Towards Purpose-aware Privacy-preserving Techniques for Predictive Applications

Towards Purpose-aware Privacy-preserving Techniques for Predictive Applications

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 9 juli 2024 om 15:00 uur

door

Manel Slokom

Master of Data Mining & Knowledge Management,
University of Nantes, France,
geboren te Tunis, Tunesië.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. A. Hanjalic
promotor: Prof. dr. M.A. Larson

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. A. Hanjalic	Technische Universiteit Delft
Prof. dr. M.A. Larson	Radboud Universiteit

Onafhankelijke leden:

Prof. dr. ir. A. Bozzon	Technische Universiteit Delft
Dr. M.S. Pera	Technische Universiteit Delft
Prof. dr. M. Pechenizkiy	Technische Universiteit Eindhoven
Prof. dr. G. Raab	Edinburgh Napier University
Prof. dr. K. Muralidhar	University of Oklahoma
Prof. dr. P. Cesar	Technische Universiteit Delft, reservelid

Keywords: Privacy, recommender systems, machine learning, threat model, privacy-preserving techniques, purpose

Printed by: ProefschriftMaken

Front & Back: Designed by Boyu Xu and Delfina Martinez Pandiani. *The cover art is created using real and fake images. The front cover illustrates the concept that "unity is strength," a quote by Mattie J.T. Stepanek. It symbolizes the multifaceted nature of identity and demonstrates the power of unity. The strength lies in diversity and difference, highlighted through a perturbation, i.e., obfuscation approach.*

ISBN 978-94-6366-883-5

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

CONTENTS

Summary **ix**

1 Introduction **1**
1.1 Privacy risks in machine learning . . . 2
1.2 Privacy-preserving techniques . . . 3
1.3 Purpose-aware privacy-preserving techniques . . . 5
1.3.1 Threat model . . . 6
1.3.2 Predictive applications . . . 8
1.4 Contributions of the thesis . . . 9
1.5 Publication related to this thesis . . . 11

I Attacking Input Data **13**

2 Towards User-Oriented Privacy for Recommender System Data: A Personalization-based Approach to Gender Obfuscation for User Profiles **15**
2.1 Introduction . . . 17
2.1.1 User-oriented Paradigm for Privacy Protection . . . 18
2.1.2 Threat Model for Gender Inference . . . 20
2.1.3 Experimental Framework . . . 22
2.2 Background and Related Work . . . 22
2.2.1 Privacy in Recommender Systems . . . 23
2.2.2 Obfuscation for Privacy . . . 24
2.2.3 Gender Inference . . . 26
2.2.4 Fairness and Diversity . . . 26
2.2.5 Imputation for user-item matrices . . . 27
2.3 Personalized Blurring (PerBlur) . . . 28
2.3.1 Standard PerBlur . . . 29
2.3.2 PerBlur with Removal . . . 31
2.4 Experimental Setup . . . 32
2.4.1 Data Sets . . . 32
2.4.2 Evaluation metrics . . . 33
2.4.3 Algorithms and Evaluation Setup . . . 35
2.5 Blocking of Gender Inference . . . 36
2.6 Recommendation Performance . . . 39
2.6.1 Evaluation Procedure . . . 39
2.6.2 Comparing Recommendation Performance . . . 42

2.7 Maintaining Fairness . . . 44
2.8 Achieving Diverse Results . . . 45
2.9 Conclusion and Outlook . . . 47
2.9.1 Summary . . . 47
2.9.2 Future Work . . . 48

3 Data Masking for Recommender Systems: Prediction Performance and Rating Hiding 51
3.1 Introduction to Data Masking with Shuffle-NNN . . . 53
3.2 Experimental Framework . . . 53
3.3 Comparative Algorithm Ranking . . . 54
3.3.1 Ranking Prediction Performance . . . 54
3.3.2 Rating Prediction Performance . . . 54
3.4 Rating Hiding . . . 55
3.5 Conclusions and Outlook . . . 56

II Attacking Output Data 59

4 When Machine Learning Models Leak: An Exploration of Synthetic Training Data 61
4.1 Introduction . . . 63
4.2 Threat Model . . . 64
4.3 Background and Related Work . . . 65
4.3.1 Propensity to Move . . . 65
4.3.2 Privacy in Machine Learning . . . 66
4.3.3 Synthetic data generation . . . 66
4.3.4 Model Inversion Attribute Inference Attack . . . 67
4.3.5 Attribute Disclosure Risk . . . 67
4.4 Label-Only MIA with Marginals . . . 68
4.5 Experimental Setup . . . 68
4.5.1 Data Set . . . 68
4.5.2 Utility Measures . . . 69
4.5.3 Adversary Resources . . . 70
4.6 Experimental Results . . . 71
4.6.1 Performance of Machine Learning Classifiers . . . 71
4.6.2 Results of Model Inversion Attribute Inference Attack . . . 72
4.7 Conclusion and Future Work . . . 75

5 Exploring Privacy-Preserving Techniques on Synthetic Data as a Defense against Model Inversion Attacks 77
5.1 Introduction . . . 79
5.2 Threat Model . . . 80
5.3 Background and Related Work . . . 81
5.3.1 Synthetic data generation . . . 81
5.3.2 Privacy-preserving techniques . . . 82
5.3.3 Model inversion attribute inference attacks . . . 82
5.3.4 Attribute disclosure risk . . . 83

5.4 Experimental Setup . . . 84
5.4.1 Data set . . . 84
5.4.2 Privacy-preserving Techniques on Synthetic Training Data . . . 85
5.4.3 Target machine learning model . . . 86
5.4.4 Model Inversion Attribute Inference Attacks . . . 87
5.5 Performance of the target models . . . 89
5.6 Results of Model Inversion Attribute Inference Attacks . . . 89
5.6.1 Attacks on the model trained on original data . . . 90
5.6.2 Attacks on the model trained on protected synthetic data . . . 90
5.7 Correct Attribution Probability . . . 91
5.8 Conclusion and Future Work . . . 93

6 A Closer Look at User Attributes in Recommendations: Implications for Privacy and Diversity 95
6.1 Introduction . . . 97
6.2 Threat Model . . . 98
6.3 Related Work . . . 100
6.3.1 Context-Aware Recommendation with User Side Information . . . 100
6.3.2 Attribute inference attack in Recommender Output . . . 101
6.3.3 Diversity and Fairness in Recommender Systems . . . 101
6.4 Experimental Setup . . . 102
6.4.1 Data Sets . . . 102
6.4.2 Recommender System Algorithms . . . 103
6.4.3 Classification Algorithms . . . 105
6.5 Leakage in the Output of Standard Recommenders . . . 105
6.6 Leakage in the Output of Context-Aware Recommenders . . . 106
6.6.1 Context-Aware Recommendation with User Side Information . . . 107
6.6.2 Measuring Leaks in Context-aware Recommenders . . . 108
6.7 Diversity and Coverage of the Recommender Output . . . 110
6.8 Countering privacy leaks . . . 111
6.8.1 GNN-based recommendation . . . 111
6.8.2 Post-processing methods . . . 113
6.9 Conclusion and Future Work . . . 114

III Conclusion and Future Work 117

7 Outlook 119
7.1 Main contributions and Discussion . . . 119
7.2 Future Work . . . 122
7.3 Reflections . . . 124

Bibliography 129

Propositions 149

Acknowledgements 151

Curriculum Vitæ 157

List of Publications **159**

SUMMARY

In the field of machine learning (ML), the goal is to leverage algorithmic models to generate predictions, transforming raw input data into valuable insights. However, the ML pipeline, consisting of input data, models, and output data, is susceptible to various vulnerabilities and attacks. These attacks include re-identification, attribute inference, membership inference, and model inversion attacks, all posing threats to individual privacy. This thesis specifically targets attribute inference attacks, wherein adversaries seek to infer sensitive information about target individuals.

The literature on privacy-preserving techniques explores various perturbative approaches, including obfuscation, randomization, and differential privacy, to mitigate privacy attacks. While these methods have shown effectiveness, conventional perturbation based techniques often offer generic protection, lacking the nuance needed to preserve specific utility and accuracy. These conventional techniques are typically purpose unaware, meaning they modify data to protect privacy while maintaining general data usefulness. Recently, there has been a growing interest in purpose-aware techniques. The thesis introduces purpose-aware privacy preservation in the form of a conceptual framework. This approach involves tailoring data modifications to serve specific purposes and implementing changes orthogonal to relevant features. We aim to protect user privacy without compromising utility. We focus on two key applications within the ML spectrum: recommender systems and machine learning classifiers. The objective is to protect these applications against potential privacy attacks, addressing vulnerabilities in both input data and output data (i.e., predictions). We structure the thesis into two parts, each addressing distinct challenges in the ML pipeline.

Part I tackles attacks on input data, exploring methods to protect sensitive information while maintaining the accuracy of ML models, specifically in recommender systems. Firstly, we explore an attack scenario in which an adversary can acquire the user-item matrix and aims to infer privacy-sensitive information. We assume that the adversary has a gender classifier that is pre-trained on unprotected data. The objective of the adversary is to infer the gender of target individuals. We propose personalized blurring (PerBlur), a personalization-based approach to gender obfuscation that aims to protect user privacy while maintaining the recommendation quality. We demonstrate that recommender system algorithms trained on obfuscated data perform comparably to those trained on the original user-item matrix. Furthermore, our approach not only prevents classifiers from predicting users' gender based on the obfuscated data but also achieves diversity through the recommendation of (non-stereotypical) diverse items. Secondly, we investigate an attack scenario in which an adversary has access to a user-item matrix and aims to exploit the user preference values that it contains. The objective of the adversary is to infer the preferences of individual users. We propose Shuffle-NNN, a data masking-based approach that aims to hide the preferences of users for individual items while maintaining the relative performance of recommendation algorithms. We demon-

strate that Shuffle-NNN provides evidence of what information should be retained and what can be removed from the user-item matrix. Shuffle-NNN has great potential for data release, such as in data science challenges.

Part II investigates attacks on output data, focusing on model inversion attacks aimed at predictions from machine learning classifiers and examining potential privacy risks associated with recommender system outputs. Firstly, we explore a scenario where an adversary attempts to infer individuals' sensitive information by querying a machine learning model and receiving output predictions. We investigate various attack models and identify a potential risk of sensitive information leakage when the target model is trained on original data. To mitigate this risk, we propose to replace the original training data with protected data using synthetic training data + privacy-preserving techniques. We show that the target model trained on protected data achieves performance comparable to the target model trained on original data. We demonstrate that by using privacy-preserving techniques on synthetic training data, we observe a small reduction in the success of certain model inversion attacks measured over a group of target individuals. Secondly, we explore an attack scenario in which the adversary seeks to infer users' sensitive information by intercepting recommendations provided by a recommender system to a set of users. Our goal is to gain insight into possible unintended consequences of using user attributes as side information in context-aware recommender systems. We study the extent to which personal attributes of a user can be inferred from a list of recommendations to that user. We find that both standard recommenders and context-aware recommenders leak personal user information into the recommendation lists. We demonstrate that using user attributes in context-aware recommendations yields a small gain in accuracy. However, the benefit of this gain is distributed unevenly among users and it sacrifices coverage and diversity. This leads us to question the actual value of side information and the need to ensure that there are no hidden 'side effects'.

The final chapter of the thesis summarizes our findings. It provides recommendations for future research directions which we think are promising for further exploring and promoting the use of purpose-aware privacy-preserving data for ML predictions.

1

INTRODUCTION

Parts of this chapter are published as Manel Slokom. Comparing recommender systems using synthetic data. In Proceedings of the 12th ACM Conference on Recommender Systems. 2018.
Giuseppe Garofalo, Manel Slokom, Davy Preuveneers, Wouter Joosen, Martha Larson. Machine Learning Meets Data Modification: the Potential of Pre-processing for Privacy Enhancement. In Security and Artificial Intelligence (pp. 130-155). Springer. 2022.

1

The huge amount of data online has played a big role in shaping artificial intelligence (AI). On one side, social websites, e-commerce, and media services collect lots of data from people every day. This data becomes a valuable resource for training and improving algorithms. On the other side, improvements in machine learning (ML) help create models that can understand complex patterns in these large data sets.

The mainframe of the ML pipeline, illustrated in Figure 1.1, depicts the three fundamental components: input data or training data, models, and output data (or predictions). The ML pipeline involves collecting input data and employs optimization methods to train models capable of extracting important features and patterns from the input training data. Then, the model is used to extract knowledge and generate predictions.

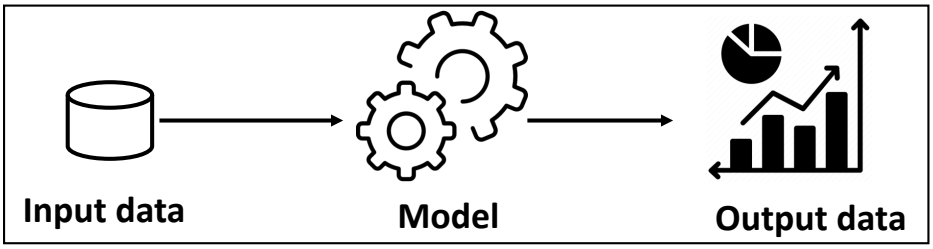


Figure 1.1: Machine learning pipeline with three components: input data, model, and output data.

However, the advancement of ML models is not without its share of challenges. These challenges include issues such as data quality and quantity, bias, interpretability, model complexity, security, and privacy [1], [2]. This thesis will focus on the susceptibility of the ML pipeline to various vulnerabilities and attacks that violate users' privacy.

1.1. PRIVACY RISKS IN MACHINE LEARNING

Privacy risks refer to potential threats or vulnerabilities that can compromise the confidentiality of sensitive information [3], [4]. These risks arise when there is a possibility of unauthorized access, disclosure, or misuse of data, leading to violations of the privacy of individuals. In the context of the ML pipeline, privacy risks can manifest in various forms, such as identity disclosure, attribute disclosure, or attribute inference.

Identity disclosure refers to the risk of an individual's identity being revealed through the analysis of anonymized data. This occurs when an adversary successfully links some of their information on an individual with the corresponding data in the anonymized data set [4]. Attribute disclosure or attribute inference attack is another privacy risk. Attribute disclosure occurs when an adversary is able to infer specific information about an individual. In this case, the focus shifts from revealing the identity to exposing or inferring sensitive or personal information associated with an individual [4]–[6]. These privacy risks manifest in various attacks, including re-identification attacks [6], [7], attribute inference attacks [6], [8]–[10], membership inference attacks [11], and model inversion attacks [12], [13]. Each poses unique challenges to individual privacy and highlights the need for robust privacy protection measures in ML applications.

In this thesis, we will focus on attribute inference attacks. We study attacks that

threaten to reveal sensitive information of the users that is explicitly or implicitly present in the input data i.e., the data used for training the algorithms. Our primary goal is to protect users' data against potential privacy attacks that may arise across the ML pipeline. We specifically focus on attacks that could happen to both the input data and output prediction data. Our first research question **(RQ1)** looks into *how can we protect sensitive information when the attacker has access to* ***the input data*** *from inference attack while maintaining utility?* This addresses the vulnerability in input data, emphasizing the need to maintain utility despite potential attacks. For attacks on output data, we investigate two distinct scenarios in which we aim to explore potential privacy risks associated with the output data of machine learning classifiers and recommender systems. When considering the output data of machine learning classifiers, our focus shifts to *how can we protect sensitive information when the attacker has access to the* ***model's predictions*** *while maintaining utility?* ***(RQ2)***. For **(RQ1)** and **(RQ2)**, we investigate the threat model, perform the attack, and solutions to protect sensitive information. When considering the output data of recommender system algorithms, we investigate *whether* ***recommender system's output data*** *leaks sensitive information about users* ***(RQ3)***. **RQ3** involves describing the threat model and conducting the attack without proposing a solution. For RQ3, we focus solely on assessing potential privacy risks associated with the output data of recommender systems.

1.2. PRIVACY-PRESERVING TECHNIQUES

Privacy-preserving techniques have been proposed in the literature as a crucial solution to protect against different attacks [14]–[16]. In this section, we briefly present the relevant literature related to privacy-preserving techniques.

Indistinguishability-based methods modify the data to prevent the identification of individuals within a data [17]. By generalizing or suppressing specific attributes within a data set, we can achieve properties such as k-anonymization, t-closeness, and l-diversity. In [18], a privacy-preserving collaborative filtering method combines microaggregation and k-anonymization for secure rating data release. Here, we also mention the Netflix Prize data competition, an anonymization-based technique, that replaced personal details with random numbers for privacy. However, as revealed in [7], 99% of records could be re-identified in 2008. This highlights the ongoing challenge mentioned in [19]: ensuring secure data release for research purposes remains an open problem. Indistinguishability based methods are out of the scope of the thesis.

Data masking techniques use data distortion (a perturbation function) to create a private representation of the data. *Randomization based perturbation* techniques were first introduced in [20] to protect data in collaborative filtering. They showed that using perturbed data, which has been perturbed such that no certain information about the ratings can be derived, may still yield acceptable recommendations. One of the most popular approaches used for data perturbation is obfuscation, where a certain percentage of users' preferences is replaced by randomized values. In [21], the authors proposed a framework for privacy-preserving recommendations using data obfuscation. In [22], [23], the authors proposed a new method called "BlurMe", that adds ratings to a user's profile to make it hard to infer the user's gender, while causing a minimal impact on recommendation quality. Besides perturbative masking and obfuscation, techniques

like swapping and suppression [24] are explored. However, one of the major problems in data perturbation techniques is that the addition of noise is inapplicable to binary data [19]. Also, using these techniques it is difficult to calibrate the magnitude of noise and it may hide key relationships in the data. Finally, in [25], the authors discussed that traditional methods (perturbative and non-perturbative) like global recoding, local suppression, and top-coding would yield too much loss of detail to produce protected data.

Differential privacy (DP)-based approaches provide a solution for protecting data sets, particularly in scenarios where the data is stored in a database, and access is exclusively through queries. DP was originally proposed for interactive statistical queries to a database [26]. Authors in [27] were the first to study the application of differential privacy to a collaborative recommendation algorithm. They used the Laplace Transform mechanism to add noise to the covariance matrix, protecting the original nearest neighbors. Later, the authors of [28] proposed an approach called *Private Neighbor Collaborative Filtering* (PNCF). In PNCF, the authors introduced recommendation-aware sensitivity and re-designed differential privacy mechanisms to select the nearest neighbors. However, the major drawback of this method lies in balancing data quality and privacy level, as the protection level is often too high to ensure data quality [29]: excessive noise can adversely affect the output, while less noise fails to hide user contributions [19]. Similarly, recent research has extended the application of DP beyond its original interactive setting, into other use cases such as data release [30], [31]. However, applying Differential Privacy (DP) to record-level data release or collection demands a large value of ϵ. It is recommended to use $\epsilon < 1$ to obtain a meaningful privacy guarantee; however, in this case, the analytical utility of DP outputs is likely to be very poor [32].

Cryptography-based approaches that have been used in machine learning include homomorphic encryption and or secure multiparty computation to hide users' private data. Several protocols have been applied in different machine learning and recommender system scenarios, such as a distributed setting [33], [34], a setting including a privacy service provider [35], and a client-server setting [36]. A drawback of cryptography-based approaches is that they require significant computational resources and time which make these protocols suitable mainly for offline evaluation, i.e., recommendations (not for online recommendations) [19]. In addition, unnecessary computational cost impacts and limits its application by normal users [29]. Further cryptography requires a key that can potentially be lost or stolen. Fully Homomorphic Encryption (FHE) and Secure Multi-Party Computation (MPC) fall outside the scope of our research.

Synthetic data generation is an alternative approach to protecting data while preserving the statistical properties of the original data set. Synthetic data generation methods first construct a model of the data and then generate artificial values for this model. Recent techniques for synthetic data generation can be divided into three categories [37]–[39], namely partially synthetic methods, fully synthetic methods, and hybrid methods. Fully synthetic data, created entirely anew, maintains privacy by replacing the original data set [37]. Fully synthetic data ensures a low disclosure risk. In contrast, partially synthetic data sets combine original and synthetic values, replacing only high-risk variables [40]. Although disclosure risk is higher than fully synthetic sets, utility is typically better. Hybrid masking combines original and synthetic data linearly, offering precise control over individual characteristics [41].

1.3. PURPOSE-AWARE PRIVACY-PRESERVING TECHNIQUES

While conventional privacy-preserving techniques have effectively protected users' data, they often struggle with the challenge of maintaining a balance between privacy and utility. The trade-off between these measures is a popular topic under continuous research [42]. Conventional privacy-preserving techniques often provide a generic level of protection that may not guarantee to maintain the specific utility or accuracy required [43]. Moreover, these techniques are generally purpose-unaware, meaning they are not specifically tailored to the intended purpose or context of data usage. Recognizing this limitation, we have adapted conventional privacy-preserving techniques to better align with specific purposes.

Purpose-unaware vs. purpose-aware privacy-preserving techniques. Purpose unaware techniques employ an indiscriminate noise, i.e., a broad and uniform application of noise, e.g., randomization, perturbation. For instance, some randomization approaches focus on single-dimensional perturbation and assume independence between attributes [44]. The traditional data perturbation approach distorts each data element independently. As a result, the distance between data records is not preserved, and the perturbed data cannot be used for many machine learning applications [45]. While effective in providing a generic level of privacy, they lack the sophistication to tailor protection according to the individual use cases [46]. Simultaneously, with the growth in the domain of machine learning where various data types from structured to unstructured coexist, the shortcomings of purpose-unaware methods become apparent. Take, for instance, recommender system data represented as a sparse user-item matrix, where the challenge lies in capturing hidden patterns and similarities. Applying purpose-unaware perturbation to such data can hinder the performance of recommender systems or machine learning algorithms in general. The process of selecting the optimal perturbation algorithm for a specific problem is known to be complex, involving various trade-offs [46].

We address the challenge by proposing a conceptual framework for purpose-aware privacy-preserving techniques that provide targeted protection against specific threats. The protection solution is aimed at a predefined set of risks, ensuring robust protection against a defined threat model. Purpose-aware privacy-preserving techniques use perturbative techniques with a predefined 'purpose' in mind. This purpose, representing the desired function of the modified data, is known before the perturbation is applied. Purpose-aware privacy-preserving techniques ensure that the modified data retains its usefulness for the intended function. In works such as [43], [47], the authors propose utility-aware privacy perturbation schemes that share similarities with our work and could fit into our framework, particularly in their emphasis on specifying the purpose of data usage. In [47], the authors propose a utility-aware data perturbation scheme based on attribute partition and budget allocation. Their three-step procedure involves quantifying attribute privacy and importance, attribute partitioning, and budget allocation, leveraging information entropy, and preserving attribute correlations. In [43], the authors present a two-step perturbation-based utility-aware privacy-preserving data-releasing framework. They apply perturbation to the original data to ensure its successful use for an intended purpose (learning to succeed) while preserving predefined privacy requirements.

Rethinking the privacy-accuracy trade-off. Purpose-aware privacy-preserving techniques aim to break from the traditional privacy-accuracy trade-off and offer a way to work on solutions that can tailor both. Within purpose-aware privacy-preserving techniques, we make *changes* (also referred to as *modifications*) to the data such that it is still useful for some purpose (training a particular type of model) but with minimal privacy-sensitive information. Changes or modifications could be any privacy-preserving technique where the focus is on protecting sensitive information that the data contains. Such purpose-aware privacy preservation has the potential to be particularly effective by introducing changes along any dimensions that are (nearly) orthogonal to the relevant features.

A key component in our purpose-aware privacy-preserving framework is the careful outlining of the threat model. Within the threat model, we define the adversary and specify the purpose for which we apply for protection while maintaining accuracy (more details in section 1.3.1). We note that while our framework aims to protect against specific threats, it does not claim universal protection against all possible attacks. Similarly, it does not assure that the protected data is impervious to unintended use or repurposing. In the next sections, we first provide an overview of threat model formulation. Next, we describe the two machine learning applications that we explore in this thesis: recommender systems and ML classifiers. Then, we follow up with the contributions and publications related to the thesis.

1.3.1. THREAT MODEL

This thesis is built on the basis of the threat model, which is the foundation of our research. The threat model serves as the guiding framework for formulating purpose-aware privacy-preserving techniques, ensuring robust protection against privacy breaches.

A threat model is a theoretical framework defining what is considered to be a privacy violation or breach i.e., identity is linked to a record, leaking sensitive information. In [48], the authors provided a widely used schema for defining a threat model. First, the threat model describes the *adversary*, including the resources at the adversary's disposal and the adversary's objective. In other words, the threat model specifies what the adversary is capable of and what the goal of the adversary is. Second, it describes the *vulnerability*, including the opportunity that makes an attack possible and the nature of the countermeasures that can be taken to prevent the attack. Note that, throughout the thesis, we use attacker and adversary interchangeably.

Adversary's objective determines what an adversary wants to do. An adversary can aim at different goals. For instance, given access to data, an adversary will be interested in identifying a target individual [6], [7], or an adversary could be interested in inferring sensitive information about target individuals [9], [10], [49]. In this thesis, we specifically focus on *inference attacks*. We use the term inference attack to refer to the use of an inference algorithm to infer something that a user may consider private. For instance, the adversary seeks to infer individuals' sensitive information such as demographic attributes i.e., gender, age, and income, or individuals' orientation i.e., political, sexual.

Adversary's resources determines what an adversary can do. An adversary can have different levels of knowledge and resources of the system. This knowledge influences the extent to which the adversary can compromise individuals' sensitive information. We distinguish between different levels of knowledge of the system [12], [49], [50]. In the worst scenario, an adversary is assumed to have full knowledge of the pipeline. This is called a white-box scenario. In this scenario, an adversary is assumed to know about the input training data, *and* the algorithm used for training including its parameters and architecture. In a more realistic scenario, an adversary is assumed to have partial knowledge of the system. This is called a gray-box scenario or black-box scenario. In a gray-box scenario, an adversary is assumed to know the input training data *and or* the algorithm and its parameters and architecture. In a black-box scenario, an adversary knows either about the input training data *or* the algorithm. In this thesis, we focus on black-box scenarios. We consider two distinct scenarios: one in which the adversary leverages ground truth data from social media and trains a basic classifier and another where the adversary possesses black-box access to a target model.

- **Ground truth collection and basic classifier training** Here, we assume that for a large enough number of users, the adversary is able to gather their demographic attributes, i.e., gender, age, income, on social media to use as ground truth [8]. We assume that the adversary has the ability to train a simple machine learning classifier. The availability of ground truth online is not an unrealistic assumption. We recall the case of Netflix de-anonymization using data scraped from the Web [7]. In this thesis, we focus on a case in which the adversary has a subset of data and is able to train a classifier. This assumption is also assumed in literature [22], [51].

- **Black-box access to a target model** We assume that the adversary can query a machine learning model and get its predictions output as class labels or confidence-scores [12], [13], [52], [53].

Vulnerability- Opportunity determines what an adversary is willing to do. We distinguish between three different possible opportunities (or vulnerabilities in the system) that might be available to the adversary to infer sensitive information about target individuals:

- **The possession of original input data** In this scenario, the adversary is assumed to have access to the raw input data that will be used to train different models. For instance, an obvious way in which the adversary can acquire the input data is via unauthorized access or data breach. Also, the adversary may be internal to the platform. In this thesis, we assume that the adversary has access to the recommender system input, i.e., called user-item matrix via a breach of the recommender system platform. Then, the adversary can train a classifier on the user-item matrix.

- **The ability to query a target model to get its output data** In this scenario, the adversary is assumed to have black-box access to target machine-learning predictions. The adversary can perform model inversion attacks. Model inversion

attacks aim to expose sensitive information inherent in the training data of a prediction model [50], [54]. In this thesis, we assume that the adversary has the access necessary to query a target machine learning model and get predictions (class labels and confidence scores) about target individuals' propensity to move. Having access to the model parameters and architecture is out of the scope of the thesis.

- **The ability to intercept output data of a recommendation system algorithm** In this scenario, the adversary cannot actively query a recommender system but has access to the output data. The adversary is assumed to be able to intercept recommendation lists generated by recommender system platforms.

Vulnerability- Countermeasure represents potential solutions that could be used to protect against a specific attack. The potential solution can be achieved by applying privacy-preserving techniques to the data. Throughout the thesis, we use different terminology to refer to the protected data. For instance, we refer to obfuscated data, for protected data on which we applied obfuscation techniques [22], [51], [55], [56]. We consider masked data to be data on which we applied data masking techniques such as shuffling [57] and swapping [58]. Perturbed data is data on which we have applied privacy-preserving perturbation techniques such as randomization [14]. Lastly, we refer to synthetic data as artificial data that reassembles the original data and mimics its structure and property [38], [59], [60].

1.3.2. PREDICTIVE APPLICATIONS

Within the larger area of machine learning, we focus on two types of predictions: (1) Predictions using recommender system algorithms, and (2) Predictions using machine learning classifiers.

Predictions using recommender system algorithms Examining recommender systems, we look at rating prediction and ranking prediction also called TopN recommendation. In the former, the objective is to predict a user's ratings on items, such as predicting movie ratings (or stars) in movie recommendation systems like Netflix or predicting user ratings for restaurants in food recommendation apps. The latter focuses on making personalized recommendations by providing a user with a list of top-N recommendations, such as suggesting personalized playlists in music streaming services like Spotify or recommending products in e-commerce platforms like Amazon. Our emphasis lies in protecting user profiles against inference attacks while maintaining the performance of the recommender systems. Additionally, our investigation extends towards achieving diversity and fairness in recommendations. We aim to achieve diverse recommendations by increasing item coverage and recommending less stereotypical items, and we aim to promote fairness by ensuring equitable recommendation performance to different groups of users.

Predictions using machine learning classifiers With regards to machine learning, we look at predicting an individual's likelihood of relocating or changing their place of residence (referred to as propensity-to-move). This case study involves different data sources

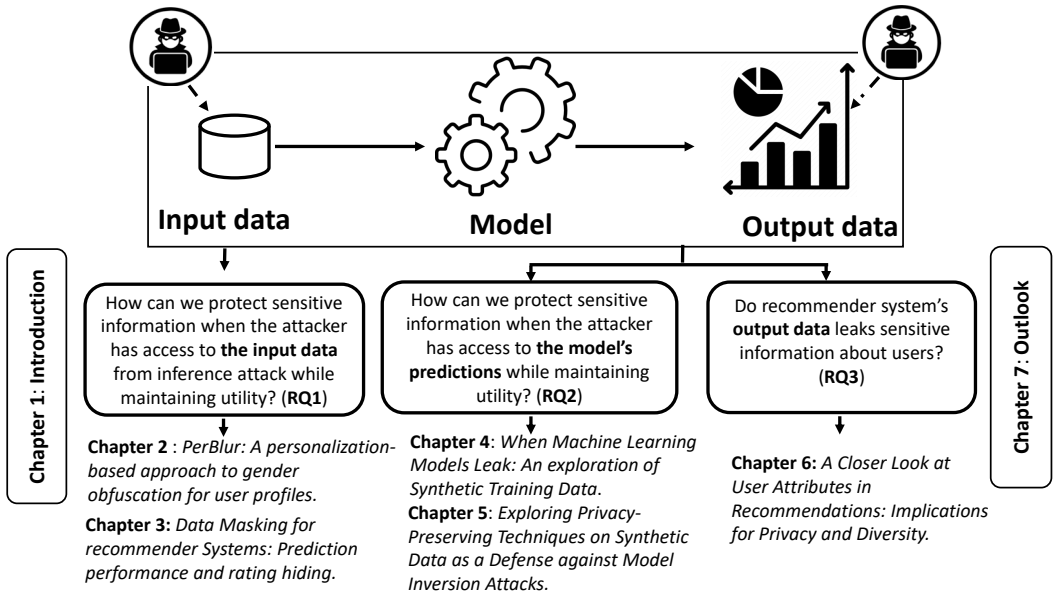
from the Dutch System of Social Statistical Data sets (SSD) [61]. The study includes evaluating various machine learning algorithms for propensity-to-move prediction. The primary objective is to assess the possibility of releasing predictions of the target machine learning without privacy concerns. This promotes transparency in the inference process, which makes it possible to assess and address possible issues of bias in machine learning models.

1.4. CONTRIBUTIONS OF THE THESIS

In this section, we elaborate on how the research questions outlined earlier are tackled in various chapters throughout this thesis.

We structure this thesis into two parts focusing on different attacks targeting the ML pipeline, attacking input data and attacking output data. Figure 1.2 shows different parts of the thesis, the main research questions, and the corresponding chapters. The first

Figure 1.2: Thesis road-map. The figure shows our research contributions, divided into two main parts with three main research questions: 1) *Attacking Input Data*, focusing on adversary access to input data, and 2) *Attacking Output Data*, involving adversary access to either the model predictions via querying or recommendation lists via interception a recommender system platform.



part explores attack models attacking input data. In part I, incorporating Chapter 2 and Chapter 3, we focus on addressing RQ1. Specifically, our focus lies in exploring methods to protect sensitive information when the attacker has access to the input data, mitigating inference attacks while maintaining utility.

Chapter 2 explores a threat model, where an attacker can acquire the entire user–item matrix via a breach of the recommender platform. As resources, we assume that the at-

tacker has a gender classifier that is pre-trained on unobfuscated data. The objective is to infer the gender of individual users. We propose PerBlur, a simple, yet effective personalized-based approach to gender obfuscation that aims to protect user privacy while maintaining the recommendation quality. We demonstrate that recommender system algorithms trained on obfuscated data can achieve performance comparable to what is achieved when they are trained on the original user-item (UI) Matrix. We show that a classifier can no longer use the obfuscated data to predict the gender of users. This indicates that implicit gender information has been removed from the UI matrix. Last but not least, we demonstrate the ability of our solution to recommend more diverse items.

Chapter 3 explores a threat model, where an attacker has access to released data i.e., data science or recommender system challenge. The objective is to infer the preferences of individual users. We introduce a data masking approach called *Shuffling Non-Nearest-Neighbors* (Shuffle-NNN) that modifies the data so that it no longer contains precise information about which user has interacted with which item. At the same time, Shuffle-NNN aims to maintain the usefulness of the data for the purpose of training and testing recommender systems, which is necessary to carry out research. Shuffle-NNN generates a masked data set by changing a large portion of the values of the preferences in a user's profile. Specifically, Shuffle-NNN aims to preserve item-item similarity information, based on the assumption that this information is the most important pattern that needs to be present in the data in order to train and test a recommender system algorithm. Shuffle-NNN applies a data shuffling technique that hides (i.e., changes) the preferences of users for individual items. We demonstrate that the relative performance of a set of recommender system algorithms, which is the key property that a data science challenge must measure, is comparable between the original data and the data masked with Shuffle-NNN.

The second part investigates other attacks attacking two different types of output: One output can be acquired through querying an ML classifier, while the other output can be obtained through the interception of a recommender system platform. In part II, we focus on answering the following research questions RQ2 and RQ3. In Chapter 4 and Chapter 5, we focus on RQ2 which is related to protecting sensitive information when the attacker has access to the ML's predictions while maintaining utility. In Chapter 6, we focus on RQ3 in which we investigate whether the recommender system's output data (recommendation list) leaks sensitive information about users.

Chapter 4 investigates privacy risks associated with model inversion attribute inference attacks. Specifically, we explore a case in which a governmental institute aims to release a *propensity-to-move model* trained machine learning model to the public (i.e., for collaboration or transparency reasons) without threatening privacy. We investigate the potential leaks that could be associated with releasing predictions of machine learning models. The attack assumes that the adversary can query the model to obtain predictions and that the marginal distributions of the data on which the model was trained are publicly available. The attack also assumes that the adversary has obtained the values of non-sensitive attributes for a certain number of target individuals. We explore how replacing the original data with synthetic data when training the model impacts how successfully the attacker can infer sensitive information.

Chapter 5 extends Chapter 4 by looking at other model inversion attribute inference attacks. To further understand the disclosure risk associated with the release of a trained ML model, we evaluate several existing model inversion attribute inference attacks that an attacker can use to infer sensitive information. The attack models differ in terms of the resources and opportunities available to the attacker. Our results first show that there is a potential leak of sensitive information, i.e., gender, age, and income when a model is trained on original data. To address this privacy risk, we propose a data synthesis + privacy preservation approach: we replace the original training data with synthetic data on top of which we apply privacy-preserving techniques. Our results show that the propensity-to-move model trained on protected data (data synthesis + privacy preservation) achieves performance comparable to the model trained on original training data. By utilizing privacy-preserving synthetic data to train the target model, before its release, we observe a reduction in the efficacy of certain model inversion attribute inference attacks measured over a group of target individuals.

Chapter 6 investigates the possibility of inferring sensitive information from recommender system output. We look at user attributes from the point of view of privacy and diversity. Our aim is to gain insight into possible unintended consequences of using user attributes as side information in context-aware recommenders. With respect to privacy, our study seeks to understand the extent to which personal attributes of a user can be inferred from a list of items recommended to that user. We are concerned about whether the use of user attributes as side information in context-aware recommendations increases the risk of exposure of users' personal information. We experiment with several categories of user attributes: gender, age, occupation, and location. With respect to diversity, we investigate the effect of user attributes on the usefulness of recommendation lists for users. We demonstrate that using user attributes in context-aware recommendations yields a small gain in accuracy. However, the benefit of this gain is distributed unevenly among users and it sacrifices coverage and diversity.

Chapter 7 concludes the thesis and provides an outlook towards the open research challenges that remain in the domain of purpose-aware privacy-preserving data.

1.5. PUBLICATION RELATED TO THIS THESIS

The list of papers below constitutes the body of the thesis. The content presented in Chapter 2 to Chapter 5 is based on original publications, to which the references are given below. Chapter 6 is currently under preparation. The used terminologies may vary slightly across chapters. Also, the background and related work sections in different chapters may be similar in terms of argumentation and the material they cover.

1. **Manel Slokom**, Alan Hanjalic, and Martha Larson. Towards User-Oriented Privacy for Recommender System Data: A Personalization-based Approach to Gender Obfuscation for User Profiles. Information Processing & Management Journal. 2021 - [**Chapter 2**]

2. **Manel Slokom**, Martha Larson and Alan Hanjalic. Data Masking for Recommender Systems: Prediction Performance and Rating Hiding. Late-breaking results paper, at ACM International Conference on Recommender Systems. 2019 - [**Chapter 3**]

3. **Manel Slokom**, Peter-Paul de Wolf, and Martha Larson. When Machine Learning Models Leak: An Exploration of Synthetic Training Data. International Conference on Privacy in Statistical Databases. 2022 - [**Chapter 4**]

4. **Manel Slokom**, Peter-Paul de Wolf, and Martha Larson. Exploring Privacy-Preserving Techniques on Synthetic Data as a Defense against Model Inversion Attacks. Information Security Conference. 2023. - [**Chapter 5**]

5. **Manel Slokom**, Jesse Brons, Özlem Özgobek and Martha Larson. A Closer Look at User Attributes in Recommendations: Implications for Privacy and Diversity. Under preparation - [**Chapter 6**]

I

ATTACKING INPUT DATA

2

Towards User-Oriented Privacy for Recommender System Data: A Personalization-based Approach to Gender Obfuscation for User Profiles

This chapter is published as Manel Slokom, Alan Hanjalic, and Martha Larson, Towards user-oriented privacy for recommender system data: A personalization-based approach to gender obfuscation for user profiles, Information Processing & Management, vol. 58, no. 6, 2021, ISSN:0306-4573.

In this chapter, we propose a new privacy solution for the data used to train a recommender system, i.e., the user-item matrix. The user-item matrix contains implicit information, which can be inferred using a classifier, leading to potential privacy violations. Our solution, called Personalized Blurring (PerBlur), is a simple, yet effective, approach to adding and removing items from users' profiles in order to generate an obfuscated user-item matrix. The novelty of PerBlur is personalization of the choice of items used for obfuscation to the individual user profiles. PerBlur is formulated within a user-oriented paradigm of recommender system data privacy that aims at making privacy solutions understandable, unobtrusive, and useful for the user. When obfuscated data is used for training, a recommender system algorithm is able to reach performance comparable to what is attained when it is trained on the original, unobfuscated data. At the same time, a classifier can no longer reliably use the obfuscated data to predict the gender of users, indicating that implicit gender information has been removed. In addition to introducing PerBlur, we make several key contributions. First, we propose an evaluation protocol that creates a fair environment to compare between different obfuscation conditions. Second, we carry out experiments that show that gender obfuscation impacts the fairness and diversity of recommender system results. In sum, our work establishes that a simple, transparent approach to gender obfuscation can protect user privacy while at the same time improving recommendation results for users by maintaining fairness and enhancing diversity.

2.1. INTRODUCTION

The data used to train a recommender system takes the form of a user-item matrix, where the columns represent items in the collection and the rows represent individual users. Each row contains a user's ratings, or interactions with items, and is referred to as a user profile. The user-item matrix does not explicitly contain specific user attributes such as gender. However, such information is implicit in each profile, since it can be predicted or inferred using machine learning, specifically, a classifier. This information represents a privacy threat for users.

As the user profiles collected and stored by online platforms increase in number and length, classifiers have a larger amount of data available for training and inference, and the privacy threat grows. To counter this threat, we need the right privacy solutions. Less obviously, we need to re-examine our underlying assumptions about user privacy and to be open to a variety of paradigms.

In this chapter, we propose a privacy protection solution for the user-item matrix called Personalized Blurring (PerBlur)[1], which applies individualized obfuscation to user profiles. Obfuscation is a privacy protection approach that uses small changes to mask sensitive information. Our solution is formulated within a user-oriented paradigm of recommender system data privacy, which strives towards privacy that is *understandable*, *unobtrusive*, and *useful* for the user. In Section 2.1.1, we explain the paradigm, present a comparison and contrast with previous work, and motivate our PerBlur approach. In Section 2.1.2, we present our threat model, which formalizes the types of scenarios to which PerBlur applies. Specifically, PerBlur addresses privacy for cases in which an attacker is able to gain control of the entire data set, as occurs with a data breach or with drifting of goals, known as "mission creep". In Section 2.1.3, we explain the experimental framework. Our chapter presents extensive experimental analysis of PerBlur, focusing on its usefulness for the user in terms of recommender system performance, fairness, and diversity. In this work, we focus on gender obfuscation, but PerBlur would also be suited for protecting other sorts of information that can be inferred from user profiles.

The chapter makes the following contributions.

- We introduce PerBlur (Section 2.3) and demonstrate its ability to effectively obfuscate user profiles to protect information on user gender (Section 2.5).
- We propose an evaluation process for obfuscated recommender system data that addresses the challenges of comparing the performance of recommender systems trained on data that has been obfuscated in different ways (Section 2.6.1).
- We show that training recommender systems on obfuscated data leads to little, if any, loss in the quality of the recommendations received by the user, i.e., recommendation performance and that PerBlur is particular effective at maintaining recommender system performance (Section 2.6.2).
- We show the interplay between user-profile obfuscation and fairness (Section 2.7) and diversity (Section 2.8) and demonstrate the potential of PerBlur to contribute in both cases.

[1] GitHub Link: `https://github.com/SlokomManel/PerBlur`

Taken together our experimental analysis constitutes compelling evidence that user-oriented privacy can be achieved with an obfuscation-based method that is useful to users, while remaining understandable and unobtrusive. To our knowledge, our work represents the most convincing case to date for recommender system data privacy within a strongly user-oriented paradigm that prefers simplicity and transparency over formality and complexity.

2.1.1. USER-ORIENTED PARADIGM FOR PRIVACY PROTECTION

The user-oriented paradigm for privacy protection expresses the requirements and priorities underlying our approach to addressing privacy threats that arise when users share their interaction data and recommender system platforms store these data as user profiles. The idea at the foundation of our paradigm is that privacy protection should center on users, serving their needs and allowing them to maintain insight and control. The idea of user-oriented privacy, defined in this way, has been around for at least a decade already in a somewhat weaker form. Two key examples of user-oriented approaches to protecting user profiles are [23], which studies the impact of obfuscation without attempting to protect a specific user attribute, and [22] (BlurMe), which is designed to protect the specific attribute or gender. These contributions make the assumption that users should have some measure of control over the obfuscation of their own profiles.

In this work, we move the design of user-oriented privacy beyond the orientation towards user control, to include other desirable, user-oriented characteristics. Specifically, the user should find privacy protection to be *understandable*, *unobtrusive*, and *useful*. The characteristics are the basis of the design of our personalization-based approach to gender obfuscation for recommender system data.

Table 2.1: User-oriented paradigm for privacy of recommender system data. The paradigm forms the basis for the design of our approach.

Desideratum	**Description**
Understandable	User understands why items have been added to or removed from the profile.
Unobtrusive	Obfuscation should not be pure "noise", but rather be consistent with the user's own preferences.
Useful	Maintain or enhance recommendation performance, fairness, diversity

Our paradigm is summarized in Table 2.1, and next we will discuss each desirable characteristic in turn. The discussion will shed light on the advantage offered by privacy approaches that prefer simplicity and transparency over formality and complexity.

OBFUSCATION SHOULD BE UNDERSTANDABLE

The *understandable* dimension of our paradigm expresses the importance that our paradigm places on approaches that the user can understand. The dimension is based on basic observations about how people protect their own privacy in offline environments. When we are offline, we protect our own privacy by choosing what we reveal about ourselves

and whom to reveal it to. Our choices are based on our intuition and experience of what we can share without getting hurt, and we are not concerned with formal guarantees.

Our paradigm aims to maintain this natural approach to privacy in the online world. We strive for privacy protection that is conceptually simple so that people can form intuitions about it, allowing them to understand, or even choose, information that has been added to or subtracted from their profiles in order to achieve obfuscation. Our approach, PerBlur, is based on the idea, originating from BlurMe of Weinsberg et al. [22], that to obfuscate gender, we should simply extend a user's profile with items that are indicative of the opposite gender. For example, in the movie domain, "Gone with the Wind" is indicative of female users and "Apocalypse Now" is indicative of male users. It is completely transparent to a male user how adding "Gone with the Wind" to his profile will obfuscate his gender.

Our work stands in contrast to paradigms which emphasize formal guarantees. An example is Yang et al. [62], which minimizes privacy leakage under a bound of the negative impact on the recommender system ranking. In this work, minimizing privacy leakage is achieved at the cost of the assumption of the existence of a detailed user profile specifying the information to be leaked. In contrast, PerBlur applies to any user profile without detailed knowledge of the user.

Our experiments demonstrate that it is possible to achieve successful obfuscation and simultaneously maintain recommender system performance with a "rough and ready" choice of an operating point, i.e., by estimating the necessary amount of obfuscation at the level of the collection rather than via a process of iterative optimization. The success of this "rough and ready" approach is quite remarkable, since the current trend is to immediately assume that obfuscation challenges require iterative optimization, i.e., using Generative Adversarial Networks (GANs). In [63], a GAN-based approach to protecting user attributes while maintaining recommender performance is proposed. The work is not directly comparable to our own, since the authors address a different threat model. Our own threat model, which is more formally specified in Section 2.1.2, protects information in the user item matrix. In contrast, [63] protects a combination of the user embeddings and the recommender output. However, this paper is relevant because it shows that we cannot assume that data obfuscated using a GAN-based approach will be capable of enabling the level of recommendation performance achieved using original data. Specifically, the GAN in [63] does not quite reach the precision and recall of the system before obfuscation. With our experiments, we will show that PerBlur, using its "rough and ready" approach to hyperparameter setting, gets very close to the performance with the original data, and in some cases surpasses it. At the same time, PerBlur obfuscation is understandable to the user and it also does not have to be recomputed from scratch as the user continues to rate or interact with items and the profile grows.

We also note that [63] claims that their approach outperforms BlurMe [22]. However, the support for the claim is weak. In [22], it is shown that BlurMe can achieve the recommendation performance achieved using the original, unobfuscated data. We also reach this conclusion on the basis of our experiments. In contrast, [63] lacks discussion of why their implementation of BlurMe falls very far short of the original data in ability to maintain recommendation performance. A possible explanation is that [63] does not adapt BlurMe for their threat model, which would be necessary in order to achieve a fair

comparison.

OBFUSCATION SHOULD BE UNOBTRUSIVE

The *unobtrustive* characteristic expresses the commitment of our paradigm to approaches that do not hamper or otherwise inconvenience or disturb the user. In other words, the user should not perceive the protection as getting in the way. This requirement is in line with previous work [23], [64] that has carried out user evaluation to test whether recommendations using the obfuscated matrix affect the satisfaction of the users.

Here, we incorporate our concern with unobtrusivness into the design of the obfuscation. Specifically, we strive to make obfuscated profiles remain as natural as possible. PerBlur goes beyond BlurMe [22] with respect to the goal of naturalness. Specifically, PerBlur does not draw heavily on the most indicative movies of the opposite gender. For example, "Gone with the Wind" could be used to obfuscate some user profiles, but if every male looking to hide his gender had "Gone with the Wind" in his profile, the obfuscation would become obvious. PerBlur also avoids the larger issue that a male user might not want to have a particular movie in his profile. For example, "Gone with the Wind" romanticizes the US Civil War, and, today, its depictions of the South are understood as racist. A user obfuscating his profile would prefer to have movies that are consistent with his tastes.

PerBlur achieves unobtrustivenss by *personalizing* obfuscation so that it matches the preferences of the user being obfuscated as well as possible. Specifically, the items that extend a user's profile are both indicative of the opposite gender and, at the same time, reflective of the user's preferences. Our paradigm stands in contrast with the paradigm used by nearly every other research effort in the direction of obfuscation for privacy in recommender systems, which obfuscate by introducing noise into the user data. For example, Differential Privacy is explicitly directed at adding noise to user profiles. An example of such an approach is [65]. We do not consider such approaches user-oriented since they miss the chance to attempt to align obfuscation with user preferences.

OBFUSCATION SHOULD BE USEFUL

The *useful* dimension expresses the commitment of our paradigm to serving users needs. First, obfuscation should strive to maintain recommender performance, i.e., the accuracy of the recommended items from the perspective of the user. Most other work on recommender system privacy, agrees on this point. However, within our paradigm we go beyond accuracy. We are also interested on maintaining the usefulness of the recommendations with respect to fairness and diversity. To our knowledge, we are the first work to experimentally demonstrate that recommender data obfuscation can impact the fairness and diversity of recommender systems trained on that data.

2.1.2. THREAT MODEL FOR GENDER INFERENCE

Our goal is to protect user privacy in the case that recommender system data, i.e., the entire user-item matrix, falls into the hands of a party whose goal is to infer gender information about individual users. We call this party the *attacker*. In this section, we specify our goal more formally in the form of a threat model.

We start with some general comments about the conditions under which an attack might occur. Perhaps the most obvious way in which the attacker can acquire the en-

tire user-item matrix is via a breach of the recommender platform. However, it is also possible that the attacker is internal to the platform. For example, a platform might collect user data without the intention to infer gender information. However, the business strategy of the company owning the platform might change, or the company might be bought by another company. In this case, so-called "mission creep" can occur. In other words, the data is used for something other than the original purpose. It is important to note that the privacy threat that we are addressing differs from that addressed by the large portion of the literature on recommender system privacy, summarized for example by [66]. Work such as [20], [21], [23] often aims to improve the privacy of users, but under the assumption that the platform does not lose control of user data. Work such as [67]–[69], adopts a federated learning approach, which assumes the existence of clients, which can also be breached individually.

Our threat model serves to make the scenario we address concrete, and clearly differentiate it from scenarios addressed by other work. Such a threat model is generally used in security and privacy research, and specifies the conditions for which protection is developed and against which protection is tested. Our model is presented in Table 2.2.

Table 2.2: Threat model: Gender inference on user-item data used for recommender systems

Component	Description
Adversary: Resources	The attacker has a gender classifier pre-trained on unobfuscated data or has the data necessary to train one.
Adversary: Objective	The inference of users' gender attribute.
Vulnerability: Opportunity	The possession of a user-item matrix.
Vulnerability: Countermeasure	Obfuscation of the user-item matrix to block the inference of gender.

The threat model follows the main dimensions set out in [48]. First, it describes the adversary, including the resources at adversary's disposal and the adversary's objective. In other words, the threat model specifies what the attacker is capable of and what the goal of the attacker is. Second, it describes the vulnerability, including the opportunity that makes an attack possible and the nature of the countermeasures that can be taken to prevent the attack.

Table 2.2 provides the specifications of our threat model for each of the dimensions. As resources, we assume that the attacker has a gender classifier that is pre-trained on unobfuscated data. The objective is to infer the gender of individual users. The data is unobfuscated because we assume that the attack is *blackbox* in the sense that the attacker does not have access to information about the obfuscation. In our experiments, the gender inference classifier is trained using data drawn from the same sources as user profiles that are subject to attack. This means that our attack is somewhat stronger than what could be expected in the real world, where the attacker would not necessarily have access to data from the same source.

The opportunity for attack is the possession of the entire user-item matrix. We note that anonymization is important but here we are not interested in whether attackers can reconstruct the identity of the users, but rather whether they can infer a gender for each user-ID. Finally, the countermeasure that we are investigating is obfuscation. Note that our focus on obfuscation does not imply that other countermeasures may not be important. For example, encryption protects privacy in the case of a data breach. However, we focus on obfuscation because user's data might actually be partially public, for example, on a social media website, and because encryption does not address the issue of mission creep.

We finish this section with some additional discussion on why we do not strive for privacy with formal guarantees. As previously stated, privacy in the real-world does not offer guarantees. Further, our experiments will show that the trade-off in privacy vs. protection is small, if it exists at all. The implication is that the user can have intuitive confidence without needing a guarantee, circumventing the question of whether the guarantee is understandable. Another interesting consideration is that formal guarantees cover *defenses* but not *meta-defenses*. In other words, formal guarantees capture the degree to which attacks are blocked, but do not cover the goal of motivating the attacker to give up entirely. In a practical situation, we should be interested not only in ensuring that attackers be unsuccessful in inferring gender, but in nudging them to abandon the effort of inferring gender. For example, the incentive for mission creep within a recommender system platform towards gender inference evaporates if gender inference requires large amounts of resources and yields only low quality information. We do not consider meta-defense further here, but mention the issue only for completeness.

2.1.3. EXPERIMENTAL FRAMEWORK

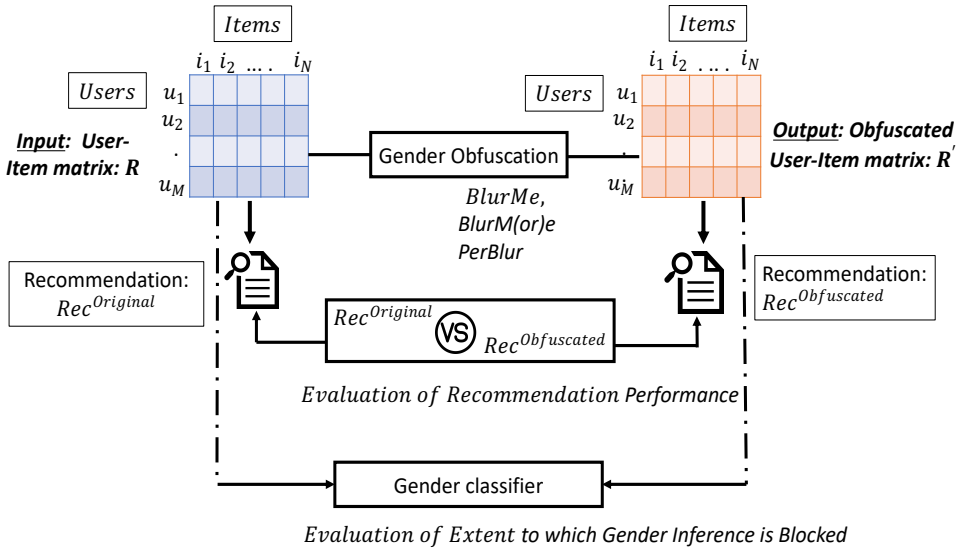
Next we present the framework that we use to carry out our analysis of gender obfuscation and demonstrate the properties of our PerBlur approach. As shown in Fig. 2.1 (top), gender obfuscation takes the original user-item matrix $\mathscr{R}$ and transforms it into the obfuscated user-item matrix $\mathscr{R}'$. In order to be successful, gender obfuscation must fulfill two criteria. First, as indicated by "Evaluation of Recommendation Performance" in Fig. 2.1 (middle), the quality of the predictions produced by the recommender system must be comparable for the original and the obfuscated data. Second, as indicated by "Evaluation of the Extent to Which Gender Information is Blocked" in Fig. 2.1 (bottom), a gender classifier must no longer be able to use the obfuscated data to reliably predict the genders of the users.

In addition to studying recommendation performance, our experiments also analyze obfuscated data with respect to its ability to support the fairness and diversity of recommendations.

2.2. BACKGROUND AND RELATED WORK

In this section, we first give a brief overview of existing work on privacy in recommender systems. Then, we cover previous work on obfuscation for privacy. Next, we provide background on gender inference, and, finally, we discuss related work on fairness and diversity.

Figure 2.1: Gender obfuscation of recommender system data (user-item matrix). Evaluation involves comparing recommendation performance on original and obfuscated data and also confirming the extent to which the inference of user gender from the obfuscated data is reduced or prevented.



2.2.1. PRIVACY IN RECOMMENDER SYSTEMS

Privacy-preserving techniques for recommender systems can be understood as falling into different groups [70]. Here, we discuss several key examples of those groups. *Indistinguishability-based techniques* [18], such as k-anonymity, t-closeness and l-diversity, are designed to protect against re-identification attacks. *Differential privacy based techniques* aim to obscure the link between a users' information in the input (the user's preferences) and output (the recommendation) [70], [71]. McSherry et al. [27], Hua et al. [72] and Friedman et al. [65] proposed different ways to apply differential privacy to matrix factorization that can prevent an untrusted recommender from learning any users' preferences. Hua et al. [72] added noise to item vectors to make them differentially private. Friedman et al. [65] perturbed the input data by introducing noise prior to the data analysis. *Data masking techniques* [20], [73] obfuscate users' information by perturbing the input data. Kandappu et al. [74] proposed "Privacy Canary", an interactive system that enables users to interact and control the privacy-utility trade-off of the recommender system to achieve a desired accuracy while maintaining privacy protection.

In this work, we are not interested in general privacy, but rather in protecting recommender system data. Different techniques have been proposed to protect recommender system data. Some techniques approach the problem from a security point of view. Focusing on securing the *system*, [75]–[77] attempt to prevent attackers from manipulating the recommendation results through the insertion of fake user profiles called profile injection attack. The objective of profile injection attack is to promote (called item push) or demote (called item nuke) the recommendations made for specific items [76]. Bad-

sha et al. [78] and Nikolaenko et al. [79] proposed to protect matrix factorization by applying homomorphic encryption that provides recommendations without knowing the actual ratings. Other techniques focus on protecting recommender system data in order to improve privacy, i.e., prevent the disclosure of users' information. It is important to differentiate between work that protects information implicit in the user-item matrix, such as [80], from work that protects the information implicit in the list of recommendations [63], [81]. Different disclosure attacks have been studied. Here we differentiate between re-identification attacks [7] and inference attacks [9], [82]. In our work, we are interested in protecting the user-item matrix against inference attacks.

2.2.2. OBFUSCATION FOR PRIVACY

DATA OBFUSCATION

Data obfuscation is a privacy preserving technique that aims to hide sensitive information in the data by adding ambiguous, confusing, or misleading information [55] in order to prevent inference attacks and sensitive information leakage. Obfuscation can be applied to different domains with different input data such as online social networks [64], location-based services [83], photos [84], text [85], and recommender systems [21]–[23], [51], [86].

In this chapter, we focus on data obfuscation for recommender systems research. Our work is most closely related to the following papers. Berkovsky et al. [23] focused on enhancing the privacy of recommender system users by distributing their profiles across multiple repositories and then, obfuscating the user profiles to partially hide the actual user ratings. Berkovsky et al. investigated three data obfuscation strategies: (1) Default obfuscation replaces real ratings in the user profile with a predefined value, (2) uniform random obfuscation replaces real ratings with random values chosen within the range of ratings, (3) distribution-based obfuscation replaces real ratings with values drawn from the distribution of ratings in the data set. In Parameswara et al. [21] a privacy preserving framework is proposed to make it possible for multiple E-commerce services to share data. The data sets are obfuscated by permuting sets of similar items.

We note that obfuscation is different from injection attacks. Obfuscation and injection (shilling) attacks are similar in the sense that they both manipulate the user profile but with different goals. Obfuscation focuses on protecting users' information existing in the user-item matrix (a defense technique) but injection attacks are generally techniques for attacking the recommender systems.

GENDER OBFUSCATION

Gender Obfuscation is a subset of data obfuscation, which aims to protect the privacy of users, while maintaining the utility of the data. Specifically, obfuscation has the goal of making it more difficult to infer the gender of the user from data using a classifier. Gender obfuscation is widely studied as a surrogate for obfuscating other sensitive information such as age or profession.

Gender obfuscation for recommender system data was originally proposed by Weinsberg et al. [22]. This work showed that a recommender system can infer binary gender of a user with high accuracy, based solely on recommender system data (i.e., rated movies). They then proposed an algorithm, called *BlurMe*, which obfuscates the user-item matrix

in a way that blocks this inference while maintaining the performance of rating prediction. The basic idea is to add fictional item ratings to every user profile that are typical for the opposite gender. Tests of BlurMe involved only obfuscating 10% of the data at a time, so the goal was not directly to protect the entire user-item matrix as we attempt to do here.

After BlurMe, Feng et al. [86] introduced a privacy preserving module (called PP module) situated between the recommender system and the user. PP module also adds a set of extra fictitious ratings of items not rated by the given user. Although Feng et al. [86] moves away from the one-size-fits all obfuscation used by BlurMe, it does not propose to leverage imputation for personalized obfuscation, as we do in this chapter. Further, Feng et al. [86] focused on rating prediction and did not propose approaches for Top-N prediction, as we do here. In [62], an approach to obfuscating an entire user-item matrix was proposed, however, this approach necessitates the use of detailed private data from users to determine whether privacy is being leaked.

In previous work, notably BlurMe [22], the goal has been to reduce the accuracy as far as possible. This goal is not particularly helpful to privacy protection. If the accuracy of a binary gender classifier is very low, and if the attacker realizes that the data has been protected, then it is possible to recover reliable gender predictions by simply flipping the classifier decision. In our work, we adopt the position that once the AUC performance has been reduced to 0.5 (where there is not benefit from a flip), then, we have succeeded to block gender classification and it is not necessary to reduce it lower.

There have been a number of approaches to gender obfuscation related to recommender systems. It is important to note, however, that these approaches differ from our work because they are protecting an aspect of the recommender system other than the data, as we do here. We mention these approaches here for completeness. Resheff et al. [87] showed that private demographic information can be leaked via the user representations used by latent factor recommender systems. Resheff et al. adapted an adversarial training framework with which they simultaneously perturb the user vectors in order to harm the readout of the private information and change the recommender parameters until the system is optimized. As mentioned above, Hu et al. [88] adopted an adversarial learning technique to learn a privacy-aware transfer model. The generator represents the attacker who tries to infer the user privacy, while the discriminator is the recommender which learns user preferences and deceives the adversary. In this work, Hu et al. focus on perturbing the representations of the system, rather than the recommender system data, as we do here. Note that in our work the obfuscation approach and the classifier can be considered to stand in an adversarial relationship. However, we do not optimize them together, as would be done with a GAN.

Note that there is some work on gender obfuscation outside of recommender systems. In particular, we mention Chen et al. [64], which focused on online social networks. Chen et al. [64] studied how the adoption of different obfuscation strategies e.g., addition, removal or replacement by different proportions of users affects the inference attacks. We mention this work to demonstrate the viability of obfuscation approaches to privacy.

2.2.3. GENDER INFERENCE

We use the term inference attack to refer to the use of an inference algorithm to infer something that a user may consider private i.e., age, gender, orientations. Most of the users are not aware of the correlation that exists between their public and private data [89]. For example, just from Facebook Likes [82] or ratings given to consumed items [10], [22], [64], [90], an attacker can accurately predict a range of highly sensitive personal attributes including [89]: sexual orientation, ethnicity, religious and political views, age, and gender.

Some authors [91], [92] have studied the problem of inference of user attributes in online social networks. Jia et al. [92] proposed a method called "AttriInfer" that combines both friends and behaviors in a social graph. AttriInfer illustrated that even when only a fraction of users provide publicly their profile attributes (such as location, interests), it is possible to infer these attributes among users who do not disclose them. Bi et al. [93] showed how user demographic traits such as age, gender, and even political and religious views can be inferred based on their search query histories. Bhagat et al. [94] presented a new inference attack that a recommender system could use to infer demographic attributes for private user profiles. In the area of online video systems, a gender inference algorithm [95] was used to infer viewers' gender based on implicit watching history.

2.2.4. FAIRNESS AND DIVERSITY

The goal of fairness is to design algorithms that make fair predictions across various (i.e., demographic) groups [96], [97]. There are different kinds of fairness [98]–[100]: *consumers fairness* (C-fairness): where the recommendations should be fair towards the users in the protected class (as defined by gender, age, nationality, ethnicity, etc.) relative to other users. *Providers fairness* (P-fairness) treat the providers of the items in a fair way [101], and *multi-sided fairness* (CP-fairness) [99], [102] requires fairness to be considered for both consumers and providers. Ekstrand et al. [103] looked at C-fairness by exploring whether different user demographic groups experience similar or different utility from the recommendation system. Ekstrand et al. proposed an empirical analysis of the effectiveness of collaborative filtering recommendation strategies stratified by the gender and age of the users. They found that not all users experience the system in the same way. Mansoury et al. [104], explored different factors (e.g., the user profile size, the entropy of users profiles and the anomaly in rating behavior) that could be associated with the unfairness of performance of recommendation algorithms for males versus females. They showed that neighborhood-based algorithms such as UserKNN and ItemKNN discriminate more against female users. For the provider fairness, Ekstrand et al. [105], [106] looked at the response of collaborative filtering recommender algorithms to the distribution of their input data with respect to the content creator gender. In the context of book recommendation, Ekstrand et al. investigated how recommender systems interact with author gender in book data. In the context of music recommendation, Shakespeare et al. [107] studied the extent to which collaborative filtering recommendation algorithms may increase or decrease artist gender bias. Epps-Darling et al. [108] studied gender representation in music streaming. They found that listeners generally tend to stream fewer female artists than male artists.

Here, we focus on consumer fairness (C-Fairness). Specifically, we are worried about the recommendation system performing well for users of one gender and not for another. We followed the same measures used in [103].

Diversity in recommender systems has been broadly studied in literature [109]–[113]. Generally, diversity applies to a set of items and it has to do with how different the items are with respect to each other [109]. Hansen et al. [112] aimed at shifting users' consumption towards the tail and less familiar content in the context of music streaming. Hansen et al. defined diversity around two factors that influence the consumption of music. First, the *taste similarity* which means how similar a piece of music is to the type of music the user has listened to previously. Second, *popularity* or how many users have recently listened to the piece of music. Mansoury et al. [114] proposed a graph-based approach, FairMatch, that works as a post-processing approach after recommendation generation for improving the aggregate diversity. Aggregate diversity is defined in literature as long-tail recommendation which refers to the fact that the recommender systems should recommend a wide variety of items across all users. FairMatch improved the visibility of high-quality items that have a low visibility in the original set of recommendations. Oliveira et al. [115] proposed an multiobjective optimization solution for music recommendations that are at the same time diverse and similar to user preferences. The recommended lists aim at balancing between the aspects that should be held fixed (maximize similarity with users actual items) and apects that should be diversified (minimize similarity with other items in the recommendation list). Vargas et al. [113] defined novelty and diversity based on three key concepts namely choice, discovery and relevance. Helberger et al. [116] highlighted a number of principles designed for exposure diversity in recommender systems.

Here we study diversity with respect to gender specificity. We look at gender specificity and investigate how to control the number of gender-stereotypical items recommended to users. Our goal is preventing users from getting overrun with items that are stereotypical for their gender. For example, a woman might want to watch one Hallmark Christmas romance movie, and if a recommender system diversifies for gender specificity, it will prevent her recommendation list from being flooded with other Hallmark Christmas romances. In this chapter, the study of gender-stereotypical items diversity is different from popularity. In gender-stereotypical items we compare the recommended items vs. items highly indicative for female (or male) users. There is no direct relation between the list of indicative items and the popularity of items.

2.2.5. IMPUTATION FOR USER-ITEM MATRICES

Imputation approaches are approaches used to fill in the missing values of user-item matrices [117], [118]. The goal of the approaches is to infer missing values in data set in such a way that improves the overall performance of recommender systems trained on that data set [119]. Su et al. [119] proposed two neighborhood based collaborative filtering imputation algorithms called imputed nearest neighborhood CF (INN-CF) and imputed densest neighborhood CF (IDN-CF). INN-CF first finds the most similar users to the target user. Then, it uses the corresponding imputed rating data to make predictions. IDN-CF makes predictions from the imputed densest neighbors (i.e., the users who have rated the most number of items).

In our work, we use imputation to personalize obfuscation. Specifically, we impute in order to derive a confidence score that allows us to choose the items that are added to the profile and also to (in the case of rating data) predict the rating that those items should have. Our choice of imputation is inspired by Su et al. [119]. We point out that our main goal is to obfuscate data, but that imputation actually has the goal of increasing recommender performance. For this reason, we can expect that PerBlur might actually be able to increase recommendation performance.

Evidence of the benefits of imputation has been given by Su et al. [119], who found that imputation boosts the predictive performance for collaborative filtering recommendations. Another example of a related paper that used imputation to improve performance is Yuan et al. [120], which proposed a novel method ISVD to incorporate imputed data into SVD framework. For imputation, ISVD chooses effective neighbors for the users and items based on the similarity relation among users and items. The imputed ratings are produced and then incorporated into the SVD model. Imputation can also provide benefit when used for augmentation. The work in [121], [122] introduced a sparsity-aware data-augmentation strategy that provides more item correlation patterns and hence improves recommendation performance.

2.3. PERSONALIZED BLURRING (PERBLUR)

In this section, we present a basic skeleton for gender obfuscation and also introduce *PerBlur*, our approach to gender obfuscation for recommender system data. The main idea of PerBlur is to obfuscate the gender of a user in the user-item matrix by extending the user's profile in a personalized manner, while simultaneously ensuring that the extension is not typical for the user's gender. Specifically, the standard PerBlur algorithm adds ratings (or interactions) to a user's profile that are consistent with the user's preferences, but are at the same time indicative for the opposite gender. PerBlur has two variants: The standard variant just adds ratings (or interactions), and the variant "PerBlur with removal" removes ratings (or interactions) that are indicative for the user's own gender.

Recall that PerBlur builds on the basic idea of BlurMe [22], which is to obfuscate by adding indicative items for the opposite gender. In our work, BlurMe is also applied differently from the original BlurMe paper [22]. First, we are focused on studying Top-N recommendation, whereas [22] studies exclusively rating prediction. Second, our goal is to protect the entire data set, and we apply obfuscation to all user profiles. In contrast, the goal of [22] is to protect individual users and in [22] obfuscation is applied only to 10% of the data at a time. In our experiments, we show, for the first time, that the basic BlurMe can maintain recommender system performance in the case of Top-N recommendation and also in the case that the entire data set is obfuscated.

PerBlur also builds on the idea of our own previous (preliminary) work, BlurM(or)e [51], which removes ratings to make the additional ratings less obvious and to prevent the user-item matrix from becoming dense, resulting in more naturalistic data. PerBlur introduces innovations beyond BlurMe and BlurM(or)e in two respects: It personalizes the extension of the user profile (personalization) and it also prioritizes the items to remove so that the most typical items for a user's gender are removed first (greedy removal).

Before presenting the details of PerBlur, we first present the basic skeleton of the gen-

der obfuscation, which we will use in our experiments for BlurMe and PerBlur in order to compare the two approaches. Input to the algorithm is the level of obfuscation, p, expressed in terms of the percentage by which the user profile is to be extended, and two lists of indicative items: L_m is the list of indicative items for male users and L_f, is the list of indicative items for female users. The lists are created by training a logistic regression model on labeled training data (the same data that are to be obfuscated). The coefficients $\beta = \{\beta_0, \beta_1, ..., \beta_M, \}$ of the logistic regression capture the extent to which each item is correlated with the class attribute gender. The coefficients are used to select the items for the two lists and order them according to the strength of the association. The higher the coefficient is, the more strongly the item is correlated with the attribute class. We extend user profiles by adding items until they are p percent longer than the original profile. When we are working with rating data (as opposed to implicit data), an added item receives either rating that is predicted for the user (using imputation, which is explained below) or average ratings.

Once an item has doubled its frequency with respect to the original data, it is no longer added. This mechanism is used by BlurM(or)e [51], where it was shown to work well and, for this reason, adopted by PerBlur. We refer to this mechanism as "stop after doubled". The goal is to help to keep the overall distribution of items naturalistic. If "stop after doubled" is not applied, then the items in the top ranks of L_m and L_f will occur in a large number of user profiles, creating a "spike" in the item histogram. Such spikes make it obvious that the data set was obfuscated and conflict with our goal to design a system in which obfuscation is unobtrusive. Such a "stop after doubled" mechanism was not relevant in the original BlurMe [22] work, since only 10% of the data was obfuscated at a time, so the items used for obfuscation would not be obvious in item histogram.

2.3.1. STANDARD PERBLUR

Now we will discuss the specifics of PerBlur. We start with standard PerBlur, which is shown in Algorithm 1. First the algorithm creates personalized lists of indicative items (Lines 1-14). PerBlur is built on the insight that if the items added to the user profile for the purpose of obfuscation could have a close match to user preferences, then recommendation performance has a better chance of being maintained when the obfuscated data is used for training. To this end, PerBlur adds ratings (or interactions) to a user's profile that are consistent with the user's preferences, but are at the same time indicative for the opposite gender. Specifically, PerBlur uses a personalized list of indicative items for each user, $Personalized_L^u$. This list is created by intersecting a personalized list of preferred items for each user with the list of indicative items for the opposite gender (L_f for male users and L_m for female users). The personalized list is a list of items ranked in order of the probability that the user will have rated the item.

To create the personalized list, we need a recommender algorithm that imputes items (i.e., predicts ratings or interactions). In the case of rating data, this algorithm must produce a confidence score (and not just a rating prediction or a ranking score) since we are interested in the chance that a user will rate the item and not the user preference. We turn to the widely used user-based collaborative filtering algorithm (userKNN). UserKNN predicts a rating for the target user u on a given item i by calculating the set of

[2] We used $\theta = 0.6$ for MovieLens, $\theta = 0.45$ for Flixster, and $\theta = 0.4$ for LastFM.

Algorithm 1: Standard PerBlur

Input:

- p: percentage of obfuscation
- Users gender information
- L_f (L_m): list of indicative items for females (respectively males)
- Original user-item matrix $\mathscr{R}$ ($\mathbb{N}$ users, $\mathbb{M}$ items)
- Initial count: user profile size at time $t = 0$

Output: Standard PerBlur user-item matrix $\mathscr{R}'$ ($\mathbb{N}$ users, $\mathbb{M}$ items)

```
                                        // 0.  PerBlur Personalized lists of indicative items
1  Confidence score for recommendation based on UserKNN[2];
2  for (user u in ℕ) do
3      for (item i in 𝕄) do
4          Similarity computation finds nearest neighbor candidates;
5          Sort selected items based on the number of possessed neighbor candidates;
6  List^u_NCounts contains a list of counts for each user u;
7  for (user u in ℕ) do
8      Fix a cutoff on L_f and L_m
9                              // we set the cutoff to Top-50, in the rest of the
       experiments
10     Create new personalized list of indicative items for u: Personalized^u_L;
11     if (u is a Female) then
12                                 // Personalized^u_L = List^u_NCounts ∩ L_m
13         for item i ∈ List^u_NCounts do
14             Personalized^u_L = Personalized^u_L. add (i) if item i ∃ L_m
15     else
16                                 // Personalized^u_L = List^u_NCounts ∩ L_f
17         Do the same steps (Line 9 to 12) but for a Male target user u
18                                  // 1.  Obfuscation:  adding extra items
19 for (user u in ℕ) do
20     count = initial count [u] ∗ p
21     added = 0
22     while added < count do
23         i = picks the item in the first position in Personalized^u_L
24         if 𝒫'[u, i] == 0 then
25             𝒫'[u, i] = value
26             added += 1
27                             // For rating data, the rating value is either predicted
           using imputation or average ratings.
28     Total added += added
```

neighbors nearer than a specific distance threshold, θ, who have also rated this item. We choose UserKNN since the count of the neighbors used to make a prediction for an item is a straightforward choice of a confidence score. We rank the items by count from high to low to arrive at $List^u_{NCounts}$, our personalized list for each user. In order to make the item list effective for obfuscation, we do not use $List^u_{NCounts}$ directly. Rather, we create a final personalized item list ($Personalized^u_L$) for each user u by intersecting $List^u_{NCounts}$ with L_f (if u is male) or L_m (if u is female).

This approach runs risk that the final personalized item list $Personalized^u_L$ contains items that are not particularly specific to the opposite gender (because they are too far down the list L_f or L_m). For this reason, we impose a threshold on L_f and L_m. Note that BlurMe never reaches the bottom of L_f or L_m since it chooses the same items for all the users. PerBlur, however, reaches further down the list since it is attempting to leverage personal items. For this reason, the cutoff L_f and L_m is important for PerBlur, as our experiments will show.

Note that it is important to use an appropriate evaluation pipeline for assessing the performance of recommender systems on obfuscated data. We will discuss this point further in Section 2.6.1. We already state a key point here: imputation is trained and operates on training data only and never predict items in the test set being used to evaluate the recommender system.

2.3.2. PERBLUR WITH REMOVAL

Next, we move to the second variant of PerBlur, namely "PerBlur with removal" shown in Algorithm 2. This algorithm takes data obfuscated by standard PerBlur as input, and removes items. Removal has two goals: First, it keeps the density of the obfuscated data close to the density of the original data. Removal is carried out so that the total number of user ratings (or user-item interactions) for each item remains close to the total number in the original data set. We spread out the items that need to be removed evenly over all users. For users with very short profiles, we do not remove items. In our experiments, we set the threshold defining very short profiles to 20, meaning that in the obfuscated data no user can have less than 20 interactions. Our exploratory experiments demonstrated that the success of obfuscation is not particularly sensitive to the threshold. Note that in standard PerBlur we keep track of the number of added items so that we can remove the same number later.

Second, removal contributes to the obfuscation. In other words, item removal can help to mask the gender of the user. Specifically, removing gender-indicative items in a controlled way, could potentially help to confuse the gender classifier, without unduly impacting recommendation performance. To this end, PerBlur proposes a new removal strategy, *greedy removal*, which removes items in the order of their indicativeness for the gender of the user whose profile is being obfuscated. The greedy removal strategy extends our previous work, BlurM(or)e, [51] which proposed random removal strategy. The random removal strategy chooses items for removal in a random manner.

When we evaluate the ability of obfuscation to block gender inference in Section 2.5, we apply greedy removal to BlurMe for comparison. We compare BlurMe with greedy removal to BlurMe with random removal. We see that removal helps to reduce gender inference and also that its contribution to recommendation lies in the area of improving

Algorithm 2: PerBlur with removal

Input:

- Standard PerBlur user-item matrix $\mathscr{R}'$ ($\mathbb{N}$ users, $\mathbb{M}$ items)
- Removal mode = { Random, Greedy }
- Total added:total number of extra ratings (interactions) added, Interaction count: user profile size after adding p% extra items.
- Removal threshold

Output: User-item matrix $\mathscr{R}''$ ($\mathbb{N}$ users, $\mathbb{M}$ items)

```
1                          // 2. Obfuscation: Removing certain items
2  for user u in 𝒩 do
3      if Interaction count ≥ Removal threshold then
4                          // Removal threshold is chosen by us to be 20.
5          remove count += 1
6  To be removed = Total added / remove count
7                          // To be removed: contains the number of ratings (or
       interactions) that will be removed from individual user profiles.
8  for user u in 𝒩 do
9      if (Interaction count ≥ Removal threshold) then
10         if (u is a Female) then
11             removed = 0
12             while (removed < To be removed [u]) do
13                 i = picks an item from L_f
14                             // i depends on the removal mode: random or greedy
15                 if 𝒳'[u, i]! = 0 then
16                     𝒳''[u, i] = 0
17                     removed += 1
18                             // 𝒳'' is 𝒳' after applying the removal
19         else
20             Do the same steps (Line 5 to 13) but for a Male target user u
```

diversity, discussed in Section 2.8.

2.4. EXPERIMENTAL SETUP

2.4.1. DATA SETS

We test our approach on three data sets. The first two are user-item matrices containing ratings (explicit feedback): MovieLens [123] and Flixster [124]. For MovieLens, we use the MovieLens 1 million (ML1M) release. For Flixster, we select users with at least 15 ratings and movies with at least 20 ratings, which results in a subset of ratings for 2.8K items by 2.4K users. The ratings are between [1,5] for both data sets. The third is a user-item matrix containing interactions (implicit feedback): LastFM data [125]. We use artists as the items. Our experimental data set contains users who listened to at least 10 artists and artists to which at least 10 users have listened. The result is a subset of 884 users and 56K artists. The three data sets contain binary information on user gender, i.e., the gender of a user can be either male or female. We choose these data sets because they contain gender information and because they are publicly available, for reproducibility

purposes. Table 2.3 summarizes the statistics of the data sets that we used. In can be seen that the MovieLens and LastFM data sets are quite sparse (4.47% and 0.01%, respectively), and the Flixster data set is somewhat less sparse (5.49%). Note also that in ML1M and LastFM, there are more male than female users (ML1M: 72% male vs. 28% female and LastFM: 57% male vs. 43% female), but in Flixster there are more female than male users (38% male vs. 62% female).

Table 2.3: Summary of data sets

Data sets	#Users	#Items	#Ratings	#Sparsity (%)	Gender (F/M)
ML1M	6040	3706	1000209	4.47	1709 / 4331
Flixster	2372	2835	369059	5.49	1480 / 890
LastFM	884	55686	655929	0.01	382 / 502

2.4.2. EVALUATION METRICS

The evaluation that we carry out in this chapter measures four different aspects of the obfuscated data: (1) the *success of the obfuscation* (2) how well *recommender system performance* is maintained on obfuscated data, (3) how obfuscation impacts *fairness* by making the difference in the quality of the recommendations between males and females larger, and (4) how obfuscation impacts *diversity* of recommended items with respect to gender-stereotypicality. In this section, we present the metrics that we use for each of these aspects.

SUCCESS OF OBFUSCATION

For gender inference, we compute the Area Under the Curve (AUC) using the mean Receiver Operating Characteristic (ROC) curve computed across ten folds. For the ROC, the true positive rate (TPR or sensitivity) is calculated as the rate of correctly classified male users out of males in the data set and the false positive rate (FPR) is calculated as the rate of users incorrectly classified as male out of females in the data set. The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR on the x-axis. We consider gender obfuscation to be successful when the prediction accuracy is close to the average accuracy of random guessing, i.e., 0.5 , which means that the classifier does not have the ability to separate the classes.

RECOMMENDER SYSTEM PERFORMANCE

In our experiments, we compare recommenders trained on the original data with recommenders trained on data obfuscated with different variants of BlurMe and PerBlur. We carry out rating prediction for comparison with previous work, but our main focus is on Top-N recommendation. The goal is to keep the performance of recommender systems trained on the obfuscated data as close as possible to recommender systems trained on the original data.

Here, we define the metrics that we use. For rating prediction we use mean absolute error (MAE), the mean of the absolute difference between each prediction and rating for all the ratings of users in the test set. If there are n held-out ratings in the test set, the

MAE is computed as follows:

$$MAE = \frac{1}{n} \sum_{u,i} \left| p_{u,i} - r_{u,i} \right|$$

where $p_{u,i}$ is the predicted rating for user u on item i and $r_{u,i}$ is the rating value of user u on item i in the test set.

For top-N recommendation, we use Hit Ratio@10 and Top10.nDCG. To compute Hit Ratio@10 (HR@10), we consider an item a "hit" if it is relevant and is ranked as one of the top-N (N = 10) items that we recommend. In the case of the rating data being used as implicit data, we threshold at 3.5, which means any predicted rating above 3.5 will be considered as relevant (= 1) and below will not be relevant (= 0). HR@10 is defined as the count of hits ($\#Hits$) divided by the total user-item pairs in the test set ($\#counts$).

$$HR = \frac{\#Hits}{\#counts}$$

Note that in order to give a ranking perspective to our rating experiments, we rank according to rating prediction and calculate HR@10, although this method would not be used to generated Top-N recommendations in a practical setting.

Normalized Discounted Cumulative Gain (Top10.nDCG) measures the utility that a user is expected to obtain from a recommender based on that user's estimated utility for individual items and the position in the list at which those items were presented [103]. In order to compute nDCG, first we truncate the recommendation list to 10. Then, we compute the discounted cumulative gain (DCG) of the recommended order and the DCG of the ideal order (iDCG). The $DCG_{L_{Rec},u}$ is defined as:

$$DCG_{L_{Rec},u} = \mu_u(l_1) + \sum_{i=2}^{|L_{Rec}|} \frac{\mu_u(l_i)}{log_2 i}$$

where l_i is the i-th item in the recommendation list L_{Rec} and $\mu_u(l_i)$ is user's u utility for item l_i. We define $\mu_u(l_i)$ as a binary utility: if a user u consumed item i then, $\mu_u(l_i) = 1$. Otherwise, items for which no data is available are assumed to have a utility of 0. Then, the $nDCG_{L_{Rec},u}$ for a recommendation list L_{Rec} generated for a target user u is the ratio of DCG of recommended order ($DCG_{L_{Rec},u}$) to DCG of ideal order ($iDCG_u$).

$$nDCG_{L_{Rec},u} = \frac{DCG_{L_{Rec},u}}{iDCG_u}$$

Note that for the rating data, nDCG is also calculated with respect to thresholded ground truth.

To illustrate the impact of obfuscated data on recommendation performance, we report the gain (+) or drop (−) of the recommender performance on obfuscated data with respect to the recommender performance on original data.

FAIRNESS

Recall that we study fairness in terms of the ability of the system to provide good recommendations for both females and males. To measure fairness, we split users in the test set into female users and male users. Then, we measure $nDCG_F$ and HR_F for female

test users and $nDCG_M$ and HR_M for male test users to illustrate the overall satisfaction obtained by each gender group and the difference between them.

For each gender, we report the gain (+) or drop (−) of the recommender performance on obfuscated data. We also report the absolute magnitude of the difference between the male users' drop and the female users' drop. If an obfuscation strategy is fair, this difference should remain as small as possible. In cases where the obfuscation strategy increases the performance of the recommender system, this difference should also remain small, but it is not as important as in cases where performance is lost.

DIVERSITY

We are interested in keeping the number of gender-stereotypical items that the recommender system recommends to users under control. Our study of diversity, for this reason, is focused on the proportion of correctly recommended items that are gender-stereotypical. In this work, we define a gender-stereotypical recommendation as an item that is highly typical for a particular gender.

We calculate the number of user-item pairs for which the item is correctly recommended to user u and is also considered a highly typical item. For this purpose, we use the top {10, 20, 50} items of the highly indicative items list L_m (if u is a male user) or L_f (if u is a female user).

2.4.3. ALGORITHMS AND EVALUATION SETUP

In this section, we describe the recommender system algorithms and gender inference algorithms that we will use in our experiments.

GENDER INFERENCE ALGORITHM

For gender inference, we choose logistic regression because it is mentioned in literature, Weinsberg et al. [22] and Chen et al. [64], as the best performing classifier for gender inference on recommender system data. This was confirmed by our exploratory experiments. We also report results here on SVM, which was the second strongest classifier in our exploratory experiments. We apply normalization (L^2-norm)[3] to the user-item matrix to scale all ratings to values in [0, 1].

To evaluate gender inference, we carry out ten fold cross-validation (using StratifiedKFold[4]). In every iteration, we train the classifier on 9 folds and we test on the 10th fold. Hyperparameters are selected from the training set with grid search (GridSearchCV[5] from Sklearn). The test results for the classifier are reported in Table 2.4 in terms of AUC. Our goal will be to reduce these scores. Recall that we consider gender inference to have been successfully blocked once AUC has been reduced to the level of a random classifier 0.5. Scores below 0.5 show that the more we add extra ratings (interactions), the lower the inference score is. However, scores lower than 0.5 are not ideal cases of blocking because they could be flipped.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
[5] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Table 2.4: Gender inference results measured in terms of AUC using logistic regression and SVM classifiers on: **original** ML1M, Flixster and LastFM data sets. The $\pm$ represents the standard deviations of the results over different ten folds.

	AUC	
	Logistic regression	*SVM*
ML1M	0.87 ± 0.02	0.82 ± 0.04
Flixster	0.87 ± 0.02	0.81 ± 0.04
LastFM	0.77 ± 0.06	0.72 ± 0.04

RECOMMENDER ALGORITHMS

For our recommendation experiments, we use two state-of-the-art algorithms commonly used in collaborative filtering recommender systems: ALS [126] and BPRMF [127]. ALS is a matrix factorization model trained with alternating least squares. We choose this algorithm because it takes a user-item matrix containing ratings (explicit data) as input. BPRMF is a matrix factorization model trained using the Bayesian Personalized Ranking from implicit data. BPRMF is a learning-to-rank algorithm that optimizes pairwise ranking. This algorithm takes a user-item matrix containing interactions (implicit data) as input. We use the implementations of the Lenskit Python (lkpy) toolkit [128].

To evaluate recommender performance, we randomly sample (without replacement) 80% of the items in each user profile as our training set and 20% as our test set. The hyperparameters for each algorithm (the number of features and the number of iterations for ALS and the number of epochs, batch size and features for BPRMF) are tuned using cross validation on the training set. We evaluate the performance of the recommender system algorithm using a special adaptation of 1+random, which is explained in Section 2.6.1.

2.5. BLOCKING OF GENDER INFERENCE

In this section, we report experimental results that demonstrate the ability of gender obfuscation to block gender inference. Table 2.5 presents the performance of the gender classifier on data obfuscated with different variants of BlurMe and Table 2.6 presents different variants of PerBlur. The classifier is trained on the original data, and tested on obfuscated data. We test four levels of obfuscation, corresponding to adding 1%, 2%, 5%, and 10% extra items to each user profile in the original data. Recall that the indicative items lists L_f and L_m used by BlurMe and PerBlur are selected using logistic regression. Here, we evaluate classification results with respect to that same logistic regression classifier. We also test an SVM in order to confirm that the gender obfuscation transfers to a classifier not used for the selection of the indicative item lists.

Tables 2.5 and 2.6 show that when data is obfuscated with any of the obfuscation approaches, classification performance is lower than on the original data (cf. Table 2.4). Recall that lower classification performance is our goal, since it represents improved user privacy. We can see the impact that obfuscation has on lowering the classification performance is evident for both logistic regression and SVM, confirming that our obfuscation approach is not specific to the logistic regression classifier.

Comparing BlurMe results in Table 2.5, we see that BlurMe with greedy removal out-

Table 2.5: Gender inference results measured in terms of AUC for different **BlurMe** obfuscations (with no removal, random removal, and greedy removal) on ML1M, Flixster, and LastFM data sets. The $\pm$ are standard deviations of the results over ten folds.

Gender Inference		Obfuscation Strategies		**Logistic Regression**				**SVM**			
				Extra Items				*Extra Items*			
Data Sets		**Personalization**	**Removal**	***1%***	***2%***	***5%***	***10%***	***1%***	***2%***	***5%***	***10%***
ML1M	***BlurMe***	*None*	*No removal*	0.76 $\pm$ 0.03	0.69 $\pm$ 0.03	0.48 $\pm$ 0.05	0.22 $\pm$ 0.06	0.74 $\pm$ 0.03	0.67 $\pm$ 0.03	0.42 $\pm$ 0.06	0.16 $\pm$ 0.06
			Random	0.75 $\pm$ 0.03	0.69 $\pm$ 0.03	0.43 $\pm$ 0.05	0.13 $\pm$ 0.05	0.74 $\pm$ 0.03	0.66 $\pm$ 0.02	0.38 $\pm$ 0.08	0.10 $\pm$ 0.04
			Greedy	0.59 $\pm$ 0.03	0.49 $\pm$ 0.03	0.22 $\pm$ 0.04	0.05 $\pm$ 0.02	0.53 $\pm$ 0.03	0.42 $\pm$ 0.03	0.14 $\pm$ 0.04	0.02 $\pm$ 0.01
Flixster	***BlurMe***	*None*	*No removal*	0.65 $\pm$ 0.05	0.59 $\pm$ 0.05	0.41 $\pm$ 0.05	0.19 $\pm$ 0.04	0.62 $\pm$ 0.05	0.55 $\pm$ 0.05	0.35 $\pm$ 0.05	0.14 $\pm$ 0.04
			Random	0.65 $\pm$ 0.05	0.59 $\pm$ 0.05	0.38 $\pm$ 0.05	0.14 $\pm$ 0.03	0.62 $\pm$ 0.05	0.55 $\pm$ 0.05	0.32 $\pm$ 0.05	0.10 $\pm$ 0.03
			Greedy	0.44 $\pm$ 0.06	0.33 $\pm$ 0.05	0.17 $\pm$ 0.03	0.06 $\pm$ 0.02	0.39 $\pm$ 0.06	0.28 $\pm$ 0.05	0.13 $\pm$ 0.03	0.04 $\pm$ 0.02
LastFM	***BlurMe***	*None*	*No removal*	0.66 $\pm$ 0.06	0.57 $\pm$ 0.07	0.35 $\pm$ 0.07	0.16 $\pm$ 0.05	0.61 $\pm$ 0.07	0.45 $\pm$ 0.08	0.15 $\pm$ 0.05	0.04 $\pm$ 0.02
			Random	0.65 $\pm$ 0.06	0.55 $\pm$ 0.07	0.27 $\pm$ 0.07	0.03 $\pm$ 0.02	0.60 $\pm$ 0.07	0.43 $\pm$ 0.08	0.08 $\pm$ 0.04	0.006 $\pm$ 0.002
			Greedy	0.52 $\pm$ 0.07	0.39 $\pm$ 0.07	0.17 $\pm$ 0.05	0.05 $\pm$ 0.02	0.39 $\pm$ 0.07	0.21 $\pm$ 0.05	0.03 $\pm$ 0.02	0.007 $\pm$ 0.003

performs the other approaches of BlurMe with no removal and BlurMe with random removal. BlurMe with greedy removal requires less obfuscation (fewer extra items) to bring the performance of the classifier close to 0.5 AUC. This is due to the fact that BlurMe with greedy removal uses our proposed removal strategy which removes items in the order of their gender indicativeness. Also, we observe that BlurMe with no removal and BlurMe with random removal perform similarly and both require heavier obfuscation, and, as such, can be considered less effective than BlurMe with greedy removal. For this reason, next in Table 2.6, we omit PerBlur with random removal and we continue with PerBlur with no removal and PerBlur with greedy removal.

Table 2.6 shows the gender inference results on PerBlur data for the case of *no removal* and the case of *greedy removal* with different personalization cutoffs. The "personalization" column gives the length at which $Personalized_L^u$ is truncated (see Line 7 to Line 14 of Algorithm 1). We recall that the personalization proposed by PerBlur attempts to create a personalized list of indicative items that are close to the user preferences. In Table 2.6, we see that PerBlur with *no removal* succeeds to lower the gender inference score but with more obfuscation (more extra items). We note that we tested the case in which there is no threshold for the personalization (we call it "All items"). We do not include this results in the table, but instead mention that we found that the inference of the classifier is at the same level as it is for the original data. This is to be expected because without the threshold there is no influence from the indicative item list.

We observe in Table 2.6 that PerBlur with greedy removal outperforms the other variants of PerBlur with no removal, since it can bring the performance of the classifier close to 0.5 AUC with less obfuscation (fewer extra items). This demonstrates the importance of using greedy removal strategy which removes items in the order of their gender indicativeness. We consider gender obfuscation to be successful at levels of obfuscation

Table 2.6: Gender inference results measured in terms of AUC on **PerBlur** (No removal and greedy removal), for ML1M, Flixster, and LastFM data sets. The $\pm$ are standard deviations of the results over ten folds. We note that for rating data, PerBlur with average ratings has quite similar results to PerBlur with predicted ratings. For this reason, we only report results of PerBlur with predicted ratings.

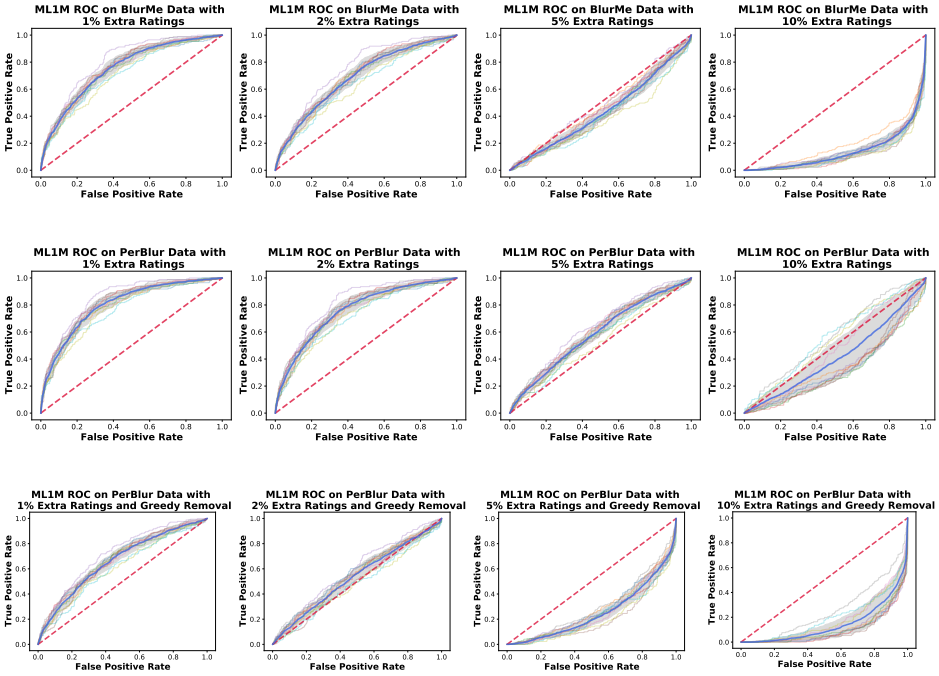
Gender Inference		Obfuscation Strategies		**Logistic Regression**				**SVM**			
				Extra Items				*Extra Items*			
Data Sets		**Personalization**	**Removal**	***1%***	***2%***	***5%***	***10%***	***1%***	***2%***	***5%***	***10%***
ML1M	***Standard PerBlur***	*Top-50 Items*	*No Remval*	0.80 $\pm$ 0.02	0.76 $\pm$ 0.03	0.59 $\pm$ 0.03	0.43 $\pm$ 0.09	0.78 $\pm$ 0.03	0.73 $\pm$ 0.03	0.52 $\pm$ 0.02	0.34 $\pm$ 0.12
		Top-100 Items		0.81 $\pm$ 0.02	0.78 $\pm$ 0.03	0.64 $\pm$ 0.04	0.39 $\pm$ 0.03	0.79 $\pm$ 0.03	0.75 $\pm$ 0.03	0.55 $\pm$ 0.04	0.28 $\pm$ 0.03
	PerBlur w/ removal	*Top-50 Items*	*Greedy*	0.66 $\pm$ 0.03	0.53 $\pm$ 0.03	0.26 $\pm$ 0.03	0.14 $\pm$ 0.05	0.61 $\pm$ 0.03	0.46 $\pm$ 0.03	0.17 $\pm$ 0.02	0.09 $\pm$ 0.05
		Top-100 Items		0.68 $\pm$ 0.03	0.56 $\pm$ 0.04	0.26 $\pm$ 0.04	0.10 $\pm$ 0.02	0.63 $\pm$ 0.03	0.48 $\pm$ 0.03	0.17 $\pm$ 0.03	0.07 $\pm$ 0.01
Flixster	***Standard PerBlur***	*Top-50 Items*	*No removal*	0.78 $\pm$ 0.04	0.75 $\pm$ 0.04	0.67 $\pm$ 0.04	0.57 $\pm$ 0.06	0.73 $\pm$ 0.04	0.69 $\pm$ 0.04	0.57 $\pm$ 0.04	0.45 $\pm$ 0.08
		Top-100 Items		0.79 $\pm$ 0.04	0.77 $\pm$ 0.04	0.69 $\pm$ 0.04	0.55 $\pm$ 0.05	0.75 $\pm$ 0.04	0.72 $\pm$ 0.04	0.59 $\pm$ 0.05	0.41 $\pm$ 0.04
	PerBlur w/ removal	*Top-50 Items*	*Greedy*	0.56 $\pm$ 0.05	0.48 $\pm$ 0.04	0.27 $\pm$ 0.03	0.13 $\pm$ 0.02	0.56 $\pm$ 0.05	0.42 $\pm$ 0.04	0.22 $\pm$ 0.03	0.10 $\pm$ 0.02
		Top-100 Items		0.57 $\pm$ 0.05	0.43 $\pm$ 0.05	0.19 $\pm$ 0.04	0.08 $\pm$ 0.02	0.47 $\pm$ 0.05	0.37 $\pm$ 0.06	0.14 $\pm$ 0.04	0.06 $\pm$ 0.02
LastFM	***Standard PerBlur***	*Top-50 Items*	*No removal*	0.70 $\pm$ 0.06	0.63 $\pm$ 0.07	0.49 $\pm$ 0.08	0.42 $\pm$ 0.14	0.68 $\pm$ 0.06	0.56 $\pm$ 0.07	0.34 $\pm$ 0.09	0.28 $\pm$ 0.15
		Top-100 Items		0.71 $\pm$ 0.06	0.64 $\pm$ 0.07	0.44 $\pm$ 0.06	0.28 $\pm$ 0.09	0.69 $\pm$ 0.06	0.58 $\pm$ 0.07	0.27 $\pm$ 0.05	0.13 $\pm$ 0.09
	PerBlur w/ removal	*Top-50 Items*	*Greedy*	0.53 $\pm$ 0.06	0.39 $\pm$ 0.06	0.21 $\pm$ 0.06	0.17 $\pm$ 0.09	0.39 $\pm$ 0.06	0.21 $\pm$ 0.04	0.07 $\pm$ 0.03	0.06 $\pm$ 0.06
		Top-100 Items		0.54 $\pm$ 0.07	0.36 $\pm$ 0.07	0.12 $\pm$ 0.04	0.06 $\pm$ 0.04	0.42 $\pm$ 0.06	0.18 $\pm$ 0.03	0.02 $\pm$ 0.01	0.02 $\pm$ 0.02

at which AUC is close to 0.5. As we previously mentioned, levels of classification performance lower than 0.5 actually reveal the gender because the point reliably in the opposite direction. Among the personalization levels, Top-50, Top-100 from $Personalized_L^u$, Top-50 items performs consistently well, and we adopt this setting for the PerBlur experiments in the rest of the chapter.

It is important to remember that the ability of gender obfuscation to block the SVM classifier seen in Table 2.5 and Table 2.6 is a demonstration of the transferrability of our approach. Recall from Section 2.3 that the indicative items lists are chosen using logistic regression. It makes sense, then, that adding these items in order to obfuscate data would be able to prevent the classifier from making accurate predictions. The SVM results assure us that the items chosen using logistic regression actually have a general blocking power, since using these items to obfuscate data is also able to block the ability of the classifier to make predictions.

A different view on the gender inference performance is presented in the ROC curves in Fig. 2.2. Here, we show ML1M, and leave out the other data sets since the pattern is similar. These curves dramatically show the level of obfuscation (extra ratings or interactions) at which the performance of the classifier collapses (i.e., the performance approaches the diagonal). On the basis of these curves, we choose the levels of obfuscation for each obfuscation approach that we will investigate for each data set in the remaining of the chapter. These settings constitute a "rough and ready" operating point

Figure 2.2: ROC AUC of a logistic regression classifier on BlurMe (with no removal), PerBlur (with no removal) and PerBlur (with greedy removal) for different degree of obfuscation (1%, 2%, 5% and 10%) for **ML1M** Data. PerBlur data is created with Top-50 personalized list of indicative items and using predicted ratings. We observe that BlurMe and PerBlur with *no removal* require 5% extra items to perform like a random classifier. PerBlur with *greedy removal* requires only 2% of extra items.



at which we know that the gender prediction performance has collapsed. Specifically, for the ML1M data set, we add 5% extra ratings to BlurMe with *no removal* and Standard PerBlur data, and we add 2% extra ratings for PerBlur with *greedy removal* (see Fig. 2.2). For the Flixster data set, we add 2% extra ratings to BlurMe with *no removal*, 10% to Standard PerBlur data, and we add 2% extra ratings to PerBlur with *greedy removal*. For the LastFM data set, we add 2% extra items to BlurMe with *no removal*, and 5% extra items to Standard PerBlur, and we add 1% extra items to PerBlur with *greedy removal*.

2.6. RECOMMENDATION PERFORMANCE

Now that we have established the effectiveness of PerBlur in blocking gender inference, we turn to the evaluation of its ability to maintain recommendation prediction performance.

2.6.1. EVALUATION PROCEDURE

In order to evaluate obfuscated data, it is necessary to have an evaluation procedure that creates a fair environment to compare top-N recommendation performance between different obfuscation techniques. Because obfuscation adds and subtracts ratings (or

interactions), designing a procedure is non-trivial. Unless specific attention is paid to how training and test splits are created, the addition and subtraction of ratings (or interactions) to the user profiles will lead to the test set being different for the different versions of the data that are being compared with each other. The result would be that the test conditions are no longer directly comparable. For example, a particular condition might add easy-to-predict and remove difficult-to-predict ratings, meaning that the prediction score no longer reflects that performance of the recommender algorithm.

We introduce a new evaluation procedure for obfuscated data that ensures that different conditions are comparable. Our procedure works as follows. We randomly sample 80% of each user profile for the training set and we keep the remaining 20% for the test set. The choice of static splitting plays a key role in preventing obfuscation from adding items into the test set. Obfuscation is applied only to the training set. The effect is that the test set remains connected to users in the training set, and will remain the same across all of the conditions.

For Top-N recommendation, the procedure needs to address an additional challenge. Specifically, we would like to be able to use the *1+random* protocol [129], [130] and still maintain a fair comparison across conditions. Under this protocol, a test item is added to a set of random candidate items (here, 1000 items) that are drawn from a set of possible candidate items. In order to maintain fairness in our evaluation procedure, we must look not only at the test set, but also at the set of possible candidate items from which the random items are drawn. The core of the challenge is the following. If *1+random* makes use of a candidate set drawn from the training data, that candidate set will change from condition to condition, since each type of data obfuscation makes different additions (and subtractions) to the data. When obfuscation adds and deletes items, it will impact the candidate set. Specifically, it will change the set of items that compete with the relevant item for each user across conditions. This issue does not occur with data that is not obfuscated.

To address this problem, we adapt *1+random* evaluation for use with obfuscated data. For each user, we define a candidate set of $C = 1000$ items. In Fig. 2.3, we describe our process of generating new candidate items. These items must be selected among items that are *not* rated (= ?) by the user. To ensure that these items are comparable across conditions, we create a large set of possible candidate items for each user by intersecting the items that are *not* rated by the user in the original training set as well as *not* rated by the user in all the obfuscated training sets. We then draw C random items to create the candidate set for each user from this set of possible candidate items. Each relevant item in the set of a user's test items is injected in turn into the user's candidate set, and then recommendation is performed and the ranking metric is calculated. We adapt this evaluation procedure for the comparison of recommendation performance carried out in the next section.

Note that this procedure requires experiments to be planned carefully in advance. Building the candidate set requires an intersection involving data from all conditions that are being compared. It is not possible to compare two types of obfuscated data, and then add a third type later because the candidate set must necessarily change.

It is important to understand why building the candidate set from the test items of the other users is not a viable solution. For this methodology [129], the list includes for

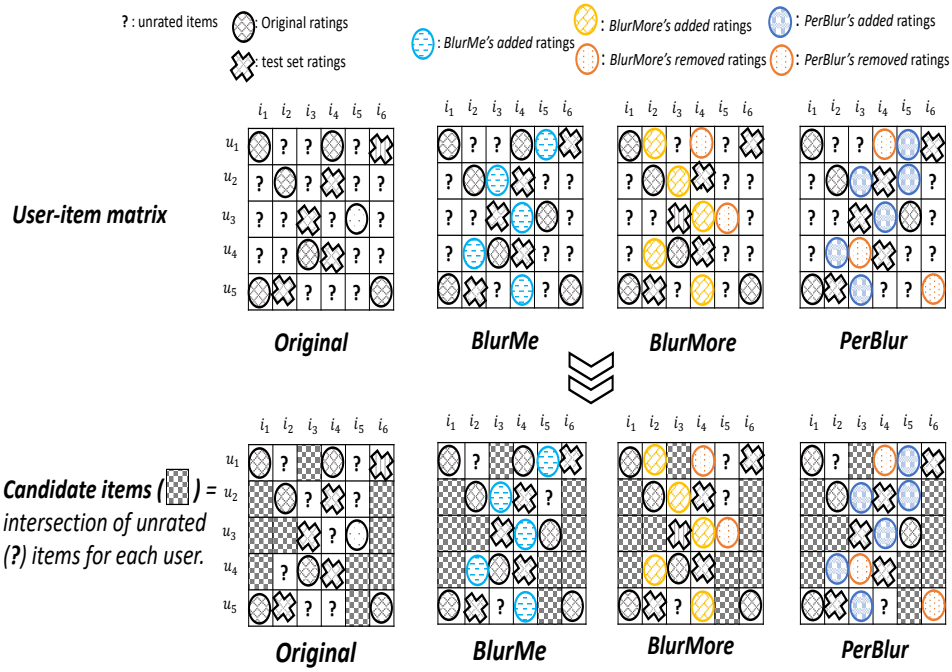


Figure 2.3: Generation of possible candidate items: the intersection of unrated items from the original training set, original test set and the training sets of different obfuscation conditions.

all users, all the items having a test rating (interaction) by some user and no training rating (interaction) by the target user. For this reason: when 1+random uses candidates drawn from the test items of other users, it has to exclude items that are in the training set of the target user's profile. The training part of the profile is exactly the part that changes from one type of obfuscation to the other. Again, we see that the candidate list will change across comparative conditions. For this reason, we build the candidate list for a given target user using items that are in the training data of all of the obfuscated data sets being compared, but are not rated by (or interacted with) the target user and are not in the test set of the target user.

It is very important to remember that only the scores of conditions that contribute to building the candidate sets can be directly compared. In order to understand this point in more detail, consider the impact that different types of obfuscation have on the candidate sets. Recall that the items added by obfuscation to the training data will not occur in the users' individual candidate sets. Consequently, if highly personalized obfuscation approaches are compared, then users' candidate sets may contain less personalized items. Such candidate sets could offer less competition with the relevant item that is being tested in 1+random evaluation, leading to higher absolute scores. The opposite could be true if less personalized obfuscation is used. In short, scores can only be compared relatively within a set of conditions that all contributed to the candidate sets.

2.6.2. COMPARING RECOMMENDATION PERFORMANCE

In this section, we compare recommendation performance in order to measure the extent to which the accuracy of a recommender is impacted when it is trained on obfuscated data. We use the evaluation procedure just described. The training/test split is kept constant. In order to understand the range of variability, we repeat the evaluation five times. Each repetition involves the selection of a new candidate set for each user. We report the average and standard deviation of the repetitions. Recall that we are testing obfuscation levels that we have previously determined to lead to collapse of the predictive ability of the gender classifier.

First, we look at the results of the ALS algorithm, which takes rating data as input and gives rating predictions as output. Remember that our main focus is Top-N recommendation, but we test rating prediction because that was the focus of previous work, most importantly [22]. The results are shown in Table 2.7. We report rating prediction with MAE. We also rank items by their predicted ratings, which allows us to get a top-N view of ALS and ranking prediction. The performance of this ranking is reported as HR@10.

Table 2.7: Rating prediction results measured in terms of MAE and HR@10 using **ALS** on Original, BlurMe, and standard PerBlur Data (personalization with Top-50 indicative items) for ML1M and Flixster data sets. The scores report the average over five repetitions of the evaluation. The standard deviation for HR@10 is around 0.0001.

ALS		***Obfuscation Strategies***		***MAE***	***HR@10***
Data Sets		**Obfuscation Level**	**Personalizaton**		
ML1M	***Original***	*0%*	*None*	0.7534	0.0125
	BlurMe	*5%*	*None*	0.7477	0.0126
	Standard PerBlur	*5%*	*Personalized w/ Average Ratings*	0.7485	0.0129
		5%	*Personalized w/ Predicted Ratings*	0.7497	0.0125
Flixster	***Original***	*0%*	*None*	0.7584	0.0071
	BlurMe	*5%*	*None*	0.7400	0.0070
	Standard PerBlur	*10%*	*Personalized w/ Average Ratings*	0.7415	0.0066
		10%	*Personalized w/ Predicted Ratings*	0.7410	0.0069

The main insight gained from Table 2.7 is that it is possible to train a recommender on obfuscated data, and still maintain a comparable performance level as is achieved when the recommender is trained on the original, unobfuscated data. This conclusion is consistent with the BlurMe [22] rating prediction experiments. Recall however, that in contrast to [22], we obfuscate the full data set, rather than just 10%. We find that the performance in the case of obfuscated data can actually exceed the performance in the case of the original data (i.e., MAE falls below the level of Original). This can be explained by the fact that in some cases, making the profile larger gives a boost. We see that in terms of MAE, PerBlur and BlurMe achieve approximately the same performance level, and both

outperform the original. In terms of HR@10 it remains close. In real-world application scenarios, we are not particularly interested in rating prediction, nor would we choose to carry out top-N recommendation by ranking on the basis of predicted ratings. However, these experiments serve to give insight into how gender obfuscation works, and link our analysis of PerBlur to the related work on BlurMe, which studied rating prediction.

Next, we look at BPRMF algorithm, which takes implicit data as input and gives a ranked list of items as output. Recall that ML1M and Flixster are binarized via thresholding and LastFM is interaction data, which is originally implicit. The results are shown in Table 2.8. We report TopN recommendation results measured with Top10.nDCG and HR@10.

Table 2.8: Ranking prediction results measured in terms of Top10.nDCG (Δ wrt original Top10.nDCG) and HR@10 (Δ wrt original HR@10) using **BPRMF** on Original, BlurMe (no removal), and Standard PerBlur Data (personalization with Top-50 indicative items and no removal). The scores report the average over five repetitions of the evaluation. The standard deviation of Top10.nDCG and HR@10 is around 0.001 on ML1M and Flixster data sets. The standard deviation of Top10.nDCG and HR@10 is around 0.005 on LastFM data.

BPRMF		***Obfuscation Strategies***		***nDCG (Δ wrt Original)***	***HR@10 (Δ wrt Original)***
Data Sets		**Obfuscation Level**	**Personalization**		
ML1M	***Original***	0%	*None*	0.1634	0.1712
	BlurMe	5%	*None*	0.1536 (-0.0098)	0.1633 (-0.0080)
	Standard PerBlur	5%	*Personalized w/ Average Ratings*	0.1603 (-0.0031)	0.1675 (-0.0037)
		5%	*Personalized w/ Predicted Ratings*	**0.1637** (+0.0003)	**0.1704** (-0.0009)
Flixster	***Original***	0%	*None*	0.1139	0.0628
	BlurMe	5%	None	0.1066 (-0.0073)	**0.0605** (-0.0023)
	Standard PerBlur	10%	*Personalized w/ Average Ratings*	0.1028 (-0.0112)	0.0602 (-0.0026)
		10%	*Personalized w/ Predicted Ratings*	**0.1099** (-0.0041)	0.0595 (-0.0033)
LastFM	***Original***	0%	*None*	0.0782	0.0603
	BlurMe	2%	*None*	**0.0839** (+0.0056)	**0.0722** (+0.0119)
	Standard PerBlur	5%	*Personalized w/ Interaction*	0.0752 (-0.0030)	0.0690 (+0.0087)

We see in Table 2.8 that when data obfuscated with PerBlur is used, the recommendation performance comes very close to what is achieved on the original data for both Top10.nDCG and HR@10. This observation stands in contrast to the conventional wisdom that privacy comes at the price of decreased recommendation performance. Recall that the obfuscation levels used here are chosen because they collapse the AUC curve to a random classifier. In other words, obfuscation defeats the classifier with a very small decrease in recommendation performance, if there is a decrease at all. Overall, PerBlur approaches the original performance more closely and more consistently than BlurMe.

Further in Table 2.8 we see that PerBlur that uses predicted ratings outperforms PerBlur

that uses average ratings. This point is interesting since predicted ratings were not found to be particularly helpful by [22].

2.7. MAINTAINING FAIRNESS

Next, we move to investigate the impact of gender obfuscation on fairness. Here, we are concerned about the extent to which recommender algorithms trained on obfuscated data are able to maintain fairness for both genders. We investigate fairness by comparing the ranking prediction results calculated separately for males and females. These results are presented in Table 2.9.

The first point to notice in Table 2.9 is that all of our data sets have a gap between the performance for the two genders (see the first line of each section of the table, reporting results on the original data). For ML1M and Flixster, the systems perform better for males and for LastFM the systems perform better for females. A gender performance gap has been observed in many systems in the literature, e.g., [103].

Table 2.9: Ranking prediction results measured in terms of Top10.nDCG and HR@10 using **BPRMF** for female and male users. The standard deviation of Top10.nDCG and HR@10 for female and male users is around 0.001 on ML1M and Flixster data sets. The standard deviation of Top10.nDCG and HR@10 for female and male users is around 0.005 on LastFM data. $|\Delta_{nDCG_F} - \Delta_{nDCG_M}|$ measures the Δ difference with respect to the original Top10.nDCG for both genders. $|\Delta_{HR_F} - \Delta_{HR_M}|$ measures the Δ difference with respect to the original HR for both genders. The scores report the average over five repetitions of the evaluation.

BPRMF		***Obfuscation Strategies***		$nDCG_F$	$nDCG_M$	$\lvert\Delta_{nDCG_F} - \Delta_{nDCG_M}\rvert$	HR_F	HR_M	$\lvert\Delta_{HR_F} - \Delta_{HR_M}\rvert$
	Data Sets	**Obfuscation Level**	**Personalizaton**	*(Δ w/ Orig)*	(Δ w/ Orig)		***(Δ w/ Orig)***	***(Δ w/ Orig)***	
ML1M	***Original***	0%	*None*	0.1478	0.1695	0.0000	0.1575	0.1768	0.0000
	BlurMe	5%	*None*	0.1338 (-0.0140)	0.1614 (-0.0082)	0.0058	0.1474 (-0.0101)	0.1697 (-0.0071)	0.0029
	Standard PerBlur	5%	*Personalized w/ Average Ratings*	0.1398 (-0.0080)	0.1684 (-0.0011)	0.0069	0.1489 (-0.0087)	0.1751 (-0.0017)	0.0069
		5%	*Personalized w/ Predicted Ratings*	**0.1435** (-0.0043)	**0.1716** (+0.0021)	0.0064	**0.1518** (-0.0057)	**0.1779** (+0.0011)	0.0068
Flixster	***Original***	0%	*None*	0.1115	0.1179	0.0000	0.0605	0.0671	0.0000
	BlurMe	5%	*None*	0.1060 (-0.0055)	0.1077 (-0.0103)	0.0048	0.0591 (-0.0014)	0.0630 (-0.0041)	0.0027
	Standard PerBlur	10%	*Personalized w/ Average Ratings*	0.1042 (-0.0074)	0.1004 (-0.0175)	0.0101	**0.0590** (-0.0015)	0.0624 (-0.0047)	0.0032
		10%	*Personalized w/ Predicted Ratings*	**0.1079** (-0.0037)	**0.1132** (-0.0047)	0.0010	0.0576 (-0.0029)	**0.0631** (-0.0040)	0.0011
LastFM	***Original***	0%	*None*	0.1052	0.0570	0.0000	0.0805	0.0445	0.0000
	BlurMe	2%	None	**0.1092** (+0.0040)	**0.0639** (+0.0069)	0.0029	0.0836 (+0.0031)	**0.0633** (+0.0188)	0.0157
	Standard PerBlur	5%	*Personalized w/ Interacted Data*	0.1033 (-0.0019)	0.0532 (-0.0038)	0.0019	**0.0873** (+0.0068)	0.0547 (+0.0102)	0.0034

We have previously seen that obfuscation sometimes improves recommendation, but often causes a small drop. Here, we see that the drop is not evenly distributed over both genders. Rather, one gender drops further than the other. The implication is that when obfuscating it is necessary to check that the recommender system performance is not impacted asymmetrically between the genders. This observation is new, and has not been previously reported in the literature.

In the columns $|\Delta_{nDCG_F} - \Delta_{nDCG_M}|$ and $|\Delta_{HR_F} - \Delta_{HR_M}|$ in Table 2.9 we report the difference between the drop (or gain) experienced by both genders. It can be seen that this value is the lowest for PerBlur with predicted ratings. The exception is ML1M where

the value for BlurMe is lowest. In this case, PerBlur with predicting ratings outperforms BlurMe for both genders, so the gap is less worrisome. In general, PerBlur appears somewhat better in preventing obfuscation from widening++ the gender performance gap. We interpret this finding as reflecting the benefit of attempting to avoid obfuscating with "noise", but instead keep obfuscation as close as possible to what users might have done themselves.

Finally we note that here again we see that PerBlur with predicted ratings is superior to PerBlur with average ratings. In the remainder of the chapter, we examine PerBlur using predicted ratings.

2.8. ACHIEVING DIVERSE RESULTS

In this section, we look at the impact of obfuscation on diversity. We first need an overview of the different variants of obfuscation we will investigate. We start by looking at the conditions for which we report Top-N recommendation performance. For this, the relevant performance levels were already reported in Table 2.8. Then, we will look at obfuscation with removal. For this purpose the Top-N recommendation performance is provided in Table 2.10. This table includes results for both random and greedy removal. We include this table here because obfuscation with removal is not discussed in detail in Section 2.6.2 due to the fact that our experiments showed that it did not have a consistent influence on recommendation performance or fairness. However, we study removal now because of its potential for enhancing diversity. Recall that the difference between the two removal strategies is that the random removal strategy removes items randomly from individual user profile and the greedy removal strategy removes items in the order of their gender indicativeness (in L_m and L_f) from individual user profile. In Table 2.10, we see that greedy removal and random removal are largely comparable. In our analysis of diversity we will argue that the choice should be made by taking diversity, and not just recommendation accuracy, into consideration.

Table 2.10: Ranking prediction results measured in terms of Top10.nDCG and HR@10 using **BPRMF** on Original, BlurMe, and PerBlur with removal Data (personalization with Top-50 indicative items and removal strategy). The scores report the average over five repetitions of the evaluation. The standard deviation of Top10.nDCG and HR@10 is around 0.001 on ML1M and Flixster data sets, and 0.003 on LastFM data.

BPRMF		***Obfuscation Strategies***		***nDCG***	***HR@10***
Data Sets		**Personalizaton**	**Removal**		
ML1M	***Original***	*None*	*None*	0.1632	0.1720
	PerBlur w/ Removal	*Personalized w/ Average Ratings*	*Random*	0.1591	0.1682
			Greedy	0.1545	0.1615
		Personalized w/ Predicted Ratings	*Random*	**0.1593**	**0.1678**
			Greedy	0.1534	0.1606
Flixster	***Original***	*None*	*None*	0.1159	0.0610
	PerBlur w/ Removal	*Personalized w/ Average Ratings*	*Random*	0.1092	0.0600
			Greedy	0.1056	0.0584
		Personalized w/ Predicted Ratings	*Random*	**0.1104**	**0.0607**
			Greedy	0.1073	0.0590
LastFM	***Original***	*None*	*None*	0.0882	0.0622
	PerBlur w/ Removal	*Personalized w/ Interacted Data*	*Random*	0.0764	0.0592
			Greedy	**0.0833**	**0.0576**

Now that we have a complete view of recommendation performance for all the rele-

Table 2.11: Diversity for Standard PerBlur: The proportion of correctly recommended items that are stereotypical for gender (female and male) using **BPRMF**. Three different cutoff levels (10, 20, 50) are used to define gender-stereotypical items. Original Data and Standard PerBlur Data (personalization with Top-50 indicative items). The scores report the average over five repetitions of the evaluation. The standard deviation is around: 0.0002 on ML1M data, 0.0005 on Flixster data, and 0.0005 on LastFM data.

BPRMF		***Obfuscation Strategies***		*Stereotypical Gender Items*					
Data Sets		**Level**	**Personalization**	**top10F**	**top10M**	**top20F**	**top20M**	**top50F**	**top50M**
ML1M	***Original***	*0%*	*None*	0.0021	0.0044	0.0040	0.0068	0.0083	0.0127
	Standard PerBlur	*5%*	*Personalized w/ Predicted Ratings*	0.0020	0.0046	0.0036	0.0070	0.0077	0.0129
Flixster	***Original***	*0%*	*None*	0.0056	0.0090	0.0114	0.0150	0.0244	0.0266
	Standard PerBlur	*10%*	*Personalized w/ Predicted Ratings*	0.0038	0.0086	0.0083	0.0142	0.0197	0.0251
LastFM	***Original***	*0%*	*None*	0.001	0.0000	0.001	0.0000	0.0026	0.0002
	Standard PerBlur	*5%*	*Personalized w/ Interacted Data*	0.000	0.0000	0.000	0.0000	0.0003	0.0000

vant variants of obfuscation, we dive into the impact of PerBlur on diversity. Remember that we study diversity by looking at the ability of PerBlur data to steer recommender systems away from providing gender-stereotypical recommendations. Recall also that we define a gender-stereotypical recommendation as an item that is highly typical for a particular gender. Our assumption is that users will appreciate a less stereotyped recommender, i.e., that women will appreciate when recommendations do not focus on stereotypical female items such as 'chick flicks'. We are not looking to eliminate gender stereotypical items from the recommendation lists, but rather to control them.

For our analysis, we assume gender-stereotypical items to be items that are specific to a user's gender. We make use of the lists of gender-indicative items, L_f and L_m, that we use for the gender obfuscation algorithms. Because these lists were derived before the training/test split, test items of users occur in these lists. Refer back to Fig. 2.3 to understand the way in which the training and test set are ensured to be disjoint. We test three different cutoffs for defining a list of gender-specific items: top10, top20, and top50 most specific items. Recall that PerBlur removes gender specific items from the training data. Note, however, that this does not impact the test data, which is a constant item set over all data sets tested (Fig 2.3).

In Table 2.11 and Table 2.12, we report the proportion of correctly recommended test-items that are gender-stereotypical. For PerBlur, these tables report the PerBlur variant that uses predicted ratings so as not to crowd the table. We choose this variant because it generally achieves better performance. The proportions in these tables are small because only a small number of top10, top20 or top50 items are in the ground truth. However, the relative difference between these proportions demonstrates the effect of PerBlur.

Table 2.11 corresponds to the recommender performance in Table 2.8. In Table 2.11, we see that PerBlur seems to lower the Top-N gender-stereotypical items that are recommended to both male and female users with respect to the original data.

Table 2.12 corresponds to the recommender performance in Table 2.10. Note that the performance on the original data is different between Table 2.8 and Table 2.10. This difference arises because the conditions in these tables were run as two separate condi-

Table 2.12: Diversity for PerBlur with removal: The proportion of correctly recommended items that are stereotypical for gender (female and male) using **BPRMF**. Three different cutoff levels (10, 20, 50) are used to define gender-stereotypical items. Original Data and PerBlur Data (personalization with Top-50 indicative items using predicted ratings, and with **greedy removal**). The scores report the average over five repetitions of the evaluation. The standard deviation is around: 0.0002 on ML1M data, 0.0005 on Flixster data, and 0.0005 on LastFM data.

BPRMF	***Obfuscation Strategies***		*Stereotypical Gender Items*					
Data Sets	**Personalization**	**Removal**	**top10F**	**top10M**	**top20F**	**top20M**	**top50F**	**top50M**
ML1M	***Original***	*None*	*None*	0.0020	0.0045	0.0038	0.0069	0.0082	0.0128
	PerBlur	*Personalized w/*	*Random*	0.0017	0.0045	0.0033	0.0070	0.0075	0.0127
	w/ Removal	*Predicted Ratings*	*Greedy*	**0.0003**	**0.0005**	**0.0014**	**0.0020**	**0.0051**	**0.0073**
Flixster	***Original***	*None*	*None*	0.0058	0.0084	0.0115	0.0147	0.0225	0.0255
	PerBlur	*Personalized w/*	*Random*	0.0048	0.0087	0.0097	0.0149	0.0219	0.0265
	w/ Removal	*Predicted Ratings*	*Greedy*	**0.0006**	**0.0018**	**0.0035**	**0.0068**	**0.0149**	**0.0169**
LastFM	***Original***	*None*	*None*	0.0013	0.0010	0.0013	0.0010	0.0026	0.0027
	PerBlur	*Personalized w/*	*Random*	0.0013	0.0010	0.0013	0.0010	0.0013	0.0020
	w/ Removal	*Interacted Data*	*Greedy*	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0008**	**0.0012**

tion sets, which means that their candidate sets are not comparable, as was described in Section 2.6.1. In Table 2.12, we see that PerBlur with greedy removal is highly effective in lowering the proportion of Top-N gender-stereotypical items. Random removal has no apparent impact on diversity. This effect can be attributed to the fact that greedy removal uses information about gender specificity and can guide the recommender away from gender-typical items. In sum, these results demonstrate the potential of using obfuscation to improve diversity at the same time as it is protecting users' privacy.

2.9. CONCLUSION AND OUTLOOK

In this section, we summarize the main findings of our chapter and also provide an outlook onto future working.

2.9.1. SUMMARY

We have introduced PerBlur, a new gender obfuscation approach for recommender system data. PerBlur extends the state of the art with its use of personalization and also greedy item removal.

Main finding The main contribution of the chapter is a demonstration that PerBlur can maintain recommender system performance, and in some cases improve it, while also blocking the inference of gender information.

We have also shown that BlurMe, an approach that does not use personalization, is effective when applied to the entire user-item matrix, which was not previously demonstrated in the literature. The picture that emerges is that PerBlur shows advantages over BlurMe, but that BlurMe is also more effective than is expected.

User-oriented paradigm The PerBlur approach was formulated within a user-oriented paradigm for user profile privacy. This paradigm requires approaches to be *understandable* to users, as well as remaining *unobtrusive* so that they do not get in the user's way.

These are two desirable characteristics informed the design of PerBlur. The paradigm also requires that privacy look at *usefulness* as going beyond accuracy to encompass also fairness and diversity aspects of recommender system data protection.

Fairness and Diversity Our experiments have shown that obfuscation interacts with fairness and diversity. When using obfuscation it is important to check that different user groups are impacted in the same way. In our experiments, we saw that PerBlur appears to have an advantage over BlurMe in controlling the difference of the impact. We also showed that PerBlur has the potential to improve recommendation diversity by reducing the percentage of gender-stereotypical items that are recommended.

Evaluation methodology We have pointed out that fairness of experimental analysis when testing obfuscated recommender system data is non-trivial. To address this challenge, we have proposed an evaluation procedure for obfuscated recommender system data, and carried out our experiments using this procedure.

2.9.2. FUTURE WORK

The user-oriented paradigm for privacy protection offers a framework in which future work can formulate new approaches to protecting user data. The formulation of PerBlur itself is independent of the specific nature of the data and the attribute being protected. In the future, PerBlur could be applied also to different demographic attributes such as age, occupation, ethnicity, political orientation. Here, we elaborate further on the insights of this chapter that are important for future work.

Obfuscation that promotes fairness and diversity Work focusing on obfuscation of other attributes has been carried out by [63], [64], but not all of these approaches focus on data set obfuscation, and none of them investigate the potential benefits for fairness and diversity. In this respect, our contribution opens an important new vista for future work.

From obfuscation to data synthesis We also mention that PerBlur can be considered to be between obfuscation and data synthesis. Because it imputes items, PerBlur is effectively synthesizing a profile extension for each user. If obfuscation can take the data far enough away from the original user, but still keep it faithful to underlying distributions, it could become an important tool in creating data sets that can be released for the research community to use in the development of new algorithms.

Moving towards more sophisticated threat models Our work is based on the threat model that is defined in Section 2.1.2. This threat model can be refined in the future to match more closely with real-world threats, in particular data breaches. A limitation of our current work is that we test gender classifiers with only one data set. A more sophisticated threat model would assume that different sources and different amounts of labeled data may be at the disposal of the attacker.

We close by summarizing the main insights that we have found important for guiding future work on data obfuscation.

Obfuscation is relatively easy First, obfuscation is a simpler task than one might think. A simple approach, fully understandable to users, works well. An approximate setting of an operating point gives a practically useful approach. Future research should not blindly assume that the problem of gender obfuscation requires iterative optimization approaches. Such approaches are not only difficult for the user to understand, but they are computationally heavy and require recomputation as users continue to rate and interact with items items.

Obfuscation needs not to add noise Second, we should not assume that obfuscation must introduce noise. In this chapter we have shown, that if we keep obfuscation close to user preferences it has the potential to be unobtrusive for the user and also allows us to maintain or even improve upon the performance of the original data.

Obfuscation should go beyond accuracy Third, maintaining recommender system accuracy should not be the sole goal of obfuscation. Instead, fairness must also be maintained. We have seen that obfuscation also opens up an interesting opportunity to improve recommender system diversity.

3

DATA MASKING FOR RECOMMENDER SYSTEMS: PREDICTION PERFORMANCE AND RATING HIDING

This chapter is published as Manel Slokom, Martha Larson, and Alan Hanjalic. Data Masking for Recommender Systems: Prediction Performance and Rating Hiding. Late breaking results, in conjunction with the 13th ACM Conference on Recommender Systems. 2019.

Data science challenges allow companies, and other data holders, to collaborate with the wider research community. In the area of recommender systems, the potential of such challenges to move forward the state of the art is limited due to concerns about releasing user interaction data. This chapter investigates the potential of privacy-preserving data publishing for supporting recommender system challenges. We propose a data masking algorithm, Shuffle-NNN, with two steps: Neighborhood selection and value swapping. Neighborhood selection preserves valuable item similarity information. The data shuffling technique hides (i.e., changes) ratings of users for individual items. Our experimental results demonstrate that the relative performance of algorithms, which is the key property that a data science challenge must measure, is comparable between the original data and the data masked with Shuffle-NNN.

3.1. INTRODUCTION TO DATA MASKING WITH SHUFFLE-NNN

We propose a masking approach, *Shuffling Non-Nearest-Neighbors* (Shuffle-NNN), which changes (i.e., hides) a large proportion of the values of the ratings in the original user-item matrix, $\mathscr{R}$, to create a masked data set, $\mathscr{R}'$. Our work is a step towards masked data that can be publicly released for data science challenges. For this reason, we adopt a relative-rank success criterion: given a set of recommender algorithms to be compared, the relative ranking of the algorithms trained and tested on $\mathscr{R}$ and on $\mathscr{R}'$ must be maintained. Shuffle-NNN is motivated by the observation that privacy-preserving techniques [131] (privacy-preserving data publishing [132], privacy-preserving data mining [133]) have not yet been systematically applied to the data released for recommender system challenges.

Shuffle-NNN generates a masked data by changing a large portion of values of the ratings in a user's profile. We chose a data shuffling technique because of its previous success in masking numerical data in other domains [57]. Shuffle-NNN works as follows. Our overall strategy is to shuffle ratings in a way that maintains key item-item similarities in the data set. First, we determine the item neighborhoods, i.e., the most similar item for every item, and join them, giving us an overall set of critical items. Then, we shuffle ratings for items not in this set (non-critical items, i.e., the rest of the items). Shuffle-NNN has two parameters. We fix the neighbhorhood size $k = 40$, since this value has been shown to be effective in practice[1]. Given k, we set θ (a minimum threshold on item-similarity that must be met for inclusion in a neighborhood) by running some test rankings. Our exploratory experiments showed that a range of θ values can be effective, and that θ can be determined using a subset of the algorithms to be ranked (meaning that it can be set in-house before releasing data).

Recall that the goal of this chapter is not to demonstrate the absolute performance of the algorithms, but rather to evaluate if the relative performance of algorithms is the same on the original and on the masked data. For this purpose, we need a selection of classic recommender algorithms, ranging from baselines that are known not to yield state-of-the-art performance, to current algorithms. We carry out experiments on both ranking prediction, and on classic rating prediction tasks. The algorithms for ranking prediction are: *Most Popular*, *KNN* is user/item-based K-Nearest Neighbor. *(BPRFM)* [134] and *BPRMF*. We followed the hyperparameters tested in [135].The algorithms for rating prediction are: *ItemKNN* [136], *UserKNN*, *SlopeOne* [137],*Co-clustering* [138]. matrix factorization *(MF)*, Biased Matrix Factorization *(BMF)* [139] and *(SVD++)* [140]. We also test baseline algorithms: *Average-Item* and *Average-User* which use the average rating value of a user or item for predictions.

3.2. EXPERIMENTAL FRAMEWORK

The experiments are implemented using WrapRec [141]. We test on two publicly available data sets (cf. Table 3.1). We choose MovieLens 100k [2], since it is well understood, and Goodbooks-10K [3], since it is larger and sparser.

[1] http://www.mymedialite.net/examples/datasets.html
[2] https://grouplens.org/datasets/movielens/
[3] https://www.kaggle.com/philippsp/book-recommender-collaborative-filtering-shiny/data

Table 3.1: Statistics of the data sets used in our experiments and analysis.

data set	#Users	#Items	#Ratings	Range	Av.rating	Density(%)	Variance
MovieLens 100k/ Random empirical /Masked	943/-/-	1682 /-/-	100.000/-/-	[1,5] /-/-	3.529/2.997/3.529	6.30 /-/-	1.1256/ 1.415/ 1.125
GoodBook/ Random empirical /Masked	53424/-/-	10000/-/-	981.756/-/-	[1,5]/-/-	3.8565/ 3.858/3.856	0.18/-/-	0.9839/ 0.9837/0.9838

Table 3.2: The ranking prediction performance on the original data, on the masked data and on the empirical data. For the masked MovieLens data we used $\theta = 0.4$ and for the masked GoodBook data we used $\theta = 0.15$.

		R@5 / R@10 for MovieLens data set		R@5 / R@10 for GoodBook data	
		Original	*Masked*	*Original*	*Masked*
The relative ranking of algorithms	*UserKNN*	0.093/0.177	0.093/0.178	0.434/0.595	0.433/0.594
	BPRMF	0.084/0.165	0.085/0.166	0.398/0.575	0.396/0.571
	BPRFM	0.082/ 0.159	0.082/0.159	0.387/0.567	0.382/0.563
	ItemKNN	0.079/ 0.156	0.079/0.155	0.315/0.452	0.314/0.451
	Popular	0.058/ 0.099	0.064/0.108	0.033/0.053	0.028/0.059

3.3. COMPARATIVE ALGORITHM RANKING

3.3.1. RANKING PREDICTION PERFORMANCE

We train and test the algorithms for ranking prediction both on the original data and and the masked data. Results are given in Table 3.2.

It can be seen that the comparative algorithm ranking is maintained. In other words, for all cases, the best algorithm on the masked data is also the best algorithm on the original data and the worst algorithm on the masked data is also the worst algorithm on the original data. These results demonstrate the success of Shuffle-NNN. In Table 3.2, we report results for specific values of θ. However, relative ranking is actually maintained for a range of values of θ (not shown here). In addition to Recall@5 and Recall@10, we found also that Precision@5/ Precision@10 maintains the same relative ranking of algorithms.

Interestingly, when we tested our algorithms on *empirical data* (which we generate by replacing individual values with values drawn randomly from the global distribution of ratings in the original data), the comparative ranking was also preserved. This means that Shuffle-NNN is sufficient, but not actually necessary in the case of Top-N recommendation. Our conclusion from these results is that (at least for these data sets) the values of the ratings are not important if the goal is a relative ranking among algorithms. What is important is that our masked data sets maintain information on which items were rated.

3.3.2. RATING PREDICTION PERFORMANCE

To dig deeper, we carry out rating prediction experiments. Results are reported in Table 3.3. Here, we observe that the relative ranking is well maintained between the original data and masked data (although not perfectly). Our focus here is on relative performance, but it is interesting to note that the absolute RMSE on the masked data remains within 5% of its value on the original data. In contrast to the ranking-prediction results, we found that the empirical distribution does less well in maintaining the ranking than Shuffle-NNN.

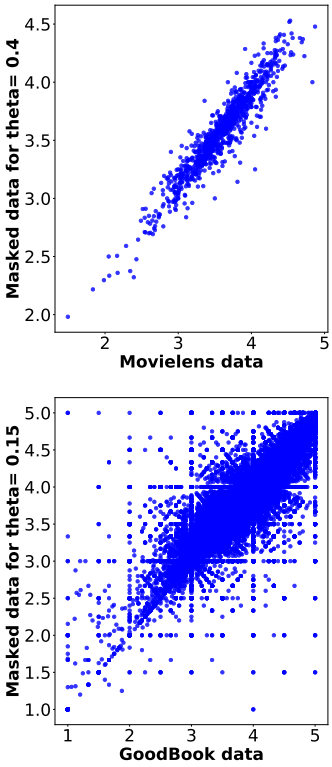


Figure 3.1: Average user ratings original vs. masked data (masked using θ= 0.4 for MovieLens; θ = 0.15 for GoodBook).

Table 3.3: The prediction performance on the original data and on the masked data and on the empirical data. Note: Lower RMSE is better. The colors show the ranking of algorithms. For the masked MovieLens data we used $\theta = 0.4$ and for the masked GoodBook data we used $\theta = 0.15$.

	RMSE for MovieLens data set		**RMSE for GoodBook data set**	
	Original	*Masked*	*Original*	*Masked*
The relative ranking of algorithms	SVD++ (0.902)	SVD++ (0.957)	BMF (0.822)	BMF (0.85)
	BMF (0.911)	BMF (0.962)	SVD++ (0.825)	SVD++ (0.854)
	ItemKNN (0.918)	MF (0.968)	ItemKNN (0.826)	ItemKNN (0.861)
	MF (0.929)	ItemKNN (0.972)	MF (0.856)	MF (0.873)
	UserKNN (0.935)	UserKNN (0.979)	SlopeOne (0.865)	UserKNN (0.894)
	SlopeOne (0.937)	SlopeOne (0.986)	UserKNN (0.868)	SlopeOne (0.903)
	Co-Clustering (0.974)	Co-Clustering (1.026)	Average-User (0.883)	Average-User (0.913)
	Average-Item (1.023)	Average-Item (1.030)	Co-Clustering (0.891)	Co-Clustering (0.924)
	Average-User (1.042)	Average-User (1.081)	Average-Item (0.948)	Average-Item (0.947)

3.4. RATING HIDING

Next, we discuss the ability of Shuffle-NNN to hide ratings. A rating is considered hidden if its value changes between the masked data and the original data [142]. First, we look at the global proportion of ratings hidden in the masked data. We find that at our operating

point, Shuffle-NNN achieved a rating hiding percentage of 0.7 for MovieLens data and 0.68 for the GoodBook data.

Then, we look at the impact of masking at the user level. Figure 3.1 illustrates the relationship between average user ratings before and after masking. Figure 3.2 shows, for different level of hidden ratings, for how many users that level was achieved. The average user rating of the masked data is impacted, but still correlated with the original values. The protection varies per user, but is relatively high.

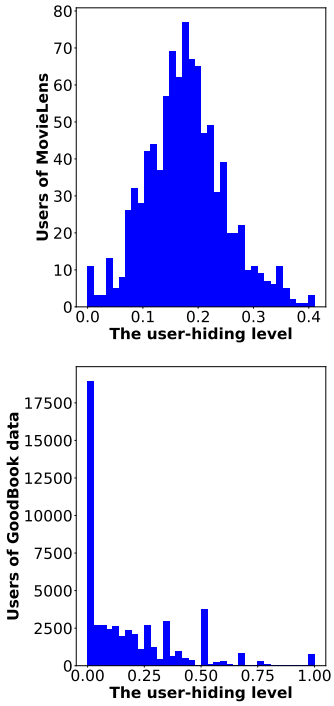


Figure 3.2: Average user ratings original vs. masked data (masked using θ= 0.4 for MovieLens; θ = 0.15 for GoodBook).

3.5. CONCLUSIONS AND OUTLOOK

Our overall conclusion is that data masking has great potential for data science challenges. It is possible to develop a masking approach, such that masked data can be used to train and test algorithms with little impact on the *relative* performance of algorithms. Shuffle-NNN provides valuable evidence about what information can be removed from the user-item matrix and what information should be maintained. When the critical items list is larger than the list of items that is shuffled, it is easier to reconstruct the original values. Future work will investigate the difficulty of reconstructing the original values from the shuffled data, which is an issue important to consider in cases where the critical items are in the majority. We note that even modest levels of rating hiding can

support deniability.

II

ATTACKING OUTPUT DATA

4

When Machine Learning Models Leak: An Exploration of Synthetic Training Data

This chapter is a corrected and updated version of the original paper, which was published as: Manel Slokom, Peter-Paul de Wolf, Martha Larson. When Machine Learning Models Leak: An Exploration of Synthetic Training Data. In: Domingo-Ferrer, J., Laurent, M. (eds) Privacy in Statistical Databases. PSD 2022. Lecture Notes in Computer Science, vol 13463. Springer, Cham.

We investigate an attack on a machine learning model that predicts whether a person or household will relocate in the next two years, i.e., a propensity-to-move classifier. The attack assumes that the attacker can query the model to obtain predictions and that the marginal distributions of the data set on which the model was trained are publicly available. The attack also assumes that the attacker has obtained the values of non-sensitive attributes for a certain number of target individuals. The objective of the attack is to infer the values of sensitive attributes for these target individuals. We explore how replacing the original data with synthetic data when training the model impacts how successfully the attacker can infer sensitive attributes.

4.1. INTRODUCTION

Governmental institutions charged with collecting and disseminating information may use machine learning (ML) models to produce estimates, such as imputing missing values or inferring attributes that cannot be directly observed. When such estimates are published, it is also useful to make the machine learning model itself publicly available, so that researchers using the estimates can evaluate it closely, or even produce their own estimates. Moreover, society also asks for more insight into the models that are used, e.g., to address possible discrimination caused by decisions based on machine learning models.

Unfortunately, machine learning models can be attacked in a way that allows an attacker to recover information about the data set that they were trained on [143]. For this reason, making machine learning models available can lead to a risk that information from the training set is leaked. In this paper, we carry out a case study of *model inversion attribute inference attacks* on a machine learning classifier to better understand the nature of the risk. Model inversion attribute inference attacks aim to reconstruct the data a model is trained on or expose sensitive information inherent in the data [50], [54]. Conventionally, they only seek to infer sensitive attributes of individuals whose data are included in the training set (Inclusive individuals). Here, we go beyond this conventional perspective to investigate the extent to which the availability of the machine learning model and the marginal distributions of the data it was trained on can support inferring sensitive attributes of individuals who are not in the training set (Exclusive individuals).

The attack scenario that we study assumes that the classifier has been made accessible and can be queried with arbitrary input an unlimited number of times, and also that the marginal distributions of the data set the model was trained on have been released. The attacker has a set of non-sensitive attributes of the target individuals including the correct value for the propensity to move attribute for "Inclusive individuals" in the training data or "Exclusive individuals" not in the training data. The attacker wishes to learn sensitive attributes for a group of victims, i.e., target individuals. The classifier that we attack in our study predicts individuals' tendency to move or relocate, i.e., whether an individual or household has the desires, expectations, or plans to move to another dwelling [144] within the next two years. For this reason, it is called a *propensity-to-move* classifier. Our investigation into propensity to move builds upon the work of [61], which studies the possibility of replacing a survey question about moving desires with a model-based prediction using a machine learning classifier.

Our experimental investigation first confirms that a machine learning classifier is able to predict the propensity to move with an accuracy comparable to that achieved by [61]. In contrast to [61], we report results for previously "unseen" individuals (Exclusive individuals) separately from results on individuals who appeared in the training data, which was collected two years before the test data (Inclusive individuals). We then attack this classifier and demonstrate that an attacker can learn sensitive attributes both for Inclusive individuals in the training data as well as for Exclusive individuals. Next, we train the machine learning classifier on synthetic training data and repeat the attacks. The resulting classifier is slightly less susceptible to attacks compared to the original classifier, which was trained on the original data. Our findings point in the direction that future research must pursue to investigate other model inversion attribute inference

Table 4.1: Threat model addressed by our approach.

Component	Description
Adversary: Objective	Specific sensitive attributes of the target individuals.
Adversary: Resources	A set of non-sensitive attributes of the target individuals, including the correct value for the propensity-to-move attribute, for "Inclusive individuals" (in the training set) or "Exclusive individuals" (not in the training set).
Vulnerability:Opportunity	Ability to query the model to obtain output plus the marginal distributions of the data set that the model was trained on.
Countermeasure	Modify the data on which the model is trained.

attacks, as well as other synthetic data techniques that could further reduce the risk of attacks when used to train machine learning models.

4.2. Threat Model

Our goal is to be able to make publicly accessible a machine learning model that has been trained on synthetic data such that the model maintains the same performance as a model trained on the original data, but is less susceptible to model inversion attribute inference attacks. In this section, we specify our goal more formally in the form of a threat model.

Inspired by [48], we include three main dimensions in our threat model. First, the threat model describes the adversary by looking at the resources at the adversary's disposal and the adversary's objective. In other words, it specifies what the attacker is capable of and what the attacker's goal is. Second, it describes the vulnerability, including the opportunity that makes an attack possible. Then, the threat model specifies the nature of the countermeasures that can be taken to prevent the attack.

Table 4.1 provides the specifications of our threat model for each of the dimensions. As objective, the attacker seeks to infer specific sensitive attributes of the target individuals. As resources, we assume that the attacker has collected a set of non-sensitive attributes of the target individuals, i.e., previously released data or data gathered from social media. The target individuals are either in the training data used to train the released model ("Inclusive individuals") or not in the training data ("Exclusive individuals"). The set of non-sensitive attributes also includes the target individuals' corresponding true labels concerning their propensity-to-move. The vulnerability is related to the opportunities available to the attacker, i.e., how the model is released and the access that has been provided to the model. The attacker is able to query the model and collect the output predictions of the model, for an unlimited number of arbitrary inputs. The attacker also has information about the marginal distribution for each attribute in the training data. Finally, the countermeasure that we are investigating is modification of the training data on which the model is trained.

4.3. BACKGROUND AND RELATED WORK

In this section, we give a brief overview of basic concepts and related work on predicting propensity to move, privacy in machine learning, and model inversion attribute inference attacks.

4.3.1. PROPENSITY TO MOVE

"Propensity to move" is defined as desires, expectations, or plans to move to another dwelling [144]. Multiple factors come into play to understand and estimate the propensity to move in a population. In [144], the authors have grouped those factors into two categories: (1) *Residential satisfaction*, which is defined as the satisfaction with the dwelling and its location or surroundings. Residential satisfaction is divided into housing satisfaction and neighborhood satisfaction. (2) *Household characteristics*, which is related to demographic and socioeconomic characteristics of the household. Gender and age are indicators of a household are important demographic attributes. For instance, a male household has different mobility patterns than a female household. Also, education and income of the household are important socioeconomic attributes.

In [145], the authors studied the social capital and propensity to move of four different resident categories in two Dutch restructured neighborhoods. They define social capital as the benefit of cursory interactions, trust, shared norms, and collective action. Using a logistic regression model, they show that (1) age, length of residency, employment, income, dwelling satisfaction, dwelling type, and perceived neighborhood quality significantly predict residents' propensity to move and (2) social capital is of less importance than suggested by previous research. In [146], the authors investigate the possible relationship between involuntary job loss and regional mobility. The authors look at whether job loss increases the probability of relocating to a different region and whether displaced workers who relocate to another region after job loss have better labor market outcomes than those staying in the same area. They find that job loss has a strong positive effect on the propensity to relocate. In [147], the authors use data collected by the British Household Panel Survey. The authors tested seven hypotheses to examine the reasons why people desire to move and how these desires affect their moving behavior. The results show that people are more likely to relocate if they desire to move for targeted reasons like job opportunities than if they desire to move for more diffuse reasons relating to area characteristics. In [61], the authors study the possibility of replacing a survey question about moving desires with a model-based prediction. To do so, they use machine learning algorithms to predict moving behavior. The results show that the models are able to predict the moving behavior about equally well as the respondents of the survey. In [148], the authors examine the residential moving behavior of older adults in the Netherlands. The authors of [148] use data collected from Housing Research Netherlands (HRN) to provide insights into the housing situation of the Dutch population and their living needs. A logistic regression model was used to assess the likelihood that respondents would report that they are willing to move in the upcoming years. They show that older adults are more often motivated by unsatisfactory conditions in the current neighborhood. Here, we follow up on the work of [61], as they evaluate a number of machine learning models to predict the propensity-to-move.

4.3.2. Privacy in Machine Learning

In this section, we will discuss challenges and possible solutions for privacy in machine learning. Existing works can be divided into three categories according to the roles of machine learning (ML) in privacy [143]: First, *making the ML model private*. This category includes keeping the ML model (its parameters) confidential and protecting the data it was trained on to control privacy threats. Second, *using ML to enhance privacy protection*. In this category, ML is used as a tool to enhance the privacy protection of the data. Third, *ML based privacy attack*. The ML model is used as an attack tool by the attacker.

Our work falls under the first category. The governmental institution that wishes to make the model available to the public needs to protect individuals in the training data and to make sure that by providing access to the model they are not making it easier to infer sensitive information about individuals not in the training data. As mentioned in Section 4.1, we approach the protection of the model by leveraging synthetic data. In prior work [149], the authors propose a one-step approach for differential private (DP) training of neural networks. They introduce differential private stochastic gradient descent (DPSGD) that achieves differential privacy by constraining the gradients to have a maximum l_2 norm for each layer. Existing DP research mainly focuses on protecting against membership inference attacks. In contrast, we employ a two-step approach in which we first synthesize data and then proceed to train a machine learning model.

4.3.3. Synthetic data generation

Synthetic data generation is based on two main steps: First, we train a model to learn the joint probability distribution in the original data. Second, we generate a new artificial data set from the same learned distribution. In recent years, advances in machine learning and deep learning models have offered us the possibility to learn a wide range of data types.

Synthetic data was first proposed for Statistical Disclosure Control (SDC) [37]. The SDC literature distinguishes between two types of synthetic data [37]. First, *fully synthetic data sets* create entirely synthetic data based on the original data set. Second, *partially synthetic data sets* contain a mix of original and synthetic values. It replaces only observed values for attributes that bear a high risk of disclosure with synthetic values.

In this paper, we are interested in fully synthetic data. For data synthesis, we used an open source and widely used R toolkit: *Synthpop*. We use a CART model for synthesis since it has been shown to perform well for other types of data [38]. Data synthesis is based on sequential modeling by decomposing a multidimensional joint distribution into conditional and univariate distributions. The synthesis procedure models and generates one attribute at a time, conditionally to previous attributes:

$$f_{x_1,x_2,..,x_n} = f_{x_1} \times f_{x_2|x_1} \times .. \times f_{x_n|x_1,x_2,..x_{n-1}} \tag{4.1}$$

Synthesis using the CART model has two important parameters. First, the order in which attributes are synthesized is called the *visiting sequence*. This parameter has an important impact on the quality of the synthetic data since it specifies the order in which the conditional synthesis will be applied. Second, the *stopping rules* are influenced by a limit

set to define the number of observations that must be present in a terminal node of the CART tree.

4.3.4. MODEL INVERSION ATTRIBUTE INFERENCE ATTACK

Model inversion attribute inference attacks try to recover sensitive features or the full data sample based on output labels and partial knowledge of some features [13], [150]. In [12], [52], the authors introduce two types of model inversion attacks: black-box attacks and white-box attacks. The difference between a black-box attack and a white-box attack lies in the resources that are available to the adversary. In a black-box setting, the adversary can only query the model and receive predictions [52]. The authors of [52] show that an attacker can use a trained classifier to extract representations of the training data. They exploit access to a model to learn information about its training data using confidence scores revealed in predictions. In [13], the authors provide a summary of possible assumptions about the adversary's capabilities and resources for different model inversion attribute inference attacks. The authors propose two types of model inversion attacks: (1) confidence score-based model inversion attack (CSMIA) and (2) label-only model inversion attack (LOMIA). The first attack, CSMIA, assumes that the adversary has access to the target model's confidence scores. The second attack, LOMIA, which is the basis of our work, assumes that the adversary has access to the target model's predictions only. The LOMIA attack uses an auxiliary machine learning model to infer sensitive information about target individuals. In our attack, we employ LOMIA to access the model's predictions, but we do not use an auxiliary model. We opt for LOMIA because it assumes the same adversary resources as in our threat model (Section 4.2). Other attacks such as [151] assume that the attacker does not have access to target individuals' non-sensitive attributes.

4.3.5. ATTRIBUTE DISCLOSURE RISK

In the context of statistical disclosure control, model inversion attribute inference attacks pose a risk to attribute disclosure. An adversary leverages predictive models (target ML model) to infer sensitive information about individuals from known attributes, increasing the likelihood of disclosure. Prior research on attribute disclosure risk has looked at various metrics to measure the risk of attribute disclosure. These metrics include matching probability, where perceived match risk, expected match risk, and true match risk are compared [152]. Additionally, a Bayesian estimation approach has been considered, where an attacker is assumed to seek a Bayesian posterior distribution [153]. Correct Attribution Probability (CAP) is another metric used to measure the risk of disclosure. CAP measures the proportion of matches between records from the original data and records from the protected data. Here, the protected data refers to the data that the adversary has used to query the target model accompanied by the inferred information. CAP calculates the ratio of correct attributions to total matches for a given individual [154], [155]. Methods for measurements of success are discussed in [6].

In the context of machine learning, we study attribute disclosure or attribute inference attacks as predictions. An attacker trains an auxiliary model to predict the value of an unknown sensitive attribute from a set of known attributes given access to raw or synthetic data [156], [157]. In this paper, we evaluate the success of our attack fol-

lowing [13], which measures the difference between the adversary's predictive accuracy given the model and the accuracy that could be achieved without the model. We consider our attack successful when its predictive accuracy surpasses that of a baseline using Marginals Only. This implies that using more information than just marginal data can reveal sensitive information about target individuals.

4.4. Label-Only MIA with Marginals

In this section, we describe our label-only MIA + Marginals attack (for short LOMIA + Marginals). LOMIA + Marginals is based on the LOMIA attack proposed by [13]. The attacker aims to predict the value of an unknown sensitive attribute from a set of known attributes. To perform the attack, the attacker needs access to the released ML model's predictions, the released marginal distribution representing possible values and actual probabilities for the sensitive attributes in the training data, and a subset of data containing information about target individuals' non-sensitive attributes. The attacker queries the target model multiple times by replacing the missing sensitive attribute with all possible values. To determine the value of the sensitive attribute, we follow Case (1) proposed in [13]. Case (1) states that when the target model's prediction is correct only for a single sensitive attribute value, e.g., $y = y_0^{'} \wedge y! = y_0^{'}$ or $y! = y_0^{'} \wedge y = y_0^{'}$, the attacker selects the sensitive attribute to be the one for which the prediction is correct. For instance, when the sensitive attribute is binary, i.e., $K = 2$, the attacker will query the model by setting the sensitive attribute value to both *yes* and *no* and leave other attributes unchanged. When the sensitive attribute is set to *no*, the returned model prediction is $y_0^{'}$. Similarly, when the sensitive attribute is set to *yes*, the returned model prediction is $y_1^{'}$. If $y = y_1^{'} \wedge y! = y_0^{'}$, the attacker predicts *yes* for the sensitive attribute. Differently, from [13], when the attacker cannot infer the sensitive attribute (for cases where the model's predictions vary across multiple sensitive attribute values, and cases where the model outputs incorrect predictions for all possible sensitive attribute values), we do not use an auxiliary machine learning model. Instead, the attacker relies on the released marginal distribution to predict the most probable value of the sensitive attribute.

In addition to the LOMIA + Marginals attack model, we also study an attack that uses *Marginals Only*, as a baseline for comparison. The Marginals Only attack uses the marginals to predict the most probable value of the sensitive attribute.

4.5. Experimental Setup

In this section, we describe our data sets, utility measures calculated by applying different machine learning classifiers, and adversary resources.

4.5.1. Data Set

For our experiments, we used existing data collected by [61] related to the propensity to move of individuals in the Netherlands. The authors of [61] linked various records from the Dutch System of Social Statistical Datasets (SSD). The data set has around 150K individuals including 100K individuals drawn randomly from SSD and 50K individuals are sampled from the Housing Survey 2015 (HS2015) respondents. The resulting data set has 700 attributes for each individual: (1) "y01" the binary target attribute indicat-

ing whether (=1) or not (=0) a person moved in year j where j= 2013, 2015. The target attribute "y01" is imbalanced and dominated by class 0. (2) time-independent personal attributes, (3) time-dependent personal, household, and housing attributes, (4) information about regional attributes.

FEATURE SELECTION

Different from [61], we applied feature selection to reduce the number of attributes. Some attributes can be noise and potentially reduce the performance of the models. Also, reducing the number of attributes helps to reduce the complexity of synthesis and to better understand the output of the ML model. To do so, we applied *SelectKBest* from *Sklearn*. We use the χ^2 method as a scoring function. We selected the top $K = 30$ attributes with the highest scores. Our final data set contains the 30 best attributes for a total of 150K individuals. In addition to the 30 attributes which include age, we added gender (binary) and income (categorical with five categories) as sensitive attributes. These sensitive attributes will be used in our model inversion attribute inference attack later (Section 4.6.2).

DATA SPLITS

As previously mentioned, our propensity to move data was collected in 2013 and 2015. Following [61], we use the 2013 data to train the propensity-to-move classifier and the 2015 data for testing. We call the 2013 data as "Inclusive individuals (2013)". Recall, that the 2015 data contains both individuals who were present in the 2013 data set ("Inclusive individuals (2015)") and also new "unseen" individuals ("Exclusive individuals (2015)"). We carry out tests on both sets individually.

In the synthesis process, we are interested in protecting the training data of the target propensity-to-move model. We use the Inclusive individuals 2013 data to train our synthesis model (cf. Section 4.3.3. The generated Inclusive individuals (2013) synthetic data is then used as input for training the target propensity-to-move model.

4.5.2. UTILITY MEASURES

In this section, we provide a description of the machine learning classifiers used in our experiments, as well as the metrics to evaluate the performance of these classifiers.

MACHINE LEARNING CLASSIFIERS

We selected a number of machine learning algorithms to predict the propensity to move. The chosen machine learning techniques provide insight into the importance of the attributes and are easy to interpret and understand [61].

In our experiments in Section 4.6.1, we used the following classifiers. *Decision Tree* creates/learns a tree by splitting the training data into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner. In *Random Forest*, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. *Extra Trees* fits a number of randomized decision trees on various sub-samples of the data set and uses averaging to improve the predictive accuracy and control overfitting. Extra Trees and Random Forest are ensemble methods. *Naïve Bayes* is a probabilistic machine learning algorithm based

on applying Bayes' theorem with strong (naïve) independence assumptions between the attributes. *K-nearest neighbors* (KNN) is a non-parametric machine learning algorithm. KNN uses proximity to make predictions about the grouping of an individual data point. We compare the performance of machine learning algorithms to the performance of a Majority-Class classifier using the most frequent strategy as a naïve baseline.

Metrics for Evaluating Performance of ML Classifiers

Similar to [61] and since our target propensity-to-move attribute is imbalanced, we used: F1-score, as a harmonic mean of precision and recall score. Matthews Correlation Coefficient (MCC), and Area Under the Curve (AUC) measure the ability of a classifier to distinguish between categories.

4.5.3. Adversary Resources

In this section, we describe different resources that are available for the attacker. As adversary resources, we assume that the attacker has access to a set of non-sensitive attributes of the target individuals (see our threat model in Section 4.2). We consider three cases corresponding to three different sets of individuals:

- `Inclusive individuals (2013)`: The attacker has access to data from the year 2013, which aligns with the data used to train the target model.
- `Inclusive individuals (2015)`: The attacker possesses more recent data from 2015, but it corresponds to the same set of individuals used in training the target model. The more recent nature of the data implies that certain (time-sensitive) attributes for specific individuals may have some changes.
- `Exclusive individuals (2015)`: The attacker's data is from 2015, but it pertains to a distinct group of individuals who were not part of the training set for the target model.

We created three different data sets for the three cases. Exclusive Individuals (2015) includes all available individuals (2904 individuals). For the Inclusive individuals (2013), used to train the target ML model as well as to create the synthetic data, and the Inclusive individuals (2015), we have randomly sampled to create data sets of the same size, each containing 2904 individuals. The attacker has access to the correct value of the propensity-to-move attribute for the target individuals but does not have information about the sensitive attributes of gender, age, and income, which are the objective of the attacker.

Understanding the vulnerability of a model to model inversion attribute inference attacks requires using the right metric to evaluate different attack models. Since our sensitive target attributes (gender, age, income) are balanced, we used precision, recall, and F1 to measure the effectiveness of the attacks. Precision measures the ability of the classifier not to label as positive a sample that is negative. Precision is the ratio of $tp/(tp+fp)$ where tp is the number of true positives and fp is the number of false positives. Recall measures the ability of the classifier to label positive samples positive Recall is the ratio of $tp/(tp+fn)$ where tp is the number of true positives and fn is the number of false negatives.

Table 4.2: Results of the performance of propensity-to-move model trained on **original data** and **synthetic data**. The test data is used in its original (unprotected) form.

Data Sets	Machine learning Algorithms	*Test set: Inclusive individuals (2015) and Exclusive individuals (2015)*			*Test set: Exclusive individuals (2015)*		
		AUC	**MCC**	**F1-Macro**	**AUC**	**MCC**	**F1-Macro**
Original Data	*Majority-Class*	0.5000	0.0012	0.4924	0.5000	0.0000	0.3758
	Naive Bayes	0.6815	0.2204	0.5968	0.5656	-0.0368	0.3331
	Random Forest	0.7532	0.2407	0.5946	0.7881	0.3425	0.5732
	Decision Tree	0.6568	0.2292	0.5767	0.7180	0.3478	0.6691
	Extra Trees	0.7219	0.2099	0.5764	0.7226	0.3197	0.6325
	KNN	0.6717	0.1744	0.5575	0.6723	0.2532	0.5981
Synthetic Data	*Majority-Class*	0.5000	0.0000	0.4900	0.5000	0.0000	0.3758
	Naive Bayes	0.6826	0.2029	0.5734	0.5657	-0.0144	0.3629
	Random Forest	0.7275	0.2426	0.5946	0.7870	0.3432	0.5900
	Decision Tree	0.6618	0.2125	0.5762	0.7189	0.3567	0.6728
	Extra Trees	0.7177	0.2082	0.5596	0.7233	0.3144	0.6425
	KNN	0.6542	0.1637	0.5418	0.6437	0.2027	0.5423

4.6. EXPERIMENTAL RESULTS

In this section, we turn to discuss our experimental results. We start by describing our results of the performance of the machine learning classifier. Then, we present the results of model inversion attribute inference attacks.

4.6.1. PERFORMANCE OF MACHINE LEARNING CLASSIFIERS

Table 4.2 shows our results of classification performance of propensity to move. In Table 4.2, the first column reports results on a test set that combines Inclusive Individuals (2015) and Exclusive Individuals (2015). This test setting is similar to [61]. The second column reports results in the case where the test set is Exclusive individuals (2015). In the latter case, we evaluate the performance of the trained propensity-to-move on unseen individuals who were not part of the training set.

As expected, all classifiers outperform the Majority-Class baseline, with classifiers using trees generally being the stronger performers. We also see that when the test set includes Inclusive individuals (2015) and Exclusive individuals (2015), the performance is better than when it includes only "unseen" individuals (Exclusive individuals (2015)). Note that if the data for the *inclusive* individuals were identical in the training and test set, we would have expected very high classification scores. However, the data is not identical because it was collected on two different occasions with two years intervening, and individuals' situations would presumably have changed.

REPRODUCING BURGER ET AL.,'S [61] RESULTS

In Table 4.2, results show that all machine learning classifiers outperform the Majority-Class baseline. Overall we observe that our results are in line with [61] across different metrics. This confirms that we can still predict individuals' moving behavior at the same level as in [61] even after reducing the number of attributes.

In addition to reproducing [61], we looked at another prediction model where train and test individuals are exclusive/different. We found that it is also possible to predict the moving behavior of new individuals from 2015 based on a classifier trained on different individuals from 2013.

Measuring the utility of synthetic data

In order to evaluate the quality of synthetic data, we run machine learning algorithms on a synthesized training set (2013 data). We used $TSTR$ (train on synthetic and test on real) [158] evaluation strategy where we train classifiers on 2013 synthetically generated data and we test on 2015 original data. Results in Table 4.2 show that the performance of machine learning classifiers trained on synthetic data is very close and comparable to the performance of machine learning algorithms trained on original data. This confirms that the synthetic training set can replace the original training set. In the remainder of the paper, we will focus on the Random Forest model. We will assume the release of a Random Forest model, as it outperforms other machine learning classifiers.

4.6.2. Results of Model Inversion Attribute Inference Attack

In this section, we discuss the results of model inversion attribute inference attacks on the propensity-to-move classifiers using gender, age, and income as the sensitive values. For comparison purposes, we begin by taking a look at the results generated by LOMIA Case (1) without adding the marginals, which we report in Table 4.3. Recall that LOMIA Case (1) (cf. Section 4.3.4) involves querying the model under attack with versions of the information of the target individual into which all possible values of the sensitive attribute have been substituted. #Predicted individuals reports the raw number of individuals for whom this querying process generates a prediction. If there is more than one version of the individual's information that produces the same result from the classifier, then that individual cannot be predicted. #Correctly predicted individuals reports the raw number of predictions that are correct.

First, we consider attacks on the ML models trained on original data. We see that Inclusive individuals (2013) shows the highest count of correct predictions across the conditions (gender, age, income). The correct predictions are relatively lower for both Inclusive individuals (2015) and Exclusive individuals (2015). This observation can be attributed to the use of Inclusive individuals (2013) for training the target ML model.

Then, we consider attacks on the ML models trained on synthetic data. For the case of Inclusive individuals (2013), we see that the model trained on synthetic data yields substantially fewer predicted individuals and also fewer correctly predicted individuals than the model trained on original data. For the case of Inclusive individuals (2015) and Exclusive individuals (2015), the prediction scores (considering both LOMIA + Marginals and Marginals Only) are also in general smaller for the model trained on synthetic data than for the model trained on original data.

The comparison in Table 4.3 illustrates that the use of synthetic data to train models is contributing to mitigate leaks, since models trained on synthetic data yield fewer correctly predicted individuals. However, it is important to note that for all models and all attributes, the vast majority of the 2904 individuals in each test set (i.e., Attacker resources) cannot be predicted with Case (1) and will be predicted using the marginals in

the attacks we discuss below. Also note that while using LOMIA during the attack, the attacker does not have the ground truth necessary to identify correctly predicted individuals and for this reason, the attack proceeds with the predicted individuals given in Table 4.3.

Table 4.3: Results of predictions returned from querying the target model using LOMIA Case (1) [13]. #Predicted individuals are the number of predictions returned from querying the target model. #Correctly predicted individuals represent correctly predicted records among all target individuals.

Attacker resources	***Target ML trained on***	***Sensitive Attributes***	**# Predicted individuals**	**# Correctly predicted individuals**
Inclusive individual (2013)	Original	*Gender*	92	92
		Age	38	37
		Income	37	37
	Synthetic (CART model)	*Gender*	79	35
		Age	17	7
		Income	27	6
Inclusive individual (2015)	Original	*Gender*	86	42
		Age	20	7
		Income	31	5
	Synthetic (CART model)	*Gender*	59	35
		Age	28	8
		Income	25	4
Exclusive individual (2015)	*Original*	*Gender*	281	148
		Age	72	16
		Income	58	24
	Synthetic (CART model)	*Gender*	124	59
		Age	47	13
		Income	57	16

Next, we move to discuss the results of our LOMIA + Marginals attack. Here, we start with the case of Inclusive individuals (2013). Table 4.4 summarizes the results of model inversion attribute inference attacks, comparing LOMIA + Marginals and Marginals Only attacks for Inclusive individuals (2013). Considering attacks on ML models trained on original data, we observe that the LOMIA + Marginals attack out performs the Marginals Only attack. Considering attacks on ML models trained on synthetic data, we see that the LOMIA + Marginals attack outperforms the Marginals Only attack for the age attribute, whereas it is surpassed by the Marginals Only attack for gender and income attributes. Recall that we saw in Table 4.3 that the vast majority of the predictions for individuals are carried out with Marginals. For this reason, we do not expect a large difference between LOMIA + Marginals and Marginals Only and it is not particularly surprising that a Margainals Only attack might sometimes outperform LOMIA + Marginals. Turning now to comparison, we see in Table 4.4 that the strongest attack on an ML model trained on original data (in this case LOMIA + Marginals) is always slightly more successful than the strongest attack on an ML model trained on synthetic data (in this case usually Marginals

Table 4.4: Case of "Inclusive individuals (2013)" as adversary resources: Evaluation results for model inversion attribute inference attacks using Marginals Only and LOMIA + Marginals attacks. Standard deviations, indicated by ±, represent variability across ten experiment runs.

Attacker resources	*Target ML Trained on*	*Attack Models*	*Gender*			*Age*			*Income*		
			F1-macro	**Precision**	**Recall**	**F1-macro**	**Precision**	**Recall**	**F1-macro**	**Precision**	**Recall**
Inclusive individual (2013)	Original	*Marginals Only*	0.4976 ± 0.0094	0.4977 ± 0.0094	0.4977 ± 0.0094	0.1237 ± 0.0068	0.1238 ± 0.0068	0.1238 ± 0.0068	0.1982 ± 0.0053	0.1982 ± 0.0052	0.1983 ± 0.0053
		LOMIA + Marginals	0.5155 ± 0.0080	0.5157 ± 0.0081	0.5157 ± 0.0080	0.1335 ± 0.0053	0.1336 ± 0.0052	0.1337 ± 0.0053	0.2105 ± 0.0072	0.2105 ± 0.0071	0.2106 ± 0.0072
	Synthetic (CART model)	*Marginals Only*	0.5035 ± 0.0072	0.5036 ± 0.0072	0.5036 ± 0.0072	0.1227 ± 0.0054	0.1228 ± 0.0055	0.1227 ± 0.0053	0.2020 ± 0.0081	0.2021 ± 0.0081	0.2020 ± 0.0081
		LOMIA + Marginals	0.4979 ± 0.0086	0.4980 ± 0.0087	0.4980 ± 0.0087	0.1259 ± 0.0057	0.1261 ± 0.0057	0.1261 ± 0.0057	0.1994 ± 0.0082	0.1995 ± 0.0082	0.1995 ± 0.0082

Only). We emphasize that the difference is very small, but the fact that it is discernible supports the conclusion that training models on synthetic data does have at least a basic potential for fighting ML model leakage.

For completeness, we present the results of LOMIA + Marginals and Marginals Only attacks when adversary resources are Inclusive individuals (2015) (Table 4.5) and Exclusive individuals (2015) (Table 4.6). Here, on the original data, the LOMIA + Marginals attack is not always more successful than the Marginals Only attack. However, we do see the trend that attacks are generally slightly less successful when the model is trained on synthetic data. We note the risk of leakage posed by the released marginals, and that, moving forward, the danger of releasing marginals must be studied alongside the danger of releasing the ML model itself.

Table 4.5: Case of "Inclusive individuals (2015)" as adversary resources: Evaluation results for model inversion attribute inference attacks using Marginals Only and LOMIA + Marginals attacks. Standard deviations, indicated by ±, represent variability across ten experiment runs.

Attacker Resources	*Target ML trained on*	*Attack Models*	*Gender*			*Age*			*Income*		
			F1-macro	**Precision**	**Recall**	**F1-macro**	**Precision**	**Recall**	**F1-macro**	**Precision**	**Recall**
Inclusive individual (2015)	Original	*Marginals Only*	0.5029 ± 0.0077	0.5029 ± 0.0076	0.5029 ± 0.0076	0.1239 ± 0.0065	0.1244 ± 0.0064	0.1241 ± 0.0067	0.1988 ± 0.0092	0.1991 ± 0.0090	0.1991 ± 0.0091
		LOMIA + Marginals	0.5034 ± 0.0124	0.5035 ± 0.0123	0.5035 ± 0.0123	0.1287 ± 0.0061	0.1291 ± 0.0062	0.1291 ± 0.0061	0.1980 ± 0.0070	0.1983 ± 0.0070	0.1984 ± 0.0070
	Synthetic (CART model)	*Marginals Only*	0.4937 ± 0.0083	0.4938 ± 0.0082	0.4938 ± 0.0082	0.1222 ± 0.0055	0.1225 ± 0.0057	0.1225 ± 0.0053	0.2031 ± 0.0066	0.2033 ± 0.0066	0.2035 ± 0.0067
		LOMIA + Marginals	0.5001 ± 0.0086	0.5003 ± 0.0085	0.5003 ± 0.0085	0.1278 ± 0.0028	0.1282 ± 0.0029	0.1281 ± 0.0028	0.1969 ± 0.0101	0.1972 ± 0.0102	0.1972 ± 0.0100

Table 4.6: Case of "Exclusive individuals (2015)" as adversary resources: Evaluation results for model inversion attribute inference attacks using Marginals Only and LOMIA + Marginals attacks. Standard deviations, indicated by ±, represent variability across ten experiment runs.

Attacker resources	*Target ML trained on*	*Attack Models*	*Gender*			*Age*			*Income*		
			F1-macro	**Precision**	**Recall**	**F1-macro**	**Precision**	**Recall**	**F1-macro**	**Precision**	**Recall**
Exclusive individual (2015)	*Original*	*Marginals Only*	0.5002 ± 0.0125	0.5012 ± 0.0126	0.5012 ± 0.0127	0.0880 ± 0.0031	0.1275 ± 0.0037	0.1323 ± 0.0185	0.1504 ± 0.0064	0.2001 ± 0.0059	0.2027 ± 0.0115
		LOMIA + Marginals	0.5007 ± 0.0065	0.5014 ± 0.0066	0.5014 ± 0.0066	0.0854 ± 0.0055	0.1234 ± 0.0050	0.1269 ± 0.0262	0.1506 ± 0.0065	0.2005 ± 0.0052	0.2008 ± 0.0123
	Synthetic (CART model)	*Marginals Only*	0.4966 ± 0.0078	0.4979 ± 0.0076	0.4979 ± 0.0076	0.0839 ± 0.0059	0.1233 ± 0.0053	0.1264 ± 0.0213	0.1447 ± 0.0058	0.1980 ± 0.0058	0.1952 ± 0.0097
		LOMIA + Marginals	0.4975 ± 0.0078	0.4989 ± 0.0078	0.4989 ± 0.0079	0.0852 ± 0.0030	0.1252 ± 0.0025	0.1242 ± 0.0146	0.1461 ± 0.0075	0.1985 ± 0.0068	0.1992 ± 0.0140

4.7. CONCLUSION AND FUTURE WORK

In this paper, we have investigated an attack on a machine learning model trained to predict an individual's propensity to move i.e., whether they will relocate in the next two years. We have studied the risk for Inclusive individuals, who are in the training data of the model, as well as for "unseen" Exclusive individuals.

To explore the ability of synthetic data to replace original data and protect against model inversion attribute inference attacks, we created fully synthetic data using a CART model. The ML model trained on the synthetic data maintained prediction performance and was found to leak in the same way or slightly less than the original classifier. This result is interesting as it shows that training a model on synthetic data will not exacerbate leaks, and may actually have the potential to reduce attribute disclosure risk. Also, our findings are interesting because until now the SDC community working with synthetic data has mainly focused on measuring the risk of identity disclosure rather than attribute disclosure [155]. In the identity disclosure literature, synthetic data has been shown to provide protection [15], [159]. However, releasing a model trained on synthetic data remains an open domain for research. Our work has highlighted the importance of considering the released marginals and not just the model.

Broadening the scope of the threat model is an essential avenue for future research (Section 4.2). Exploring additional attack scenarios, such as scenarios involving an attacker with access to confidence scores or confusion matrix from the target machine learning model or scenarios where the attacker lacks access to certain attributes within the data, would contribute to a better understanding of potential vulnerabilities associated with making a trained model publicly available. In terms of evaluation, future work should consider alternative metrics [157] from both statistical disclosure control (SDC) and machine learning perspectives to evaluate and quantify the success of model inversion attribute inference attacks for a given target individual. Also, it would be interesting to explore different synthesis approaches ranging from ML and generative models. If the inference attack is still possible, then, a second protection using privacy-preserving techniques on sensitive attributes during synthesis, e.g., data perturbation or masking sensitive attributes, might provide extra protection and reduce the risk of attribute disclosure. Furthermore, the choice of sensitive attributes is important given its impact on the output of model inversion attribute inference attacks. This consideration extends to understanding the nature of the relationship between sensitive attributes and the target attribute within the machine learning model.

5

EXPLORING PRIVACY-PRESERVING TECHNIQUES ON SYNTHETIC DATA AS A DEFENSE AGAINST MODEL INVERSION ATTACKS

This chapter is published as Manel Slokom, Peter-Paul de Wolf, and Martha Larson. Exploring Privacy-Preserving Techniques on Synthetic Data as a Defense against Model Inversion Attacks. Information Security Conference.

In this work, we investigate privacy risks associated with model inversion attribute inference attacks. Specifically, we explore a case in which a governmental institute aims to release a trained machine learning model to the public (i.e., for collaboration or transparency reasons) without threatening privacy. The model predicts change of living place and is important for studying individuals' tendency to relocate. For this reason, it is called a propensity-to-move model. *Our results first show that there is a potential leak of sensitive information when a propensity-to-move model is trained on the original data, in the form collected from the individuals. To address this privacy risk, we propose a data synthesis + privacy preservation approach: we replace the original training data with synthetic data on top of which we apply privacy preserving techniques. Our approach aims to maintain the prediction performance of the model, while controlling the privacy risk. Related work has studied a one-step synthesis of privacy-preserving data. In contrast, here, we first synthesize data and then apply privacy-preserving techniques. We carry out experiments involving attacks on individuals included in the training data ("inclusive individuals") as well as attacks on individuals not included in the training data ("exclusive individuals"). In this regard, our work goes beyond conventional model inversion attribute inference attacks, which focus on individuals contained in the training data. Our results show that a propensity-to-move model trained on synthetic training data protected with privacy-preserving techniques achieves performance comparable to a model trained on the original training data. At the same time, we observe a reduction in the efficacy of certain attacks.*

5.1. INTRODUCTION

A governmental institute that is responsible for providing reliable statistical information may use machine learning (ML) approaches to estimate values that are missing in their data or to infer attributes whose values are not possible to collect. Ideally, the machine learning model that is used to make the estimates can be made available outside of the institute in order to promote transparency and support collaboration with external parties. Currently, however, an important unsolved problem stands in the way of providing external access to machine learning models: the models may pose a privacy threat because they are susceptible to *model inversion attribute inference attacks*. In other words, they may leak information about sensitive characteristics of individuals whose data they were trained on ("inclusive individuals"). Further, going beyond the strict definition of model inversion, access to models may enable the inference of attributes of individuals whose data is not included in the original training set ("exclusive individuals").

In this paper, we investigate the potential leaks that could occur when external access is provided to machine learning models. We carry out a case study on a model that is trained to predict whether an individual is likely to move or to relocate within the next two years. Such models are helpful for understanding tendencies in the population to change their living location and are, for this reason, called *propensity-to-move models*. We study the case in which an institute would like to provide access to the model by allowing external parties to query the model and receive output predictions and by releasing the marginal distributions of the data the model is trained on. Additionally, the output might include confidence scores. Finally, access might include releasing a confusion matrix of the model calculated on the training data. Attackers wish to target a certain set of target individuals to obtain values of sensitive attributes for these individuals. We assume that for this set of target individuals, attackers possess a set of non-sensitive attributes that they have previously obtained, e.g., by scraping social media, including the correct value for the propensity-to-move attribute.

First, we show the effectiveness of our propensity-to-move prediction model. Then, we evaluate a number of existing model inversion attribute inference attacks [12], [13] and demonstrate that, if access would be provided to the model, a privacy threat would occur. Next, we address this threat by proposing a synthesis + privacy preservation approach, which applies privacy preserving techniques designed to inhibit attribute inference attacks on top of synthetic data. This two-step approach is motivated by the fact that within our case study, training models on synthetic data is an already established practice and the goal is to address the threat posed by synthetic data. In our previous work [160], we demonstrated that training on synthetic data has the potential to provide a small measure of protection, and here we build on that result.

Our results show that a propensity-to-move model trained on data created with our synthesis + privacy preservation approach achieves performance comparable to a propensity-to-move model trained on original training data. We also observe that the data created by our synthesis + privacy preservation approach contributes to the reduced success of certain attacks over a certain group of target individuals. Last but not least, we use the Correct Attribution Probability (CAP) metric [154] from Statistical Disclosure Control as a disclosure risk measure to calculate the risk of attribute disclosure for individuals.

We summarize our contributions as follows:

- **Threat Model**: Our attacks consider both target individuals who are included in the data on which the model is trained ("inclusive individuals") and target individuals who are *not* ("exclusive individuals"). Studying exclusive individuals goes beyond the strict definition of model inversion and is not well-studied in the literature.

- **Data synthesis + privacy preservation**: We explore a two-step approach that applies privacy-preserving techniques on top of synthetic data. Our approach aims to maintain model utility, i.e., the prediction performance of the model, while at the same time inhibiting inference of the sensitive attributes of target individuals.

- **Disclosure Risk**: In contrast to measures that rely on machine learning metrics, which often average or aggregate scores, we employ the Correct Attribution Probability (CAP) to quantify the level of disclosure risk for individual cases.

5.2. THREAT MODEL

We start characterizing the case we study in terms of a threat model [48], a theoretical formulation that describes: the adversary's objective, the resources at the adversary's disposal, the vulnerability that the adversary seeks to exploit, and the types of countermeasures that come into consideration. Table 5.1 presents our threat model. We cover each of the dimensions, in turn, explaining their specification for our case.

Table 5.1: Model inversion attribute inference threat model, defined for our case.

Component	Description
Adversary: Objective	Specific sensitive attributes of the target individuals.
Adversary: Resources	A set of non-sensitive attributes of the target individuals, including the correct value for the propensity-to-move attribute, for "inclusive individuals" (in the training set) or "exclusive individuals" (not in the training set).
Vulnerability:Opportunity	Ability to query the model to obtain output plus the marginal distributions of the data that the model was trained on. Additionally, the output might include confidence scores and a confusion matrix calculated on the training data might be available.
Countermeasure	Modify the data on which the model is trained.

As objective, the attacker seeks to infer sensitive information about a set of target individuals. As resources, we assume that the attacker has collected a set of data for each target individual, i.e., from previous data releases or social media. The set contains non-sensitive attributes of the target individuals and that includes the individual's ID and the corresponding true label for propensity-to-move. The target individuals are either in the training data used to train the released model ("inclusive individuals") or not in the training data ("exclusive individuals"). The vulnerability is related to how the model is released, i.e., the access that has been provided to the model. The attacker can query the model and collect the output of the model, both predictions and confidence scores, for unlimited number of inputs. The attacker also has information about the marginal distribution for each attribute in the training data. The countermeasure that we study is

a change in the model that is released, which is accomplished by modifying the training data.

5.3. BACKGROUND AND RELATED WORK

In this section, we provide a brief overview of existing literature on data synthesis, privacy-preserving techniques, and model inversion attribute inference attacks.

5.3.1. SYNTHETIC DATA GENERATION

Synthetic data generation methods involve constructing a model of the data and generating synthetic data from this model. These methods are designed to preserve specific statistical properties and relationships between attributes in the original data [14], [16], [159]. Synthetic data generation techniques fall into two categories [25]: partially synthetic data and fully synthetic data. Partially synthetic data contain a mix of original and synthetic records [37]. Techniques to achieve partial synthesis replace only observed values for attributes that bear a high risk of disclosure (i.e., sensitive attributes) [40]. Fully synthetic data, which we use in our experiments, creates an entirely synthetic data set based on the original data [37], [40]. Next, we discuss existing work on fully synthetic data generation from Statistical Disclosure Control [14], [159] and deep learning [161], [162].

Data synthesis in Statistical Disclosure Control Several approaches have been proposed in the literature for generating synthetic data, such as data distortion by probability distribution [163], synthetic data by multiple imputation [164], and synthetic data by Latin Hypercube Sampling [41]. In [38], the authors proposed an empirical evaluation of different machine learning algorithms, e.g., classification and regression trees (CART), bagging, random forests, and Support Vector Machines for generating synthetic data. The authors showed that data synthesis using CART results in synthetic data that provides reliable predictions and low disclosure risks. CART, being a non-parametric method, helps in handling mixed data types and effectively captures complex relationships between attributes [38].

Data synthesis using generative models A lot of research has been carried out lately focusing on tabular data synthesis [162], [165], [166]. In [165], the authors proposed *MedGAN*, one of the earliest tabular GAN-based data synthesis used to generate synthetic Health Records. MedGAN transformed binary and categorical attributes into a continuous space by combining an auto-encoder with GAN. In [166], the authors proposed *TableGAN*, a GAN-based method to synthesize fake data that are statistically similar to the original data while protecting against information leakage, e.g., re-identification attack and membership attack. TableGAN uses a convolutional neural network that optimizes the label column's quality such that the generated data can be used to train classifiers. In [162], the authors pointed out different shortcomings that were not addressed in previous GAN models, e.g., a mixture of data types, non-Gaussian and multimodal distribution, learning from sparse one-hot encoded vectors and the problem of highly imbalanced categorical attributes. In [162], a new GAN model called *CTGAN* is introduced, which uses a conditional generator to properly model continuous and categorical columns.

5.3.2. PRIVACY-PRESERVING TECHNIQUES

In this section, we provide an overview of existing work on privacy-preserving techniques. Privacy-preserving techniques can be categorized as perturbative or non perturbative methods. Perturbative methods involve introducing slight modifications or noise to the original data to protect privacy, while non perturbative methods achieve privacy through data transformation techniques without altering the data itself [14]. These techniques, which have been studied for many years, include randomization, data shuffling, data swapping [20], [57], obfuscation [55], post-randomization [167]. We discuss the privacy-preserving techniques that we use in our experiments in more depth:

Data swapping is a non-perturbative method that is based on randomly interchanging values of an attribute across records. Swapping maintains the marginal distributions in the shuffled data. By shuffling values of sensitive attributes, data swapping provides a high level of utility while minimizing risk of disclosure [57].

Post-randomization (PRAM) is a perturbative method. Applying PRAM to a specific attribute (or a number of attributes) means that the values of the record in the PRAMmed attribute will be changed according to a specific probability. Following notations used in [167], let ξ denote the categorical attribute in the original data to which PRAM will be applied. X denotes the same categorical attribute in the PRAMmed data. We suppose that ξ and X have K categories $1,\dots,K$. $p_{kl} = \mathbb{P}(X = l|\xi = k)$ denotes the transition probabilities that define PRAM. This means the probability that an original value $\xi = k$ is changed to value $X = l$ for $k, l = 1,\dots,K$. Using the transition probabilities as entries of a $K \times K$ matrix, we obtain $\boldsymbol{P}$ (called the PRAM-matrix).

Differential privacy has gained a lot of attention in recent years [30], [31]. Differential privacy (DP) uses a mathematical formulation to measure privacy. DP creates differentially private protected data by injecting noise expressed by ϵ into the original data. In [168] a differentially private Bayesian Network, PrivBayes is proposed to make possible the release of high-dimensional data. PrivBayes first constructs a Bayesian network that captures the correlations among the attributes and learns the distribution of data. After that, PrivBayes injects noise to ensure differential privacy and it uses the noisy marginals and the Bayesian network to construct an approximation of the data distribution. In [169], the authors introduced two methods for creating differentially private synthetic data. The first method adds noise to a cross-tabulation of all the attributes and creates synthetic data by a multinomial sampling from the resulting probabilities. The second method uses an iterative proportional fitting algorithm to obtain a fit to the probabilities computed from noisy marginals. Then, it generates synthetic data from the resulting probability distributions. A more recent work, Differentially Private CTGAN (DPCTGAN) [170] adds a differentially private noise to CTGAN. Specifically, DPCTGAN adds $\epsilon - \delta$ noise to the discriminator $\mathscr{D}$ and clips the norm to make it differentially private. We consider DPCTGAN to be a one-step synthesis approach, as it combines the application of noise and the synthesis process. Here, we test DPCTGAN, alongside our two-step synthesis + privacy preservation approaches.

5.3.3. MODEL INVERSION ATTRIBUTE INFERENCE ATTACKS

Privacy attacks on data [143] include identification (or identity disclosure) attacks [6], [133], [162], membership inference attacks [11], and attribute inference attacks (or at-

tribute disclosure) [6], [157], [171]. A lot of attention has been given to identification attacks on synthetic data [156], [172], [173]. However, less attention has been given to attribute inference attacks on synthetic data [172]. Attacks on data include attacks on models aimed at acquiring information about the training data. Here we investigate a model inversion attribute inference attack.

Model inversion attacks (MIA) aim to reconstruct the data a model is trained on or expose sensitive information inherent in the data [50], [54]. Attribute inference attacks use machine learning algorithms to predict, and perform attacks that infer sensitive attributes, i.e., gender, age, income. In a model inversion attribute inference attack, the attacker is interested in inferring sensitive information, e.g., demographic attributes, about an individual [12], [13], [143].

We distinguish between three categories of model inversion attribute inference attacks [50], [143]. An attack is black-box if the attacker only gets access to predictions generated by the model, i.e., can query the model with target individuals to receive the model's output. An attack is gray-box if the structure of the model and or some auxiliary information is further known, e.g., the attacker knows that the prediction is based on decision tree model, or attacker knows about the estimated weights of the model. An attack is white-box if an attacker has the full model, e.g., predictions, estimated weights or structure of model, and other information about training data.

In [12], [52], the authors showed that it is possible to use black-box access to prediction models (access to commercial machine learning as a service APIs such as BigML) to learn genomic information about individuals. In [12], the authors developed an attack model that exploits adversarial access to a model to learn information about its training data. To perform the attack, the adversary uses the confidence scores included with the predictions as well as the confusion matrix of the target model and the marginal distributions of the sensitive attributes. In [13], the authors proposed two attack models: confidence score-based MIA (CSMIA) and label-only MIA (LOMIA). CSMIA exploits confidence scores returned by the target model. Different from Fredrikson et al. [12], in CSMIA an attacker is assumed to not have access to the marginal distributions or confusion matrix. LOMIA uses only the model's predicted labels. CSMIA, LOMIA, and Fredrikson et al., [12] are the attacks we study in our work. The three attacks aim to achieve the adversary's objective of inferring sensitive attributes about target individuals, while assuming different resources and opportunities available to the attacker. (Further details are in Section 5.4.4). Other model inversion attacks use variational inference [54] or imputation [174] to infer sensitive attributes.

5.3.4. ATTRIBUTE DISCLOSURE RISK

Previous work on identity and attribute disclosure risk has looked either at matching probability by comparing perceived, expected, and true match risk [152], or at a Bayesian estimation approach, assuming that an attacker seeks a Bayesian posterior distribution [153]. Similar to [152], other work [154], [155], [157] has looked at the concept of Correct Attribution Probability (CAP).

CAP assumes that the attacker knows the values of a set of key attributes for an individual in the original data set, and aims to learn the respective value of a target attribute. The key attributes encompass all attributes within the data, excluding the sensitive at-

tribute that is the target attribute. Correct Attribution Probability (CAP) measures the disclosure risk of the individual's real value in the case where an adversary has access to protected data, and was originally proposed for synthetic data [155], [157]. The basic idea of CAP is that an attacker is supposed to search for all records in the synthetic data that match records in the original data for given key attributes. The CAP score is the proportion of matches leading to correct attribution out of the total matches for a given individual [155]. In [155], the authors extended their previous preliminary work [154]. They proposed a new CAP measure called differential correct attribution probability (DCAP). DCAP captures the effect of multiple imputations on the disclosure risk of synthetic data. The authors of [155] stated that DCAP is well-suited for fully synthetic data. In [175], the authors introduced TCAP, for targeted correct attribution probability. TCAP calculates CAP value for targeted individuals that the attacker knows their existence in the original data. In our experiments, we use the CAP measure introduced in [154].

5.4. EXPERIMENTAL SETUP

In this section, we describe our experimental setup. First, we provide an overview of our data set. Second, we describe how we synthesize data and the privacy protection techniques that we use. Next, we discuss target machine learning algorithms that we will use to predict propensity-to-move. Then, we describe the model inversion attribute inference attacks we study in our experiments.

5.4.1. DATA SET

For our experiments, we use a data set from a governmental institute. The data set was previously collected and first used in [61]. It combines different registers from the System of Social Statistical Data sets (SSD). In our experiments, we use the same version of the data set used in [160]. Our data contains 150K individuals' records between 2013 and 2015. We have 40 attributes (categorical and numerical) containing information about individual demographic attributes such as gender and age, and time-dependent personal, household, and housing attributes. The target attribute "$y01$" is binary, indicating whether (=1) or not (=0) a person moved in year j where $j = 2013, 2015$. The target attribute is imbalanced with 129428 0s (majority class) and 24971 1s (minority class).

We have three distinct groups of individuals within the data. The difference between the three groups resides in the fact that there are some individuals who are in the data in the year 2013 (called Inclusive individuals 2013). The same individuals appear again in the year 2015 (called Inclusive individuals 2015), where they may have different values for the time-dependent attributes than they did in 2013. The last group (called Exclusive individuals 2015) contains individuals who are "new in the country". We have a total of: 76904 Inclusive individuals 2013, 74591 Inclusive individuals 2015, and 2904 Exclusive individuals 2015.

Our propensity-to-move classifier (i.e., the target model) is trained on all 2013 data (76904 records). The classifier is tested on the 2015 data (77495 records) as in [160]. For the target model trained on (privacy-preserving) synthetic data, we use $TSTR$ evaluation strategy such that we train classifiers on 2013 (privacy-preserving) synthetically generated data and we test on 2015 original data [158], [160].

As adversary resources, we assume that the attacker has access to a set of non-sensitive attributes of the target individuals (see our threat model in Section 5.2). As in [160], we consider three cases:

- `Inclusive individuals (2013)`: the attacker has access to data from the year 2013, which aligns with the data used to train the target model.
- `Inclusive individuals (2015)`: Here, the attacker possesses more recent data from 2015, but it corresponds to the same set of individuals used in training the target model. The data being more recent implies that some of the (time-sensitive) attributes for particular individuals may have changed somewhat.
- `Exclusive individuals (2015)`: In this case, the attacker's data is from 2015, but it pertains to a distinct group of individuals who were not part of the training set for the target model.

We create data sets for each of the three cases. As in [160], for Exclusive individuals (2015) we use all 2904 individuals and for the other two cases we randomly sample to create data sets of the same size (2904 individuals each). The attributes of the target individuals that are in the possession of the attacker include the correct value of the propensity-to-move attribute but do not include the sensitive attributes gender, age, and income, which are targeted by the attack.

5.4.2. PRIVACY-PRESERVING TECHNIQUES ON SYNTHETIC TRAINING DATA

In this section, we describe how we synthesized data, and how we then applied privacy preserving approaches to it. The synthesis and privacy-preserving techniques are applied to the training data of the target model (the 76904 Inclusive individuals 2013), which is intended for release.

Our experiments with our two-step synthesis + privacy protection approach use a *classification and regression tree* (CART) model to synthesize data since it is shown to perform the best in the literature [38], [176]. Recall that CART is a non-parametric method that can handle mixed data types and is able to capture complex and non-linear relationships between attributes. We apply CART to the training data of the target model, which includes individuals from 2013. We use the open public R package, Synthpop for our implementation of the CART model [177][1]. Within Synthpop, there are a number of parameters that can be optimized to achieve a good quality of synthesis [177]. *Visiting.sequence* parameter specifies the order in which attributes are synthesized. The order is determined institute-internally by a human expert. *Stopping rules* parameter dictates the number of observations that are assigned to a node in the tree. Stopping rules parameter helps to avoid over-fitting.

Following synthesis using CART, we apply privacy-preserving techniques, data swapping and PRAM (cf. Section 5.3.2), to the synthetic data. We use two data swapping approaches, referred to as *Swapping* and *Conditional swapping*. For Swapping, we perform data swapping separately for each sensitive attribute, which includes gender, age, and income. Specifically, for the age attribute, we interchange numerical age values among

[1] `http://www.synthpop.org.uk/`

individuals and subsequently map these values to their respective age groups. For Conditional swapping, we perform simultaneous data swapping for gender, age, and income conditioned on the propensity-to-move target attribute. Conditional data swapping ensures that sensitive attributes are swapped while preserving the influence of the target attribute. Additionally, we apply Post-randomization (PRAM) independently to the attributes of gender, age, and income within the synthetic data generated using CART. Our transition matrices can be found in supplementary material.[2] We use the sdcMicro toolkit. [3] It is important to note that our evaluation includes separate testing of PRAM and data-swapping techniques.

In addition to experiments with our two-step synthesis + privacy protection approach, we explore a GAN-based one-step approach for generating (privacy preserving) synthetic data generation. We use *CTGAN*, a popular and widely used GAN-based generative model [162]. The data synthesis procedure of CTGAN involves three key elements, namely: the conditional vector, the generator loss, and the training-by-sampling method. CTGAN uses a conditional generator to deal with the class imbalance problem. The conditional generator generates synthetic rows conditioned on one of the discrete columns. With training-by-sampling, the conditional and training data are sampled according to the log frequency of each category. We used open public toolkit Synthetic Data Vault (SDV)[4] implemented in Python [39]. In our implementation, hyperparameter tuning is applied to batch size, number of epochs, generator dimension, and discriminator dimension. We left other parameters set to default. We generate differentially private CTGAN data using DPCTGAN, which takes the state-of-the-art CTGAN and incorporates differential privacy. We chose to make a comparison with CTGAN and DPCTGAN because of the success of the two techniques reported in the literature [162].

5.4.3. TARGET MACHINE LEARNING MODEL

In this section, we discuss the target machine learning algorithm used to predict the propensity to move. We trained and tested a number of machine learning algorithms, including decision tree, random forest, naïve Bayes, and extra trees. We found that all classifiers outperform the majority-class classifier, with classifiers using trees generally being the best performers. For simplicity, in the rest of the paper, we will use random forest classifier as it is shown to perform the best on the original data and on the synthetic data. We report the results of a random classifier using the most frequent (majority-class) strategy as a naïve baseline.

Recall that we must ensure that the prediction performance of the model is maintained when it is trained on synthetic + privacy-preservation data. To this end, we use the following metrics: F1-Macro, Matthews Correlation Coefficient (MCC), geometric mean (G-mean), True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP). Our choice is motivated by the imbalance of the target attribute.

The macro-averaged F1 score (F1-Macro) is computed using the arithmetic mean (i.e., unweighted mean) of all the per-class F1 scores. This method treats all classes equally regardless of their support values.

[2] Supplemental material is at this link in Section.2: `PRAM`

[3] `https://cran.r-project.org/web/packages/sdcMicro/sdcMicro.pdf`

[4] `https://github.com/sdv-dev/SDV`

The Geometric mean (G-mean) is the geometric mean of sensitivity and specificity [178]. G-mean takes all of the TP, TN, FP, and FN into account.

$$\text{G-mean} = \sqrt{\frac{TP}{TP+FN} * \frac{TN}{TN+FP}} \tag{5.1}$$

Matthews Correlation Coefficient (MCC) metric is a balanced measure that can be used especially if the classes of the target attribute are of different sizes [179]. It returns a value between -1 and 1.

$$\text{MCC} = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP+FP) * (TP+FN) * (TN+FP) * (TN+FN)}} \tag{5.2}$$

5.4.4. MODEL INVERSION ATTRIBUTE INFERENCE ATTACKS

In this section, we describe three model inversion attacks that we use in our paper: confidence-score MIA (CSMIA) [13], label-only MIA (LOMIA + Marginals), and the Fredrikson et al. MIA (FMIA) [12].

Confidence-Score MIA (CSMIA) [13] uses the output and confidence scores returned when the attacker queries the target propensity-to-move model. The attacker also has knowledge of the possible values for the sensitive attribute. For each target individual, the attacker creates different versions of the individual's records by substituting in for the missing sensitive attribute all values that would be possible for that attribute. The attacker then queries the model with each version and obtains the predicted class labels and the corresponding model confidence scores. Then, the attacker uses the predicted labels and confidence scores as follows [13]:

`Case (1)`: when the target model's prediction is *correct for only a single* sensitive attribute value, then, the attacker selects the sensitive attribute value to be the one for which the prediction is correct.

`Case (2)`: when target model's prediction is *correct for multiple* sensitive attribute values, then the attacker selects the sensitive value to be the one for which prediction confidence score is maximum.

`Case (3)`: when target model's prediction is *incorrect for all* sensitive attribute values, then the attacker selects the sensitive value to be the one for which prediction confidence score is minimum.

Label-Only MIA with Marginals (LOMIA + Marginals) is based on the LOMIA attack proposed by [13]. LOMIA + Marginals uses the output returned when the attacker queries the target propensity-to-move model and the marginal distributions of the training data (which includes the information about the possible values of sensitive attributes).

As with CSMIA, for each target individual, the attacker queries the target model multiple times, varying the value of the sensitive attribute. To determine the value of the sensitive attribute, the attacker follows `Case (1)` of CSMIA, as described in [13]. Specifically, if the target model's prediction is correct for a single sensitive attribute value, the attacker selects that value as the sensitive attribute. Differently from [13], for cases where the attacker cannot infer the sensitive attribute, we do not run an auxiliary machine learning model. Instead, the attacker uses the released marginal distribution to predict the most probable value of the sensitive attribute.

The Fredrikson et al. MIA (FMIA) [12] uses the output returned when the attacker queries the target propensity-to-move model and the marginal distributions of the training data. Following the threat model of [12], the attacker also has access to a confusion matrix of the target model's predictions on its training data. As with CSMIA and LOMIA + Marginals, the attacker queries the target model multiple times for each target individual, changing the sensitive attribute to take on all possible values and obtaining the predicted labels. Next, the attacker calculates the product of the probability that the target model's prediction aligns with the true label and the marginal distribution for each potential sensitive attribute value across all possibilities. Then, the attacker predicts the sensitive attribute value for which this product is maximized.

Measuring success of attribute inference attack We use two ways to measure attribute inference attacks:

(1) From a machine learning perspective, we evaluate the success of the attack by measuring precision (also called the positive predicted value (PPV) [174]). The precision metric measures the ratio of true positive predictions considering all positive predictions. A precision score of 1 indicates that the positive predictions of the attack are always correct.

(2) From statistical disclosure control, we use CAP to measure the disclosure risk of the individuals. Following [155], we define D_{org} as the original data and K_{org} and T_{org} as vectors for the key and target sensitive attributes of the original data: $D_{org} = \{K_{org}, T_{org}\}$. Similarly, we denote by D_{syn} as the synthetic data and K_{syn} and T_{syn} as the vectors for the key and target sensitive attributes of the synthetic data: $D_{syn} = \{K_{syn}, T_{syn}\}$. Note that when we are calculating CAP, the synthetic data we use is the data reconstructed by the attacker by inferring the missing sensitive value and adding it to the previously-possessed non-sensitive attributes used for the attack. We consider gender, age, and income as target sensitive attributes, evaluating CAP for each sensitive attribute separately. Key attributes are all other attributes for an individual except for the sensitive attribute being measured by CAP. The CAP for a record j is the probability of its target attributes given its key attributes.

$$\text{CAP}_{org,j} = Pr(T_{org,j}|K_{org,j}) = \frac{\sum_{i=1}^{M}\left[T_{org,i} = T_{org,j}, K_{org,i} = K_{org,j}\right]}{\sum_{i=1}^{M}(K_{org,i} = K_{org,j})} \tag{5.3}$$

where M is the number of records. The CAP score for the original data is considered as an approximate upper bound. Then, the CAP for the record j based on a corresponding synthetic data D_{syn} is the same probability but derived from synthetic data D_{syn}.

$$\text{CAP}_{syn,j} = (Pr(T_{org,j}|K_{org,j}))_{syn} = \frac{\sum_{i=1}^{M}\left[T_{syn,i} = T_{org,j}, K_{syn,i} = K_{org,j}\right]}{\sum_{i=1}^{M}(K_{syn,i} = K_{org,j})} \tag{5.4}$$

CAP has a score between 0 and 1: a low score (close to 0) indicates that the synthetic data has a little risk of disclosure and a high score (close to 1) indicates a high risk of disclosure.

5.5. PERFORMANCE OF THE TARGET MODELS

In this section, we compare the performance of the target propensity-to-move models. We evaluate whether a random forest classifier trained on protected synthetic data can attain performance comparable to a random forest classifier trained on the original data. Our results are reported in Table 5.2. Column "privacy-preservation" provides differ-

Table 5.2: Classification performance of the target model. We generate synthetic data using CART and CTGAN. For privacy-preserving techniques, we used swapping, conditional swapping, PRAM, and differential privacy ($\epsilon = 3$). In each case, the test data is used in its original (unprotected) form.

Target MLs to be Released	**Data sets**	**Privacy-preservation**	**F1-Macro**	**MCC**	**G-mean**	**TN**	**FP**	**FN**	**TP**
Majority-class	**Original data**	*None*	0.4924	0.0012	0.4924	46452	9539	17818	3686
Random Forest	**Original Data**	*None*	0.5946	0.2407	0.5779	61907	2363	10677	2548
Random Forest	**Synthetic data using CART**	*None*	0.5946	0.2426	0.5793	61848	2422	10628	2597
		Swapping	0.5881	0.2389	0.5742	62174	2096	10831	2394
		Conditional swapping	0.4654	0.0216	0.5028	63704	566	13034	191
		PRAM	0.5941	0.2415	0.5789	61844	2426	10638	2587
	Synthetic data using CTGAN	*None*	0.4586	0.0392	0.5021	64207	63	13155	70
		Differential privacy	0.4534	0.000	0.5000	64270	0	13225	0

ent privacy-preserving techniques that we applied to synthetic training data. "Privacy-preservation" with "None" means that there are no privacy-preserving techniques applied on top of the synthesis.

In Table 5.2, we see that random forest classifier trained on synthetic data using CART with *None* (i.e., no privacy-preserving technique applied) has quite close and comparable results to random forest classifier trained on original data. As a sanity check, we observe that both outperform the majority-class classifier.

We observe that in two cases the model trained on our synthesis + privacy preservation data retains a level of performance comparable to a model trained on the original data: CART with *Swapping* and CART with *PRAM*. Surprisingly, we find that when the training data is created with CART synthesis and *Conditional swapping* or CTGAN (with or without *Differential privacy*) the performance is comparable to that of a majority-class classifier. This result suggests that the use of conditional swapping and differential privacy may not effectively preserve the utility of the propensity-to-move data. For the rest of the paper, we will assume that we intend to release machine learning models trained on synthetic data using CART with: *None*, *Swapping*, and *PRAM* as privacy-preserving techniques.

5.6. RESULTS OF MODEL INVERSION ATTRIBUTE INFERENCE ATTACKS

In this section, we report the performance of different model inversion attribute inference attacks. We evaluate the performance of attacks on the model when it is trained on the original training data. Then, we investigate whether training the model on synthesis + privacy preservation data can protect against model inversion attribute inference attacks.

5.6.1. ATTACKS ON THE MODEL TRAINED ON ORIGINAL DATA

First, we look at the performance of model inversion attribute inference attacks on the target model trained on original training data. The results are reported in Table 5.3. The attack models show varying performances compared to the Marginals Only Attack.

Table 5.3: Results of model inversion attribute inference attacks measured using precision (positive predictive value) for three different target individual sets. The target propensity-to-move model is trained on **original training data**. Numbers in bold and italic represent the first and second best inference scores across conditions. A high precision indicates that the attack is good at correctly inferring the sensitive attribute values. We run experiments ten times and we report average scores. The standard deviation is below 0.01.

Adversary Resources	**Inclusive individuals (2013)**			**Inclusive individuals (2015)**			**Exclusive individuals (2015)**		
Attack models	*Gender*	*Age*	*Income*	*Gender*	*Age*	*Income*	*Gender*	*Age*	*Income*
Marginals Only	0.4977	0.1238	0.1982	0.5029	0.1244	0.1991	0.5012	0.1275	0.2001
CSMIA	0.3206	0.0105	0.0514	0.4660	0.0638	0.1581	0.4943	0.0721	0.1602
LOMIA + Marginals	*0.5157*	*0.1336*	*0.2105*	**0.5035**	**0.1291**	0.1983	*0.5014*	0.1234	**0.2005**
FMIA	**0.7563**	**0.6777**	**0.6898**	0.4647	0.0170	**0.2499**	**0.5205**	0.1091	0.1452

We observe that attribute inference scores for the attack models "LOMIA + Marginals" and "FMIA" outperform the inference scores of the Marginals Only Attack. In particular, FMIA for Inclusive individuals (2013) achieves the highest precision for all three sensitive attributes gender, age, and income. It outperforms other attack models in terms of correctly predicting positive instances. LOMIA + Marginals shows moderate performance, obtaining precision values higher than Marginals Only Attack. The fact that the attack performance for Inclusive individuals (2013) is highest is not surprising since these individuals are in the training set of the target model. For Inclusive individuals (2015) and Exclusive individuals (2015), we see that the performance for all attack models is relatively low and comparable to the Marginals Only Attack, except for a few cases such as FMIA on age for Inclusive individuals (2015). Recall that for FMIA, the attacker is exploiting a larger opportunity for attack than for the other attacks. Specifically, the attacker can query the model but also possesses the marginal distributions of the training data and a confusion matrix (cf. Section 5.4.4. For this reason, it is not particularly surprising that FMIA is the strongest attack).

5.6.2. ATTACKS ON THE MODEL TRAINED ON PROTECTED SYNTHETIC DATA

Second, we investigate whether we can counter the attack by replacing original data used to train target model by a privacy-preserving synthetic data. The results of the model inversion attribute inference attacks are reported in Table 5.4.

Overall we see that the effectiveness of the synthesis + privacy-preserving techniques varies across different attributes, attack models, and adversary resources (target sets). While some attributes have an inference score higher than the inference score of the Marginals Only attack, others only have comparable performance to the Marginals Only attack. We notice a decrease in the performance of attack models specifically for Inclusive individuals (2013) compared to the performance of attack models for the same group of individuals in Table 5.3. For Inclusive individuals (2015) and Exclusive individ-

Table 5.4: Results of model inversion attribute inference attacks measured using precision for three different target individual sets. The target propensity to move model is trained on *privacy-preserving (PP) + synthetic training* data. Numbers in bold and italic represent the first and second best inference scores across conditions. We run experiments ten times and we report average scores. The standard deviation is below 0.02.

PP+ Synthetic data	Attack Models	Inclusive individuals (2013)			Inclusive individuals (2015)			Exclusive individuals (2015)		
		Gender	*Age*	*Income*	*Gender*	*Age*	*Income*	*Gender*	*Age*	*Income*
Synthesis Only	*Marginals Only*	0.5036	0.1228	0.2021	0.4938	0.1225	0.2033	0.4979	**0.1233**	0.1980
	CSMIA	0.4901	0.0675	0.1423	0.4947	0.0775	0.1544	***0.5018***	0.1012	0.1826
	LOMIA + Marginals	0.4980	*0.1261*	0.1995	*0.5003*	**0.1282**	0.1972	*0.4989*	**0.1252**	0.1985
	FMIA	**0.5153**	0.0498	**0.3453**	**0.5007**	0.0588	**0.2772**	**0.5069**	0.1080	0.1452
Synthesis + Swapping	*Marginals Only*	0.4980	0.1238	0.1974	0.4979	0.1233	0.2060	0.4975	0.1248	**0.1973**
	CSMIA	0.4958	0.1198	**0.2032**	**0.4996**	0.1175	0.1848	***0.5093***	**0.1457**	*0.1986*
	LOMIA + Marginals	**0.5012**	**0.1280**	*0.1984*	0.4972	*0.1265*	0.1984	*0.5032*	**0.1242**	***0.1988***
	FMIA	0.4473	0.0901	0.0792	0.4320	**0.1362**	**0.3098**	**0.5351**	0.1020	0.1452
Synthesis + PRAM	*Marginals Only*	0.5002	0.1259	0.2010	0.5063	0.1239	0.2039	0.5002	0.1255	0.2000
	CSMIA	0.4967	0.1175	0.1701	0.4913	0.1059	0.1827	0.4895	**0.1371**	**0.2070**
	LOMIA + Marginals	**0.5038**	**0.1274**	0.1963	0.5004	0.1238	0.2002	0.5004	**0.1247**	**0.1987**
	FMIA	0.4827	0.0282	0.1635	**0.5286**	0.1129	0.1188	**0.5120**	0.1019	0.1452

uals (2015) which were not part of the training of the synthesis nor the training of the target model, we do not see a clear impact of privacy-preserving techniques on attack models. In most cases, the leak of sensitive information is low and comparable to the performance of the Marginals Only attack.

5.7. CORRECT ATTRIBUTION PROBABILITY

Now, we shift our focus to calculate the risk of attribute disclosure for individual target subjects using CAP (Correct Attribution Probability). CAP captures how many specific individuals face a high risk of attribute disclosure and how many a lower risk. We measure CAP using equation 5.4, where D_{org} is the attacker's data with key attributes K_{org} and the original target sensitive attribute T_{org} (gender, age, income). D_{syn} represents the attacker's data where $K_{syn} = K_{org}$ are the key attributes and T_{syn} is the outcome of the model inversion attribute inference attacks.

Figure 5.1 and Figure 5.2 show the frequency of CAP scores for sensitive attributes age and income, respectively. Due to space limitation, we specifically, focus on FMIA attack because it outperformed other attack models in Table 5.3. The top row of Figure 5.1 and Figure 5.2 shows the frequency of CAP scores on the original data (unprotected data). We see that across all three cases, Inclusive individuals (2013), Inclusive individuals (2015), and Exclusive individuals (2015), there is a high CAP score, signifying a high disclosure risk. However, when we calculate CAP scores based on the outcome of the model inversion attack, we observe that the risk of disclosure is relatively low, with approximately up to 92% of individuals considered protected. Only for the remaining individuals (8% individuals), we observe that an attacker can easily infer sensitive attributes age, and in-

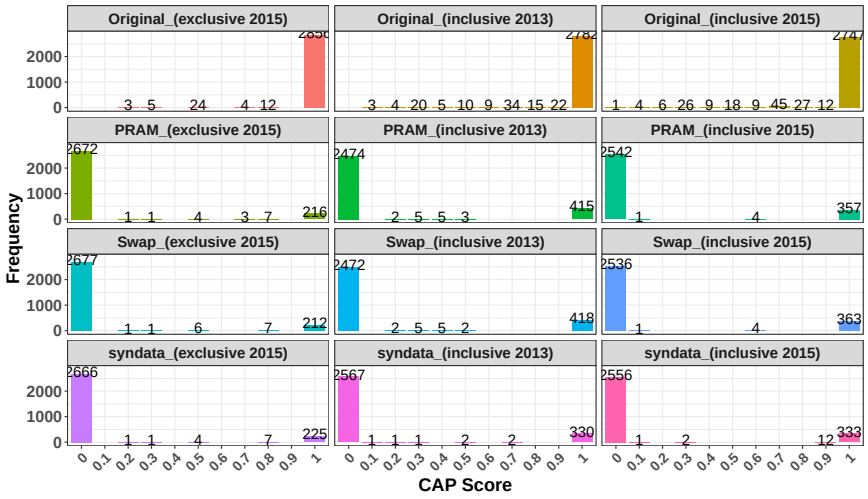


Figure 5.1: Frequency of CAP scores for attribute *age*. The total number of queries is 2904. The numbers inside the bars represent the count of individuals with corresponding CAP scores.

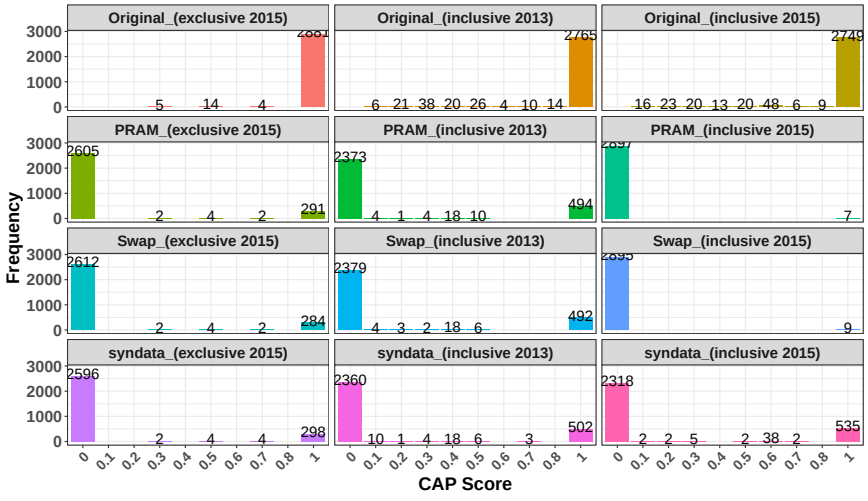


Figure 5.2: Frequency of CAP scores for attribute *income*. The attack model is FMIA. The total number of queries is 2904. The numbers inside the bars represent the count of individuals with corresponding CAP scores.

come with high CAP scores. Also, the number of disclosed individuals varies depending on the privacy-preserving technique applied. Comparing different resources, we see that for sensitive attribute age, Inclusive individuals (2013) have the highest number of disclosed individuals, next are Inclusive individuals (2015), and finally, Exclusive individuals (2015) have the lowest number of disclosed individuals. This aligns with the findings in Table 5.4. Notably, even though we generated privacy-preserving synthetic training

data sets, the target model appears to retain some information about the original data, leading to a risk of disclosure for certain individuals.

5.8. CONCLUSION AND FUTURE WORK

We have conducted an investigation aimed at protecting sensitive attributes against model inversion attacks, with a specific focus on a case study for a governmental institute. Our objective was to determine the feasibility of releasing a trained machine learning model predicting propensity-to-move to the public without causing privacy concerns. To accomplish this, we evaluated a number of existing privacy attacks, including CSMIA, LOMIA + Marginals, and FMIA, each distinguished by the resources available to the attacker. Our findings revealed that FMIA presented the highest degree of information leakage, followed by LOMIA + Marginals, while CSMIA exhibited the least leakage.

To mitigate these privacy risks, we employed privacy-preserving techniques on top of synthetic data utilized to train the machine learning model prior to its public release. Our results indicated that, in specific cases, such as with Inclusive individuals (2013), our privacy-preserving techniques successfully reduced information leakage. However, in other cases Inclusive individuals (2015) and Exclusive individuals (2015), the leakage remained comparable to that of a Marginals Only Attack, which uses the marginal distributions of the training data. We found a high disclosure risk, measured with CAP, when the target model is trained on original data. When the target model is trained on data protected with our two step synthesis + privacy preservation approach a lower percentage of individuals risk disclosure.

Furthermore, we think that the performance of the target machine learning model, as well as the correlation between the sensitive attribute and the target attribute, play a key role in the success of model inversion attacks. Future work should explore other case studies, in which this correlation might be different. Also, future work can look at other threat models such as white-box attacks, where the model predictions, model parameters, and explanation of the model's output are made public.

6

A Closer Look at User Attributes in Recommendations: Implications for Privacy and Diversity

This chapter is under preparation as Manel Slokom, Jesse Brons, Özlem Özgobek and Martha Larson. A Closer Look at User Attributes in Recommendations: Implications for Privacy and Diversity.

For over a decade, researchers have investigated the use of user attributes (gender, age, occupation, and location) to improve recommender systems. In this chapter, we take a closer look at user attributes from the perspective of privacy and diversity. With respect to privacy, we demonstrate that it is possible to infer attributes of a user from the list of items that a recommender system generates for that user. Since user attributes are potentially privacy sensitive, the risk of such inference constitutes a privacy leak. We carry out extensive experiments, with a context-aware recommender system, adding one category of user attribute at a time. Our results show that adding user attributes as side information during training can increase the size of the leak. With respect to diversity, further experiments show that adding user attributes can negatively impact diversity and coverage. In sum, our findings demonstrate that it is important for recommender system platforms to carefully consider their use of user attributes, since it may have unintended consequences for privacy and diversity.

6.1. INTRODUCTION

From the days of demographic recommender systems, mentioned in [180], researchers have attempted to make use of user attributes to improve recommendations. Context-based recommenders have included user attributes as side information, mixed with item and interaction attributes [181], [182]. With the rise of Graph Neural Networks, interest in leveraging user attributes has been recently renewed [182]–[184].

In this chapter, we look at user attributes from the point of view of privacy and diversity. Our aim is to gain insight into possible unintended consequences of using user attributes as side information in context-aware recommenders.[1] With respect to privacy, our study seeks to understand the extent to which personal attributes of a user can be inferred from a list of items recommended to that user. We are concerned about whether the use of user attributes as side information in context-aware recommendation increases the risk of exposure of users' personal information. We experiment with several categories of user attributes: gender, age, occupation and location. Note that although these specific attributes may not always be perceived as privacy sensitive, our conclusions can be expected to extend to attribute categories that clearly pose a threat to user privacy (e.g., ethnicity, religion, and health state), which are too sensitive to release in publicly available data sets. With respect to diversity, we investigate the effect of user attributes on the usefulness of recommendation lists for users. To the best of our knowledge, we are the first to take a closer look at the impact of user attributes as side information on recommender system algorithms and to investigate the user-level implications of inferring user attributes from recommender system outputs. Taking perspective of privacy and diversity allows us to open the broader question of whether the use of user attributes in context-aware recommendation is worth the unintended side effects.

The novelty of our work is twofold. First, we provide, a systematic study of how recommendation is improved when individual categories of user side information are used in context-aware recommendation. Previous work, e.g., [185], [186] has, to our knowledge, always studied user side information in combination with item and/or interaction side information. Isolating the impact of user attributes, as we do here, requires studying individual categories of user side information. We carry out a series of experiments on top-N recommender systems that make use of implicit (unary) user interactions, the currently dominant type of recommendation. Our experiments use three data sets: MovieLens 100K, MovieLens 1M [123], and LastFM [125]. These are the only publicly available data sets that we could identify at the time of writing that contain both user attributes and the temporal data that we need for recommender experiments using temporal splitting. MovieLens is well-studied, but must be transformed to yield implicit interactions. LastFM is a 'born implicit' data set, and we consider it more realistic for that reason. Second, we study privacy implications of the inference of user attributes from lists of recommendations. Previous work has also measured privacy threats in recommender systems using a classifier that infers user attributes; however, it has involved information internal to the system. Specifically, studies have been carried out on inferring user information from user representations [63], [187] and on inferring user information from

[1] Supplemental material with additional results is at this link: `All_detailed_results`.

a combination of recommendations and historical user data [81], [188]. In our work, we are interested in how user information can *leak* into the outside world, and for this reason, we perform inference of user attributes solely on the basis of the recommender system output, i.e., recommendation lists. Previously, one other paper [189] has investigated gender inference from recommendation lists. However, this work is substantially different from our own since they focus on fairness with respect to items and do not consider, as we do, user-oriented privacy or diversity.

We make the following contributions:

- We show that standard recommenders leak personal user information into the recommendation lists that they produce for a range of different user attributes.
- We report on extensive experiments with Factorization Machines, which are well suited to isolate the contribution of user attributes to recommendations. The results reveal that the use of user attributes as user side information in context-aware recommenders has the potential to increase the leak of personal information about a user via that user's recommendations.
- We demonstrate that using user attributes in context-aware recommendation yields a small gain in accuracy. However, the benefit of this gain is distributed unevenly over users and it sacrifices coverage and diversity.
- We discuss two possible approaches for reducing the privacy issues and their implications for diversity.

The chapter is structured as follows. In Section 6.2, we introduce the threat model under which we study recommendation list leakage. Section 6.3 covers the related work that is most relevant to our own. Section 6.4 describes the experimental setup and Sections 6.5– 6.8 present experimental results and analysis. Finally, in Section 6.9, we conclude and provide an outlook on how future work can work to reduce privacy leaks in recommender systems and maintain diversity.

6.2. THREAT MODEL

Studies of security threats and privacy leaks are carried out within a well-defined threat model, which characterizes the attacker in terms of objectives, opportunities, and resources [48]. In our case, the *objective* of the attacker is to infer sensitive attributes of a user. The opportunities and resources are defined on the basis of an attack scenario.

The *opportunity* available to the attacker is the ability to intercept recommendations that are provided by a recommender system to a set of users. This ability is also assumed given by [81], which studies transaction inference. The attacker does not have any knowledge of how the recommendations are generated, which makes our scenario a black-box attribute inference attack. The only assumption we make is that the attacker is able to listen in on an unprotected network connection. This assumption is stricter than that used by [63], [81], [187]. The attacker collects recommendations that are used for training a classifier capable of inferring sensitive information.

The *resources* available to the attacker are the ability to train a basic classifier and the availability of ground truth. Here, we assume that for a large enough number of

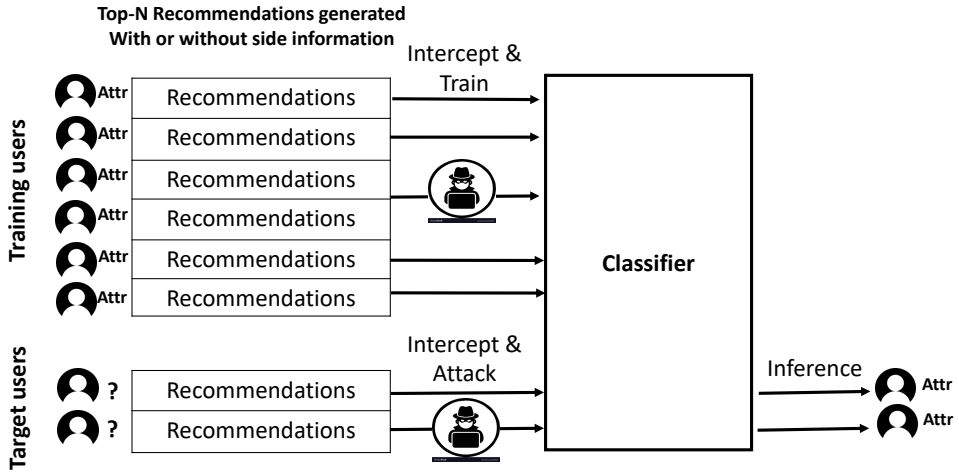


Figure 6.1: Our recommendation and inference attack general diagram. First, we generate recommendation outputs using different *Recommender system algorithms*. Second, *inference attack*, the attacker intercepts recommendation outputs and trains a classifier on training users, i.e., data scrapped from social media. Then, the attacker intercepts and attacks to infer sensitive information about target users.

users, the attacker is able to gather their demographic attributes on social media to use as ground truth [8]. The availability of ground truth online is not an unrealistic assumption. We recall the case of NetFlix de-anonymization uses data scraped from the Web [7].

In some work, a threat model also includes a specification for the possible *countermeasures*. In our work, we are focusing on exploring the existence of privacy leaks and discuss countermeasures in Section 6.8.

Figure 6.1 summarizes the inference attack that we study in this chapter: The input for training the recommender system is a user-item matrix for standard recommender system algorithms and a user-item matrix plus a category of user attribute for context-aware recommender system algorithms. To measure the privacy leak of users' recommendation lists, we carry out an inference attack. To train the classifier used for inference, the attacker intercepts recommendation lists that are provided by a recommender system, one list per user. For some of these users, personal information can be gathered from social media, which serves as ground truth to train the classifier. The attacker is targeting a set of users for which recommendation lists can be intercepted, but for which no personal information is available on social media. The success of the inference attack is measured in terms of the accuracy of the classifier. In our work, we consider a leak to be present when a classifier is able to outperform majority-class baseline. In this way, we avoid imposing assumptions on how large a privacy leak must be before it is considered dangerous. Note that even small leaks can be dangerous, because even uncertain information can harm users if it can be accumulated over a large number of sources.

6.3. RELATED WORK

In this section, we cover related work in areas that are most important for the study carried out in this chapter.

6.3.1. CONTEXT-AWARE RECOMMENDATION WITH USER SIDE INFORMATION

Context-aware recommenders integrate one or more of three types of side information: information related to users (e.g., age, gender), items (e.g., genre, price), and the interaction between users and items (e.g., time, location) [190], [191]. In this chapter, we study user side information since we are concerned about its privacy implications and we have not found another chapter that studies the isolated contribution of user attributes to context-aware Top-N recommendation.

Use of user attributes in recommender systems dates at least back to demographic recommender systems [180], as previously mentioned. Our experiments on context-aware recommendations focus on Factorization Machines (FMs) [192], a tried-and-true recommender that allows easy integration of side information via extension of the user-item vector. A Factorization Machine models pair-wise interactions with factorized parameterization and is suited to ranking problems with implicit feedback.

Recently, Graph Neural Networks (GNNs) have gained attention in the research community. Early GNN-based recommenders such as NGCF [193] and LightGCN [194] showed promising results. However, these recommenders, as with most of current graph-based recommenders use a bipartite user-item graph and do not offer the possibility of integrating side information [184]. Recently, [184] extended existing GNN-based recommenders using a pre-training scheme, which makes it possible to leverage user and item side information [184]. First, embeddings for entities, i.e., users and items, are pre-trained using side information. Specifically, [184] proposed two pre-training models: *Single-P* model and *Multi-P* model. The *Single-P* model learns entity embeddings on an undirected graph in which edges reflect a single symmetric relationship. The *Multi-P* model learns entity embeddings on a graph that encodes multiple, weighed relationships. Then, the pre-trained embeddings are fine-tuned using an existing recommender system algorithm such as Matrix Factorization, LightGCN [194], NGCF [193]. Since the performance of the two pre-training models is comparable, we use *Single-P* due to ease of implementation. The embeddings are integrated into LightGCN [194], which outperforms NGCF [193].

For completeness, we briefly cover other examples of recent context-aware recommenders that are not used in the experiments in this chapter, since they do not have publicly released implementations. Variational Autoencoder approaches include [195], which stacks denoising auto-encoders (SDAE) to integrate side information into the latent factors, and [186], which uses a collective Variational Autoencoder (cVAE) for integrating side information for Top-N recommendation. Recently, a clustering-based collaborative filtering algorithm that integrates user-side information (such as age, gender and occupation) in a deep neural network [196] has been proposed. Also, a Gaussian process-based recommendation framework that leverages side information [181] has been introduced. For further examples of recent work, see [197]–[199].

6.3.2. ATTRIBUTE INFERENCE ATTACK IN RECOMMENDER OUTPUT

Previous research has examined the issue of privacy-sensitive information leakage for a variety of sources of user data. Research on social media has shown that based on users' relationships i.e., users' friends and social circle, it is possible to infer users' locations [92], [200]–[202]. In [82], [203], the authors showed that based solely on the users' likes on Facebook it is possible to infer a range of sensitive attributes, such as gender, sexual orientation, ethnicity, and political view. In [204], the authors applied some strategies to estimate which of a user's friends are likely to be the most predictive of the user's location. In [202], the authors showed that an adversary is able to infer users' geolocations at given points of time, if s/he has access to a collection of locations previously disclosed by the users. This work is related to our own because it assumes the availability of previous information on social media. Note that in our work, this information is not needed for the specific users that are targeted by the attack.

Specifically regarding recommender systems, work has been carried out on the vulnerability of the user interaction data, used for training, to inference attack. An attacker can infer users' private attributes e.g., age, gender, occupation, location from the *input* data used to train a recommender system [22], [56]. As with our work, these authors find relatively small leaks important to study and are concerned about cases in which the classifier outperforms random guessing.

In this chapter, we are interested in the privacy leak in the *output* of recommender systems. In [188], the authors investigated the problem of user behavior leakage in recommender systems. Experiments showed that an attacker can infer information about the items that a user has clicked on the basis of that user's recommendation list. In [81], the output of a recommender system is combined with a limited number of known transactions to infer unknown transactions of a target user. In contrast, our work focuses on inference attacks that predict personal attributes of the user, rather than past interactions. Previous work studying attribute inference attacks on recommender output is limited, as previously mentioned. The most closely related work [63], [187] infers user attributes based on recommendation lists combined with additional information. In [63], the additional information is user embeddings that represent users internal to the recommender system. In [187], the additional information is the user's original profile, which is also internal to the recommender system.

6.3.3. DIVERSITY AND FAIRNESS IN RECOMMENDER SYSTEMS

Diversity in recommender systems has drawn attention in recent years [205]. Diversity can be defined as the potential of recommender system algorithms to recommend different or diverse content, e.g., recommending less popular items and targeting more niche items, while making personalized recommendations to users. In [206], the authors provide an overview of different definitions and measurements for diversity. Here, we are interested in the impact of side information on the diversification of the recommendation output. Diversity is important for recommendations to be useful to the user. Its importance is reflected in a surge of recent work on improving diversity, such as [207], a multi-attribute diversification method, and [208], attribute-aware diversifying sequential recommender (SR). Other work on diversity has focused on enhancing the user experience with system, such as [209], which showed the importance of diversity. In this

Table 6.1: Statistics of the data sets used for the experiments, including user attributes and item attributes. Class distribution shows how imbalanced the user attributes are.

Data set	#Users	#Items	#Interactions	Sparsity (%)	User attributes	Item Attributes
ML100K	845	1574	80961	6.08	Gender (**M**: 612, **F**: 233), Age (**0**: 47, **1**: 444, **2**: 184, **3**: 153, **4**: 17) Occupation (21), States (52)	Genres (19)
ML1M	5755	3624	831745	3.98	Gender (**M**: 4148, **F**: 1607), Age (**0**: 1245, **1**: 2017, **2**: 1664, **3**: 829) Occupation (21), States (52)	Genres (19)
LastFM	836	12155	501827	4.93	Gender (**M**: 484, **F**: 352), EU vs. rest (**EU**: 433, **Rest**: 403), Continent (7)	TrackIDs (500)

chapter, our focus is measuring diversity rather than attempting to improve it.

Additionally, fairness in recommender system has attracted a lot of attention recently. The goal of fairness is to make fair predictions along various dimensions [96], [210], [211]. In [103], the authors looked at pre-processing input to achieve consumer fairness, e.g., fairness oriented towards users. They explored whether different user demographic groups experience similar or different utility from the recommendation system. In [189], as previously mentioned, the authors investigate item-oriented fairness, seeking to ensure that items are recommended with equal frequency to both males and females. Indirectly, this notion of fairness is intended to benefit individual users, but the user-level effects are not measured. In our work, we analyze users with respect to the length of the user profiles, i.e., the number of items they have interacted with. We are interested in ascertaining if the use of user attributions in recommendation impacts users with different profile lengths.

6.4. EXPERIMENTAL SETUP

In this section, we describe the data sets, recommender algorithms and classifiers used in our experiments.

6.4.1. DATA SETS

Our experiments use two MovieLens data sets ML100K and ML1M [123] and LastFM [125], a music data set. As mentioned, to our knowledge these are the only publicly available data sets suit our needs for this study (i.e., contain user attributes and the timestamps needed for temporal splitting). Table 6.1 summarizes the statistics of the data sets as they were used in our experiments. MovieLens data sets include user gender, age, occupation, zipcode. We used zipcode to generate the State attribute. For our GNN experiment, which also involves item side information, we use movie genre (19 categories). As common in the literature we convert MovieLens data ratings to implicit feedback [212]. Here, we set the *cutoff* >= 3, such that items with ratings >= 3 are defined to be relevant, and the rest as non-relevant. We retain the users who have at least 20 interactions, to ensure at least two test items per user. For LastFM [125], we use artists as the items. For each user in LastFM data, gender and country location attributes are provided. We used the Country attribute to generate the Continent and the EU vs Rest attributes. As item side information, we use trackID (500 categories). We retain users who listened to at least 20 artists and artists to which at least 10 users have listened.

Our recommender system experiments use a temporal splitting strategy to mimic an online recommender, which cannot train on data from the future. The key idea is that the model should only learn from interactions that are available before a test instance [213]. For each user, the most recent 10% of the data is chosen as the test set, the next most recent 10% as the validation set, and the rest as the training set.

6.4.2. RECOMMENDER SYSTEM ALGORITHMS

For our recommendation experiments, first (Section 6.5), we investigate a number of *standard collaborative filtering algorithms* that are commonly used in recommender systems:

- MostPop is a non-personalized algorithm recommending most popular items.
- User-based (UserKNN) and item-based collaborative filtering (ItemKNN) [136], and
- BPRMF is a matrix factorization algorithm using Bayesian personalized ranking for implicit data [127].

We use Elliot and Lenskit toolkits for our implementation of standard recommender system algorithms.[2]

Then (Section 6.6.2), we study *context-aware recommendation* using a Factorization Machine recommendation algorithm [192], which is competitive with other approaches and allows for easy integration of user side information. User attributes (gender, age, occupation, and location) are one-hot-encoded as user side information for use by FM. We used the RankFM implementation,[3] which includes two variants for the loss: Bayesian Personalized Ranking (BPR) [214] and Weighted Approximate Rank Pairwise (WARP) [215] to learn model weights via Stochastic Gradient Descent (SGD) [214]. WARP loss is often described as performing better than BPR loss [216], [217], which was confirmed by our exploratory experiments. We adopt WARP loss for our investigation.[4]

Finally (Section 6.8), we study recommendations generated by a *Graph Neural Network* to investigate the leakage. We used GNN implementation of [184],[5] which uses two types of graph-structured data: *Single-P* model and *Multi-P* model to capture the interdependent and hidden relationships between entities. The *Single-P* model learns the entity embeddings on a single relational graphs using GNNs. The *Multi-P* model learns the entity embeddings on a mutlti-relational graphs using Compositional based Multi-Relational GNNs. We adopt the *Single-P* approach from [184], which is fine-tuned with LightGCN to generate recommendations. Our choice of pre-train GNN model is twofold: First, reproducibility and flexible incorporation of side information. Second, the authors of [184] showed that pre-training the embeddings with both the users and items' side information improved existing models in terms of both effectiveness and stability.

Our experiments use the validation set for tuning hyper-parameters including: batch size, the learning rate (lr), user and bias regularization, and the number of latent factors.

[2] `https://elliot.readthedocs.io/en/latest/index.html`
[3] `https://rankfm.readthedocs.io/en/latest/`
[4] Table with results of FM with BPR loss and FM with WARP loss can be found in the online folder `FM_WARP_BPR`
[5] `https://github.com/pretrain/pretrain`

For our factorization machine implementation, we search for the best: learning rate in {0.001,..,0.1}, number of training epochs in {5,..,500}, and latent factor in {5,..,200}. We left alpha and beta parameters at default. For GNN (Single-P), we followed hyper-parameters suggested in [184]. We adopt the Adam optimizer. We tune Single-P by varying the learning rate in [0.00001, 0.0001, 0.001] regularization weight in { 0.000001, .., 0.001}, the depth of GCN based recommender is kept to 3 with each layer having a size of 64. The maximum number of epochs is set to 500, batch size to 1000 and the latent dimension to 64. We use an early stopping strategy if the model on validation set does not increase for 50 successive epochs. Last but not least, we used MAP metric for optimizing hyper-parameters, as in [218].

In our experiments, we used *AllItems* methodology [129]. AllItems selects the whole set of items except the items that the user has interacted with in the training set. Let us assume I represents the set of items in the data set, Tr_u is the training set vector of user u (in other words, Tr_u represents the set of items that u has interacted with). Then, the list of candidate items for user u can be formulated as $L_u = I \setminus Tr_u$. For GNN, to speed up the process, we sample 1000 candidate items for each user.

We compute common top-N recommendation metrics: Precision (P@N), Recall (R@N), normalized Discounted Cumulative Gain (TopN.nDCG), and HR@N. In our results, we only report TopN.nDCG for simplicity. Recommendation results of Precision (P@N), Recall (R@N), and HR@N can be found in our supplementary material. We test recommendation lists of size $N = 5$ and $N = 10$.

The diversity of the recommendation lists is measured with item coverage, Gini index [205], and Shannon entropy. Item coverage computes the proportion of items that a recommender system recommends from the entire item catalog, taking all recommendation lists for all users into account. Gini index and Shannon entropy are two different metrics used to measure distributional inequality [205].

The Gini index reflects the difference between a given recommender system and an ideal system that recommends all items equally frequency. Gini index scores in the tables are represented through a reversed scale, obtained as *1 - Gini Index*: a high Gini index score is better and a low Gini score corresponds to a scenario in which items are not equally chosen.

The Shannon entropy is calculated over the distribution over all items of the probability that the recommender recommends each item. A Shannon entropy score is equal to 0 if one single item is recommended and reaches $log(N)$ if N items are recommended [205]. Also, we measure the log likelihood of the recommendation list for each test user. We first calculate $P(i)$, as the number of times an item i was recommended to test users divided by total number of recommendations to test users. Then, in order to get the log likelihood of a user's recommendation list for each item, we calculate: $\sum_i^N log(P(i))$. A low log likelihood means that a user received diverse recommendations.

Since we use different implementations of our recommendation algorithms ranging from conventional to context-aware and GNN, we unify our evaluation of recommendation performance by using ProxyRecommender which is a standard framework used to evaluate an already computed recommendation outputs.[6] In this way, we avoid mismatch that could happen in different implementations of evaluation metrics [219].

[6] https://elliot.readthedocs.io/en/latest/guide/proxy_model.html

6.4.3. CLASSIFICATION ALGORITHMS

We test three machine learning algorithms: Logistic Regression (LogReg), Random Forest (RF) and Neural Network (NN) because they are widely used for inference attacks in recommender systems literature [22], [63], [64], [187]. In our experimental results, we found that the LogReg classifier has close and comparable results to the RF classifier and NN classifier, with LogReg somewhat better. We report LogReg results here and the others in the supplemental material.[7] We adopt a random classifier using the most frequent strategy as our baseline. Our classifiers take users' top-N recommendation lists as input. We create a user-TopN recommendation matrix, where rows represent individual users and columns represent items that are in the catalog of items. If an item is recommended to a user, we mark it in the matrix with 1, otherwise 0. We create a 70/30 training/test split of the user-topN recommendation matrix for each classifier by user. We use stratified splitting with respect to the attribute classes that the classifier is trained to infer resulting in different splits for each attribute.

We carry out hyper-parameter tuning on the training data (stratified k-fold cross-validation with $k = 5$). We measure the performance of classifiers using F1-score with macro-average. We choose F1-score with macro-average because user attributes in our data sets are highly imbalanced. We calculate Brier score that measures the accuracy of probabilistic predictions. The Brier score measures the mean squared difference between the predicted probability of a classifier and the actual observed values [220]. The idea is to use confidence scores generated by classifier to measure the accuracy of prediction per user. A lower Brier score implies accurate predictions and vice versa.

6.5. LEAKAGE IN THE OUTPUT OF STANDARD RECOMMENDERS

In this section, we measure leakage of user attributes in five standard recommender systems (MostPop, ItemKNN, UserKNN, BPRMF, and FM) that do not use user-side information. Recommendation performance is reported in Table 6.2.[8] MostPop is outperformed by the other algorithms. The best algorithm is BPRMF or FM, depending on the data set.

We analyze the leakage of the recommender algorithms in terms of the performance of a classifier trained to predict user attributes. Results are shown in Table 6.3. Recall that we consider recommendations to leak when the classifier beats the majority-class baseline. According to this definition, ItemKNN, UserKNN, BPRMF, and FM leak all of the attributes. Note that MostPop leaks in all but two cases, although it is not a personalized algorithm. The leak arises because the implementation removes the items in a user's profile from that user's recommendation list, which introduces personal information into the list. The results in Table 6.3 suggest that there is not a single recommender algorithm that is more severely prone to leakage than others. We see that, with the exception of MostPop, for each recommender, there are several combinations of data set and leaking attribute for which that recommender has the largest leak.

[7] Full tables of three classifiers can be found here (Section 2 and Section 3) `All_Inference_results`.

[8] Full table with results of other metrics can be found in our additional materials (Section 1)`Recommendation_Performance`.

Table 6.2: TopN (N=5, 10) recommendation performance measured in terms of TopN.nDCG on conventional recommender system algorithms. FM with WARP loss is trained with no side information.

Data Sets	ML100K		ML1M		LastFM	
Algorithms	**N = 5**	**N = 10**	**N = 5**	**N = 10**	**N = 5**	**N = 10**
MostPop	0.0484	0.0583	0.0275	0.0383	0.2135	0.2079
ItemKNN	0.0704	0.0831	0.0342	0.0479	0.2790	0.2671
UserKNN	**0.0795**	0.0898	0.0334	0.0468	0.3190	0.3036
BPRMF	0.0748	0.0848	0.0561	0.0586	**0.3436**	**0.3132**
FM	0.0771	**0.0905**	**0.0639**	**0.0687**	0.3088	0.2888

Table 6.3: Classification results measured in terms of F1-score with macro-average. Recommendation lists are generated using standard recommender system algorithms. Majority-class classifier uses most frequent strategy. Values in bold represent the highest leak score. Values in italic represent cases where the LogReg classifier does not beat the majority-class baseline.

			Top-N = 5				**Top-N = 10**			
Data Sets	*Recommenders*	Classifiers	**Gender**	**Age**	**Occupation**	**State**	**Gender**	**Age**	**Occupation**	**State**
ML100K		*Majority-class*	0.4330	0.1381	0.0154	0.0053	0.4330	0.1381	0.0154	0.0053
	MostPop	*LogReg*	0.4428	*0.1354*	0.0412	0.0081	0.4464	0.1629	0.0280	0.0087
	UserKNN	*LogReg*	0.4865	0.1923	0.0492	**0.0165**	0.5012	0.1847	**0.0631**	0.0161
	ItemKNN	*LogReg*	0.5196	0.1789	**0.0674**	0.0095	0.4944	**0.2165**	0.0599	**0.0233**
	BPRMF	*LogReg*	**0.5334**	0.1390	0.0403	0.0145	**0.5642**	0.1631	0.0383	0.0081
	FM	*LogReg*	0.5015	**0.2012**	0.0394	0.0149	0.5470	0.1884	0.0329	0.0129
ML1M		*Majority-class*	0.4160	0.1299	0.0104	0.0058	0.4160	0.1299	0.0104	0.0058
	MostPop	*LogReg*	*0.4158*	0.2257	0.0308	0.0075	0.4327	0.2623	0.0421	0.0108
	UserKNN	*LogReg*	0.5665	0.3276	0.0587	0.0165	0.5840	0.3333	**0.0737**	0.0161
	ItemKNN	*LogReg*	0.5758	0.3330	0.0618	0.0164	0.6077	0.3354	0.0540	**0.0193**
	BPRMF	*LogReg*	0.5305	0.3364	**0.0627**	0.0103	0.5607	0.3602	0.0615	0.0182
	FM	*LogReg*	**0.6163**	**0.3671**	0.0520	**0.0171**	**0.6346**	**0.3730**	0.0723	0.0154
			Gender	**Continent**	**EU vs. Rest**		**Gender**	**Continent**	**EU vs. Rest**	
LastFM		*Majority-class*	0.3646	0.1126	0.3377		0.3646	0.1126	0.3377	
	MostPop	*LogReg*	0.5035	0.1298	0.4963		0.4990	0.1321	0.4704	
	UserKNN	*LogReg*	0.5323	0.1914	**0.5456**		0.5249	0.1897	0.5015	
	ItemKNN	*LogReg*	0.5250	**0.2092**	0.5171		0.5275	0.1776	0.4957	
	BPRMF	*LogReg*	**0.5479**	0.1721	0.5337		**0.5595**	0.1892	0.5328	
	FM	*LogReg*	0.5015	0.1719	0.5111		0.5160	**0.2205**	**0.5635**	

6.6. LEAKAGE IN THE OUTPUT OF CONTEXT-AWARE RECOMMENDERS

Now that we have established that recommender system output leaks, we turn to investigate context-aware recommenders. Our concern is that using user attributes as side information during training of a context-aware recommender will make it easier to infer these attributes from the recommender system output, i.e., increase the size of the leak. For our experiments we choose the Factorization Machine (FM) recommender, which is known for easy incorporation of side information, as previously mentioned. We have just seen, in Section 6.5, that FM has competitive performance and is prone to the same leakage as the other algorithms. In Section 6.6.1, we carry out experiments with individual categories of user side information, and measure the contribution of each category to recommendation performance. In Section 6.6.2, we measure the extent to which adding user information in the training data increases leaks.

6.6.1. CONTEXT-AWARE RECOMMENDATION WITH USER SIDE INFORMATION

The results in Table 6.4 show the difference between the performance of a recommender that does not use user side information ('None') and a series of recommenders in which individual categories of side information have been added. Our FM uses WARP loss, but we note that results of FM using BPR loss are comparable. From the results in Table 6.4, we observe that it is possible to obtain improvements in recommendation performance by using user attributes as side information. However, the attribute that is most useful and the size of the contribution differs across data sets. This variation is not surprising, since varying performance of algorithms across data sets has been observed in the literature [221], [222].

Table 6.4: TopN (N=5, 10) recommendation performance measured in terms of TopN.nDCG on FM with WARP loss ("None" means FM with no side information). In bold we mark recommendation with highest accuracy score. In italics, we mark scores of FM with user side information that are lower than scores of FM with 'None'.

Data Sets	ML100K		ML1M		LastFM		
User Attributes	**Top-N = 5**	**Top-N = 10**	**Top-N = 5**	**Top-N = 10**	*User Attributes*	**Top-N = 5**	**Top-N = 10**
None	0.0771	0.0905	0.0639	0.0687	*None*	0.3088	0.2888
Gender	0.0932	**0.1097**	0.0647	0.0688	*Gender*	**0.3196**	0.3049
Age	0.0888	0.1013	0.0644	*0.0684*	*continent*	0.3125	0.2996
Occupation	0.0903	0.1025	*0.0620*	*0.0657*	*EU vs Rest*	0.3188	**0.3061**
State	**0.0933**	0.1082	**0.0665**	**0.0721**			

Next, we order users by profile length and compare the users with the longest and the shortest profiles with the rest of the users. Results are provided in Table 6.5. We note that it is expected that the 10% of the users with the longest profiles have a low nDCG. This effect has been observed by [103], who remark that users with more items in their profile have already interacted with most of the "easy" items, so recommending for them is a harder problem. We observe that long-profile users benefit substantially from adding user side information. The 10% of the users with the shortest profiles often do not benefit at all. The rest of the users enjoy only a moderate benefit. These observations are interesting because at first our expectation would be that users with short profiles do not have enough information in their profiles and we would anticipate that these profiles would experience the most change. Our comparison of the benefits of user side information for users with different profile lengths provides first evidence that it is important to be

Table 6.5: The average TopN.nDCG on FM with WARP loss ("None" means FM with no side information) for long, medium, and short profile users on LastFM data.

Data Sets	LastFM					
User Attributes	**Top-N = 5 (short/ mid / long)**			**Top-N = 10 (short/ mid / long)**		
None	0.0550	0.0834	0.0450	0.0714	0.1153	0.0641
Gender	0.0529	0.0863	0.0532	0.0633	0.1202	0.0788
continent	0.0521	0.0847	0.0489	0.0693	0.1181	0.0739
EU vs Rest	0.0556	0.0840	0.0565	0.0821	0.1183	0.0817

careful when adding user side information to a recommender system, since it may not be helping users across the board.

6.6.2. MEASURING LEAKS IN CONTEXT-AWARE RECOMMENDERS

The results in Table 6.6 show the difference between the leak that occurs without side information ('None') and the leak that occurs when each individual category of user attribute is added as side information to the recommender during training. In a majority of cases, but not in all, adding a user attribute during training increases the size of the leak of that user attribute in the recommender output, as measured by the accuracy of the classifier. For LastFM, which we consider more realistic, than MovieLens, adding user attributes consistently increases the size of the leak.

Table 6.6: Classification results measured in terms of F1-score with macro-average. Recommendation lists are generated using FM with WARP loss. Values in bold represent the highest leak score between FM with None and FM with user side information using LogReg.

		Top-N = 5				**Top-N = 10**			
Data Sets	*User Attributes*	**Gender**	**Age**	**Occupation**	**State**	**Gender**	**Age**	**Occupation**	**State**
ML100K	None	**0.5015**	**0.2012**	0.0394	0.0149	**0.5470**	0.1884	0.0329	**0.0129**
	With side information	0.4871	0.1843	**0.0476**	**0.0162**	0.5269	**0.2112**	**0.0533**	0.0128
ML1M	None	0.6163	0.3671	0.0520	**0.0171**	0.6346	0.3730	**0.0723**	0.0154
	With side information	**0.6275**	**0.4025**	**0.0531**	0.0148	**0.6520**	**0.4401**	0.0704	**0.0197**
		Gender	**Continent**	**EU vs. Rest**		**Gender**	**Continent**	**EU vs. Rest**	
LastFM	None	0.5015	0.1719	0.5111		0.5160	0.2205	0.5635	
	With side information	**0.5578**	**0.2031**	**0.6197**		**0.5427**	**0.2430**	**0.6250**	

In Table 6.7, we provide the raw difference in performance of the recommender before and after user side information is added (rows labeled 'Recommendation') juxtaposed with the raw difference in classification performance on the recommendation lists before and after user side information is added (rows labeled 'Classification'). The highlight indicates cases in which improvement in recommendation and size of the leak move in the same direction. We can draw two conclusions from this table. First, across data sets there is not a connection between user attributes improving recommendation and user attributes increasing leaks. For ML1M, we see three cases (age and occupation for $N = 5$ and age for $N = 10$) in which adding user attributes increases the leak, but does not increase the recommender performance. Recommender system platforms should be careful to avoid this kind of use of user attributes: They are not helping the recommender and they are increasing leaks. Second, for LastFM we see that user attributes consistently improve recommendation and consistently increase the leaks. Recommender system platforms should consider carefully whether the magnitude of this improvement is worth the added privacy threat for their users.

To gain additional insight into the case of LastFM, we plot the nDCG against the Brier loss in Figure 6.2. The figure shows the FM with and without the user attribute 'Gender'. This figure reveals the lack of correlation between the performance of the recommender and the size of the leak at the user level. The lack of correlation holds for the FM with and

Table 6.7: Leakage and recommendation improvement reported as raw difference in classification between recommendation without and with user side information. Recommendation lists are generated using FM with WARP loss and values calculated using nDCG metric. The positive values in 'Recommendation' mean that user attribute helps to improve recommendation performance. The positive values in 'Classification' mean that the leak is larger when using user attributes. Gray marks cases where both classification and recommendation agree + or -.

		Top-N = 5				Top-N = 10			
Data Sets	***Task***	*Gender*	*Age*	*Occupation*	*State*	*Gender*	*Age*	*Occupation*	*State*
ML100K	Recommendation	+0.0161	+0.0117	+0.0132	+0.0162	+0.0192	+0.0108	+0.0120	+0.0177
	Classification	-0.0144	-0.0169	+0.0082	+0.0013	-0.0201	+0.0228	+0.0204	-0.0001
ML1M	Recommendation	+0.0005	-0.0019	-0.0019	+0.0026	+0.0001	-0.0003	-0.0030	+0.0034
	Classification	+0.0112	+0.0354	+0.0011	-0.0023	+0.0174	+0.0671	-0.0019	+0.0043
		Gender	*Continent*	*EU vs. Rest*		*Gender*	*Continent*	EU vs. Rest	
LastFM	Recommendation	+0.0108	+0.0037	+0.0100		+0.0161	+0.0108	+0.0173	
	Classification	+0.0563	+0.1086	+0.0312		+0.0267	+0.0615	+0.0225	

without user side information. The plots for other user attributes (included in supplementary material.[9]) also did not show correlation. These plots support the conclusion that future research should not assume the existence of a privacy/performance trade-off, but rather that there is possibility that reducing recommendation leaks can be achieved without reducing recommender system performance.

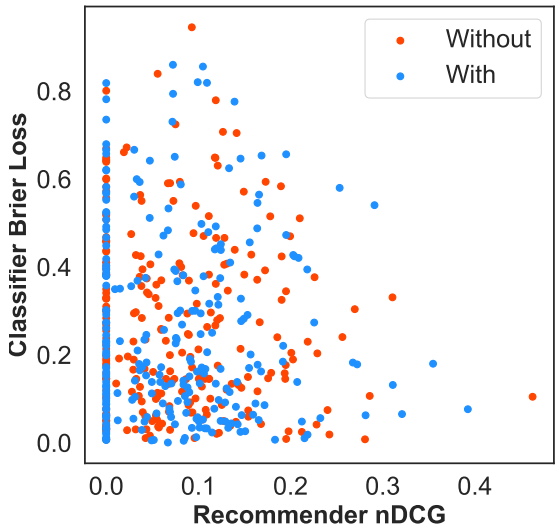


Figure 6.2: Scatter plot of users: The potential of LogReg to infer Gender vs. recommender performance (LastFM, Top-5 recommendation, FM with WARP loss). Blue: with user side information. Orange: without side information. (Classification test set contains 251 users.)

To gain further insight on the user level, we plotted the distribution of the gain in

[9] Results of correlation plots can be found here (in Section 5): `User_Analysis`.

nDCG over users when the user attribute 'Gender' is added as side information in Figure 6.3 (left). Instead, there are a large number of users that receive a small gain from adding user side information, but also users who do not benefit. We plot the raw change in Brier score in Figure 6.3 (right). The users that experience a larger leak has a negative change in Brier score. The plot shows us that users do not experience increases in leakage evenly. We see that there is a small group of users that experiences a large jump in their leak when user side information is used for recommendation. Protecting privacy should entail protecting all users. Seeking reductions in the average leak over all user may leave some users still unacceptably threatened.

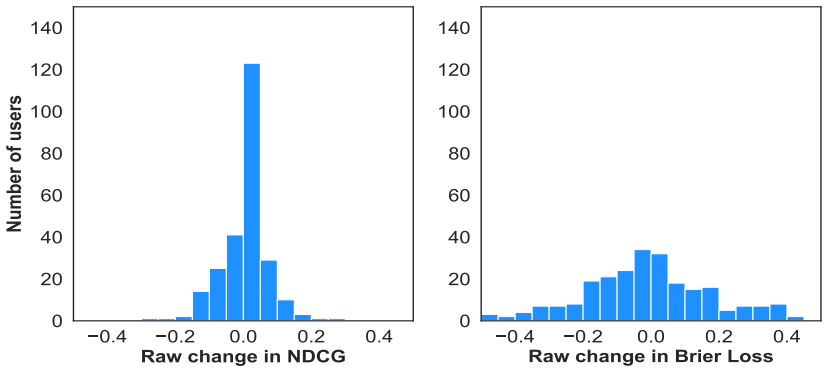


Figure 6.3: Left: Raw change in nDCG (score of user u with Gender minus the score of user u without Gender). A *positive* value in raw change in nDCG mean that adding user side information helped to improve recommendation performance. (STD = 0.067) Right: Raw change in Brier score (score of user u with Gender minus score of user u without Gender). A *negative* value in raw change in Brier loss means a larger leak when user side information is added. (STD = 0.216) (LastFM, Top-5 recommendation, FM with WARP loss; Classification test set contains 251 users.)

6.7. DIVERSITY AND COVERAGE OF THE RECOMMENDER OUTPUT

Now, we move to investigate the impact of user side information on coverage and diversity. Coverage is reported as the percent of items recommended and, diversity is measured in Shannon entropy and Gini index (higher is more diverse). Table 6.8 contains the results of recommendation using FM with WARP loss with and without user side information. We observe that compared to the recommender without side information ('None'), most user attributes depress coverage. We also observe that user attributes deteriorate diversity. Attribute categories with few subcategories (such as Gender and Age) show the fewest exceptions. The results support our conclusion that user side information has the potential to cause unintended side effects in terms of loss of coverage and diversity.

Table 6.8: Item coverage and diversity of recommendation lists from Factorization Machine with WARP loss. The higher the scores the better the diversity and coverage. ↓ and ↑ indicate the change with respect to 'None' for the conditions with user side information.

Data Sets	User Attributes	Top-N = 5			Top-N = 10		
		Item coverage	Shannon Entropy	Gini index	Item coverage	Shannon Entropy	Gini index
ML100K	*None*	415	7.770	0.109	546	8.097	0.136
	Gender	315↓	7.275↓	0.077↓	422↓	7.657↓	0.099↓
	Age	336↓	7.075↓	0.070↓	461↓	7.564↓	0.096↓
	Occupation	424↑	7.697↓	0.105↓	563↑	8.072↓	0.134↓
	State	369↓	7.514↓	0.092↓	507↓	7.888↓	0.117↓
ML1M	*None*	840	7.995	0.056	1110	8.376	0.072
	Gender	647↓	7.572↓	0.042↓	1181↑	8.363↓	0.074↑
	Age	687↓	7.308↓	0.037↓	902↓	7.741↓	0.049↓
	Occupation	779↓	7.657↓	0.047↓	985↓	8.058↓	0.058↓
	State	901↑	8.031↑	0.059↑	1181↑	8.363↓	0.074↑
LastFM	*None*	1180	9.173	0.041	1802	9.547	0.054
	Gender	1107↓	9.167↓	0.039↓	1625↓	9.543↓	0.051↓
	Continent	1152↓	9.246↑	0.042↑	1668↓	9.595↑	0.053↓
	EU vs Rest	814↓	8.483↓	0.025↓	1195↓	8.895↓	0.033↓

6.8. COUNTERING PRIVACY LEAKS

In this section, we take a look at two approaches that are capable of countering privacy leaks and discuss their implications for recommendion performance and diversity.

6.8.1. GNN-BASED RECOMMENDATION

In this section, we turn to investigate whether combining user and item attributes as side information to context-aware recommenders could cause the items in recommendation lists to be blended in such a way that it becomes more difficult to infer sensitive user attributes.

First, we carry out an experiment with a GNN (Single-P) that combines user attributes and item attributes [184]. Table 6.9 reports recommendation results. We compare performance of FM with no side information ('None') and FM with all user and item attributes, and GNN (Single-P) with all user and item attributes. User attributes and item attributes are one-hot-encoded as user and item side information for use by FM and GNN (Single-P).

Comparing FM 'None' and FM with all user and item attributes, we note that when all user and item attributes are added, the performance drops. This result is surprising, since in Table 6.4 we observed that adding individual user attributes improves recommendation performance. However, the literature has previously observed that user side information is difficult to exploit, citing this challenge as a motivation for adopting GNN [184], Specifically, in [184], the authors state that most of existing context-aware recommenders adopt an integration scheme for side information that could result in a conflict or disagreement between the recommendation loss and side information-aware loss, resulting in low recommendation performance and propose a GNN designed to address this issue.

Second, we turn to consider the GNN [184]. Interestingly, LastFM is the only data set on which the GNN using all available side information consistently outperforms the FM using no side information and FM using all user and item attributes across both con-

Table 6.9: TopN (N=5, 10) recommendation performance measured in terms of TopN.nDCG on FM with 'None', FM using using *all* user attributes and item attributes and GNN with Single-P model using *all* user attributes and item attributes.

Algorithms	*Side information*	ML100K		ML1M		LastFM	
		Top-N = 5	**Top-N = 10**	**Top-N = 5**	**Top-N = 10**	**Top-N = 5**	**Top-N = 10**
FM	*None*	0.0771	0.0905	0.0639	0.0687	0.3088	0.2888
	All User & item Attributes	0.0518	0.0759	0.0444	0.0488	0.2811	0.2200
GNN (Single-P)	*All User & item Attributes*	0.0757	0.0985	0.0626	0.0800	0.4928	0.4827

Table 6.10: Classification results measured in terms of F1-score with macro-average. Recommendation lists are generated using FM (with WARP) and GNN (using Single-P). For FM, upper row we use one attribute at a time and bottom row we use FM with *all* user and item attributes. For GNN (Single-P) we use all user and item attributes. We focus on Logistic Regression classifier (LogReg).

Classifier= LogReg			Top-N = 5				Top-N = 10			
Data Sets	*Algorithms*	*Side information*	**Gender**	**Age**	**Occupation**	**State**	**Gender**	**Age**	**Occupation**	**State**
ML100K	*FM*	User attribute	0.4871	0.1843	0.0476	0.0162	0.5269	0.2112	0.0533	0.0128
		User & Item Attributes	0.5032	0.1604	0.0643	0.0127	0.5404	0.2202	0.0350	0.0100
	GNN (Single-P)	User & Item Attributes	0.4807	0.1513	0.0327	0.0145	0.4823	0.2018	0.0390	0.0224
ML1M	*FM*	User attribute	0.6275	0.4025	0.0531	0.0148	0.6520	0.4401	0.0704	0.0197
		User & Item Attributes	0.6288	0.4176	0.0696	0.0160	0.6405	0.4625	0.0834	0.0161
	GNN (Single-P)	User & Item Attributes	0.4179	0.1759	0.0289	0.0061	0.4234	0.2037	0.0341	0.0106
			Gender	**Continent**	**EU vs. Rest**		**Gender**	**Continent**	**EU vs. Rest**	
LastFM	*FM*	User attribute	0.5578	0.2031	0.6197		0.5427	0.2430	0.6250	
		User & Item Attributes	0.5781	0.1941	0.5458		0.5162	0.1820	0.4817	
	GNN (Single-P)	User & Item Attributes	0.4711	0.1366	0.5062		0.5164	0.1806	0.5592	

ditions and all metrics. This results lead to the observation that recommender system platforms should test carefully before assuming that the combination of GNN and all possible side information will necessarily provide improved recommendations.

Next, we move to consider whether adding all user and item attributes can block classifiers from inferring sensitive user attributes from recommender system output. We report our results in Table 6.10. First, we compare FM adding only a single user attribute with FM adding all user and item attributes. We see that using all user and item attributes sometimes reduces the leak, but sometimes makes it worse. Moving to GNNs that add user and item side information, we see more promise: GNNs reduce the leak across the board, with the exception of State on ML100K. The results point to the conclusion that the large number of different information sources added to the recommender training data is having an obfuscating effect on the recommender list.

Unfortunately, the picture is less positive when we look at the coverage and diversity scores, which are provided in Table 6.11. Here, again we compare the FM with no side information to FM and the GNN with all user and item attributes. We see that adding all user and item attributes to FM reduces the coverage and diversity scores across the

Table 6.11: Item coverage and diversity of recommendation lists generated by FM using "None", FM using *all* user attributes and item attributes, and GNN (Single-P) model using all user and item attributes. ↓ and ↑ indicate the change with respect to 'None' for the conditions with side information.

Data Sets	Algorithms	Side Information	Top-N = 5			Top-N = 10		
			Items coverage	***Shannon Entropy***	***Gini index***	***Items coverage***	***Shannon Entropy***	***Gini index***
ML100K	*FM*	*None*	415	7.770	0.109	546	8.097	0.136
		All User & item Attributes	215↓	6.357↓	0.039↓	302↓	6.853↓	0.0553↓
	GNN (Single-P)	*All User & item Attributes*	128 ↓	6.145↓	0.033↓	163↓	6.698↓	0.050↓
ML1M	*FM*	*None*	840	7.995	0.056	1110	8.376	0.072
		All User & item Attributes	431↓	6.985↓	0.027↓	575↓	7.448↓	0.037↓
	GNN (Single-P)	*All User & item Attributes*	125↓	4.749 ↓	0.005↓	220↓	5.650↓	0.010↓
LastFM	*FM*	*None*	1180	9.173	0.041	1802	9.547	0.054
		All User & item Attributes	732↓	8.365↓	0.022↓	1107↓	8.816↓	0.0296↓
	GNN (Single-P)	*All User & item Attributes*	162↓	6.034↓	0.004↓	299↓	6.879↓	0.007↓

board. Unfortunately, moving to the GNN makes the situation even worse, with even larger reductions of diversity and coverage. These results demonstrate that combining user and item attributes is an effective approach for addressing leaks of recommender system output, but that more work is necessary to ensure that these combinations do not impact coverage an diversity.

A possible way forward is GNNs such as [223], which achieves diversified GNN-based recommendations by improving the embedding generation procedure. This GNN mainly focuses on finding a subset of diverse neighbors to aggregate for each GNN node and the learning of items belonging to long-tail categories. We mention it as possible future work, but do not test it here since it is not designed to integrate side information.

6.8.2. POST-PROCESSING METHODS

Next, we discuss post-processing methods, which modify the recommender system output, could be used to reduce the leak. Specifically, we take a closer look at [189], a post-processing method is proposed that iteratively updates recommendation lists until it is not longer possible to infer gender from recommendation lists that have been output by a recommender system. As previously noted, the authors of [189] seek to improve item-oriented fairness, and are not concerned with privacy or diversity. Specifically, their goal is to balance the number of times a given item is recommended to males and to females, making it as equal as possible. They show that blocking the ability to infer gender from the recommendation list achieves this goal. Future work should investigate the potential of their work for protecting privacy, but must also broaden the notion of fairness. To motivate this point, we plotted (Fig. 6.4) the Brier score against the user-level log-likelihood of the recommendation lists, which reflects the extent to which recommendation lists contain popular items or a mix of popular and non-popular. We see that there is no

correlation, meaning that there is no clear impact of blocking inference of gender on the ability of the recommender to recommend niche items. Contrary to what is implied by [189], blocking inference of user attributes from recommender list falls short of being a fail safe approach to achieving arbitrary forms of fairness.

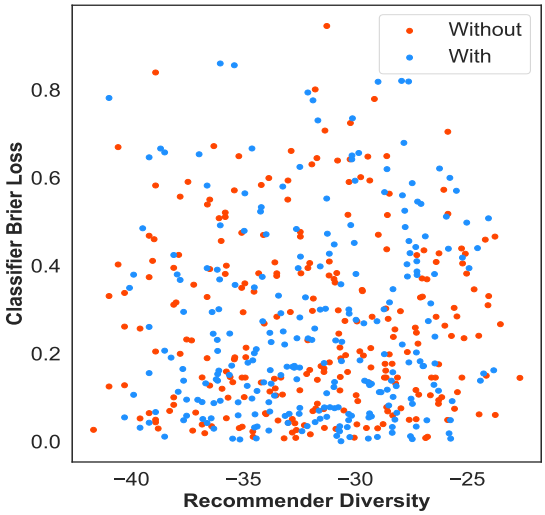


Figure 6.4: Scatter plot of users: The potential of LogReg to infer Gender vs. diversity of recommendation (LastFM, Top-5 recommendation, FM with WARP loss). Blue: with side information. Orange: without side information. (Classification test set contains 251 users.) A low Brier score implies accurate predictions and vice versa.

6.9. Conclusion and Future Work

In this chapter, we have looked at user attributes from a perspective of privacy and diversity. In terms of privacy, we have seen that standard recommender systems leak and that using user attributes as side information during the training of a context-aware recommender system may exacerbate this leak. In terms of diversity, we have seen that using user attributes restricts the coverage of a recommender system and lowers the diversity. On the basis of these results, we conclude that recommender system platforms should consider carefully whether it is actually advantageous to make use of user attributes for training recommender systems. If platforms do not use them, it is wise not to collect and store them in the first place, which helps to control the privacy risk should the system be breached.

By demonstrating the presence of leaks we have opened the door for future research on techniques to reduce them. We carried out an initial experiment (cf. Tab. 6.10) that showed that the combination of many different pieces of side information might make inference more difficult. However, it is important to consider whether side information

is actually bringing a substantial benefit and to ensure that there are no hidden 'side effects' of side information, such as negative implications for diversity.

Another possible approach is to obfuscate the input data. Previous work has shown such obfuscation to be capable of blocking inference attacks on user profiles [22], [56], and the blocking effect might transfer to recommender output. Further, perturbations could be applied to user and item embeddings, as done by [63], [184]. These approaches involve simultaneously optimizing for maximizing recommendation performance and minimizing leaks by combining user interactions and user recommendations. Future work should follow our threat model and test the leak of the recommendation list alone.

In closing, we would like to emphasize an important negative result. In Fig. 6.2 we have seen no clear correlation between the recommender system performance (nDCG) for individual users and the ease with which those users' profiles can be used to infer user attributes (Classifier Brier Loss). This observation directly contradicts the assumption of [189], who assume that recommendation performance must get worse as it becomes more and more difficult to infer user attributes from recommendation lists. This observation is important for future work in recommender systems in two respects. First, we should not assume that making recommendation lists more indicative of a particular user attribute, i.e., 'female' will better satisfy users with that attribute. Instead, overfitting on user attributes risks adversely impacting diversity. Second, on the basis of this observation, we conclude that future research that develops methods to reduce privacy leaks, should not assume that there is a trade-off between leak reduction and recommender system performance. It seems that the best of both is worth pursuing.

III

Conclusion and Future Work

7

OUTLOOK

In this thesis, we have presented various purpose-aware privacy-preserving techniques for machine learning and recommender system algorithms. Our approaches, along with our methodologies, findings, and empirical results, have been presented across five technical chapters. In this chapter, we summarize our contributions in Section 7.1. We outline potential avenues for future research in Section 7.2. We close the thesis with reflections that we think are interesting and promising in the context of user privacy and ML applications in Section 7.3 .

7.1. MAIN CONTRIBUTIONS AND DISCUSSION

Throughout the thesis, we explored distinct aspects related to purpose-aware privacy-preserving techniques for predictive applications. Firstly, we specified the threat models in terms of the resources at the adversary's disposal, the adversary's objective, the opportunity that makes an attack possible, and the nature of the countermeasures that can be taken to prevent the attack. Secondly, we introduced our conceptual framework of purpose-aware privacy-preserving techniques. Our conceptual framework extends existing privacy-preserving techniques by emphasizing the importance of "the purpose". The essence of purpose-aware techniques lies in the necessity to specify the intended purpose for which data modifications are made. Thirdly, we applied purpose-aware privacy-preserving techniques to two ML prediction applications: recommender systems and machine learning classifiers. Within these applications, we focused not only on maintaining the accuracy performance of the models, i.e., predictions using machine learning classifiers and recommendation performance but also on protecting individuals' sensitive information. In this section, we summarize our contributions and findings.

Attacking the input data The first part of the thesis focused on answering the first research question (RQ1): *How can we protect sensitive information when the attacker has access to* ***the input data*** *from inference attack while maintaining utility?* Specifically, we

focused on attack scenarios in which the adversary aims to infer users' sensitive information by attacking the input data.

In **Chapter 2**, we assumed that the adversary has a gender classifier that is pre-trained on unobfuscated data or has data to train a classifier. The adversary's objective is to infer the gender of individual users. We proposed, PerBlur, a purpose-aware privacy-preserving users' gender for recommender systems. PerBlur extended state-of-the-art data obfuscation techniques with its use of personalization and greedy item removal. PerBlur is formulated within a user-oriented paradigm for user privacy. The paradigm involves three dimensions: obfuscation should be understandable, obfuscation should be unobtrusive, and obfuscation should be useful. We showed how obfuscation is a simpler task than one might think. It is a very simple approach fully *understandable* to users. Also, we should not assume that obfuscation must always introduce noise. If we keep obfuscation close to the user preferences it has the potential to be *unobtrusive* for the user. We demonstrated how obfuscation is *useful* as it maintains the quality of the recommendation. Also, we showed how recommendation performance should not be the sole goal of obfuscation, but instead diversity and fairness as well.

In **Chapter 3**, we assumed that the adversary has access to released data i.e., Data science or RecSys Challenge. The objective is to infer the preferences of individual users. We proposed Shuffle Non-Nearest Neighbors (Shuffle-NNN for short), a purpose-aware privacy-preserving users' preferences for recommender systems. Shuffle-NNN extended state-of-the-art data masking techniques with its use of neighborhood selection and value-swapping steps. Neighborhood selection preserved valuable item similarity information. The data shuffling technique hid ratings of users for individual items. We demonstrated that our data masking approach has great potential for data science challenges. We showed that it is possible to develop a masking approach, such that masked data can be used to train and test algorithms with little impact on the relative performance of algorithms. We demonstrated that Shuffle-NNN provides valuable evidence about what information can be removed from the user-item matrix and what information should be maintained.

Attacking the output data The second part of the thesis was dedicated to addressing two distinct research questions, RQ2 on *how can we protect sensitive information when the attacker has access to the* ***model's predictions*** *while maintaining utility?* and RQ3 on *whether* ***recommender system's output data*** *leaks sensitive information about users.* We looked at attack scenarios where the adversary's objective is to infer users' sensitive information by targeting the output data, which comprises predictions generated using an ML model. In RQ2, we focused on a machine learning classifier predicting individuals' propensity to move (Chapter 4 and Chapter 5). Here, we explored how to protect sensitive information when the attacker has access to the model's predictions while maintaining utility. In RQ3, we focused on the output of a recommender system algorithm predicting users' next preferred items to be consumed, commonly referred to as the recommendation list (Chapter 6). Our investigation was related to determining whether recommendation lists leak sensitive user information.

In **Chapter 4**, we investigated an attack on a machine learning classifier that predicts the propensity of a person or household to move (i.e., relocate) in the next two years. The

attack assumes that the classifier has been made publicly available and that the adversary has collected a set of non-sensitive attributes of target individuals, i.e., previously released data or data gathered from social media. The objective of the adversary is the inference of specific sensitive attributes of the target individuals. We investigated Label-Only MIA + Marginals (LOMIA + Marginals) attack model, allowing unlimited queries to the model and leveraging marginal distributions for predictions. The adversary queries the model and collects the output predictions of the model. First, we found that the attack is possible where LOMIA + Marginals outperforms the Marginals only attack. Then, we investigated whether training the classifier on a data set that is synthesized from the original training data, rather than using the original training data directly, would help to mitigate the attack. Our experimental results indicated that the risk of attribute disclosure is somewhat comparable, and in certain cases even lower, when using synthetic training data to train the machine learning model.

In **Chapter 5**, we extended the threat model of Chapter 4 by evaluating a number of existing privacy attack models, including Label-Only, LOMIA + marginals, confidence-score based MIA (CSMIA), Fredrikson et al. MIA (FMIA). The attack models differ based on the opportunities available to the attacker. First, we found that FMIA presented the highest degree of information leakage, followed by LOMIA with Marginals, while CSMIA exhibited the least leakage when a model was trained on original data. We proposed to replace the original data used to train the target model prior to its release with protected data with data synthesis + privacy-preserving techniques. We demonstrated that, in specific cases, our protected data successfully reduced information leakage. However, in other cases, the leakage remained comparable to Marginals Only attack. Also, we found a high disclosure risk, measured with CAP, when the target model is trained on original data. But, when the target model is trained on data protected with our two step synthesis + privacy preservation approach a lower percentage of individuals risk disclosure.

In **Chapter 6**, we assumed that the adversary is able to intercept recommendations that are provided by a recommender system to a set of users, i.e., to listen in on an unprotected network connection. The adversary's objective is to infer the sensitive attributes of a user. We generated recommendation lists using different recommender systems algorithms, ranging from standard collaborative filtering techniques such as ItemKNN, userKNN, and BPRMF, to context-aware recommenders using factorization machine (FM) with and without user attributes as side information and graph neural networks (GNNs). We investigated the potential of recommender system algorithms to reveal (or leak) users' sensitive information from the recommendation lists. We found that standard recommender system algorithms leak and that using user attributes as side information during the training of context-aware recommenders may increase this leak. We also found that using user attributes reduces the coverage and lowers the diversity of the recommendation. We provided two countermeasure approaches that are capable of countering privacy leaks. Firstly, we showed that the combination of user attributes as side information might make the inference more difficult. Secondly, we discussed post-processing methods, which modify the recommender system list before being recommended to the users.

To summarize, in Chapter 2 to Chapter 6, we explored different threat models. The common objective of the adversary is to infer or expose sensitive private information

about target users. The difference between the chapters relies on the resources available to the adversary as well as the opportunity for the adversary to make an attack possible. Then, we proposed several purpose-aware privacy-preserving techniques. We focused on two ML applications, machine learning classifiers and recommender systems. Throughout the chapters, we studied how purpose-aware privacy-preserving techniques are connected to protecting individuals' sensitive or personal information within these ML applications.

7.2. FUTURE WORK

In this section, we provide possible future work that we think promising in the context of purpose-aware privacy-preserving data. Several future research directions could extend the contributions proposed in this thesis.

Exploring other threat models The first potential direction involves the exploration of other threat models. The exploration could look at any dimension of the threat model. For instance, future research can look at other sensitive attributes, other adversary resources, and capabilities.

- *Other user sensitive attributes* researchers can investigate other sensitive attributes that we did not study due to the limited availability of open data. Beyond users' demographic attributes such as gender, age, and income, there exist plenty of sensitive attributes that deserve examination. One direction could be to investigate the inference of users' orientation, e.g., sexual, or religious, based on their preferences. Exploring other sensitive attributes can reveal other challenges and opportunities in maintaining accurate predictions while protecting users' private information.

- *Moving beyond black-box attack models* The growing evolution of ML and the availability of vast amounts of data on social media provide the adversary with extra capabilities and resources to perform attacks. With the growth of ML, future work has to adjust and put the focus on more sophisticated gray-box and white-box approaches. In a gray-box attack, the adversary is assumed to have knowledge either about the data used for training a model or about the model itself including which algorithm was used, the parameters, and the architecture. In a white-box attack, the adversary has knowledge about both the training data and the model.

Investigating other purpose-aware privacy-preserving solutions The second potential direction involves the investigation of other purpose-aware privacy-preserving data techniques.

- *Synthetic data for testing and advancing ML applications* Synthetic data is generally intended to take the place of original data. However, in order to take full advantage of synthetic data, we must also invest research efforts in developing the potential of synthetic data to transcend conventional data and be used for purposes for which conventional data is not suited. For instance, one promising application of synthetic data lies in facilitating the development and testing of ML

models. Synthetic data offers an adequate environment where researchers can experiment with data without any constraints. One notable benefit of synthetic data is its utility in identifying and addressing bias in ML algorithms. By simulating diverse scenarios, synthetic data allows researchers to proactively detect and mitigate bias that may emerge when these models interact with real-world data. This approach helps to ensure that ML applications such as recommender systems are fair and unbiased.

- *Protecting the embedding* With the rapid advancements in AI, particularly in the domains of deep learning, generative networks, and Graph Neural Networks (GNNs), we notice an extensive use of embeddings. The embeddings are a way to represent complex data into lower-dimensional representations such that it is easy for machines to understand. However, while being important in various ML applications and domains, the embeddings have also raised concerns about potential leakage of sensitive information [63], [187]. An example of a case study could be that a company is interested in sharing the embeddings with external parties, i.e., researchers, and collaborators. It is important that future research looks at this critical area by exploring potential attacks that may compromise the privacy of individuals through embeddings. Addressing these privacy concerns in the context of ML embeddings is essential to ensure that the advancements in ML are in a trustworthy and responsible manner.

- *Vertically distributed (synthetic) data* refers to data that is distributed across multiple locations or systems, such as different servers, databases, or even among various organizations. Our purpose-aware privacy-preserving framework can also be extended to vertically distributed data. This approach could be employed in scenarios where data sharing or collaboration between different parties or organizations is necessary while maintaining privacy or confidentiality for specific attributes. For instance, consider a collaboration between different organizations, such as the police and the Office of Statistics, aiming to predict whether there is a correlation between poverty and crime. While the data is distributed between these organizations, the ultimate goal is to generate synthetic data that protects individuals' sensitive information while preserving its intended purpose. In this context, the primary objective is to make accurate predictions that enable drawing correct conclusions, just as if the model were trained on real data. There are two approaches for the vertically distributed synthetic data generation, : (1) locally distributed synthetic data generation, (2) globally distributed synthetic data generation. As for the locally distributed synthetic data generation, the synthesis happens locally and separately in each organization. As for the globally distributed synthetic data generation, the synthesis happens in a trusted server.

Quantifying the risk of disclosure While working on this thesis and moving forward, we noticed a gap in the literature on privacy-preserving techniques. Existing research on privacy-preserving techniques has extensively focused on re-identification attacks, but there is less literature regarding attribute inference attacks (attribute disclosure) [156], [172], [173].

To measure the success of inference attacks, we currently rely on two approaches. The first involves using machine learning algorithms to quantify the attribute disclosure risk. However, the effectiveness of this approach heavily depends on both the data and choice of machine learning algorithm. The second approach is from statistical disclosure control, which is based on matching records in the real data with records in the synthetic data using the correct attribution probability (CAP). CAP score describes the proportion of matches leading to correct attribution out of total matches [154], [155]. However, CAP has its limitations, particularly when no matching exists between records in real and synthetic data, leading to a situation where CAP treats the disclosure risk as zero or undefined [155]. In both cases, we encounter challenges in providing a quantified risk of disclosure for individual cases.

Future research should explore other methods to quantify the risk of disclosure on an individual basis. This includes moving beyond aggregated scores that provide generalized assessments of disclosure risk, aiming instead for a more precise evaluation of attribute disclosure risks for each individual. Such quantification of risk for individuals would not only advance the technical understanding but also bridge the gap and facilitate the communication between technical and non-technical communities, including legal and privacy officers.

Other ML applications Throughout this thesis, our primary focus has centered on machine learning focusing on classifiers and recommender system algorithms, including standard collaborative filtering techniques, context-aware recommenders, and graph-neural network recommenders that incorporate user attributes as side information. We showed that as part of our purpose-aware privacy-preserving data, we always look at protecting users' private data while maintaining the intended purpose. Importantly, this framework is not restricted solely to machine learning classifiers and recommender system algorithms. It holds the potential for extension and application to a broader spectrum of machine learning algorithms and diverse recommender system techniques. Beyond this, the applicability of our framework could be further extended to various other ML domains such as lifestyle (e.g., facial recognition), education (e.g., personalized learning), healthcare, and more.

7.3. REFLECTIONS

In this section, we share our reflections and provide our perspectives and insights on various facets of our framework of purpose-aware privacy-preserving data.

Reproducibility and Reliable evaluation Throughout our chapters, we have emphasized the need to adapt our evaluation setup to ensure the reliability of our results and the reproducibility of experiments. Since the early age of machine learning and recommender systems, researchers have pointed to the importance of completely controlling the dimensions of an evaluation in order to achieve a fair comparison [129], [224]. This is not new and has been reported in previous research [219].

Here, we point to two important challenges for achieving reproducibility in published research, namely, the non-availability of data and the rigorousness of the eval-

uation setup. Addressing the first challenge, a potential solution to mitigate this issue could be via the use of synthetic data. It is worth noting that synthetic data serves various purposes including data release, education, testing algorithms, and testing technologies [225]. The generation of synthetic data heavily depends on the use case. We should provide high privacy protection to the synthetic data before being released while trying to maintain the utility. The second challenge in reproducibility is related to adopting a rigorous evaluation setup. An ideal evaluation setup involves providing the readers with all the information related to: the splitting strategy, the candidate items selection in recommender system, end-to-end optimization of hyper-parameter tuning, the choice of strong baseline to compare against, and the choice of the metrics. In addition to these design choices in the evaluation setup, extra attention should be given to develop evaluation frameworks that are suitable for use in evaluating ML models when using synthetic data. In this case, evaluating the evaluation itself must be an object of research. Such an evaluation involves comparing the performance metrics of predictive models trained on synthetic and real data (called model compatibility). The performance of machine learning models trained and tested on real and or synthetic data is compared based on different scenarios depending on the synthetic data use case [158], [226], [227]: Train on Real and Test on Synthetic data ($\mathbb{TRTS}$) Train on Synthetic and Test on Real ($\mathbb{TSTR}$), Train on Real, Test on Real ($\mathbb{TRTR}$) and Train on Synthetic, Test on Synthetic ($\mathbb{TSTS}$), and lastly trained and tested on a mixture of real and synthetic data ($\mathbb{TMTM}$). To sum-up, specific attention should be paid to our evaluation setup and more specifically to the conclusion that we can (and cannot) draw when using synthetic data.

User-oriented paradigm for privacy protection aims at making privacy solutions understandable, unobtrusive, and useful for the user. The idea of our paradigm is that privacy protection should center on users, serving their needs and allowing them to maintain insight and control. The *understandable* dimension of our paradigm expresses the importance that our paradigm places on approaches that the user can understand. The user must understand why a privacy-preserving solution is applied, i.e., why items have been added to or removed from the profile in obfuscation. The *unobtrusive* characteristic expresses the commitment of our paradigm to approaches that do not hamper or otherwise inconvenience or disturb the user. In other words, our purpose-aware privacy-preserving techniques should not be pure "noise", but rather be consistent with the user's preferences. Finally, the *useful* dimension expresses the commitment of our paradigm to serving users' needs such as maintaining or improving the recommendation performance, fairness, and diversity.

Privacy for all Privacy and fairness often appear as trade-offs in various ML applications. Achieving one may come at the expense of the other, leading to searching for the right balance between protecting users' private information and ensuring fair treatment. It is important to make fair predictions as well as to protect users' private data. So, it is also important to make fair protection in which we make sure that all users are protected. In this case, the privacy protection should adapt to individual needs. As mentioned in our user-oriented privacy paradigm, privacy protections should be understandable to users, allowing them to actively participate in the process by choosing what to hide and

what to share. Taking recommender systems as an example, user profiles exhibit variations; while some user profiles may share certain attributes like watching the same genres of movies or listening to the same genre of music. The users' preferences differ and evolve over time. Additionally, user profiles vary in size. A one-size-fits-all privacy budget applied uniformly to all users can lead to the unintentional removal of valuable information essential for recommender system algorithms. Consequently, this approach may provide robust protection to some users while affording minimal protection to others.

Evaluating protection levels becomes crucial in this context. Having 80% protection to user X and only 10% protection to user Y does not signify comprehensive user protection. Instead, it reveals disparities in the levels of protection, potentially leaving the majority of users inadequately protected.

Rethinking the Trade-off Paradigm Traditionally, privacy and utility have been seen as opposing forces, pulling organizations and researchers in different directions. Similarly, the notions of fairness/diversity and accuracy or transparency and privacy are often seen as trade-offs in the literature. The common agreement has been that to improve one, we must compromise the other. For instance, in the context of recommender systems, the more personalized the recommendations, the higher the demand to collect and use the users' private data. Also, in machine learning models, more accurate predictions often necessitate the use of more sensitive data. This trade-off mindset has led to suboptimal solutions where one part is often sacrificed to achieve the second part (privacy vs. accuracy of the predictions, accuracy of the recommendations vs. diversity or fairness, fairness vs. privacy). However, we believe that it's time to question this trade-off paradigm: Why should we accept the premise that to gain one, we must lose the other? Why not looking for a solution or agreement that could satisfy all parties?

Our purpose-aware data framework challenges the conventional trade-off by introducing a more nuanced and purpose-dependent approach. For instance, instead of applying a uniform level of privacy protection to all data, we advocate for tailoring data protection measures to the specific purpose of data usage. This approach recognizes that different machine learning applications have distinct objectives and, therefore, different privacy requirements. In practice, it means that we don't need to choose between privacy and utility. we can have both. Throughout our contributions, we showed that we can provide personalized recommendations while ensuring that users' sensitive information remains protected. We can ensure users' protection while achieving diverse recommendations and without impacting the users' fairness. In machine learning models, we showed that we can make accurate predictions without compromising the privacy of individuals whose data contributes to the model's training, as well as for exclusive individuals (not part of the training data).

We rather call to go from *trade-offs* to *synergy*. For example, by aligning privacy protection with the intended purpose, we can create synergy between the two objectives. Our paradigm of "purpose-aware data" encourages future research to: First, we should prioritize individual needs by recognizing that individuals have varying preferences and by involving users in the protection process. Second, we should optimize data use by tailoring data protection (or other goals such as fairness, diversity, and explainability) to the specific purpose of data usage. As a result, we aim to build trust with users, stakeholders,

and the public.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. Tufail, H. Riggs, M. Tariq, and A. I. Sarwat, "Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms," *Electronics*, vol. 12, no. 8, 2023, ISSN: 2079-9292.

[2] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Computing Surveys*, vol. 56, no. 4, 2023, ISSN: 0360-0300.

[3] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *31st IEEE Computer Security Foundations Symposium*, 2018, pp. 268–282.

[4] V. Torra, "Privacy models and disclosure risk measures," in *Data Privacy: Foundations, New Developments and the Big Data Challenge*. Springer International Publishing, 2017, pp. 111–189, ISBN: 978-3-319-57358-8.

[5] S. Garfinkel *et al.*, *De-identification of Personal Information:*. US Department of Commerce, National Institute of Standards and Technology, 2015.

[6] A. Andreou, O. Goga, and P. Loiseau, "Identity vs. attribute disclosure risks for users with multiple social profiles," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM, Sydney, Australia, 2017, pp. 163–170.

[7] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE Symposium on Security and Privacy*, 2008, pp. 111–125.

[8] G. Beigi and H. Liu, "A survey on privacy in social media: Identification, mitigation, and applications," *ACM/IMS Transaction on Data Science*, vol. 1, no. 1, 2020, ISSN: 2691-1922.

[9] N. Z. Gong and B. Liu, "Attribute inference attacks in online social networks," *ACM Transactions on Privacy and Security*, vol. 21, no. 1, 2018, ISSN: 2471-2566.

[10] S. Salamatian, A. Zhang, F. d. P. Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "How to hide the elephant- or the donkey- in the room: Practical privacy against statistical inference for large data," in *Global Conference on Signal and Information Processing*, 2013, pp. 269–272.

[11] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy*, 2017, pp. 3–18.

[12] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM International Conference on Computer and Communications Security*, Denver, Colorado, USA, 2015, pp. 1322–1333.

[13] S. Mehnaz, S. V. Dibbo, E. Kabir, N. Li, and E. Bertino, "Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models," in *31st USENIX Security Symposium*, Boston, MA: USENIX Association, 2022, pp. 4579–4596.

[14] V. Torra, "Privacy in data mining," in *Data Mining and Knowledge Discovery Handbook*, Springer, 2009, pp. 687–716.

[15] M. Templ, *Statistical disclosure control for microdata: methods and applications in R*. Cham: Springer, 2017.

[16] G. Garofalo, M. Slokom, D. Preuveneers, W. Joosen, and M. Larson, "Machine learning meets data modification," in *Security and Artificial Intelligence: A Cross-disciplinary Approach*, L. Batina, T. Bäck, I. Buhan, and S. Picek, Eds. Cham: Springer International Publishing, 2022, pp. 130–155, ISBN: 978-3-030-98795-4.

[17] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[18] F. Casino, J. Domingo-Ferrer, C. Patsakis, D. Puig, and A. Solanas, "A k-anonymous approach to privacy preserving collaborative filtering," *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 1000–1011, 2015.

[19] A. Friedman, B. P. Knijnenburg, K. Vanhecke, L. Martens, and S. Berkovsky, "Privacy aspects of recommender systems," in *Recommender Systems Handbook*, Springer, 2015, pp. 649–688.

[20] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *3rd IEEE International Conference on Data Mining*, 2003, pp. 625–628.

[21] R. Parameswaran and D. M. Blough, "Privacy preserving collaborative filtering using data obfuscation," in *IEEE International Conference on Granular Computing*, 2007, pp. 380–380.

[22] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "Blurme: Inferring and obfuscating user gender based on ratings," in *Proceedings of the 6th ACM International Conferencerence on Recommender systems*, 2012, pp. 195–202.

[23] S. Berkovsky, T. Kuflik, and F. Ricci, "The impact of data obfuscation on the accuracy of collaborative filtering," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5033–5042, 2012.

[24] J. Parra-Arnau, D. Rebollo-Monedero, and J. Forné, "A privacy-protecting architecture for collaborative filtering via forgery and suppression of ratings," in *Data Privacy Management and Autonomous Spontaneus Security*, Springer, 2012, pp. 42–57.

[25] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. S. Nordholt, K. Spicer, and P.-P. De Wolf, *Statistical disclosure control*. John Wiley & Sons, 2012.

[26] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[27] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 627–636, ISBN: 9781605584959.

[28] T. Zhu, Y. Ren, W. Zhou, J. Rong, and P. Xiong, "An effective privacy preserving algorithm for neighborhood-based collaborative filtering," *Future Generation Computer Systems*, vol. 36, pp. 142–155, 2014.

[29] R. Wei, H. Tian, and H. Shen, "Improving k-anonymity based privacy preservation for collaborative filtering," *Computers & Electrical Engineering*, 2018, ISSN: 0045-7906.

[30] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, "Privacy preserving synthetic data release using deep learning," in *Machine Learning and Knowledge Discovery in Databases*, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, Eds., Springer International Publishing, 2019, pp. 510–526.

[31] H. Li, L. Xiong, L. Zhang, and X. Jiang, "DPSynthesizer: Differentially private data synthesizer for privacy preserving data sharing," in *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, NIH Public Access, vol. 7, 2014, p. 1677.

[32] J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "The limits of differential privacy (and its misuse in data release and machine learning)," *Communications of the ACM*, vol. 64, no. 7, pp. 33–35, 2021.

[33] J. Canny, "Collaborative filtering with privacy," in *IEEE Proceedings Symposium on Security and Privacy*, 2002, pp. 45–57.

[34] Z. Erkin, M. Beye, T. Veugen, and R. L. Lagendijk, "Privacy enhanced recommender system," in *31st symposium on information theory in the Benelux*, 2010, pp. 35–42.

[35] Z. Erkin, T. Veugen, and R. L. Lagendijk, "Generating private recommendations in a social trust network," in *IEEE International Conference on Computational Aspects of Social Networks (CASoN)*, 2011, pp. 82–87.

[36] A. Basu, J. Vaidya, H. Kikuchi, and T. Dimitrakos, "Privacy-preserving collaborative filtering on the cloud and practical implementation experiences," in *IEEE 6th International Conference on Cloud Computing (CLOUD)*, 2013, pp. 406–413.

[37] J. Drechsler, *Synthetic datasets for statistical disclosure control: theory and implementation*. Springer Science & Business Media, 2011, vol. 201.

[38] J. Drechsler and J. P. Reiter, "An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets," *Computational Statistics & Data Analysis*, vol. 55, no. 12, pp. 3232–3243, 2011.

[39] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *IEEE International Conference on Data Science and Advanced Analytics*, 2016, pp. 399–410.

[40] J. Drechsler, S. Bender, and S. Rässler, "Comparing fully and partially synthetic datasets for statistical disclosure control in the German IAB establishment panel.," *Transaction of Data Privacy*, vol. 1, no. 3, pp. 105–130, 2008.

[41] R. A. Dandekar, M. Cohen, and N. Kirkendall, "Sensitive micro data protection using latin hypercube sampling technique," in *Inference Control in Statistical Databases*, Springer, 2002, pp. 117–125.

[42] T. Carvalho, N. Moniz, P. Faria, and L. Antunes, "Survey on privacy-preserving techniques for microdata publication," *ACM Computing Survey*, vol. 55, no. 14s, 2023, ISSN: 0360-0300.

[43] D. Zhuang and J. M. Chang, "Utility-aware privacy-preserving data releasing," *arXiv preprint arXiv:2005.04369*, 2020.

[44] K. Chen and L. Liu, "A survey of multiplicative perturbation for privacy-preserving data mining," in *Privacy-Preserving Data Mining: Models and Algorithms*, C. C. Aggarwal and P. S. Yu, Eds. Springer US, 2008, pp. 157–181, ISBN: 978-0-387-70992-5.

[45] B. K. Pandya, U. K. Singh, and K. Dixit, "A robust privacy preservation by combination of additive and multiplicative data perturbation for privacy preserving data mining," *International Journal of Computer Applications*, vol. 120, no. 1, 2015.

[46] M. A. P. Chamikara, P. Bertók, I. Khalil, D. Liu, and S. Camtepe, "Ppaas: Privacy preservation as a service," *Computer Communications*, vol. 173, pp. 192–205, 2021.

[47] X. Li, G. Wu, L. Yao, Z. Zheng, and S. Geng, "Utility-aware privacy perturbation for training data," *ACM Transactions on Knowledge Discovery from Data*, 2024, ISSN: 1556-4681.

[48] C. Salter, O. S. Saydjari, B. Schneier, and J. Wallner, "Toward a secure system engineering methodology," in *Proceedings of the Workshop on New Security Paradigms*, ser. NSPW, 1998, pp. 2–10.

[49] X. Zhang, C. Chen, Y. Xie, X. Chen, J. Zhang, and Y. Xiang, "A survey on privacy inference attacks and defenses in cloud-based deep neural network," *Computer Standards & Interfaces*, vol. 83, p. 103 672, 2023, ISSN: 0920-5489.

[50] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka, "Exposing private user behaviors of collaborative filtering via model inversion techniques," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 3, pp. 264–283, 2020.

[51] C. Strucks, M. Slokom, and M. Larson, "BlurM (or) e: Revisiting gender obfuscation in the user-item matrix," in *Recommendation in Multi-stakeholder Environments (RMSE), in conjunction with the 13th ACM International Conferencerence on Recommender Systems*, 2019.

[52] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An End-to-End case study of personalized warfarin dosing," in *23rd USENIX Security Symposium*, San Diego, CA: USENIX Association, 2014, pp. 17–32.

[53] M. Kahla, S. Chen, H. A. Just, and R. Jia, "Label-only model inversion attacks via boundary repulsion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 045–15 053.

[54] K.-C. Wang, Y. Fu, K. Li, A. Khisti, R. Zemel, and A. Makhzani, "Variational model inversion attacks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9706–9719, 2021.

[55] F. Brunton and H. Nissenbaum, *Obfuscation: A user's guide for privacy and protest*. MIT Press, 2015.

[56] M. Slokom, A. Hanjalic, and M. Larson, "Towards user-oriented privacy for recommender system data: A personalization-based approach to gender obfuscation for user profiles," *Information Processing & Management*, vol. 58, no. 6, 2021, ISSN: 0306-4573.

[57] K. Muralidhar and R. Sarathy, "Data shuffling: A new masking approach for numerical data," *Management Science*, vol. 52, no. 5, pp. 658–670, 2006.

[58] T. Dalenius and S. P. Reiss, "Data-swapping: A technique for disclosure control," *Journal of statistical planning and inference*, vol. 6, no. 1, pp. 73–85, 1982.

[59] J. P. Reiter, "Synthetic data: A look back and a look forward.," *Transaction on Data Privacy*, vol. 16, no. 1, pp. 15–24, 2023.

[60] J. Drechsler and A.-C. Haensch, "30 years of synthetic data," *Statistical Science*, vol. 39, no. 2, pp. 221–242, 2024.

[61] J. Burger, B. Buelens, T. de Jong, and Y. Gootzen, "Replacing a survey question by predictive modeling using register data," *ISI World Statistics Congress*, pp. 1–6, 2019.

[62] D. Yang, B. Qu, and P. Cudré-Mauroux, "Privacy-preserving social media data publishing for personalized ranking-based recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 507–520, 2019.

[63] G. Beigi, A. Mosallanezhad, R. Guo, H. Alvari, A. Nou, and H. Liu, "Privacy-aware recommendation with private-attribute protection using adversarial learning," in *Proceedings of the 13th ACM International Conference on Web Search and Data Mining*, Houston, TX, USA, 2020, pp. 34–42, ISBN: 9781450368223.

[64] T. Chen, R. Boreli, M.-A. Kaafar, and A. Friedman, "On the effectiveness of obfuscation techniques in online social networks," in *International Symposium on Privacy Enhancing Technologies Symposium*, Springer, 2014, pp. 42–62.

[65] A. Friedman, S. Berkovsky, and M. A. Kaafar, "A differential privacy framework for matrix factorization recommender systems," *User Modeling and User-Adapted Interaction*, vol. 26, no. 5, pp. 425–458, 2016.

[66] A. Friedman, B. P. Knijnenburg, K. Vanhecke, L. Martens, and S. Berkovsky, "Privacy aspects of recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA: Springer US, 2015, pp. 649–688, ISBN: 978-1-4899-7637-6.

[67] V. W. Anelli, Y. Deldjoo, T. Di Noia, A. Ferrara, and F. Narducci, "Federank: User controlled feedback with federated recommender systems," in *Advances in Information Retrieval*, D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, and F. Sebastiani, Eds., Cham: Springer International Publishing, 2021, pp. 32–47.

[68] Y. Qiang, "Federated recommendation systems," in *IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1–1.

[69] S. Badsha, X. Yi, I. Khalil, and E. Bertino, "Privacy preserving user-based recommender system," in *IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 1074–1083.

[70] A. J. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. L. Lagendijk, and Q. Tang, "Privacy in recommender systems," in *Social media retrieval*, Springer, 2013, pp. 263–281.

[71] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds., Springer Berlin Heidelberg, 2008, pp. 1–19, ISBN: 978-3-540-79228-4.

[72] J. Hua, C. Xia, and S. Zhong, "Differentially private matrix factorization," in *Proceedings of the 24th International Conference on Artificial Intelligence*, Buenos Aires, Argentina: AAAI Press, 2015, pp. 1763–1770.

[73] J. Parra-Arnau, D. Rebollo-Monedero, and J. Forné, "Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems," *Entropy*, vol. 16, no. 3, pp. 1586–1631, 2014.

[74] T. Kandappu, A. Friedman, R. Boreli, and V. Sivaraman, "PrivacyCanary: Privacy-aware recommenders with adaptive input obfuscation," in *22nd IEEE International Symposium on Modelling, Analysis Simulation of Computer and Telecommunication Systems*, 2014, pp. 453–462.

[75] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transaction Internet Technology*, vol. 7, no. 4, 23–es, 2007, ISSN: 1533-5399.

[76] R. Burke, M. P. O'Mahony, and N. J. Hurley, "Robust collaborative recommendation," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Springer US, 2015, pp. 961–995, ISBN: 978-1-4899-7637-6.

[77] Y. Deldjoo, T. D. Noia, and F. A. Merra, "A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks," *ACM Computing Survey*, vol. 54, no. 2, 2021, ISSN: 0360-0300.

[78] S. Badsha, X. Yi, and I. Khalil, "A practical privacy-preserving recommender system," *Data Science and Engineering*, vol. 1, no. 3, pp. 161–177, 2016.

[79] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proceedings of the International Conference on Computer & Communications Security*, 2013, pp. 801–812, ISBN: 9781450324779.

[80] M. Slokom, M. Larson, and A. Hanjalic, "Data masking for recommender systems: Prediction performance and rating hiding," *Late breaking results, in conjunction with the 13th ACM International Conferencerence on Recommender Systems*, 2019.

[81] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, ""you might also like:" privacy risks of collaborative filtering," in *IEEE Symposium on Security and Privacy*, 2011, pp. 231–246.

[82] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5802–5805, 2013.

[83] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. Di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *IFIP Annual Conference on Data and Applications Security and Privacy*, Springer, 2007, pp. 47–60.

[84] Y. Li, N. Vishwamitra, B. P. Knijnenburg, H. Hu, and K. Caine, "Effectiveness and users' experience of obfuscation as a privacy-enhancing technology for sharing photos," *Proceeding of the ACM Human-Computing Interaction*, vol. 1, 2017.

[85] S. Reddy and K. Knight, "Obfuscating gender in social media writing," in *Proceedings of the 2016 EMNLP Workshop on NLP and Computational Social Science*, ACL, 2016, pp. 17–26.

[86] T. Feng, Y. Guo, and Y. Chen, "Can user privacy and recommendation performance be preserved simultaneously?" *Computer Communications*, vol. 68, pp. 17–24, 2015.

[87] Y. S. Resheff, Y. Elazar, M. Shahar, and O. S. Shalom, "Privacy and fairness in recommender systems via adversarial training of user representations," *arXiv preprint arXiv:1807.03521*, 2018.

[88] G. Hu and Q. Yang, "PrivNet: Safeguarding private attributes in transfer learning for recommendation," in *Findings of the Association for Computational Linguistics: EMNLP*, 2020, pp. 4506–4516.

[89] S. Salamatian, A. Zhang, F. du Pin Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "Managing your private and public data: Bringing down inference attacks against your privacy," *Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1240–1255, 2015.

[90] T. Feng, Y. Guo, and Y. Chen, "Can user gender and recommendation performance be preserved simultaneously?" In *International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2015, pp. 227–231.

[91] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: Inferring user profiles in online social networks," in *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, 2010, pp. 251–260, ISBN: 9781605588896.

[92] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "AttriInfer: Inferring user attributes in online social networks using markov random fields," Perth, Australia: ACM International World Wide Web Conferences, 2017, pp. 1561–1569, ISBN: 9781450349130.

[93] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel, "Inferring the demographics of search users: Social data meets search queries," in *Proceedings of the 22nd ACM International Conference on World Wide Web*, Rio de Janeiro, Brazil, 2013, pp. 131–140, ISBN: 9781450320351.

[94] S. Bhagat, U. Weinsberg, S. Ioannidis, and N. Taft, "Recommending with an agenda: Active learning of private attributes using matrix factorization," in *Proceedings of the 8th ACM International Conferencerence on Recommender Systems*, Foster City, USA, 2014, pp. 65–72, ISBN: 9781450326681.

[95] T. Feng, Y. Guo, Y. Chen, X. Tan, T. Xu, B. Shen, and W. Zhu, "Tags and titles of videos you watched tell your gender," in *IEEE International Conference on Communications (ICC)*, 2014, pp. 1837–1842.

[96] S. A. Friedler, C. Scheidegger, and S. Venkatasubramanian, "The (im)possibility of fairness: Different value systems require different mechanisms for fair decision making," *Communication of the ACM*, vol. 64, no. 4, pp. 136–143, 2021, ISSN: 0001-0782.

[97] S. Yao and B. Huang, "Beyond parity: Fairness objectives for collaborative filtering," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 2925–2934.

[98] R. Burke, N. Sonboli, and A. Ordonez-Gauger, "Balanced neighborhoods for multi-sided fairness in recommendation," in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, ser. Proceedings of Machine Learning Research, vol. 81, 2018, pp. 202–214.

[99] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, and L. Pizzato, "Beyond personalization: Research directions in multistakeholder recommendation," *arXiv preprint arXiv:1905.01986*, 2019.

[100] M. D. Ekstrand, R. Joshaghani, and H. Mehrpouyan, "Privacy for all: Ensuring fair and equitable privacy protections," in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, ser. Proceedings of Machine Learning Research, vol. 81, 2018, pp. 35–47.

[101] A. Ferraro, X. Serra, and C. Bauer, "Break the loop: Gender imbalance in music recommenders," in *Proceedings of the International Conference on Human Information Interaction and Retrieval*, Canberra ACT, Australia, 2021, pp. 249–254, ISBN: 9781450380553.

[102] R. Burke, "Multisided fairness for recommendation," in *FATREC 2017 Workshop on Fairness, Accountability, and Transparency in Recommender Systems, in conjunction with the 11th ACM International Conferencerence on Recommender Systems*, 2017.

[103] M. D. Ekstrand, M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, and M. S. Pera, "All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness," in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, ser. Proceedings of Machine Learning Research, vol. 81, 2018, pp. 172–186.

[104] M. Mansoury, H. Abdollahpouri, J. Smith, A. Dehpanah, M. Pechenizkiy, and B. Mobasher, "Investigating potential factors associated with gender discrimination in collaborative recommender systems," in *Proceedings of the 13th FLAIRS Conference*, Miami, FL, USA, 2020.

[105] M. D. Ekstrand and D. Kluver, "Exploring author gender in book rating and recommendation," *User Modeling and User-Adapted Interaction*, pp. 1–44, 2021.

[106] M. D. Ekstrand, M. Tian, M. R. I. Kazi, H. Mehrpouyan, and D. Kluver, "Exploring author gender in book rating and recommendation," in *Proceedings of the 12th ACM International Conferencerence on Recommender Systems*, Vancouver, British Columbia, Canada, 2018, pp. 242–250, ISBN: 978-1-4503-5901-6.

[107] D. Shakespeare, L. Porcaro, E. Gómez, and C. Castillo, "Exploring artist gender bias in music recommendation," in *ImpactRS Workshop in conjunction with the 14th ACM International Conferencerence on Recommender Systems*, 2020.

[108] A. Epps-Darling, R. T. Bouyer, and H. Cramer, "Artist gender representation in music streaming," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[109] P. Castells, N. J. Hurley, and S. Vargas, "Novelty and diversity in recommender systems," in *Recommender systems handbook*, Springer, 2015, pp. 881–918.

[110] M. Kaminskas and D. Bridge, "Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems," *ACM Transactions on Interactive Intelligent Systems*, vol. 7, no. 1, 2016, ISSN: 2160-6455.

[111] M. Kunaver and T. Požrl, "Diversity in recommender systems – a survey," *Knowledge-Based Systems*, vol. 123, pp. 154–162, 2017, ISSN: 0950-7051.

[112] C. Hansen, R. Mehrotra, C. Hansen, B. Brost, L. Maystre, and M. Lalmas, "Shifting consumption towards diverse content on music streaming platforms," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, Virtual Event, Israel, 2021, pp. 238–246, ISBN: 9781450382977.

[113] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the 5th ACM International Conferencerence on Recommender Systems*, Chicago, Illinois, USA, 2011, pp. 109–116, ISBN: 9781450306836.

[114] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke, "Fairmatch: A graph-based approach for improving aggregate diversity in recommender systems," in *Proceedings of the 28th ACM International Conferencerence on User Modeling, Adaptation and Personalization*, Genoa, Italy, 2020, pp. 154–162, ISBN: 9781450368612.

[115] R. S. Oliveira, C. Nóbrega, L. B. Marinho, and N. Andrade, "A multiobjective music recommendation approach for aspect-based diversification.," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 414–420.

[116] N. Helberger, K. Karppinen, and L. D'Acunto, "Exposure diversity as a design principle for recommender systems," *Information, Communication & Society*, vol. 21, no. 2, pp. 191–207, 2018.

[117] K. Lakshminarayan, S. A. Harp, and T. Samad, "Imputation of missing data in industrial databases," *Applied intelligence*, vol. 11, no. 3, pp. 259–275, 1999.

[118] D. Bertsimas, C. Pawlowski, and Y. D. Zhuo, "From predictive methods to missing data imputation: An optimization approach," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 7133–7171, 2017.

[119] X. Su, T. M. Khoshgoftaar, and R. Greiner, "Imputed neighborhood based collaborative filtering," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1, 2008, pp. 633–639.

[120] X. Yuan, L. Han, S. Qian, G. Xu, and H. Yan, "Singular value decomposition based recommendation using imputed data," *Knowledge-Based Systems*, vol. 163, pp. 485–494, 2019.

[121] Q. Li, X. Zheng, and X. Wu, "Collaborative autoencoder for recommender systems," *ArXiv e-prints*, 2017.

[122] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, San Francisco, California, USA, 2016, pp. 153–162.

[123] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, 2015, ISSN: 2160-6455.

[124] R. Zafarani and H. Liu, *Social computing data repository at ASU*, 2009.

[125] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011.

[126] I. Pilászy, D. Zibriczky, and D. Tikk, "Fast als-based matrix factorization for explicit and implicit feedback datasets," in *Proceedings of the 4th ACM International Conferencerence on Recommender Systems*, Barcelona, Spain, 2010, pp. 71–78.

[127] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, Montreal, Quebec, Canada: AUAI Press, 2009, pp. 452–461.

[128] M. D. Ekstrand, "Lenskit for python: Next-generation software for recommender systems experiments," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, Virtual Event, Ireland, 2020, pp. 2999–3006, ISBN: 9781450368599.

[129] A. Bellogin, P. Castells, and I. Cantador, "Precision-oriented evaluation of recommender systems: An algorithmic comparison," in *Proceedings of the 5th ACM International Conferencerence on Recommender Systems*, Chicago, Illinois, USA, 2011, pp. 333–336.

[130] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the 4th ACM International Conferencerence on Recommender Systems*, Barcelona, Spain, 2010, pp. 39–46, ISBN: 9781605589060.

[131] V. Torra, "Masking methods," in *Data Privacy: Foundations, New Developments and the Big Data Challenge*, Springer, 2017, pp. 191–238.

[132] B. C. Fung, K. Wang, A. W.-C. Fu, and S. Y. Philip, *Introduction to privacy-preserving data publishing: Concepts and techniques*. Chapman and Hall/CRC, 2010.

[133] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *ACM Sigmod Record*, vol. 29, 2000, pp. 439–450.

[134] W. Guo, S. Wu, L. Wang, and T. Tan, "Personalized ranking with pairwise factorization machines," *Neurocomputing*, vol. 214, no. C, pp. 191–200, 2016.

[135] B. Loni, R. Pagano, M. Larson, and A. Hanjalic, "Top-n recommendation with multi-channel positive feedback using factorization machines," *ACM Transaction on Information Systems*, vol. 37, no. 2, 15:1–15:23, 2019.

[136] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, Hong Kong: Association for Computing Machinery, 2001, pp. 285–295.

[137] D. Lemire and A. Maclachlan, "Slope one predictors for online rating-based collaborative filtering," in *Proceedings of the SIAM International Conference on Data Mining*, 2005, pp. 471–475.

[138] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *5th IEEE international conference on Data Mining*, 2005, pp. 625–628.

[139] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[140] Y. 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*, 2008, pp. 426–434.

[141] B. Loni and A. Said, "WrapRec: An easy extension of recommender system libraries," in *Proceedings of the 8th ACM International Conferencerence on Recommender Systems*, California, USA, 2014, pp. 377–378, ISBN: 978-1-4503-2668-1.

[142] E. Bertino, I. N. Fovino, and L. P. Provenza, "A framework for evaluating privacy preserving data mining algorithms," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 121–154, 2005.

[143] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Computing Survey*, vol. 54, no. 2, 2021.

[144] S. R. Crull, "Residential satisfaction, propensity to move, and residential mobility: A causal model," in *Digital Repository at Iowa State University*, `http://lib.dr.iastate.edu/`, 1979.

[145] R. Kleinhans, "Does social capital affect residents' propensity to move from restructured neighbourhoods?" *Housing Studies*, vol. 24, no. 5, pp. 629–651, 2009.

[146] D. Fackler and L. Rippe, "Losing work, moving away? regional mobility after job loss," *Labour*, vol. 31, no. 4, pp. 457–479, 2017.

[147] R. Coulter and J. Scott, "What motivates residential mobility? re-examining self-reported reasons for desiring and making residential moves," *Population, Space and Place*, vol. 21, no. 4, pp. 354–371, 2015.

[148] P. A. de Jong, "Later-life migration in the Netherlands: Propensity to move and residential mobility," *Journal of Aging and Environment*, pp. 1–10, 2020.

[149] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the ACM International Conference on Computer and Communications Security*, 2016, pp. 308–318.

[150] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *ACM Computing Survey*, 2023, ISSN: 0360-0300.

[151] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka, "Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes," in *15th IEEE Annual Conference on Privacy, Security and Trust (PST)*, 2017, pp. 115–11 509.

[152] J. P. Reiter and R. Mitra, "Estimating risks of identification disclosure in partially synthetic data," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, 2009.

[153] J. P. Reiter, Q. Wang, and B. Zhang, "Bayesian estimation of disclosure risks for multiply imputed, synthetic data," *Journal of Privacy and Confidentiality*, vol. 6, no. 1, 2014.

[154] Mark Elliot. "'Final report on the disclosure risk associated with synthetic data produced by the SYLLS Team." `http://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/reports/`, Online; Last accessed 26-June-2022. (2014).

[155] J. Taub, M. Elliot, M. Pampaka, and D. Smith, "Differential correct attribution probability for synthetic data: An exploration," in *Privacy in Statistical Databases*, J. Domingo-Ferrer and F. Montes, Eds., Springer International Publishing, 2018, pp. 122–137, ISBN: 978-3-319-99771-1.

[156] T. Stadler, B. Oprisanu, and C. Troncoso, "Synthetic data–anonymisation groundhog day," in *29th USENIX Security Symposium*, USENIX Association, 2020.

[157] M. Hittmeir, R. Mayer, and A. Ekelhart, "A baseline for attribute disclosure risk in synthetic data," in *Proceedings of the 10th ACM International Conferencerence on Data and Application Security and Privacy*, 2020, pp. 133–143.

[158] R. Heyburn, R. R. Bond, M. Black, M. Mulvenna, J. Wallace, D. Rankin, and B. Cleland, "Machine learning using synthetic and real data: Similarity of evaluation metrics for different healthcare datasets and for different algorithms," in *Data Science and Knowledge Engineering for Sensing Decision Support: Proceedings of the 13th International FLINS Conference*, World Scientific, 2018, pp. 1281–1291.

[159] J. Domingo-Ferrer, "A survey of inference control methods for privacy-preserving data mining," in *Privacy-preserving data mining*, Springer, 2008, pp. 53–80.

[160] M. Slokom, P.-P. de Wolf, and M. Larson, "When machine learning models leak: An exploration of synthetic training data," in *Proceedings of the International Conference on Privacy in Statistical Databases*, Paris, France: Springer-Verlag, 2022, pp. 283–296, ISBN: 978-3-031-13944-4.

[161] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," in *57th IEEE Annual Allerton Conference on Communication, Control, and Computing*, 2019, pp. 495–505.

[162] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche-Buc, E. Fox, and R. Garnett, Eds., 2019, pp. 7335–7345.

[163] C. K. Liew, U. J. Choi, and C. J. Liew, "A data distortion by probability distribution," *ACM Transactions on Database Systems*, vol. 10, no. 3, pp. 395–411, 1985.

[164] D. B. Rubin, "Discussion statistical disclosure limitation," *Journal of official Statistics*, vol. 9, no. 2, pp. 461–468, 1993.

[165] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," F. Doshi-Velez, J. Fackler, D. Kale, R. Ranganath, B. Wallace, and J. Wiens, Eds., ser. Proceedings of Machine Learning Research, vol. 68, PMLR, 2017, pp. 286–305.

[166] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," *Proceedings of the 44th International Conference on Very Large Data Bases (VLDB Endowment)*, vol. 11, no. 10, pp. 1071–1083, 2018.

[167] P.-P. de Wolf, "Risk, utility and PRAM," in *Privacy in Statistical Databases*, J. Domingo-Ferrer and L. Franconi, Eds., Springer Berlin Heidelberg, 2006, pp. 189–204, ISBN: 978-3-540-49332-7.

[168] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via bayesian networks," *ACM Transaction on Database Systems*, vol. 42, no. 4, 2017.

[169] G. M. Raab, "Utility and disclosure risk for differentially private synthetic categorical data," in *Proceedings of International Conference on Privacy in Statistical Databases*, Paris, France: Springer-Verlag, 2022, pp. 250–265, ISBN: 978-3-031-13944-4.

[170] M. L. Fang, D. S. Dhami, and K. Kersting, "DP-CTGAN: Differentially private medical data generation using CTGANs," in *Proceedings of the 20th International Conference on Artificial Intelligence in Medicine, AIME*, Berlin, Heidelberg: Springer-Verlag, 2022, pp. 178–188, ISBN: 978-3-031-09341-8.

[171] M. Sun, C. Li, and H. Zha, "Inferring private demographics of new users in recommender systems," in *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, 2017, pp. 237–244.

[172] N. Shlomo, "How to measure disclosure risk in microdata?" *The Survey Statistician*, vol. 86, no. 2, pp. 13–21, 2022.

[173] P.-H. Lu, P.-C. Wang, and C.-M. Yu, "Empirical evaluation on synthetic data generation with generative adversarial network," in *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*, 2019, pp. 1–6.

[174] B. Jayaraman and D. Evans, "Are attribute inference attacks just imputation?" In *Proceedings of the ACM International Conference on Computer and Communications Security*, Los Angeles, CA, USA, 2022, pp. 1569–1582, ISBN: 9781450394505.

[175] C. Little, M. Elliot, and R. Allmendinger, "Comparing the utility and disclosure risk of synthetic data with samples of microdata," in *Proceedings of the International Conference on Privacy in Statistical Databases*, J. Domingo-Ferrer and M. Laurent, Eds., Paris, France: Springer-Verlag, 2022, pp. 234–249, ISBN: 978-3-031-13944-4.

[176] J. P. Reiter, "Using CART to generate partially synthetic public use microdata," *Journal of Official Statistics*, vol. 21, no. 3, p. 441, 2005.

[177] B. Nowok, G. M. Raab, and C. Dibben, "Synthpop: Bespoke creation of synthetic data in R," *Journal of Statistical Software*, vol. 74, no. 11, pp. 1–26, 2016.

[178] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International journal of pattern recognition and artificial intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[179] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, pp. 1–13, 2020.

[180] R. Burke, "Hybrid web recommender systems," *The adaptive web*, pp. 377–408, 2007.

[181] W. Zhou, J. Li, Y. Yang, and F. Shah, "Leverage side information for top-n recommendation with latent gaussian process," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 12, e5534, 2021.

[182] P. Do and P. Pham, "Heterogeneous graph convolutional network pre-training as side information for improving recommendation," *Neural Computing and Applications*, pp. 1–17, 2022.

[183] J. Wu, X. He, X. Wang, Q. Wang, W. Chen, J. Lian, and X. Xie, "Graph convolution machine for context-aware recommender system," *Frontiers of Computer Science*, vol. 16, no. 6, 2022, ISSN: 2095-2228.

[184] S. Liu, Z. Meng, C. Macdonald, and I. Ounis, "Graph neural pre-training for recommendation with side information," *ACM Transaction Information System*, 2022, ISSN: 1046-8188. DOI: 10.1145/3568953.

[185] K. Haruna, M. Akmar Ismail, S. Suhendroyono, D. Damiasih, A. C. Pierewan, H. Chiroma, and T. Herawan, "Context-aware recommender system: A review of recent developmental process and future research direction," *Applied Sciences*, vol. 7, no. 12, 2017, ISSN: 2076-3417.

[186] Y. Chen and M. de Rijke, "A collective variational autoencoder for top-n recommendation with side information," in *Proceedings of the 3rd workshop on deep learning for recommender systems*, 2018, pp. 3–9.

[187] S. Zhang, H. Yin, T. Chen, Z. Huang, L. Cui, and X. Zhang, "Graph embedding for recommendation against attribute inference attacks," in *Proceedings of the ACM Web Conference*, Ljubljana, Slovenia, 2021, pp. 3002–3014, ISBN: 9781450383127.

[188] X. Xin, J. Yang, H. Wang, J. Ma, P. Ren, H. Luo, X. Shi, Z. Chen, and Z. Ren, "On the user behavior leakage from recommender system exposure," *ACM Transaction on Information Systems*, 2022.

[189] B. Edizel, F. Bonchi, S. Hajian, A. Panisson, and T. Tassa, "Fairecsys: Mitigating algorithmic bias in recommender systems," *International Journal of Data Science and Analytics*, vol. 9, pp. 197–213, 2020.

[190] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1–45, 2014.

[191] G. Adomavicius, K. Bauman, A. Tuzhilin, and M. Unger, "Context-aware recommender systems: From foundations to recent developmentscontext-aware recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Springer US, 2022, pp. 211–250.

[192] S. Rendle, "Factorization machines," in *IEEE International conference on data mining*, 2010, pp. 995–1000.

[193] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[194] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. 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*, 2020, pp. 639–648.

[195] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," in *Proceedings of the AAAI Conference on artificial intelligence*, vol. 31, 2017.

[196] C. Zhao, X. Shi, M. Shang, and Y. Fang, "A clustering-based collaborative filtering recommendation algorithm via deep learning user side information," in *International Conference on Web Information Systems Engineering*, Springer, 2020, pp. 331–342.

[197] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo, J. Zhang, and R. Burke, "Research commentary on recommendations with side information: A survey and research directions," *Electronic Commerce Research and Applications*, vol. 37, 2019, ISSN: 1567-4223.

[198] S. Kulkarni and S. F. Rodd, "Context aware recommendation systems: A review of the state of the art techniques," *Computer Science Review*, vol. 37, p. 100 255, 2020.

[199] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Transactions on Recommender Systems (TORS)*, 2022.

[200] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: Improving geographical prediction with social and spatial proximity," in *Proceedings of the 19th ACM international conference on World wide web*, 2010, pp. 61–70.

[201] D. Jurgens, T. Finethy, J. McCorriston, Y. Xu, and D. Ruths, "Geolocation prediction in twitter using social networks: A critical analysis and review of current practice," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 9, 2015.

[202] R. Shokri, "Quantifying and protecting location privacy," *it-Information Technology*, vol. 57, no. 4, pp. 257–263, 2015.

[203] A. Chaabane, G. Acs, M. A. Kaafar, *et al.*, "You are what you like! information leakage through users? interests," in *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS)*, Citeseer, 2012.

[204] L. Kong, Z. Liu, and Y. Huang, "Spot: Locating social media users based on social network context," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1681–1684, 2014.

[205] P. Castells, N. J. Hurley, and S. Vargas, "Novelty and diversity in recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., Springer US, 2015, pp. 881–918, ISBN: 978-1-4899-7637-6.

[206] M. Kunaver and T. Požrl, "Diversity in recommender systems – a survey," *Knowledge-Based Systems*, vol. 123, pp. 154–162, 2017, ISSN: 0950-7051.

[207] T. Di Noia, J. Rosati, P. Tomeo, and E. D. Sciascio, "Adaptive multi-attribute diversity for recommender systems," *Information Sciences*, vol. 382-383, pp. 234–253, 2017, ISSN: 0020-0255.

[208] A. Steenvoorden, E. Di Gloria, W. Chen, P. Ren, and M. de Rijke, "Attribute-aware diversification for sequential recommendations," *arXiv preprint arXiv:2008.00783*, 2020.

[209] C.-H. Tsai and P. Brusilovsky, "Leveraging interfaces to improve recommendation diversity," in *the 25th Conference on User Modeling, Adaptation and Personalization*, Slovakia, 2017, pp. 65–70.

[210] F. Tramer, V. Atlidakis, R. Geambasu, D. Hsu, J.-P. Hubaux, M. Humbert, A. Juels, and H. Lin, "Fairtest: Discovering unwarranted associations in data-driven applications," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017, pp. 401–416.

[211] Y. R. Shrestha and Y. Yang, "Fairness in algorithmic decision-making: Applications in multi-winner voting, machine learning, and recommender systems," *Algorithms*, vol. 12, no. 9, p. 199, 2019.

[212] D. Lim, J. McAuley, and G. Lanckriet, "Top-n recommendation with missing implicit feedback," in *Proceedings of the 9th ACM International Conference on Recommender Systems*, Vienna, Austria, 2015, pp. 309–312, ISBN: 9781450336925.

[213] Y. Ji, A. Sun, J. Zhang, and C. Li, "A critical study on data leakage in recommender system offline evaluation," *ACM Transaction on Information System*, 2022, ISSN: 1046-8188.

[214] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, 2014, pp. 273–282, ISBN: 9781450323512.

[215] J. Weston, H. Yee, and R. J. Weiss, "Learning to rank recommendations with the k-order statistic loss," in *Proceedings of the 7th ACM International Conferencerence on Recommender Systems*, 2013, pp. 245–248.

[216] F. Abbas and X. Niu, "One size does not fit all: Modeling users' personal curiosity in recommender systems," *ArXivorg*, 2019.

[217] M. Kula, "Metadata embeddings for user and item cold-start recommendations," in *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems*, T. Bogers and M. Koolen, Eds., ser. CEUR Workshop Proceedings, vol. 1448, CEUR-WS.org, 2015, pp. 14–21.

[218] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "Tfmap: Optimizing map for top-n context-aware recommendation," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, pp. 155–164, ISBN: 9781450314725.

[219] M. Ferrari Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," in *Proceedings of the 13th ACM International Conferencerence on Recommender Systems*, Copenhagen, Denmark, 2019, pp. 101–109, ISBN: 9781450362436.

[220] R. Dinga, B. W. Penninx, D. J. Veltman, L. Schmaal, and A. F. Marquand, "Beyond accuracy: Measures for assessing machine learning models, pitfalls and guidelines," *bioRxiv*, 2019.

[221] Y. Deldjoo, A. Bellogin, and T. Di Noia, "Explaining recommender systems fairness and accuracy through the lens of data characteristics," *Information Processing & Management*, vol. 58, no. 5, p. 102 662, 2021.

[222] G. Adomavicius and J. Zhang, "Impact of data characteristics on recommender systems performance," *ACM Transaction on Management Information Systems*, vol. 3, no. 1, 2012.

[223] L. Yang, S. Wang, Y. Tao, J. Sun, X. Liu, P. S. Yu, and T. Wang, "Dgrec: Graph neural network for recommendation with diversified embedding generation," in *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*, Singapore, 2023, pp. 661–669, ISBN: 9781450394079.

[224] A. Said and A. Bellogın, "Comparative recommender system evaluation: Benchmarking recommendation frameworks," in *Proceedings of the 8th ACM International Conferencerence on Recommender Systems*, Foster City, Silicon Valley, California, USA, 2014, pp. 129–136.

[225] U. N. E. C. for Europe *et al.*, "Synthetic data for official statistics: A starter guide," 2023.

[226] J. Jordon, J. Yoon, and M. van der Schaar, "Measuring the quality of synthetic data for use in competitions," *arXiv preprint arXiv:1806.11345*, 2018.

[227] M. N. Fekri, A. M. Ghosh, and K. Grolinger, "Generating energy data for machine learning with recurrent generative adversarial networks," *Energies*, vol. 13, no. 1, p. 130, 2020.

PROPOSITIONS

1. More data does not necessarily lead to a better model performance.
   This proposition pertains to this thesis.
2. Privacy-accuracy trade-offs should not exist.
   This proposition pertains to this thesis.
3. Every type of attack requires a careful selection of privacy protection.
   This proposition pertains to this thesis.
4. Synthetic data amplifies societal harms as much as real data do.
   This proposition pertains to this thesis.
5. Top-rated toolboxes fail to guarantee the reproducibility of results.
6. Perfection stifles productivity.
7. The potential of negative results needs more attention.
8. Social media distorts our perception of reality.
9. The path to self-discovery in life lies not in finding our passion but in finding our purpose.
10. Years of experience lose value if not paired with self-doubt.

These propositions are regarded as opposable and defendable, and have been approved as such by the promotors prof. dr. A. Hanjalic and prof. dr. M.A. Larson.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my promoter, Prof. Alan Hanjalic. Your guidance and insightful advice have been invaluable throughout my PhD journey. Especially in the final stages, your support and thoughtful comments were instrumental in bringing this thesis to completion. Thank you for being a good listener and a great promoter.

To my co-promoter, Prof. dr. Martha A. Larson, words cannot fully capture my appreciation. Our journey began with an extensive interview process in late 2016, a period I remember fondly. Accepting your PhD offer was a dream come true, granting me the opportunity to attend RecSys. Meeting you for the first time at RecSys in Como in August 2017 was a special moment. Our research collaboration has been a source of immense learning and growth for me. From writing academic papers to presenting results, your openness to my ideas and unlimited support and tolerance (e.g., PSD'2022 paper and "others"), have been crucial. Martha, a huge part of who I am today is thanks to you. Your belief in me has allowed me to further develop myself. During the challenging COVID period, your role surpassed that of a supervisor. Your concern for my well-being, including checking on my meals, encouraging short walks, and proposing to be in Nijmegen for one month, meant a great deal to me. Your kindness and support strengthened our relationship beyond the academic environment. Without you, Alan, and Saskia, I wouldn't be here writing this acknowledgment and completing my (*our*) PhD thesis. Thank you, Martha, Nick, and Martha's mum, for making me feel like part of your family.

I would also like to extend my heartfelt thanks to my committee members, Prof. dr. ir. A. Bozzon, Dr. M.S. Pera, Prof. dr. M. Pechenizkiy, Prof. dr. G. Raab, and Prof. dr. K. Muralidhar, and Prof. dr. P. Cesar. Your time, effort, and invaluable feedback have been instrumental in shaping this thesis. I am deeply grateful for your contributions and for being part of my PhD defence.

In 2017 , I moved to the Netherlands, embarking on a new challenge and adventure. Leaving my family to follow my dreams was made easier with Martha's support. Even before my arrival, Martha introduced me to MMC colleagues Soode and Babak. Meeting you both gave me the courage and reassurance that I would find another family at TU Delft. To my *MMC family*, I am grateful for the incredible culture of sharing and caring within our group. The Monday group meetings, dinners, social activities, drinks, coffee/tea breaks, and Christmas lunches are moments I will memorize forever. Babak and Soode, our conversations went beyond life in the Netherlands to in-depth discussions about recommender systems, research, and conferences. Your advice has been invaluable. Raynor, I appreciate your continued presence in our team even after finishing your PhD. I enjoyed our Dutch lessons and cultural talks. Karthik, thank you for being a nice colleague and neighbor, recalling the anecdote when my keys didn't work one summer at 1 am in 2021. Jay, I remember our first discussion about synthetic data (MIT paper) during my first Monday group meeting. Thanks to that conversation, I joined the syn-

thetic data field. Xiuxiu, saying goodbye when you returned to China was tough. You brought so much pleasure into my life, from teaching me badminton and Chinese cooking to playing Zumba. We shared many ups and downs. I tell people that you are my sister in China. Roger, my office-mate and friend in crime, was honored to share the PhD journey with you, attend your wedding, and be your paranymph. Our friendship will last a lifetime. In 2018, Sandy and Alberto joined the MMC family. Alberto, we had many discussions about Italian culture and food, despite discovering you were a "not real" Italian. Sandy, I miss our random walks and talks around in Delft, and your move to Lyon left a space in my life. Omar, I enjoyed our chats about research and daily life. I admire your perseverance and determination. Li, you arrived at just the right time in my life, continuing the camaraderie I had with Xiuxiu. You are not just a colleague but also a wonderful neighbor. Andrew, I will always remember our funny jokes and enjoyable moments, whether they were about research or other daily life topics. Your sense of humor makes every interaction memorable. Matteo, I still remember the time you joined our group. It feels like I have known you for a long time, thanks to your openness and readiness to help, especially during RecSys'2021. Your support has been invaluable. Patrick, Anthony, Dimme, and Shilun, although we did not meet often, I always enjoyed our conversations and the times we checked in on each other. Your kindness and thoughtfulness are greatly appreciated. To everyone, including Bishwadeep, Mohamed, Shishir, Maosheng, Tianqi, Yuanyuan, Chengen, and Andrea, I wish you all brilliant futures filled with success. Thank you for being part of this journey at MMC.

Aside from my PhD colleagues, I want to acknowledge the incredible staff members at MMC. Coming from a slightly different environment but a culture where we share many things, including knowledge and food, I deeply appreciated having Julian (and Monica), Huijuan, Cynthia, and Saskia. Your presence in our group meetings and your willingness to share both work-related and personal stories openly have been inspiring. Julian, thank you for involving me in your course. That was an amazing experience in my academic journey. Cynthia, I admire your interdisciplinary work, team, and work. Your advice has been invaluable. Huijuan, I admire your involvement with your students. Your care for all of us, and your efforts in planning group dinners and other activities, have created a supportive and warm environment. Saskia, from the moment I arrived at MMC, I felt you were like a sister to me. Your kindness, care for our well-being, and willingness to help with everything have been extraordinary. We are incredibly lucky and glad to have you with us. Pablo, even though you worked only one day a week, your presence was always felt. You never hesitated to help or contribute to making MMC a happier place. Now, as colleagues at CWI and ELSA Lab, I am grateful for our continued collaboration. Our MMC group keeps growing, and I have been fortunate to share special moments with new members (joining at the end of my PhD contract). Odette, Elvin, Jorge, Megha, Zhengjun, and Masoud, thank you for the enjoyable coffee/tea breaks and group dinner in 2023. To the bachelor and master students I supervised from TU Delft, it was a pleasure to work with you: Manoj, Mark, Christopher, and Dimitris. Your dedication and enthusiasm made our collaborations enjoyable. Outside of MMC, I also had the pleasure of talking and sharing great moments with the PRB, Cyber-security, and WIS groups at TU Delft. Our work would not have been smooth without the help of Robbert and Bart (retired). To all members of MMC and TU Delft, I wish you continued success

and happiness. You remain my first family in the Netherlands, and I appreciate every moment spent together.

Getting to 2021, when I decided to do an internship at Statistics Netherlands (CBS). This opportunity was made possible thanks to my part-time position at Radboud University. I extend my warmest thanks to Arjan de Vries and everyone involved for approving my application. During this time, I had the privilege of teaching the "Research Methods" course alongside Martha, Yana, and Ilona. Yana and Ilona, it was a pleasure working with you both. Martha, thank you (again and again) for educating me on how to design, communicate, and evaluate students' work. In addition to teaching, I had the pleasure of supervising several bachelor and master students: Joost, Jasper, Danny, Janneke, Jesse, Gea, Gerhard, Micha, Fleur, Aylin, Clara, Selin, Esmee, Jorane, and Mark. Watching you all graduate was a pure moment of happiness for me. Your success is my success, and I am incredibly proud of each one of you. I also got the chance to meet amazing PhD candidates and staff members from the data science group. To my best friend at Radboud, Zhuoran, I appreciate our bi-weekly meetings during the COVID period. They were enjoyable moments for me.

In parallel with my part-time position at Radboud, I began my internship journey at Statistics Netherlands (CBS). I was part of the Methodology team, working closely with my supervisor and colleague, Peter-Paul. Peter-Paul, I consider myself incredibly fortunate to have worked with you. One question you asked me in PSD'2020 changed the scope of my PhD thesis. I am proud of our achievements and if I were given the chance to go back in time, I wouldn't change a thing. I thoroughly enjoyed our collaboration. You were a good listener, open to new ideas, always encouraging me, believing in my potential, and vigilant about my health, among many other things. My achievements in the synthetic data domain, and recently winning the first prize in the Young Statisticians Awards (with COACH), are partly thanks to your guidance and support. I cannot thank you enough. Reinoud, you are more than just the manager of the Methodology team. I admire your hard work and your concern for our well-being. I appreciate your belief in me, the contract offer, your flexibility, and your encouragement to pursue my dreams. Marieke, Nynke (Lucy), Naomi, Nino, and Iris—my CBS colleagues and Dutch best friends—you were there for me in my hardest moments with open arms and endless support. I love you all and will never forget your kindness and friendship. To all my Methodology colleagues, I have been inspired by and learned a lot from each one of you. At CBS, I also had the pleasure of interacting and working with colleagues from different departments, including Process Development, EBM, Software Experts, Policy and Output Checking, as well as people from Security, Management, Canteen, and the Synthetische Data group. To Barteld, Eric, Jel, Fannie, Reinier, Liesbeth, Ran, Silvia, Mohamed, Shruti, Erwin, Lucas, Kim, Janelle, Chris, Fatima, Martin, Hilde, Isabel, Sonia, Amber, Marret, Ahmad, Henrico, Anco, Vanessa, and Arjan (and many more), thank you for being great listeners and for the amazing moments we shared, directly or indirectly.

In 2023, while being offered a contract at CBS, I also received an offer for a Postdoc researcher position at Centrum Wiskunde & Informatica (CWI). Thank you once again to Martha for encouraging me to pursue this opportunity. This position would not have been possible without the flexibility of Reinoud from CBS and Laura from CWI. Laura, my current supervisor and manager at the Human-Centered Data Analytics (HCDA) group,

thank you for everything you have done and continue to do to support me. I am having a wonderful time at HCDA and CWI, thanks to your guidance. I feel incredibly fortunate to be part of the HCDA group. Davide and Lynda, thank you for all your advice on career, academic, and daily life matters. Savvina, Delfi, Andrei, Dirk, Atefeh, Thomas, Boyu, and Ghazeleh, it is a pleasure to share dinners, coffee/tea breaks, conversations about chocolate, and brainstorming sessions about our research. I have special moments with each of you: Andrei, I remember how you gave me a beautiful bouquet in July 2023. Atefeh, Delfi, and Savvina, you never hesitate to hug me whenever I feel down and need support. Dirk, your help with HPC and improving my code have been invaluable. Thomas (and Rackoon), you are a great listener and office-mate. Ghazeleh, your kindness and artistic talent bring joy to our group. Boyu, you are the best graphic designer I know, and thanks to you, I have a beautiful thesis cover. Jacco, it is a pleasure to co-supervise Jovan with you. Anaïs, I enjoy our conversations. CWI is not only about HCDA but also includes the incredible management and support staff: Minnie, Bikkie, Martine, Karima, Ramona, David, Carla, Amber, Paul, Lex, Gael, and Silke. My journey at CWI is already amazing thanks to all of you. I look forward to more fun and achievements.

My project at CWI allows me to be part of the AI, Media, and Democracy Lab (AIMD) and the ELSA Lab. A special mention to Claes and Natali, the co-founders of AIMD. In AIMD, I have the pleasure of working with an interdisciplinary team. This would not be possible without great and hard-working colleagues who share the same goal of success: Abdo, Sanne, Valeria, Hannes, Tomás, Max, Anna, Nick, Leyla, Sophie, Stanislaw, Laurens, Aqsa, Kimon, Zilin, Pippa, Wouter, and Gionata. In the context of the work at AIMD, we collaborate closely on various topics. This brings me to thank the DIS group (Pablo, Karthik, Simone, Moonisa, Abdo) and the IAS group (Valentin, Eric, Han) at CWI. Our collaborations would not be possible without Sara, our manager and the "heart and soul of AIMD". Sara, thank you for believing in me and supporting me in all moments.

From 2017 to now, I have had the privilege of meeting many successful, kind, and wonderful friends. At the ACM International Recommender Systems Conference (RecSys), I enjoyed talking to Michael Ekstrand, Robin Burke, Christine Bauer, Nava, Alan, Kim, Lien, Vito Walter, Andrès, Lorenzo, Joe Konstan, Helma, Alain, Sole, Alejandro, Pablo, Hanna, Maria, and many more. At RecSys, I had the pleasure of being a student volunteer for many years and met numerous friends. To Marcel and Andrès—our friendship has grown beyond RecSys and being SVs into monthly meetings, which I cherish deeply. To my 2x Andrea, I treasure our friendship, which I know will last a lifetime. Within RecSys, I had the pleasure of collaborating with Özlem from NTNU Norway. Özlem, thank you for inviting me to give talks about my work and for being a fantastic collaborator. At the Privacy in Statistical Databases (PSD) conference, I met incredible people working on privacy and synthetic data. I have enjoyed learning from them. Krish, Gillian, Josep, Sara, Mark Elliot, Joerg, Felix, Aleksandra, and Vicenc, thank you for teaching me so much. Being a researcher does not only mean having publications; it also involves being active in society and creating a rich network. In addition to attending various conferences, summer/winter schools, and Dagstuhl, I want to express my thanks to the Kennisnetwerk Synthetische Data, a place where we share our knowledge on synthetic data. Thank you, Lotte, Barteld, Shannon, Charissa, Thiery, Jim, and others, for your work in making this initiative successful. I am also honored to be part of the

UNECE and IEEE Synthetic Data Guidelines.

To everyone I have met in my life, know that you mean a lot to me. Your presence has enriched my journey, and I am grateful for every one of you. I wish you all success and happiness in your lives.

Back in Tunisia and before 2017, I was surrounded by the kindest and most lovely people in the world. Safsouf, our 14-year-old friendship has been a constant source of support. Thank you for your continuous messages, calls, and endless care. Thank you for being by my side during the hardest moments and for introducing me to our group of "tilawat al couran". That was one of the best decisions I have made in my life. A special thanks to our teacher, your mum Nejia, for every letter and word we learned from her. Thank you for your time and dedication. "Allah ibereklek". To Si Salim, you were, are, and will always be my best coach in life. I met you as the president of our handball team, CSMannouba. I admire your openness, ability to listen to all kinds of people equally, endless support, and caring nature. You have inspired me and continue to inspire me immensely. "Stay strong, Si Salim!" More friends who will never be forgotten, even with fewer interactions. To Feryel, Waazzaa, Walid, Cherif, Aouatef, Amira, and Nida, thank you for crossing my path. You have left a mark on me, and I memorized our great moments together.

I cannot end without thanking my blood family and relatives. To my family, both near and far, I love you all. The first thank you goes to you for your endless and unlimited love, support, and care. THANK YOU, Dad, Mum, Mamoucha, Nousty, Tata, Amour, and Mallouka: "Nhebkom barcha." To my nieces and nephews: Lajjounette, Doudouwety, Bayyou, Lilianou, Joujouty, Mayyan, Poupoune, Poupette Sousou, Eyouta, Lindoucha, and Mimi, you have brought so much joy to our family. Bouzid, Kamel, and Ismailo, you are the best and kindest brothers-in-law. My Wiss Wiss, my little sister, I am happy for you and wish you a happy marriage and a lot of happiness in your life. Ma Fifi, thanks to my PhD, I got to know you very closely, and since then, I have treated you as my sister. To my aunts and cousins (including Samia, Mohamed), thank you for being there and being my family. Hannoun x2, I love our infrequent calls that always bring love. Radhouane, thank you for checking on me from time to time and encouraging me.

Thierno, words cannot fully express my gratitude for having you in my life. Since the time I met you, I knew you were exceptional. I never hesitate to reach out to you whenever I am in need, and you have never said "No" to me. I have cried to you many times, and you have never let me down. Your love, support, care, and attention have enriched my life. Your success is my success, and my success is your success. "Bras dessus, bras dessous. Namounala, sama coco bané."

This part of the thesis is the most emotionally challenging one. While the text is fully mine, I thank ChatGPT for smoothing the writing process.

CURRICULUM VITÆ

Manel SLOKOM

24-07-1991	Born in Tunis, Tunisia.

EDUCATION

2011–2013	A bachelor's degree in computer science applied to management ISG Tunis, Tunisia
2013–2015	A master's degree in science and technologies of business intelligence ISG Tunis, Tunisia
2014–2015	A master's degree in data mining Polytech Nantes, University of Nantes, France
2017	A Ph.D. in computer science Multimedia Computing Group, Delft University of Technology, the Netherlands *Thesis:* Towards Purpose-aware Privacy-preserving Techniques for Predictive Applications *CoPromotor:* Prof. dr. Martha Larson *Promotor:* Prof. dr. Alan Hanjalic

PROFESSIONAL EXPERIENCE

2021-2022	Part-time research intern at Statistics the Netherlands.
2021-2022	Part-time research assistant at Radboud University.
2022-2024	Part-time data scientist at Statistics the Netherlands (CBS) until May 2024.
2023-2025	Postdoc reseracher at the National Research Institute for Mathematics and Computer Science (CWI), the Netherlands.

LIST OF PUBLICATIONS

14. Manel Slokom, Jel Vankan, Peter-Paul de Wolf. From COACH to COACH+: Automating Output Checking with Human-in-the-Loop. First prize in the IAOS Prize for Young Statisticians. 2024.

13. Manel Slokom, Jesse Brons, Özlem Özgobek and Martha Larson. A Closer Look at User Attributes in Recommendations: Implications for Privacy and Diversity. Under preparation.

12. Manel Slokom, Jel Vankan, Peter-Paul de Wolf, and Martha Larson. COACH: COmputer-Assisted output CHecking with Human-in-the-Loop. UNECE Expert Meeting on Statistical Data Confidentiality. 2023.

11. Manel Slokom, Peter-Paul de Wolf, and Martha Larson. Exploring Privacy-Preserving Synthetic Data as a Defense against Model Inversion Attacks. Information Security Conference. Springer. 2023.

10. Manel Slokom, Peter-Paul de Wolf, and Martha Larson. When Machine Learning Models Leak: An Exploration of Synthetic Training Data. International Conference on Privacy in Statistical Databases. Springer. 2022.

9. Giuseppe Garofalo, Manel Slokom, Davy Preuveneers, Wouter Joosen, Martha Larson. Machine Learning Meets Data Modification: the Potential of Pre-processing for Privacy Enhancement. In Security and Artificial Intelligence (pp. 130-155). Springer. 2022.

8. Manel Slokom and Martha Larson. Doing Data Right: How Lessons Learned Working with Conventional Data Should Inform the Future of Synthetic Data for Recommender Systems. SimuRec Workshop at the ACM International Conference on Recommender Systems. 2021.

7. Manel Slokom, Alan Hanjalic, and Martha Larson. Towards User-Oriented Privacy for Recommender System Data: A Personalization-based Approach to Gender Obfuscation for User Profiles. Information Processing & Management Journal. 2021.

6. Manel Slokom, Martha Larson and Alan Hanjalic. Partially Synthetic Data for Recommender Systems. In the USB/INTRANET of the International Conference on Privacy in Statistical Databases. 2020.

5. Martha Larson and Manel Slokom. Up Close, but not too Personal: Hypotargeting for Recommender Systems. ImpactRS Workshop at the ACM International conference on Recommender Systems. 2019.

4. Christopher Strucks, Manel Slokom and Martha Larson. BlurM(or)e: Revisiting Gender Obfuscation in the User-Item Matrix. RMSE Workshop at the ACM International conference on Recommender Systems. 2019.

3. Manel Slokom, Martha Larson and Alan Hanjalic. Data Masking for Recommender Systems: Prediction Performance and Rating Hiding. Late-breaking results paper, at the ACM International conference on Recommender Systems. 2019.

2. Martha Larson, Jaeyoung Choi, Manel Slokom, Zekeriya Erkin, Gerald Friedland, Arjen P. de Vries (2019). Privacy and Audiovisual Content: Protecting Users as Big Multimedia Data Grows Bigger. In: Stefanos Vrochidis, Benoit Huet, Edward Y. Chang, Ioannis Kompatsiaris (eds) Big Data Analytics for Large-Scale Multimedia Search, 2019.

1. Manel Slokom (2018). Comparing recommender systems using synthetic data. In Proceedings of the 12th ACM Conference on Recommender Systems.