returns a desired but unattainable utopia value for the i-th metric. For a (non-generalized) utopia point $f^\diamond$, we have $h_i = \min_{\mathrm{x}} f_i(\mathrm{x})$. Its definition can be driven both by system or dataset properties and by the choices of the system designer. For instance, in Section 8.4.1, we define h_2 (see Equation (8.12)) to quantify the users' popularity tendencies stemming from their past interactions with the items in a recommendation scenario.

Given a Pareto-optimal solution $\mathrm{x}^\star \in \mathbb{R}^k$, we can assume that it is the image of an aggregation function applied to a set of m points x_j in $\mathbb{R}^k$, with $j \in \{1, \ldots, m\}$. In our previous example, the points represent the values of the pairs $\langle nDCG, ID \rangle$ (with $k = 2$) for the m users in the system. Suppose a generalized utopia point $\boldsymbol{f}_j^\circ \in \mathbb{R}^k$, with $j \in \{1, \ldots, m\}$, is associated to each point x_j.

Definition 8.1. *The Population Distance from Utopia (PDU) is:*

$$PDU = \log\left(\sum_{j=1}^{m} e(\boldsymbol{f}_j^\circ, \mathrm{x}_j)^2\right), \tag{8.2}$$

where $e : \mathbb{R}^k \longrightarrow \mathbb{R}$ is an error function that satisfies the conditions of identity, symmetry, and triangle inequality. The Pareto-optimal point having the minimum PDU is the selected solution. The error function $e(\cdot)$ is parametric, i.e., we can set any error or distance metric as $e(\cdot)$, like Euclidean distance or mean squared error.

Derivation. Let us consider an objective function space $\mathbb{R}^k$, where k is the number of objectives, and a dataset $\mathcal{D}$ of m samples (users/queries). For each sample, we suppose to know the best possible value of each objective. Then, we can associate each sample with a k-dimensional vector $\boldsymbol{f}_j^\circ$, with $j \in \{1, \ldots, m\}$, which constitutes its generalized utopia point in the objective function space $\mathbb{R}^k$. We use $\mathrm{F} = \{\boldsymbol{f}_j^\circ \mid j \in \{1, \ldots, m\}\}$ to denote the set of all the generalized utopia points referring to the m samples. Let us now consider a model η that returns k objectives performance values for each sample in $\mathcal{D}$. As before, each sample corresponds to a k-dimensional vector x_j, with $j \in \{1, \ldots, m\}$, which represents the model performance for that sample in $\mathbb{R}^k$. We denote $\mathcal{P} = \{\mathrm{x}_j \mid j \in \{1, \ldots, m\}\}$. Thus, each sample j is represented by $\boldsymbol{f}_j^\circ$ and x_j in the objective function space: the closer the points, the better the model η performs. Let us introduce an error function $e : \mathbb{R}^k \longrightarrow \mathbb{R}$ satisfying the conditions of identity, symmetry, and triangle inequality. The error of the model η on the j-th sample is $e(\boldsymbol{f}_j^\circ, \mathrm{x}_j)$. By supposing the error term follows the IID property, it has a Gaussian distribution with mean $\mu = 0$ and variance σ^2, i.e., $e(\boldsymbol{f}_j^\circ, \mathrm{x}_j) \sim \mathcal{N}(0, \sigma^2)$, whose probability density function is:

$$p(e(\boldsymbol{f}_j^\circ, \mathrm{x}_j)) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e(\boldsymbol{f}_j^\circ, \mathrm{x}_j)^2}{2\sigma^2}\right). \tag{8.3}$$

We can note that if $\boldsymbol{f}_j^\circ$ and x_j are close, the exponent part of Equation (8.3) tends to 1, and the probability increases while tending to 0 when the two points are far apart and the probability decreases.

Then, we compute the error probability density function of the error for the entire dataset $\mathcal{D}$. We observe that the model η has some parameters Θ. Hence, $\mathcal{P}$ can be expressed as a function g of the parameters Θ: $\mathcal{P} = g(\Theta)$. Then, a vector

$\mathrm{x}_j \in \mathcal{P}$ can be rewritten as $\mathrm{x}_j = g(\Theta)_j$. By assuming the samples to be independent, we obtain the following expression for the likelihood function:

$$p(e(\mathrm{F}, g(\Theta))) = \prod_{j=1}^{m} p(e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)). \tag{8.4}$$

Since $\boldsymbol{f}_j^{\circ}$ is the (generally unattainable) output we desire to have, we are interested in finding the parameters Θ for the model η such that the likelihood function $p(e(\mathrm{F}, g(\Theta)))$ is the highest. As the logarithmic function is increasing monotone, it does not modify the maximum positions. Hence, we can compute the log-likelihood instead of the likelihood to simplify calculations:

$$\log p(e(\boldsymbol{f}_j^{\circ}, g(\Theta))) = \log \prod_{j=1}^{m} p(e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)) \tag{8.5}$$

$$= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2. \tag{8.6}$$

At this point, we explicit the variance term σ^2. Since we have supposed that the error term $e(\boldsymbol{f}_j^{\circ}, \mathrm{x}_j)$ has a Gaussian distribution with $\mu = 0$, the variance σ^2 is defined as $\frac{\sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2}{m}$. By introducing this term in Equation (8.6), we obtain that the log-likelihood is:

$$\begin{aligned}\log p(e(\boldsymbol{f}_j^{\circ}, g(\Theta))) = m \log & \frac{1}{\sqrt{2\pi}\sqrt{\frac{1}{m}\sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2}} \\ & - \frac{1}{\frac{2}{m}\sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2} \sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2\end{aligned} \tag{8.7}$$

$$= -m \log(\sqrt{2\pi}) + m \log m - \frac{1}{2} \log \left(\sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2 \right) - \frac{m}{2}. \tag{8.8}$$

By supposing to train the model η on the same dataset $\mathcal{D}$ with several configurations of Θ, the terms depending on the dataset size m and the constant $1/2$ in Equation (8.8) can be removed as they are constant when choosing the highest log-likelihood. Hence, the only variable quantity among the different log-likelihoods is:

$$-\log \left(\sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2 \right). \tag{8.9}$$

Therefore, we are looking for the model η with parameters Θ having the maximum value of the term in Equation (8.9):

$$\max \left[-\log \left(\sum_{j=1}^{m} e(\boldsymbol{f}_j^{\circ}, g(\Theta)_j)^2 \right) \right]. \tag{8.10}$$

Finally, this remainder term can be easily rewritten with a positive sign as long as we choose the configuration of Θ for the model η having the minimum value for this quantity:

$$\min\left[\log\left(\sum_{j=1}^{m} e(f_j^{\circ}, g(\Theta)_j)^2\right)\right] = \min\left[\log\left(\sum_{j=1}^{m} e(f_j^{\circ}, \mathrm{x}_j)^2\right)\right]. \tag{8.11}$$

□

8.3.1 Calibrated PDU

PDU allows setting a generalized utopia point for each sample of the dataset, i.e., queries and users in an IR or RS scenario, respectively. This feature allows choosing a solution not only based on the "global" performance achieved by the IR/RS model, but also in a more fine-grained resolution that now considers multiple quality criteria expressed on a sample level. We call such feature ***calibration*** since it can be usefully exploited in specific scenarios, e.g., personalization in RS, where it is possible to define generalized utopia points according to individual users' preferences. These generalized utopia points can be fixed apriori, e.g., they can be identified by the system designer or computed through functions that numerically quantify the users' tendencies, similarly to what has been done in previous works regarding *calibrated recommendations* [97, 137, 170]. We refer to this feature as *Calibrated*-PDU (C-PDU).

8.3.2 Feature Comparison

In Section 8.2.1, we have presented the most-used techniques to choose a single best solution belonging to a Pareto frontier. However, as also stated by Wu et al. [204], there is no consensus on the strategy to solve this task in the IR and RS communities. Not surprisingly, all methods have some advantages and limitations, leading to a lack of an ideal strategy [109]. Hence, a comparison of the features provided by PDU and state-of-the-art techniques is needed. Specifically, we identify some desirable features the techniques should have. Table 8.1 discusses the main properties of PDU and other state-of-the-art techniques. First, **the strategy should be suitable even when dealing with more than two objectives**. In this regard, the angle-based knee point is the only ineffective method. Second, **the strategy should not need any additional knowledge**. Most techniques require additional problem information, i.e., the reference point ($\mathcal{HV}$), the (generalized) utopia point (*ED*, PDU), and a weights set (*WM*). Since the results of a given strategy can largely depend on such information, a fair strategy should require as less additional information as possible. The weights should be set by a decision-maker with deep knowledge of the hierarchy among the objectives. In contrast, the reference and the (generalized) utopia points are ordinarily intrinsic to the problem. Despite some common practices (e.g., nadir point) [109], it has been shown that determining a reference point r for $\mathcal{HV}$ is generally more challenging [79, 109], and a badly defined reference point

Table 8.1. Overview of the properties of PDU and other selection strategies. The symbols ✓ (✗,—) indicate that the method has (does not have, could not have) the specified property.

Method	*A-KP*	*U-KP*	$\mathcal{HV}$	*ED*	*WM*	PDU
Suitable With >2 Objectives	✗	✓	✓	✓	✓	✓
No Additional Knowledge	✓	✓	r	$f^\diamond$	w	f°
No Scaling before Calculation	✓	—	✓	—	—	—
Deterministic	✓	✗	✓	✓	✓	✓
Equal Treatment of PF Regions	✗	✓	✗	✓	✓	✓
Calibration	✗	✗	✗	✗	✗	✓

can lead to inconsistent evaluation results [102]. Indeed, having a reference point slightly different from the nadir point could lead to incongruous evaluation, as experimentally demonstrated by Ishibuchi et al. [88]. Therefore, the utopia point is the most effortlessly additional information that can be exploited for this task. Third, **the strategy should not require to scale the range of the objectives**. Scaling may be needed for strategies whose calculation involves objective blending, i.e. *U-KP, ED, WM,* and PDU. When the objectives have different scales, the bigger the range of an objective, the bigger its contribution to the selection of a solution. However, the choice of scaling the objectives is left to the system designer. Fourth, **the strategy should be deterministic**. The *U-KP* strategy requires randomly extracting a set of weights from a uniform distribution. This could potentially affect the consistency and reproducibility of results. Fifth, **the strategy should equally promote the solutions despite their position on the Pareto frontier**. The strategies blending the objectives are not biased to select solutions based on particular Pareto frontier regions. This is not true for the $\mathcal{HV}$ strategy that tends to promote the solutions on the concave region of a Pareto frontier.

Final Observations and Calibration

To summarize, none of the strategies own all the properties. However, some considerations can be made. *A-KP* and *U-KP* are characterized by huge drawbacks. The former can be utilized only in contexts considering two objectives. The latter is nondeterministic. Furthermore, none of the techniques is able to select a solution irrespective of its position on the Pareto frontier and to be independent of scaling the objective ranges before calculation simultaneously. In this regard, a system designer could prefer to adopt a technique able to fairly choose a solution despite its position on the Pareto frontier (as done by *U-KP, ED, WM,* and PDU). Indeed, scaling the objectives can be easily performed with a simple operation such as min/max normalization. Furthermore, this operation is subject to the system designer, who can consider the objectives range in specific applications. Concerning the additional knowledge problem, only *A-KP* and *U-KP* do not need supplementary information. However, as stated before, they are characterized by main drawbacks. Then, such additional knowledge is required. Among the remainder techniques, PDU and *ED* exploit easier-to-define additional material, i.e., the utopia point.

By looking beyond, the proposed PDU allows us to define a utopia point for each sample in the dataset. While the other approaches exploit only aggregated models' performance, PDU opens to a novel "calibrated" way to select one—best Pareto-optimal solution tailored to individual sample characteristics. To the best of our knowledge, this is the first attempt to introduce this kind of feature in the task of Pareto-optimal solutions selection strategy.

From now on, when no confusion arises, we will use *utopia point* to refer also to a *generalized utopia point*.

8.4 Experimental Evaluation

We now present an experimental evaluation based on public data that aims at answering the following research questions:

RQ1: How do PDU and other state-of-the-art selection strategies behave w.r.t. the discussed properties? (see Section 8.3.2)

RQ2: How does the distribution of the points composing the points on the Pareto frontier influence the selection of a solution?

RQ3: How does the calibration feature impact the selection of a solution?

8.4.1 Experimental Scenarios

Driven by the observation that, in IR and RS settings, the Pareto frontier is populated by points representing aggregated results, we analyze the selection strategies in these two settings.

Information Retrieval Scenario

Concerning the IR scenario, we focus on an ad-hoc search task by dealing with the efficiency / effectiveness / energy-consumption trade-off of query processing in IR systems based on machine-learned ranking models [33]. IR systems heavily exploit supervised techniques for learning document ranking models that are both effective and efficient, i.e., able to retrieve within a limited time budget high-quality documents relevant to users' queries. State-of-the-art learning-to-rank models include ensembles of regression trees trained with gradient boosting algorithms, e.g., LambdaMART [33, 209], and deep neural networks, e.g., NeuralNDCG [142]. Since ranking is a complex task and the training datasets are large, the learned models are complex and computationally expensive at inference time. The tight constraints on query response time thus require suitable solutions to provide an optimal trade-off between efficiency and ranking quality [40, 72, 116].

In this scenario, we use the LambdaMART [33, 209] implementation available in LightGBM [100] to train ranking models based on ensembles of regression trees and Neural Networks (NN) trained in Pytorch [138] following the optimization methodology proposed in [132]. The models are trained on `MSN30K` [143], a public and widely-used dataset for learning to rank. The evaluation employs 11 LambdaMART and 5 Neural Networks ranking models, each tested on the 6,306 queries of the `MSN30K` test set. We measure the ranking quality of each model in terms of average nDCG@10 (f_1), and average ranking time (seconds per document) (f_2). For the LambdaMART configurations, we also measure the average energy consumption (Joules per document) (f_3). The average ranking time of each model has been measured by using QuickScorer [116], while energy consumption has been measured by using the Mammut library [159]. Efficiency experiments are performed on a dedicated Intel Xeon CPU E5-2630 v3 clocked at 2.4 GHz in single-thread execution. QuickScorer is compiled using GCC 9.2.1 with the `-O3` option.

In this IR experimental scenario, we focus on selecting the best efficiency/effectiveness trade-off for query processing.

Recommendation Scenario

Concerning the RS scenario, we consider two of the main problems of recommendation algorithms, i.e., the accuracy of the recommendations and the tendency to over-suggest popular items. Often, the ability of RS to provide accurate recommendations is competing with the capability of including long-tail items in such suggestions [131], inducing a trade-off. Hence, we consider two objectives. We compute the Recall@10 (f_1) to measure the accuracy of suggestions and the average percentage of items in the long-tail (APLT) [3] (f_2) to measure to what extent a RS can recommend unpopular items (see Section 2.4).

Specifically, we interpret APLT from two perspectives, identifying two experimental scenarios. On the one hand, we assess APLT from provider-side fairness. The provider side fairness can be quantified as the models' ability to expose items to users evenly [1, 3, 204]. Indeed, the over-recommendation of popular items, i.e., the so-called unfairness of popularity bias, may be felt as unfair by providers who get long-tail items under-represented in the suggestions. Hence, in this scenario, the goal is to choose a model that promotes relevant items without affecting niche items' visibility.

In this first RS experimental scenario, we focus on selecting the best recommendation model dealing with multiple objectives.

On the other hand, we evaluate APLT from the final user point of view. Indeed, certain users may prefer to consume popular items, while others niche items. Consequently, exclusively recommending mainstream items would hurt the experience of long-tail users, and vice versa. The approach of calibrated recommendation has shown a valuable solution toward this direction of research [137, 170]. A recommendation list is calibrated concerning popularity when the set of items it covers matches the user's profile in terms of item popularity [4]. Inspired by the concept of popularity-based calibrated recommendation, for each user, we compute the values of the APLT target (f_2) stemming from their popularity profile. To this end, we compute the user-level APLT utopia values using the *weighted combination of mean and standard deviation* method described by Jugovac et al. [97]. We consider the set of users $\mathcal{U}$, the set of items $\mathcal{I}$, and the mean number of transactions T in the training set. For each item $i \in \mathcal{I}$, we assess its popularity pop_i by counting the number of transactions the item is involved in. For each user $u \in \mathcal{U}$, we define the set $\Gamma_u = \{pop_i \mid u \text{ interacted with } i\}$. We quantify the user u popularity tendencies as $pop_u = \alpha \cdot \mu(\Gamma_u) + \beta \cdot \sigma(\Gamma_u)$, where α and β are set to a fixed value of 1 as done in [97], $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean and standard deviation operators, respectively. The higher is pop_u, the most user u has consumed mainstream items in her past interactions. Finally, we normalize pop_u and compute the APLT utopia value for each user:

$$f_2^\circ = h_2(u) = \frac{pop_\Psi - pop_u}{pop_\Psi - pop_\Phi}, \tag{8.12}$$

where Φ and Ψ are the sets composed by pop_i values such that i is one of the less and most T consumed items, respectively. With this normalization, the higher is f_2°, the less popular is the user profile.

In this second RS experimental scenario, we show how important a calibrated technique is for choosing the best recommendation model dealing with multiple objectives.

In the two experimental scenarios presented for RS, we exploit the EASE^R recommendation model [171], which works like a shallow autoencoder. This model is characterized by a single hyper-parameter to tune, i.e., the L2-norm regularization (λ). Nevertheless, it has been shown that it often outperforms other state-of-the-art recommender systems [13]. Specifically, we explore the hyper-parameter λ by training 48 configurations on the book-domain dataset *Goodreads* [196] (18,892 users, 25,475 items, and 1,378,033 transactions) and on the music-domain dataset *Amazon Music* [13] (14,354 users, 10,027 items, and 145,523 transactions). We split the datasets following the 70-10-20 hold-out strategy. Thus, the evaluation of this scenario employs 48 solutions on the objective function space, each tested on the remaining users of the test set (18,070 of *Goodreads*, and 14,354 of *Amazon Music*).

8.4.2 Experimental Methodology

The different hyperparameter configurations introduced before, for the two IR and RS settings, generate solutions in the objectives function space for each specific experimental scenario. Once the Pareto-optimal solutions that compose the Pareto frontier are identified, we select one by applying PDU and the other selection strategies we analyzed in this work. The selected solutions are then analyzed according to the features introduced in Section 8.3.2. Moreover, we investigate in detail how the formulation of PDU and its calibration feature influence the choice of the one—best solution by looking at the distribution of points composing that solution. We refer to the reference point and the utopia point with r and $\boldsymbol{f}^\circ$, respectively. Furthermore, we use the Euclidean distance as $e(\cdot)$ in the formulation of PDU, to have an immediate comparison with *ED* to assess the impact of the points distribution composing a solution. Tables 8.2, 8.3, and 8.4 report the results for the solutions chosen by at least one strategy. For the sake of completeness, the reader may find the complete sets of results in the GitHub repository. The best values for each metric are in bold, while the arrows indicate if better stands for lower ↓ or higher ↑ values.

Experimental settings for the IR scenario

A nadir point cannot be established for the IR scenario because two of the objectives, i.e., efficiency and energy consumption, are not bounded in the opposite direction of the optimization target. For this reason, we define the reference point by slightly worsening the worst values reached by the optimal solutions available. By doing so, we end up setting $\mathrm{r} = (0.5, 0.00002, 0.001)$ for $\mathcal{HV}$. Moreover, we set $\boldsymbol{f}^{\circ} = (1, 0, 0)$ for *ED*, and for each sample in the dataset in PDU. For what regards *WM*, we equally treat the objectives by setting each weight to 0.5. Finally, in this scenario, we do not apply any normalization to the objective values achieved with the different models.

Experimental settings for the RS scenario

Differently from the IR scenario, a nadir point can be established here because the two objectives under consideration, i.e., Recall and APLT, are bounded. We thus set $\mathrm{r} = (0, 0)$ for $\mathcal{HV}$, and $\boldsymbol{f}^{\circ} = (1, 1)$ for *ED*. As before, we give equal importance to the objectives in *WM* by setting each weight to 0.5. Concerning PDU, we set $f_1^{\circ} = 1$ for each sample utopia point as we want all users to have accurate recommendations. Instead, we set $f_2^{\circ} = 1$ in the first RS experimental scenario, while we compute specific values of f_2° for each user as in Equation (8.12) in the second RS experimental scenario. Finally, in both RS scenarios, we apply a min-max normalization to the objectives.

We first divide the results discussion according to both IR and RS scenarios for RQ1 and RQ2. Then, we answer RQ3 by exploiting the second RS scenario.

8.4.3 Performance Comparison (RQ1)

IR scenario

We answer RQ1 by first focusing on the IR scenario. The results for this scenario are summarized in Tables 8.2 (LambdaMART) and 8.3 (Neural Networks). The plots in Figures 8.1a and 8.1c show the Pareto-optimal points selected by the different techniques for the cases considering two and three objectives regarding the LambdaMART models, respectively. Figure 8.1b shows the points selected in the case of the Neural Networks models.

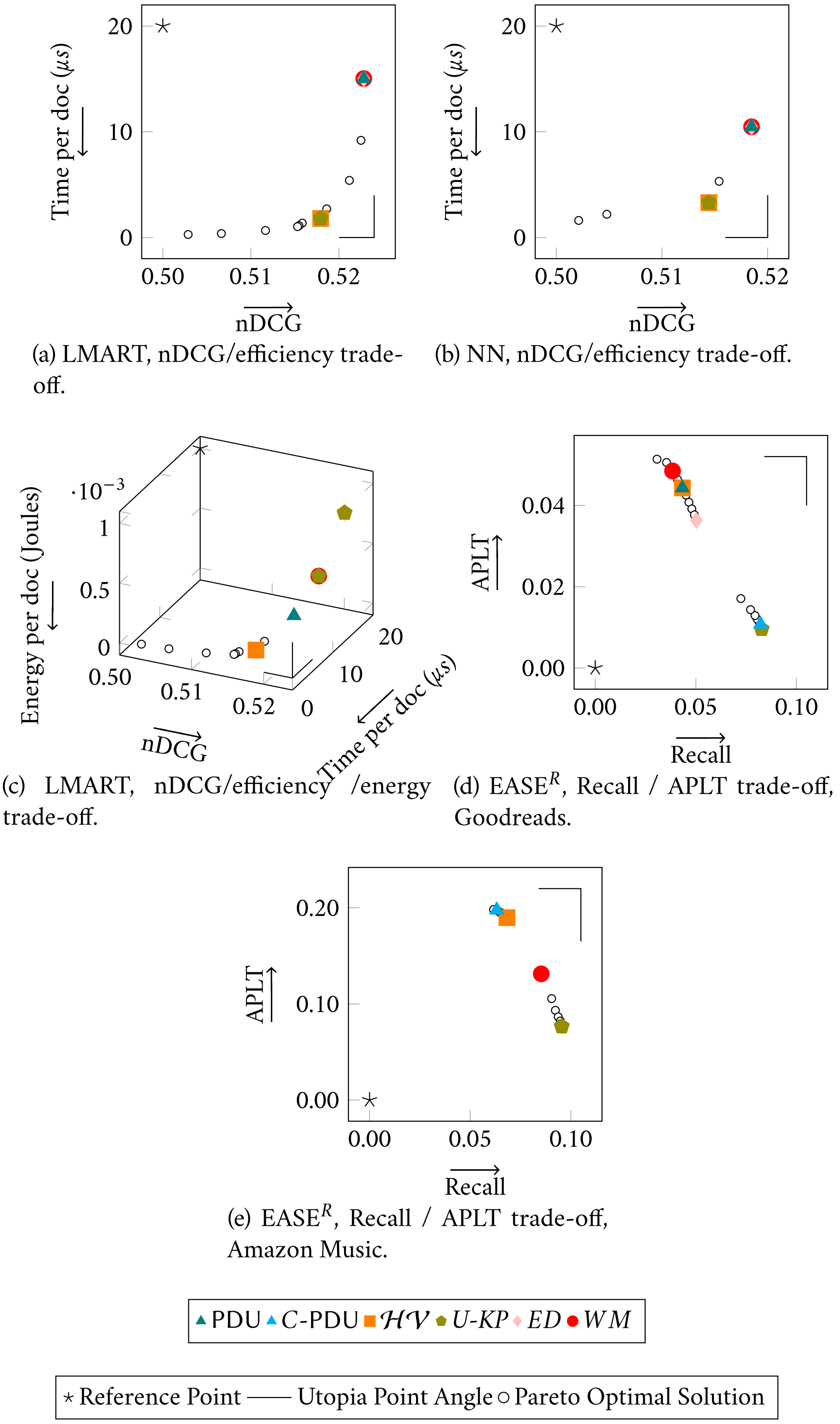
Regarding the two-objective case, we observe that the methods blending the objectives (PDU, *ED*, *WM*) select the same Pareto-optimal solution lying on the boundary of the Pareto frontier for both families of models, thus maximizing the accuracy at the cost of efficiency. In contrast, $\mathcal{HV}$ chooses an inner solution of

Table 8.2. LambdaMART selected solutions for the IR scenario. The objectives are accuracy (*nDCG*), efficiency (*Seconds*), and energy consumption (*Joules*).

Models		**Objectives**			**Selection Strategies**									
					Accuracy / Efficiency					Accuracy / Efficiency / Energy				
Trees	Leaves	nDCG $\uparrow$	Seconds $\downarrow$	Joules $\downarrow$	PDU $\downarrow$	$\mathcal{HV}\uparrow$	*U-KP*$\uparrow$	*ED* $\downarrow$	*WM* $\uparrow$	PDU $\downarrow$	$\mathcal{HV}\uparrow$	*U* $\uparrow$	*ED* $\downarrow$	*WM* $\uparrow$
300	32	0.5179	18.0544×10^{-5}	10.8515×10^{-5}	7.4953	3.2612×10^{-7}	**107**	0.4821	0.1295	7.4960	2.9236×10^{-10}	85	0.4821	0.0863
300	64	0.5212	54.0393×10^{-5}	31.7795×10^{-5}	7.4837	3.0924×10^{-7}	102	0.4788	0.1303	**7.4904**	2.1097×10^{-10}	93	0.4788	0.0868
500	64	0.5225	91.9204×10^{-5}	54.5946×10^{-5}	7.4799	2.4323×10^{-7}	103	0.4775	0.1306	7.4996	1.1044×10^{-10}	**102**	0.4775	**0.0870**
878	64	**0.5228**	150.355×10^{-5}	89.4260×10^{-5}	**7.4768**	1.1328×10^{-7}	98	**0.4772**	**0.1307**	7.5289	0.1198×10^{-10}	**102**	**0.4772**	0.0870

Table 8.3. Neural Networks selected solutions in the IR scenario. The objectives are accuracy (*nDCG*) and efficiency (*Seconds*).

Models				**Objectives**		**Selection Strategies**				
L1	L2	L3	L4	nDCG $\uparrow$	Seconds $\downarrow$	PDU $\downarrow$	$\mathcal{HV}\uparrow$	*U-KP*$\uparrow$	*ED* $\downarrow$	*WM* $\uparrow$
100	50	50	10	0.5144	3.3003×10^{-6}	7.5069	2.4099×10^{-7}	**221**	0.4856	0.1286
200	100	100	50	**0.5185**	1.0476×10^{-5}	**7.4959**	1.7598×10^{-7}	204	**0.4815**	**0.1296**



(a) LMART, nDCG/efficiency trade-off.

(b) NN, nDCG/efficiency trade-off.

(c) LMART, nDCG/efficiency /energy trade-off.

(d) EASE^R, Recall / APLT trade-off, Goodreads.

(e) EASE^R, Recall / APLT trade-off, Amazon Music.

Figure 8.1. Pareto-optimal solutions for the IR and RS scenarios. The colored shapes represent the best—Pareto-optimal—point selected by the strategies under evaluation.

the Pareto frontier in both cases, i.e., more efficient models, that however show a significantly lower performance in terms of nDCG compared to the selection provided by PDU (0.5225 vs. 0.5179 for LambdaMART, and 0.5185 vs. 0.5144 for the Neural Network). It is worth noting that, in this case, no transformation has been applied to the scale of the objectives, and the values of the Pareto solutions for what regards the efficiency scale lead the points to be closer to the utopia value $f_2^\circ = 0$. If a min/max normalization is applied to the objective, PDU still selects the same solution. Another essential implication arising from this analysis is that, in this scenario, we cannot establish the nadir point, making challenging the definition of the reference point. Consequently, this potentially leads to different results based on how we define the reference point. Indeed, as we push the reference point away from the Pareto frontier, $\mathcal{HV}$ selects a boundary solution, as done by PDU. In light of the above results, we observe that if the information related to the nadir point is unavailable, the definition of the reference point can strongly affect the selection of the final solution. Moreover, if the reference point is estimated by looking at the collection of the considered solutions, i.e., by slightly increasing the worst values reached by them, $\mathcal{HV}$ promotes the solution in the middle. Indeed, the definition of the reference point in such a way makes the volume of those solutions, computed as in Equation (8.1), higher than any other. Thus, $\mathcal{HV}$ unequally considers the remaining points lying on the boundaries of the Pareto frontier. Finally, it is worth highlighting that *U-KP*, although reported in Figures 8.1a and 8.1b, is not deterministic. Indeed, by executing this method several times, it may choose different points as the weights of the utility function (see Section 8.2.1) are randomly extracted from a uniform distribution.

Moving to the three-objective formulation of the IR scenario for the LambdaMART models, Figure 8.1c shows that when introducing the energy consumption objective, the methods tend to choose a more efficient model than the one selected in the two-objectives scenario. As before, PDU and *ED* tend to maximize the accuracy with respect to $\mathcal{HV}$ that still select solutions in the middle. The three-dimensional scenario confirms two behaviors observed in the two-dimensional one. First, the solution selected by $\mathcal{HV}$ depends on the chosen reference point since it is not possible to define a nadir point. Second, *U-KP* still exhibits a non-deterministic behavior.

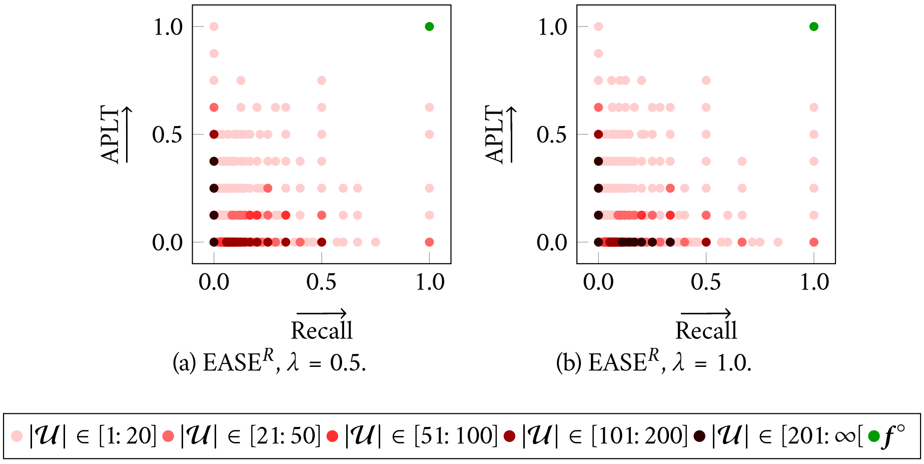
Finally, we claim that PDU and *ED* perform the most convenient selection from a qualitative perspective. By looking at Tables 8.2 and 8.3, we see that they choose the models with higher values of nDCG for all IR cases. Indeed, both efficiency and energy consumption objectives are closer to their respective utopia values. This means that more complex models, chosen by PDU and *ED*, guarantee considerable improvement in ranking accuracy at a small reduction of efficiency and energy consumption. Conversely, $\mathcal{HV}$ chooses models that exhibit a considerable decrease in terms of nDCG.

Table 8.4. EASE^R selected solutions (for Goodreads and Amazon Music) in the RS scenario with *Recall* and *APLT* objectives. For *APLT*, the higher the better refers to the provider side.

Models	**Objectives**		**Selection Strategies**					
λ	Recall ↑	APLT ↑*	PDU ↓	C-PDU ↓	$\mathcal{HV}$ ↑	*U-KP*↑	*ED* ↓	*WM* ↑
				Goodreads				
0.3	0.0384	0.0485	10.4113	10.0898	0.1861×10^{-2}	55	0.8546	**0.2699**
0.5	0.0433	0.0443	**10.4066**	10.0829	0.1919×10^{-2}	16	0.7761	0.2686
1	0.0503	0.0363	10.4098	10.0819	0.1826×10^{-2}	0	**0.7191**	0.2546
60	0.0822	0.0108	10.4126	**10.0706**	0.0885×10^{-2}	86	0.9651	0.2556
90	0.0827	0.0096	10.4134	10.0711	0.0791×10^{-2}	**101**	0.9938	0.2510
				Amazon Music				
0.3	0.0632	0.1976	**10.0104**	**9.8604**	0.1249×10^{-1}	79	0.9524	0.2608
1	0.0683	0.1898	10.0147	9.8628	0.1295×10^{-1}	49	0.8074	0.2819
10	0.0853	0.1313	10.0784	9.9160	0.1120×10^{-1}	4	**0.6177**	**0.2896**
80	0.0955	0.0766	10.1268	9.9570	0.0731×10^{-1}	**89**	0.9780	0.2542

RS scenario

For the first RS experimental scenario, we report the results achieved in Table 8.4 for the Goodreads dataset (Figure 8.1d) and for the Amazon Music dataset (Figure 8.1e). For both datasets, we notice that two well-separated clusters characterize the Pareto frontier. On the one hand, in Goodreads the EASE^R configurations with lower L2 norm (λ) values, which belong to the top-center cluster, account for the accommodation of the objectives. In contrast, the second cluster (bottom-right), i.e., λ between 10 and 100 in Table 8.4, maximizes Recall at the expense of the exposure of the items (lower values of APLT). On the other hand, in Amazon Music, these two clusters of configurations follow the opposite behavior. On the one side, the configurations with λ between 0.2 and 1 maximize APLT at the detriment of Recall (top-left cluster). On the other side, the remaining configurations do not promote either Recall or APLT (bottom-right cluster). In this scenario, $\mathcal{HV}$ suffers less from the problem of promoting solutions in the center of the Pareto frontier. Indeed, differently from the IR scenario, here it is possible to define the nadir point as a reference point because we know the lowest bounds (0 for both APLT and Recall). Consequently, even though $\mathcal{HV}$ selects an inner solution in the Goodreads case, it chooses a point that tends to maximize APLT for the Amazon Music dataset. PDU follows the behaviour of $\mathcal{HV}$ when selecting the solutions for both datasets. By considering that it selects an outer point of the Pareto frontier in the IR scenario, this endorses the ability of PDU to equally promote the available solutions despite their positioning on the Pareto frontier. *WM* and *ED* select a solution belonging to the top-center cluster in Goodreads and to the bottom-right cluster in Amazon Music, thus enhancing the trade-off between accuracy measured in terms of Recall and items exposure in both cases. Finally, *U-KP* still exhibits a nondeterministic performance.



(a) EASE^R, λ = 0.5.

(b) EASE^R, λ = 1.0.

Figure 8.2. Distribution of users data points in the objective function space *Recall* / *APLT* for the solutions selected by PDU (left) and *ED* (right). The color of the point indicates the number of users in the point.

To answer RQ1 we conclude observing that PDU overcomes some limitations of $\mathcal{HV}$ and U-KP competitors. Indeed, PDU selects one—best—Pareto-optimal solution regardless of its position on the Pareto frontier in a deterministic way. Moreover, it exploits the concept of Utopia point as additional information. Such a concept is more convenient to use than the reference point used in $\mathcal{HV}$, since, depending on the problem addressed, the nadir point is difficult to be defined.

8.4.4 Impact of the Points Distribution (RQ2)

We now answer RQ2 by investigating the impact on selecting the distribution of the points that compose a solution on the Pareto frontier. Indeed, PDU is the only strategy considering these points in a more fine-grained resolution. This analysis is done on both the IR (Tables 8.2 and 8.3) and RS (Table 8.4) scenarios. To this end, we remember that we have set $e(\cdot)$ as the Euclidean Distance in the formulation of PDU (Equation (8.2)). Hence, even if both PDU and *ED* rely on the Euclidean distance, they work differently in the two experimental scenarios. This observation provides insights on the impact of the points distribution on the selection.

RS scenario

PDU and *ED* choose different solutions for both RS datasets. In this regard, the user data points' distribution in the objective function space plays a crucial role, as visually depicted by Figure 8.2 for the Goodreads dataset. Indeed, the distribution of the solution with λ = 0.5, chosen by PDU, shows that more points are oriented to the Utopia point than the ones of the solution selected by *ED*. To confirm this fact, we compute the users points' mean Euclidean distances to the utopia point of both solutions. Results confirm that the EASE^R configuration selected by PDU has a lower value of average Euclidean distance, i.e., 1.3498 for $\lambda = 0.5$, w.r.t. the configuration chosen by *ED*, i.e., 1.352 for $\lambda = 1$. The same impact is observed regarding the Amazon Music dataset. Here, PDU and *ED* select different configuration models having $\lambda = 0.3$ and $\lambda = 10$, respectively. As before, the EASE^R configuration selected by PDU ($\lambda = 0.3$) has a lower value of average Euclidean distance, i.e., 1.2361 than the configuration chosen by *ED* ($\lambda = 10$), i.e., 1.279.

IR scenario

Concerning the IR two-objectives cases, PDU and *ED* choose the same solution for both LambdaMART and Neural Networks models. When introducing energy consumption as the third objective for the LambdaMART models, *ED* still selects the same configuration with 878 trees and 64 leaves. Conversely, PDU chooses a more efficient model (300 trees and 64 leaves). Once more, the query points' mean Euclidean distances to the common utopia point of the model selected by PDU are lower than the ones of the model chosen by *ED* (0.4813 vs. 0.4945).

To conclude, the answer to RQ2 is that the distribution of the points composing a solution with respect to a common utopia point has a significant impact on the final selection. This is an important fact, as it paves the way to defining selection strategies that take the distribution of the points into account while performing a selection that can be done in a more—fine-grained—sample-level way.

8.4.5 Impact of Calibration on the Selection (RQ3)

Finally, we analyze the impact of the calibration introduced for PDU using the second RS scenario, where we aim to tailor the selection according to the users' item popularity tastes. To this end, we assess the selection performed by Calibrated-PDU (C-PDU).

Starting from the Amazon Music dataset, the average of the APLT utopia values computed with Equation (8.12) (0.83) reveals that the dataset's users generally prefer less popular items. Indeed, C-PDU selects the EASE^R model with $\lambda = 0.3$. This solution lies on the top-left cluster of Figure 8.1e, by maximizing APLT with a loss of Recall. In this case, C-PDU behaves similarly to PDU and $\mathcal{HV}$. Moving to the Goodreads dataset, it is characterized by users with more mainstream tastes, since the average of the APLT utopia values is equal to 0.65. Surprisingly, C-PDU is the only strategy among the ones tested selecting a model configuration belonging to the bottom-right cluster in Figure 8.1d where the solutions achieve higher accuracy values without promoting APLT and following the mainstream users tastes — along with *U-KP* that, however, has a non-deterministic behavior. These experimental results already qualitatively show the impact of defining a utopia point for each user on the final selection, since C-PDU is the only strategy to capture the users' popularity profiles for both datasets. We deepen the analysis further by considering the model configurations chosen by PDU and C-PDU for Goodreads, i.e., $\lambda = 0.5$ and $\lambda = 60$, respectively. We observe that, although the model with $\lambda = 0.5$ performs better on average APLT, the model with $\lambda = 60$ has a lower variance of the mean absolute error (0.036 for $\lambda = 60$ vs. 0.039 for $\lambda = 0.5$) between the utopia values and the model performance values for each user. This indicates that C-PDU selects the model that generally follows better the users' popularity profile. In addition, this model provides more accurate recommendations on average. Hence, C-PDU chooses the model that performs better in terms of accuracy and also tailors the popular tastes of the users.

To conclude, the answer to RQ3 is that the calibration feature of PDU allows dealing with the ideal targets for each sample. This confirms that calibration is a viable way to move the selection of the Pareto-optimal solution to a more fine-grained resolution that can lead to significantly different choices in terms of the trade-off selected.

8.5 Summary

In this work, we proposed PDU, a novel, theoretically-justified *post-hoc* technique to select one—best—Pareto-optimal solution among the ones lying in the Pareto frontier in search and recommendation scenarios. To our knowledge, PDU is the only selection technique in the literature that can be "calibrated", i.e., it can choose the best Pareto-optimal solution based on ideal targets expressed on single queries or users. We comprehensively compared the properties of PDU with those of competitor techniques. We conducted an extensive experimental evaluation focusing on both IR and RS scenarios, showing that the formulation and the calibration feature of PDU have a notable impact on the solution's selection. In the future, we will explore PDU by exploiting other distance metrics (e.g., Chebyshev and Manhattan). Moreover, it could be interesting to perform online A/B tests to assess the impact of the calibrated selection. Finally, this work could open to the formulation of a new loss function based on the PDU derivation, to directly train a ranking model on multiple objectives simultaneously.

Chapter 9

Flex-MORe: A Flexible Multi-Objective Recommendation Framework

Conventional Recommender Systems (RS) deliver users a ranked list of relevant items based on their historical preferences. Nevertheless, most approaches overlook other aspects of a recommendation process. Beyond-accuracy metrics—e.g., those focusing on novelty, diversity, and fairness of recommendation—are not usually a core part of the optimization process behind the training of the recommendation model. Multi-Objective Recommender Systems (MORSs) aim to consider beyond-accuracy objectives in their training process. However, current state-of-the-art works fail to provide a general framework that can encompass any kind of objective in the training procedure. This chapter introduces a Flexible Multi-Objective Recommendation framework (Flex-MORe). The framework extends RS training by incorporating an objective-agnostic and objective scale-aware additional loss function term, blending different—conflicting—metrics guided by their deviation from the corresponding utopia point. Since most ranking-based metrics are inherently non-differentiable, we present a method that makes them suitable for use with back-propagation. Experiments on three real-world datasets show the ability of Flex-MORe to balance diverse objectives, achieving state-of-the-art performance. We release code and datasets at: `https://github.com/vincpapa/Flex-MORe`.[1]

[1] This chapter is based on the work "Flex-MORe: A Flexible Multi-Objective Recommendation Framework", currently under review.

9.1 Introduction

In recent years, Recommender Systems (RSs) have become crucial parts of online platforms, offering personalized recommendations based on users' preferences [37, 153]. However, conventional approaches focus on accuracy metrics to generate ranked lists of relevant items, often neglecting other important aspects. On the one hand, as these systems shape user experience, there is a pressing need to extend their capabilities beyond accuracy, e.g., to consider the diversity and novelty of suggestions [68, 168]. Indeed, accuracy-centric RS can lead to user satisfaction erosion, the creation of filter bubbles, and the perpetuation of biases in historical data [96, 133]. On the other hand, these limitations become evident in marketplaces and platforms that cater to diverse stakeholders with conflicting objectives [1, 165], such as producer and consumer interests. Some approaches solely focus on consumer [111] or producer-sided [70] fairness, while others try to accomplish both user and provider fairness [147, 205]. Toward these increasing needs of satisfying multiple facets of recommendation, Multi-Objective Recommender Systems (MORSs) [89, 224] are gaining attention. MORSs are built by considering multiple objectives through Multi-Objective Optimization (MOO) [161]. Ideally, the process of MOO leads to optimal solutions adhering to the Pareto optimality principle, where no objective can improve without compromising others [30]. Strategies for optimizing multiple objectives can be broadly categorized into *heuristic search* [31, 152]. and *scalarization* [124]. Between the two, the most commonly adopted MOO strategy is the latter [94, 114, 205].

Scalarization methods (linearly) combine several objectives by weighting them through scalars. When adopting this approach, researchers have to face two main challenges: (i) how the weights of the linear function should be chosen depends on the application scenario or business needs. Therefore, these weights should be set according to a decision-maker's knowledge [107] or through automatic routines [56, 94]; (ii) the blended objectives must be carefully designed to make them differentiable and suitable to train an effective model with back-propagation. In this work, we focus on the latter challenge. This challenge is even harder in the recommendation domain, where the metrics that possibly measure the achievement of the objectives imply the ranking of the items into a recommendation list. Unfortunately, the item's ranking employs the sorting operation, which is non-differentiable [205]. Consequently, the requirement of designing differentiable functions led to the recent strides in MORSs [70, 168, 205, 224]. However, even though many works address quite effectively specific scenarios [70, 168, 205], often they lack generalization. Indeed, they often overlook the scale of the devised functions, with the risk of having a dominating objective – i.e., the one with the largest scale of values. To further illustrate these limitations, we present an experiment in Section 9.2 using a recent state-of-the-art MORS framework, MultiFR [205], showing that altering the scale of one objective can significantly affect the model's performance, resulting in entirely different outcomes. Overcoming these limitations requires the development of more flexible frameworks that can seamlessly accommodate diverse recommendation scenarios without the need for manual metric adjustments.

To address the abovementioned limitations, we present a Flexible Multi-Objective Recommendation framework, named Flex-MORe. In this framework, we propose a novel smoothing general approach to make ranking metrics that apply a cutoff differentiable and suitable for back-propagation routines. Then, the framework encompasses a general multi-objective loss function term that, adopting the scalarization strategy, can be incorporated into the training procedure of any Bayesian Pairwise Ranking-based recommendation algorithm. The loss function is designed to measure the discrepancies between the chosen metrics and their corresponding utopia points. Contrarily to the previous work, each objective's error is also normalized to avoid the problem of dominating objectives due to different scales. To summarize, the novel contributions are:

- We introduce Flex-MORe, a novel multi-objective-agnostic framework for recommendation. Flex-MORe provides a strategy to make any metric suitable for backpropagation and combines multiple metrics to create a normalized loss function.
- We evaluate Flex-MORe in a marketplace scenario, where balancing provider interests (measured by APLT) and user needs (represented by nDCG) is crucial. Flex-MORe demonstrates its efficacy in improving the beyond-accuracy performance and achieves state-of-the-art performance compared to other MORSs.
- We show that Flex-MORe can control the objectives according to the scalarization weights, avoiding undesired and unexpected behaviors. Furthermore, we demonstrate that normalizing the objective related errors achieves a balance among the objectives. Finally, we study the training efficiency of our method.

9.2 Motivating Example

This section highlights the importance of carefully assessing the scale of the loss functions in a scalarization-based Multi-Objective Recommender System. We conduct an experiment on MultiFR [205][2]. This framework integrates Bayesian Pairwise Ranking (BPR) [149] loss function alongside group-based consumer and provider fairness loss functions. These functions are linearly combined adopting the scalarization technique, computing the weights assigned to each objective through the Multi-Gradient Descent Algorithm. In MultiFR, the BPR loss is designed to sum the components related to the users into a batch. However, it is commonly recognized that for batch data, averaging the loss rather than summing it ensures consistency in the loss scale, regardless of batch size. This common practice[3] [18, 178, 202] promotes more stable and manageable training [73]. In the experiment, we trained MultiFR with NGCF [200] on two datasets (Facebook Books and Amazon Music) under two

[2] More details on the experimental settings will be provided in Section 9.5.2

[3] This approach is evident when inspecting the source codes.

Table 9.1. Performance comparison between MultiFR w/s and MultiFR w/a as motivating example.

Model	nDCG ↑	RSP ↓	MAD ↓	nDCG ↑	RSP ↓	MAD ↓
	Facebook Books			**Amazon Music**		
MultiFR w/s	0.0913	0.9340	0.0069	0.0562	0.7950	0.0089
MultiFR w/a	0.0096	0.3231	0.0041	0.0059	0.2590	0.0042

different settings: (i) summing the components of the BPR loss (MultiFR w/s) and (ii) averaging its components (MultiFR w/a) within the batch. Interestingly, Table 9.1 shows that MultiFR's performance varies significantly with this modification. Indeed, the accuracy performance (measured with nDCG) of MultiFR w/a completely drops compared to MultiFR w/s, benefiting the fairness-oriented performance (assessed with RSP [228] and MAD [55]). As a result, averaging the BPR loss function reduces the scale of its values, thereby lowering its gradient intensity during training.

Overall, experiments show that the scales of the objectives play a crucial role in the final recommendation performance in a multi-objective setup. Hence, this highlights the need for a recommendation framework considering the scale of any potential objective metric.

9.3 Related Work

Over a decade ago, Rodríguez et al. [154] laid the groundwork for multi-objective optimization for recommender systems. In the subsequent years, numerous research efforts have aimed to integrate diverse objectives. For a comprehensive overview of these studies, readers can refer to [89, 161, 224].

Re-ranking-based Approaches. A straightforward approach is to re-rank the suggestions to accommodate other objectives. Li et al. [111] introduced a user-fairness-focused re-ranking strategy, aiming to ensure fair recommendations for different user groups according to their activity level. Rahmani et al. [147] replicate the study, finding that the user-focused re-ranking strategy negatively affects the popularity bias. Following this direction, Naghiaei et al. [131] implement a re-ranking strategy addressing both user and provider fairness.

Scalarization and Heuristic Search Approaches. Two common techniques are scalarization and heuristic search. Ribeiro et al. [152] proposed the first heuristic search method for recommendation. They introduced a Pareto-efficient hybrid approach that combines various recommendation models — each excelling in different aspects like precision, novelty, and diversity — using a weighted sum to determine the optimal weights through a multi-objective evolutionary algorithm. Among the works belonging to the scalarization category, Lin et al. [114] proposed differentiable formulations for the Gross Merchandise Value and Click-Through Rate (CTR) objectives, coordinating these objectives using weighted aggregation. Recently, Wu et al. [205] proposed MultiFR, a multi-objective optimization framework for fairness-aware recommendations, which adaptively balances accuracy and fairness for various stakeholders. They employ the Karush–Kuhn–Tucker conditions [158] to blend the objectives in a scalarization function to gather a single Pareto optimal solution through the Multi-Gradient Descent Algorithm (MGDA) [56]. Conversely, Carmel et al. [41] propose Stochastic Label Aggregation, that performs label aggregation by randomly selecting a label per training example according to a given label distribution.

Reinforcement Learning-based Approaches. An emerging approach utilizes Multi-Objective Reinforcement Learning (MORL) to pursue several objectives simultaneously. Ge et al. [70] propose a fairness-aware multi-objective reinforcement learning approach, optimizing CTR and item exposure as signals for relavance and fairness. Conversely, Stamenkovic et al. [168] introduce SMORL, a Scalarized MORL framework, to simultaneously achieve accuracy, diversity, and novelty in session-based recommender systems.

The literature shows that existing scalarization approaches typically focus on specific objectives that the authors aim to achieve. As a result, the constraints, loss functions, and rewards are closely tied to these objectives. In contrast, this study explores a more generalized, scale-aware approach to simultaneously optimizing multiple objectives.

9.4 The proposed framework

This section introduces the Flexible Multi-Objective Recommendation framework (Flex-MORe). Flex-MORe is flexible approach that considers multiple objectives while holistically training a model. It combines a multi-objective and a recommendation backbone loss function together. The loss function can include any metric beyond accuracy, making the approach versatile and adaptable to different scenarios due to the normalization of loss scales. Additionally, we propose a general smoothing approach to make ranking-based metrics differentiable.

9.4.1 Multi-objective Loss

The Flex-MORe loss function is designed to optimize multiple recommendation metrics simultaneously. Imagine a perfect scenario where every important metric reaches its ideal value. This ideal state is called the "utopia point." Flex-MORe tries to bring the recommendation system performance as close to this utopia point as possible. To do this, we calculate the squared difference between the actual metric values and their ideal counterparts for each user. However, these errors can assume different magnitudes for several reasons: (i) the scales of the metrics can vary in range, and (ii) it could be easier to reach higher performance for some metrics than others, thus having lower errors for them. To achieve this, we define the Flex-MORe loss function as follows.

Definition 9.1 (Flex-MORe Loss Function)**.** *Given a set of m differentiable metrics*

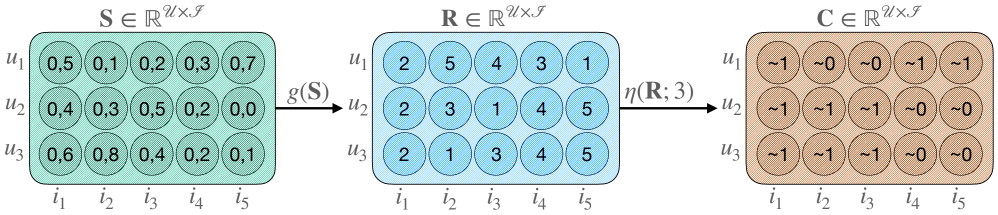

$\mathbf{S} \in \mathbb{R}^{\mathcal{U}\times\mathcal{J}}$
$\mathbf{R} \in \mathbb{R}^{\mathcal{U}\times\mathcal{J}}$
$\mathbf{C} \in \mathbb{R}^{\mathcal{U}\times\mathcal{J}}$
u_1 0,5 0,1 0,2 0,3 0,7
u_2 0,4 0,3 0,5 0,2 0,0
u_3 0,6 0,8 0,4 0,2 0,1
$g(\mathbf{S})$
u_1 2 5 4 3 1
u_2 2 3 1 4 5
u_3 2 1 3 4 5
$\eta(\mathbf{R}; 3)$
u_1 ~1 ~0 ~0 ~1 ~1
u_2 ~1 ~1 ~1 ~0 ~0
u_3 ~1 ~1 ~1 ~0 ~0
i_1 i_2 i_3 i_4 i_5

Figure 9.1. The steps followed by our proposed smoothing approach to obtain the matrices $R \in \mathbb{R}^{\mathcal{U}\times\mathcal{I}}$ and $C \in \mathbb{R}^{\mathcal{U}\times\mathcal{I}}$ to compute approximated differentiable ranking-based metrics, for $k = 3$.

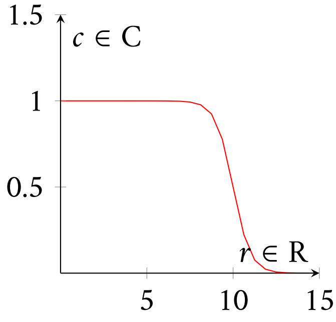


Figure 9.2. The plot of the function in Eq. (9.2) for $k = 10$. The function returns 1 (or close to) if r is less than k, 0 otherwise.

with cutoff k and a set $\mathcal{U}$ of users, let $\mathrm{x}_u \in \mathbb{R}^m$ be the m metric performance values of user $u \in \mathcal{U}$. Let the metrics' utopia point be $\boldsymbol{f}^\circ \in \mathbb{R}^m$. The multi-objective loss function is

$$\mathcal{L}_{\textit{Flex-MORe}} = \arg\min_{\Theta} \frac{1}{|\mathcal{U}|} \sum_{\mu=1}^{m} \omega_\mu \sum_{u=1}^{\mathcal{U}} \sigma\left(\zeta\left(\left(f_\mu^\circ - x_{\mu_u}(\Theta)@k\right)^2\right)\right), \tag{9.1}$$

where x_{μ_u} represents the performance value of the differentiable approximated metric μ for the user u that depends on the model's parameters Θ, and f_μ° is the ideal value for metric μ. ω_μ is the weight associated to metric μ. Finally, $\sigma(\cdot)$ and $\zeta(\cdot)$ are the sigmoid and z-score normalization functions, respectively.

The z-score normalization function $\zeta(\cdot)$ is used to normalize the squared errors regardless of their scale. To prevent negative values, the sigmoid function $\sigma(\cdot)$ is applied to the normalized errors.

9.4.2 *Differentiable Approximation of the Metrics*

The formulation of $\mathcal{L}_{\text{Flex-MORe}}$ in Eq. (9.1) leans on computing m metric values for each user. Generally, the metrics employed in a recommendation scenario are required to calculate the items ranking for each specific user, which is inherently a non-differentiable operation. Many practical approaches have been proposed to make the ranking and sorting operations differentiable [27, 128]. Given an array of scores, these methods typically compute another array containing the score's approximated rankings (or their sorting) of all scores. However, recommendation metrics generally apply a cutoff, i.e., the top-k recommended items solely contribute to their calculus. Therefore, such metrics not only ask for the ranking computation of the items but also require to know if a ranked item is within the top-k recommended items, e.g., nDCG@k or APLT@k. In literature, some methods are specifically designed to account for this requirement for accuracy metrics like nDCG@k and Precision@k [32, 142]. However, we need a general strategy to gather the top-k items once the approximated rankings have been calculated to apply it to any cutoff and ranking-based metric. To this end, after leveraging on any method to compute differentiable rankings, we propose a differentiable smoothing approach to retrieve the top-k items and compute a differentiable approximation of any ranking-based metrics with cutoff, making them suitable for use with back-propagation. Figure 9.1 shows the steps involved in the approach.

Firstly, we gather the score matrix $\mathrm{S} \in \mathbb{R}^{\mathcal{U} \times \mathcal{I}}$ predicted by the recommendation model, where $s_{u,i}$ is the predicted relevance score between the user u and the item i. Then, we can use any continuous approximator of the ranking function [128, 144], that we denote with $g(\cdot)$, to compute the differentiable approximated item positions in each user's recommendation list. Formally, we obtain the matrix of rankings $\mathrm{R} = g(\mathrm{S})$, where $\mathrm{R} \in \mathbb{R}^{\mathcal{U} \times \mathcal{I}}$, and $r_{u,i}$ is the approximated predicted ranking position of the item i for the user u. In this way, we make the computation of the rankings differentiable. Now, we need a differentiable method to retrieve the top-k recommended items for each user given the approximated ranking matrix R. To this end, we devise a differentiable function $\eta(\mathrm{R}; k)$ obtaining the matrix $\mathrm{C} \in \mathbb{R}^{\mathcal{U} \times \mathcal{I}}$:

$$\mathrm{C} = \eta(\mathrm{R}; k) = \frac{\tanh(-\mathrm{R} + k) + 1}{2} \tag{9.2}$$

where k determines the top-k recommended items, and $c_{u,i} \in \mathrm{C}$ is 1 (or close to) if the approximated ranking $r_{u,i}$ is less than k, 0 otherwise. Figure 9.2 plots the function in Eq. (9.2) for $k = 10$. Thus, matrices R and C suffice to calculate the differentiable approximation of any ranking-based metric and to compute Eq. (9.1).

9.4.3 Model Training with Flex-MORe loss

In section 9.4.2, we explained how the Flex-MORe loss function term could encompass approximated differentiable metrics and be suitable for back-propagation. We can now integrate $\mathcal{L}_{\text{Flex-MORe}}$ into a recommendation backbone. We adopt the Bayesian Pairwise Ranking (BPR) [149] as the recommendation backbone's loss function. BPR formulates the recommendation problem as a pairwise ranking task, where the goal is to ensure that, for each user $u \in \mathcal{U}$, an observed (positive) item $i^+ \in \mathcal{S}^+$ is ranked higher than an unobserved (negative) item $i^- \in \mathcal{S}^- := \mathcal{I} \setminus \mathcal{S}^+$. Hence, built a training dataset $\mathcal{D} := \{(u, i^+, i^-) \mid i^+ \in \mathcal{S}_u^+ \wedge i^- \in \mathcal{S}_u^-\}$, we define the

Algorithm 1: Training Procedure with Flex-MORe

```
Inputs :$m \leftarrow$ number of objectives
        $\mu_1, \ldots, \mu_m \leftarrow$ the metrics to incorporate
        $g(\cdot) \leftarrow$ continuous approximator
        $model \leftarrow$ recommendation backbone
        $\omega_{\text{BPR}}, \omega_{\mu_1}, \ldots, \omega_{\mu_m} \leftarrow$ weights in Eq. (9.3)
        $n_{epoch} \leftarrow$ number of epochs
        $n_{batch} \leftarrow$ number of batches
Output:$\Theta$
Random initialization: $\Theta$                    // Model parameters
for $metric \in \{\mu_1, \ldots, \mu_m\}$ do
    construct $\mathcal{L}_{\text{Flex-MORe}}(\Theta)$ based on $metric$ and $\omega_{\text{metric}}$
for $epoch \in \{1, \ldots, n_{epoch}\}$ do
    for $batch \in \{1, \ldots, n_{batch}\}$ do
        forward propagation depending on $model$
        compute score matrix $\mathbf{S}$
        compute differentiable ranking appr. $\mathrm{R} = g(\mathrm{S})$
        compute differentiable ranking cutoff $\mathrm{C} = \eta(\mathrm{R}; k)$
        for $metric \in \{\mu_1, \ldots, \mu_m\}$ do
            foreach $user$ in $batch$ do
                select C[$user$] as the row in C corresponding to $user$
                compute $metric$ based on C[$user$]
            update $\mathcal{L}_{\text{Flex-MORe}}(\Theta)$ based on $metric$ and $\omega_{\text{metric}}$
        construct $\mathcal{L}(\Theta) = \omega_{\text{BPR}}\mathcal{L}_{\text{BPR}}(\Theta) + \mathcal{L}_{\text{Flex-MORe}}(\Theta)$
        compute $\nabla_{\Theta}\mathcal{L}(\Theta)$
        update $\Theta$                            // Back-propagation
```

recommendation backbone loss function as:

$$\mathcal{L}_{\text{BPR}} = \arg\min_{\Theta} \frac{1}{|\mathcal{D}|} \sum_{(u,i^+,i^-)\in\mathcal{D}} - \ln \sigma(s_{ui^+}(\Theta) - s_{ui^-}(\Theta)) + \lambda_{\Theta}||\Theta||_2^2,$$

where $\sigma(\cdot)$ denotes the sigmoid function, λ_{Θ} is the regularization term, s_{ui^+} and s_{ui^-} represent the predicted scores for the positive and negative items, respectively, and Θ are the model's parameters. Finally, we scalarize [53, 205] $\mathcal{L}_{\text{BPR}}$ and $\mathcal{L}_{\text{Flex-MORe}}$:

$$\mathcal{L} = \omega_{\text{BPR}}\mathcal{L}_{\text{BPR}} + \mathcal{L}_{\text{Flex-MORe}}, \tag{9.3}$$

where ω_{BPR} is the weight that controls the intensity of the model's loss function gradient. This scalar, along with the scalars ω_{μ} in Eq. (9.1), can be manually set — expressing the knowledge of a multi-criteria decision maker [107] — or computed through some MOO methods such as Multi-Gradient Descent [56, 158] or EPO Search [118]. The overall training procedure is shown in Algorithm 1.

9.5 Evaluation

We evaluate the proposed Flex-MORe framework by answering the following research questions:

RQ1: Can we adjust the importance of different goals in Flex-MORe by changing the values of weights in Eq. (9.1) and (9.3)?

RQ2: Does adding Flex-MORe to the recommendation system improve its performance beyond just accuracy?

RQ3: How does our Flex-MORe system compare to other similar systems that focus on fairness?

RQ4: Does our choice of combining sigmoid and z-score normalization functions in Flex-MORe effectively balance the various goals of the system?

RQ5: How does the training overhead of Flex-MORe scale when optimizing multiple objectives in various dataset sizes?

9.5.1 Experimental Scenario

Flex-MORe allows designers to incorporate desired metrics into a general loss function. To illustrate, we present a case study and identify the metrics to include in the Flex-MORe loss function to represent our objectives.

The Recommendation Scenario

Recently, there has been a growing interest in the multi-stakeholder fairness [70, 111, 131, 205]. Indeed, nearly every online platform works as a marketplace that links consumers with service/item providers, positioning them as the main stakeholders involved in the recommendation process. While fairness can be defined in various ways, a commonly adopted scenario is as follows. From the consumers' perspective, fairness is concerned with evenly achieving effective performance across users on the accuracy side [111, 131]. Meanwhile, item provider fairness focuses mainly on the even exposure of different item categories, such as mainstream and niche items [70, 205]. This experimental evaluation considers the above-described recommendation scenario, focusing on consumer and provider fairness.

Provider Fairness Metric

On the provider fairness side, we partition the items into popular and unpopular groups according to the Pareto distribution [4]. The items constituting 80% of transactions in a dataset are identified as short-head (popular items), while the remaining items are categorized as unpopular (long-tail). We inject into $\mathcal{L}_{\text{Flex-MORe}}$ the Average Percentage of Long-Tail metric (APLT) [3]. This metric measures the presence of long-tail items in the recommendation lists. Pushing this metric towards its utopia point, i.e., 1, means promoting the exposure of niche items for each user. This choice is reasonable since RSs generally suffer from popularity bias. This often results in greater exposure for popular items [3, 4].

Consumer Fairness Metric

On the consumer fairness side, we measure the effectiveness of the suggestions with nDCG. Here, we deal with the *individual* consumer fairness, where ideally, each user receives recommendations with the same relevance quality [210]. We aim to optimize the relevance of recommendations towards an ideal nDCG of 1 for each user in the Flex-MORe loss function.

Flex-MORe loss function

Given the scenario described above, we define the specific Flex-MORe loss function as follows. As assumed in Section 9.1, how to set or compute the weights of the scalarization approach is out of this chapter scope. Without loss of generality, we set $\omega_\mu = 1 - \omega_{\text{BPR}}$ for each metric μ in Eq. (9.1). Hence, the Flex-MORe loss function for this specific setup is:

$$\mathcal{L}_{\text{Flex-MORe}} = \arg\min_{\Theta} \frac{1}{|\mathcal{U}|} \Bigg((1 - \omega_{\text{BPR}}) \sum_{u=1}^{\mathcal{U}} \sigma\left(\zeta\left((1 - \text{APLT}_u(\Theta)@k)^2\right)\right) + (1 - \omega_{\text{BPR}}) \sum_{u=1}^{\mathcal{U}} \sigma\left(\zeta\left((1 - \text{nDCG}_u(\Theta)@k)^2\right)\right) \Bigg).$$

In this setup, we accommodate three objectives within the Flex-MORe framework: consumer and provider fairness through $\mathcal{L}_{\text{Flex-MORe}}$, and recommendation relevance through $\mathcal{L}_{\text{BPR}}$ (Eq. (9.3)). We point out that this scenario serves only as a showcase of the Flex-MORe performance. In fact, all the metrics used in the above equation can be substituted to consider other criteria.

9.5.2 Experimental Settings

To answer our research questions, we first introduce the experimental settings, i.e., the datasets, the baselines considered for comparison, and the evaluation metrics.

Datasets

We use the following three public datasets from various domains having a different number of users, items, and transactions: *Amazon Baby* [134], *Facebook Books*[4], and *Amazon Music* [134]. Amazon Baby is an e-commerce dataset from the Amazon catalog that contains 5,842 users, producing 35,475 feedbacks on 7,925 items. Facebook Books

[4] https://2015.eswc-conferences.org/important-dates/call-RecSys.html

is a dataset in the book domain. It collects 18,978 implicit feedback from 1,398 users for 2,933 items. Amazon Music is a music domain dataset containing 145,523 transactions performed by 14,354 users over a catalog of 10,027 items.

Baselines

To measure the effectiveness of Flex-MORe, we compare its performance with two well-known recommendation models:

- **BPRMF** [149]. Bayesian Personalized Ranking is one of the most widely used factorization models for pair-wise ranking.
- **NGCF** [200]. Neural Graph Collaborative Filtering propagates embeddings on the user-item interaction graph, capturing non-linear relationships between users and items.

In addition, identifying appropriate multi-objective-based baselines for our work poses several challenges since differing interpretations and assumptions of fairness are assessed in the literature [148]. Nonetheless, we include two state-of-the-art fairness-oriented multi-objective-based frameworks:

- **MultiFR** [205]. It is a fairness-aware recommendation framework based on multi-objective optimization that leverages the application of the Multi-Gradient Descent Algorithm [56]. This model jointly optimizes accuracy and fairness for consumers and producers. Since it adopts a classic recommendation model as a backbone, we consider two versions of MultiFR, specifically **BPRMF-MultiFR** and **NGCF-MultiFR**.
- **CPFair** [131]. It is a re-ranking strategy taking into account consumer (C)- and provider (P)-side fairness constraints. We consider **BPRMF-CPFair** and **NGCF-CPFair**.

These frameworks promote group-based consumer and provider fairness. We utilize both these baselines by dividing the users into active and inactive groups and the items into popular and unpopular groups according to the Pareto distribution.

Reproducibility Details

We implement the baselines using *Pytorch* with Adam Optimizer. For all of the models, we fix the embedding size to 64. For NGCF, we set the number of layers equal to 3. The learning rate is fine-tuned in the range {0.005, 0.001, 0.0005}, and the regularization term in {0.01, 0.005, 0.001}. In MultiFR, the patience parameter is varied in {0.5, 0.75}, while the smooth temperature is set to 0.00001 to guarantee better performance of SmoothRank. Regarding CPFair, we explored a wide range of λ_1 and λ_2 values, but did not observe substantial differences in performance. Finally, for Flex-MORe, the *k* value in Eq. (9.2) used for computing the approximated differentiable metrics is set to 20. We examine several values of ω_{BPR} in the set {0.25, 0.5, 0.75, 0.95}. As the continuous approximator $g(\cdot)$, we adopt the Adaptive Implicit Likelihood Estimation (AIMLE) method proposed by Minervini et al. [128]. We train all the baselines with a batch size of 2,048 for the three datasets, except for MultiFR-NGCF on Amazon Music, where we chose a batch size of 1,024 due to GPU memory constraints.

Evaluation Protocol

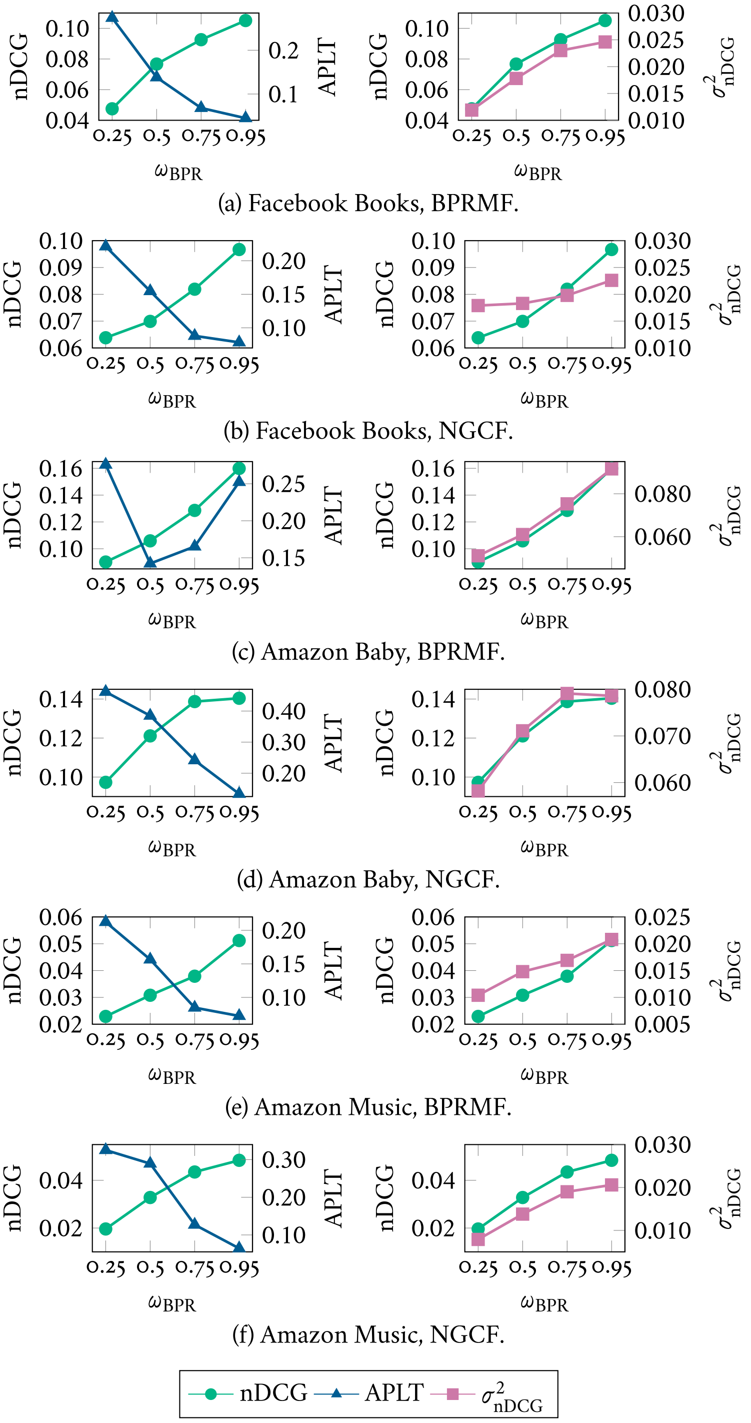
We treat the datasets in an implicit feedback setting and split them following a 70-10-20 hold-out strategy. We evaluate the performance on the validation set every ten epochs. Given the recent insights about the selection of the best iteration of multi-objective recommendation models [188], we explicitly choose the best iteration of the training process according to the value of nDCG@20 on the validation set to follow the standard evaluation practices in RecSys community [20]. We then report the performance achieved on the test set by the model obtained on that iteration.

Evaluation Metrics

We select several metrics to assess the performance of Flex-MORe and baselines. We measure the accuracy of the recommendations with nDCG and Recall. For the consumer-fairness evaluation, we measure the variance of the consumer's recommendation quality measured by nDCG (σ^2_{nDCG}) [210]. Lower variance indicates fairer recommendation results. Conversely, we utilize APLT [3] to measure to what extent unpopular items are inserted into the recommendation lists. Higher APLT values indicate that more long-tail items are included in the recommendation lists. To enhance a fair comparison with the multi-objective-based baselines, we employ the following metrics to assess the group-based provider and consumer fairness of the models:

- **RSP** [228], i.e., "Ranking-based Statistical Parity". Based on statistical parity, this metric measures the extent to which the ranking probability distributions of different item categories c_i, with $i \in \{1, \dots, n\}$, are the same. Lower values of RSP indicate that the recommendation is less biased. The items are partitioned into short-head and long-tail (see Section 9.5.1).
- **MAD** [55], i.e., "Mean Absolute Deviation," computes the equity of the ranking quality measured in terms of nDCG among two user groups, active and inactive. Lower MAD values suggest higher consumer fairness.

Furthermore, we also evaluate the diversity of recommendations with the Item Coverage (**IC**) and the **Gini** index. Item Coverage quantifies the number of unique items that appear in the top-k recommendations across all users. The Gini index is a statistical measure of dispersion used to quantify a distribution's inequality. We report the values of 1 - Gini [75], with higher values corresponding to greater diversification. Finally, we employ a multi-objective evaluation metric to measure



(a) Facebook Books, BPRMF.

(b) Facebook Books, NGCF.

(c) Amazon Baby, BPRMF.

(d) Amazon Baby, NGCF.

(e) Amazon Music, BPRMF.

(f) Amazon Music, NGCF.

Figure 9.3. Performance of Flex-MORe on Amazon Baby, Facebook Books, and Amazon Music by varying the weight ω_{BPR}. All metrics are assessed with cutoff 20 (i.e., Metric@20). The greater the value of ω_{BPR}, the greater is the backbone loss' influence.

how a model simultaneously performs on two metrics. In detail, we calculate the hypervolume (HV) [230] by considering nDCG and APLT as objectives.

9.5.3 Controlling the Loss Functions (RQ1)

This section aims to assess to what extent Flex-MORe allows controlling the influence of each loss function in Eq. (9.3). We recall that, without loss of generality, we set $\omega_{\mu} = 1 - \omega_{\mathrm{BPR}}$ for each objective integrated into Eq. (9.1). Hence, by varying the value ω_{BPR}, we expect a lower or higher influence of each objective into $\mathcal{L}_{\text{Flex-MORe}}$ on the final result. Lowering the value of ω_{BPR} increases the influence of $\mathcal{L}_{\text{Flex-MORe}}$, leading to better system performance in terms of the considered objectives (i.e., beyond-accuracy metrics in this scenario). Table 9.2 and Figure 9.3 report the results for $\omega_{\mathrm{BPR}} \in \{0.25, 0.5, 0.75, 0.95\}$ given the different models and datasets introduced in Section 9.5.2. It is important to note that controlling the objectives can be challenging due to their complex interrelationships. Indeed, they could depend on the positive or negative correlation between a couple of objectives. Except for one out of twelve case combinations (Figure 9.3c, see discussed limitation in section 9.5.4), Flex-MORe can fetch adequate control of the objectives. As expected, decreasing the value of ω_{BPR} generally brings benefits to both provider (with APLT, the higher, the better) and consumer (with σ^2_{nDCG}, the lower, the better) fairness at the expense of the relevance of suggestions on the three datasets with both BPRMF and NGCF as backbones. Conversely, higher values of ω_{BPR} increase the gradient intensity of the BPR loss function, resulting in more accurate recommendations, as indicated by the higher nDCG and Recall values (see Table 9.2).

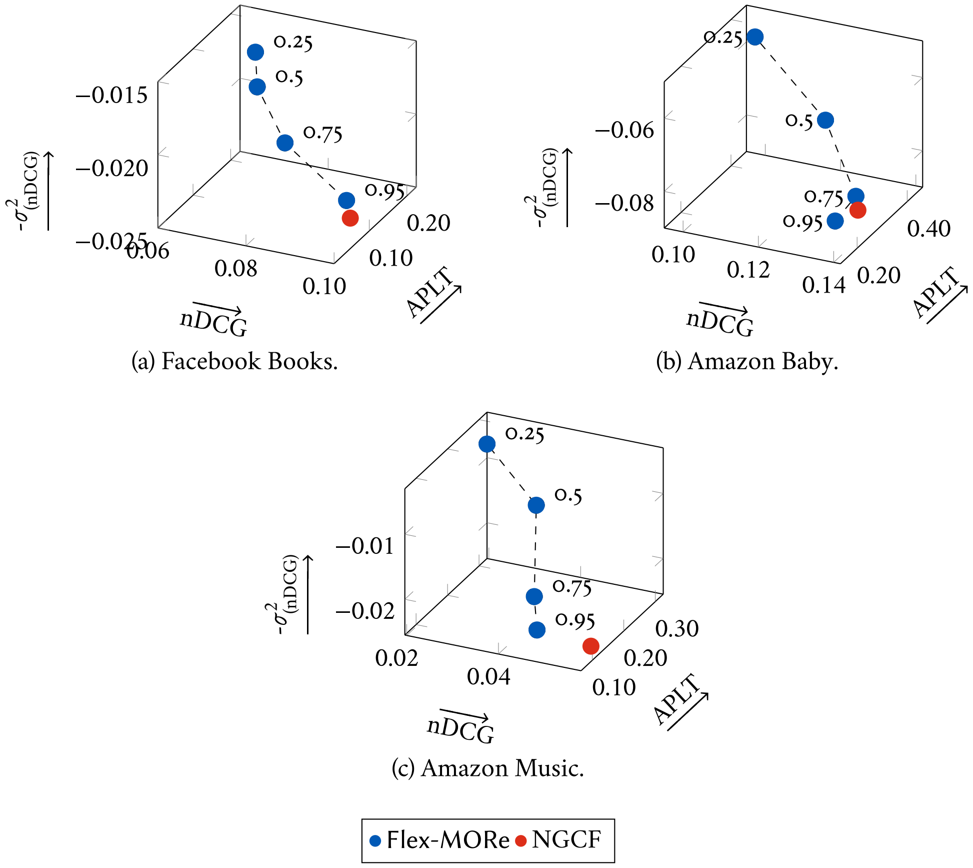
In conclusion, adjusting ω_{BPR} affects gradient intensity and enables Flex-MORe to effectively balance fairness and accuracy objectives.

9.5.4 Beyond-Accuracy Performance (RQ2)

This section investigates to what extent Flex-MORe can improve the beyond-accuracy performance of the recommendation backbones. In Table 9.2, we compare the vanilla backbones results with those gathered when combining them with Flex-MORe, i.e., BPRMF-Flex-MORe and NGCF-Flex-MORe. For a better visualization, Figure 9.4 depicts the Pareto frontiers obtained by varying the ω_{BPR} values in Eq. (9.3) adopting the NGCF backbone. Generally, it could be observed that, even assigning low gradient intensity to the multi-objective loss term $\mathcal{L}_{\text{Flex-MORe}}$, i.e., $\omega_{\mathrm{BPR}} = 0.95$, the framework improves the performance of the backbone on the fairness side, showing comparable values of nDCG and Recall, especially on Facebook Books and Amazon Baby datasets. This behavior becomes more noticeable by decreasing the value of ω_{BPR}, where Flex-MORe pays a drop in accuracy performance at the advantage of fairness metrics values (higher values of APLT and lower values of σ^2_{nDCG}). For instance, when utilizing NGCF as the backbone, Flex-MORe gets 13,7% and 12,6 % of improvements on APLT and σ^2_{nDCG} (on average on the three datasets), respectively,

Table 9.2. Comparison of backbones (see Section 9.4.3) and Flex-MORe performance on Amazon Baby, Facebook Books, and Amazon Music. Best and second-best results are in bold and underlined, respectively. Arrows indicate the descending or ascending order for better values. For statistical hypothesis testing, we use the paired t-test to compare Flex-MORe and the backbone ($p < 0.05$) and the Bonferroni adjustment to compare the Flex-MORe variants. Differences are statistically significant, unless denoted with † (‡ for the Bonferroni test).

Model	ω_{BPR}	Accuracy		Diversity		PF	CF	MO
		nDCG ↑	Recall ↑	Gini ↑	IC ↑	APLT ↑	$\sigma^2_{(\text{nDCG})}$ ↓	HV ↑
				Facebook Books				
BPRMF		**0.1059**†	$\underline{0.1734}^{\dagger}$	0.0957	$\underline{1381}$	0.0406	0.0259	0.0043
BPRMF-Flex-MORe	0.95	$\underline{0.1051}^{\dagger}$	**0.1748**†	0.1013	1372	0.0448	0.0246	0.0047
BPRMF-Flex-MORe	0.75	0.0926	0.1536	0.1161	1341	0.0681	0.0230	0.0063
BPRMF-Flex-MORe	0.5	0.0767	0.1326	$\underline{0.1281}$	1337	$\underline{0.1380}$	$\underline{0.0178}$	$\underline{0.0106}$
BPRMF-Flex-MORe	0.25	0.0475	0.0793	**0.1816**	**1477**	**0.2739**	**0.0119**	**0.0130**
NGCF		**0.0983**†	**0.1648**†	0.1438	$\underline{1736}$	0.0695	0.0235	0.0068
NGCF-Flex-MORe	0.95	$\underline{0.0967}^{\dagger}$	$\underline{0.1604}^{\dagger}$	0.1550	1714	0.0783	0.0226	0.0076
NGCF-Flex-MORe	0.75	0.0819	0.1396	$\underline{0.1710}$	1672	0.0880	0.0198	0.0072
NGCF-Flex-MORe	0.5	0.0699‡	0.1192‡	**0.2088**	**1850**	$\underline{0.1547}$	$\underline{0.0183}$	$\underline{0.0108}$
NGCF-Flex-MORe	0.25	0.0638‡	0.0998‡	0.0490	736	**0.2214**	**0.0179**	**0.0141**
				Amazon Baby				
BPRMF		**0.1622**†	**0.2049**†	$\underline{0.2700}$	**7439**	0.1848	0.0905	$\underline{0.0300}$
BPRMF-Flex-MORe	0.95	$\underline{0.1599}^{\dagger}$	$\underline{0.1978}^{\dagger}$	**0.2852**	$\underline{7340}$	$\underline{0.2524}$	0.0916	**0.0404**
BPRMF-Flex-MORe	0.75	0.1286	0.1674	0.1817	5699	0.1652	0.0753	0.0212
BPRMF-Flex-MORe	0.5	0.1059	0.1486	0.1617	5340	0.1422	$\underline{0.0610}$	0.0151
BPRMF-Flex-MORe	0.25	0.0901	0.1312	0.1074	3545	**0.2758**	**0.0511**	0.0248
NGCF		**0.1418**†	$\underline{0.1887}^{\dagger}$	0.2968	7466	0.2033	0.0798	0.0288
NGCF-Flex-MORe	0.95	$\underline{0.1404}^{\dagger\ddagger}$	**0.1917**†	0.2341	7274	0.1336	0.0786	0.0188
NGCF-Flex-MORe	0.75	0.1387‡	0.1861	$\underline{0.3276}$	$\underline{7684}$	0.2417	0.0791	0.0335
NGCF-Flex-MORe	0.5	0.1211	0.1633	**0.4227**	**7801**	$\underline{0.3850}$	$\underline{0.0711}$	**0.0466**
NGCF-Flex-MORe	0.25	0.0973	0.1344	0.1372	4568	**0.4617**	**0.0582**	$\underline{0.0449}$
				Amazon Music				
BPRMF		**0.0625**	**0.1064**	**0.2090**	**8418**	0.0505	0.0270	0.0032
BPRMF-Flex-MORe	0.95	$\underline{0.0512}$	$\underline{0.0943}$	$\underline{0.1753}$	7935	0.0723	0.0208	0.0037
BPRMF-Flex-MORe	0.75	0.0379	0.0659	0.1731	8072	0.0847	0.0169	0.0032
BPRMF-Flex-MORe	0.5	0.0308	0.0512	0.1662	$\underline{8381}$	$\underline{0.1565}$	$\underline{0.0148}$	$\underline{0.0048}$
BPRMF-Flex-MORe	0.25	0.0229	0.0398	0.0793	6159	**0.2124**	**0.0104**	**0.0049**
NGCF		**0.0563**	**0.0971**	**0.3270**	**9500**	0.1205	0.0245	0.0068
NGCF-Flex-MORe	0.95	$\underline{0.0484}$	$\underline{0.0886}$	0.2110	8594	0.0643	0.0206	0.0031
NGCF-Flex-MORe	0.75	0.0435	0.0789	0.2257	8802	0.1271	0.0190	0.0055
NGCF-Flex-MORe	0.5	0.0328	0.0607	$\underline{0.2317}$	$\underline{9174}$	$\underline{0.2894}$	$\underline{0.0138}$	**0.0095**
NGCF-Flex-MORe	0.25	0.0196	0.0371	0.1962	8545	**0.3254**	**0.0079**	$\underline{0.0064}$



(a) Facebook Books.

(b) Amazon Baby.

(c) Amazon Music.

Figure 9.4. Pareto frontiers obtained using Flex-MORe on the three datasets adopting NGCF as the backbone by varying the values of ω_{BPR}. The blue points refer to the models trained with Flex-MORe. Their labels are the ω_{BPR} values. The red points refer to the "vanilla" NGCF. For $\sigma^2_{(\text{nDCG})}$, the negative values are reported to have "the higher the better" semantic.

paying a drop of 13,5% on nDCG. This demonstrates a satisfactory balance between the considered objectives. A limitation of the results in Table 9.2 emerges when inspecting the diversity metrics values. In some cases, lowering the ω_{BPR} value can lead to a decline in Gini index and IC values. This behavior could occur due to an over-recommendation of niche items. This can be observed when setting $\omega_{\text{BPR}} = 0.25$. As a confirmation, in those experimentas we can notice high values of APLT. A similar situation arises for the specific case when using Flex-MORe with BPRMF on the Amazon Baby dataset, with a little impact on APLT. It is worth mentioning that the chosen scenario only considers fairness objectives. This diversity-oriented analysis is included to provide the reader with a more comprehensive view of overall performance [147, 188].

In conclusion, when Flex-MORe loss function comprises beyond-accuracy metrics, the Flex-MORe-enhanced backbone outperforms the vanilla backbones for all the considered objectives.

9.5.5 Performance Comparison with other MORSs (RQ3)

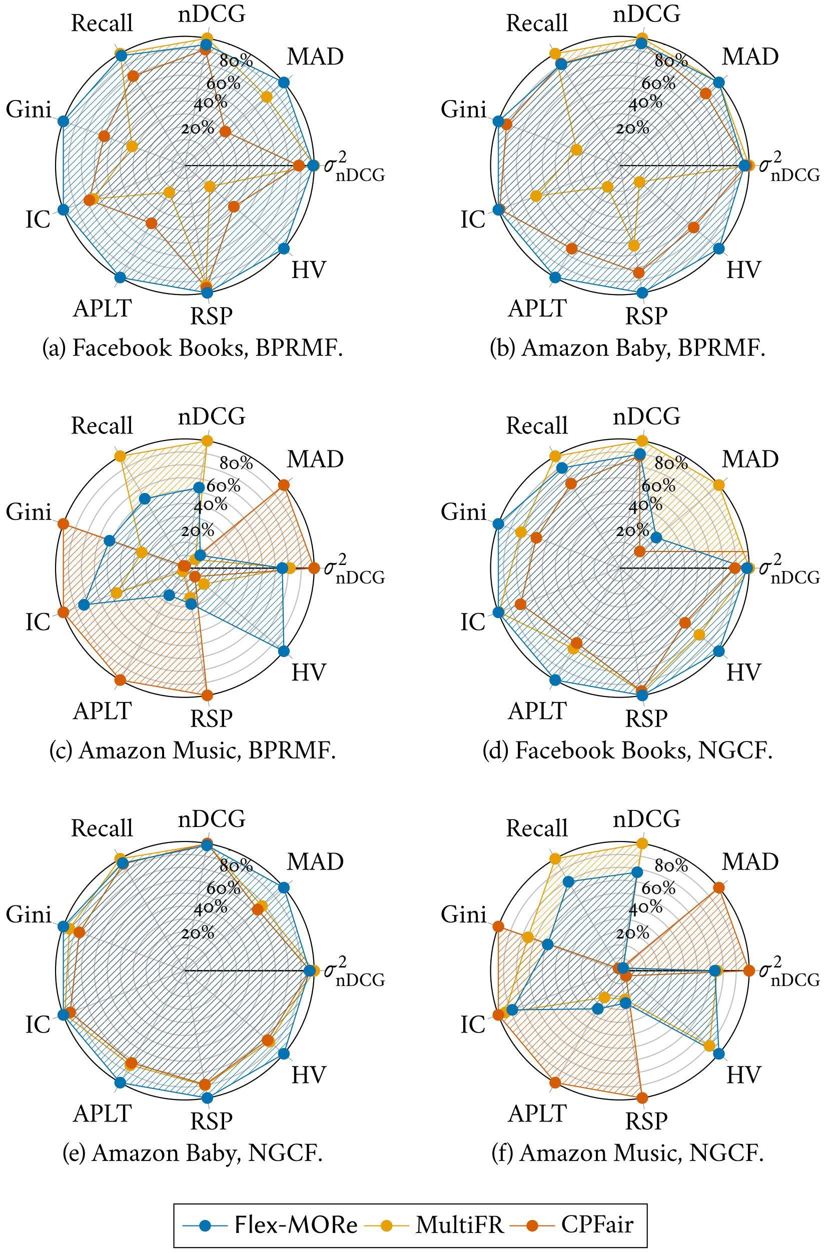
This experiment compares Flex-MORe with state-of-the-art MORSs designed explicitly for addressing provider and consumer fairness concerns. The radar plots in Figure 9.5 illustrate the relative performance achievement of Flex-MORe compared to MultiFR and CPFair. The radar plot provides a comprehensive overview of the model's performance. On Facebook Books and Amazon Baby, Flex-MORe shows balanced performance across the various metrics. It performs the best on the provider-fairness side (APLT and RSP) compared to MultiFR and CPFair, maintaining comparable performance on accuracy and consumer-fairness. Conversely, MultiFR prioritizes relevant recommendations and achieves a provider-fairness performance that is lower than the vanilla backbones.[5] On Amazon Music, CPFair fails to provide relevant suggestions, showing almost a 100% reduction in performance compared to MultiFR, which still achieves the best accuracy. In addition to these observations, the previously discussed patterns are confirmed. Although Flex-MORe is a flexible and general multi-objective recommendation framework (unlike MultiFR and CPFair that are explicitly designed for fairness purposes), it achieves a satisfactory balance among the analyzed objectives.

In conclusion, Flex-MORe can balance the trade-off among the considered objectives, achieving overall state-of-the-art multi-objective recommendation performance.

9.5.6 Impact of Normalization on Flex-MORe Performance (RQ4)

To avoid the presence of uncontrolled dominating objectives during model training, Flex-MORe normalizes the components related to each objective. In this section,

[5] Numerical detailed results are available in the online repository.



(a) Facebook Books, BPRMF.

(b) Amazon Baby, BPRMF.

(c) Amazon Music, BPRMF.

(d) Facebook Books, NGCF.

(e) Amazon Baby, NGCF.

(f) Amazon Music, NGCF.

Figure 9.5. Performance comparison among Flex-MORe, Multi-FR, and CPFair on Amazon Baby, Facebook Books, and Amazon Music. All metrics are assessed with cutoff 20. For metrics where a higher value is preferable, their values are divided by the metric best value. Conversely, for metrics where a lower value is better, each method's value is first converted to its reciprocal. The same division operation is then applied. Therefore, in these plots, a higher percentage indicates a more favorable outcome. For Flex-MORe, we consider $\omega_{\text{BPR}} = 0.75$, except for BPRMF-Flex-MORe on Amazon Baby ($\omega_{\text{BPR}} = 0.95$).

Table 9.3. Performance of Flex-MORe with and without normalizing the objective's components in $\mathcal{L}_{\text{BPR}}$.

Model	nDCG ↑	APLT ↑	σ^2_{nDCG} ↓	nDCG ↑	APLT ↑	σ^2_{nDCG} ↓
	Facebook Books			**Amazon Baby**		
BPRMF-Flex-MORe	0.0926	0.0681	0.0230	0.1599	0.2524	0.0916
BPRMF-Flex-MORe w/o σ	0.1022	0.0483	0.0261	0.1653	0.1850	0.0939
BPRMF-Flex-MORe w/o ζ	0.0572	0.6271	0.0202	0.1015	0.0867	0.0497
BPRMF-Flex-MORe w/o n	0.0471	0.6562	0.0177	0.0664	0.1077	0.0291
NGCF-Flex-MORe	0.0819	0.0783	0.0226	0.1387	0.2417	0.0791
NGCF-Flex-MORe w/o σ	0.0922	0.0558	0.0216	0.1326	0.1117	0.0707
NGCF-Flex-MORe w/o ζ	0.0742	0.1493	0.0184	0.1219	0.3241	0.0711
NGCF-Flex-MORe w/o n	0.0482	0.2925	0.0128	0.0498	0.3509	0.0260

we study the effectiveness of applying such normalization, i.e., the sigmoid ($\sigma(\cdot)$) and the z-score ($\zeta(\cdot)$) normalization functions in Eq. (9.1). To this end, we re-train the configurations of Flex-MORe seen in section 9.5.5 — that show a satisfactory trade-off among the objectives — by removing the computation of the sigmoid and the z-score normalization. Table 9.3 reports the results for the Facebook Books and Amazon Baby datasets. With "backbone"-Flex-MORe w/o n, we denote the variants of Flex-MORe without both $\sigma(\cdot)$ and $\zeta(\cdot)$, while "backbone"-Flex-MORe w/o $\sigma(\cdot)$ and "backbone"-Flex-MORe w/o $\zeta(\cdot)$ denotes the variants without the sigmoid and the z-normalization, respectively. The results reveal that removing the computation of both $\sigma(\cdot)$ and $\zeta(\cdot)$ in $\mathcal{L}_{\text{Flex-MORe}}$ causes an accuracy decrease of at least 41%, thus making the provided recommendations less relevant and suitable, although the higher beyond-accuracy performance. By applying $\sigma(\cdot)$ or $\zeta(\cdot)$ individually, the performance on the accuracy side improves, confirming that the re-scaling of the objective's losses benefits the trade-off balance. Notably, the variants w/o $\sigma(\cdot)$ achieve comparable results with Flex-MORe in accuracy and consumer fairness. However, they consistently fail from the provider fairness perspective. Conversely, the variants w/o $\zeta(\cdot)$ show fluctuating performance that depends on the adopted backbone. On the one hand, NGCF-Flex-MORe w/o $\zeta(\cdot)$ demonstrates the ability to achieve an effective trade-off. On the other hand, BPRMF-Flex-MORe w/o $\zeta(\cdot)$ exhibits a considerable loss in accuracy without ensuring gains on the provider fairness side.

To conclude, the application of $\sigma(\cdot)$ and $\zeta(\cdot)$ in Eq.(2) is crucial to achieve balanced performance among the accuracy and fairness-oriented metrics.

9.5.7 Training Efficiency (RQ5)

In this section, we empirically analyze the training efficiency of Flex-MORe. We observe that the training loss overhead in Eq. (9.1) is mainly caused by applying $\zeta(\cdot)$ and $\sigma(\cdot)$ on the squared difference between a differentiable approximated metric performance for a user and its utopia value, repeating these operations for each

Table 9.4. Training efficiency comparison of Flex-MORe by varying dataset and the number of objectives. The chosen backbone is BPRMF. The training time is reported in seconds. The symbol ✓ indicates whether the provider (PF) or consumer (CF) fairness objectives are involved during the training.

Objectives		Datasets		
PF	**CF**	**Facebook Books**	**Amazon Baby**	**Amazon Music**
✓		55.62	937.53	8115.51
	✓	58.82	921.20	8116.78
✓	✓	70.06	963.39	8288.44

metric and user. Therefore, we expect an increased training time as the number of objectives involved in the training and the number of users and items in the dataset increase. For this reason, we evaluate the empirical training efficiency of Flex-MORe for each dataset by varying the number of fairness-related objectives involved in the training. We conduct these experiments on a machine equipped with an Intel(R) Core(TM) i7-5820K CPU, 64 GB RAM, and NVIDIA GeForce RTX 3090 GPU. Table 9.4 reports the training duration in seconds of Flex-MORe with different combinations of objectives for the consumer (CF) and producer (PF) sides. BPRMF is used as the base model to assess training performance. The results reveal that the proposed framework achieves suitable training times, even as the number of fairness constraints increases. Indeed, by adding more objectives, the training time grows marginally. Furthermore, the training overhead scales well according to the dataset's size, demonstrating the model's capacity to optimize multiple objectives simultaneously in real-world applications.

In conclusion, Flex-MORe demonstrates reasonable training efficiency, showing adequate scalability as the number of objectives and the dataset's size increase.

9.6 Summary

This work introduces Flex-MORe, a Flexible Multi-Objective Recommendation framework that extends recommender system (RS) training, which often focuses solely on accuracy, by incorporating an objective-agnostic and scale-aware loss function. A key contribution of Flex-MORe is its smoothing approach, which makes ranking-based metrics differentiable and allows their incorporation into the framework. Furthermore, Flex-MORe addresses the challenge of metrics having different scales by normalizing the squared errors between actual metric values and their ideal counterparts, ensuring that no single objective dominates the optimization process. Experimental analysis demonstrates that Flex-MORe effectively balances diverse objectives, leading to state-of-the-art performance. Moreover, Flex-MORe lets us adjust the weights in the loss function to control the influence of the objectives, while maintaining competitive accuracy. Future research will explore the possibility of personalizing utopia points at the user level for each metric to better calibrate recommendations according to individual preferences [97, 189]. Additionally, investigating methods to dynamically adjust the weights of the scalarization within the multi-objective loss function during training presents another promising direction further enhancing the framework's ability to achieve diverse trade-off levels.

Part IV
Conclusion

Chapter 10

Closing Remarks

Traditional research in Recommender Systems (RSs) has predominantly focused on their ability to deliver relevant recommendations during optimization and evaluation phases. However, as real-world recommendation tasks become increasingly complex, it is evident that a more holistic approach that accounts for multiple quality dimensions of recommendations and addresses the diverse needs of various stakeholders is necessary. This dissertation aims to enhance the multiple perspectives of RSs by tackling key challenges in their multi-objective evaluation and optimization.

Multi-objective evaluation. The first part of this work concentrates on advancing the methodologies for the multi-objective evaluation of RSs. In Chapter 4, we examined the underexplored performance of graph-based RSs regarding consumer and provider fairness. Leveraging Pareto frontiers, we qualitatively analyzed the trade-offs between accuracy and beyond-accuracy dimensions. Our findings reveal that user-user and item-item message-passing strategies can significantly enhance accuracy/fairness trade-offs. Conversely, implicit message-passing mechanisms in recent approaches were found to adversely impact consumer-provider fairness scenarios, underscoring the need for careful algorithmic design. To move a step forward to a quantitative evaluation, Chapter 5 introduced an approach for assessing RS performance across multiple objectives using quality indicators of Pareto frontiers. This methodology demonstrated that ranking models solely based on predictive accuracy can overlook their broader potential. By incorporating beyond-accuracy dimensions, we showed that this multi-objective evaluation approach could overturn traditional performance rankings, offering a more balanced perspective. To further aid practitioners and researchers, we proposed in Chapter 6 an analytical framework for assessing the sensitivity of RS models to hyper-parameter tuning in multi-objective scenarios. This framework provides actionable insights into how specific hyper-parameters influence trade-offs among objectives, enabling more informed decision-making in RS optimization.

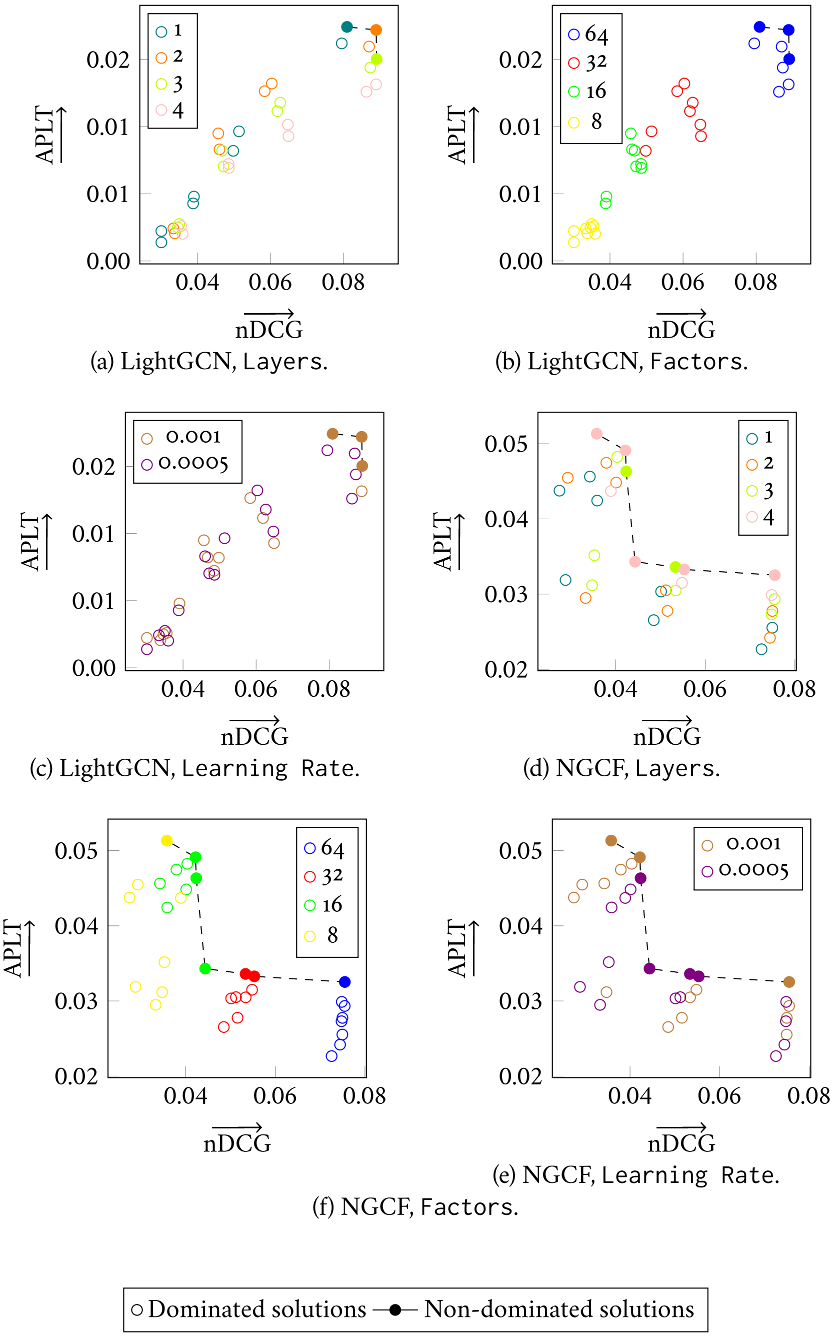
Multi-objective recommender systems. The second part of this dissertation focuses on addressing challenges in designing multi-objective RSs that simultaneously optimize multiple goals. In Chapter 7, we conducted a reproducibility study to uncover ambiguities and challenges in multi-objective RS research. Our study highlighted the critical importance of transparency in experimental methodologies, particularly in reporting model selection strategies, which are often overlooked but essential for fair performance evaluation. Inspired by these findings, Chapter 8 introduced Population Distance from Utopia (PDU), a novel strategy for selecting a single Pareto-optimal solution tailored to RS tasks. Unlike traditional methods, PDU introduces a unique property, named calibration, that aligns the selected solution more closely with user preferences. This approach allows for a more nuanced and personalized selection of optimal solutions from the Pareto frontier. Finally, in Chapter 9, we presented Flex-MORe, a flexible framework for multi-objective recommendations. Flex-MORe integrates an objective-agnostic and scale-aware loss function into the training process of RS backbones, enabling simultaneous optimization of multiple objectives. Additionally, Flex-MORe incorporates a novel technique to make ranking-based recommendation metrics differentiable, ensuring seamless integration of beyond-accuracy objectives into the optimization pipeline.

Future directions. The research contributions presented in this dissertation seek to advance recommendation approaches that move beyond accuracy while maintaining its importance. While our focus has been on methodologies rather than a restricted set of objectives, the broader perspectives on recommendation extend beyond those explored in this work. On the evaluation front, future directions include studying trade-offs between accuracy and dimensions related to trustworthy artificial intelligence, such as privacy, explainability, and data minimization in recommendations. From an optimization perspective, we aim to extend Flex-MORe to achieve calibrated recommendations, incorporating the concept of a generalized utopia point within this framework, as introduced in PDU. Additionally, we plan to integrate multi-objective optimization into matrix factorization, assigning objective-driven semantics to latent factors, thereby enriching their interpretability and utility.

Personal Thought. The entire dissertation is written in the first person plural, recognizing all my co-authors and colleagues, among whom some have positively contaminated my research, and others have tried to guide me during these three years. Now, it is just Vincenzo speaking. Looking at the content of this thesis, I am proud of what I have achieved in these three years. Not because I have achieved extraordinary scientific results, considering that knowing I do not know continually alienates me. I am proud because I remember how this journey began, evolved, and ended. I am proud of not losing the compass of my research theme and making my way through, aware that I only spilled a drop in the ocean. It has been three years in which *research questions* were flowing. These I have tried to answer. One in particular remains unresolved. I have experienced a thousand difficulties, especially psychological ones, often finding myself facing them alone, not feeling that anyone could fully understand me. I have made sacrifices, made little money compared to the mental health lost, and cried. *But then, was it worth it?* I cannot answer, and giving one today is impossible. Despite this lack, I have made many efforts over the past three years. And I can say that I would do it again because I did it to give value to the me of the past of *"four years, four months and twenty one days"* who thought, *"I care about me of the future, today's problem should not be his problem."* For once, I didn't care about me of the future, but I loved me of the past despite of being aware of what I would be up against. And it didn't even end.

Chapter A

Appendix A



(a) LightGCN, `Layers`.

(b) LightGCN, `Factors`.

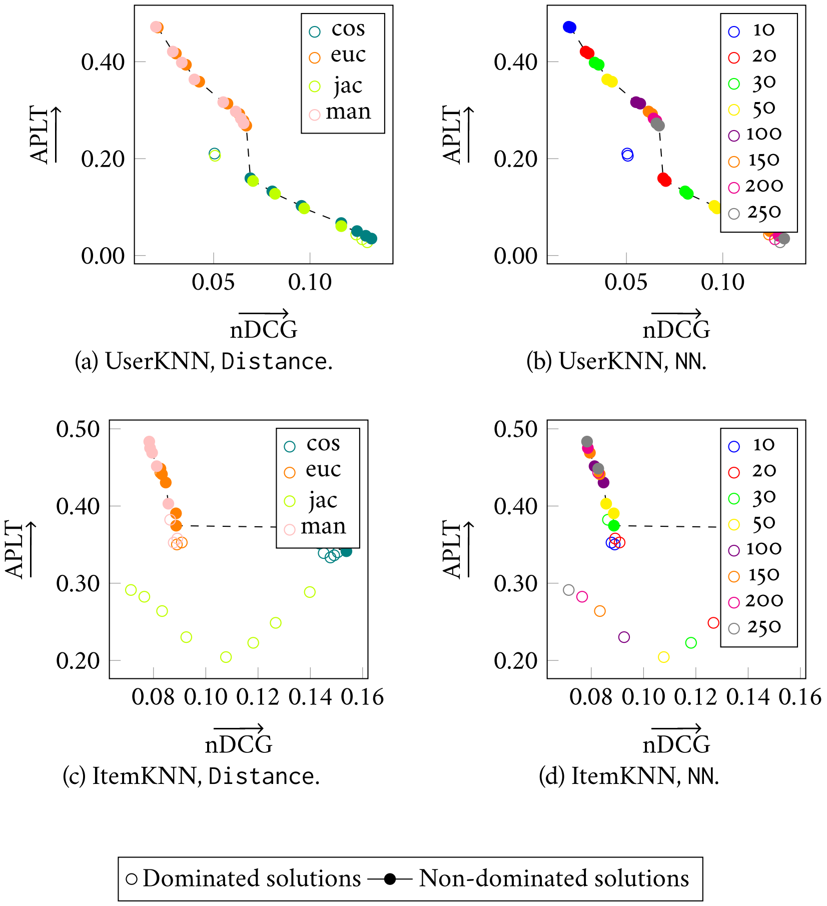
(c) LightGCN, `Learning Rate`.

(d) NGCF, `Layers`.

(e) NGCF, `Learning Rate`.

(f) NGCF, `Factors`.

Figure A.1. Accuracy/Bias trade-offs on Amazon Book, assessed through *nDCG*/*APLT*, for LightGCN and NGCF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



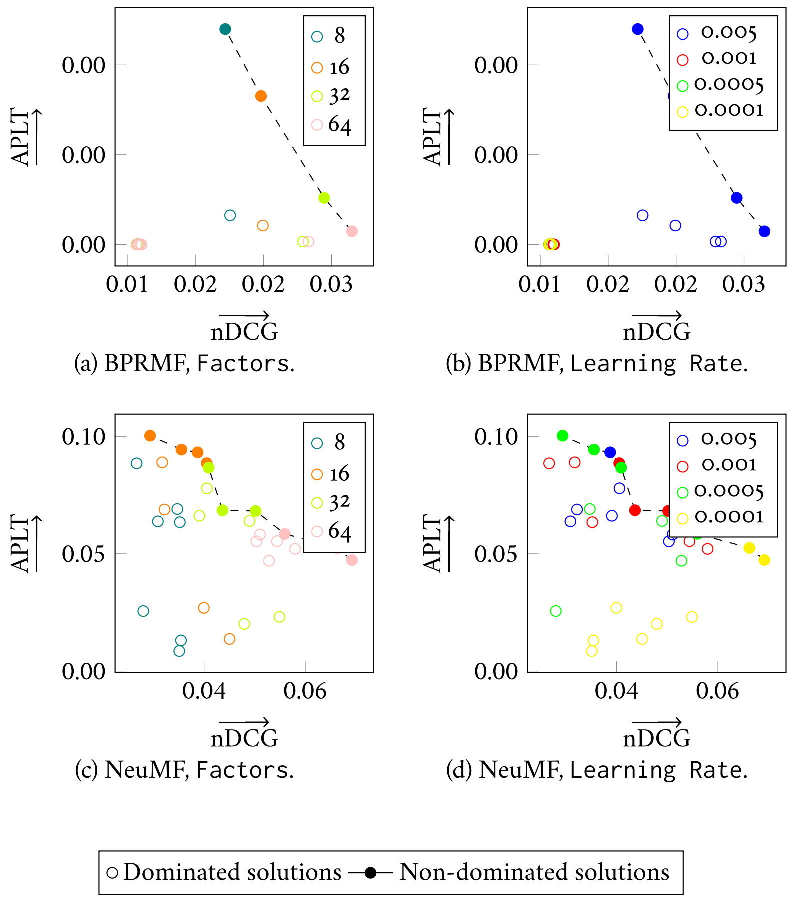
(a) UserKNN, `Distance`.

(b) UserKNN, `NN`.

(c) ItemKNN, `Distance`.

(d) ItemKNN, `NN`.

Figure A.2. Accuracy/Bias trade-offs on Amazon Book, assessed through *nDCG/APLT*, for UserKNN and ItemKNN. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



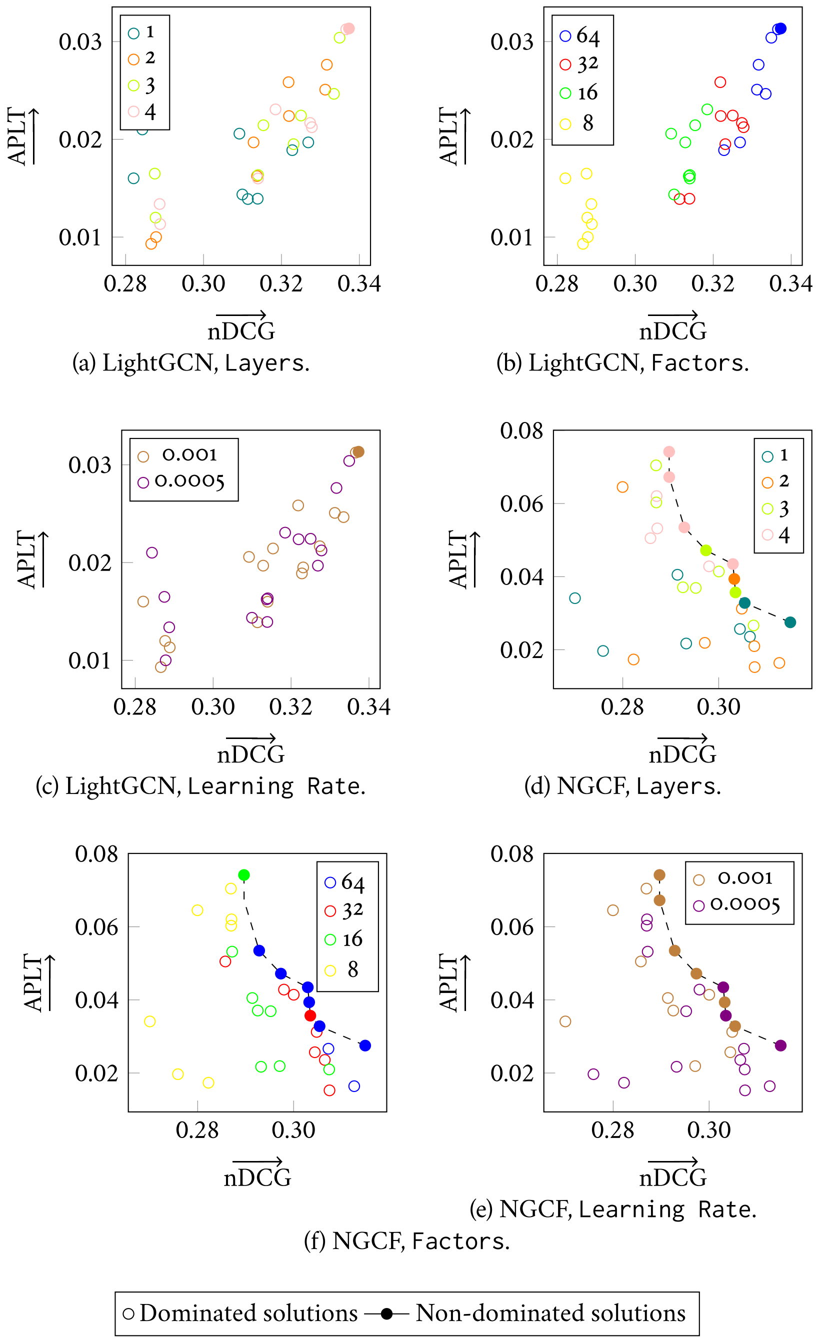
(a) BPRMF, Factors.

(b) BPRMF, Learning Rate.

(c) NeuMF, Factors.

(d) NeuMF, Learning Rate.

Figure A.3. Accuracy/Bias trade-offs on Amazon Book, assessed through *nDCG*/*APLT*, for BPRMF and NeuMF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



(a) LightGCN, Layers.

(b) LightGCN, Factors.

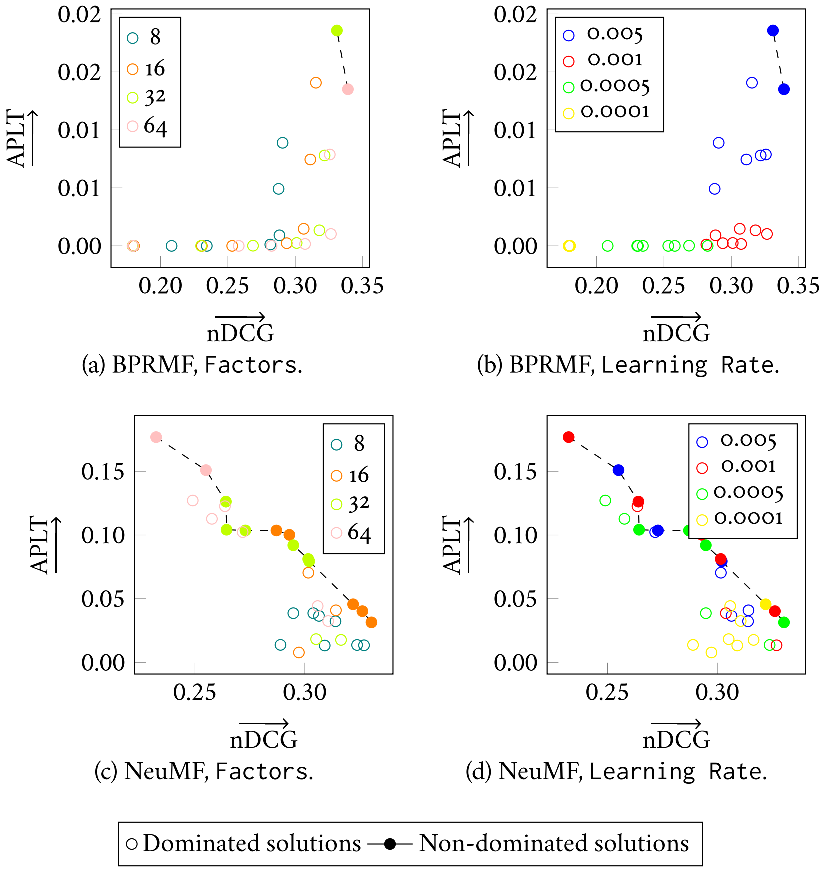
(c) LightGCN, Learning Rate.

(d) NGCF, Layers.

(e) NGCF, Learning Rate.

(f) NGCF, Factors.

Figure A.4. Accuracy/Bias trade-offs on Movielens1M, assessed through *nDCG*/*APLT*, for LightGCN and NGCF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



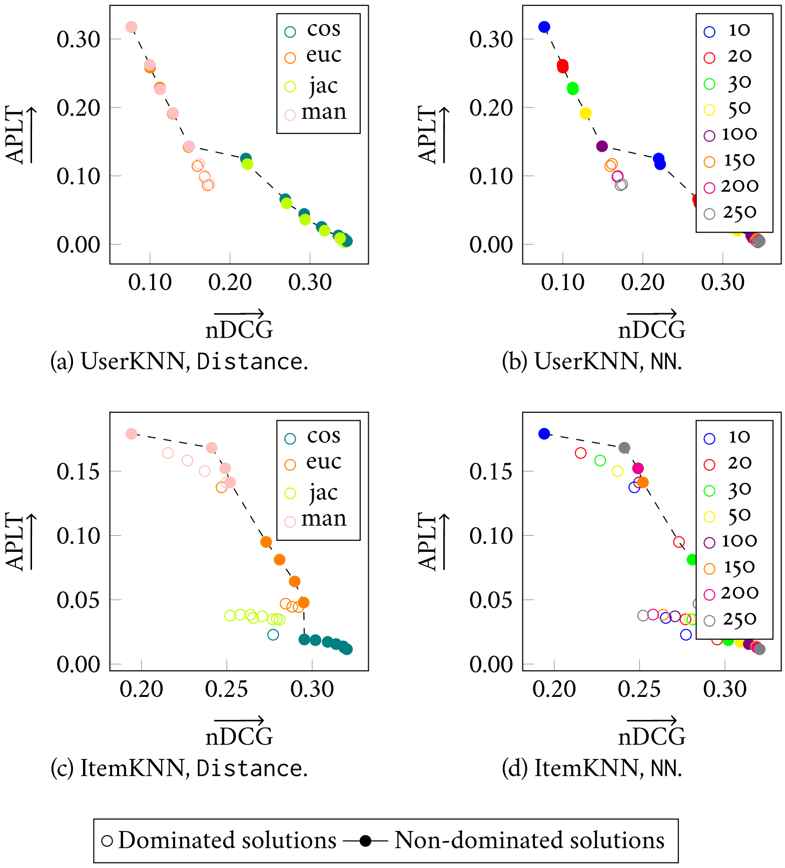
(a) BPRMF, `Factors`.

(b) BPRMF, `Learning Rate`.

(c) NeuMF, `Factors`.

(d) NeuMF, `Learning Rate`.

Figure A.5. Accuracy/Bias trade-offs on Movielens 1M, assessed through *nDCG/APLT*, for BPRMF and NeuMF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



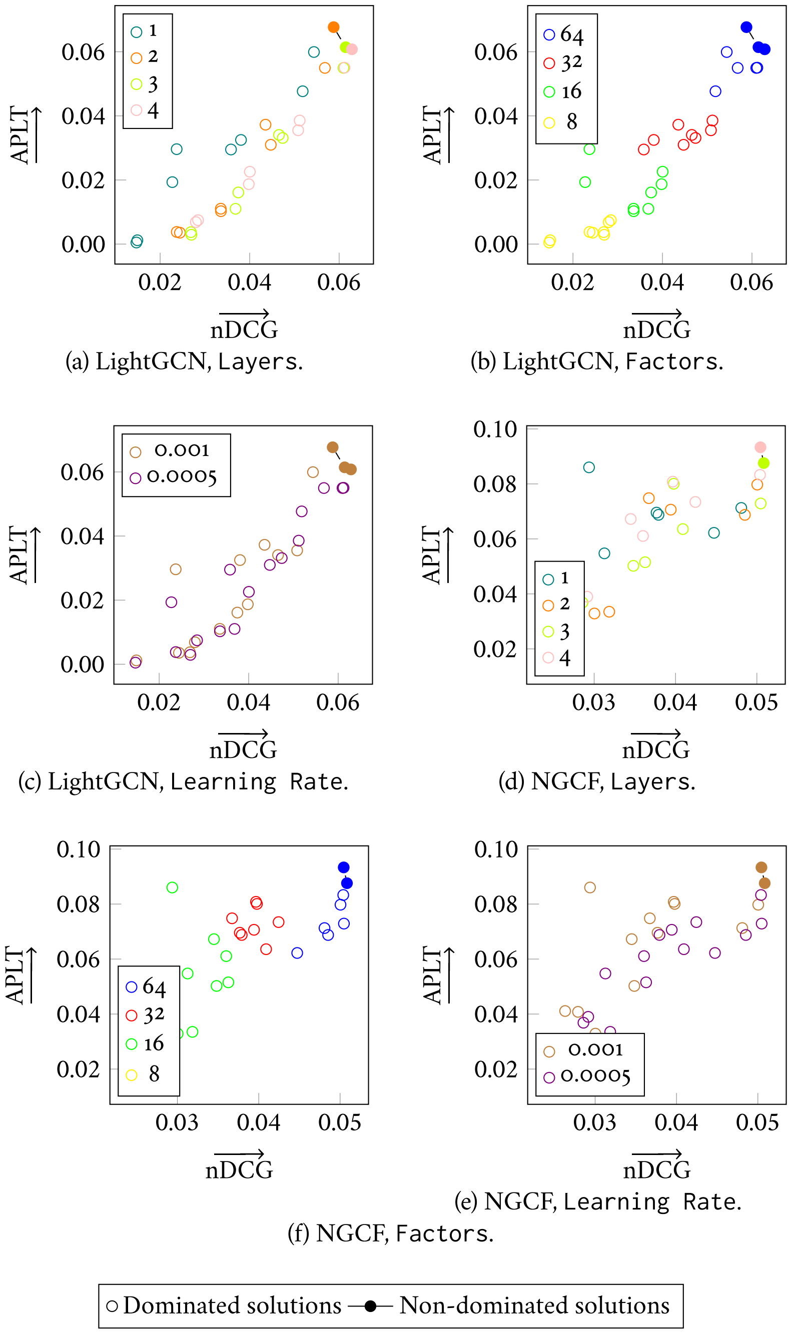
(a) UserKNN, `Distance`.

(b) UserKNN, `NN`.

(c) ItemKNN, `Distance`.

(d) ItemKNN, `NN`.

Figure A.6. Accuracy/Bias trade-offs on Movielens1M, assessed through *nDCG*/*APLT*, for UserKNN and ItemKNN. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



(a) LightGCN, `Layers`.

(b) LightGCN, `Factors`.

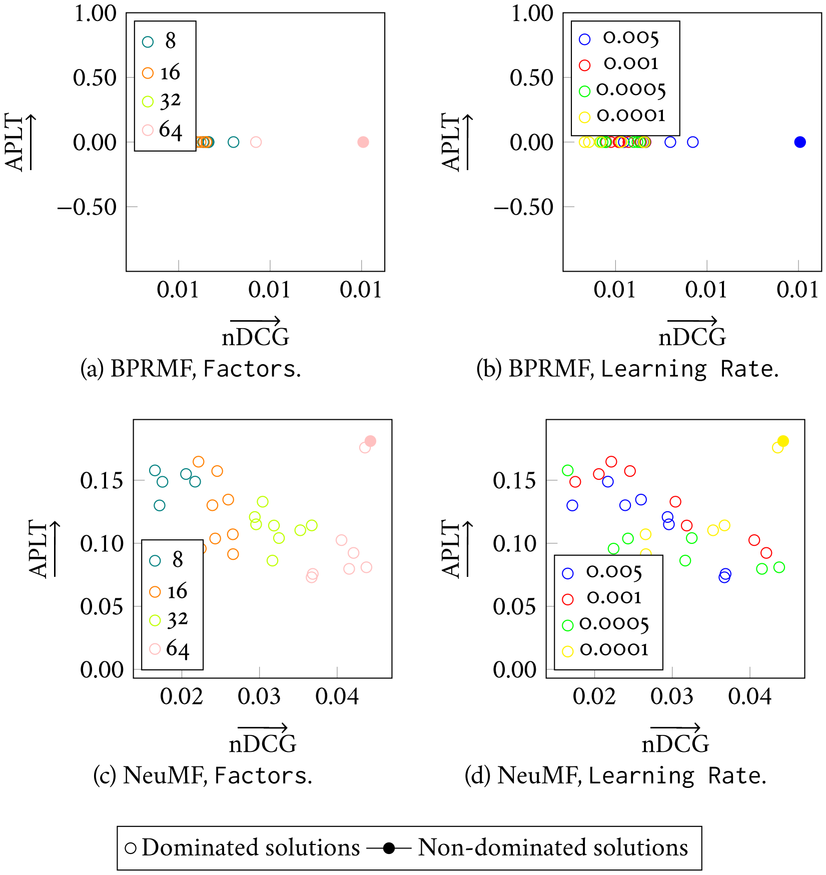
(c) LightGCN, `Learning Rate`.

(d) NGCF, `Layers`.

(e) NGCF, `Learning Rate`.

(f) NGCF, `Factors`.

Figure A.7. Accuracy/Bias trade-offs on Amazon Music, assessed through *nDCG*/*APLT*, for LightGCN and NGCF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



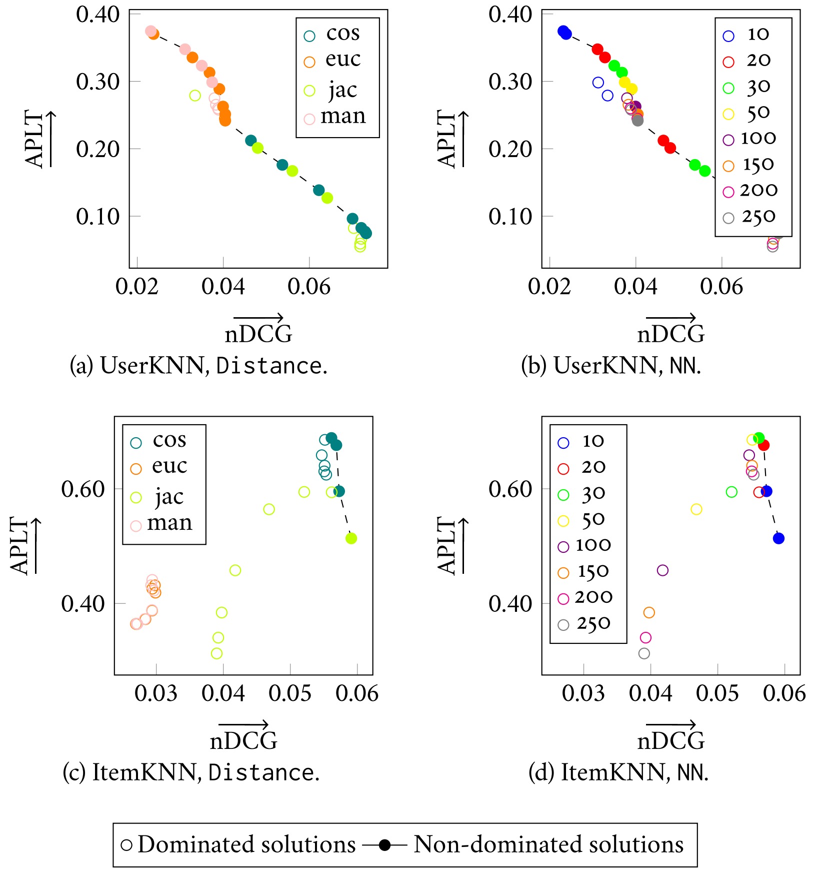
(a) BPRMF, `Factors`.

(b) BPRMF, `Learning Rate`.

(c) NeuMF, `Factors`.

(d) NeuMF, `Learning Rate`.

Figure A.8. Accuracy/Bias trade-offs on Amazon Music, assessed through *nDCG*/*APLT*, for BPRMF and NeuMF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



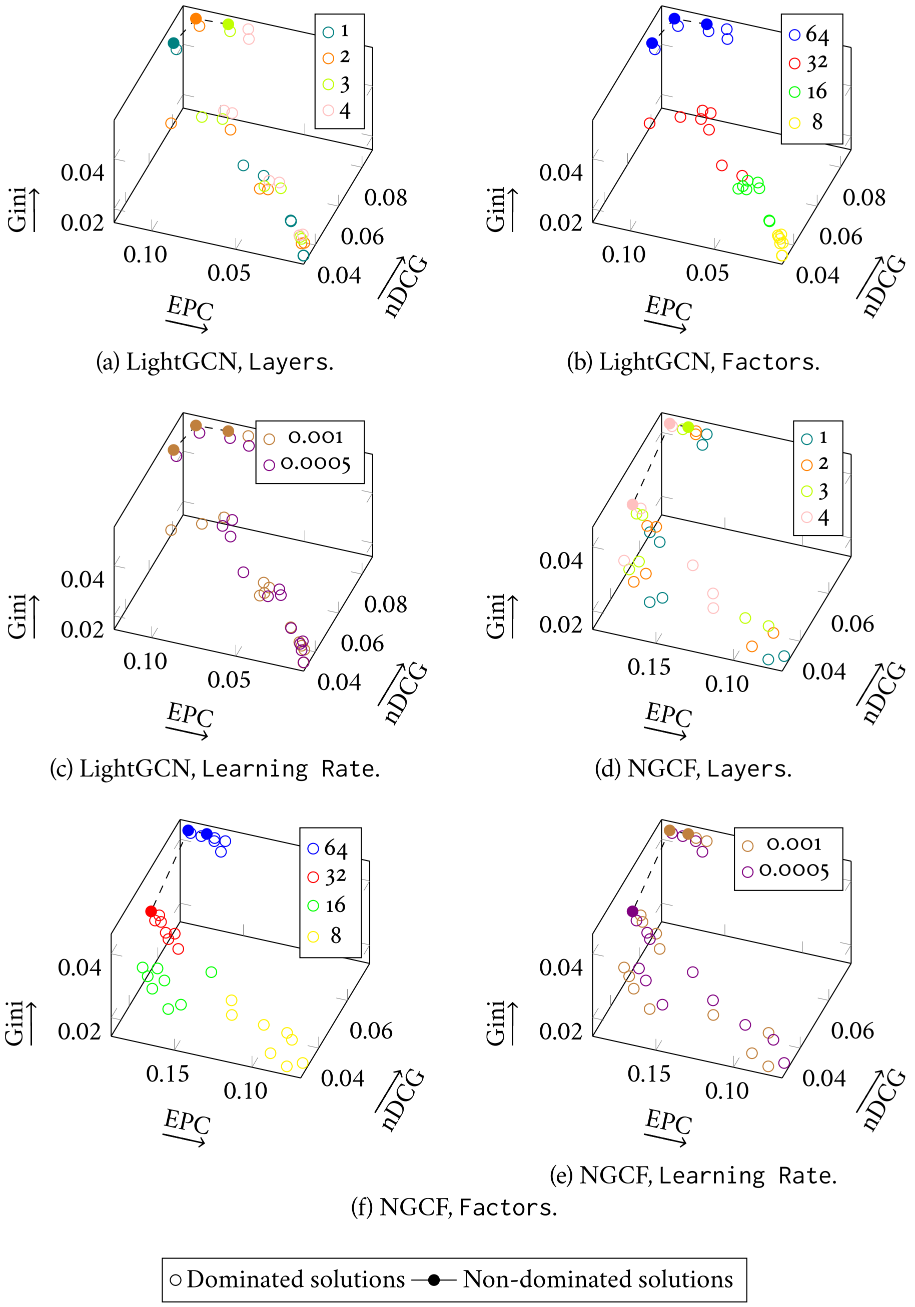
(a) UserKNN, `Distance`.

(b) UserKNN, `NN`.

(c) ItemKNN, `Distance`.

(d) ItemKNN, `NN`.

Figure A.9. Accuracy/Bias trade-offs on Amazon Music, assessed through *nDCG/APLT*, for UserKNN and ItemKNN. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



(a) LightGCN, `Layers`.

(b) LightGCN, `Factors`.

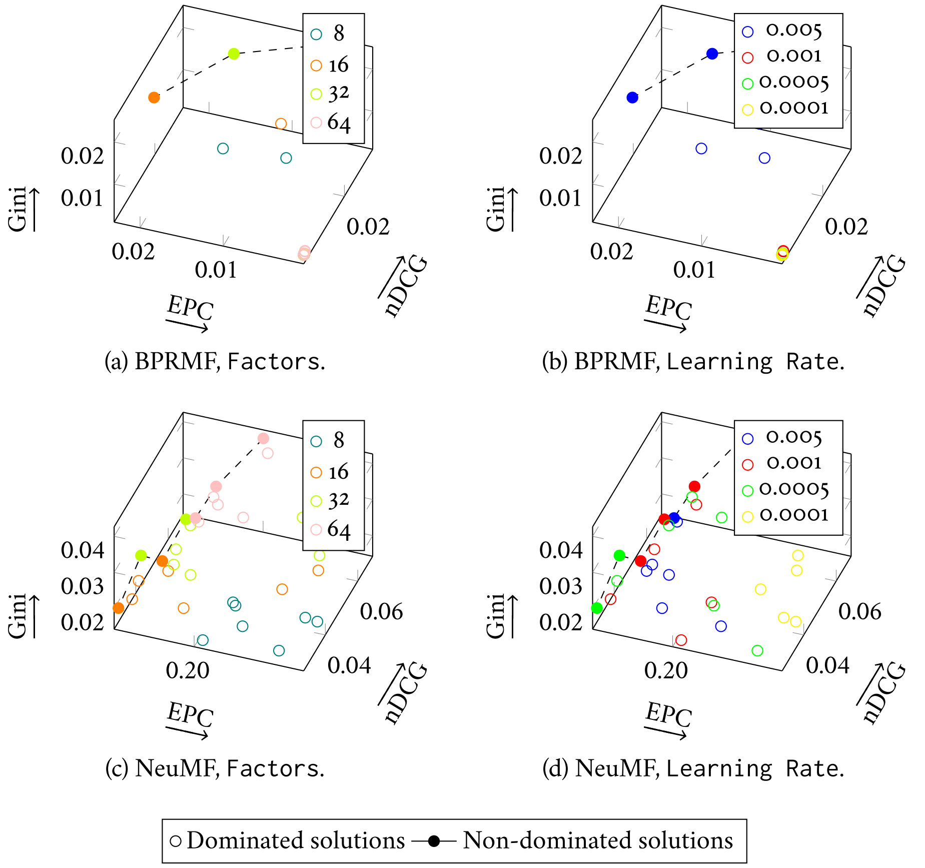
(c) LightGCN, `Learning Rate`.

(d) NGCF, `Layers`.

(e) NGCF, `Learning Rate`.

(f) NGCF, `Factors`.

Figure A.10. Accuracy/Novelty/Diversity trade-offs on Amazon Book, assessed through *nDCG*/*EPC*/*Gini*, for LightGCN and NGCF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



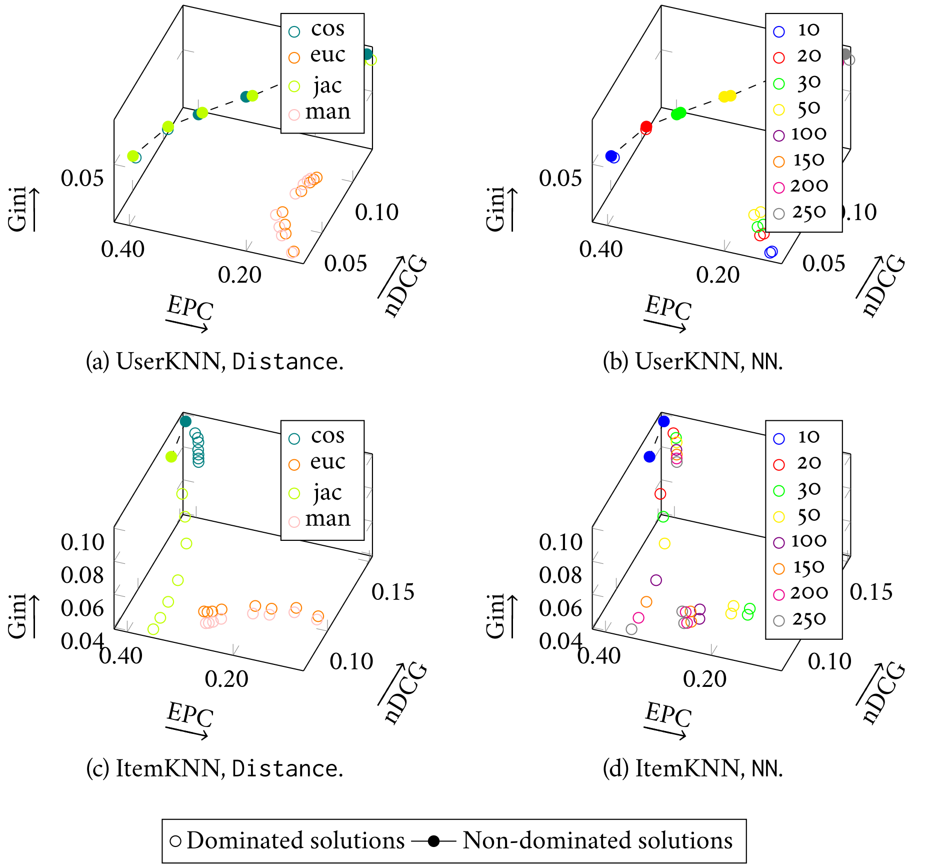
(a) BPRMF, `Factors`.

(b) BPRMF, `Learning Rate`.

(c) NeuMF, `Factors`.

(d) NeuMF, `Learning Rate`.

Figure A.11. Accuracy/Novelty/Diversity trade-offs on Amazon Book, assessed through *nDCG*/*EPC*/*Gini*, for BPRMF and NeuMF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



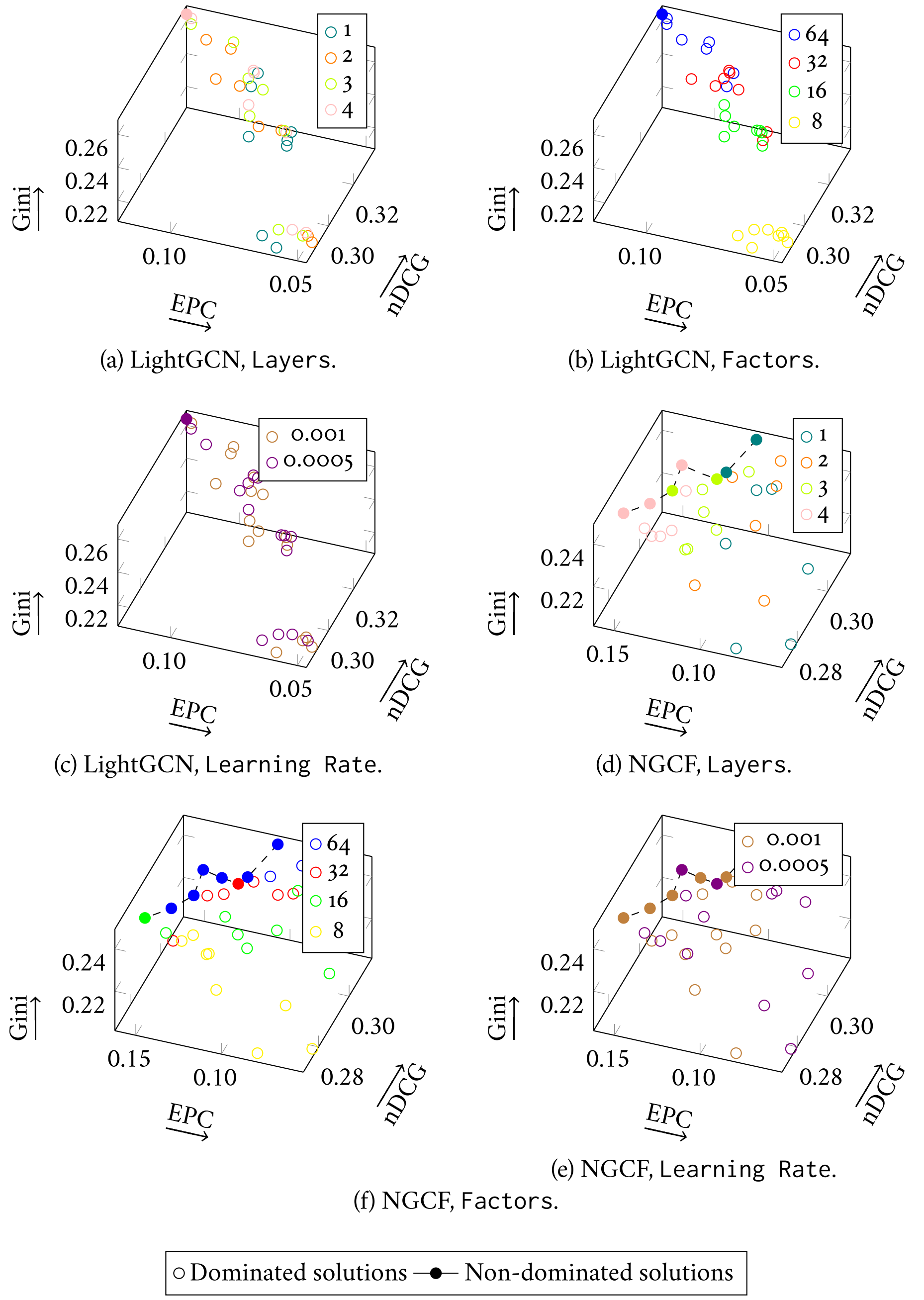
(a) UserKNN, `Distance`.

(b) UserKNN, `NN`.

(c) ItemKNN, `Distance`.

(d) ItemKNN, `NN`.

Figure A.12. Accuracy/Novelty/Diversity trade-offs on Amazon Book, assessed through *nDCG*/*EPC*/*Gini*, for UserKNN and ItemKNN. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



(a) LightGCN, Layers.

(b) LightGCN, Factors.

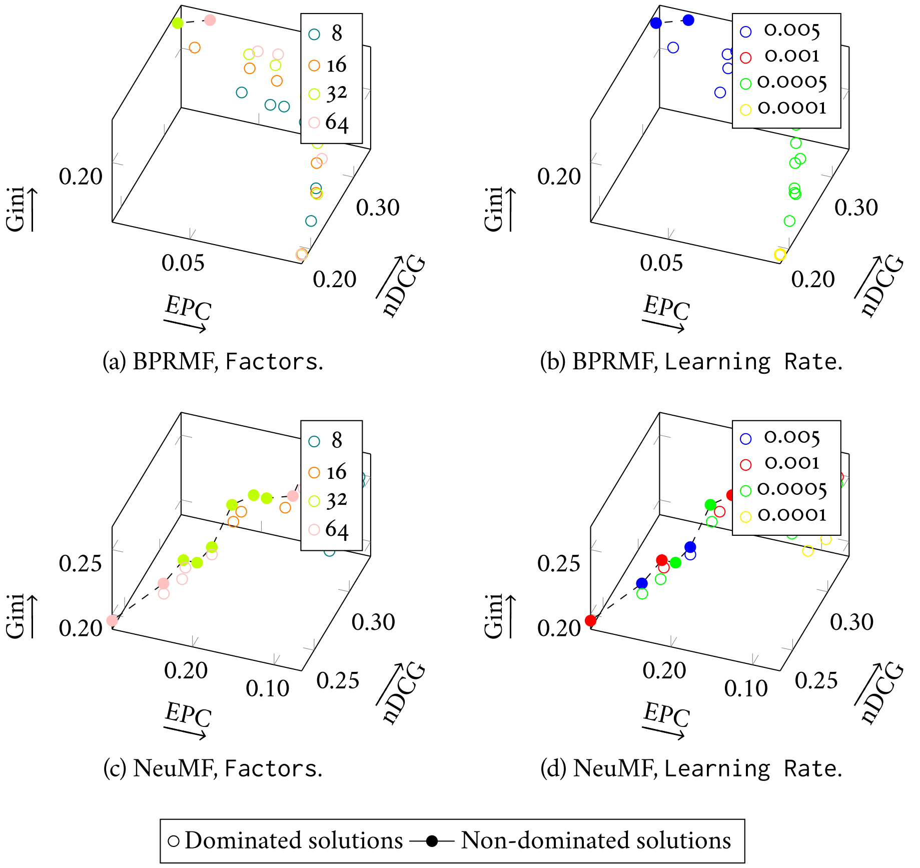
(c) LightGCN, Learning Rate.

(d) NGCF, Layers.

(e) NGCF, Learning Rate.

(f) NGCF, Factors.

Figure A.13. Accuracy/Novelty/Diversity trade-offs on Movielens1M, assessed through *nDCG*/*EPC*/*Gini*, for LightGCN and NGCF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



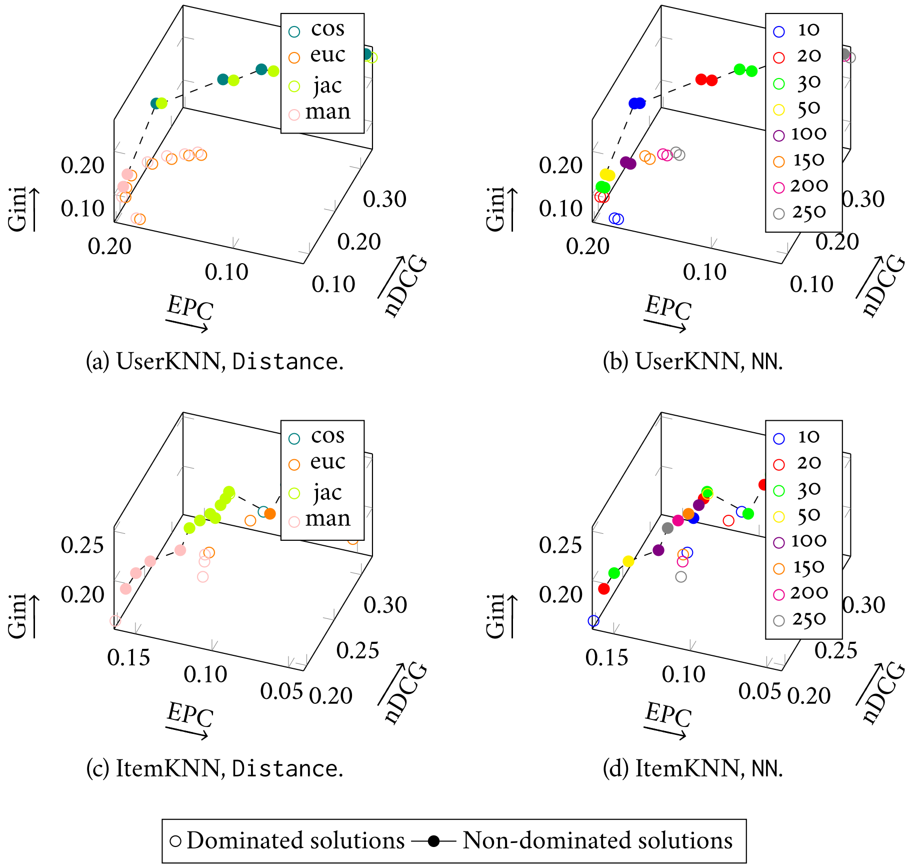
(a) BPRMF, `Factors`.

(b) BPRMF, `Learning Rate`.

(c) NeuMF, `Factors`.

(d) NeuMF, `Learning Rate`.

Figure A.14. Accuracy/Novelty/Diversity trade-offs on Movielens1M, assessed through *nDCG*/*EPC*/*Gini*, for BPRMF and NeuMF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



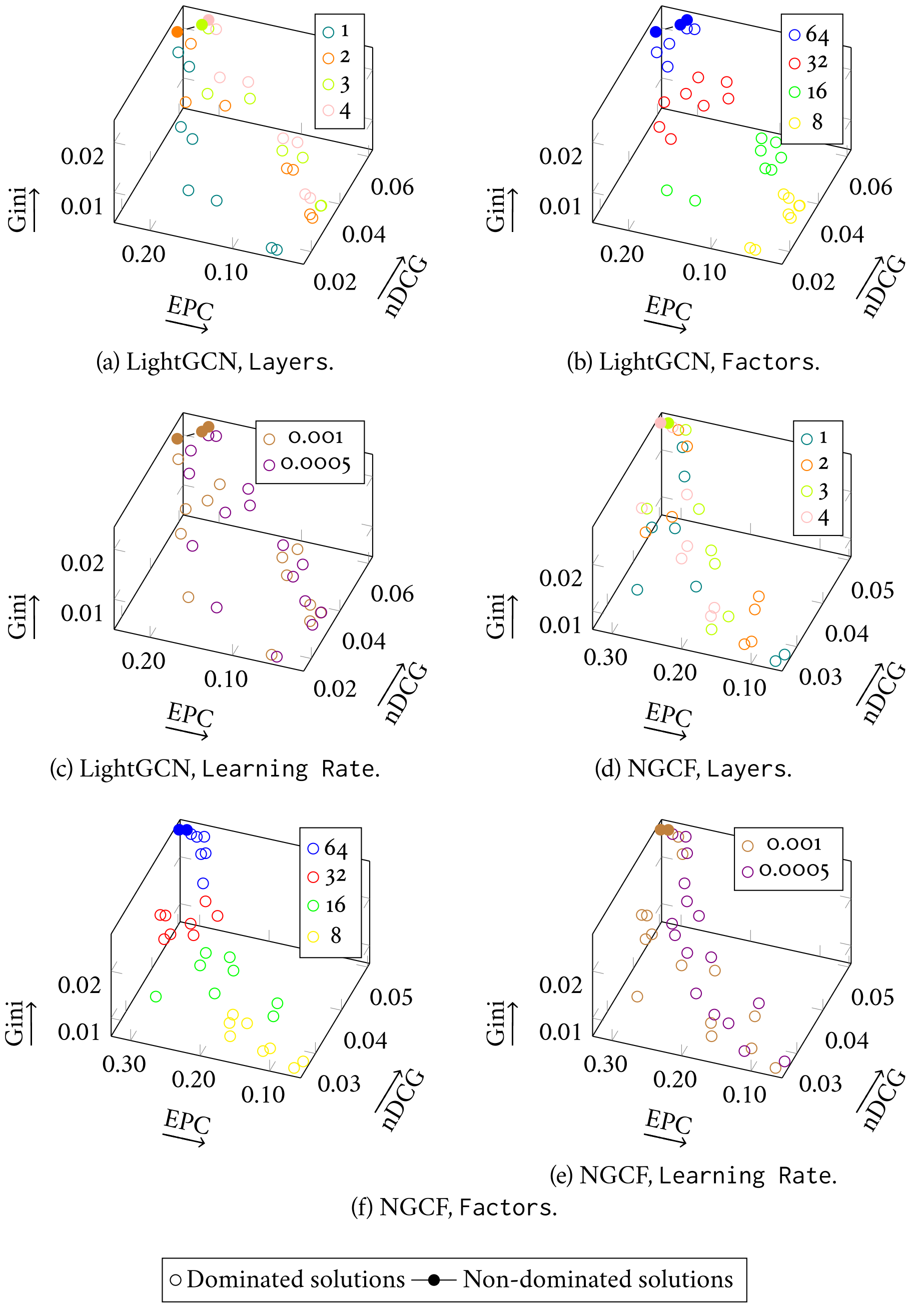
(a) UserKNN, `Distance`.

(b) UserKNN, NN.

(c) ItemKNN, `Distance`.

(d) ItemKNN, NN.

Figure A.15. Accuracy/Novelty/Diversity trade-offs on Movielens1M, assessed through *nDCG*/*EPC*/*Gini*, for UserKNN and ItemKNN. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



(a) LightGCN, `Layers`.

(b) LightGCN, `Factors`.

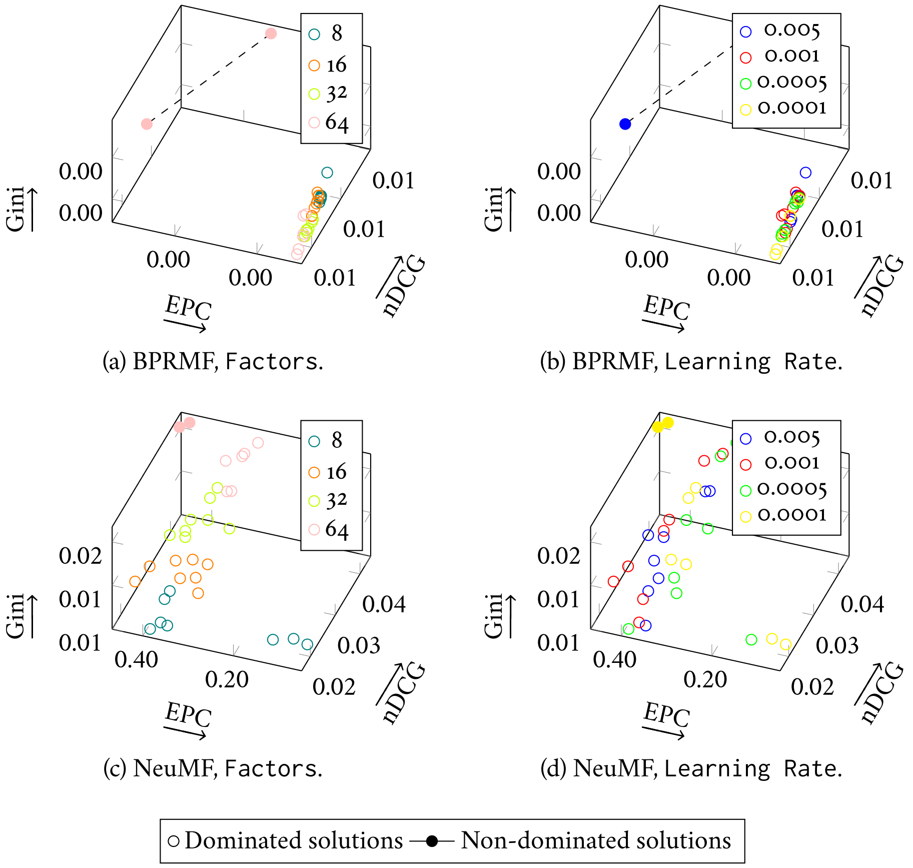
(c) LightGCN, `Learning Rate`.

(d) NGCF, `Layers`.

(e) NGCF, `Learning Rate`.

(f) NGCF, `Factors`.

Figure A.16. Accuracy/Novelty/Diversity trade-offs on Amazon Music, assessed through *nDCG*/*EPC*/*Gini*, for LightGCN and NGCF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



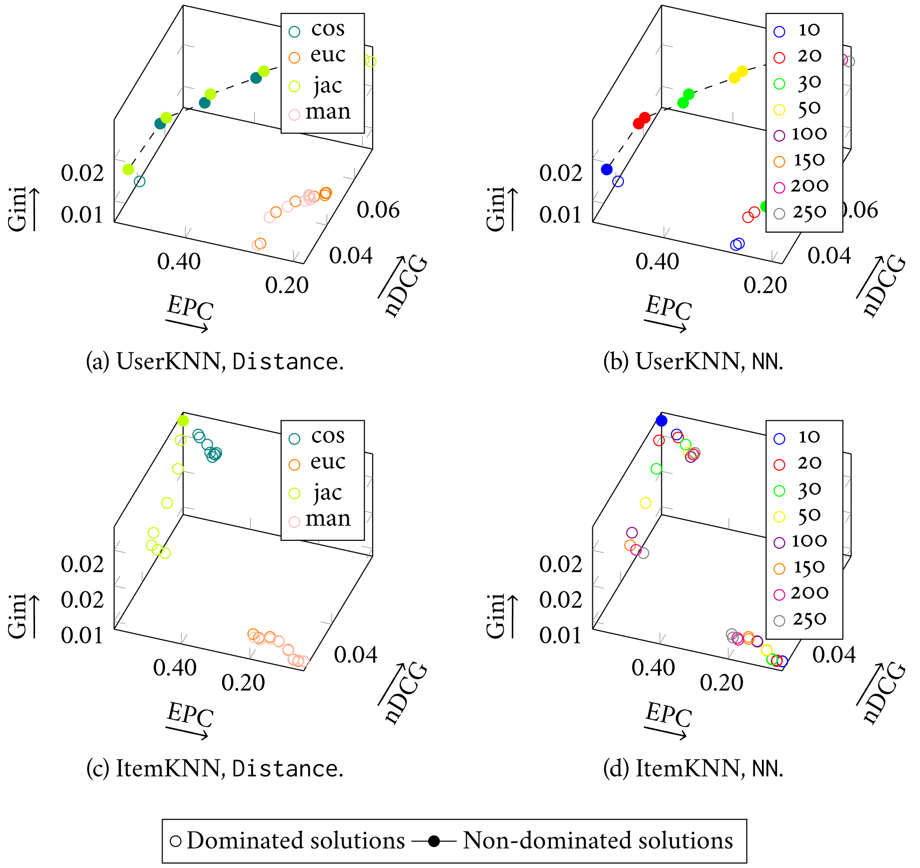
(a) BPRMF, `Factors`.

(b) BPRMF, `Learning Rate`.

(c) NeuMF, `Factors`.

(d) NeuMF, `Learning Rate`.

Figure A.17. Accuracy/Novelty/Diversity trade-offs on Amazon Music, assessed through *nDCG*/*EPC*/*Gini*, for BPRMF and NeuMF. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.



(a) UserKNN, `Distance`.

(b) UserKNN, `NN`.

(c) ItemKNN, `Distance`.

(d) ItemKNN, `NN`.

Figure A.18. Accuracy/Novelty/Diversity trade-offs on Amazon Music, assessed through *nDCG*/*EPC*/*Gini*, for UserKNN and ItemKNN. The cutoff is 10. Each point depicts a model hyper-parameter configuration set in the objective function space. The filled dots are on the Pareto frontier, while the empty dots are dominated points. Colors refer to a value of a selected hyper-parameter. Arrows indicates the optimization direction for each metric on x and y axes.

Bibliography

[1] Himan Abdollahpouri, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Augusto Pizzato. "Multistakeholder recommendation: Survey and research directions". In: *User Model. User Adapt. Interact.* 30.1 (2020), pp. 127–158. DOI: 10.1007/s11257-019-09256-1.

[2] Himan Abdollahpouri and Robin Burke. "Multistakeholder Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Springer US, 2022, pp. 647–677. DOI: 10.1007/978-1-0716-2197-4_17.

[3] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. "Controlling Popularity Bias in Learning-to-Rank Recommendation". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*. Ed. by Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, and Alexander Tuzhilin. ACM, 2017, pp. 42–46. DOI: 10.1145/3109859.3109912.

[4] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. "Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking". In: *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019*. Ed. by Roman Barták and Keith W. Brawner. AAAI Press, 2019, pp. 413–418.

[5] Himan Abdollahpouri, Mehdi Elahi, Masoud Mansoury, Shaghayegh Sahebi, Zahra Nazari, Allison Chaney, and Babak Loni. "MORS 2021: 1st Workshop on Multi-Objective Recommender Systems". In: *RecSys*. ACM, 2021, pp. 787–788.

[6] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. "The Unfairness of Popularity Bias in Recommendation". In: *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019*. Ed. by Robin Burke, Himan Abdollah-

pouri, Edward C. Malthouse, K. P. Thai, and Yongfeng Zhang. Vol. 2440. CEUR Workshop Proceedings. CEUR-WS.org, 2019.

[7] Himan Abdollahpouri, Shaghayegh Sahebi, Mehdi Elahi, Masoud Mansoury, Babak Loni, Zahra Nazari, and Maria Dimakopoulou. "MORS 2022: The Second Workshop on Multi-Objective Recommender Systems". In: *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*. Ed. by Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge. ACM, 2022, pp. 658–660. DOI: 10.1145/3523227.3547410.

[8] Gediminas Adomavicius and YoungOk Kwon. "Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques". In: *IEEE Trans. Knowl. Data Eng.* 24.5 (2012), pp. 896–911. DOI: 10.1109/TKDE.2011.15.

[9] Gediminas Adomavicius and Alexander Tuzhilin. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". In: *IEEE Trans. Knowl. Data Eng.* 17.6 (2005), pp. 734–749. DOI: 10.1109/TKDE.2005.99.

[10] Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. "Discrimination through Optimization: How Facebook's Ad Delivery Can Lead to Biased Outcomes". In: *Proc. ACM Hum. Comput. Interact.* 3.CSCW (2019), 199:1–199:30. DOI: 10.1145/3359301.

[11] Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. "Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation". In: *SIGIR*. ACM, 2021, pp. 2405–2414.

[12] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, Francesco Maria Donini, **Vincenzo Paparella**, and Claudio Pomo. "An Analysis of Local Explanation with LIME-RS". In: *Proceedings of the 12th Italian Information Retrieval Workshop 2022, Milan, Italy, June 29-30, 2022*. Ed. by Gabriella Pasi, Paolo Cremonesi, Salvatore Orlando, Markus Zanker, David Massimo, and Gloria Turati. Vol. 3177. CEUR Workshop Proceedings. CEUR-WS.org, 2022.

[13] 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: *UMAP '22: 30th ACM Conference on User Modeling, Adaptation and Personalization, Barcelona, Spain, July 4 - 7, 2022*. Ed. by Alejandro Bellogín, Ludovico Boratto, Olga C. Santos, Liliana Ardissono, and Bart P. Knijnenburg. ACM, 2022, pp. 121–131. DOI: 10.1145/3503252.3531292.

[14] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, and Claudio Pomo. "Reenvisioning the comparison between Neural Collaborative Filtering and Matrix Factorization". In: *RecSys*. ACM, 2021, pp. 521–529.

[15] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Daniele Malitesta, Vincenzo Paparella, and Claudio Pomo. “Auditing Consumer- and Producer-Fairness in Graph Collaborative Filtering”. In: *Advances in Information Retrieval - 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2-6, 2023, Proceedings, Part I*. Ed. by Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo. Vol. 13980. Lecture Notes in Computer Science. Springer, 2023, pp. 33–48. DOI: 10.1007/978-3-031-28244-7_3.

[16] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Eugenio Di Sciascio, Antonio Ferrara, Daniele Malitesta, and Claudio Pomo. “How Neighborhood Exploration influences Novelty and Diversity in Graph Collaborative Filtering”. In: *MORS@RecSys*. Vol. 3268. CEUR Workshop Proceedings. CEUR-WS.org, 2022.

[17] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Eugenio Di Sciascio, Antonio Ferrara, Daniele Malitesta, and Claudio Pomo. “Reshaping Graph Recommendation with Edge Graph Collaborative Filtering and Customer Reviews”. In: *DL4SR@CIKM*. Vol. 3317. CEUR Workshop Proceedings. CEUR-WS.org, 2022.

[18] Vito Walter Anelli, Daniele Malitesta, Claudio Pomo, Alejandro Bellogín, Eugenio Di Sciascio, and Tommaso Di Noia. “Challenging the Myth of Graph Collaborative Filtering: a Reasoned and Reproducibility-driven Analysis”. In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, 2023, pp. 350–361. DOI: 10.1145/3604915.3609489.

[19] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Antonio Ferrara, and Alberto Carlo Maria Mancino. “Sparse Feature Factorization for Recommender Systems with Knowledge Graphs”. In: *RecSys ’21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*. Ed. by Humberto Jesús Corona Pampín, Martha A. Larson, Martijn C. Willemsen, Joseph A. Konstan, Julian J. McAuley, Jean Garcia-Gathright, Bouke Huurnink, and Even Oldridge. ACM, 2021, pp. 154–165. DOI: 10.1145/3460231.3474243.

[20] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Claudio Pomo, and Azzurra Ragone. “On the discriminative power of hyper-parameters in cross-validation and how to choose them”. In: *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*. Ed. by Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk. ACM, 2019, pp. 447–451. DOI: 10.1145/3298689.3347010.

[21] Christine Bauer, Ben Carterette, Nicola Ferro, Norbert Fuhr, and Guglielmo Faggioli. "Frontiers of Information Access Experimentation for Research and Education (Dagstuhl Seminar 23031)". In: *Dagstuhl Reports* 13.1 (2023), pp. 68–154. DOI: 10.4230/DAGREP.13.1.68.

[22] Christine Bauer, Alan Said, and Eva Zangerle. "Evaluation Perspectives of Recommender Systems: Driving Research and Education (Dagstuhl Seminar 24211)". In: *Dagstuhl Reports* 14.5 (2024), pp. 58–172. DOI: 10.4230/DAGREP.14.5.58.

[23] Robert M. Bell and Yehuda Koren. "Lessons from the Netflix prize challenge". In: *SIGKDD Explor.* 9.2 (2007), pp. 75–79. DOI: 10.1145/1345448.1345465.

[24] Alejandro Bellogín, Pablo Castells, and Iván Cantador. "Statistical biases in Information Retrieval metrics for recommender systems". In: *Inf. Retr. J.* 20.6 (2017), pp. 606–634. DOI: 10.1007/S10791-017-9312-Z.

[25] Rianne van den Berg, Thomas N. Kipf, and Max Welling. "Graph Convolutional Matrix Completion". In: *CoRR* abs/1706.02263 (2017).

[26] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. "Equity of Attention: Amortizing Individual Fairness in Rankings". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. Ed. by Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz. ACM, 2018, pp. 405–414. DOI: 10.1145/3209978.3210063.

[27] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. "Fast Differentiable Sorting and Ranking". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 950–959.

[28] Georgios Boltsis and Evaggelia Pitoura. "Bias disparity in graph-based collaborative filtering recommenders". In: *SAC*. ACM, 2022, pp. 1403–1409.

[29] Ludovico Boratto, Gianni Fenu, and Mirko Marras. "Interplay between upsampling and regularization for provider fairness in recommender systems". In: *User Model. User Adapt. Interact.* 31.3 (2021), pp. 421–455. DOI: 10.1007/s11257-021-09294-8.

[30] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. "Finding Knees in Multi-objective Optimization". In: *Parallel Problem Solving from Nature - PPSN VIII, 8th International Conference, Birmingham, UK, September 18-22, 2004, Proceedings*. Ed. by Xin Yao, Edmund K. Burke, José Antonio Lozano, Jim Smith, Juan Julián Merelo Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiño, Ata Kabán, and Hans-Paul Schwefel. Vol. 3242. Lecture Notes in Computer Science. Springer, 2004, pp. 722–731. DOI: 10.1007/978-3-540-30217-9_73.

[31] Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, eds. *Multiobjective Optimization, Interactive and Evolutionary Approaches [outcome of Dagstuhl seminars]*. Vol. 5252. Lecture Notes in Computer Science. Springer, 2008. ISBN: 978-3-540-88907-6. DOI: 10.1007/978-3-540-88908-3.

[32] Sebastian Bruch, Masrour Zoghi, Michael Bendersky, and Marc Najork. "Revisiting Approximate Metric Optimization in the Age of Deep Neural Networks". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. Ed. by Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer. ACM, 2019, pp. 1241–1244. DOI: 10.1145/3331184.3331347.

[33] Christopher JC Burges. "From ranknet to lambdarank to lambdamart: An overview". In: *Learning* 11.23-581 (2010), p. 81.

[34] Robin Burke. "Multisided Fairness for Recommendation". In: *CoRR* abs/1707.00093 (2017). arXiv: 1707.00093.

[35] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. "Balanced Neighborhoods for Multi-sided Fairness in Recommendation". In: *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*. Ed. by Sorelle A. Friedler and Christo Wilson. Vol. 81. Proceedings of Machine Learning Research. PMLR, 2018, pp. 202–214.

[36] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. "Balanced neighborhoods for multi-sided fairness in recommendation". In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 202–214.

[37] Robin D. Burke. "Hybrid Web Recommender Systems". In: *The Adaptive Web, Methods and Strategies of Web Personalization*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Vol. 4321. Lecture Notes in Computer Science. Springer, 2007, pp. 377–408. DOI: 10.1007/978-3-540-72079-9_12.

[38] Fidel Cacheda, Victor Carneiro, Diego Fernández, and Vreixo Formoso. "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems". In: *ACM Trans. Web* 5.1 (2011), 2:1–2:33. DOI: 10.1145/1921591.1921593.

[39] California State Legislature. *The California Consumer Privacy Act of 2018*. 2018. URL: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

[40] Gabriele Capannini, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Nicola Tonellotto. "Quality Versus Efficiency in Document Scoring with Learning-to-rank Models". In: *Information Processing Management* 52.6 (Nov. 2016), pp. 1161–1177. ISSN: 0306-4573.

[41] David Carmel, Elad Haramaty, Arnon Lazerson, and Liane Lewin-Eytan. "Multi-Objective Ranking Optimization for Product Search Using Stochastic Label Aggregation". In: *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. Ed. by Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen. ACM / IW3C2, 2020, pp. 373–383. DOI: 10.1145/3366423.3380122.

[42] Pablo Castells, Neil Hurley, and Saúl Vargas. "Novelty and Diversity in Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Springer US, 2022, pp. 603–646. DOI: 10.1007/978-1-0716-2197-4_16.

[43] Zheng-Yi Chai, Ya-Lun Li, and Sifeng Zhu. "P-MOIA-RS: a multi-objective optimization and decision-making algorithm for recommendation systems". In: *J. Ambient Intell. Humaniz. Comput.* 12.1 (2021), pp. 443–454.

[44] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. "Measuring and Relieving the Over-Smoothing Problem for Graph Neural Networks from the Topological View". In: *AAAI*. AAAI Press, 2020, pp. 3438–3445.

[45] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. "Revisiting Graph Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach". In: *AAAI*. AAAI Press, 2020, pp. 27–34.

[46] David W. Corne, Nick R. Jerram, Joshua D. Knowles, and Martin J. Oates. "PESA-II: Region-Based Selection in Evolutionary Multiobjective Optimization". In: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. GECCO'01. San Francisco, California: Morgan Kaufmann Publishers Inc., 2001, pp. 283–290. ISBN: 1558607749.

[47] David W. Corne, Joshua D. Knowles, and Martin J. Oates. "The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization". In: *Parallel Problem Solving from Nature PPSN VI*. Ed. by Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 839–848.

[48] Paolo Cremonesi and Dietmar Jannach. "Progress in Recommender Systems Research: Crisis? What Crisis?" In: *AI Mag.* 42.3 (2021), pp. 43–54. DOI: 10.1609/AIMAG.V42I3.18145.

[49] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. "Performance of recommender algorithms on top-n recommendation tasks". In: *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*. Ed. by Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker. ACM, 2010, pp. 39–46. DOI: 10.1145/1864708.1864721.

[50] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research". In: *ACM Trans. Inf. Syst.* 39.2 (2021), 20:1–20:49. DOI: 10.1145/3434185.

[51] 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 2019, Copenhagen, Denmark, September 16-20, 2019*. Ed. by Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk. ACM, 2019, pp. 101–109. DOI: 10.1145/3298689.3347058.

[52] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197. DOI: 10.1109/4235.996017.

[53] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience series in systems and optimization. Wiley, 2001. ISBN: 978-0-471-87339-6.

[54] Kalyanmoy Deb and Shivam Gupta. "Understanding knee points in bicriteria problems and their implications as preferred solution principles". In: *Engineering Optimization* 43.11 (2011), pp. 1175–1204. DOI: 10.1080/0305215X.2010.548863. eprint: https://doi.org/10.1080/0305215X.2010.548863.

[55] Yashar Deldjoo, Vito Walter Anelli, Hamed Zamani, Alejandro Bellogín, and Tommaso Di Noia. "A flexible framework for evaluating user and item fairness in recommender systems". In: *User Model. User Adapt. Interact.* 31.3 (2021), pp. 457–511. DOI: 10.1007/S11257-020-09285-1.

[56] Jean-Antoine Désidéri. "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization". In: *Comptes Rendus Mathematique* 350.5 (2012), pp. 313–318. ISSN: 1631-073X. DOI: https://doi.org/10.1016/j.crma.2012.03.014.

[57] Jean-Antoine Désidéri. "Multiple-gradient Descent Algorithm for Pareto-Front Identification". In: *Modeling, Simulation and Optimization for Science and Technology*. Ed. by William Fitzgibbon, Yuri A. Kuznetsov, Pekka Neittaanmäki, and Olivier Pironneau. Vol. 34. Computational Methods in Applied Sciences. Springer, 2014, pp. 41–58. DOI: 10.1007/978-94-017-9054-3_3.

[58] Dario Di Palma, Vito Walter Anelli, Daniele Malitesta, Vincenzo Paparella, Claudio Pomo, Yashar Deldjoo, and Tommaso Di Noia. "Examining Fairness in Graph-Based Collaborative Filtering: A Consumer and Producer Perspective". In: *Proceedings of the 13th Italian Information Retrieval Workshop (IIR 2023), Pisa, Italy, June 8-9, 2023*. Ed. by Franco Maria Nardini, Nicola Tonellotto, Guglielmo Faggioli, and Antonio Ferrara. Vol. 3448. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 79–84.

[59] Michael D Ekstrand, Anubrata Das, Robin Burke, Fernando Diaz, et al. "Fairness in Information Access Systems". In: *Foundations and Trends® in Information Retrieval* 16.1-2 (2022), pp. 1–177.

[60] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. "Fairness in Information Access Systems". In: *Found. Trends Inf. Retr.* 16.1-2 (2022), pp. 1–177. DOI: 10.1561/1500000079.

[61] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. "Fairness in Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Springer US, 2022, pp. 679–707. DOI: 10.1007/978-1-0716-2197-4_18.

[62] Michael D. Ekstrand, John Riedl, and Joseph A. Konstan. "Collaborative Filtering Recommender Systems". In: *Found. Trends Hum. Comput. Interact.* 4.2 (2011), pp. 175–243.

[63] European Commission. *2018 reform of EU data protection rules*. 2018. URL: https://ec.europa.eu/info/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules/eu-data-protection-rules_en.

[64] Hui Fang, Xu Feng, Lu Qin, and Zhu Sun. "Towards Fair and Rigorous Evaluations: Hyperparameter Optimization for Top-N Recommendation Task with Implicit Feedback". In: *CoRR* abs/2408.07630 (2024). DOI: 10.48550/ARXIV.2408.07630. arXiv: 2408.07630.

[65] Matthias Feurer and Frank Hutter. "Hyperparameter Optimization". In: *Automated Machine Learning - Methods, Systems, Challenges*. Ed. by Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. The Springer Series on Challenges in Machine Learning. Springer, 2019, pp. 3–33. DOI: 10.1007/978-3-030-05318-5_1.

[66] M. Fleischer. "The Measure of Pareto Optima". In: *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 8-11, 2003, Proceedings*. Ed. by Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Vol. 2632. Lecture Notes in Computer Science. Springer, 2003, pp. 519–533. DOI: 10.1007/3-540-36970-8_37.

[67] Carlos M. Fonseca and Peter John Fleming. "Genetic Algorithms for Multiobjective Optimization: FormulationDiscussion and Generalization". In: *ICGA*. 1993.

[68] Reinaldo Silva Fortes, Daniel Xavier de Sousa, Dayanne Gouveia Coelho, Anísio Mendes Lacerda, and Marcos André Gonçalves. "Individualized extreme dominance (IndED): A new preference-based method for multi-objective recommender systems". In: *Inf. Sci.* 572 (2021), pp. 558–573. DOI: 10.1016/J.INS.2021.05.037.

[69] Zuohui Fu et al. "Fairness-Aware Explainable Recommendation over Knowledge Graphs". In: *SIGIR*. ACM, 2020, pp. 69–78.

[70] Yingqiang Ge, Xiaoting Zhao, Lucia Yu, Saurabh Paul, Diane Hu, Chu-Cheng Hsieh, and Yongfeng Zhang. "Toward Pareto Efficient Fairness-Utility Trade-off in Recommendation through Reinforcement Learning". In: *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*. Ed. by K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang. ACM, 2022, pp. 316–324. DOI: 10.1145/3488560.3498487.

[71] Bingrui Geng, Lingling Li, Licheng Jiao, Maoguo Gong, Qing Cai, and Yue Wu. "NNIA-RS: A multi-objective optimization based recommender system". In: *Physica A: Statistical Mechanics and its Applications* 424 (2015), pp. 383–397. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2015.01.007.

[72] Veronica Gil-Costa, Fernando Loor, Romina Molina, Franco Maria Nardini, Raffaele Perego, and Salvatore Trani. "Ensemble Model Compression for Fast and Energy-Efficient Ranking on FPGAs". In: *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part I*. Ed. by Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty. Vol. 13185. Lecture Notes in Computer Science. Springer, 2022, pp. 260–273. DOI: 10.1007/978-3-030-99736-6_18.

[73] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016. ISBN: 978-0-262-03561-3.

[74] Alexey Grishanov, Anastasia Ianina, and Konstantin V. Vorontsov. "Multiobjective Evaluation of Reinforcement Learning Based Recommender Systems". In: *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*. Ed. by Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge. ACM, 2022, pp. 622–627. DOI: 10.1145/3523227.3551485.

[75] Asela Gunawardana and Guy Shani. "Evaluating Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Springer, 2015, pp. 265–308. DOI: 10.1007/978-1-4899-7637-6_8.

[76] Mounir Hafsa, Pamela Wattebled, Julie Jacques, and Laetitia Jourdan. "A Multi-Objective E-learning Recommender System at Mandarine Academy". In: *Proceedings of the 2nd Workshop on Multi-Objective Recommender Systems co-located with 16th ACM Conference on Recommender Systems (RecSys 2022), Seattle, WA, USA, 18th-23rd September 2022*. Ed. by Himan Abdollahpouri, Shaghayegh Sahebi, Mehdi Elahi, Masoud Mansoury, Babak Loni, Zahra

Nazari, and Maria Dimakopoulou. Vol. 3268. CEUR Workshop Proceedings. CEUR-WS.org, 2022.

[77] Haimes, Lasdon, and Wismer. "On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-1.3 (1971), pp. 296–297. DOI: 10.1109/TSMC.1971.4308298.

[78] Christian Hansen, Rishabh Mehrotra, Casper Hansen, Brian Brost, Lucas Maystre, and Mounia Lalmas. "Shifting Consumption towards Diverse Content on Music Streaming Platforms". In: *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*. Ed. by Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich. ACM, 2021, pp. 238–246. DOI: 10.1145/3437963.3441775.

[79] Michael Pilegaard Hansen and Andrzej Jaszkiewicz. *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical Universityof Denmark, 1994.

[80] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of Opportunity in Supervised Learning". In: *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. Ed. by Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett. 2016, pp. 3315–3323.

[81] F. Maxwell Harper and Joseph A. Konstan. "The MovieLens Datasets: History and Context". In: *ACM Trans. Interact. Intell. Syst.* 5.4 (2016), 19:1–19:19. DOI: 10.1145/2827872.

[82] Ruining He and Julian J. McAuley. "Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering". In: *WWW*. ACM, 2016, pp. 507–517.

[83] Ruining He and Julian J. McAuley. "VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback". In: *AAAI*. AAAI Press, 2016, pp. 144–150.

[84] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. Ed. by Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu. ACM, 2020, pp. 639–648. DOI: 10.1145/3397271.3401063.

[85] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. "Neural Collaborative Filtering". In: *WWW*. ACM, 2017, pp. 173–182.

[86] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. "Evaluating collaborative filtering recommender systems". In: *ACM Trans. Inf. Syst.* 22.1 (2004), pp. 5–53. DOI: 10.1145/963770.963772.

[87] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. "Session-based Recommendations with Recurrent Neural Networks". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2016.

[88] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. "How to Specify a Reference Point in Hypervolume Calculation for Fair Performance Comparison". In: *Evol. Comput.* 26.3 (2018). DOI: 10.1162/evco_a_00226.

[89] Dietmar Jannach and Himan Abdollahpouri. "A survey on multi-objective recommender systems". In: *Frontiers in Big Data* 6 (2023). ISSN: 2624-909X. DOI: 10.3389/fdata.2023.1157899.

[90] Dietmar Jannach and Himan Abdollahpouri. "A survey on multi-objective recommender systems". In: *Frontiers Big Data* 6 (2024). DOI: 10.3389/FDATA.2023.1157899.

[91] Dietmar Jannach and Christine Bauer. "Escaping the McNamara Fallacy: Towards more Impactful Recommender Systems Research". In: *AI Mag.* 41.4 (2020), pp. 79–95. DOI: 10.1609/AIMAG.V41I4.5312.

[92] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. "What recommenders recommend: an analysis of recommendation biases and possible countermeasures". In: *User Model. User Adapt. Interact.* 25.5 (2015), pp. 427–491. DOI: 10.1007/s11257-015-9165-3.

[93] Dietmar Jannach and Markus Zanker. "Value and Impact of Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Springer US, 2022, pp. 519–546. DOI: 10.1007/978-1-0716-2197-4_14.

[94] Olivier Jeunen, Jatin Mandav, Ivan Potapov, Nakul Agarwal, Sourabh Vaid, Wenzhe Shi, and Aleksei Ustimenko. "Multi-Objective Recommendation via Multivariate Policy Learning". In: *CoRR* abs/2405.02141 (2024). DOI: 10.48550/ARXIV.2405.02141. arXiv: 2405.02141.

[95] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. "Do Loyal Users Enjoy Better Recommendations?: Understanding Recommender Accuracy from a Time Perspective". In: *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*. Ed. by Fabio Crestani, Gabriella Pasi, and Éric Gaussier. ACM, 2022, pp. 92–97. DOI: 10.1145/3539813.3545124.

[96] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. "Degenerate Feedback Loops in Recommender Systems". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019*. Ed. by Vincent Conitzer, Gillian K. Hadfield, and Shannon Vallor. ACM, 2019, pp. 383–390. DOI: 10.1145/3306618.3314288.

[97] Michael Jugovac, Dietmar Jannach, and Lukas Lerche. "Efficient optimization of multiple recommendation quality factors according to individual user tendencies". In: *Expert Syst. Appl.* 81 (2017), pp. 321–331. DOI: 10.1016/j.eswa.2017.03.055.

[98] Marius Kaminskas and Derek Bridge. "Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems". In: *ACM Trans. Interact. Intell. Syst.* 7.1 (2017), 2:1–2:42.

[99] Wang-Cheng Kang and Julian J. McAuley. "Self-Attentive Sequential Recommendation". In: *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 2018, pp. 197–206. DOI: 10.1109/ICDM.2018.00035.

[100] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 3149–3157. ISBN: 9781510860964.

[101] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *ICLR (Poster)*. OpenReview.net, 2017.

[102] Joshua D. Knowles and David Corne. "Properties of an adaptive archiving algorithm for storing nondominated vectors". In: *IEEE Trans. Evol. Comput.* 7.2 (2003), pp. 100–116. DOI: 10.1109/TEVC.2003.810755.

[103] Joshua D. Knowles and David W. Corne. "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy". In: *Evolutionary Computation* 8.2 (June 2000), pp. 149–172. ISSN: 1063-6560. DOI: 10.1162/106365600568167. eprint: https://direct.mit.edu/evco/article-pdf/8/2/149/1493180/106365600568167.pdf.

[104] Yehuda Koren. "Factorization meets the neighborhood: a multifaceted collaborative filtering model". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*. Ed. by Ying Li, Bing Liu, and Sunita Sarawagi. ACM, 2008, pp. 426–434. DOI: 10.1145/1401890.1401944.

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

[106] Walid Krichene and Steffen Rendle. "On Sampled Metrics for Item Recommendation". In: *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. Ed. by Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash. ACM, 2020, pp. 1748–1757. DOI: 10.1145/3394486.3403226.

[107] Charles R. Leake. "Multicriterion Decision in Management: Principles and Practice". In: *J. Oper. Res. Soc.* 52.5 (2001), p. 603. DOI: 10.1057/palgrave.jors.2601200.

[108] Cheng-Te Li, Cheng Hsu, and Yang Zhang. "FairSR: Fairness-aware Sequential Recommendation through Multi-Task Learning with Preference Graph Embeddings". In: *ACM Trans. Intell. Syst. Technol.* 13.1 (2022), 16:1–16:21.

[109] Miqing Li and Xin Yao. "Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey". In: *ACM Comput. Surv.* 52.2 (2019), 26:1–26:38. DOI: 10.1145/3300148.

[110] Miqing Li and Xin Yao. "Quality evaluation of solution sets in multiobjective optimisation: A survey". In: *ACM Computing Surveys (CSUR)* 52.2 (2019), pp. 1–38.

[111] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. "User-oriented Fairness in Recommendation". In: *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021.* Ed. by Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia. ACM / IW3C2, 2021, pp. 624–632. DOI: 10.1145/3442381.3449866.

[112] 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 on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018.* Ed. by Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis. ACM, 2018, pp. 689–698. DOI: 10.1145/3178876.3186150.

[113] M. Lightner and S. Director. "Multiple criterion optimization for the design of electronic circuits". In: *IEEE Transactions on Circuits and Systems* 28.3 (1981), pp. 169–179. DOI: 10.1109/TCS.1981.1084969.

[114] Xiao Lin, Hongjie Chen, Changhua Pei, Fei Sun, Xuanji Xiao, Hanxiao Sun, Yongfeng Zhang, Wenwu Ou, and Peng Jiang. "A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation". In: *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.* Ed. by Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk. ACM, 2019, pp. 20–28. DOI: 10.1145/3298689.3346998.

[115] Greg Linden, Brent Smith, and Jeremy York. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering". In: *IEEE Internet Comput.* 7.1 (2003), pp. 76–80. DOI: 10.1109/MIC.2003.1167344.

[116] Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, Nicola Tonellotto, and Rossano Venturini. "QuickScorer: A Fast Algorithm to Rank Documents with Additive Ensembles of Regression Trees". In: *Proc. ACM SIGIR.* 2015, pp. 73–82.

[117] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. “Disentangled Graph Convolutional Networks”. In: *ICML*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 4212–4221.

[118] Debabrata Mahapatra and Vaibhav Rajan. “Multi-Task Learning with User Preferences: Gradient Descent with Controlled Ascent in Pareto Optimization”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 6597–6607.

[119] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. “FairMatch: A Graph-based Approach for Improving Aggregate Diversity in Recommender Systems”. In: *UMAP*. ACM, 2020, pp. 154–162.

[120] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. “A Graph-Based Approach for Mitigating Multi-Sided Exposure Bias in Recommender Systems”. In: *ACM Trans. Inf. Syst.* 40.2 (2022), 32:1–32:31.

[121] Masoud Mansoury, Bamshad Mobasher, Robin Burke, and Mykola Pechenizkiy. “Bias Disparity in Collaborative Recommendation: Algorithmic Evaluation and Comparison”. In: *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019*. Ed. by Robin Burke, Himan Abdollahpouri, Edward C. Malthouse, K. P. Thai, and Yongfeng Zhang. Vol. 2440. CEUR Workshop Proceedings. CEUR-WS.org, 2019.

[122] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. “SimpleX: A Simple and Strong Baseline for Collaborative Filtering”. In: *CIKM ’21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. Ed. by Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong. ACM, 2021, pp. 1243–1252. DOI: 10.1145/3459637.3482297.

[123] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. “UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation”. In: *CIKM*. ACM, 2021, pp. 1253–1262.

[124] R. Marler and Jasbir Arora. “Survey of Multi-Objective Optimization Methods for Engineering”. In: *Structural and Multidisciplinary Optimization* 26 (Apr. 2004), pp. 369–395. DOI: 10.1007/s00158-003-0368-6.

[125] Pawel Matuszyk, Renê Tatua Castillo, Daniel Kottke, and Myra Spiliopoulou. “A comparative study on hyperparameter optimization for recommender systems”. In: *Workshop on Recommender Systems and Big Data Analytics (RS-BDA’16) 2016*. 2016, pp. 13–21.

[126] Sean M. McNee, John Riedl, and Joseph A. Konstan. “Being accurate is not enough: how accuracy metrics have hurt recommender systems”. In: *Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*. Ed. by Gary M. Olson and Robin Jeffries. ACM, 2006, pp. 1097–1101. DOI: 10.1145/1125451.1125659.

[127] Kaisa Miettinen. *Nonlinear multiobjective optimization*. Vol. 12. International series in operations research and management science. Kluwer, 1998. ISBN: 978-0-7923-8278-2.

[128] Pasquale Minervini, Luca Franceschi, and Mathias Niepert. “Adaptive Perturbation-Based Gradient Estimation for Discrete Latent Variable Models”. In: *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*. Ed. by Brian Williams, Yiling Chen, and Jennifer Neville. AAAI Press, 2023, pp. 9200–9208. DOI: 10.1609/AAAI.V37I8.26103.

[129] Marta Moscati, Yashar Deldjoo, Giulio Davide Carparelli, and Markus Schedl. “Multiobjective Hyperparameter Optimization of Recommender Systems”. In: *Proceedings of the 3rd Workshop Perspectives on the Evaluation of Recommender Systems 2023 co-located with the 17th ACM Conference on Recommender Systems (RecSys 2023), Singapore, Singapore, September 19, 2023*. Ed. by Alan Said, Eva Zangerle, and Christine Bauer. Vol. 3476. CEUR Workshop Proceedings. CEUR-WS.org, 2023.

[130] Hossam Mossalam, Yannis M. Assael, Diederik M. Roijers, and Shimon Whiteson. “Multi-Objective Deep Reinforcement Learning”. In: *CoRR* abs/1610.02707 (2016). arXiv: 1610.02707.

[131] Mohammadmehdi Naghiaei, Hossein A. Rahmani, and Yashar Deldjoo. “CP-Fair: Personalized Consumer and Producer Fairness Re-ranking for Recommender Systems”. In: *SIGIR ’22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*. Ed. by Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai. ACM, 2022, pp. 770–779. DOI: 10.1145/3477495.3531959.

[132] Franco Maria Nardini, Cosimo Rulli, Salvatore Trani, and Rossano Venturini. “Distilled Neural Networks for Efficient Learning to Rank”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[133] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren G. Terveen, and Joseph A. Konstan. “Exploring the filter bubble: the effect of using recommender systems on content diversity”. In: *23rd International World Wide Web Conference, WWW ’14, Seoul, Republic of Korea, April 7-11, 2014*. Ed. by Chin-

Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel. ACM, 2014, pp. 677–686. DOI: 10.1145/2566486.2568012.

[134] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. "Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Association for Computational Linguistics, 2019, pp. 188–197. DOI: 10.18653/V1/D19-1018.

[135] Tommaso Di Noia, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. "Adaptive multi-attribute diversity for recommender systems". In: *Inf. Sci.* 382-383 (2017), pp. 234–253. DOI: 10.1016/j.ins.2016.11.015.

[136] Tommaso Di Noia, Nava Tintarev, Panagiota Fatourou, and Markus Schedl. "Recommender systems under European AI regulations". In: *Commun. ACM* 65.4 (2022), pp. 69–73. DOI: 10.1145/3512728.

[137] Jinoh Oh, Sun Park, Hwanjo Yu, Min Song, and Seung-Taek Park. "Novel Recommendation Based on Personal Popularity Tendency". In: *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*. Ed. by Diane J. Cook, Jian Pei, Wei Wang, Osmar R. Zaöane, and Xindong Wu. IEEE Computer Society, 2011, pp. 507–516. DOI: 10.1109/ICDM.2011.110.

[138] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 8024–8035.

[139] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. "Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications". In: *ACM Trans. Interact. Intell. Syst.* 7.1 (2017), 1:1–1:34.

[140] Michael J. Pazzani and Daniel Billsus. "Content-Based Recommendation Systems". In: *The Adaptive Web: Methods and Strategies of Web Personalization*. Ed. by Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 325–341. ISBN: 978-3-540-72079-9. DOI: 10.1007/978-3-540-72079-9_10.

[141] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. "SVD-GCN: A Simplified Graph Convolution Paradigm for Recommendation". In: *CIKM*. ACM, 2022, pp. 1625–1634.

[142] Przemyslaw Pobrotyn and Radoslaw Bialobrzeski. "NeuralNDCG: Direct Optimisation of a Ranking Metric via Differentiable Relaxation of Sorting". In: *CoRR* abs/2102.07831 (2021). arXiv: 2102.07831.

[143] Tao Qin and Tie-Yan Liu. "Introducing LETOR 4.0 Datasets". In: *CoRR* abs/1306.2597 (2013). arXiv: 1306.2597.

[144] Tao Qin, Tie-Yan Liu, and Hang Li. "A general approximation framework for direct optimization of information retrieval measures". In: *Inf. Retr.* 13.4 (2010), pp. 375–397. DOI: 10.1007/S10791-009-9124-X.

[145] Massimo Quadrana, Antoine Larreche-Mouly, and Matthias Mauch. "Multi-objective Hyper-parameter Optimization of Behavioral Song Embeddings". In: *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*. Ed. by Preeti Rao, Hema A. Murthy, Ajay Srinivasamurthy, Rachel M. Bittner, Rafael Caro Repetto, Masataka Goto, Xavier Serra, and Marius Miron. 2022, pp. 437–445.

[146] Tahleen A. Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. "Fairwalk: Towards Fair Graph Embedding". In: *IJCAI*. ijcai.org, 2019, pp. 3289–3295.

[147] Hossein A. Rahmani, Mohammadmehdi Naghiaei, Mahdi Dehghan, and Mohammad Aliannejadi. "Experiments on Generalizability of User-Oriented Fairness in Recommender Systems". In: *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*. Ed. by Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai. ACM, 2022, pp. 2755–2764. DOI: 10.1145/3477495.3531718.

[148] Hossein A. Rahmani, Mohammadmehdi Naghiaei, and Yashar Deldjoo. "A Personalized Framework for Consumer and Producer Group Fairness Optimization in Recommender Systems". In: *Trans. Recomm. Syst.* 2.3 (2024), 19:1–19:24. DOI: 10.1145/3651167.

[149] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. "BPR: Bayesian Personalized Ranking from Implicit Feedback". In: *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*. Ed. by Jeff A. Bilmes and Andrew Y. Ng. AUAI Press, 2009, pp. 452–461.

[150] Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. "Neural Collaborative Filtering vs. Matrix Factorization Revisited". In: *RecSys*. ACM, 2020, pp. 240–248.

[151] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". In: *CSCW '94, Proceedings of the Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, October 22-26, 1994*. Ed. by John B. Smith, F. Donelson Smith, and Thomas W. Malone. ACM, 1994, pp. 175–186. DOI: 10.1145/192844.192905.

[152] Marco Túlio Ribeiro, Anísio Lacerda, Adriano Veloso, and Nivio Ziviani. "Pareto-efficient hybridization for multi-objective recommender systems". In: *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*. Ed. by Padraig Cunningham, Neil J. Hurley, Ido Guy, and Sarabjot Singh Anand. ACM, 2012, pp. 19–26. DOI: 10.1145/2365952.2365962.

[153] Francesco Ricci, Lior Rokach, and Bracha Shapira, eds. *Recommender Systems Handbook*. Springer US, 2022. ISBN: 978-1-0716-2196-7. DOI: 10.1007/978-1-0716-2197-4.

[154] Mario Rodríguez, Christian Posse, and Ethan Zhang. "Multiple objective optimization in recommender systems". In: *RecSys*. ACM, 2012, pp. 11–18.

[155] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. "Search Result Diversification". In: *Foundations and Trends® in Information Retrieval* 9.1 (2015), pp. 1–90. ISSN: 1554-0669. DOI: 10.1561/1500000040.

[156] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. "Item-based collaborative filtering recommendation algorithms". In: *WWW*. ACM, 2001, pp. 285–295.

[157] Jason R Schott. *Fault tolerant design using single and multicriteria genetic algorithm optimization.* Tech. rep. Air force inst of tech Wright-Patterson afb OH, 1995.

[158] Ozan Sener and Vladlen Koltun. "Multi-Task Learning as Multi-Objective Optimization". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. Ed. by Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett. 2018, pp. 525–536.

[159] Daniele De Sensi, Massimo Torquati, and Marco Danelutto. "Mammut: High-level management of system knobs and sensors". In: *SoftwareX* 6 (2017), pp. 150–154. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2017.06.005.

[160] Guy Shani and Asela Gunawardana. "Evaluating Recommendation Systems". In: *Recommender Systems Handbook*. Springer, 2011, pp. 257–297.

[161] Shubhkirti Sharma and Vijay Kumar. "A Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future". In: *Archives of Computational Methods in Engineering* 29.7 (Nov. 2022), pp. 5605–5633. ISSN: 1886-1784. DOI: 10.1007/s11831-022-09778-9.

[162] Faisal Shehzad and Dietmar Jannach. “Everyone’s a Winner! On Hyperparameter Tuning of Recommendation Models”. In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, 2023, pp. 652–657. DOI: 10.1145/3604915.3609488.

[163] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, Khaled B. Letaief, and Dongsheng Li. “How Powerful is Graph Convolution for Recommendation?” In: *CIKM*. ACM, 2021, pp. 1619–1629.

[164] Ashudeep Singh and Thorsten Joachims. “Fairness of Exposure in Rankings”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. Ed. by Yike Guo and Faisal Farooq. ACM, 2018, pp. 2219–2228. DOI: 10.1145/3219819.3220088.

[165] Nasim Sonboli, Robin Burke, Michael D. Ekstrand, and Rishabh Mehrotra. “The Multisided Complexity of Fairness in Recommender Systems”. In: *AI Mag.* 43.2 (2022), pp. 164–176. DOI: 10.1002/AAAI.12054.

[166] Paolo Sorino, **Vincenzo Paparella**, Domenico Lofù, Tommaso Colafiglio, Eugenio Di Sciascio, Fedelucio Narducci, Rodolfo Sardone, and Tommaso Di Noia. “A Pareto-Optimality-Based Approach for Selecting the Best Machine Learning Models in Mild Cognitive Impairment Prediction”. In: *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2023, Honolulu, Oahu, HI, USA, October 1-4, 2023*. IEEE, 2023, pp. 3822–3827. DOI: 10.1109/SMC53992.2023.10394057.

[167] N. Srinivas and Kalyanmoy Deb. “Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms”. In: *Evolutionary Computation* 2.3 (1994), pp. 221–248. DOI: 10.1162/evco.1994.2.3.221.

[168] Dusan Stamenkovic, Alexandros Karatzoglou, Ioannis Arapakis, Xin Xin, and Kleomenis Katevas. “Choosing the Best of Both Worlds: Diverse and Novel Recommendations through Multi-Objective Reinforcement Learning”. In: *WSDM ’22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*. Ed. by K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang. ACM, 2022, pp. 957–965. DOI: 10.1145/3488560.3498471.

[169] Alain Starke, Sanne Vrijenhoek, Lien Michiels, Johannes Kruse, and Nava Tintarev. “NORMalize 2024: The Second Workshop on Normative Design and Evaluation of Recommender Systems”. In: *Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024, Bari, Italy, October 14-18, 2024*. Ed. by Tommaso Di Noia, Pasquale Lops, Thorsten Joachims, Katrien Verbert, Pablo Castells, Zhenhua Dong, and Ben London. ACM, 2024, pp. 1242–1244. DOI: 10.1145/3640457.3687103.

[170] Harald Steck. “Calibrated recommendations”. In: *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*. Ed. by Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O’Donovan. ACM, 2018, pp. 154–162. DOI: 10.1145/3240323.3240372.

[171] Harald Steck. “Embarrassingly Shallow Autoencoders for Sparse Data”. In: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. Ed. by Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia. ACM, 2019, pp. 3251–3257. DOI: 10.1145/3308558.3313710.

[172] 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 2023, Taipei, Taiwan, July 23-27, 2023*. Ed. by Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete. ACM, 2023, pp. 2629–2638. DOI: 10.1145/3539618.3591931.

[173] Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. “HGCF: Hyperbolic Graph Convolution Networks for Collaborative Filtering”. In: *WWW*. ACM / IW3C2, 2021, pp. 593–601.

[174] Jianing Sun et al. “A Framework for Recommending Accurate and Diverse Items Using Bayesian Graph Convolutional Neural Networks”. In: *KDD*. ACM, 2020, pp. 2030–2039.

[175] 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”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 45.7 (2023), pp. 8206–8226. DOI: 10.1109/TPAMI.2022.3231891.

[176] 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: *RecSys*. ACM, 2020, pp. 23–32.

[177] Yan-Martin Tamm, Rinchin Damdinov, and Alexey Vasilev. “Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently?” In: *RecSys ’21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*. Ed. by Humberto Jesús Corona Pampín, Martha A. Larson, Martijn C. Willemsen, Joseph A. Konstan, Julian J. McAuley, Jean Garcia-Gathright, Bouke Huurnink, and Even Oldridge. ACM, 2021, pp. 708–713. DOI: 10.1145/3460231.3478848.

[178] Hao Tang, Guoshuai Zhao, Yujiao He, Yuxia Wu, and Xueming Qian. “Ranking-based contrastive loss for recommendation systems”. In: *Knowl. Based Syst.* 261 (2023), p. 110180. DOI: 10.1016/J.KNOSYS.2022.110180.

[179] Jiaxi Tang and Ke Wang. "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*. Ed. by Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek. ACM, 2018, pp. 565–573. doi: 10.1145/3159652.3159656.

[180] Zhulin Tao, Yinwei Wei, Xiang Wang, Xiangnan He, Xianglin Huang, and Tat-Seng Chua. "MGAT: Multimodal Graph Attention Network for Recommendation". In: *Inf. Process. Manag.* 57.5 (2020), p. 102277.

[181] Matús Tomlein, Branislav Pecher, Jakub Simko, Ivan Srba, Róbert Móro, Elena Stefancova, Michal Kompan, Andrea Hrckova, Juraj Podrouzek, and Mária Bieliková. "An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting and Recent Behavior Changes". In: *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*. Ed. by Humberto Jesús Corona Pampín, Martha A. Larson, Martijn C. Willemsen, Joseph A. Konstan, Julian J. McAuley, Jean Garcia-Gathright, Bouke Huurnink, and Even Oldridge. ACM, 2021, pp. 1–11. doi: 10.1145/3460231.3474241.

[182] Virginia Tsintzou, Evaggelia Pitoura, and Panayiotis Tsaparas. "Bias Disparity in Recommendation Systems". In: *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019*. Ed. by Robin Burke, Himan Abdollahpouri, Edward C. Malthouse, K. P. Thai, and Yongfeng Zhang. Vol. 2440. CEUR Workshop Proceedings. CEUR-WS.org, 2019.

[183] David Allen Van Veldhuizen. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Air Force Institute of Technology, 1999.

[184] Saúl Vargas. "Novelty and diversity enhancement and evaluation in recommender systems and information retrieval". In: *SIGIR*. ACM, 2014, p. 1281.

[185] Saul Vargas and Pablo Castells. "Rank and relevance in novelty and diversity metrics for recommender systems". In: *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*. Ed. by Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius. ACM, 2011, pp. 109–116.

[186] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In: *ICLR (Poster)*. OpenReview.net, 2018.

[187] **Vincenzo Paparella**. "Pursuing Optimal Trade-Off Solutions in Multi-Objective Recommender Systems". In: *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*. Ed. by Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge. ACM, 2022, pp. 727–729. DOI: 10.1145/3523227.3547425.

[188] **Vincenzo Paparella**, Vito Walter Anelli, Ludovico Boratto, and Tommaso Di Noia. "Reproducibility of Multi-Objective Reinforcement Learning Recommendation: Interplay between Effectiveness and Beyond-Accuracy Perspectives". In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, 2023, pp. 467–478. DOI: 10.1145/3604915.3609493.

[189] **Vincenzo Paparella**, Vito Walter Anelli, Franco Maria Nardini, Raffaele Perego, and Tommaso Di Noia. "Post-hoc Selection of Pareto-Optimal Solutions in Search and Recommendation". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*. Ed. by Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos. ACM, 2023, pp. 2013–2023. DOI: 10.1145/3583780.3615010.

[190] **Vincenzo Paparella**, Dario Di Palma, Vito Walter Anelli, Alessandro De Bellis, and Tommaso Di Noia. "Unveiling the Potential of Recommender Systems through Multi-Objective Metrics". In: *Proceedings of the 14th Italian Information Retrieval Workshop, Udine, Italy, September 5-6, 2024*. Ed. by Kevin Roitero, Marco Viviani, Eddy Maddalena, and Stefano Mizzaro. Vol. 3802. CEUR Workshop Proceedings. CEUR-WS.org, 2024, pp. 119–122.

[191] **Vincenzo Paparella**, Dario Di Palma, Vito Walter Anelli, and Tommaso Di Noia. "Broadening the Scope: Evaluating the Potential of Recommender Systems beyond prioritizing Accuracy". In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, 2023, pp. 1139–1145. DOI: 10.1145/3604915.3610649.

[192] **Vincenzo Paparella**, Alberto Carlo Maria Mancino, Antonio Ferrara, Claudio Pomo, Vito Walter Anelli, and Tommaso Di Noia. "Knowledge Graph Datasets for Recommendation". In: *Proceedings of the Fifth Knowledge-aware and Conversational Recommender Systems Workshop co-located with 17th ACM Conference on Recommender Systems (RecSys 2023), Singapore, September 19th, 2023*. Ed. by Vito Walter Anelli, Pierpaolo Basile, Gerard de Melo, Francesco Maria Donini, Antonio Ferrara, Cataldo Musto, Fedelucio Narducci, Az-

zurra Ragone, and Markus Zanker. Vol. 3560. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 109–117.

[193] Michael Matthias Voit and Heiko Paulheim. "Bias in Knowledge Graphs - An Empirical Study with Movie Recommendation and Different Language Editions of DBpedia". In: *LDK*. Vol. 93. OASIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 14:1–14:13.

[194] Sanne Vrijenhoek, Lien Michiels, Johannes Kruse, Alain Starke, Nava Tintarev, and Jordi Viader Guerrero. "NORMalize: The First Workshop on Normative Design and Evaluation of Recommender Systems". In: *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. Ed. by Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song. ACM, 2023, pp. 1252–1254. DOI: 10.1145/3604915.3608757.

[195] D. Wallach and B. Goffinet. "Mean squared error of prediction as a criterion for evaluating and comparing system models". In: *Ecological Modelling* 44.3 (1989), pp. 299–306. ISSN: 0304-3800. DOI: https://doi.org/10.1016/0304-3800(89)90035-5.

[196] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. "Fine-Grained Spoiler Detection from Large-Scale Review Corpora". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 2605–2610. DOI: 10.18653/v1/p19-1248.

[197] Nan Wang, Lu Lin, Jundong Li, and Hongning Wang. "Unbiased Graph Embedding with Biased Graph Observations". In: *WWW*. ACM, 2022, pp. 1423–1433.

[198] Shanfeng Wang, Maoguo Gong, Haoliang Li, and Junwei Yang. "Multi-objective optimization for long tail recommendation". In: *Knowl. Based Syst.* 104 (2016), pp. 145–155. DOI: 10.1016/j.knosys.2016.04.018.

[199] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. "KGAT: Knowledge Graph Attention Network for Recommendation". In: *KDD*. ACM, 2019, pp. 950–958.

[200] 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 2019, Paris, France, July 21-25, 2019*. Ed. by Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer. ACM, 2019, pp. 165–174. DOI: 10.1145/3331184.3331267.

[201] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. "Disentangled Graph Collaborative Filtering". In: *SIGIR*. ACM, 2020, pp. 1001–1010.

[202] Yifan Wang, Peijie Sun, Weizhi Ma, Min Zhang, Yuan Zhang, Peng Jiang, and Shaoping Ma. "Intersectional Two-sided Fairness in Recommendation". In: *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*. Ed. by Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee. ACM, 2024, pp. 3609–3620. DOI: 10.1145/3589334.3645518.

[203] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. "DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation". In: *CIKM*. ACM, 2020, pp. 1605–1614.

[204] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. "Multi-FR: A Multi-objective Optimization Framework for Multi-stakeholder Fairness-aware Recommendation". In: *Transactions on Information Systems (TOIS)*. ACM, 2022.

[205] Haolun Wu, Chen Ma, Bhaskar Mitra, Fernando Diaz, and Xue Liu. "A Multi-Objective Optimization Framework for Multi-Stakeholder Fairness-Aware Recommendation". In: *ACM Trans. Inf. Syst.* 41.2 (2023), 47:1–47:29. DOI: 10.1145/3564285.

[206] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. "Self-supervised Graph Learning for Recommendation". In: *SIGIR*. ACM, 2021, pp. 726–735.

[207] Junkang Wu, Wentao Shi, Xuezhi Cao, Jiawei Chen, Wenqiang Lei, Fuzheng Zhang, Wei Wu, and Xiangnan He. "DisenKGAT: Knowledge Graph Embedding with Disentangled Graph Attention Network". In: *CIKM*. ACM, 2021, pp. 2140–2149.

[208] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. "Learning Fair Representations for Recommendation: A Graph-based Perspective". In: *WWW*. ACM / IW3C2, 2021, pp. 2198–2208.

[209] Q. Wu, C.J.C. Burges, K.M. Svore, and J. Gao. "Adapting boosting for information retrieval measures". In: *Information Retrieval* (2010).

[210] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. "TFROM: A Two-sided Fairness-Aware Recommendation Model for Both Customers and Providers". In: *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. Ed. by Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai. ACM, 2021, pp. 1013–1022. DOI: 10.1145/3404835.3462882.

[211] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. "Collaborative Denoising Auto-Encoders for Top-N Recommender Systems". In: *WSDM*. ACM, 2016, pp. 153–162.

[212] Ruobing Xie, Yanlei Liu, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. "Personalized Approximate Pareto-Efficient Recommendation". In: *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. Ed. by Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia. ACM / IW3C2, 2021, pp. 3839–3849. DOI: 10.1145/3442381.3450039.

[213] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. "Self-Supervised Reinforcement Learning for Recommender Systems". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. Ed. by Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu. ACM, 2020, pp. 931–940. DOI: 10.1145/3397271.3401147.

[214] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. "Challenging the Long Tail Recommendation". In: *Proc. VLDB Endow.* 5.9 (2012), pp. 896–907.

[215] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. "Graph Convolutional Neural Networks for Web-Scale Recommender Systems". In: *KDD*. ACM, 2018, pp. 974–983.

[216] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. "A Simple Convolutional Generative Network for Next Item Recommendation". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*. Ed. by J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman. ACM, 2019, pp. 582–590. DOI: 10.1145/3289600.3290975.

[217] Yv Haimes Yv, Leon S. Lasdon, and Dang Da. "On a bicriterion formation of the problems of integrated system identification and system optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics* (1971), pp. 296–297.

[218] Fatima Ezzahra Zaizi, Sara Qassimi, and Said Rakrak. "Multi-objective optimization with recommender systems: A systematic review". In: *Information Systems* 117 (2023), p. 102233. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2023.102233.

[219] Fatima Ezzahra Zaizi, Sara Qassimi, and Said Rakrak. "Multi-objective optimization with recommender systems: A systematic review". In: *Inf. Syst.* 117 (2023), p. 102233. DOI: 10.1016/J.IS.2023.102233.

[220] Eva Zangerle and Christine Bauer. "Evaluating Recommender Systems: Survey and Framework". In: *ACM Comput. Surv.* 55.8 (2023), 170:1–170:38. DOI: 10.1145/3556536.

[221] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. "Mitigating Unwanted Biases with Adversarial Learning". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*. Ed. by Jason Furman, Gary E. Marchant, Huw Price, and Francesca Rossi. ACM, 2018, pp. 335–340. DOI: 10.1145/3278721.3278779.

[222] Minghao Zhao, Le Wu, Yile Liang, Lei Chen, Jian Zhang, Qilin Deng, Kai Wang, Xudong Shen, Tangjie Lv, and Runze Wu. "Investigating Accuracy-Novelty Performance for Graph-based Collaborative Filtering". In: *SIGIR*. ACM, 2022, pp. 50–59.

[223] 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". In: *ACM Trans. Inf. Syst.* 41.2 (2023), 32:1–32:41. DOI: 10.1145/3545796.

[224] Yong Zheng and David (Xuejun) Wang. "A survey of recommender systems with multi-objective optimization". In: *Neurocomputing* 474 (2022), pp. 141–153. DOI: 10.1016/j.neucom.2021.11.041.

[225] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. "DGCN: Diversified Recommendation with Graph Convolutional Networks". In: *WWW*. ACM / IW3C2, 2021, pp. 401–412.

[226] Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. "Towards Deeper Graph Neural Networks with Differentiable Group Normalization". In: *NeurIPS*. 2020.

[227] Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. "BARS: Towards Open Benchmarking for Recommender Systems". In: *SIGIR*. ACM, 2022, pp. 2912–2923.

[228] Ziwei Zhu, Jianling Wang, and James Caverlee. "Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems". In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. Ed. by Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu. ACM, 2020, pp. 449–458. DOI: 10.1145/3397271.3401177.

[229] E. Zitzler and L. Thiele. "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach". In: *IEEE Transactions on Evolutionary Computation* 3.4 (1999), pp. 257–271. DOI: 10.1109/4235.797969.

[230] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. "The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration". In: *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007, Proceedings*. Ed. by Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and

Tadahiko Murata. Vol. 4403. Lecture Notes in Computer Science. Springer, 2007, pp. 862–876. DOI: 10.1007/978-3-540-70928-2_64.

[231] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. "Comparison of multiobjective evolutionary algorithms: Empirical results". In: *Evolutionary computation* 8.2 (2000), pp. 173–195.

[232] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization". In: vol. 3242. Jan. 2001.

[233] Eckart Zitzler and Lothar Thiele. "Multiobjective optimization using evolutionary algorithms—a comparative case study". In: *Parallel Problem Solving from Nature—PPSN V: 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings 5*. Springer. 1998, pp. 292–301.

UNIONE EUROPEA
Fondo Sociale Europeo

Ministero dell'Università
e della Ricerca

PON
RICERCA
E INNOVAZIONE
2014 - 2020

REACT EU

La borsa di dottorato è stata cofinanziata con risorse del
Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, risorse FSE REACT-EU
Azione IV.4 "Dottorati e contratti di ricerca su tematiche dell'innovazione"
e Azione IV.5 "Dottorati su tematiche Green"