Université de Montréal

Towards Privacy-Preserving and Fairness-Enhanced Item Ranking in Recommender Systems

par

Jia Ao Sun

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

July 8, 2023

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Towards Privacy-Preserving and Fairness-Enhanced Item Ranking in Recommender Systems

présenté par

Jia Ao Sun

a été évalué par un jury composé des personnes suivantes :

Michel Boyer
(président-rapporteur)

Esma Aïmeur
(directeur de recherche)

Golnoosh Farnadi
(codirecteur)

Frédéric Dupont-Dupuis
(membre du jury)

Résumé

Nous présentons une nouvelle approche de préservation de la vie privée pour améliorer l'équité des éléments dans les systèmes de classement. Nous utilisons des techniques de post-traitement dans un environnement de recommandation multipartite afin d'équilibrer l'équité et la protection de la vie privée pour les producteurs et les consommateurs. Notre méthode utilise des serveurs de calcul multipartite sécurisés (MPC) et une confidentialité différentielle (DP) pour maintenir la confidentialité des utilisateurs tout en atténuant l'injustice des éléments sans compromettre l'utilité. Les utilisateurs soumettent leurs données sous forme de partages secrets aux serveurs MPC, et tous les calculs sur ces données restent cryptés. Nous évaluons notre approche à l'aide d'ensembles de données du monde réel, tels qu'Amazon Digital Music, Book Crossing et MovieLens-1M, et analysons les compromis entre confidentialité, équité et utilité. Notre travail encourage une exploration plus approfondie de l'intersection de la confidentialité et de l'équité dans les systèmes de recommandation, jetant les bases de l'intégration d'autres techniques d'amélioration de la confidentialité afin d'optimiser l'exécution et l'évolutivité pour les applications du monde réel. Nous envisageons notre approche comme un tremplin vers des solutions de bout en bout préservant la confidentialité et promouvant l'équité dans des environnements de recommandation multipartites.

Mots clés: Confidentialité, Équité, Classement, Calcul multipartite sécurisé, Confidentialité différentielle

Abstract

We present a novel privacy-preserving approach to enhance item fairness in ranking systems. We employ post-processing techniques in a multi-stakeholder recommendation environment in order to balance fairness and privacy protection for both producers and consumers. Our method utilizes secure multi-party computation (MPC) servers and differential privacy (DP) to maintain user privacy while mitigating item unfairness without compromising utility. Users submit their data as secret shares to MPC servers, and all calculations on this data remain encrypted. We evaluate our approach using real-world datasets, such as Amazon Digital Music, Book Crossing, and MovieLens-1M, and analyze the trade-offs between privacy, fairness, and utility. Our work encourages further exploration of the intersection of privacy and fairness in recommender systems, laying the groundwork for integrating other privacy-enhancing techniques to optimize runtime and scalability for real-world applications. We envision our approach as a stepping stone towards end-to-end privacy-preserving and fairness-promoting solutions in multi-stakeholder recommendation environments.

Keywords: Privacy, Fairness, Ranking, Secure Multi-party Computation, Differential Privacy

Contents

List of Tables

List of Figures

List of acronyms and abbreviations

SVD	Singular Value Decomposition
EOAA	Equity of Amortized Attention
PET	Privacy-Enhancing Technologies
DP	Differential Privacy
MPC	Secure Multi-Party Computation
PRP	Probability Ranking Principle
MSE	Mean Squared Error
ILP	Integer Linear Programming
NDCG	Normalized Discounted Cumulative Gain
LTR	Learning to Rank

Acknowledgements

First and foremost, I am truly grateful to my advisors, Esma Aïmeur and Golnoosh Farnadi, for welcoming me into their group, and providing me with invaluable guidance and support that have greatly contributed to my academic and personal growth.

I am deeply appreciative of my collaborators: Sikha Pentyala, for the close mentorship and assistance, and Martine De Cock, for the insightful feedback and commitment to ensuring the quality of my work.

I would like to express my gratitude to my colleagues in Golnoosh's lab for their camaraderie and intellectual exchange. Our weekly group meetings were immensely educational, and I cherished our social outings. A special thanks to Rebecca Salganik for her commitment to helping me navigate life decisions and providing emotional support.

I would like to extend my heartfelt thanks my friends Bonnie Li, David Qian, Arjun Gupta, William Gao, Shahrukh Rahman, and Raag Kashyap for their steadfast presence during both challenging and joyous moments, offering honest and constructive advice, and consistently bringing laughter and joy into my life.

Last but not least, I express my deepest gratitude to my parents for their unconditional love, endless encouragement, and unwavering belief in my abilities.

Chapter 1

Introduction

The proliferation of digital platforms and recommender systems has led to an increased demand for personalized recommendations that cater to individual user preferences, driving consumer engagement and enhancing user satisfaction. However, as the influence of recommender systems grows, so does the need to address the challenges of fairness and privacy in these environments.

Recommender systems often generate ranked lists to optimize utility for each user based on their relevance scores or ratings. However, this can result in position bias [**13, 14**], a phenomenon where users tend to focus their attention on the highest-ranked items in the list, inadvertently disadvantaging lower-ranked items by allocating them disproportionately less attention relative to their relevance. Over time, this position bias can accumulate, causing a systemic inequality of exposure for items and leading to negative economic and social impacts on the item providers (a.k.a. producers), resulting in economic disparity, bias toward underrepresented producers, and unhealthy markets [**15, 19**].

While several methods have been proposed to mitigate unfairness in rankings [**34, 20, 28, 35, 21**], these approaches often require centralization of user data, which can lead to privacy concerns [**5**]. In response to these challenges and in compliance with privacy regulations like the General Data Protection Regulation (GDPR)[1], privacy-enhancing technologies (PETs) have gained prominence in recommender systems [**12, 33, 2, 7, 32**].

Our work is fundamentally focused on the potential for data leakage, a concern we tackle by prioritizing both input and output privacy. Input privacy requires that users mask their identity and obscure their data prior to sharing it with other individuals or a centralized authority. This requirement aids in safeguarding a user's data integrity. Output privacy, in contrast, is designed to ensure that any received data cannot be traced back to its origin, thus preventing the identification of the source or precise location from which the data was extracted. Relying solely on one PET may not be adequate for assuring both types of privacy.

[1] European General Data Protection Regulation `https://gdpr-info.eu/`

As such, the nature of the task may necessitate the intricate integration of several PETs. Consider secure multi-party computation (MPC), for instance. This would be the natural choice for upholding input privacy given its technique of splitting user data before sharing with others or MPC servers. The benefit here is that it prevents any single entity from having full access to comprehensive information about another user, while simultaneously enabling collaborative endeavors. However, once the partitioned data is reassembled, without proper protective measures, the delivery to the end user might expose sensitive information about other participants. On the other hand, differential privacy (DP) can be effective in preserving output privacy, as it has been the gold standard in industry for preventing user identification when dealing with the dissemination of aggregate data [**1**]. By introducing strategically computed noise into the collective data, DP can disrupt the information flow before it reaches the recipient. In essence, MPC and DP manage different facets of privacy within our research, and utilizing just one of these tools is insufficient to guarantee comprehensive protection for both input and output privacy. Consequently, it is necessary to devise strategies that effectively integrate both these technologies together so that we can preserve the sanctity of both input and output privacy. It is only by harnessing their combined strengths that we can offer users a more robust and comprehensive guarantee of privacy.

To the best of our knowledge, there have not been any studies that simultaneously tackle the challenge of promoting fairness for producers (a.k.a. items) while safeguarding the privacy of consumers (a.k.a. users) within the context of ranking systems. This thesis focuses on developing a ranking system that addresses this gap by both preserving the privacy of consumers and enhancing the fairness for producers in multi-stakeholder recommendation settings. Specifically, the goal is to modify each user's relevance-based ranking during a post-processing phase to optimize equity of amortized attention [**4**], without disclosing the ranking to a centralized entity. We propose a novel approach that combines MPC to protect input privacy and DP to ensure output privacy. Our method allows users to perform local computations using perturbed intermediate results from MPC servers, thus ensuring both fairness and privacy. We evaluate our approach on three real-world datasets, demonstrating its effectiveness in reducing position bias, improving fairness for items without compromising user privacy or ranking quality.

1.1. Thesis Layout

Chapter 2 presents the published article on which this thesis is based, and outlines each author's contributions.

Chapter 3 provides an overview of the background and relevant concepts in ranking systems, focusing on the topics of singular value decomposition (SVD), fairness in rankings with an emphasis on equity of amortized attention (EOAA), and privacy-enhancing technologies

(PETs), including differential privacy (DP), the Laplace mechanism, and secure multi-party computation (MPC), with a particular focus on secret sharing. This foundation sets the stage for a more in-depth exploration of the intersection of fairness and privacy in ranking systems.

Chapter 4 provides an extensive review of relevant literature, encompassing fairness in rankings, privacy-preserving ranking systems, and fair and privacy-preserving ranking systems.

Chapter 5 introduces the core of the thesis–the proposed method for a fair and privacy-preserving ranking system. This method employs integer linear programming, secure multi-party computation, and differential privacy to achieve the desired objectives. We evaluate the performance of our method through experiments conducted on real-world datasets, such as Amazon Digital Music, Book Crossing, and MovieLens-1M. In doing so, we analyze the trade-offs between privacy, fairness, and utility, offering valuable insights into the effectiveness of our approach.

Chapter 6 concludes the thesis by highlighting the potential for further research in this field and the prospects for integrating other privacy-enhancing techniques to improve runtime and scalability in real-world applications. By showcasing the potential for integrating additional privacy-enhancing techniques to optimize runtime and scalability, we hope to inspire the development of end-to-end privacy-preserving and fairness-promoting solutions for both producers and consumers in multi-stakeholder recommendation environments.

Chapter 2

Prologue to Article

2.1. Article Details

Privacy-Preserving Fair Item Ranking. By: Jia Ao Sun, Sikha Pentyala, Martine De Cock, and Golnoosh Farnadi. This article was accepted to the 45th European Conference on Information Retrieval (ECIR 2023) [**29**].

2.2. Author Contributions

The main constributions of Jia Ao Sun for this article are presented.

- Jointly formulated the research problem with Golnoosh Farnadi.
- Implemented all aspects, including the singular value decomposition ranking system for generating user relevance-rankings, the equity of amortized attention integer linear program, and various differential privacy methods such as randomized response, Laplace mechanism, and exponential mechanism.
- Conducted all experiments, testing various combinations of MPC and DP methods within the EOAA framework.
- Performed comprehensive analyses on the interplay between privacy and fairness, as well as privacy and ranking utility.
- Prepared the manuscript drafts.
- The code and data for replicating our experiments is available on GitHub: `https://github.com/sunosaj/privacy-fair-ranking`

Sikha Pentyala proposed the privacy-preserving approach that integrates MPC and DP within the EOAA mechanism, implemented of secure multi-party computation techniques, and revised and edited the manuscript.

Martine De Cock provided feedback, and revised and edited the manuscript.

Golnoosh Farnadi jointly formulated the research problem with Jia Ao Sun, provided feedback, and revised and edited the manuscript.

Chapter 3

Background

3.1. Ranking Systems

Ranking systems are essential tools in categorizing and arranging items in a collection based on their significance or pertinence to a user or specific inquiry. Their widespread application spans search engines, recommender systems, social media platforms, and online marketplaces. In order to optimize these systems, adherence to the Probability Ranking Principle (PRP) [**24**] is critical. The PRP states that items should be arranged in descending order of their likelihood of relevance, which maximizes the overall utility for users.

The foundation of PRP lies in calculating a relevance score for each item, taking into account factors such as content relevance, item popularity, and similarity to other items in the collection. A common approach to compute these scores is to use the singular value decomposition (SVD) algorithm, which is a matrix factorization technique that can capture latent factors to generate relevance scores for items. In this thesis, SVD was chosen as the ranking model due to its high accuracy and scalability [**18**], as well as its simplicity, though our work can be applied to any ranking model.

In the following section, we will delve deeper into the technical details of the SVD algorithm and its application in ranking systems.

3.1.1. Singular Value Decomposition (SVD)

Singular value decomposition (SVD) is a widely-used matrix factorization technique for discovering latent factors and generating personalized recommendations in ranking systems. It decomposes an $m \times n$ matrix $\mathbf{M}$ into

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}, \tag{3.1.1}$$

where $\mathbf{U}$ is an $m \times r$ matrix, $\mathbf{\Sigma}$ is an $r \times r$ diagonal matrix, and $\mathbf{V}^{\mathbf{T}}$ is an $r \times n$ matrix, with r being the chosen number of latent factors. Latent factors are also known as hidden variables, which are not explicitly detailed by the dataset but rather inferred by the model through the use of matrix factorization.

In recommender systems, $\mathbf{M}$ represents user-item ratings, with m users and n items. The matrices $\mathbf{U}$ and $\mathbf{V^T}$ correspond to users' and items' latent factors, respectively. By selecting a specific number of latent factors and applying SVD to $\mathbf{M}$, the model estimates missing user-item ratings through the dot product of the corresponding rows and columns in $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V^T}$. To refine the model, the error between predicted and actual ratings is minimized using a cost function such as mean squared error (MSE) in a process called matrix factorization, which utilizes gradient descent techniques to iteratively update the values of $\mathbf{U}$ and $\mathbf{V^T}$.

Once the SVD model is trained and evaluated, it can predict ratings for unrated items. Items are then ranked by their predicted ratings for each user, generating a personalized list sorted by relevance. This ranking process is commonly used in experiments that employ SVD as the ranking model.

3.1.2. Equity of Amortized Attention (EOAA)

Ranking algorithms are designed to optimize utility by generating lists of items sorted according to their predicted relevance to a specific query or user. Items of higher relevance are placed at the top of the list, thereby receiving the most attention due to position bias. Ensuring fairness in rankings involves distributing exposure equitably among the listed items to mitigate the impact of this inherent position bias. In this thesis, we examine the notion of fairness centered on the attention received by items, which is determined by their exposure in the list. This concept, known as the equity of amortized attention (EOAA), was introduced by Biega et al. [**4**]. The EOAA model takes into account the decay of user attention across the ranking positions and aims to fairly distribute this attention among the items, considering the items' relevance.

Formally, Biega et al. introduced the concept of EOAA to attain "amortized individual fairness" for a collection of items $\mathcal{D} = d_1, d_2, \ldots, d_i, \ldots, d_n$ that appear in a series of relevance-based rankings ρ_1,ρ_2, $\ldots$,ρ_l, $\ldots$,ρ_L corresponding to users $u_1, u_2, \ldots, u_l, \ldots, u_L$ [**4**]. The attention an item d_i garners depends on its position within ranking ρ_l. Equity of amortized attention is achieved when the accumulated attention A_i received by each item d_i is proportional to its total relevance R_i over the series of rankings. The authors quantify this fairness notion by calculating the sum of the absolute differences between A_i and R_i for $i = 1 \ldots n$, as presented in Eq. 3.1.2:

$$unfairness(\rho_1, ..., \rho_L) = \sum_{i=1}^{n} |A_i - R_i| = \sum_{i=1}^{n} \left| \sum_{l=1}^{L} a_i^l - \sum_{l=1}^{L} r_i^l \right| . \tag{3.1.2}$$

For each item d_i, r_i^l represents its relevance score concerning user u_l, and a_i^l denotes the attention it obtains within ranking ρ_l.

The unfairness of a series of relevance-based rankings $\rho_1,\rho_2,\ldots,\rho_l,\ldots,\rho_L$ is reduced by sequentially reranking to generate $\rho_1^*,\rho_2^*,\ldots,\rho_l^*,\ldots,\rho_L^*$. The reranking of ρ_l, considering the previously calculated rerankings $\rho_1^*,\rho_2^*,\ldots,\rho_{l-1}^*$, is obtained by solving the following integer linear programming (ILP) problem:

$$\text{Minimize} \sum_{i=1}^{n}\sum_{j=1}^{n}\left|A_i^{l-1} + \hat{w}_j - (R_i^{l-1} + \hat{r}_i^l)\right| \cdot X_{i,j} \tag{3.1.3}$$

$$\text{Subject to} \sum_{j=1}^{k}\sum_{i=1}^{n}\frac{2^{\hat{r}_i^l}-1}{\log_2(j+1)}X_{i,j} \geq \theta \cdot DCG@k(\rho_l) \tag{3.1.4}$$

$$X_{i,j} \in \{0,1\}, \forall_{i,j} \text{ and } \sum_i X_{i,j} = 1, \forall_j \text{ and } \sum_j X_{i,j} = 1, \forall_i. \tag{3.1.5}$$

In this ILP, we have n^2 decision variables, $X_{i,j}$, which determine the optimal reranking ρ_l^* of items in ranking ρ_l. The accumulated attention and relevance of item d_i over rerankings $\rho_1^*,\rho_2^*,\ldots,\rho_{l-1}^*$ are represented by A_i^{l-1} and R_i^{l-1}, respectively. Normalized attention weight, $\hat{w}_j$, is calculated as

$$\hat{w}_j = \frac{0.5^j}{1-0.5^n}. \tag{3.1.6}$$

The normalized relevance score, $\hat{r}_i^l$, is computed using the maximum and minimum relevance scores a user can have for an item, $r_{\max}$ and $r_{\min}$, respectively:

$$\hat{r}_i^l = \frac{r_i^l - r_{\min}}{\sum_{t=1}^{n}\left(r_i^l - r_{\min}\right)}. \tag{3.1.7}$$

The unfairness measure for placing any item d_i at position j in the current l^{th} reranking is given by the the expression $\left|A_i^{l-1} + w_j - (R_i^{l-1} + r_i^l)\right|$. The constraint in Eq. 3.1.4 ensures that the quality of the top-k items in the reranking is no lower than $0 \leq \theta \leq 1$ times the discounted cumulative gain (DCG),

$$DCG@k(r) = \sum_{i=1}^{k}\frac{2^{r_i}-1}{\log_2(i+1)}, \tag{3.1.8}$$

of the top-k items in the original relevance-based ranking. The constraints in Eq. 3.1.5 also specify that the decision variables $X_{i,j}$ are binary and there is a single 1 per j for all i's, and a single 1 per i for all j's.

The quality of reranking ρ_l^* is evaluated by its divergence from the original relevance-based ranking ρ_l, as measured by the normalized discounted cumulative gain (NDCG) metric:

$$NDCG(\rho_l,\rho_l^*) = \frac{DCG(\rho_l^*)}{DCG(\rho_l)}. \tag{3.1.9}$$

The maximum NDCG value is 1, which occurs when either $\rho_l = \rho_l^*$ or when items with equal relevance scores are rearranged.

3.2. Differential Privacy (DP)

Differential privacy (DP) [**10**] is a mathematical framework for measuring the privacy guarantees of an algorithm. It aims to provide strong privacy guarantees such that the presence or absence of a single individual in a data set does not significantly affect the outcome of the algorithm or reveal sensitive information about that individual. This framework addresses the paradox of learning useful information about a population while learning nothing about an individual. This means output privacy is guaranteed, so users do not need to worry about being identified after their information has been processed by our method. DP is an important part of our work, but it is not sufficient on its own as we need to consider input privacy as well.

Formally, a randomized algorithm $\mathcal{M}$ is ε-differentially private if, for all pairs of neighboring datasets D and D' that differ in a single element, and for all subsets $\mathcal{S}$ of $\mathcal{M}$'s range:

$$|\ln(\mathrm{P}[\mathcal{M}(D) \in \mathcal{S}]) - \ln(\mathrm{P}[\mathcal{M}(D') \in \mathcal{S}])| \leq \varepsilon, \tag{3.2.1}$$

where ε is the privacy budget or privacy loss. The level of privacy protection offered by a differentially private algorithm is determined by ε, which specifies the maximum amount of noise that can be added to the data to achieve privacy. The value of ε determines the trade-off between privacy and accuracy: smaller values of ε provide stronger privacy guarantees but result in noisier output, while larger values of ε result in less noise but weaker privacy guarantees.

Algorithms can use various techniques to achieve differential privacy, including randomization; in our work, we employ the Laplace mechanism, which is a popular and efficient approach for achieving differential privacy. This technique is described in more detail in the following section.

3.2.1. Laplace Mechanism

The Laplace distribution is a probability distribution with a scale parameter b, and is represented by the probability density function $\mathrm{Lap}(b)$ defined by:

$$\mathrm{Lap}(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{3.2.2}$$

To achieve differential privacy using the Laplace mechanism, we let $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ be a deterministic function that maps databases to k real numbers. The ℓ_1-sensitivity of a function

$f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is defined as:

$$\Delta f = \max_{\substack{x,y \in \mathbb{N}^{|\mathcal{X}|} \\ \|x-y\|_1 = 1}} \|f(x) - f(y)\|_1. \tag{3.2.3}$$

The Laplace mechanism computes f, and perturbs each coordinate with noise drawn from the Laplace distribution [**10**]. The scale of the noise is calibrated to the sensitivity of f, divided by ε. For any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is thus defined as:

$$\mathcal{M}_L(x,f(\cdot),\varepsilon) = f(x) + (Y_1, \ldots, Y_k), \tag{3.2.4}$$

where Y_i are independent and identically distributed (i.i.d.) random variables drawn from $\text{Lap}(\Delta f/\varepsilon)$. This means that the Laplace mechanism adds independent noise values to each component of $f(x)$, where the amount of noise added is proportional to the sensitivity of the function and inversely proportional to the privacy budget ε. We show in Chapter 5 that our method employs the Laplace mechanism to achieve differential privacy while preserving the utility of the data.

3.3. Secure Multi-Party Computation (MPC)

Secure multi-party computation (MPC) is a cryptographic technique that facilitates joint computation of a function on private data by multiple parties, while only revealing the desired output. It allows parties to carry out distributed computing tasks where a number of distinct, yet connected, computing devices or parties jointly compute a function in a secure manner. Specifically, MPC enables n parties to jointly compute a function $(y_1, ..., y_n) \leftarrow f(x_1, \ldots, x_n)$, where each party P_i holds an input x_i, obtains an output y_i, and learns nothing except for (x_i, y_i, f), where function f is typically modeled as a Boolean or arithmetic circuit. MPC is designed to handle possible malicious behavior by adversarial entities, including protocol attacks by external entities or even a subset of participating parties. This attack may aim to learn private information or cause incorrect computation results.

In the MPC setting, some parties may have good intentions, while others may intend to compromise the privacy of the other parties' inputs. We consider a setting where an adversarial entity controls some subset of parties and wishes to attack the protocol execution. The parties under the adversary's control are corrupted and follow the adversary's instructions. In contrast, non-corrupt parties are called honest parties. Secure protocols should withstand any adversarial attack. Two types of adversaries exist: semi-honest adversaries and malicious adversaries. Semi-honest adversaries correctly follow the protocol specification but obtain the internal state of all corrupted parties to learn private information. Malicious adversaries can arbitrarily deviate from the protocol specification.

The literature on MPC distinguishes two settings based on the maximum number t of corrupted parties: dishonest majority ($n/2 \leq t < n$, particularly we often adopt $t = n - 1$) and honest majority ($t < n/2$).

MPC has shown itself to be an appealing technique for ensuring input privacy, so users can feel confident in providing their private information for our method. MPC, in conjunction with DP, allows for the preservation of both input and output privacy, providing end-to-end privacy guarantees to the user that would not be possible through the sole use of either MPC or DP.

3.3.1. Secret Sharing

Over the past three decades, many different techniques have been developed for constructing MPC protocols with different properties, and for different settings. Secret sharing is a fundamental MPC concept that allows a group of parties to jointly compute a function on their private data while ensuring that no individual party can access the private data of the others. Secret sharing involves splitting a secret into multiple shares, with each share distributed to a different party. The secret can only be reconstructed by combining the shares held by a specified number of parties called the threshold, which is often less than the total number of parties. This thesis focuses on secret-sharing-based MPC protocols, which are concretely efficient and have been proposed for various settings.

MPC computations are typically performed on integers modulo q; specifically, in the ring $\mathbb{Z}_q$. In a dishonest majority 2-party (2PC) computation setting with passive adversaries, a data holder converts its input data into $x = x_0 + x_1 \mod q$ and sends x_0 to MPC server S_0, and x_1 to S_1. Throughout this thesis, the shorthand for a secret sharing of x is denoted as $[\![x]\!] = (x_0,x_1)$. If servers S_0 and S_1 receive secret shares y_0 and y_1, respectively, of a value y from another data holder, the servers can compute a secret sharing of $x + y$ as $[\![x + y]\!] = (x_0 + y_0,x_1 + y_1)$ without learning the values of x, y, or their sum. In addition, we use an MPC protocol π_{LAP}, which allows the MPC servers to sample secret shares of noise drawn from a Laplace distribution. The protocol π_{LAP} requires communication between the MPC servers, in addition to operations on their own local shares.

Chapter 4

Related Work

4.1. Fairness in Rankings

Fairness in rankings has been a widely studied topic, particularly in the context of fairly distributing exposure to the elements of the ranked list. These elements, referred to as items, could include people, products, or places. Existing work in this area has largely focused on group fairness, where exposure is distributed equitably among different groups with protected attributes such as gender or race. For instance, Yang and Stoyanovich [**34**] proposed extending the concept of statistical parity to ranking systems in order to eliminate the influence of protected group membership on a person's ranking position. Zehlike and Castillo [**35**] proposed a fair top-k ranking algorithm, called FA*IR, based on affirmative action, guaranteeing a minimum number of protected group members in every top-k ranking. Other works, such as that of Celis et al. [**6**], addressed the constrained ranking maximization problem by ensuring that no group dominates and that there is a limit to the number of sensitive items per protected group in the ranking.

While several studies have investigated exposure or group fairness, limited research has been conducted on individual item fairness in rankings. One notable work in this area is Biega et al.'s introduction of equity of attention [**4**], which aims to achieve fairness over a sequence of rankings by distributing attention proportional to item relevance. Recently, our work has extended Biega et al.'s approach by implementing privacy-preserving measures at various stages of the ranking mechanism to ensure fairness without compromising user privacy.

Singh and Joachims [**27**] also introduced fairness of exposure in rankings by proposing to optimize for the maximum utility in a ranking, subject to group fairness constraints using linear programming. On the other hand, Sapiezynski et al. [**25**] used a geometric distribution to model user attention as an analogue to exposure and applied statistical parity in their study [15].

Overall, the majority of existing research on fairness in rankings has focused on group fairness, while limited attention has been given to individual item fairness. Our work aims to address this gap by extending an existing approach and implementing privacy-preserving measures to ensure fairness in rankings.

4.2. Privacy-Preserving Ranking Systems

Previous works on privacy-preserving learning-to-rank (LTR) systems have employed various privacy-enhancing technologies (PETs). For instance, Dehghani et al. [**8**] utilized the mimic learning approach to share only the trained model instead of the sensitive data itself. They introduced a framework that allowed a central server to train a model by mimicking the behavior of distributed client models, and they used Laplace noise during aggregation as part of the differential privacy (DP) guarantee.

Yang et al. [**33**] employed the information-theoretic privacy approach in their privacy-preserving LTR system, where they used DP to obfuscate data and satisfy a data distortion budget. This technique ensures that user privacy is preserved by adding noise to the data before sharing it with the central server. In contrast, Kharitonov [**17**] utilized a federated learning setup with evolutionary strategies optimization in his privacy-preserving LTR system. He incorporated local DP and proposed a method that enabled a central server to evolve the ranking model by aggregating the models of distributed clients. This approach maintains the privacy of the user data while allowing the central server to learn from it.

Wang et al. proposed two different privacy-preserving LTR systems with different optimization approaches. In their first work, Wang et al. [**31**] extended Kharitonov's federated learning setup with local DP to larger datasets. However, they found a substantial loss in utility compared to other non-private online LTR systems. In their second work, Wang et al. [**30**] used a federated learning setup and local DP, similar to Kharitonov, but incorporated a pairwise differentiable gradient descent (PDGD) optimization approach instead. By adding noise to the gradient updates before sharing them with the central server, they achieved privacy preservation while maintaining high ranking quality.

Ge et al. [8] incorporated the Paillier homomorphic encryption algorithm in their Priv-Item2Vec model for top-N recommendation. Their privacy-preserving LTR system is based on MPC, allowing multiple parties to jointly train a recommendation model while keeping their private data encrypted. They proposed a method that allows parties to share their private data with each other without revealing it to others and used a privacy-preserving algorithm to train the recommendation model.

However, among these previous works in privacy-preserving ranking systems, none have explored the use of MPC together with DP. This presents an opportunity for further research to explore the potential of combining MPC and DP for privacy-preserving LTR systems. Our research explores this combination by adding noise to the data during computation, such that multiple parties can compute a function on their private data without revealing it to others while providing a DP guarantee.

4.3. Fair and Privacy-Preserving Ranking Systems

The intersection of fairness and privacy in ranking systems has received limited attention in the literature, despite both topics being well-studied in isolation. Resheff et al. [**23**] proposed a privacy-adversarial training approach for improving group fairness while preserving user privacy in their recommender system. They used adversarial training to obtain user representations that obfuscate sensitive attributes, preventing implicit private information attacks.

In contrast, Sato [**26**] presented a local ranking system framework that allows users to post-process the rankings they receive by setting their own fairness constraints based on their preferences. This approach offers privacy preservation by allowing each user to develop their own recommender system, independent of a centralized recommender system. However, the computational cost of maintaining these individual systems may be prohibitive for some users.

While both approaches address privacy and fairness concerns related to users' protected attributes, there has been no work to date that addresses privacy and fairness with respect to the exposure items receive relative to their relevance. This thesis delves into this unexplored research domain.

Chapter 5

Privacy-Preserving Fair Item Ranking

5.1. Problem Description

This study focuses on a regression-style recommendation model, denoted as $\mathcal{M}$, which is trained in a privacy-preserving manner using techniques such as [**32, 17**] to ensure that no information about the training data is leaked. The model $\mathcal{M}$ is deployed at each user u_l to predict their relevance scores $\mathbf{r}^l = (r_1^l, r_2^l, \ldots, r_i^l, \ldots, r_n^l)$ on a global set of items $\mathcal{D} = (d_1, d_2, \ldots, d_i, \ldots, d_n)$, where $r_i^l = \mathcal{M}(u_l, d_i)$. Notably, the users' raw data, including their preferences, demographics, and embeddings, are not disclosed to any third party. Each user u_l's relevance-based ranking ρ_l is a list of items sorted in decreasing order of their relevance scores.

The proposed post-processing technique, detailed in 3.1.2, assumes that a central server S has access to the relevance-based rankings $\rho_1, \rho_2, \ldots, \rho_l, \ldots, \rho_L$ of all users and can rerank them into ρ_1^*,$\rho_2^*, \ldots, \rho_l^*, \ldots, \rho_L^*$ to achieve equity of amortized attention without compromising ranking utility beyond a set threshold. However, this centralized setup can potentially leak sensitive user information to the central server S, including:

(1) the preference of the user for all items in the form of relevance scores,

(2) the top-k items that the user is most likely to be interested in, and

(3) the order of the top-k items that the user is most likely to be interested in.

To address privacy concerns (P_1)–(P_3) and achieve individual fairness for the items while preserving user privacy, we have adapted our post-processing method.

5.2. Proposed Method

Our approach for computing an aggregated unfairness measure involves the use of MPC protocols to ensure input privacy and correctness. In our MPC protocol design, each data holder encrypts their input data value x by converting it into secret shares, which are then

[1] Figure adapted from `https://sepior.com/mpc-blog`

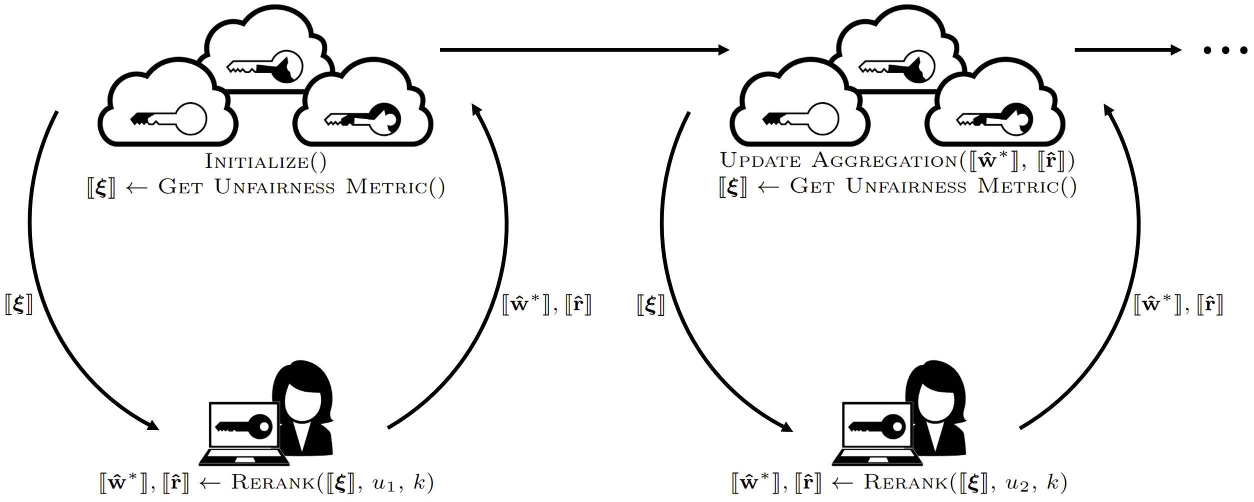

INITIALIZE()
$[\![\boldsymbol{\xi}]\!] \leftarrow$ GET UNFAIRNESS METRIC()
UPDATE AGGREGATION($[\![\hat{\mathbf{w}}^*]\!]$, $[\![\hat{\mathbf{r}}]\!]$)
$[\![\boldsymbol{\xi}]\!] \leftarrow$ GET UNFAIRNESS METRIC()
$[\![\boldsymbol{\xi}]\!]$
$[\![\hat{\mathbf{w}}^*]\!]$, $[\![\hat{\mathbf{r}}]\!]$
$[\![\boldsymbol{\xi}]\!]$
$[\![\hat{\mathbf{w}}^*]\!]$, $[\![\hat{\mathbf{r}}]\!]$
$[\![\hat{\mathbf{w}}^*]\!]$, $[\![\hat{\mathbf{r}}]\!] \leftarrow$ RERANK($[\![\boldsymbol{\xi}]\!]$, u_1, k)
$[\![\hat{\mathbf{w}}^*]\!]$, $[\![\hat{\mathbf{r}}]\!] \leftarrow$ RERANK($[\![\boldsymbol{\xi}]\!]$, u_2, k)
...

Figure 5.1. Flow diagram of privacy-preserving fair item ranking algorithm[1]

distributed among a set of MPC servers. By combining all the shares, the original value of x can be reconstructed, but none of the MPC servers learn anything about the value of x individually.

The MPC servers then execute protocols to perform computations over the secret shared values, resulting in a secret sharing of the desired output value. In our case, an aggregated unfairness measure perturbed with noise to provide DP. The MPC protocols we use are mathematically proven to guarantee input privacy and correctness, while preventing attacks by adversaries who may corrupt one or more of the parties to learn private information or alter the computation result.

Our protocols are designed to be generic enough to be used in both dishonest-majority and honest-majority settings, and to defend against both passive and active adversaries. We can achieve this by adapting the underlying MPC scheme to align with the desired security setting. The use of MPC protocols ensures that each data holder's input data remains private, while still allowing for computation of an aggregated unfairness measure that is necessary to assess the fairness of ranking systems.

Our method for privacy-preserving reranking is based on the observation that each user u_l has all the information needed to solve the ILP (Eqs. 3.1.3, 3.1.4, 3.1.5) to rerank their original ranking ρ_l into ρ_l^*, except for the values of A_i^{l-1} and R_i^{l-1} in Eq. 3.1.3. These values depend on sensitive information from users $u_1, \ldots, u_{l-1}$, which we do not want to disclose to user u_l or a central server. Therefore, we maintain encrypted versions of A_i^{l-1} and R_i^{l-1} which initially are 0 (Procedure INITIALIZE in Algorithm 1) and updated in a privacy-preserving manner each time a user completes their ILP computation. These values are split into secret shares and distributed over MPC servers who can jointly perform operations to update the shares, without ever learning the true values of the inputs or the results of the computations. We use $\mathbf{A}^{l-1}$ and $\mathbf{R}^{l-1}$ to denote the vectors $[A_1^{l-1}, \ldots, A_n^{l-1}]$ and $[R_1^{l-1}, \ldots, R_n^{l-1}]$, respectively.

This high-level flow of our solution is illustrated in Figure 5.1, and Algorithm 1 provides the pseudo-code.

Our method works by updating user rankings in a sequence from $l = 1 \ldots L$. When user u_l is reached, u_l requests the current $\mathbf{A}^{l-1} - \mathbf{R}^{l-1}$ from the MPC servers. This prompts the servers to compute a secret sharing $[\![\mathbf{A}^{l-1} - \mathbf{R}^{l-1}]\!]$ from their local secret shares of $\mathbf{A}^{l-1}$ and $\mathbf{R}^{l-1}$ through the Get Unfairness Metric procedure.

Ideally, each MPC server could send their secret shares of $\mathbf{A}^{l-1} - \mathbf{R}^{l-1}$ to user u_l, who could combine them to construct the value of $\mathbf{A}^{l-1} - \mathbf{R}^{l-1}$. However, even though $\mathbf{A}^{l-1} - \mathbf{R}^{l-1}$ only consists of aggregated information, it could still reveal information about previous users $u_1, \ldots, u_{l-1}$ to user u_l, especially if u_l is one of the first to rerank.

As part of the Get Unfairness Metric procedure, the MPC servers perturb the value at each index of $[\![\mathbf{A}^{l-1} - \mathbf{R}^{l-1}]\!]$ with Laplace noise to mitigate privacy loss. The perturbed secret sharing is referred to as $[\![\boldsymbol{\xi}]\!]$. The DP guarantee that the MPC servers provide is that the probability of returning any specific value of $\boldsymbol{\xi}$ is similar to the probability of returning that value if the data of a previous user u_i ($i = 1 \ldots l-1$) was not included in the computation of $[\![\mathbf{A}^{l-1} - \mathbf{R}^{l-1}]\!]$, as per Eq. 3.2.1. This guarantee ensures that the value of $\boldsymbol{\xi}$ returned to u_l does not reveal any information about the users who computed their rerankings before u_l.

Using the MPC-protocol π_{LAP}, the MPC servers generate secret shares of Laplace noise to ensure secure sampling from a Laplace distribution. These secret shares are added to $[\![\mathbf{A}^{l-1} - \mathbf{R}^{l-1}]\!]$, emulating the global DP paradigm without the need for a central trusted aggregator.

Upon receiving the secret shares of $\boldsymbol{\xi}$ from all MPC servers, user u_l combines them to obtain $\mathbf{A}^{l-1} - \mathbf{R}^{l-1} + \pi_{\mathsf{LAP}}(b)$, which serves as an approximation for $\mathbf{A}^{l-1} - \mathbf{R}^{l-1}$. Subsequently, user u_l solves the ILP program mentioned in Sec. 3.1.2 (Procedure Rerank) to determine the values of $X_{i,j}$ ($i = 1 \ldots n$ and $j = 1 \ldots n$).

To simplify the computation of the ILP solution, the implementation scales $\mathbf{A}^{l-1} - \mathbf{R}^{l-1} + \pi_{\mathsf{LAP}}(b)$ by a positive factor ϵ/L. Importantly, this scaling doesn't influence the outcome of $X_{i,j}$. Using $X_{i,j}$, a vector $\mathbf{s}$ is generated to reorder the normalized attention weights $\hat{\mathbf{w}}$. The reordered attention weights form a vector called $\hat{\mathbf{w}}^*$, which, along with the original rankings $\hat{\mathbf{r}}$, constitutes the reranking ρ_l^*.

User u_l then encrypts the values in both $\hat{\mathbf{w}}^*$ and $\hat{\mathbf{r}}$ by converting them into secret shares $[\![\hat{\mathbf{w}}^*]\!]$ and $[\![\hat{\mathbf{r}}]\!]$ and distributing these shares among the MPC servers. This step allows the servers to update their secret shares of the aggregated values, $[\![\mathbf{A}^l]\!]$ and $[\![\mathbf{R}^l]\!]$, as part of the Update Aggregations procedure. The servers require these updated values when the subsequent user, u_{l+1}, sends a request. The process is iteratively executed until all users have completed their reranking tasks.

To achieve ϵ-DP in our algorithm, noise is added to the true aggregate by the MPC servers while responding to each query. They do this by drawing noise from a Laplace distribution

Algorithm 1 Privacy-Preserving Fair Item Ranking

```
Achieving EOAA privately over $L$ users
$n \leftarrow$ number of items per ranking
$L \leftarrow$ number of rankings to rerank
$k \leftarrow$ number of items in quality constraint Eq. 3.1.4
$\mathbf{w} \leftarrow$ attention weights
$\hat{\mathbf{w}} \leftarrow$ NORMALIZE($\mathbf{w}$)
INITIALIZE()
for each $u_l$, where $l = 1 \ldots L$ do
    $\llbracket \boldsymbol{\xi} \rrbracket \leftarrow$ GET UNFAIRNESS METRIC()
    $\llbracket \hat{\mathbf{w}}^* \rrbracket, \llbracket \hat{\mathbf{r}} \rrbracket \leftarrow$ RERANK($\llbracket \boldsymbol{\xi} \rrbracket$, $u_l$, $k$)
    UPDATE AGGREGATION($\llbracket \hat{\mathbf{w}}^* \rrbracket$, $\llbracket \hat{\mathbf{r}} \rrbracket$)
end for

User $u_l$ subroutine
procedure RERANK($\llbracket \boldsymbol{\xi} \rrbracket, u_l, k$)
    $\boldsymbol{\xi} \leftarrow (\epsilon/L) \sum \llbracket \boldsymbol{\xi} \rrbracket$
    $\mathbf{r} \leftarrow \mathcal{M}(u_l, \mathcal{D})$
    $\hat{\mathbf{r}} \leftarrow$ NORMALIZE($\mathbf{r}$)
    $\mathbf{s} \leftarrow$ ILP($\boldsymbol{\xi}, \hat{\mathbf{r}}, k$)
    $\hat{\mathbf{w}}^* \leftarrow \hat{\mathbf{w}}[\mathbf{s}]$
    Return $\llbracket \hat{\mathbf{w}}^* \rrbracket$, $\llbracket \hat{\mathbf{r}} \rrbracket$
end procedure

MPC servers' subroutines
procedure INITIALIZE()
    $\Delta f \leftarrow$ sensitivity calculated from Eq. 5.2.5
    $\epsilon \leftarrow$ privacy budget
    $\llbracket \mathbf{A} \rrbracket \leftarrow \llbracket 0 \rrbracket$
    $\llbracket \mathbf{R} \rrbracket \leftarrow \llbracket 0 \rrbracket$
end procedure

procedure GET UNFAIRNESS METRIC()
    //MPC protocol for global DP
    $b \leftarrow \Delta f / (\epsilon / (n \cdot L))$
    $\llbracket \boldsymbol{\xi} \rrbracket \leftarrow \llbracket \mathbf{A} - \mathbf{R} \rrbracket + \pi_{\mathsf{LAP}}(b)$
    Return $\llbracket \boldsymbol{\xi} \rrbracket$
end procedure

procedure UPDATE AGGREGATION($\llbracket \hat{\mathbf{w}}^* \rrbracket, \llbracket \hat{\mathbf{r}} \rrbracket$)
    //MPC protocol to perform aggregation
    $\llbracket \mathbf{A} \rrbracket \leftarrow \llbracket \mathbf{A} \rrbracket + \llbracket \hat{\mathbf{w}}^* \rrbracket$
    $\llbracket \mathbf{R} \rrbracket \leftarrow \llbracket \mathbf{R} \rrbracket + \llbracket \hat{\mathbf{r}} \rrbracket$
end procedure
```

with a mean of 0 and a scale of $b = \Delta f / \epsilon'$, where Δf represents sensitivity and ϵ' is the privacy budget allocated for each query.

In Algorithm 1, each user u_l ($l = 1 \ldots L$) requests aggregated information $A_i^{l-1} - R_i^{l-1}$ about every item d_i ($i = 1 \ldots n$) from the MPC servers via the GET UNFAIRNESS METRIC procedure. This means the MPC servers must answer a total of $n \cdot L$ queries. Since these queries are executed on overlapping datasets (u_l requests aggregated information from users $u_1, \ldots, u_{l-1}$; u_{l+1} requests aggregated information from users $u_1, \ldots, u_l$, etc.), we allocate a privacy budget of $\epsilon' = \epsilon/(n \cdot L)$ for each query.

The sensitivity Δf was previously defined in Eq. 3.2.3; in our context of summation queries [**22, 9**], the sensitivity for the aggregate ${A_i}^{l-1} - {R_i}^{l-1}$ is calculated in Eq. 5.2.5, which determines the maximum value that a single user can contribute to this aggregate.

$$\hat{r}_{\min} = \frac{r_{\min} - r_{\min}}{r_{\max} - r_{\min}} \bigg/ \left(\frac{r_{\min} - r_{\min}}{r_{\max} - r_{\min}} + (n-1) \left(\frac{r_{\max} - r_{\min}}{r_{\max} - r_{\min}} \right) \right) = \frac{0}{0 + (n-1)(1)} = 0 \quad (5.2.1)$$

$$\hat{r}_{\max} = \frac{r_{\max} - r_{\min}}{r_{\max} - r_{\min}} \bigg/ \left(\frac{r_{\max} - r_{\min}}{r_{\max} - r_{\min}} + (n-1) \left(\frac{r_{\min} - r_{\min}}{r_{\max} - r_{\min}} \right) \right) = \frac{1}{1 + (n-1)(0)} = 1 \quad (5.2.2)$$

$$\hat{w}_{\min} = \frac{w_n}{\sum_{j=1}^{n} w_j} \quad (5.2.3)$$

$$\hat{w}_{\max} = \frac{w_1}{\sum_{j=1}^{n} w_j} \quad (5.2.4)$$

$$\Delta f = \max(|\hat{w}_{\max} - \hat{r}_{\min}|, |\hat{w}_{\min} - \hat{r}_{\max}|) \quad (5.2.5)$$

Eqs. 5.2.3, 5.2.4 are based on the normalized geometric attention model and Eqs. 5.2.1, 5.2.2 are based on the range of the normalized relevance scores that the MPC servers may receive. Computation of Δf and b is independent of the users' data and can be precomputed by one of the MPC servers in the clear, *i.e.*, without encryption (see Procedure Initialize).

Eqs. 5.2.3, 5.2.4 are based on the normalized geometric attention model, while Eqs. 5.2.1, 5.2.2 are based on the range of normalized relevance scores that the MPC servers might receive. The process of determining Δf and b does not depend on the data provided by users, and one of the MPC servers can precompute these values without the need for encryption, as demonstrated in the Initialize procedure.

5.3. Experiments

5.3.1. Datasets

In our experiments, we utilize three real-world datasets from various recommender systems: Amazon Digital Music[2], Book Crossing[3], and MovieLens-1M[4]. Each dataset comprises information about individual users and items, as well as the ratings assigned by users to these items. Table 5.1 provides a detailed breakdown of the number of users, items, ratings, and the potential range of ratings for each dataset.

Table 5.1. Statistics of datasets used to train each SVD model

Dataset	# Users	# Items	# Ratings	Rating Levels
Amazon Digital Music	478,235	266,414	836,006	$1, 2, ..., 5$
Book Crossing	77,805	185,973	433,671	$1, 2, ..., 10$
MovieLens 1M	6,040	3,706	1,000,209	$1, 2, ..., 5$

5.3.2. Setup

In our experiments, we employed an SVD model[5] for each of the three datasets—Amazon Digital Music, Book Crossing, and MovieLens-1M—to predict the relevance scores for every user-item pair combination, under the assumption that these models had been trained with privacy protection. For all datasets, we opted to rerank $n = 100$ items and focused on reranking $L = 3000$ users' rankings for the Amazon Digital Music and Book Crossing datasets, while addressing all $L = 6040$ users' rankings for the MovieLens-1M dataset.

[2] `http://jmcauley.ucsd.edu/data/amazon/links.html`

[3] `http://www2.informatik.uni-freiburg.de/~cziegler/BX/`

[4] `https://grouplens.org/datasets/movielens/1m/`

[5] `https://surpriselib.com/`

To solve the ILP equations (Eq. 3.1.3–3.1.5), we utilized the Gurobi[6] optimization software. We set $k = 100$ as the number of items in the top-k list, assigned the quality loss constraint at $\theta = 0.8$, and computed the sensitivity ($\Delta f = 1$) for all datasets in accordance with Eq. 5.2.5. Our empirical analysis explored a range of privacy budgets $\epsilon \in \{0.5, 1, 10, 100, 1000, 10000, 100000\}$ to observe the impact of varying levels of privacy protection.

All MPC processes were conducted using the MP-SPDZ framework [**16**], operating over a ring $\mathbb{Z}_q$ with $q = 2^{64}$. We carried out the experiments within a dishonest majority security setting, involving two computing parties (2PC) and passive adversaries.

Our evaluation of the proposed approach took into account both unfairness (Eq. 3.1.2) and utility (Eq. 3.1.9) metrics, allowing us to study the trade-offs among privacy, fairness, and utility in the context of our privacy-preserving reranking algorithm.

5.4. Discussion

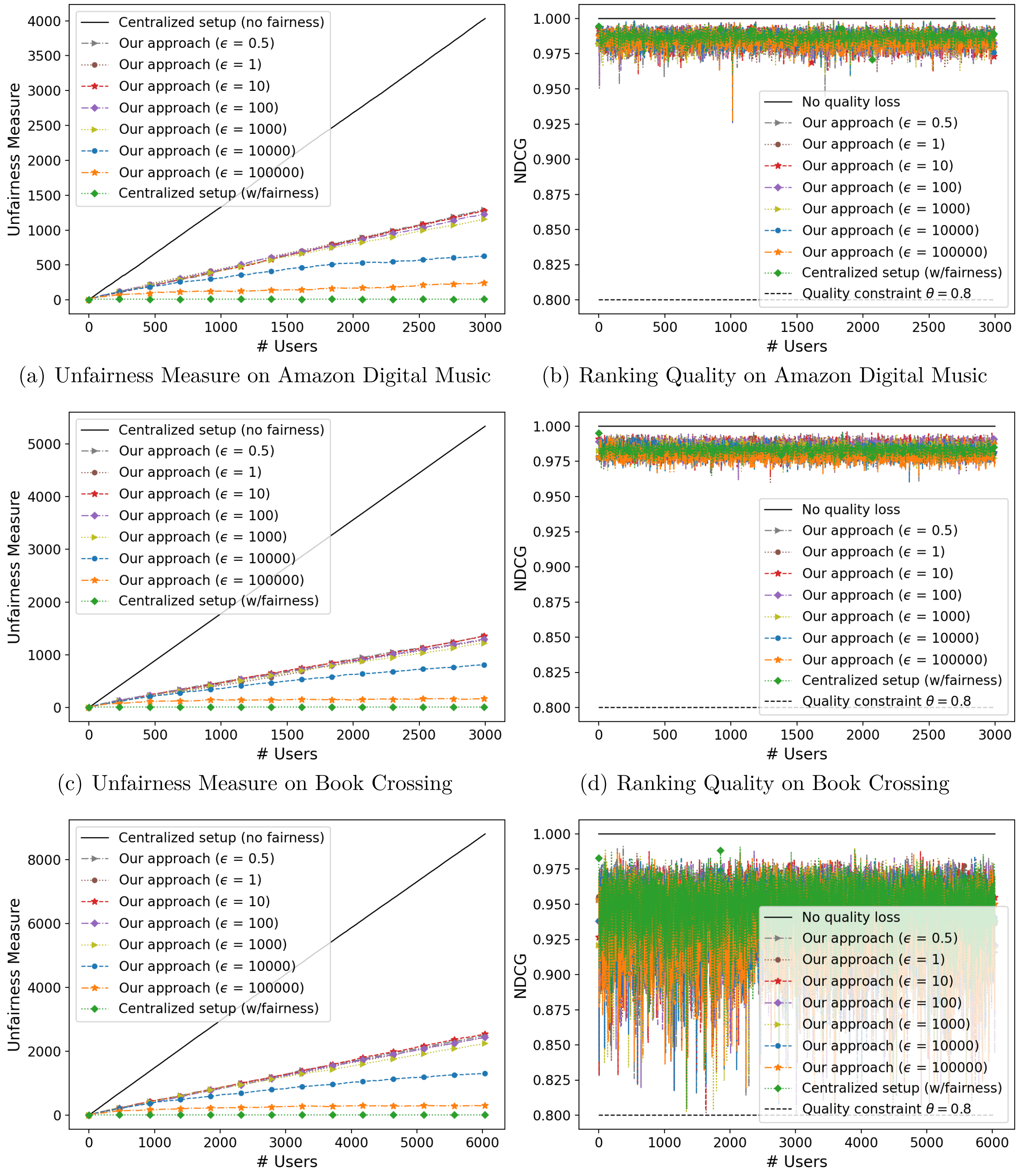
5.4.1. Fairness vs. Privacy Trade-offs

In the left column of Figure 5.2, specifically Figures 2(a), 2(c), and 2(e), we showcase the impact on item fairness when preserving users' privacy. The "Centralized setup (no fairness)" corresponds to the unfairness measurements obtained without any bias mitigation efforts, whereas "Centralized setup (w/fairness)" represents the unfairness measurements after employing the post-processing technique from Sec. 3.1.2 in a non-privacy-preserving centralized setup. Ideally, we aim to develop privacy-preserving approaches that result in unfairness metrics closely resembling those of the "Centralized setup (w/fairness)".

Our findings reveal that our proposed method can still improve fairness even when implementing privacy-preserving techniques. Specifically, our approach ensures user privacy at every stage, both when users send $[\![\hat{\mathbf{w}}^*]\!]$ and $[\![\hat{\mathbf{r}}]\!]$ to the servers, and when the servers transmit $[\![\boldsymbol{\xi}]\!]$ back to the users. We observe a trade-off between privacy and unfairness in our results, wherein an increased level of privacy (i.e., a lower privacy budget) leads to a higher cost to fairness. This observation aligns with the findings reported in existing literature.

The trade-off arises because the addition of noise disrupts the values of the unfairness measure, which in turn affects the effectiveness of the ILP in computing rerankings when compared to a non-differentially private centralized setup. Notably, our solution is capable of preserving input privacy even when faced with higher values of ϵ. As a result, our method offers a promising avenue for achieving item fairness while maintaining user privacy across a range of privacy budgets.

[6] `https://www.gurobi.com/`



(a) Unfairness Measure on Amazon Digital Music

(b) Ranking Quality on Amazon Digital Music

(c) Unfairness Measure on Book Crossing

(d) Ranking Quality on Book Crossing

(e) Unfairness Measure on MovieLens-1M

(f) Ranking Quality on MovieLens-1M

Figure 5.2. Model performance on each dataset

5.4.2. Utility vs. Privacy Trade-offs

Figure 5.2 in the right column (Figures 2(b), 2(d), and 2(f)) demonstrates the implications of incorporating user privacy protection on the quality of rerankings. An NDCG value of 1 represents the highest possible NDCG score, indicating no change in the ordering of item relevance scores when compared to the original ranking. The dashed line at NDCG = 0.8 corresponds to the quality constraint θ set within the ILP, indicating the minimum reranking quality.

Our findings consistently show that reranking quality remains within the defined boundary of $0.8 \leq NDCG \leq 1$, irrespective of the level of noise added for privacy preservation. This implies that our approach allows for user privacy protection while maintaining reranking quality within the acceptable range, as determined by the initial θ threshold set in the ILP. The observed privacy-fairness trade-offs are a consequence of maintaining output privacy, while the utility-fairness trade-offs stem from the application of bias mitigation techniques, both in scenarios with and without privacy protection. This highlights the effectiveness of our method in striking a balance between privacy, fairness, and utility.

5.4.3. Runtime

In our experiments, we utilized a computer with a 2.6 GHz 6-Core Intel Core i7 processor and 16 GB of RAM. Reranking the user's ranking for $n = 100$ items in a non-private, centralized setting takes an average of approximately 0.67 seconds, considering $L =$ 6,040 users. The integration of privacy-preserving mechanisms adds a modest overhead to the runtime. In a 2PC passive security setting with mixed circuits, as discussed in [**11**], the added runtime is less than 5 seconds per client. It is important to note that the actual runtimes can vary based on the specific security settings chosen for the implementation.

When using a 3PC passive security setting, as described in [**3**], the additional runtime for privacy protection can be substantially reduced to less than 1 second. This increase in runtime is a reasonable trade-off when considering the enhanced privacy protection offered. The selection of specific security settings and privacy-preserving techniques may have a direct impact on runtime performance, and these factors should be carefully considered when designing and implementing privacy-preserving algorithms.

Chapter 6

Conclusion & Future Work

In this thesis, we introduced an innovative concept that focuses on enhancing producer (item) fairness while preserving the privacy of consumers (users) in a recommendation environment using post-processing techniques. Our proposed approach involves users collaborating with secure multi-party computation (MPC) servers to reorder items, taking into account both relevance scores, calculated using methods like singular value decomposition (SVD), and attention weights based on a normalized geometric attention model.

Users submit their data as secret shares to the MPC servers, ensuring that all calculations performed on this data remain encrypted. This is achieved by utilizing the MP-SPDZ framework for all MPC computations and working in a dishonest majority security setting with two computing parties (2PC) and passive adversaries. To further protect user privacy, the MPC servers add Laplace noise to any aggregated information they release, providing differential privacy (DP) guarantees based on sensitivity (Δf) and privacy budget (ϵ) parameters, thereby preventing any user's data from being exposed to others.

We showcased the effectiveness of our privacy-focused approach in mitigating unfairness without sacrificing utility by comparing it to a centralized method in which all users reveal their data to a single server. We conducted experiments on real-life datasets, such as Amazon Digital Music, Book Crossing, and MovieLens-1M, and analyzed trade-offs between privacy, fairness, and utility. Our method can be adapted to various other fairness concepts in rankings and applied to other bias reduction techniques.

We hope that our work fosters exploration at the intersection of privacy and fairness in recommender systems and inspires the development of end-to-end privacy-preserving and fairness-promoting solutions for both producers and consumers in multi-stakeholder recommendation environments. Our approach paves the way for integrating other privacy-enhancing techniques to optimize runtime and scalability for real-world applications.

Bibliography

[1] Learning with privacy at scale differential. 2017.

[2] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*, 2019.

[3] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara. High-throughput semi-honest secure three-party computation with an honest majority. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 805–817, 2016.

[4] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 405–414, 2018.

[5] John Canny. Collaborative filtering with privacy. In *IEEE Symposium on Security and Privacy*, pages 45–57, 2002.

[6] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:15, 2018.

[7] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *IEEE Intelligent Systems*, 36(5):11–20, 2020.

[8] Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. Share your model instead of your data: Privacy preserving mimic learning for ranking. *arXiv preprint arXiv:1707.07605*, 2017.

[9] Damien Desfontaines. Differential privacy in practice (easy version). `https://desfontain.es/privacy/differential-privacy-in-practice.html`, 11 2018. Ted is writing things (personal blog).

[10] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[11] Daniel Escudero, Satrajit Ghosh, Marcel Keller, Rahul Rachuri, and Peter Scholl. Improved primitives for MPC over mixed arithmetic-binary circuits. In *Annual International Cryptology Conference*, pages 823–852. Springer, 2020.

[12] Zhengqiang Ge, Xinyu Liu, Qiang Li, Yu Li, and Dong Guo. PrivItem2Vec: a privacy-preserving algorithm for top-N recommendation. *International Journal of Distributed Sensor Networks*, 17(12), 2021.

[13] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2):7–es, 2007.

[14] Thorsten Joachims and Filip Radlinski. Search engines that learn from implicit feedback. *Computer*, 40(8):34–40, 2007.

[15] Matthew Kay, Cynthia Matuszek, and Sean A Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828, 2015.

[16] Marcel Keller. MP-SPDZ: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1575–1590, 2020.

[17] Eugene Kharitonov. Federated online learning to rank with evolution strategies. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pages 249–257, 2019.

[18] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[19] Yudan Liu, Kaikai Ge, Xu Zhang, and Leyu Lin. Real-time attention based look-alike model for recommender system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2765–2773, 2019.

[20] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2243–2251, 2018.

[21] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd international ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 429–438, 2020.

[22] Joseph P. Near and Chiké Abuah. *Programming Differential Privacy*, volume 1. 2021.

[23] Yehezkel S Resheff, Yanai Elazar, Moni Shahar, and Oren Sar Shalom. Privacy and fairness in recommender systems via adversarial training of user representations. *arXiv preprint arXiv:1807.03521*, 2018.

[24] Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 1977.

[25] Piotr Sapiezynski, Wesley Zeng, Ronald E Robertson, Alan Mislove, and Christo Wilson. Quantifying the impact of user attention on fair group representation in ranked lists. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 553–562, 2019.

[26] Ryoma Sato. Private recommender systems: How can users build their own fair recommender systems without log data? In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 549–557. SIAM, 2022.

[27] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2219–2228, 2018.

[28] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. *Advances in Neural Information Processing Systems*, 32, 2019.

[29] Jia Ao Sun, Sikha Pentyala, Martine De Cock, and Golnoosh Farnadi. Privacy-preserving fair item ranking. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II*, pages 188–203. Springer, 2023.

[30] Shuyi Wang, Bing Liu, Shengyao Zhuang, and Guido Zuccon. Effective and privacy-preserving federated online learning to rank. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 3–12, 2021.

[31] Shuyi Wang, Shengyao Zhuang, and Guido Zuccon. Federated online learning to rank with evolution strategies: a reproducibility study. In *European Conference on Information Retrieval*, pages 134–149. Springer, 2021.

[32] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. FedGNN: federated graph neural network for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021.

[33] Dingqi Yang, Bingqing Qu, and Philippe Cudré-Mauroux. Privacy-preserving social media data publishing for personalized ranking-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(3):507–520, 2018.

[34] Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 1–6, 2017.

[35] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578, 2017.