Protecting User Privacy with Social Media Data and Mining

by

Ghazaleh Beigi

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved January 2020 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Subbarao Kambhampati
Hanghang Tong
Tina Eliassi-Rad

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

The pervasive use of the Web has connected billions of people all around the globe and enabled them to obtain information at their fingertips. This results in tremendous amounts of user-generated data which makes users traceable and vulnerable to privacy leakage attacks. In general, there are two types of privacy leakage attacks for user-generated data, i.e., identity disclosure and private-attribute disclosure attacks. These attacks put users at potential risks ranging from persecution by governments to targeted frauds. Therefore, it is necessary for users to be able to safeguard their privacy without leaving their unnecessary traces of online activities. However, privacy protection comes at the cost of utility loss defined as the loss in quality of personalized services users receive. The reason is that this information of traces is crucial for online vendors to provide personalized services and a lack of it would result in deteriorating utility. This leads to a dilemma of privacy and utility.

Protecting users' privacy while preserving utility for user-generated data is a challenging task. The reason is that users generate different types of data such as Web browsing histories, user-item interactions, and textual information. This data is heterogeneous, unstructured, noisy, and inherently different from relational and tabular data and thus requires quantifying users' privacy and utility in each context separately. In this dissertation, I investigate four aspects of protecting user privacy for user-generated data. First, a novel adversarial technique is introduced to assay privacy risks in heterogeneous user-generated data. Second, a novel framework is proposed to boost users' privacy while retaining high utility for Web browsing histories. Third, a privacy-aware recommendation system is developed to protect privacy w.r.t. the rich user-item interaction data by recommending relevant and privacy-preserving items. Fourth, a privacy-preserving framework for text representation learning is presented to safeguard user-generated textual data as it can reveal private information.

To my loving parents, husband and brother, for making me be who I am.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Huan Liu, for his continuous guidance, encouragement, inspiration and support during my Ph.D. study. I am very fortunate to be his student and I will always be indebted. During the past five years, Dr. Liu gives me freedom through my Ph.D. to explore various research problems and establish myself as an independent researcher, but meanwhile, his critical feedback teaches me to see the big vision and always set a high standard for myself. Dr. Liu is more of a mentor and friend than an advisor for research. He always encourages me to be a better version of myself and gives me constant career advice and mentorship, not only about research, but also about life. Working with him is definitely my lifelong assets.

I would like to thank my thesis committee, Subbarao Kambhampati, Hanghang Tong, and Tina Eliassi-Rad, for their valuable interaction and constructive feedback. I have a diverse committee and they have all been an inspiration for my research. I took the information retrieval course from Subbarao Kambhampati, which provided me with a new angle to look at research challenges. I would like to thank him for his comments on my thesis proposal that helped me rethink my research and inspired novel ideas. I also took the semantic web mining course from Hanghang Tong, which impacted how I look at applied machine learning and data mining. Tina Eliassi-Rad has been a role model for me as a female researcher in the field. I would like to thank her for her early advice on work-life balance when I was just a second year Ph.D. student. I will be always grateful for the time each committee member has spent to help improve my research. I also want to thank my intern mentor at Google, Tim Pan, who gave me a great internship experience by providing a transparent environment for research and letting me contribute my knowledge to the existing projects. Tim is not only a great mentor but also an easygoing friend.

I really appreciated the time working with Data Mining and Machine Learning (DMML) lab members. I would like to thank Jiliang Tang, Ali Abbasi, Reza Zafarani, Huiji Gao, Xia Hu, Shamanth Kumar, Pritam Gundecha, Philippe Christophe Faucon, Fred Morstatter, Suhas Ranganath, Robert Trevino, Isaac Jones, Suhang Wang, Jundong Li, Liang Wu, Ruocheng Guo, Kai Shu, Tahora H.Nazer, Nur Shazwani Kamrudin, Vineeth Rakesh Mohan, Lu Cheng, Justin Sampson, Harsh Dani, Matthew Davis, Kaize Ding, Mansooreh Karami, Raha Moraffah, and Alex Nou for their valuable suggestions and discussions. In particular, Jiliang helped me a lot in my early stages of research and helped me finish my first paper at ASU. I have also had the privilege of supervising undergraduate and Ph.D. students, Alex and Nur. I am grateful to them for helping me to develop my skills as a mentor. I would like to thank Ali Abbasi and Zahra Abbasi for helping me settle down at the beginning of my Immigration. I would also like to thank Ericsson Marin, Narges Masoumi, Rouzbeh Khodadadeh, Lydia Manikonda and all my other friends for this companionship.

Last but not least, I am the first Ph.D. in my family and this dissertation would not be possible without company and support of my entire family. I am deeply indebted to my loving parents, Maryam Moghaddam and Hossein Beigi, for always being there for me no matter where I am, for all unconditional support and patience, for believing in me and for making me be who I am today. I would like to especially acknowledge my husband, Hamidreza Alvari, whom this journey would not have been possible without his love, encouragement, support, patience and wit. Hamidreza was the most motivating factor for starting and continuing my research in social media mining. He was my first mentor -and still is- and helped me publish my very first paper. I am very lucky to have him in my life! Also, many thanks to my brother, Alimohammad Beigi, for his encouragements and moral support. This dissertation is dedicated to all of them.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

The explosive Web growth in the last decade has drastically changed the way billions of people all around the globe conduct numerous activities such as surfing the web, creating online profiles in social media platforms, interacting with other people, and sharing posts and various personal information in a rich environment. This results in tremendous amounts of user-generated data. The massive amounts of user information and the availability of up-to-date data makes the Web a place for organizations to collect and aggregate this information either for legitimate purposes or nefarious goals (Bonneau *et al.*, 2009). On one hand, the user-generated data provides opportunities for researchers and business partners to study and understand individuals at unprecedented scales (Backstrom *et al.*, 2007; Beigi *et al.*, 2018; Beigi, 2018) and therefore provide personalized services for each online user.

On the other hand, the resultant rich user-generated data contains individuals' sensitive and private information, leading to privacy leakage and traceability of online users (Ji *et al.*, 2016a; Narayanan and Shmatikov, 2009; Beigi, 2018; Beigi and Liu, 2019). For example, users may share their vacation plans publicly on Twitter without knowing that this information could be used by adversaries for break-ins and thefts in the future (Zhang *et al.*, 2018; Mao *et al.*, 2011). Publishing complete and intact user data could even result in inferring sensitive information that users do not wish to disclose such as location (Li *et al.*, 2012; Mahmud *et al.*, 2014) and age (Wang *et al.*, 2016).

Internet Service Providers (ISPs) such as AT&T and Verizon also have full access to their users' web browsing histories. ISPs can infer different types of personal in-

formation such as users' political views, sexual orientations and financial information based on the sites they visit. Besides that, a recent study shows the fingerprintability of user's web browsing history (Su *et al.*, 2017). Another example is users' textual data such as reviews, tweets, search queries and posts. User-generated textual data also contains sufficient information that allows people in the textual database to be re-identified Zhang *et al.* (2018) and leaks their private-attribute. One example is AOL search data leak Barbaro *et al.* (2006) in which users were re-identified according to their textual search queries. Moreover, if malicious attackers have access to the system's output and unrestricted auxiliary information about their target users, they are able to extract users' entire user-item interactions history and therefore infer their identity and private-attribute information Ramakrishnan *et al.* (2001); Machanavajjhala *et al.* (2011); Calandrino *et al.* (2011); McSherry and Mironov (2009). These identity exposures may result in harms ranging from persecution by governments to targeted frauds (Christin *et al.*, 2010).

Privacy issues could be prominent when the data is published by a data publisher or a service provider. In general, privacy leakage attacks for user-generated data could be categorized into two types: identity disclosure and private-attribute disclosure attacks (Duncan and Lambert, 1986; Lambert, 1993; Li *et al.*, 2007). Identity disclosure occurs when the adversary maps a targeted individual to an instance in a released dataset. Private-attribute disclosure happens when the adversary could infer new private-attribute information regarding a targeted individual based on the released data. These user privacy issues mandate data publishers to protect users' privacy by sanitizing user-generated data before it is published publicly and leverage privacy-aware personalized services and frameworks as well.

The Goal of privacy protection and data anonymization techniques is to remove or perturb data to prevent adversaries from inferring sensitive information while ensuring

the utility of the published data. One straightforward anonymization technique is to remove "Personally Identifiable Information" (a.k.a. PII) such as names, user ID, age and location information. This solution has been shown to be far from sufficient in preserving privacy (Backstrom *et al.*, 2007; Narayanan and Shmatikov, 2008). An example of this insufficient approach is the anonymized dataset published for the Netflix prize challenge. As a part of the Netflix prize contest, Netflix publicly released a dataset containing movie ratings of 500,000 subscribers. The data was supposed to be anonymized with all PII removed from it, however, users' records were mapped to their corresponding profiles on IMDB (Narayanan and Shmatikov, 2008). The results of this attack show that the structure of the data carries enough information for a potential breach of privacy to re-identiy anonymized users.

Privacy protection comes at the cost of utility loss where utility is defined as the quality of personalized service users receive. User-generated data is critical for online vendors to profile users' preferences from their online activities to predict their future needs. The utility of this data therefore affects the quality of the provided online personalized services and user's satisfaction from them. This leads to a dilemma of privacy and utility and highlights the need to address the trade-off between them.

1.1 Research Challenges

Protecting user privacy and preserving utility for user-generated data is far more challenging than structured one as it is heterogeneous, highly unstructured, noisy and inherently different from relational and tabular data. In this dissertation, we investigate if user privacy can be protected for user-generated data considering the aforementioned dilemma. In particular, we study this problem from different aspects including (1) protecting user privacy in heterogeneous social media data, (2) protecting user privacy in Web browsing history data, (3) protecting user privacy in

user-item interactions data, and (4) protecting user privacy in textual data. To protect user privacy and address the dilemma between privacy and utility, we are faced with several challenges:

- User-generated social media data is heterogeneous and the existing anonymization techniques often make a specific assumption regarding the way this data is anonymized. In particular, these works assume that it's enough to anonymize each aspect of heterogeneous data (e.g., structure, textual, and location information) independently. However, sensitive information could be still leaked from the anonymized data, but we lack conclusive evidence. How can we assay privacy level of anonymized social media data? Is the data considered as private if just one of its two aspects is anonymized? Is it sufficient to independently anonymize all aspects of social media data?

- Users leave traces of Web browsing histories when they are surfing online. This Web browsing history information is rich in content and fingerprintable and thus needs privacy protection. Intuitively, the more dummy links we add to a web browsing history, the more privacy we can preserve. An extreme case is when the added links completely change a user's browsing history to perfectly obfuscate the user's fingerprints. However, such approach largely disturbs user profiles and thus results in utility loss–the maximum utility can only be achieved at the complete sacrifice of privacy. How can we design an effective web browsing history anonymizer that tackles the privacy and utility trade-off? How privacy and utility should be quantified in the context of web browsing histories? How many links and what links should be added to a user's browsing history to boost user privacy while retaining high utility?

- Users make interactions with various entities through recommenders such as

leaving online reviews and rating products. Little has been done to protect users against private-attribute inference attacks through leaked users' interactions history in recommendation systems (Jia and NZhenqiang, 2018; Weinsberg *et al.*, 2012). These works focus on anonymizing user-item data before publishing and address the utility loss by minimizing the amount of changes made to the data (Jia and NZhenqiang, 2018; Weinsberg *et al.*, 2012). However, in the context of recommendation, the utility loss due to this approach can lead to degraded recommendation results. Moreover, just sharing perfectly obfuscated user-item data with a recommendation system does not necessarily prevent the adversary from inferring users' private information in future when they receive and accept new recommendations (e.g., when purchasing new products). How can we develop a personalized privacy-aware recommendation system to guard user-item interaction data against private-attribute inference attacks? How can we ensure that the user's private attributes are effectively obscured after receiving personalized recommendation in future?

- Textual information is one of the most significant portions of data that users generate online. This data not only can reveal the identity of the user but also may contain individual's private-attribute information. Traditional privacy preserving techniques are inefficient for user-generated textual data because this data is highly unstructured, noisy and unlike traditional documental content, consists of large numbers of short and informal posts (Fung *et al.*, 2010). Moreover, these works do not explicitly include utility into the design objective of the privacy preserving model. How should textual information be perturbed to prevent the adversary from text reconstruction and users' re-identification? How should textual data be protected against private-attribute leakage? How

can we ensure that the semantic meaning of the text is preserved with respect to a given task?

1.2 Contributions

In this dissertation, we investigate the problem of protecting user privacy and addressing the dilemma between privacy and utility. The contributions of this dissertation are summarized as follows:

- Studying novel problem of protecting users' privacy while preserving the utility for different types of user-generated data from different aspects using social media data and mining;
- Protecting user privacy in heterogeneous social media data with an adversarial approach by proposing a novel de-anonymization attack applicable which assesses the privacy level of anonymized heterogeneous social media data;
- Protecting user privacy in Web browsing history data by proposing an efficient web browsing history anonymization framework with measures for quantifying the trade-off between user privacy and the quality of online services;
- Protecting user privacy in user-item interaction data by proposing a privacy-aware recommendation system which guards against the inference of private-attribute information while maintaining the user utility;
- Protecting user privacy in textual data by proposing a text representation learning framework that generates a text representation such that it is differentially private and does not contain users' private-attribute information while retaining the utility for a given task;

- Conducting experiments on real-world datasets to verify and demonstrate the effectiveness of the above proposed frameworks.

1.3 Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we introduce some basic privacy concepts and review related work in user-generated data privacy preserving. In Chapter 3, we study the problem of identifying privacy risks in heterogeneous social media data. We first introduce a new generation of adversarial technique applicable to social media network data. Then, we propose a novel de-anonymization technique ATHD to assess the privacy level of anonymized heterogeneous data. We conduct experiments to evaluate ATHD on two real world datasets. In Chapter 4, we study the problem of mitigating the dilemma between privacy and utility for web browsing history data. We first quantify the trade-off between user privacy and utility and then propose an efficient framework PBOOSTER to address the problem of anonymizing web browsing histories while retaining the utility. We conduct experiments and evaluate the proposed approach in terms of privacy and utility. In Chapter 5, we address protecting user privacy problem for user-item interaction data by proposing a privacy-aware recommendation system which protects users' privacy even after they received recommendations. We first devise a mechanism RAP to counter private-attribute inference attacks in the context of recommendation systems using adversarial learning. Then, we detail the experimental results of RAP. In Chapter 6, we investigate protecting user privacy for textual information with respect to the different types of privacy leakage attacks. We first introduce a double privacy preserving text representation learning framework, DPTEXT. Then, we demonstrate the effectiveness of DPTEXT theoretically and empirically. We conclude the dissertation and present broader impacts and promising research directions in Chapter 7.

Chapter 2

PRELIMINARIES AND RELATED WORK

Explosive growth of the Web has drastically changed the way people conduct activities and acquire information. It has not only raised security issues such as finding sybil accounts Al-Qurishi *et al.* (2017), identifying extremist Alvari *et al.* (2019a, 2020) and pathogenic social media accounts Alvari *et al.* (2019b, 2018); Alvari and Shakarian (2019, 2018) and detecting human trafficking Alvari *et al.* (2017, 2016b), but also has raised privacy issues such as leakage of users' identities (Narayanan and Shmatikov, 2009; Beigi *et al.*, 2019d) and private-attribute information Beigi *et al.* (2018); Beigi and Liu (2018a); Beigi *et al.* (2019d). Identifying and mitigating user privacy issues has been studied from different perspectives on the Web and social media (for a comprehensive survey refer to (Beigi and Liu, 2018a, 2020)). Our work is related to a number of research which we discuss below.

First we introduce two different types of users' privacy disclosure. Then, we briefly review traditional privacy preserving techniques such as k-anonymity, l-diversity, t-closeness, and differential privacy. Next, we overview the privacy risks from different aspects and categorize the related work into five groups, 1) social graphs and privacy, 2) web search and privacy, 3) private-attribute information and privacy, 4) recommendation systems and privacy, and 5) textual data and privacy.

2.1 Types of Privacy Leakage

Privacy preserving techniques were first introduced for tabular and micro data. With the emergence of social media, the issue of online user privacy was raised. Researchers then focus on studying privacy leakage issues as well as anonymization and

privacy preserving techniques specialized for social media data. There are two types of information disclosure in the literature: identity disclosure and private-attribute disclosure attacks (Duncan and Lambert, 1986; Lambert, 1993; Li *et al.*, 2007). We can formally define these attacks as:

Definition 2.1.1. ***Identity Disclosure Attack****. Assume $\mathcal{D}$ is a snapshot of user-generated data in social media platforms. $\mathcal{D}$ can include different types of data, i.e., a social graph $\mathbf{G} = (V, E)$ where V is the set of users and E demonstrates the social relations between them, users' behavioral information $\mathbf{A}$ (e.g., reviews, item ratings, etc.), and attribute information $\mathbf{B}$. Given $\mathcal{D} = (\mathbf{G}, \mathbf{A}, \mathbf{B})$, the identity disclosure attack is to re-identify all users in the list of target users V_t by mapping them to their known identities. For each $v \in V_t$, we have the information of her social friends and behavior.*

Definition 2.1.2. ***Private-Attribute Disclosure Attack****. Private attribute information contains those attributes that users do not wish to disclose such as political view, occupation, marital status, medical condition, location, age, and gender. Assume $\mathcal{D}$ is a snapshot of user-generated data in social media platforms. $\mathcal{D}$ can include different types of data, i.e., a social graph $\mathbf{G} = (V, E)$ where V is the set of users and E demonstrates the social relations between them, users' behavioral information $\mathbf{A}$ (e.g., reviews, item ratings, etc.), and attribute information $\mathbf{B}$. Given $\mathcal{D} = (\mathbf{G}, \mathbf{A}, \mathbf{B})$, the private-attribute disclosure attack is used to infer the private-attributes a_v for all $v \in V_t$ where V_t is a list of targeted users. For each $v \in V_t$, we have the information of her social friends and behavior.*

Network graph de-anonymization and author identification are examples of identity disclosure attacks that exists in social media. Examples of private-attribute disclosure attack include inferring users' private-attribute through different types of

user-generated data such as users' textual information, items rating behaviors, and social graphs.

Before we discuss privacy leakage in social media, we first overview the traditional privacy models for structured data. Traditional privacy models such as k-anonymity (Sweeney, 2002), l-diversity (Machanavajjhala *et al.*, 2006), t-closeness (Li *et al.*, 2007) and differential privacy (Dwork, 2008) are defined over structured databases and cannot be directly applied to unstructured user generated data in social media platforms. The reason is that quasi-identifiers and sensitive attributes are not clear in the context of social media data. These techniques are further adopted for social media data which we will discuss more next.

2.2 k-Anonymity, l-Diversity and t-Closeness

k-anonymity was one of the first techniques introduced for protecting data privacy (Sweeney, 2002). The aim of k-anonymity is to anonymize each instance in the dataset so that it is indistinguishable from at least $k-1$ other instances with respect to certain identifying attributes. k-anonymity could be achieved through suppression or generalization of the data instances. The goal here is to anonymize the data such that k-anonymity is preserved for all instances in the dataset with a minimum number of generalizations and suppressions while maximizing the utility of the resultant data. It has been shown that this problem is NP-hard (Aggarwal *et al.*, 2005). k-anonymity was initially defined for tabular data, but then researchers start to adopt it for solving privacy issues in social media data. In social media related problems, k-anonymity ensures that users cannot be identified and there are $k-1$ other users with the same set of features which makes these k users indistinguishable. These features may include users' attributes and structural properties.

Although k-anonymity is among the first techniques proposed for protecting the

privacy of datasets, it is still vulnerable against specific types of privacy leakage. Machanacajjhala et al. (Machanavajjhala *et al.*, 2006) introduces two simple attacks which defeats k-anonymity. The first attack is homogeneity attack in which the adversary can infer an instance's (in this case, a users in social media) sensitive attributes when sensitive values in an equivalence class lack diversity. In the second attack the adversary can infer an instance's sensitive attributes when he has access to background knowledge even in the case that the data is k-anonymized. The second attack is known as background knowledge attack. Variations of background knowledge attacks are proposed and used for inferring social media users' attributes. The background knowledge could be users' friends' or behavioral information. We will discuss more about different types of the attribute inference attacks problem in Sections 6 and 7.

To protect data against homogeneity and background knowledge attacks, Machanacajjhala et al. (Machanavajjhala *et al.*, 2006) introduce the concept of l-diversity. It ensures that the sensitive attribute values in each equivalence class are diverse. More formally, a set of records in an equivalence is l-diverse if the class contains at least l *well represented* values for the sensitive attributes. The dataset is then l-diverse if every class is l-diverse. Two instantiations of the l-diversity concept are then introduced, entropy l-diversity and recursive (c, l)-diversity. With entropy l-diversity, each equivalence must not only have enough different sensitive values, but also each sensitive value must be distributed evenly enough. More formally, the entropy of the distribution of sensitive values in each equivalence class is at least $log(l)$. For recursive (c, l)-diversity, the most frequent value should appear frequent enough in the dataset. Interested readers could refer to the work of (Machanavajjhala *et al.*, 2006) for more details.

After l-diversity, Li et al. (Li *et al.*, 2007) studies the vulnerabilities of l-diversity

and introduce a new privacy concept, t-closeness. They show that l-diversity cannot protect the privacy of data when the distribution of sensitive attributes in the equivalence class is different from the distribution in the whole dataset. If the distribution of sensitive attributes is skewed, then l-diversity presents a serious privacy risk. This attack is known as the skewness attack. l-diversity is also vulnerable against similarity attacks. This attack can happen when the sensitive attributes in an equivalence class are distinct but semantically similar (Li *et al.*, 2007). Li et al. (Li *et al.*, 2007) thus introduce a new privacy concept t-closeness which ensures that the distribution of a sensitive attribute in any equivalence class is close to the distribution in the overall table. More formally speaking, an equivalence class satisfies t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution in the whole dataset is no more than a certain threshold. The whole dataset is said to have t-closeness if all equivalence classes have t-closeness. It's valuable to mention that t-closeness protects the data against attribute disclosure but not identity disclosure.

k-anonymity, l-diversity and t-closeness are further adopted for unstructured social media data. Table.2.1 summarizes different approaches that leverage adopted versions of these techniques for privacy problems in social media. For a thorough discussion on these works, interested readers can refer to (Beigi and Liu, 2018a, 2020).

2.3 Differential Privacy

Differential privacy is a powerful technique which protects a user's privacy during statistical query over a database by minimizing the chance of privacy leakage while maximizing the accuracy of queries (Dwork, 2008). Differential privacy provides a strong privacy guarantee and has been leveraged for many privacy preserving application such as graph data (Xiao *et al.*, 2014), textual information (Zhang *et al.*, 2018),

Table 2.1: Application of k-Anonymity, l-Diversity and t-Closeness Techniques in User Privacy in Social Media.

Technique	Type of Information	Paper
k-degree anonymity	graph structure	(Liu and Terzi, 2008)
k-neighborhood anonymity	graph structure	(Zhou and Pei, 2008)
k-automorphism	graph structure	(Zou *et al.*, 2009)
k-isomorphic	graph structure	(Cheng *et al.*, 2010)
k-anonymity	graph structure and attribute information	(Yuan *et al.*, 2010)
(θ, k)-matching anonymity	graph structure and attribute information	(Andreou *et al.*, 2017)
(k, d)-anonimity	graph structure and attribute information	(Backes *et al.*, 2016)
l-diversity	attribute information	(Machanavajjhala *et al.*, 2006)
t-closeness	attribute information	(Li *et al.*, 2007)

location data (Wang *et al.*, 2017) and recommendation systems (Meng *et al.*, 2018). The intuition behind differential privacy is that the risk of user's privacy leakage should not increase as a result of participating in a database (Dwork, 2008). Differential privacy guarantees that existence of an instance in the database does not pose a threat to its privacy as the statistical information of data would not change significantly in comparison to the case that the instance is absent (Dwork, 2008). This makes it harder for the adversary to re-identify an instance and infer whether the instance is in the database or not or decides which record is associated with it (Kifer and Machanavajjhala, 2011). Differential privacy can be formally defined:

Definition 2.3.1. ϵ **- *Differential Privacy***. *Given a query function $\mathcal{A}(.)$, a mechanism $K(.)$ with an output range $\mathcal{R}$ satisfies ϵ-differential privacy for all datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ differing in at most one element iff:*

$$\frac{Pr[K(\mathcal{A}(\mathcal{D}_1)) = r \in \mathcal{R}]}{Pr[K(\mathcal{A}(\mathcal{D}_2)) = r \in \mathcal{R}]} \leq e^{\epsilon} \tag{2.1}$$

where r is some point in the output range $\mathcal{R}$.

Here ϵ is called privacy budget and it can be also shown that Eq. 2.1 is equivalent to $|\log\left(\frac{Pr[K(\mathcal{A}(\mathcal{D}_1))=r\in\mathcal{R}]}{Pr[K(\mathcal{A}(\mathcal{D}_2))=r\in\mathcal{R}]}\right)| \leq \epsilon$ for some point r in the output range. Note that larger values of ϵ (e.g., 10) results in larger privacy loss while smaller values (e.g., $\epsilon \leq 0.1$) indicate the opposite. For example, a small ϵ means that the output probabilities of $\mathcal{D}_1$ and $\mathcal{D}_2$ at r are very similar to each other which demonstrates more privacy. According to Dwork et al. (Dwork *et al.*, 2014), an uncertainty should be introduced in the output of a function (i.e., algorithm) to be able to hide the participation of an individual in the database. This is quantified by sensitivity, which is the amount of the change in the output of query function $\mathcal{A}$ made by a single data point in the worst case:

Definition 2.3.2. L_1**-*sensitivity***. *The L_1-sensitivity of a vector-valued function $\mathcal{A}$ is the maximum change in the L_1 norm of the value of the function $\mathcal{A}$ when one input changes. More formally, the L_1-sensitivity $\Delta(\mathcal{A})$ if $\mathcal{A}$ is defined as (Dwork* et al., *2014):*

$$\Delta(\mathcal{A}) = \max_{\substack{\mathcal{X},\mathcal{X}' \\ |\mathcal{X}-\mathcal{X}'|=1}} \|\mathcal{A}(\mathcal{X}) - \mathcal{A}(\mathcal{X}')\|_1 \tag{2.2}$$

where $\mathcal{X}$ and $\mathcal{X}'$ are two datasets differ in one entry.

Note that the differential privacy is just a condition on a mechanism which releases the dataset. The mechanism which achieves ϵ-differential privacy is called

sanitization. Laplacian mechanism is one popular sanitization technique which gives differential privacy for rela valued queries by adding a Laplacian noise (Dwork, 2008). Assume that $\mathcal{A}(\mathcal{D})$ is the real value resposne to a certain function (algorithm) $\mathcal{A}$. Then, a random noise $\mathcal{Y}(\mathcal{D})$ is generated from Laplacian distribution and added to $\mathcal{A}(\mathcal{D})$ as:

$$K(\mathcal{A}(\mathcal{D}) = \mathcal{A}(\mathcal{D}) + \mathcal{Y}(\mathcal{D}) \tag{2.3}$$

The Laplacian distribution has zero mean and a scale parameter $\Delta(\mathcal{A})\epsilon$. The density function of the Laplacian noise will be computed as:

$$p(x) = \frac{\epsilon}{2\Delta(\mathcal{A})} e^{-\frac{|x|\epsilon}{\Delta(\mathcal{A})}} \tag{2.4}$$

Note that higher sensitivity $\Delta(\mathcal{A})$ of the query function $\mathcal{A}$) with fixed ϵ, implies more Laplacian noise added to $\mathcal{A})(\mathcal{D})$.

There also exists a relaxed version of ϵ-differential privacy, known as (ϵ, δ)-differential privacy which was developed to deal with very unlikely outputs of $K(.)$ (Dwork *et al.*, 2006; Dwork, 2008). It could be defined as:

Definition 2.3.3. (ϵ, δ)***-differential privacy***. *Given a query function $\mathcal{A}(.)$, a mechanism $K(.)$ with an output range $\mathcal{R}$ satisfies (ϵ, δ)-differential privacy for all datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ differing in at most one element iff:*

$$Pr[K(\mathcal{A}(\mathcal{D}_1)) = r \in \mathcal{R}] \leq e^{\epsilon} \times Pr[K(\mathcal{A}(\mathcal{D}_2)) = r \in \mathcal{R}] + \delta \tag{2.5}$$

Table.2.2 summarizes different works that utilize differential privacy in social media data. For a thorough discussion on these works, interested readers can refer to (Beigi and Liu, 2018a, 2020).

2.4 Social Graphs and Privacy

A large amount of data generated by users in social media platforms has graph structure. Friendship and following/followee relations, mobility traces (e.g. WiFi

Table 2.2: Application of Differential Privacy Technique in User Privacy in Social Media.

Type of Information	Paper
graph structure	(Sala *et al.*, 2011; Proserpio *et al.*, 2014; Xiao *et al.*, 2014; Wang and Wu, 2013; Liu *et al.*, 2016)
recommender systems	(McSherry and Mironov, 2009; Machanavajjhala *et al.*, 2011; Zhu *et al.*, 2013; Jorgensen and Yu, 2014; Shen and Jin, 2014; Hua *et al.*, 2015; Guerraoui *et al.*, 2015; Zhu and Sun, 2016; Meng *et al.*, 2018)
textual data	(Zhang *et al.*, 2018)

contacts, Instant Message contacts) and spatio-temporal data (latitude, longitude, timestamps) all could be modeled as graphs. This mandates paying attention to privacy issues of graph data. We will first overview graph de-anonymization works and then survey the proposed solutions for anonymizing graph data.

Graph De-anonymization. De-anonymization approaches on social networks aim to re-identify the anonymous user data by using previously collected background information. Existing de-anonymization methods can be categorized into i) *seed-based* and ii) *seed-free*, according to whether pre-annotated seed users exist or not. Seed-based de-anonymization attack on social network was proposed to use only structural information and propagates node mappings based on seed user pairs (Narayanan and Shmatikov, 2009). Later, Narayanan *et al.* (Narayanan *et al.*, 2011) employed a simplified attack using less heuristics rules for link prediction problem. Nilizadeh *et al.* further proposed a community-enhanced de-anonymization scheme. Community de-

tection has been extensively studied in the literature of social network analysis Alvari *et al.* (2016a, 2014a, 2013); Yang and Leskovec (2013) and has been used in variety of tasks such as trust prediction Beigi *et al.* (2014) and guild membership prediction Alvari *et al.* (2014b); Hajibagheri *et al.* (2018). This work first de-anonymizes data in community-level and then de-anonymizes the users within the communities (Nilizadeh *et al.*, 2014). Yartseva *et al.* proposed a percolation-based de-anonymization method using neighborhood overlap information (Yartseva and Grossglauser, 2013). Seed-free approaches assume there is no seed users available. Pedarsani *et al.* presented a Bayesian model to iteratively perform a maximum weighted bipartite graph matching starting from the nodes with the highest degree (Pedarsani *et al.*, 2013). Moreover, Ji *et al.* proposed to use optimization based methods to minimize the edge difference between anonymized network and background information (Ji *et al.*, 2014). Recently, another group of works have focused on exploiting additional sources of information such as profile information (Fu *et al.*, 2015) and users attributes (Qian *et al.*, 2016) for social graph de-anonymization. Fu *et al.* proposed to use structural and descriptive information to de-anonymize users without seed nodes (Fu *et al.*, 2015). A thorough survey on graph data anonymization and de-anonymization is presented in (Ji *et al.*, 2016b). Note that de-anonymization methods are similar to those of user identity linkage across social network when only network information is available (Shu *et al.*, 2017). In addition to the different goals of these two research direction, the main difference is that the given graph structured is not anonymized in case of user identity linkage problem. This makes the de-anonymization much more challenging.

Graph Anonymization. Social networks contain private profile information and sensitive social relationships which provide opportunities for researchers to study and understand individuals at unprecedented scales (Beigi *et al.*, 2016b,c, 2019b,e). However, this information may leak users' privacy (Backstrom *et al.*, 2007). Anonymiza-

tion methods serve as an important role to maintain data utility as well as protecting privacy (Wu *et al.*, 2010). Existing social network anonymization methods can be categorized mainly into three categories: *k-anonymity*, *edge randomization*, *clustering-based generalization* and *differential privacy*. The aim of k-anonymity methods is to anonymize each node so that it is indistinguishable from at least $k-1$ other nodes (Sweeney, 2002). Liu *et al.* proposed to achieve k-degree anonymization (Liu and Terzi, 2008) through edge addition/deletion strategies (Liu and Terzi, 2008). Zhou *et al.* further considered the assumption that the adversary knows subgraph constructed by the immediate neighbors of a target node, and aims to achieve k-neighborhood anonymity (Zhou and Pei, 2008). Edge randomization algorithms for social networks usually utilize edge-based randomization strategies to anonymize data, such as random adding/deleting and random switching (Ying and Wu, 2009). Clustering-based anonymization methods group nodes and edges, and only reveal the density and size so that individual attributes are protected (Tassa and Cohen, 2013). Another work seeks to generate an anonymized graph which guarantees differential privacy (Sala *et al.*, 2011).

2.5 Web Search amd Privacy

Web search has become a regular activity where a user composes a query formed by one or more keywords and sends it to the search engine. The engine returns a list of web pages according to the user query. These search queries are a rich source of information for user profiling. Jones et. al. (Jones *et al.*, 2007) studies the potential vulnerabilities of the search engine query logs for the first time. This work first proposes an attack which infer a user's private attributes (e.g., age, gender and location) from her query logs using classification approaches. The next proposed attack is trace attack which maps a particular search trace to an actual user profile

by exploiting the inferred private-attribute information in the previous step. The last proposed attack is person attack which is given a user identity and the goal is to identify the search query log stream. Privacy preserving web search approaches focus on anonymizing users search queries.

One group of works focused on the protection of post-hoc logs (Korolova *et al.*, 2009; Cooper, 2008; Gotz *et al.*, 2012; Zhang *et al.*, 2016). The work of (Korolova *et al.*, 2009) releases a private query click graph which nodes correspond to either queries and URLs and edges represent the number of users who click on the URL given the search query. This graph is used in many applications such as spelling corrections, query classification and keyword generation (Baeza-Yates and Tiberi, 2007; Craswell and Szummer, 2007). This work ensures (ϵ, δ)-differential privacy (Dwork *et al.*, 2006) over released query click graph. It adds Laplacian noise to query counts and number of users who clicked on a link. After adding noise to queries counts, only those with a value greater than a threshold are released. The work of Zhang et al. (Zhang *et al.*, 2016) makes a significant improvement over (Korolova *et al.*, 2009) in which their approach provides an (ϵ)-differential privacy by expanding the query set using an external stochastic query pool. This potential set could be simulated using high frequency n-grams in general English (Davies, 2011).

Another group of approaches including client-side ones focuses on search query obfuscation (Ye *et al.*, 2009; Balsa *et al.*, 2012; Gervais *et al.*, 2014; Howe and Nissenbaum, 2009). These approaches are user-centric and automatically generate fake search queries on behalf of user. For example, TrackmeNot (TMN) (Howe and Nissenbaum, 2009) is implemented as a Web browser add-on and forges search queries. It has an initial seed of query terms which is collected from a set of RSS feeds from popular websites and recently searched popular query terms. Then, TMN sends keywords selected uniformly as search queries from the prepared set. Peddinti et al. (Peddinti

and Saxena, 2010) show that query obfuscation can be broken by an adversarial search engine.

2.6 Private-Attribute Information and Privacy

A user's private-attribute information contains those attributes that users may not wish to disclose such as political view, occupation, medical condition, age, gender, and location. To address the privacy of users, social networks usually offer the option for users to limit the access to their private-attributes, i.e. they are only visible to friends or friends of friends. A user could also create a profile without explicitly disclosing any private-attribute information. However, there exists one privacy attack which focuses on inferring users' private attribute information from their publicly avialble information. This attack is known as private-attribute inference attack. The attacker could be any party who is interested in this information such as social network service providers, cyber criminals, data brokers, and advertisers. Data brokers benefit from selling individuals' information to other parties such as banks, advertisers, and insurance companies [1] . Social network providers and advertisers leverage users' attribute information to provide more targeted services and advertisements. Cyber criminals exploit attribute information to perform targeted social engineering, spear phishing [2] and backup authentication attacks (Gupta *et al.*, 2013). This attribute information could be also used for linking users across multiple sites (Goga *et al.*, 2013) and records (e.g., vote registration records) (Sweeney, 2002; Minkus *et al.*, 2015). Private-attribute inference attacks could be categorized into three groups, 1) friend based, 2) behavior based, and 3) friend and behavior based.

First group of these attacks, i.e., friend based, uses the homophily theory (McPher-

[1] `https://bit.ly/1AwePQE`

[2] `http://www.microsoft.com/protect/yourself/phishing/spear.mspx`

son *et al.*, 2001) and assumes that two friends are more probable to share similar attributes rather than two strangers. This group of attacks leverages a target user's friends' information (He *et al.*, 2006; Lindamood *et al.*, 2009; Gong *et al.*, 2014) and community membership information (Zheleva and Getoor, 2009; Mislove *et al.*, 2010) to infer target's private attributes. Another set of works in this category focuses on predicting both network structure and missing users' private attributes (Yin *et al.*, 2010a,b; Gong *et al.*, 2014). The reason for simultaneously solving these two problems is that users with similar attributes tend to link to one another and individuals who are friends are likely to adopt similar attributes.

Second group of these attacks, i.e., behavior based, are those works which leverage users' behavioral information to infer their private attribute information. Weinsberg et al. (Weinsberg *et al.*, 2012) infers users' attributes (i.e., gender) according to their movie-rating behavior by exploiting different classifiers such as logistic regression, SVM and Naïve Bayes. Kosinski et al. (Kosinski *et al.*, 2013) leverage Facebook likes information into a logistic regression classifier to infer various attributes for each user. Another work (Chaabane *et al.*, 2012) seeks to infer users attributes based on the different types of musics they like. This approach first learns semantic interest topics for each user by using an ontologized version of Wikipeda related to each music and exploiting topic modeling techniques (i.e. Latent Dirichlet Allocation, LDA (Blei *et al.*, 2003)). Then, a user is predicted to have similar attributes as those who like similar types of musics as the user. Luo et al. (Luo *et al.*, 2014) combines graph-based semi-supervised learning with non-parametric regression and uses it to learn a classifier for inferring the household structure based on the users' log of watched TV programs (Luo *et al.*, 2014). Another work (Bhagat *et al.*, 2014) proposes an active learning based attack which infers users' attributes via interactive questions.

The third group of works exploits both friend and behavioral information (Gong

and Liu, 2016, 2018; Jia *et al.*, 2017). Gong et al. (Gong and Liu, 2016, 2018) make a social-behavior-attribute network in which all users' behavioral and friendship information is integrated in a unified framework. Nodes of this graph are either users, behaviors or attributes and edges represents the relationship between these attributes. Private attributes are then inferred through a vote distribution attack model. Another work (Jia *et al.*, 2017) incorporates structural and behavioral information from users who do not have the attribute in the training process, i.e. negative training samples. Then it learns the prior probability of each user having a specified attribute by incorporating the user's behavior information. Next, it models the joint probability of users as a pairwise Markov Random Field according to their social relationships and uses the final model to infer posterior probability of attributes for each target user.

Little work focuses on protecting users against private-attribute inference attacks (Weinsberg *et al.*, 2012; Jia and NZhenqiang, 2018). In (Weinsberg *et al.*, 2012), a predefined number of dummy items is added to each user's profile which are negatively correlated with his actual attributes before publishing anonymized user-item ratings data. Ratings are also added for each dummy item based on either the average item rating or the rating predicted using recommendation approaches such as matrix factorization. In a recent paper (Jia and NZhenqiang, 2018), after a value is sampled for the given private attribute w.r.t. a certain probability distribution which is different from the user's actual attribute, the minimum noise is found and added to the user-item data via adapting evasion attacks such that the malicious attacker predicts the sampled attribute value as the user's private attributes.

2.7 Recommendation Systems and Privacy

Existing privacy preserving works in recommendation systems focus on protecting users against re-identification attacks in which an adversary tries to infer a targeted user's actual ratings and investigate if the target is in the database. They could be categorized into differential privacy based (McSherry and Mironov, 2009; Machanavajjhala *et al.*, 2011; Jorgensen and Yu, 2014; Hua *et al.*, 2015; Zhu and Sun, 2016; Meng *et al.*, 2018) and perturbation based (Rebollo-Monedero *et al.*, 2011; Polat and Du, 2003; Luo and Chen, 2014) approaches. Some methods utilize differential privacy strategy (Dwork, 2008) to modify the answers of the recommendation algorithm so the the presence of a user's data (either a single user-item rating or entire user's history) is masked by increasing the chance that two arbitrary records have close probabilities to generate the same noisy data. McSherry et al. (McSherry and Mironov, 2009) utilize differential privacy to construct private covariance matrices to be further used by recommender. Another work (Jorgensen and Yu, 2014) clusters users w.r.t. the social relations and generates differentially private average of users' preferences in each cluster. Hua et al. (Hua *et al.*, 2015) propose a private matrix factorization which adds noise to item vectors to make them differentially private. Similarly, (Meng *et al.*, 2018) proposes another differentially private matrix factorization which only perturbs users' sensitive ratings. Bassily et al. (Bassily and Smith, 2015) modify user-item ratings data to satisfy differential privacy and then share it with recommender. Another work (Zhu and Sun, 2016) makes items list differentially private and then sends it to recommender. Perturbation based techniques obfuscate user's interactions history by adding fake items and ratings to it. Rebollo et al. (Rebollo-Monedero *et al.*, 2011) propose an information theoretic based privacy metric and then find the obfuscation rate for generating forged user profiles so that the privacy

risk is minimized. Similarly, (Parra-Arnau *et al.*, 2014) proposes to add or remove items and ratings from user profiles minimize privacy risk. Polat et al. (Polat and Du, 2003) use a randomized perturbation technique (Agrawal and Srikant, 2000) by sharing disguised z-score for items a given user have rated. In another work (Luo and Chen, 2014), similar users are grouped to each other. Aggregated ratings of the users within the same group is then used to estimate a group preference vector. Similar to (Polat and Du, 2003), randomness is then added to the preference vector to be shared with the recommender.

2.8 Textual Data and Privacy

People have the right to have anonymous free speech over different topics such as Politics (Narayanan *et al.*, 2012). However, an author's identity can be unmasked by adversaries through providing her real name or IP address to a service provider. However, authors can use tools such as Tor to protect their identity at the network level (Dingledine *et al.*, 2004). Manually generated content will always reflect some characteristics of the person who authored it. For example, some anonymous online author is prone to several specific spelling errors or has other recognizable idiosyncrasies (Narayanan *et al.*, 2012). These characteristics could be enough to figure out whether authors of two pieces of content are same or not. Therefore, with material authored by the true identity of the author, the adversary can discover the identity of a content posted online by the same author anonymously.

Identifying the author of a text according to her writing style, a.k.a stylometry, has been studied a long time ago (Mendenhall, 1887; Mosteller and Wallace, 1964; Stamatatos, 2009). With the adverse of machine learning techniques, researches start to extract textual features and discriminate between 100–300 authors (Abbasi and Chen, 2008). The application of author identification includes identifying authors

of terroristic threats and harassing messages (Chaski, 2005), detecting fraud (Afroz *et al.*, 2012), and extracting author's demographic information (Koppel *et al.*, 2009).

Privacy implications of stylometry have been studied recently. For example, Rao et al. (Rao *et al.*, 2000) investigate whether people who are posting under different pseudonyms to USENET newsgroup can be linked based on their writing style. They use a dataset of 117 people having 185 different pseudonyms and exploit function words and Principal Component Analysis (PCA) to perform matching between newsgroups posting and email domains. Another work from Koppel et al. (Koppel *et al.*, 2006, 2011), studies author identification at the scale of over 10,000 blog authors. They use 4-grams of characters which is a context specific feature. The problem with this work is that it is not clear whether their approach is solving author recognition or context recognition. In another work, Koppel et al. (Koppel *et al.*, 2009) use both content-based and stylistic features to identify 10,000 authors in the blog corpus dataset. There are also several works on identifying authors of academic papers under blind review based on the citations of the paper (Bradley *et al.*, 2008; Hill and Provost, 2003) or other sources from unblind texts of potential authors (Nanavati *et al.*, 2011).

Narayanan et al. (Narayanan *et al.*, 2012) propose another author identification attack which exploits 1,188 real-valued features from each post, such as frequency of characters, capitalization of words, syntactic structure (extracted by Stanford Parser (Klein and Manning, 2003), e.g. noun phrases containing a personal pronoun, noun phrases containing a singular proper noun), and distribution of word length. These features capture the writing style of the author regardless of the topic at hand and can re-identify large number of authors. However this approach will not work when authors anonymize their writing style. Almishari et al. (Almishari and Tsudik, 2012) proposed a new linkage attack which investigates the linkability

of prolific reviews that users post on social media platforms. More specifically, given a subset of information on reviews made by an anonymous user, this approach seeks to map it to a known identified record. This approach first extracts four types of tokens, unigrams, digrams, ratings and category of reviewed entity. Then, it uses Naive bayes and Kullback-Leibler divergence models to re-identify the anonymized information. This approach could be also used for identity disclosure attack across multiple platforms using people's posts and reviews.

Few works consider addressing privacy issues of user-generated textual data (Hakkini-Tur *et al.*, 2006; Anandan *et al.*, 2012; Bowers *et al.*, 2015; Mack *et al.*, 2015; Zhang *et al.*, 2018; Li *et al.*, 2018). The work of (Hakkini-Tur *et al.*, 2006) introduces possible privacy threats of document repositories, 1) name entity recognition, and 2) author identification. It then introduces the concept of k-author anonymity to address the latter issue. However, this work failed to provide technical solutions to address the privacy challenges. Another work from Anandan et al. (Anandan *et al.*, 2012) studies removing PII from text. It first introduces t-Plausibility notion and then propose information theoretic based algorithms which select and generalize sensitive keywords to satisfy t-Plausibility. Its drawback is that it does not address textual representation re-identification and removal of hidden private information.

Bowers et al. (Bowers *et al.*, 2015) propose an anonymization approach which uses iterative language translation (ILT) to conceal one's writing style. This approach first translates English text into another foreign language (e.g., Spanish, Chinese, etc.) and then turns it back to English again for three iterations. Another work from Nathan et al. (Mack *et al.*, 2015) evaluates Bowers's work by introducing a feature selection approach, namely Generative and Evolutionary Feature Selection (GEFES) over the set of predefined features which mask out non-salient previously extracted features. Both (Bowers *et al.*, 2015) and (Mack *et al.*, 2015) are tested over a set of blog posts

by users and the results show the efficiency of ILT-based anonymization.

The work of (Zhang *et al.*, 2018) first introduces a verified version of differential privacy specified for textual data, namely, ϵ-Text Indistinguishability to overcome the curse of dimensionality problem when original differential privacy is deployed on high-dimensional textual data. It then proposes a framework which perturbs user-keyword matrix by adding Laplacian noise to satisfy ϵ-Text Indistinguishability. Another work (Li *et al.*, 2018) uses the idea of adversarial learning to generate text representation. Their framework consists of a generator which generates representation w.r.t. given task and a discriminator which ensures the representation does not contain private information.

Chapter 3

PROTECTING USER PRIVACY IN HETEROGENEOUS SOCIAL MEDIA DATA

Existing anonymization techniques often make a specific assumption regarding the way social media data is anonymized. In particular, these works assume that it's enough to anonymize each aspect of heterogeneous social media data (e.g., structure, textual, and location information) independently. At the first glance, this assumption makes sense as anonymization takes time and effort. Moreover, users privacy is protected while the data utility is preserved at the highest possible level. For example, lets consider the simplest case study in which published data includes only two aspects such as (i) structural (e.g., friendship, follower/followee links) and (ii) textual (e.g., posts) information. We will then have options as shown in Table 3.1 to anonymize the data: no anonymization for either aspect, anonymization for one aspect, and anonymization for both. To ensure anonymization efficiency, as each aspect can be of different data types, a common practice is to anonymize each aspect independently. With two aspects as shown in Table 3.1, case 4 is the backbone of the anonymization techniques for publishing data which is clearly the strongest protection of privacy.

Privacy advocates have argued that sensitive information could be still leaked from the dataset anonymized considering each of these cases, but we lack conclusive evidence. It is unclear how the latent relation between different aspects of the data could be captured, whether the sensitive information with the scale of millions of users could be still leaked and what the success rate of such an attack could be. In particular, in this research, we are interested to study these issues by answering the following research questions:

Table 3.1: Four Different Cases for Social Media Data Anonymization. Each Check Mark Corresponds to the Aspect of Data Being Anonymized.

	Case 1	Case 2	Case 3	Case 4
Structural Anonymization	✗	✗	✓	✓
Textual Anonymization	✗	✓	✗	✓

- **(RQ1)**: Is the data private if just one of its two aspects is anonymized?
- **(RQ2)**: Is case 4, the strongest among four cases, sufficient for anonymizing social media data?

Following the work of (Narayanan and Shmatikov, 2009), we seek to answer these questions by taking an adversary approach to assay the privacy level of anonymized social media data. However, existing de-anonymization attacks require a list of target users. A target user is an individual v with the known identity in social media network $\mathcal{T}$ which will be mapped to a user in the given anonymized dataset. These techniques also require background knowledge $\mathcal{B}_v$ for each targeted user v before initiating the attack. These methods require time and effort to find a proper set of target users and gather their knowledge which may not be realistic in practice. To address these challenges, we first introduce a new generation of adversarial attacks specialized for social media data which does not require collecting information before initiating the attack. Furthermore, to assess different ways of the social media dataset anonymization and answer the aforementioned questions, we propose a novel Adversarial Technique for Heterogeneous Data, namely, ATHD (Beigi *et al.*, 2018) which utilizes the latent relationship between different aspects of data. This new approach particularly well suits for social media data in which it is concerned with assessing the strengths of anonymizing different aspects of data. Our contributions could be

summarized as follows:

- We introduce a new generation of adversarial attacks applicable to social media network data.
- We propose a novel de-anonymization technique ATHD to assess the privacy level of anonymized heterogeneous social media data.
- We implement and evaluate ATHD on two real world datasets to study the strengths of anonymization techniques in context of heterogeneous social media data. Our results demonstrates hidden relations between different aspects of the heterogeneous data make data anonymization techniques inefficient.

3.1 Data Preprocessing and Anonymization

In this section, we review the technical preliminaries of protecting user privacy in social media data, i.e. data anonymization, which is required for the rest of this discussion. Without loss of generality, in this chapter, we assume that the published social media data consists of two aspects, namely, structure and textual information. More formally, we model the social network data as $\mathcal{D} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$ where $\mathcal{V} = \{i | i \text{ is a node}\}$ is the set of nodes or users, $\mathcal{E} = \{e_{i,j} | i, j \in \mathcal{V} \wedge$ there is a link from user i to user $j\}$ is the set of links between any two nodes in $\mathcal{V}$ (e.g., friend and follower/followee relations), and $\mathcal{P} = \{\mathcal{P}_i | i \in \mathcal{V}\}$ is the set of all posts (textual information) associated with users in $\mathcal{V}$. $\mathcal{P}_i = \{p_1^i, p_2^i, ..., p_{m_i}^i\}$ denotes posts by user i where m_i is the number of posts for user i. Note that links in social networks could be either directed (e.g., follower/followee relation in Twitter) or indirected (e.g., friend relation in Facebook). We focus on directed graphs, although it is straightforward to apply the settings on undirected graphs as well. In order to preserve users' privacy, data publisher should anonymize the social media data $\mathcal{D}$ using privacy preservation

techniques. Next, we will discuss techniques deployed to secure structural and textual information.

3.1.1 Structural Information Anonymization

To anonymize structural information, we first remove users' personally identifiable information (PII) such as user's name and ID. Techniques such as k-degree anonymity (Liu and Terzi, 2008), sparsification, perturbation and switching (Ji *et al.*, 2015) are used for adding or removing nodes and links. The aim of k-anonymity methods is to anonymize each node so that it is indistinguishable from at least $k-1$ other nodes (Sweeney, 2002). Liu *et al.* proposed to achieve k-degree anonymization (Liu and Terzi, 2008) through edge addition/deletion strategies (Liu and Terzi, 2008). Sparsification technique randomly removes a set of $p|\mathcal{E}|$ edges (p is the anonymiztion coefficient) while switching methods switches $\frac{p|\mathcal{E}|}{2}$ pairs of edges. Perturbation approach first removes a set of $p|\mathcal{E}|$ edges and then add same amount of edges randomly (Ji *et al.*, 2015).

3.1.2 Textual Information Anonymization

In this work, we anonymize the textual information using ϵ-differential privacy (Dwork, 2008) by first converting each user's post into a numerical vector using tokenizing and calculating Term Frequency Inverse Document Frequency (TF-IDF) scores and then adding Laplacian noise to the text vector. Details are discussed next.

Text Processing. To anonymize user i's posts, we first remove user's PII such as user ID (including mentioning and retweeting), name and link information from her texts. Then, we follow a standard process to convert each of user's posts to a numerical vector. To do so, we first consider posts by all users in the dataset and perform some pre-processing including stop word removal. The unigram model is then deployed

to construct the word feature space $\mathcal{W}$. Finally, we use Term Frequency Inverse Document Frequency (TF-IDF) as a feature weight to derive the vector $\mathbf{x}_l^i$ for each post p_l^i of user i. TF-IDF score for each word t is calculated as:

$$\mathbf{x}_l^i(t) = f_l^i(t) * \log \frac{M}{n_t} \tag{3.1}$$

where, $f_l^i(t)$ is the number of times word t appeared in the post p_l^i, M is the total number of posts in the data and n_t is the number of posts that the word t was used in them. We can represent p_l^i with the corresponding vector $\mathbf{x}_l^i$. All users' posts can be then denoted by the post-word matrix $\mathbf{X} \in \mathbb{R}^{M \times |\mathcal{W}|}$ where $|\mathcal{W}|$ denotes the size of the word space. Relations between users and posts can be also represented via a user-post matrix $\mathbf{W} \in \mathbb{R}^{N \times M}$ where N is the number of users and $\mathbf{W}_{ij} = 1$ if post j was posted by user i and $\mathbf{W}_{ij} = 0$ otherwise. Next, we will discuss how we leverage differential privacy technique to anonymize the textual information.

Anonymizing Textual Information with Differential Privacy. We use differential privacy technique discussed in Section 2.3 to anonymize the textual information. Differential privacy aims at maximizing privacy of users when a statistical query is submitted over a database and an answer is retrieved. We use Laplacian mechanism in order to satisfy differential privacy for real valued queries by adding a Laplacian noise (Dwork, 2008). Assume that $\mathcal{A}(\mathcal{D})$ is the real value response to a certain query $\mathcal{A}$. Then, a random noise $\mathcal{Y}(\mathcal{D})$ is generated from Laplacian distribution and added to $\mathcal{A}(\mathcal{D})$ as:

$$K\left(\mathcal{A}\left(\mathcal{D}\right)\right) = \mathcal{A}\left(\mathcal{D}\right) + \mathcal{Y}\left(\mathcal{D}\right) \tag{3.2}$$

In order to anonymize the post-word matrix $\mathbf{X}$ in a way that ϵ-differential privacy is preserved, we need to apply the discussed mechanism $K(.)$ on the original matrix $\mathbf{X}$ and transform it into a new one $X' = K(X)$. Instead of transforming the entire matrix $\mathbf{X}$ at once, we can transform each individual row of the matrix by adding

a Laplacian noise to $\mathbf{X}_i$ to create a new row $\mathbf{X'}_i$. Considering the identity query function $\mathcal{A}_I(\cdot)$ where $\mathcal{A}_I(D) = D$, the sensitivity of $\mathcal{A}_I(\cdot)$ can be defined as follows:

$$\Delta(\mathcal{A}_I) = \max\|\mathbf{X}_i - \mathbf{X}_j\|_1 \tag{3.3}$$

where $\mathbf{X}_i$ and $\mathbf{X}_j$ are any two random row vectors from $\mathbf{X}$. Following the equation 3.2, a Laplacian noise will be added to each vector $\mathbf{X}_i$:

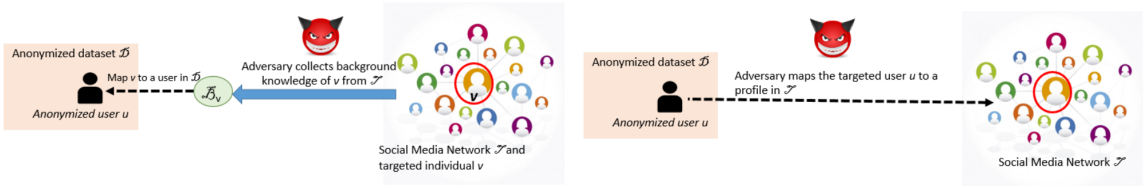
$$K(\mathcal{A}_I(\mathbf{X}_i)) = \mathbf{X}_i + \left[\mathcal{Y}_{i1}, ...\mathcal{Y}_{i|\mathcal{W}|}\right], i = 1, ..., n \tag{3.4}$$

Similarly, Y_{ij}'s are drawn i.i.d. from Laplacian distribution with zero mean and $\Delta(\mathcal{A}_I)/\epsilon$ scale parameter. After anonymizing the textual information, the anonymized post-word $\mathbf{X}$ and user-post $\mathbf{W}$ matrices will be published. The information regrading the word feature space $\mathcal{W}$ will be released by the data publisher as well.

3.2 Social Media Adversarial Attack

De-anonymization techniques have been proposed in the literature as a counterpart to data anonymization research direction (Yartseva and Grossglauser, 2013; Pedarsani *et al.*, 2013; Ji *et al.*, 2016a; Fu *et al.*, 2015; Qian *et al.*, 2016). De-anonymization works further help improve anonymization techniques and reduce privacy breach by probing the potential drawbacks of anonymization techniques. Figure 3.1(a) depicts how these de-anonymization approaches work. These works assume that the adversary has been given a list of target users to de-anonymize requiring adversarial to collect background knowledge about target users before initiating the attack (Abawajy *et al.*, 2016).

Narayanan et.al. (Narayanan and Shmatikov, 2009) discuss different ways of collecting background knowledge such as crawling data via social media networks API. Since these methods require time and effort to gather knowledge, it may not be realistic in practice for two reasons: (1) the number of target users can be very large,



(a) Traditional De-anonymization. **(b) Proposed Social Media Adversarial Attack**

Figure 3.1: Traditional De-anonymization vs. Proposed Social Media Adversarial Attack.

thinking about the number of users in Twitter; and (2) most of the online social media APIs have rate limits on the number of request a user can make through their APIs in a specific time window. Also, these APIs can only provide a random small portion of available data for each search query. This makes it infeasible to collect the background information for a significant number of users in $\mathcal{T}$ in order to find the one-to-one mapping between users in $\mathcal{D}$ and $\mathcal{T}$. Therefore, the above target-user-based approach cannot be applied to social media users when no list of target user is given. To address these shortcomings, we introduce a new generation of adversarial attacks (Figure 3.1(b)) specialized for social media network data. This approach does not require the attacker to gather background knowledge $\mathcal{B}$ before starting the attack. In fact, users registered in social media which are available via online APIs are the adversaries' only source of information. The adversary can send queries to these APIs, anytime during the adversarial process. It is formally defined bellow: Next, we will accordingly discuss the details of our proposed de-anonymization approach, ATHD, which does not require collecting target users and their background information and is proposed to further evaluate heterogeneous social media anonymization.

3.3 Adversarial Technique for Heterogeneous Data

Our proposed de-anonymization technique, adversarial technique for heterogeneous data (ATHD), uses different aspects of data, i.e., graph structure and users' textual information to identify the real identity of users in the anonymized dataset $\mathcal{D} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{W}, \mathcal{W})$. We posit that this attack could be applied on various aspects of data and is not limited to only two data aspects or structural and textual information. The main idea behind de-anonymization is to find the most similar user in social media $\mathcal{T}$ to the user u in the anonymized dataset. Here, we follow the same approach as the existing works, meanwhile our goal is to design a new framework which exploits the hidden relations between different aspects of the data, to eventually map the users to their real profile in $\mathcal{T}$.

Our proposed de-anonymization consists of three main steps. Given the anonymized dataset $\mathcal{D}$, we first extract the most revealing information for u. Second, we search those information in search engine of the targeted social media $\mathcal{T}$. This search returns a list of people whose posts include the inquired query. We save all the returned candidates as a candidate set. Third, we identify the profile from the candidate set most similar to the user u . The details of each of three steps are discussed next.

3.3.1 Step 1: Extracting the Most Revealing Information

The first step includes extracting the most revealing information for user u via social media API. In this work, we rather use textual information since it is not straightforward to look up information related to links. We are thus interested in extracting the most revealing textual information of user u. We assign a score s_l to each post l of u, $\{l \in \{1, ..., M\} | \mathbf{W}_{ul} = 1\}$ to measure how unique each post l is. Each post l has been vectorized using tf-idf approach and is represented in l-th row

of the post-word matrix $\mathbf{X}$. Given the vector representation $\mathbf{X}_l$ of post l, the score s_l is calculated as,

$$s_l = \frac{\sum_{t=1}^{\mathcal{W}} \mathbf{X}_l(t)}{|\mathcal{W}|} \tag{3.5}$$

The higher this score is, the more unique and thus the more revealing post l would be. Based on this, we rank user u's posts and select the top-k posts as the most revealing information.

3.3.2 Step 2: Finding a Set of Candidates

The goal of this step is to find a set of candidates for each user u, given the top-k most revealing posts. To do so, for each nominated post l from step 1, we select set of words $\mathcal{S}$ whose tf-idf scores are greater than the average of the tf-idf scores for the words in the post l, $\mathcal{S}^l = \{t|\mathbf{X}_l(t) > s_l\}$. This approach helps to not to select useless words which have non-zero tf-idf values only due to data distortion during the anonymization process. Therefore, the words with higher chances of being posted in a real text are selected. This step results in a set of queries $\mathcal{Q}_u = \{q_u^{(1)}, q_u^{(2)}, ..., q_u^{(k)}\}$. We construct the query $q_u^{(i)}$ from set $\mathcal{S}^i$, $i \in \{1, ..., k\}$ as $q_u^{(i)} = \{word \in \mathcal{S}^i\}$. Each of $q_u^{(i)} \in \mathcal{Q}_u$ is queried through the $\mathcal{T}$'s search engine. Result includes a set of users who have published posts including keywords in $q_u^{(i)}$.

Integrating results from all queries in $\mathcal{Q}_u$, we have a set of candidate users for user u which is denoted by $\mathcal{C} = \{c_1, c_2, ..., c_{|\mathcal{C}|}\}$. Combining steps 1 and 2, we first find posts which are the most revealing for user u and then for each selected post we select the words that are more likely to be used by the same user. The result will be a set of candidates for u.

3.3.3 Step 3: Matching-Up Candidates to Target

In the last step, we find the most similar candidate to user u. We shall define a metric which measures the similarity between each user u and ith candidate $c_i \in \mathcal{C}$. Previous works (Narayanan and Shmatikov, 2009; Ji *et al.*, 2016b,a; Qian *et al.*, 2016; Nilizadeh *et al.*, 2014) have solely leveraged the structural properties to find the similarity between a target user v and users in an anonymized dataset. However, given the properly anonymized network, the attacker is not be able to accurately find the similarity between users by just incorporating structural properties. We use other aspects of the data (even if they are anonymized) along with structural properties to reveal interesting information that could be leveraged for inferring the similarity. Location, textual and profile information are good examples of such social media data aspects. We consider textual information as the second aspect of the data. We stress that out proposed approach is not limited to textual and structural information and could be generalized to any data type. We also assume that the adversary is not aware of details of deployed anonymization techniques. Next, we define two sets of features to calculate the similarity between u and her i-th candidate c_i.

Structural Features. It has been also shown that users can be uniquely identified using their neighbors degree distributions (Hay *et al.*, 2007). Following previous works (Sharad, 2016), we thus leverage degree distribitons of u's neighbors $\mathcal{N}(u)$ (i.e. all followers and followees of u) in order to represent her structural features. Note that properties such as betweenness, closeness, and eigenvector centrality cannot be considered as u's structural features since it requires having access to the complete network of users in $\mathcal{T}$ which is not feasible in practice. We quantify degree distributions by categorizing them into b bins with size of δ in a way that each bin contains the number of neighbors that have the degree in assigned range of that bin. For

directed graphs, neighbors of each user can be divided into two groups of follower and followee and final feature set is computed by concatenating result of each group.

Textual Features. Remind that for user u, the attacker is given a set of m_u textual vectors as well as word space features $\mathcal{W}$. For each candidate c_i, we collect a set of θ recent posts by sending requests to the $\mathcal{T}$ API. The collected posts are then concatenated in one unified document. Next, c_i's PII will be removed from the document and the corresponding text vector is then created given the word space $\mathcal{W}$ following the similar approach disccused for text processing. Textual features for users u and c_i are thus represented by a set $m_u = \{t_1, t_2, ...t_{m_u}\}$ and a textual vector t_{c_i}, respectively.

Calculating Users Similarity. Given two groups of structural and textual features, similarity between u and c_i is computed as the linear combination of their textual and structural similarities,

$$Sim\left(u, c_i\right) = \alpha Sim_{struct}\left(u, c_i\right) + (1-\alpha) Sim_{text}\left(u, c_i\right) \tag{3.6}$$

where α controls the contribution of structural similarity. We further define $Sim_{struct}(u, c_i)$ as the cosine similarity between the two structural vectors computed as $Sim_{struct}\left(u, c_i\right) = \cos\left(s_u, s_{c_i}\right)$. Textual similarity between u and c_i is also computed as the average of cosine similarity between $\forall t_j \in m_u$ and t_{c_i},

$$Sim_{text}(u, c_i) = \frac{\sum_{j=1}^{|m_u|} \cos\left(t_j, t_{c_i}\right)}{m_u} \tag{3.7}$$

Improving Similarity Measure. Merely checking the structural and textual similarity between the two users' features may lead to biased and not accurate results. Moreover, the attacker needs a more powerful similarity metric which could reduce the effect of anonymization. To handle this issue, we follow a fundamental well-defined problem in the field of image processing (Buades *et al.*, 2005), *image denoising*. Non-local mean filters are a traditional way to remove noise from image data (Buades

et al., 2005). This approach replaces a pixel's value with the weighted average of all other pixels around it. The amount of weighting for neighboring pixels is based on the degree of similarity between a small patch centered on that pixel and a small patch centered on the pixel being denoised (Buades *et al.*, 2005). Inspired by the idea behind non-local mean filters (Buades *et al.*, 2005), we use the feature values of other users similar to user u in order to reduce effect of anonymization. To apply this idea, we first need to find similar users to u– here is where the concept of *homophily* comes in handy. Homophily is one of the most important social correlation theories which is also observed in social media and explains the tendency of individuals to associate and create relationship with similar ones (McPherson *et al.*, 2001; Crandall *et al.*, 2008).

Following the similar idea to non-local means filtering, we leverage homophily and consider user u's neighbor set $\mathcal{N}(u)$ as set of similar users to her. Utilizing homophily also helps in capturing the hidden relations between different aspects of the data. We thus calculate the similarity between $\mathcal{N}(u)$ and neighbors set $\mathcal{N}(c_i)$ for candidate c_i. We first quantify the degree distributions for *all* users in both neighbors set $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ as discussed earlier for strucural features. The structural similarity of neighbors are then calculated based on the cosine similarity between $s_{\mathcal{N}(u)}$ and $s_{\mathcal{N}(c_i)}$.

Following the procedure introduced for extracting textual features, we collect and concatenate θ recent posts for all neighbors in $\mathcal{N}(c_i)$. Textual similarity between $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ is then computed by taking average over the cosine similarities between textual vector of each user in $\mathcal{N}(u)$ and the textual vector of $t_{\mathcal{N}(c_i)}$. The total similarity between neighbors will be then calculated as follows,

$$Sim\left(\mathcal{N}(u), \mathcal{N}(c_i)\right) = \alpha Sim_{struct}\left(\mathcal{N}(u), \mathcal{N}(c_i)\right) + (1-\alpha) Sim_{text}\left(\mathcal{N}(u), \mathcal{N}(c_i)\right) \tag{3.8}$$

This metric quantifies the fitness of $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ as the similarity scores of

their structural and textual properties. It reduces the effect of data anonymization and also aligns well with the assumption that if u and c_i correspond to the same identity, their neighbors $\mathcal{N}(u)$ and $\mathcal{N}(c_i)$ should also match (Fu *et al.*, 2015). Finally, the total similarity between u and c_i can be computed as the combination of their individual similarity and the fitness of their neighbors:

$$Sim_{total}(u, c_i) = \beta Sim\,(u, c_i) + (1 - \beta) Sim\,(\mathcal{N}(u), \mathcal{N}(c_i)) \tag{3.9}$$

We empirically find that random selection of c_i's neighbors with the size λ works well in our problem and we are not required to collect all neighbors information from $\mathcal{T}$'s API. This will make the de-anonymization approach more efficient. Note that many noise removal approaches have been designed for specific kinds of noise (e.g., Guassian noise) which could be used to remove the noise from the data and particularly the vector of textual information. However, using certain noise removal approaches may not always have positive effects. In fact, it can lead to a wrong estimation of users' properties when the attacker does not have any prior knowledge of the deployed anonymized technique.

The proposed ATHD approach is shown in Algorithm 1. The input to the algorithm is the anonymized dataset and the output is the top-h mapped profile accounts in T. Lines 2–5, correspond to the first step of ATHD. The set of candidate set (step 2) is then found through lines 6–9. The similarity between u and each of the selected candidates is calculated in lines 10–12. Finally, top-h candidates with the maximum similarity to u will be returned. This re-identification procedure is then run over all users in the anonymized dataset. Note ATHD is independent of deployed anonymization techniques either for the textual or structural information. In the next section we will discuss how our proposed de-anonymization could be generalized to the social media data with any type of components.

Algorithm 1 Adversarial Technique for Heterogeneous Data

Input: user u, Anonymized Data $\mathcal{D}$= $\{\mathcal{V}, \mathcal{E}, \mathbf{W}, \mathbf{X}, \mathcal{W}\}$, k, λ, θ, h, α, β

Output: Top-h mapped accounts in targeted social network $\mathcal{T}$

Initialize the candidate set $\mathcal{C} = \phi$.
for each anonymized text vector of post l for u **do**
    Calculate score s_l according to Eq.3.5.
end for
Select top-k posts with the highest score s_l as the most revealing information.
for For each text vector l in top-d posts **do**
    Select words with tf-idf scores $\mathbf{X}_l(t) > s_l$ to create search query $q_u^{(l)}$.
    Search query $q_u^{(l)}$ in $\mathcal{T}$ search engine and add results to $\mathcal{C}$.
end for
for each candidate c_i in $\mathcal{C}$ **do**
    Calculate similarity between u and c_i according to Eq.3.9.
end for
Return the top-h candidates with maximum similarity

3.3.4 Generalizability of ATHD

Our framework can be generalized through abstraction to different social media data, assuming that our anonymized data consists of two different aspects, $\mathcal{A}_1$ and $\mathcal{A}_2$ and the attacker is willing to initiate an attack by mapping u to a real profile in the targeted social media $\mathcal{T}$. As discussed before, the first step is to extract the most revealing information from $\mathcal{A}_1$ for user u by using the same concept as tf-idf scores. The second step includes selecting a set of candidate profiles for u by searching for the extracted information from the previous step through $\mathcal{T}$'s search engine. Finally, the similarity between u and her candidates are calculated using the combination of

features of existing data components, $\mathcal{A}_1$ and $\mathcal{A}_2$. Features of the most similar users to u (e.g., neighbors) are also incorporated as well to reduce the anonymization effect while capturing the hidden relation between different aspects of the data.

3.4 Experiments

In this section, we seek to answer the introduced research questions, but we first need to evaluate the efficiency of proposed adversarial technique ATHD. We begin this section by introducing the dataset and anonymization techniques we used. Then, we compare the results of ATHD against the state-of-the-art de-anonymization benchmarks to evaluate its effectiveness. Next, we use ATHD to assess the anonymization power of each of the four cases to answer the research questions:

- **(RQ1)**: Is the data private if just one of its two aspects is anonymized?
- **(RQ2)**: Is case 4, the strongest among four cases, sufficient for anonymizing social media data?

3.4.1 Datasets

We use two different datasets from two large social media websites, Twitter and Foursquare. Twitter is a prevalent and well-known microblogging social media allows millions of active users interacting with each other via short posts, called tweet. Foursquare is a location based social media in which users share their location with friends. Users can also leave tips about different places. We collect the Twitter dataset using Twitter API using the snowball sampling technique as follows. We begin with a random initial seed of users and for each user u in the seed, we obtain a random subset of size 100 of her posted tweets as well as a subset of size 500 of her follower/followee information. We repeat the same process for each u's followers/followees. This way

Table 3.2: Statistics of the Crawled Datasets.

(a) Twitter

# of Users	# of Edges	Avg. Clustering Coefficient
6,789	244,480	0.219
Density	# of Tweets	# of Unigrams
0.005	478,129	208,483

(b) Foursquare

# of Users	# of Edges	Avg. Clustering Coefficient
22,332	229,234	0.295
Density	# of Tips	# of Unigrams
0.0005	124,744	103,264

we build our final dataset which consists of the users in the initial seed and their 2-hops connections. We follow the same procedure to collect the data from Foursquare API by considering a random initial seed of users. We collect each user friends as well as her tips on different locations. We build the final dataset by repeating this process for 2-hops connections. Note that in both datasets, we only keep the information of users who have posted at least one tweet or tip.

Next, we will apply various anonymization techniques on the obtained dataset– this is described in the next section. Also, we utilize the Twitter's advanced search engine [1] and Foursquare search [2] during the de-anonymization process for Twitter and Foursquare data, respectively. It would be also worthwhile to add that we already have the ground truth for the re-identification, since the real profiles of the crawled

[1]https://twitter.com/search-advanced?lang=en

[2]https://foursquare.com/explore?

users are known to us beforehand. Table 3.2 summarizes the statistics of our datasets.

3.4.2 Anonymization Approaches

We use different anonymization techniques to evaluate the introduced different anonymization cases in Table. 3.1. Following previous work (Fu *et al.*, 2015), we choose different algorithms for *structural information* anonymization as follows:

- **Naive Anonymization**. This approach only masks users' identifiers (PII), and does not change the graph structure. This is the simplest approach and thus we would expect the highest vulnerability and hence best de-anonymization result.

- **Sparsification**. This work randomly eliminates $p|\mathcal{E}|$ edges where p is the anonymiztion coefficient.

- **k-deg(add)** (Liu and Terzi, 2008). This anonymization method ensures that k-degree anonymity is preserved by only adding edges.

- **k-degree(add & del)** (Liu and Terzi, 2008). This method ensures that k-degree anonymity is preserved by performing simultaneous add/removal of the edges.

- **Switching**. This method selects two random edges (i_1, j_1) and (i_2, j_2) from the original graph such that $\{(i_1, j_2) \notin \mathcal{E} \wedge (i_2, j_1) \notin \mathcal{E}\}$. Then, it switches pairs of edges, i.e. remove edges (i_1, j_1) and (i_2, j_2) and add new edges (i_1, j_2) and (i_2, j_1) instead. This step is repeated $\frac{p|\mathcal{E}|}{2}$ times which results in $p|\mathcal{E}|$ edge removals/additions.

- **Perturbation**. This method is also known as *unintended* anonymziation and has two main steps. It first removes $p|\mathcal{E}|$ edges in a same way as sparsification method does. Then, it adds random false edges until the number of edges in the anonymized graph is the same as the original one.

Furthermore, the *Textual information* is anonymized using the techniques discussed earlier in Section 3.1.2 as follows:

- **Naive Anonymization**. This approach first removes users' identifiers and links from the tweets and then vectorize it.
- **Diff Privacy**. This method takes the output of the naive anonymization technique and then ensures differential privacy by adding Laplacian noise to the generated text vector.

3.4.3 Experimental Settings

We evaluate de-anonymization approaches by a metric called *success rate* $\mathcal{X} = \frac{n_c}{N}$, where n_c is the total number of users that have been successfully re-identified and N is the total number of users in the anonymized dataset!(Narayanan and Shmatikov, 2009). Larger values of this measure correspond to higher privacy breach.

Following the previous works (Fu *et al.*, 2015; Qian *et al.*, 2016), we set $k = 10$ for k-degree anonymity and $p = 0.1$ for sparsification, purturbation and switching methods. The ϵ for differential privacy technique is set as $\epsilon = 0.01$. We also set the parameters of ATHD as follows: $\{k = 10, \alpha = 0.5, \beta = 0.7, \lambda = 20, \theta = 50, b = 7, \delta = 50\}$. The values of δ and b for quantifying degree distributions are chosen such that it can accommodate higher degrees variation. Empirical results showed that the choice of δ and b does not have a huge impact on the final results. We also set the number of returned profiles as $h = 1$. Clearly, increasing the value of h will increase the de-anonymization success rate. To answer the research questions, we make 12 copies of the original data and sanitize each copy with a different combination of structural and textual anonymization techniques discussed earlier. For evaluation, we define two different variants of our proposed approach, ATHD, as follows:

- ATHD-Simple: This uses Eq.3.6 and Eq.3.7 to calculate similarity.

- ATHD-Improved: This variant uses Eq.3.9 to improve similarity measure by incorporating features from neighbors to reduce the anonymization effect.

3.4.4 Performance Comparison

To evaluate the effectiveness of ATHD, we benchmark its two variants, ATHD-Simple and ATHD-Improved, against the following two baselines.

- **Narayanan et. al.** (Narayanan and Shmatikov, 2009): It computes the similarity between an unmapped user u and a candidate c_i, by using the number of neighbors of u that have been mapped to neighbors of c_i.

- **ADA** (Ji *et al.*, 2016a): This method considers a combination of structural, relative distance and inheritance similarity. We only use degree centrality for measuring structural similarity as we do not have access to the global structure of c_i in $\mathcal{T}$.

In general, these baselines are seed-based approaches, meaning that they map a known target user v in $\mathcal{T}$ to a user in the anonymized data by utilizing a small set of initially mapped seed users and then propagating the mappings through the whole data. These works also need a previously collected background knowledge $\mathcal{B}$. We need to use same settings to make a fair comparison between the baselines and our proposed framework. To do so, we first make an initial seed set of the size $\nu = 20$, by mapping a set of random users in the anonymized dataset to their real identities for each of Twitter and Foursquare data. Then, we repeat the same 3-step procedure as in the ATHD for the baselines, except that we the similarity metric in the last step is replaced with those of the baselines. Performance comparison results for both datasets are demonstrated in Table 3.3 with the following observations:

Table 3.3: Comparison of the De-anonymization Success Rates for Various Anonymization Techniques. Higher Values Imply Higher Privacy Breach. Numbers in Parentheses Demonstrate the Corresponding Case Number in Table 3.1.

(a) Twitter

	ATHD-Improved		ATHD-Simple		ADA		Narayanan et. al.	
	Naive	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**
Naive	0.943(1)	0.802(2)	0.820(1)	0.695(2)	0.672(1)	0.551(2)	0.507(1)	0.410(2)
Sparsification	0.808(3)	0.699(4)	0.732(3)	0.621(4)	0.609(3)	0.511(4)	0.431(3)	0.343(4)
k-deg(add)	0.789(3)	0.681(4)	0.690(3)	0.612(4)	0.589(3)	0.498(4)	0.397(3)	0.313(4)
k-deg(add & del)	0.758(3)	0.653(4)	0.689(3)	0.582(4)	0.580(3)	0.472(4)	0.381(3)	0.299(4)
Switching	0.691(3)	0.581(4)	0.601(3)	0.518(4)	0.497(3)	0.401(4)	0.352(3)	0.261(4)
Perturbation	0.650(3)	0.568(4)	0.536(3)	0.424(4)	0.432(3)	0.361(4)	0.298(3)	0.201(4)

(b) Foursquare

	ATHD-Improved		ATHD-Simple		ADA		Narayanan et. al.	
	Naive	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**	**Naive**	**Diff Privacy**
Naive	0.800(1)	0.679(2)	0.710(1)	0.598(2)	0.569(1)	0.482(2)	0.440(1)	0.375(2)
Sparsification	0.723(3)	0.629(4)	0.640(3)	0.549(4)	0.511(3)	0.453(4)	0.396(3)	0.302(4)
k-deg(add)	0.694(3)	0.599(4)	0.611(3)	0.528(4)	0.513(3)	0.415(4)	0.348(3)	0.274(4)
k-deg(add & del)	0.661(3)	0.573(4)	0.591(3)	0.498(4)	0.486(3)	0.394(4)	0.302(3)	0.263(4)
Switching	0.613(3)	0.543(4)	0.551(3)	0.461(4)	0.430(3)	0.352(4)	0.298(3)	0.212(4)
Perturbation	0.564(3)	0.493(4)	0.451(3)	0.367(4)	0.340(3)	0.283(4)	0.230(3)	0.187(4)

- Narayanan et. al. is the least effective de-anonymization on both datasets. The reason is because its utilized similarity metric relies on the set of previously mapped neighbors and ignores the available structural and textual information provided in the data.

- ADA approach is more powerful than Narayanan et. al. since it incorporates structural properties of the data.

- Anonymized data is more vulnerable to ATHD-Simple compared to ADA and Narayanan et. al. This is because both structural and textual information are incorporated in the similarity metric used in ATHD-Simple. This confirms that integrating different components of data plays an important role in de-anonymization for heterogeneous social media data.

- ATHD-Improved technique achieves the best results for both Twitter and Foursquare datasets. This demonstrates the effectiveness of utilizing homophily and the features of neighbors for more effective de-anonymization.

To recap, the above observations confirm the efficiency of our proposed approach ATHD.

3.4.5 Assessing Effectiveness of Anonymization

Having discussed the efficiency of the proposed ATHD de-anonymization approach, we now seek the answer to the last two questions. The performance results w.r.t. the four anonymization cases are demonstrated in Table 3.3. The numbers in parentheses demonstrate the corresponding case number defined earlier in the introduction. We make the following observations for both datasets:

- Publishing the data with no anonymization for either aspect (i.e., case 1) resulted in

a large information breach in both ATHD-Simple and ATHD-Improved approaches which suggests the least amount of protection as expected.

- In general, anonymizing either aspect of the data (i.e., cases 2 and 3) protects users privacy more than case 1.

- Case 4 is the strongest protection among the four cases. Accordingly, the answer to the second question is no.

- Although case 4 provides the strongest protection, ATHD-Improved was able to re-identify at least 56% of the users in the anonymized dataset, which is a significant number in the field of privacy. This shows that case 4 is far from sufficient for data anonymization.

- Sparsification is the most vulnerable anonymization approach against both ATHD-Simple and ATHD-Improved techniques as it makes the least amount of changes to the link information.

- Although the switching and perturbation methods both add and deletes the same number of edges, switching is more vulnerable to the de-anonymization since it preserves the node degrees.

- Despite the fact that k-degree anonymity based approaches guarantee the user re-identification probability to be at most $\frac{1}{k}$, but they fail because of using extra textual information.

According to these observations, the answers to the introduced research questions are no. These results further indicate that despite anonymization of all aspects of data is essential, but it is not sufficient to anonymize each aspect independently from others. This is because an adversary could easily breach privacy no matter

what anonymization algorithm has been used. Consequently, serious privacy breach could happen when the published data is heterogeneous. This necessitates taking into account the latent relations in different portions of the social media data for anonymization.

3.5 Conclusion

In this chapter, we study a new problem of user data privacy for social media via an adversarial approach. Our work differs from the existing works due to unique properties of social media data: a social media site has an inordinate number of users and the site only allows for a limited number of data queries. Since anonymization takes time and requires dedicated efforts, anonymization efficiency should be maximized. Thus, we evaluate the strengths of anonymization techniques in the context of social media data and verify if it is sufficient. We propose ATHD, a novel adversarial technique by exploiting heterogeneous characteristics of social media data. Our results illustrate that anonymizing even all aspects of data is not sufficient for protecting user privacy due to hidden relations between different aspects of the heterogeneous data.

Chapter 4

PROTECTING USER PRIVACY IN WEB BROWSING HISTORY DATA

The results of our study in the previous chapter 3 highlights the dilemma between protecting user privacy and preserving utility. One type of user-generated data is the web browsing traces individuals leave online. The web browsing history is the list of web pages a user has visited in past browsing sessions and includes the name of the web pages as well as their corresponding URLs. Online users usually expect a secure environment when surfing the Web wherein their personally identifiable information (a.k.a. PII) could be kept hidden from prying eyes. However, the web browsing history log is stored by the web browser on the device's local hard drive. In addition to the web browser, users' browsing histories are recorded via third-party trackers embedded on the web pages to help improve online advertising and web surfing experience. Moreover, Internet Service Providers (ISPs) such as AT&T and Verizon, have full access to individuals' web browsing histories. ISPs can infer different types of personal information such as users' political views, sexual orientations and financial information based on the sites they visit. Some countries have policies for protecting individuals' privacy. For example, European Union (EU) has regulated a new data protection and privacy policy for all individuals within the European Union and the European Economic Area (a.k.a. General Data Protection Regulation (GDPR)). [1] United States government also had Federal Communications Commission's (FCC) landmark Internet privacy protections for users such that ISPs could have been punished by the Federal Trade Commission (FTC) for violating their customers' privacy. However, not all countries have such policies. FCC's Internet privacy protection has

[1] https://bit.ly/1lmrNJz

been also removed in late March of 2017. This new legislation allows ISPs to monitor, collect, share and sell their customer's behavior online such as detailed Web browsing histories without their consent and any anonymization. [2]

Assuming that ISPs and online trackers make browsing history data pseudonymous before sharing, a recent study has shown the fingerprintability of such data by introducing an attack which maps a given browsing history to a social media profile such as Twitter, Facebook, or Reddit accounts (Su *et al.*, 2017). Although linking browsing history to social media profiles may not always lead to figuring out one's real identity, it is a stepping stone for attackers to infer real identities. This identity exposure may result in harms ranging from persecution by governments to targeted frauds (Christin *et al.*, 2010; Beigi *et al.*, 2018).

The onus is now on the users to protect their browsing history from any kind of adversaries like ISPs and online trackers. There are approaches to help users shield their web browsing history such as browser add-ons or extensions (e.g., 'Ghostery', 'Privacy Badger' and 'HTTPS everywhere'), Virtual Private Networks (VPN) services, Tor, and HTTPS. However, none of the above solutions can prevent ISPs from collecting users' web browsing history and protect users' identities when such information is revealed because de-anonymization attacks will still work (Su *et al.*, 2017). Moreover, using these solutions could result in a severe decrease in the quality of online personalization services due to the lack of customer's information. This information is critical for online vendors to profile users' preferences from their online activities to predict their future needs. So users face a dilemma between user privacy and service satisfaction. Hereafter, we refer to a user's satisfaction of online personalization services, as *online service utility*, or simply, *utility*. The aforementioned challenges highlight the need to have a web browsing history anonymizer framework, which can help users

[2]http://wapo.st/2mvYKGa

strike a good balance between their privacy and utility. Traditional privacy preserving web search techniques such as (Yang *et al.*, 2016; Zhang *et al.*, 2016; Zhu *et al.*, 2010) are designed for different purposes and are thus ineffective in accomplishing our goals.

Intuitively, the more links we add to a web browsing history, the more privacy we can preserve. An extreme case is when the added links completely change a user's browsing history to perfectly obfuscate the user's fingerprints. Some existing methods include ISPPolluter, [3] Noiszy, [4] and RuinMyHistory [5] which pollute a web browsing history by adding links randomly. However, such methods largely disturb user profiles and thus results in the loss of utility of online services. Similarly, the maximum service utility can only be achieved at the complete sacrifice of user privacy. It is challenging to design an effective browsing history anonymizer that retains high utility. In this chapter, we aim to study the following problem: *how many* links and *what* links should be added to a user's browsing history to boost user privacy while retaining high utility.

Note that links cannot be removed from the browsing history as all of user's activities have been already recorded by ISPs. The research requires quantifying the privacy of users and the utility of their services. We address these challenges within a novel framework, called PBooster (Beigi *et al.*, 2019a). This framework exploits publicly available information in social media networks as an auxiliary source of information to help anonymizing web browsing history while preserving utility. Our contributions can be summarized as follows.

- We address the problem of anonymizing web browsing histories while retaining high

[3] https://github.com/essandess/isp-data-pollution

[4] https://noiszy.com/

[5] https://github.com/FascinatedBox/RuinMyHistory

service utility. We show that this problem cannot be solved in polynomial time.

- We propose an efficient framework, PBooster, with measures for quantifying the trade-off between user privacy and the quality of online services.

- We conduct experiments and evaluate the proposed approach in terms of privacy and utility. Results demonstrate the efficiency of PBooster in terms of privacy-utility trade-off.

4.1 Threat Model and Problem Statement

Before discussing the details of the proposed solution, we first formally define browsing history, then review the web browsing history de-anonymization and finally introduce the problem of web browsing history anonymization. For each user, web browsing history is defined as the list of web pages a user has visited in his past browsing sessions and includes the corresponding URLs of the visited web pages. This log is recorded by the browser, third-party trackers and ISPs. In addition to his browsing history, other private data components such as cache, cookies and saved passwords are also saved during a browsing session which are sometimes referred to under the browsing history umbrella. However, in this work, we separate these pieces of information from web browsing history. Given a user u, we assume his web browsing history $\mathcal{H}^u$ is generated by a sequence of n links $\mathcal{H}^u = \{l_1, ...l_n\}$ where l_i corresponds to the i-th URL visited by the user u.

4.1.1 Threat Model

De-anonymizing browsing histories is a type of linkage attack which is introduced by Su et al. (Su *et al.*, 2017). This de-anonymization attack links web browsing histories to social media profiles. The main idea behind this threat model is that

people tend to click on the links in their social media feed. These links are mainly provided by the set of user's friends. Since each user has a distinctive set of friends on social media and he is more likely to click on a link posted by any of his friends rather than a random user, these distinctive web browsing patterns remain in his browsing history. Assuming that the attacker knows which links in the history have resulted from clicks on social media feeds, a maximum likelihood based framework is developed as a de-anonymization attack which identifies the feed in the system that has more probably generated the browsing history. This attack can be formally defined as:

Problem 1. *Given user u's web browsing history* $\mathcal{H}^u = \{l_1, ...l_n\}$ *which is consisted of n links, map u to a social media profile whose feed has most probably generated the browsing history (Su* et al.*, 2017).*

Let's assume that each user u has a personalized set of recommender links. For example, this recommendation set could be a set of links appeared in the user's social media Feed (e.g., Twitter) which includes links posted by the user's friends on the network. Su et. al. (Su *et al.*, 2017) assume that each user visits links in his recommendation set. Given a browsing history $\mathcal{H}^u$, the attacker finds the most likely recommendation set that corresponds to the given user u: the recommendation set which contains many of the URLs in the browsing history and is not too big. This de-identifies the browsing history. For the detailed proof and implementation of this attack please refer to (Su *et al.*, 2017). Twitter is selected as a mapping platform for evaluation of this attack. This work shows that users' activities in social media can be used to re-identify them. We next introduce the problem of web browsing history anonymization.

4.1.2 Problem Statement

In this work, we define a privacy preserving framework which protects user's privacy by combating the de-anonymizing web browsing history threat model we discussed in Section 4.1.1. In addition, utility here is also defined as user's satisfaction of online personalized services. This could also be measured by comparing the quality of manipulated web browsing history after anonymization with the original one. Given user u's browsing history $\mathcal{H}^u$, the goal is to anonymize u's browsing history by adding new links to $\mathcal{H}^u$ in an efficient manner, such that both the user's privacy and utility are preserved, i.e., web browsing history is robust against de-identification attack and maintains its utility.

We first need to convert links to a structured dataset. One straightforward solution is to leverage the content of each web page and then map it to a category or a topic selected from a predefined set. This way, each user will be represented by a set of categories extracted from all of the web pages he has visited. One typical way for extracting topics is to manually define them (e.g., sports, fashion, knowledge, etc.) and then map each web page to the corresponding category. This method requires a set of keywords related to each topic and then inferring the web page's topic by calculating the similarity of its textual content to the given keywords. This solution is not feasible in practice since it needs frequent updates of keywords for each category due to the fast growth of the Internet. Moreover, this only provides a coarse-grained categorization of web pages' contents. In order to have a finer level of granularity we follow the same approach as in (Phuong *et al.*, 2014) and adopt Latent Dirichlet Allocation (LDA) topic modeling technique (Blei *et al.*, 2003). We use the following procedure to assign topics for each web page:

1. We retrieve a set of web pages to construct a corpus and then use LDA to learn

topic structures from the corpus.

2. For each web page, the learned topic model in the previous step is used to infer the topic proportion and topic assignment based on the textual content of the page.

3. The topic with highest probability from the topic distribution is selected as the representative topic of the page.

We use $\mathcal{T} = \{t_1, ..., t_m\}$ to denote the set of learned topics. Then each link in the browsing history $\mathcal{H}^u$ is mapped to a topic in the topic set, $t_l \in \mathcal{T}$. Matrix $\mathbf{T}^u \in \mathbb{R}^{n \times m}$ is then used to represent the link-topic relationship for all the links in $\mathcal{H}^u$ where $\mathbf{T}^u_{ij} = 1$ indicates that i-th link of user u is correlated to the topic t_j. The problem of anonymizing browsing history of user u is then formally defined as:

Problem 2. *Given user u's browsing history $\mathcal{H}^u$, and link-topic matrix $\mathbf{T}^u$, we seek to learn an anonymizer f to create a manipulated browsing history $\widetilde{\mathcal{H}^u}$ by adding links to H^u to preserve the privacy of user u while keeping the utility of $\widetilde{\mathcal{H}^u}$ for future applications.*

$$f : \{\mathcal{H}^u, \mathbf{T}^u\} \rightarrow \{\widetilde{\mathcal{H}^u}\} \tag{4.1}$$

We stress that links cannot be removed from the browsing history as all of user's activities have been already recorded by ISPs.

4.2 A Framework for Privacy Boosting

The goal of the web browsing history anonymizer is to manipulate the user's browsing history by adding links in a way that: 1) user privacy is preserved even when the adversary publishes the data with the weakest level of anonymization (i.e., just removing PIIs) and 2) browsing history still demonstrates user's preferences so that the quality of personalized online services is preserved.

An immediate solution that may come to mind is to add links from popular web sites. This approach cannot preserve privacy as the adversary can easily remove popular links from the history and then deploy the attack. Another solution could be adding links from the browsing history of users who are very similar to u, i.e., his friends in social media. This approach can preserve the utility of browsing history but fail to make the user robust to the adversary attack. This is also observed in (Su *et al.*, 2017) where it was shown that the more a user's history contains links from his friends' browsing activities in social media, the more fingerprints he leaves behind. All these emphasize the need for an effective solution which can handle the utility-privacy trade-off.

In this section we will discuss how our proposed algorithm PBooster, can handle utility-privacy trade-off. To better guide the PBooster and to assess the quality of the altered history, we need measures for quantifying the effect of adding links on user privacy and utility. We first present these measures and then detail the PBooster.

4.2.1 Measuring User Privacy

The best case for user privacy is when a user's visited links (i.e., interests) are distributed uniformly over different topics. This improves the user privacy by increasing ambiguity of his interests distribution. This makes it harder for the adversary to infer the real characteristic of the user's preferences and then re-identify him by mapping his anonymized information to a real profile. Entropy is a metric which measures the degree of ambiguity. We leverage the entropy of the user's browsing history distribution over a set of predefined topics as a measure of privacy.

We first introduce the topic-frequency vector $\mathbf{c}_u \in \mathbb{R}^{m\times 1}$ as $\langle c_{u1}, c_{u2}, ..., c_{um}\rangle$ for each user u, where c_{uj} is the number of links in u's history related to the topic t_j. Note that $\sum_{j=1}^{m} c_{uj} = |\mathcal{H}^u|$ where $|.|$ denotes the size of a set. The topic probability

distribution for each user can be then defined as $\mathbf{p}_u = J(\mathbf{c}_u) = \langle p_{u1}, p_{u2}, ..., p_{um} \rangle$ where J is the normalization function of input vector $\mathbf{c}_u$ where $p_{uj} = \frac{c_{uj}}{|H^u|}$ and $\sum_{j=1}^{m} p_{uj} = 1$. The privacy of user u, which is the degree of ambiguity of his browsing history, can be captured by the entropy of the topic probability distribution $\mathbf{p}_u$. This measures the spread of the user's interests across different topics. Given topic probability distribution, privacy is measured as:

$$Privacy(p_u) = -\sum_{j=1}^{m} p_{uj} \log p_{uj} \tag{4.2}$$

The higher this metric is, the greater the user privacy. The optimal value of this measure is thus achieved when the user's browsing links topics are distributed uniformly across the set of topics.

4.2.2 Measuring Utility Loss

Utility or quality of online services is a measurement of a user's satisfaction from the online personalized services he receives based on his online activities. This measurement should be able to estimate the loss of quality of services after manipulating the user's browsing history by the PBooster. We quantify utility loss as the difference between a user's topic distribution before and after browsing history manipulation. Finding the difference between topic distributions has been exploited in other applications such as recommender systems (Li *et al.*, 2011). We use the same notion used in (Li *et al.*, 2011) and quantify the utility loss between $\mathbf{p}_u$ and $\hat{\mathbf{p}_u}$ as:

$$utility\_loss(\mathbf{p}_u, \hat{\mathbf{p}_u}) = 0.5 \times (1 - sim(\mathbf{p}_u, \hat{\mathbf{p}}_u)) \tag{4.3}$$

where $\hat{\mathbf{p}_u}$ denotes the new topic probability after manipulating history. One typical choice for the *sim* is cosine similarity (Li *et al.*, 2011):

$$sim(\mathbf{p}_u, \hat{\mathbf{p}_u}) = \frac{\mathbf{p}_u . \hat{\mathbf{p}}_u}{\|\mathbf{p}_u\| . \|\hat{\mathbf{p}}_u\|} \tag{4.4}$$

Since $sim \in [-1, 1]$, the output of $utility_{loss}$ function will be in $[0, 1]$. According to this measure, the minimum value for utility loss is when $\mathbf{p}_u = \hat{\mathbf{p}}_u$ and the maximum is reached when $\hat{\mathbf{p}}_u$ does not have any non-zero value in common with $\mathbf{p}_u$.

4.2.3 PBooster *Algorithm*

We have discussed so far how to quantify a user's utility and privacy according to his browsing history. The goal is now to find a set of new links $\mathcal{A}$ to add to the browsing history such that, 1) $privacy(\hat{\mathbf{p}_u})$ is as large as possible, and 2) $utility_{loss}(\mathbf{p}_u, \hat{\mathbf{p}_u})$ is as small as possible. However, as we discussed earlier, the optimal value for privacy is reached when the user's interests are spread uniformly across different topics, while the utility loss is minimized when no changes have been done to the topic distribution $\mathbf{p}_u$. This raises a trade-off issue between user's privacy and utility loss. Simply put, maximizing privacy results in the loss of utility and vice versa. In order to optimize the trade-off between utility loss and privacy for each user u, we define a new scalar objective function:

$$G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda) = \lambda * privacy(J(\hat{\mathbf{c}_u})) - utility_{loss}(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u})) \tag{4.5}$$

where $\hat{\mathbf{c}_u}$ is the topic-frequency vector after manipulating browsing history and λ controls the contribution of privacy in G. We aim to find a set of links $\mathcal{A}$ by solving the following optimization problem:

$$\mathcal{A}^* = \underset{\mathcal{A}}{\operatorname{argmax}}\, G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda) \tag{4.6}$$

where $\hat{\mathbf{c}_u}$ could be made from $\widetilde{\mathcal{H}^u} = \mathcal{H}^u \cup \mathcal{A}$. Topic distribution $\hat{\mathbf{p}_u}$ is constructed from $\hat{\mathbf{c}_u}$ accordingly. It's notable to say that the value of λ has impact on the inferred set of links $\mathcal{A}^*$ in a sense that larger values of λ will lead to a browsing history $\widetilde{\mathcal{H}^u}$ with higher privacy while lower λ values result in lower utility loss.

It is worthwhile to mention that the search space for this problem (Eq.4.6) is exponential to N ($\mathcal{O}(m \times 2^N)$), where N is the maximum of the number of links w.r.t. a topic. Considering this fact, it can be expensive and even infeasible to search for the optimal solution. We thus decide to approach this problem in an alternative way. We divide the optimization problem in Eq.4.6 into two subproblems :

1. **Topic Selection:** Selecting a subset of topics and calculating the number of links which should be added to each topic in order to maximize the function G as follows:

$$a^* = \underset{a}{\operatorname{argmax}}\, G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda) \tag{4.7}$$

   where $\mathbf{a} = \langle a_1, .., a_m \rangle \in \mathbb{R}^{m \times 1}$ such that each non-zero element a_i indicates the *number of to-be added new links* which are related to the topic t_i. Zero value means that none of the new links are associated with the topic t_i. Consequently, $\hat{\mathbf{c}_u}$ is defined as $\hat{\mathbf{c}_u} = \langle c_{u1} + a_1, .., c_{um} + a_m \rangle$.

   This step indicates the number of links which should be added to each topic to maximize G.

2. **Link Selection:** Selecting a proper set of links which corresponds to the identified topics and their numbers found in the previous step.

To recap, the PBooster algorithm anonymizes a user's browsing history by first selecting a subset of topics with the proper number of links for each topic (topic selection phase) and then finding corresponding links for each of them (link selection phase). Next, we will discuss the possible solutions for each step.

Topic Selection

One brute-force solution to the optimization problem in Eq.4.7, is to evaluate all possible combinations of a set of topics with different sizes to find the best $\mathbf{a}^*$. The

exponential computational complexity of this algorithm makes it unacceptable and even impractical when quick results are required. We thus need a more efficient solution.

According to a recent study (Guerraoui *et al.*, 2017), having more information in the browsing history will not necessarily increase either the utility or the privacy. In other words, with large information available on user's preferences, observing a new link would have little to no impact on enhancing utility and privacy of the user. Simply put, adding more data to the history, could make the user less secured, with no specific improvement observed in the utility. The submodularity concept formally captures this intuition. A real valued function f is submodular if for a finite set $\mathcal{E}$ and two of its subsets $\mathcal{X}$, $\mathcal{Y}$ where $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{E}$, and $e \in \mathcal{E} \backslash Y$, the following property holds:

$$f(\mathcal{X} \cup \{e\}) - f(\mathcal{X}) \geq f(\mathcal{Y} \cup \{e\}) - f(\mathcal{Y}) \tag{4.8}$$

This means that adding one element $\{e\}$ to the set $\mathcal{X}$ increases f more than adding $\{e\}$ to the set $\mathcal{Y}$ which is superset of $\mathcal{X}$ (Nemhauser *et al.*, 1978). This intuitive diminishing return property exists in different areas such as social media networks and recommender systems. Recall from Eq. 4.5 that the function G is consisted of two components, namely privacy and utility loss. Given $\lambda \in [0, 1]$ and topic-frequency vector $\mathbf{c}_u$, we can rewrite the optimization problem in Eq.4.7 as:

$$\begin{aligned} \underset{\mathbf{a}}{\text{argmax}} - \lambda(\sum_j \hat{p}_{uj} \log \hat{p}_{uj}) - 0.5 \times (1 - \frac{\sum_j p_{uj}\hat{p}_{uj}}{\sqrt{\sum_j p_{uj}^2}\sqrt{\sum_j \hat{p}_{uj}^2}}) \\ \textit{subject to } -\hat{c}_{uj} \leq -c_{uj}\ , \hat{c}_{uj} \in \mathbb{N}_0 \end{aligned} \tag{4.9}$$

where $\hat{p}_{uj} = \frac{\hat{c}_{uj}}{|\hat{\mathcal{H}}^u|}$ is the topic probability distribution after applying PBooster. Privacy is calculated using the entropy function which is submodular in the set of random variables (Krause *et al.*, 2008). The defined utility loss is also naturally

submodular (Li *et al.*, 2011). Since nonnegative linear combinations of submodular functions are submodular as well, the objective function G is submodular. G is also non-monotone and thus the problem in Eq.4.9 is equal to maximizing a non-monotone nonnegative submodular function. This problem has been shown to be NP-hard (Feige *et al.*, 2011) and there is no optimal solution for it in an efficient amount of time.

However, the problem of maximizing non-monotone non-negative submodular function has been solved earlier (Feige *et al.*, 2011). A greedy local search algorithm, LS, has been introduced for solving this problem which was proved to guarantee a near-optimal solution. The greedy LS achieved a value of at least $\frac{1}{3}$ of the optimal solution (Feige *et al.*, 2011). Formally speaking, if we assume solution $\mathbf{a}_G$ is provided by the greedy LS algorithm, and $\hat{\mathbf{c}_G} = \mathbf{c}_u + \mathbf{a}_G$, and the optimal solution is a_{OPT}, and $\mathrm{OPT}(\mathbf{c}_u) = \mathbf{c}_u + \mathbf{a}_{\mathrm{OPT}}$, the following theorem holds:

Theorem 1. *If $G(.,.)$ is a nonnegative non-monotone submodular function, the set of topics $\mathbf{a}_G$ found by the greedy algorithm has the following lower bound (Feige* et al.*, 2011):*

$$G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda) \geq (\frac{1}{3} - \frac{\epsilon}{n}) G(J(\mathbf{c}_u), J(\mathrm{OPT}(\mathbf{c}_u)), \lambda) \tag{4.10}$$

Here, $\epsilon > 0$ is a small number. Local search algorithm iteratively adds an element to the final set or removes one from it to increase the value of G until no further improvement can be achieved. Algorithm 2 shows the topic selection algorithm which deploys the greedy local search. Elements of $\mathbf{a} = \langle a_1, .., a_m \rangle$ will be increased or decreased iteratively to increase value of G until it cannot be improved anymore.

We emphasize that according to (Feige *et al.*, 2011), there is no efficient algorithm which could select the best set of links to maximize aggregation of both privacy and utility in polynomial time. Following the Theorem 1, the proposed greedy algorithm can select a set with a lower bound of $\frac{1}{3}$ of the optimal solution, providing the maxi-

Algorithm 2 Greedy Local Search for Topic Selection

Input: topic-frequency vector $\mathbf{c}_u$, λ, ϵ

Output: $\mathbf{a} = \langle a_1, a_2, ..., a_m \rangle$

1: Initialize $\mathbf{a} = \langle 0, 0, ..., 0 \rangle$, $\hat{\mathbf{c}_u} = \mathbf{c}_u + \mathbf{a}$ and $val \longleftarrow 0$

2: **while** there is increase in in value of $G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda)$ **do**

3: Select $t_j, j \in \{1, ..., m\}$ such that by updating $a_j \longleftarrow a_j + 1$ and $\hat{\mathbf{c}_u} = \mathbf{c}_u + \mathbf{a}$, then $G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda)$ is maximazed

4: Update $val \longleftarrow G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda)$

5: **if** $\exists$ t_j such that updating $a_j \longleftarrow a_j + 1$ and $\hat{\mathbf{c}_u} = \mathbf{c}_u + a$ results in $G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda) > (1 + \frac{\epsilon}{n^2})$. val **then**

6: Update $a_j \longleftarrow a_j + 1$, $val \longleftarrow G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda)$

7: Repeat from step 5

8: **end if**

9: **if** $\exists$ t_j such that $a_j \geq 1$ and updating $a_j \longleftarrow a_j - 1$ and $\hat{\mathbf{c}_u} = \mathbf{c}_u + \mathbf{a}$ results in $G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda) > (1 + \frac{\epsilon}{n^2})$. val **then**

10: Update $a_j \longleftarrow a_j - 1$, $val \longleftarrow G(J(\mathbf{c}_u), J(\hat{\mathbf{c}_u}), \lambda)$

11: Repeat from step 5

12: **end if**

13: **end while**

mum user privacy and utility in polynomial time.

Link Selection

Previously, we discussed the solution for selecting a subset of topics and the proper number of links for each topic to preserve user privacy while keeping the new topic distribution as close as possible to the original one. The second step in PBooster is to select links which correspond to the selected set of topics. Let us assume that

user u has at least one active [6] account on a social media site and PBooster has access to the list of user's friends $\mathcal{F}^u \neq \emptyset$.

Algorithm 3 Link Selection

Input: $\mathcal{F}^u$, q, $\mathbf{a} = \langle a_1, a_2, ..., a_m \rangle$

Output: Set of links $\mathcal{A}$

1: $\mathcal{A} = \emptyset$
2: **for** each update ω in a **do**
3: Let t_j be the corresponding topic of update ω
4: Select a user v randomly such that $v \notin \mathcal{F}^u$
5: Simulate a browsing history $\mathcal{H'}_v$ for v with the size of q. Make c_v and link-topic matrix $\mathbf{T}^u$ from $\mathcal{H'}_v$
6: **if** $c_{vj} = 0$ **then** : Go to line 4 and repeat, **else**
7: Select a non-zero row r randomly from $\mathbf{T}^v[:, j]$
8: Select corresponling link l to row r
9: $\mathcal{A} = \mathcal{A} \cup \{l\}$
10: **end if**
11: **end for**

We propose the following solution for the link selection problem. For each single update ω in the vector a, we randomly select a user v with a public social media profile from outside of the list of u's friends, $v \notin \mathcal{F}^u$. We then simulate v's browsing history $\mathcal{H'}_v$, with the size of $|\mathcal{H'}_v| = q$. The detail of this simulation is discussed in the next section. Link-topic relation matrix $\mathbf{T}^v$ will be constructed from the history $\mathcal{H'}_v$. If there is no link in $\mathcal{H'}_v$ which corresponds to the topic of ω, then the process will be repeated for another random user, otherwise, a random related link will be

[6] Here, user activity does not refer to posting contents. In this work, we assume a user as active if he visits his feed and have non-empty list of friends.

chosen. The pseudocode of this algorithm is shown in Algorithm 3.

To recap, PBooster uses the greedy local search algorithm for submodular maximization to first find the topics which need to be updated and then infer the number of links which should be added to those topics in a way that user privacy and utility is maximized.

4.3 Experimental Evaluation

In this section we conduct experiments to evaluate the effectiveness of PBooster in terms of both privacy and utility. In particular, we seek to answer the following questions: (1) how successful is the proposed defense in protecting users' privacy? (2) how does PBooster affect the quality of online services? (3) how successful is PBooster in handling privacy-utility trade-off?

4.3.1 Dataset

Su et al. (Su *et al.*, 2017) evaluate their de-anonymization strategy by examining it on a set of synthetically generated histories as well as real, user-contributed web browsing histories. Synthetic history is generated for a set of users based on their activities in social media. These users are selected semi-randomly from social media real-time streaming API– the more active a user is, the more likely he is to be chosen. The histories are simulated in a way that mimic users' real online behaviors–they mostly click on links posted to their news feed, and sometimes click on links posted by their friends-of-friends (Su *et al.*, 2017). These friends-of-friends links may be clicked due to the organic exploration behavior of people or the Social media's algorithmic recommendation system that tries to get users visit their friends-of-friends links (Su *et al.*, 2016). Their results on real user generated browsing history is consistent with the results of synthetic histories. This confirms the procedure of simulating synthetic

browsing history as well as the efficiency of the generated data (Su *et al.*, 2017).

Similar to Su et al (Su *et al.*, 2017), we examine the performance of PBooster on a set of synthetically generated browsing history. We follow the same procedure as in (Su *et al.*, 2017) to simulate the browsing history dataset. To generate the synthetic history for each user u, friend's links and friends-of-friends' links are generated accordingly (Su *et al.*, 2017). Friends' links are generated by pulling links from a randomly selected friend of u. Friends-of-friends' links are also generated by first picking one of u's friends, say v, uniformly at random, and then pulling a link from one of v's friends. Following (Su *et al.*, 2017), we select Twitter as the source of users' activities to simulate data because of two reasons. First, many users activities on Twitter are public, and second Twitter has real-time API which helps avoid the need for large-scale web crawling. We select a total number of 1200 users and following (Su *et al.*, 2017), we generate histories of various sizes including $\{30, 50, 100\}$ for each user. For each history, 16% of links are from friends-of-friends and the rest are from friends. Note that we only select links that are related to web pages in English to make the textual analysis easier.

4.3.2 Experiment Setting

To simulate the real-world browsing situation, we divide the browsing history into $\frac{|\mathcal{H}^u|}{h}$ batches of links with size of h. These batches will be added to the history incrementally and PBooster will anonymize the updated history after taking each batch. We set the values $h = 25$, $q = 20$ (used in link selection algorithm) and trade-off coefficient $\lambda = \{0, 0.1, 0.5, 1, 10, 20, 50, 70, 100\}$. We use LDA topic modeling from Python package gensim (Rehurek and Sojka, 2010) and set the number of topics $m = 20$ and LDA parameters $\alpha = 0.05, \beta = 0.05$. We compare PBooster with the following baselines:

- RANDOM: Assuming x new links are added by PBOOSTER, this approach selects x links randomly from the browsing history of users who are not from u's friends. Note that this method does not consider the topics of the links. We compare our model against this method to investigate whether the topics of the chosen links will have effect on the privacy of the users, or in other words, how well topic selection technique in Algorithm 2 performs?

- JUSTFRIENDS: This approach is quite similar to PBOOSTER except that in the link selection phase, it adds links from a user's friends' simulated browsing history. We use this method to see how well our link selection technique in Algorithm 3 performs.

- ISPPOLLUTER [7] : The goal of this method is to eliminate the mutual information between actual browsing history and the manipulated one. According to (Ye *et al.*, 2009) mutual information vanishes if:

$$n_{Noise} \geq (n_{Calls} - 1) \times n_{PossibleCall} \tag{4.11}$$

  where $n_{PossibleCall}$ is the number of domains that a user might visit per day, and n_{Calls} is the number of visited domains. For instance, if a user visits 100 domains and requests 200 calls per day, then *ISPPolluter* adds 20,000 links randomly to the history. We choose this method to see if eliminating mutual information can preserve privacy in practice.

4.3.3 Privacy Analysis

To answer the first question, we first compare each user's privacy before and after anonymization for browsing histories with size 100 ($|\mathcal{H}^u| = 100$). Fig. 4.1 depicts

[7] https://github.com/essandess/isp-data-pollution

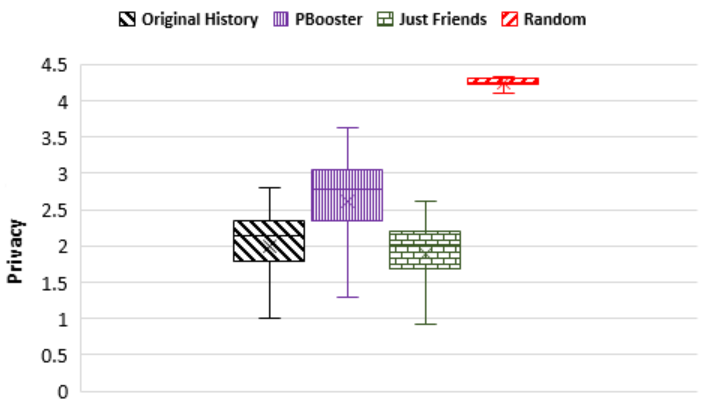
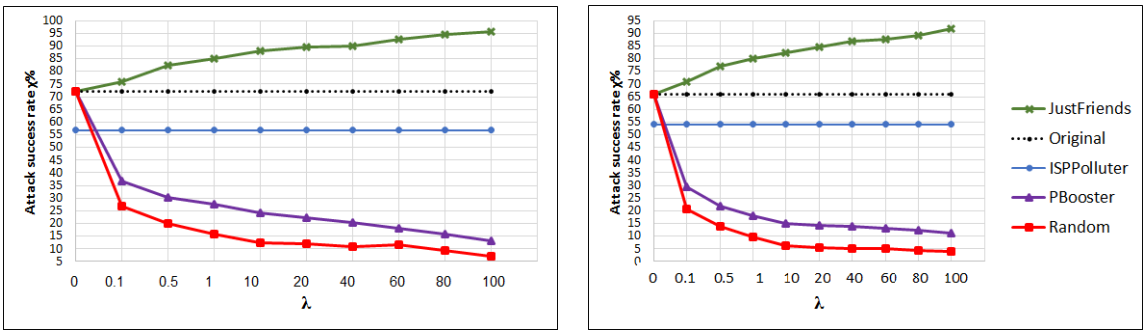


Figure 4.1: Privacy Distributions Before and After Running Anonymization Techniques.

box plots of the distributions of users' privacy measured using Eq.4.2. The privacy-utility trade-off coefficient λ is also fixed to $\lambda = 10$. Results demonstrate how privacy increases after deploying PBooster in comparison to JustFriends approach and original history. This shows that adding links from friends cannot make significant change in privacy. This is because of Homophily effect (McPherson *et al.*, 2001). The Random technique leads to the most uniform topics distribution and thus highest privacy among others.

We now evaluate the efficiency of PBooster against the de-anonymization attack introduced in (Su *et al.*, 2017). We measure the attack success rate by the metric $\mathcal{X}\% = \frac{n_c}{N} \times 100$ where n_c is the total number of users that have been successfully mapped to their Twitter accounts, and N indicates the total number of users in the dataset. We consider the attack as successful if the user is among the top 10 results returned by the attack. Lower values of this measure translates to the higher privacy and stronger defense. We evaluate all methods on histories with different sizes. The results for browsing histories with different λ are demonstrated in Fig. 4.2. Note due to the lack of space, we have removed the similar trend that we observed for



(a) Browsing History of Size 50 **(b)** Browsing History of Size 100

Figure 4.2: Attack Success Rate for Different Sizes of History.

$|\mathcal{H}^u| = 30$. We observe the following:

- ISPPOLLUTER does not work properly in practice and is not robust to the attack which leverages traces of users' activities in social media. This confirms the idea of selecting links from non-friend users which inhibits the adversary to find the targeted user.

- RANDOM is more robust to the attack than PBOOSTER and JUSTFRIENDS. This demonstrates that adding random links from non-friends could perform better in terms of privacy.

- JUSTFRIENDS decreases the privacy in comparison to the original history. This aligns well with the observations of (Su *et al.*, 2017) suggesting that adding links from friends can even decrease the privacy.

- Attack success rate decreases to 15% after applying PBOOSTER. We conclude that the generated history from PBOOSTER is more robust to the attacks in comparison to original history and those generated from JUSTFRIENDS and ISPPOLLUTER. This confirms the effectiveness of PBOOSTER for preserving privacy.

- PBooster performs better when $|\mathcal{H}^u| = 100$. This means larger history can help PBooster to model user's interests better and manipulate the history accordingly.

- PBooster is much more robust than JustFriends. This clearly shows the efficiency of the link selection approach.

- In PBooster,the attack success rate first decreases with the increase of λ and then it gets almost stable (for $\lambda \geq 10$). This makes the selection of λ easier and suggests that the privacy will not increase significantly after some point, confirming that adding more links does not always necessarily lead to more privacy.

- By deploying PBooster, the attack success rate decreases even when λ slightly changes from 0 to 0.1, which confirms the effectiveness of PBooster in anonymizing browsing histories.

Table 4.1: Attack Success Rate after Applying PBooster for Different Values of h with $\lambda = 10$.

	$h = 5$	$h = 15$	$h = 25$	$h = 50$	$h = 100$
$\mathcal{X}$	27.83	19.58	15.13	7.83	5.33

To study the effect of h (size of batches of links in browsing history), we repeat the attack with different values of h for $|\mathcal{H}^u| = 100$ with $\lambda = 10$ which was empirically found to work well in our problem. Results are demonstrated in Table 4.1 suggesting that increasing h can help to model users' preferences more accurately and PBooster can further decrease the traceability of users by making the profiles more ambiguous. Although this increases the privacy, it increases the anonymization waiting time which could result in sudden publishing of history without proper anonymization.

4.3.4 Utility Analysis

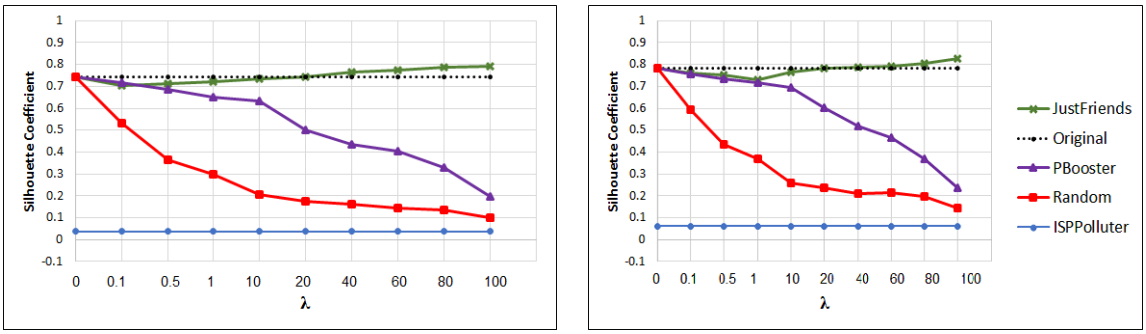
To answer the second question, we investigate the utility of the manipulated histories to estimate the change in quality of services. We evaluate the utility of manipulated history via a well-known machine learning task, i.e., clustering. Prior works (Ungar and Foster, 1998; Sarwar *et al.*, 2002) have indicated the benefits of applying clustering in personalization which can help to offer similar services to same cluster of people.

We use k-means to cluster users into $k = 5$ groups based on their topic preferences distribution $\hat{p_u}$. We evaluate the utility of browsing histories according to the quality of generated clusters via Silhouette coefficient. Silhouette coefficient ranges from $[-1, 1]$, where a higher value indicates better clusters while a negative value indicate that a sample has been assigned to the wrong cluster. Values near zero indicate overlapping clusters (i.e., all users are similar to each other). The results are demonstrated in Fig.4.3. The same trend was observed for $|\mathcal{H}^u| = 30$ but we remove it due to space limitations. We make the following observations:

- Clusters by ISPPOLLUTER has the lowest Silhouette coefficient close to 0 (i.e., clusters are almost overlapping). This shows that adding a large number of random links results in making all users similar to each other and thus severe utility degradation.

- The quality of clusters formed by RANDOM decreases by increasing λ. This confirms that adding links randomly decreases the utility of browsing history and thus shows the importance of the topic and link selection phases.

- JUSTFRIENDS can even increase the utility of the manipulated browsing history. This is not surprising and the reason is that friends have more similar tastes to each



(a) Browsing History of Size 50 **(b)** Browsing History of Size 100

Figure 4.3: Silhouette Coefficient After k-means with $k = 5$ for Different Sizes of History.

other than random people (Homophily effect (McPherson *et al.*, 2001)). Therefore, adding links from a friends' history will not change the preferences distributions significantly. Utility also improves slightly with increase in value of λ.

- Generated history by PBooster has better quality when $|\mathcal{H}^u| = 100$ in comparison to $|\mathcal{H}^u| = 50$. This shows that PBooster works better when more user's information is fed to it.

- The quality of clusters by PBooster decreases with increase in value of λ. The change is even sensible when $\lambda \geq 20$.

- The quality of data generated by PBooster is comparable to the original data when $\lambda \leq 10$. Moreover, PBooster reaches the optimal point in privacy-utility trade-off by fixing $\lambda = 10$.

We repeat k-means with different values of h for $|\mathcal{H}^u| = 100$ with $\lambda = 10$. Results are demonstrated in Table 4.2 and suggest that increasing h will lead to more accurate representation of users and thus improvement in the utility of data. However, as

Table 4.2: Silhouette Coefficient after Applying PBooster for Different Values of h with $\lambda = 10$.

	$h = 5$	$h = 15$	$h = 25$	$h = 50$	$h = 100$
S	0.477	0.5699	0.694	0.731	0.762

discussed earlier, the main drawback with increasing value of h is increasing the risk of sudden history publishing without proper anonymization.

4.3.5 Privacy-Utility Trade-off

To answer the third question, we plot the privacy and utility gain values for each user after applying different approaches over histories with size 100. We measure the privacy by Eq.4.2 and utility gain as $1 - utility_{loss}$ using the Eq.4.3. Different colors and markers represent different approaches. Each marker represents a user, with measures over his manipulated history with $h = 25$ and $\lambda = 10$.

- The original history gains the utility of 1 and the privacy to some extent. Random reaches the highest privacy but loses utility. JustFriends results in higher data utility gain in comparison to other methods but reaches a lower level of privacy. The result of PBooster varies for different users, achieving different levels of privacy and utility according to their original browsing behavior, whereas all users gain similar level of privacy by Random.

- Users achieve higher privacy with PBooster than the original data comparing with other approaches. The achieved utility by PBooster is more than the utility by Random but less than the utility by JustFriends. The reason lies at the intrinsic trade-off between utility and privacy–higher privacy results in less utility.

We compare the privacy and utility of browsing history manipulated by different

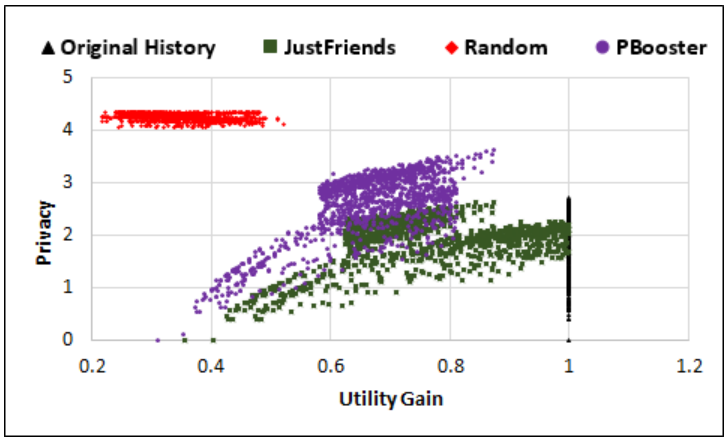


Figure 4.4: Privacy vs Utility Gain for Different Approaches.

techniques demonstrated in Fig.4.2 and Fig.4.3:

- JustFriends achieves the highest utility among all approaches while it is the most vulnerable method. Random approach is the most robust technique against de-anonymization attack, however has the most utility lost. PBooster provides high privacy but can sacrifice utility for high values of λ ($\lambda \geq 20$).

- PBooster is the most efficient approach in terms of both privacy and utility. Setting $\lambda = 10$, it returns the highest possible privacy while maintaining comparable utility with the original data.

4.4 Conclusion

The need arises for users to protect their sensitive information such as browsing history from potential adversaries. Some users resort to Tor, VPN and HTTPS to remove their traces from browsing history to assure their privacy. However, these solutions may hinder personalized online services by degrading the utility of browsing history. In this chapter, we first quantify the trade-off between user privacy and utility and then propose an efficient framework PBooster to address the problem

of anonymizing web browsing histories while retaining the utility. Our experiments demonstrate the efficiency of the proposed model by increasing the user privacy and preserving utility of browsing history for future applications.

Chapter 5

PROTECTING USER PRIVACY IN USER-ITEM INTERACTIONS DATA

Individuals in social media interact with various entities on a daily basis. Examples of these interactions are buying different products from Websites such as Amazon, making connection with different people on social media platforms, and purchasing a service. All of these interactions result in user-generated user-item interaction data. With the growth of the Web, information has increased at an unprecedented scale and therefore users face information overload problem. Recommendation systems seeks to address this challenge by suggesting relevant and reliable information that is potential of interests to online users. Therefore, recommendation systems play an important role in helping users quickly find relevant information that is buried in a large amount of irrelevant information (Koren, 2009; Beigi and Liu, 2018b; Alvari, 2017). These recommendation systems build profiles that represent user's interests (Konstan and Riedl, 2012) based on user-generated user-item interaction data and then recommend relevant items to the users based on the constructed profiles (Rashid *et al.*, 2002). Despite the effectiveness of recommendation systems, they can be sources of user privacy breach. Existing work has shown that if malicious attackers have access to the system's output and unrestricted auxiliary information about their targets, they are able to extract their entire user-item interactions history (Ramakrishnan *et al.*, 2001; Machanavajjhala *et al.*, 2011; Calandrino *et al.*, 2011; McSherry and Mironov, 2009). One main reason is that recommendation systems' outputs (i.e., product recommendation) are partially derived from other users' choices (i.e., user-item interactions history). Thus, privacy concerns arise.

One of privacy issues is the re-identification attack where a malicious adversary

attempts to infer user's actual ratings by seeking if a target user is in the database. Prior research on privacy preserving recommendation systems has extensively addressed this type of privacy breach. Common techniques include (1) modifying the output of the recommendation system algorithm so that the absence or presence of a single rating or an entire user data is masked (i.e., differential privacy based techniques) (McSherry and Mironov, 2009; Machanavajjhala *et al.*, 2011; Hua *et al.*, 2015; Zhu and Sun, 2016); and (2) coarsening the user's interactions history by adding dummy items and ratings such that the adversary cannot deduce the user's actual ratings and preferences (i.e., perturbation based techniques) (Rebollo-Monedero *et al.*, 2011; Polat and Du, 2003; Luo and Chen, 2014).

Another privacy issue is the disclosure of user private-attribute information through leaked users' interactions history (Weinsberg *et al.*, 2012). Private attribute information contains those attributes that users do not wish to disclose such as age, gender, occupation and location. This type of privacy breach is known as the private-attribute inference attack in which the adversary's goal is to infer private attributes of target users given their interactions history. Little has been done to protect users against this attack of private-attribute inference (Jia and NZhenqiang, 2018; Weinsberg *et al.*, 2012) with focus on anonymizing user-item data before publishing it. Data obfuscation comes at the cost of utility loss where utility is defined as the quality of service users receive. The existing work addresses the utility loss by minimizing the amount of changes made to the data (Jia and NZhenqiang, 2018; Weinsberg *et al.*, 2012). However, in the context of recommendation, the utility loss due to this approach can lead to degraded recommendation results. Moreover, just sharing perfectly obfuscated user-item data with a recommendation system does not necessarily prevent the adversary from inferring users' private information in future when they receive and accept new recommendations (e.g., when purchasing new products).

This research aims to devise a mechanism to counter private-attribute inference attacks in the context of recommendation systems. We propose a privacy-aware *R*ecommender with *A*ttribute *P*rotection, namely RAP (Beigi *et al.*, 2020), which offers relevant products in a way that makes any inference of user's private attributes difficult from his interactions history and recommendations. The proposed model seeks to concurrently prevent the leakage of users' private attribute information while retaining high utility for users.

Recommendation while countering private-attribute inference attack can be naturally formulated as a problem of adversarial learning (Goodfellow *et al.*, 2014). In our proposed RAP, there are two components: a Bayesian personalized ranking recommender and a private-attribute inference attacker (illustrated in Figure 5.1). The private-attribute inference attacker seeks to accurately infer users' private attribute information. The attacker aims to iteratively adapt its model with respect to the existing recommender. The recommender extracts latent representations of users and items for personalized recommendation, and simultaneously utilizes the private-attribute inference attacker to regularize the recommendation process by incorporating necessary constraints to fool the attacker. Therefore, RAP optimizes a composition of two conflicting objectives, modeled as a min-max game between recommender and attacker components. Its objective is to recommend relevant, ranked items to users such that a potential adversary cannot infer their private attribute information.

In essence, we investigate the following research issues: (1) whether we can develop a personalized privacy-aware recommendation system to guard against private-attribute inference attacks; and (2) how we can ensure that the user's private attributes are effectively obscured after receiving personalized recommendation.

Our research on these issues results in a novel framework RAP with the following main contributions:

- To the best of our knowledge, this is the first effort in proposing a recommendation system with guarding against the inference of private attribute information while maintaining the user utility.
- The proposed RAP model uses an attacker component that regularizes the recommendation process to protect users against private-attribute inference attack.
- The proposed RAP model is a general framework for recommendation systems. Both of the integrated Bayesian personalized recommender and the private-attribute attacker can be easily replaced by different models designed for specific tasks.
- We conduct experiments on real-world data to demonstrate the effectiveness of RAP. Our empirical results show that RAP preserves user utility and privacy. The results demonstrates that RAP outperforms the state-of-the-art related work and enables an adjustable balance between private-attribute protection and personalized recommendation.

5.1 Problem Statement

Before formally defining our problem, we first describe the notations used in this chapter. Let $\mathcal{I} = \{i_1, i_2, ..., i_M\}$ denotes the whole set of items, and $\mathcal{U} = \{u_1, u_2, ..., u_N\}$ denotes the whole set of users. Moreover, $\mathcal{I}_h$ represents the set of items rated by user h, and $\mathcal{R}_h$ is set of items recommended to h. $\mathcal{P} = \{p_1, p_2, ..., p_T\}$ denotes a set of T private attributes (e.g., age, gender, etc.). $\mathbf{R}$ also represents user-item rating matrix.

The goal of recommendation systems is to recommend products to people that would be interesting for them. However, we want to protect people's privacy against a malicious adversary who attempts to infer their private attribute information according to the user's list of items information. Items list $\mathcal{S}_h$ for each user h is union

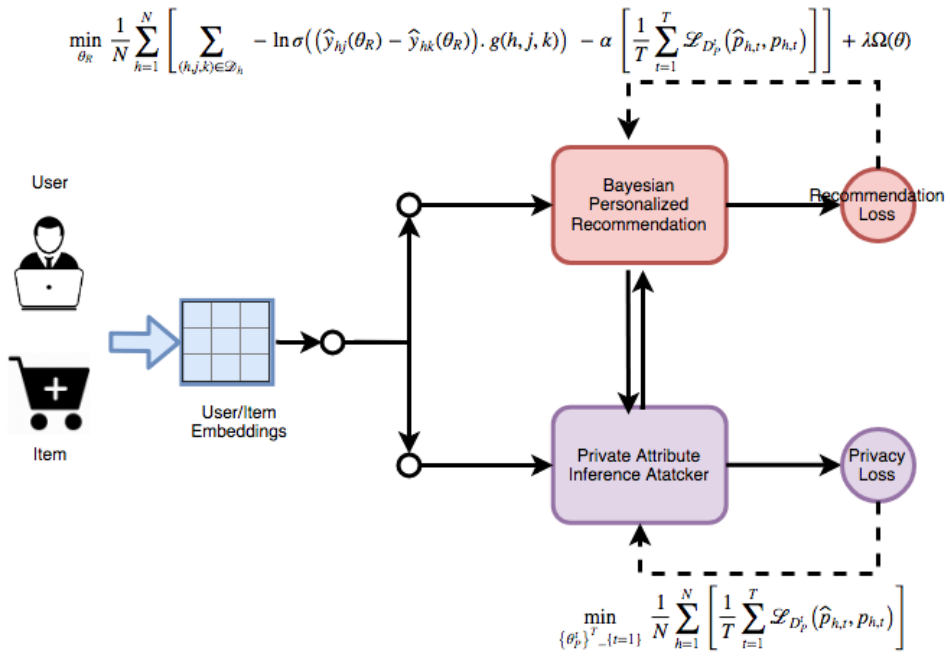


Figure 5.1: The Architecture of Recommendation with Protection (RAP) with two Components: a Bayesian Personalized Recommender and a Private-Attribute Inference Attacker.

of his previously rated and newly recommended items, i.e., $\mathcal{S}_h = \{\mathcal{I}_h \cup \mathcal{R}_h\}$. In particular, the malicious attacker has a framework which takes a target user's interactions and infers the user's private attribute. In this chapter, we study the following problem:

Problem 3. *Given a set of users $\mathcal{U}$, set of items $\mathcal{I}$, user-item rating matrix $\mathbf{R}$, set of sensitive attributes $\mathcal{P}$, we aim to learn a function f that can recommend interesting and relevant products $\mathcal{R}_h$ to each user u_h such that, 1) the adversary cannot infer the targeted user's private attribute information $\mathcal{P}$ from the user's list of items information, $\mathcal{S}_h = \{\mathcal{I}_h \cup \mathcal{R}_h\}$ and 2) the set of recommended items $\mathcal{R}_h$ is interesting for the user. The problem can be formally defined as:*

$$\mathcal{R}_h = f(\mathcal{I}_h, \mathbf{R}, \mathcal{P}) \tag{5.1}$$

Note that in this work, the goal is to protect users against a malicious adversary who have access to the users' items list, but not against the recommender itself which we assume is trusted.

5.2 Recommendation with Attribute Protection (RAP)

Our proposed recommendation framework, RAP, aims to concurrently recommend interesting items to users and protect them against private attribute leakage. The entire model is illustrated in Figure. 5.1. This framework consists of two major components, 1) a Bayesian personalized recommender, and 2) a private-attribute inference attacker. The personalized ranking recommender D_R aims to extract users' actual preferences and recommend relevant items to them. The private-attribute inference attacker D_P seeks to develop a model which can deduce users' private information w.r.t. the existing recommendation system. Recommendation component then utilizes D_P to guide the recommendation process by ensuring that the union of previously rated and newly recommended items does not leak user's attributes and further fools the adversary in D_P. Inspired by adversarial machine learning, we model this objective as a min-max game between two components, i.e. attacker D_P seeks to maximize its gain and recommender D_R aims to minimize both its recommendation loss and attacker D_P's gain. The final output of RAP for each user, is a list of top-K items which are interesting yet safe for them.

5.2.1 Bayesian Personalized Recommendation

In this section, we propose a new Bayesian personalized recommendation model. The proposed model structure is shown in Fig. 5.2. This model first extracts users and

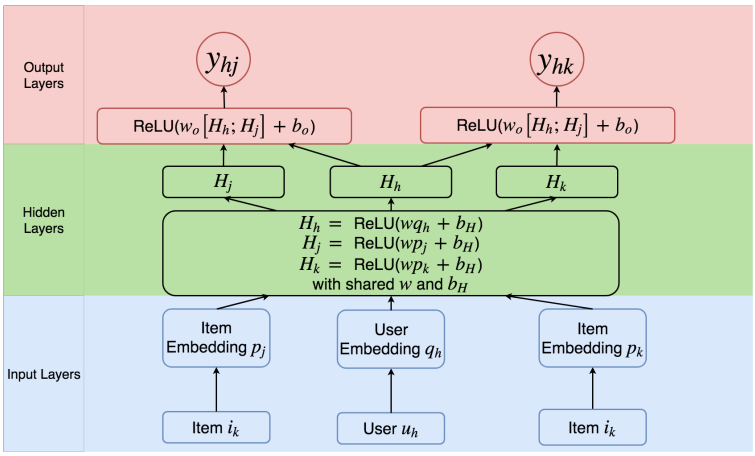


Figure 5.2: Overview of the Bayesian Personalized Recommendation Component.

items latent embeddings and then utilizes learning to rank approach to recommend items to users.

Learning to rank methods have been introduced to optimize recommendation systems toward personalized ranking. Inspired by recent success of Bayesian Personalized Ranking (BPR) (Rendle *et al.*, 2009) in image and friend recommendation systems (Niu *et al.*, 2018; Ding *et al.*, 2017), we choose BPR aver other approaches. The idea behind BPR is that observed user-item interactions should be ranked higher than unobserved ones. Learning from implicit feedback, BPR goal is to maximize the margin between an observed user-item interaction and its unobserved counterparts. In particular, BPR behavior could be interpreted as a classifier in which given a positive triplet instance of user h and items j and k, (h, j, k), it determines whether the user-item interaction (h, j) should have a higher rank score than (h, k).

This recomemndation component has three inputs, the user h and items j and k. We denote the user and items indices by a tuple of vectors (u_h, i_j, i_k) which are one-hot encodings of users and items. Since there are N users and M items, the dimensions of u_h, i_j, and i_k are M, N and N, respectively. Following the input layer, each input layer is fully connected to the corresponding embedding layer to learn the

latent representation of the users and items, $\mathbf{q}_h \in \mathbb{R}^d$, $\mathbf{p}_j \in \mathbb{R}^d$, where d is the number of dimensions. Note that the embedding dimensions for both users and items are the same. This can be formally defined as:

$$\mathbf{q}_h = \mathbf{W}_h u_h, \qquad \mathbf{p}_j = \mathbf{W}_j i_j, \qquad \mathbf{p}_k = \mathbf{W}_k i_k \tag{5.2}$$

where $\mathbf{W}_h$, $\mathbf{W}_j$ and $\mathbf{W}_k$ are embedding matrices for users and items. In the next layer, user and item embedding vectors are passed to the hidden layers H_h, H_j, and H_k for further calculations. For example, the hidden layer produces H_h for user h as:

$$H_h = ReLU(wq_h + b_H) \tag{5.3}$$

where $ReLU$ is simply defined as $ReLU(x) = \max(0, x)$ and w and b_H are the weights and bias for units, respectively.

Using H_h, H_j, and H_k, the next layer produces the user's preference $\hat{y}_{hj}$, $\hat{y}_{hk}$ toward items j and k, respectively. For example:

$$\hat{y}_{hj} = ReLU(w_o[H_h; H_j] + b_o) \tag{5.4}$$

where b_o is the bias parameter in the output layer. The activation function is $RelU$ function and $[.;.]$ represents concatenation. Note that due to the model simplicity, all users share the same latent representation learning parameters $\{w, b_H\}$ and $\{w_o, b_o\}$ in the hidden layer and output layer, respectively.

We use BPR to learn how to rank in the problem of recommendation. The final objective function is to minimize the following loss function w.r.t. θ_R:

$$\mathcal{L}_{D_R} = \frac{1}{N} \sum_{h=1}^{N} \sum_{(h,j,k) \in \mathcal{D}_h} - \ln \sigma((\hat{y}_{hj}(\theta_R) - \hat{y}_{hk}(\theta_R)).g(h,j,k)) + \lambda_{\theta_R} \|\theta_R\|^2 \tag{5.5}$$

where, $g(h, j, k)$ is the ground truth value for our model training:

$$g(h,j,k) = \begin{cases} 1, & \text{if user } u_h \text{ prefers item } i_j \text{ over item } i_k \\ -1, & \text{otherwise} \end{cases} \tag{5.6}$$

where set $\mathcal{D}_h = \{(h, j, k)|j \in \mathcal{I}_h \text{ and } k \in \mathcal{I}/\ \mathcal{I}_h\}$ also denotes the training pairwise instances in which $\mathcal{I}$ and $\mathcal{I}_h$ represent the whole set of items and the set of items rated by user u, respectively. Moreover, y_{hj} is the actual rating that user h gives to item j. θ_R is also defined as $\theta_R = \{\mathcal{W}_\mathcal{U}, \mathcal{W}_\mathcal{I}, w, b_H, w_o, b_o\}$ such that $\mathcal{W}_\mathcal{U} = \{\mathbf{W}_1, ..., \mathbf{W}_N\}$ and $\mathcal{W}_\mathcal{I} = \{\mathbf{W}_1, ..., \mathbf{W}_M\}$ represent the set of embedding matrices for N users and M items, respectively. The new proposed model considers the recommendation problem as a binary classification problem to ensure that the pairwise preference relations hold.

After training the recommendation model, given a user h, for every item j that the user has not rated, i.e., $j \in \{\mathcal{I}/\ \mathcal{I}_h\}$, his preference score $\hat{y}_{hj}$ is predicted by the recommender. In order to calculate the preference score $\hat{y}_{hj}$, we pass the tuple (h, j, j) to the recommender, and get $\hat{y}_{hj}$ and $\hat{y'}_{hj}$ as the model's output. The final preference score of user h toward item j is calculated as $\hat{y}_{hj} = 0.5(\hat{y}_{hj} + \hat{y'}_{hj})$. All of the unrated items will be then sorted based on their preference scores descendingly and the top-K items are then returned as the recommendation $\mathcal{R}_h$ to the user.

5.2.2 Training an Attacker against Inferring Private Attribute Information

The goal of our model is to recommend ranked items to users such that any potential adversary cannot infer users' private attribute information such as age, gender and occupation. However, a challenge is that the recommendation system does not know the malicious attacker's model. To address this challenge, we add a private-attribute inference attacker D_P component to our model which seeks to learn a classifier that can accurately identify the private information of users from their previous interactions. Then, we leverage this component to regularize the recommendation process by incorporating necessary constraints in order to fool the adversary D_P and further avoid the leakage of private attributes after recommendation. This part is discussed

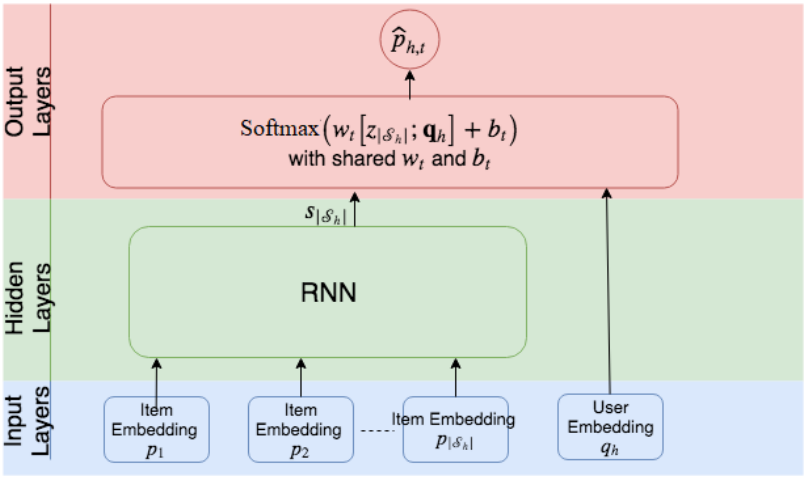


Figure 5.3: Overview of the Private-Attribute Inference Attacker Component for one Attribute.

in details in Section. 5.2.3.

The goal of the private-attribute attacker is now to predict target user h's private attribute information by leveraging the information of his latent representation as well as the the latent representation of his items list. The user h's items list $\mathcal{S}_h = \{\mathcal{I}_h \cup \mathcal{R}_h\}$ includes both items $\mathcal{I}_h$ that user has rated previously and new recommended items $\mathcal{R}_h$. Given T private attributes (e.g., age, gender), the set of $\{\theta_{P^t}\}_{t=1}^{T}$ represents all the parameters included in the private-attribute inference attacker component D_P. The output of the private attribute attacker component for user h w.r.t. t-th private attribute is the probability that user h has t-th attribute. We use $p_{h,t}$ to represent the actual value for user h's t-th private attribute. The structure of private attribute inference attacker is represented in Fig.5.3. The input to this model for each user h is the latent embedding representations of each item p_j in his items list $\mathbf{p}_j \in \mathcal{S}_h$, $j = 1, 2, ..., |\mathcal{S}_h|$ and h's latent embedding representation $\mathbf{q}_h$. Given the input, the items embeddings are passed to a single-layer recurrent neural network (RNN) and the output of RNN ($z_{|\mathcal{S}_h|}$) is then concatenated with user's embedding. The last layer

produces the predicted t-th sensitive attribute for user h, $\hat{p}_{h,t}$, which is calculated as:

$$\hat{p}_{h,t} = softmax(w_t[z_{|\mathcal{S}_h|}; \mathbf{q}_h] + b_t) \tag{5.7}$$

where $[.;.]$ represents concatenation. Also, w_t and b_t are the weights and bias for units, respectively and are shared among all users due to the model simplicity. We then minimize the private-attribute inference attacker component loss function $\mathcal{L}_{D_P^t}$ for all private attributes by seeking the optimal parameters $\{\theta_P^t\}_{t=1}^T$. The objective function for all users can be formally written as follows:

$$\mathcal{L}_{D_P} = \frac{1}{N}\sum_{h=1}^{N}\left[\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{D_P^t}(\hat{p}_{h,t}, p_{h,t})\right] \tag{5.8}$$

where $\mathcal{L}_{D_P^t}$ denotes the cross entropy loss for t-th private attribute.

5.2.3 Adversarial Learning for Recommendation with Private-Attribute Protection

Thus far, we have discussed how we 1) learn users and items representations to recommend ranked items to each user based on his personalized preferences; and 2) train an attacker which can accurately infer a target user's private attribute information given a list of his rated items and received recommendations. We stress that the adversary always has the upper hand and adapts his private-attribute inference attack in order to minimize his inference loss w.r.t. the existing recommendation system. The final objective is thus to recommend relevant ranked items to users such that a potential adversary cannot infer their private attribute information. To achieve two goals together, we design an optimization problem to minimize the recommendation loss of our model *and* maximize the inference loss of a determined attacker who adaptively minimizes his loss. Inspired by the idea of adversarial learning, we model this optimization as a min-max game between two components, Bayesian personalized recommender and private-attribute attacker.

In our proposed model, the adversary tries to adapt itself and gets the maximum gain, while the recommendation system seeks to recommend ranked items to users. The recommended items not only align well with the users' preferences, but also minimize the adversary's gain. We reformulate the objective function of the recommendation system as minimizing attacker's gain and recommendation loss simultaneously. We formalize the new objective function as follows:

$$\underbrace{\min_{\theta_R}\left(\mathcal{L}_{D_R} \overbrace{-\alpha \max_{\{\theta_P^t\}_{t=1}^T} \mathcal{L}_{D_P}}^{\text{private-attribute attacker}}\right)}_{\text{privacy-aware recommendation system}} \tag{5.9}$$

The inner part learns the most determined adversary which adaptively minimizes its loss regarding private-attribute inference given the users and items information. The outer part seeks to both minimize the recommendation loss and fool the given adversary. The parameter α controls the contribution of the private-attribute inference attacker in the learning process. Objective function in Eq. 5.9 can be written as follows:

$$\begin{aligned}\min_{\theta_R} \max_{\{\theta_P^t\}_{t=1}^T} \Bigg(\frac{1}{N}\sum_{h=1}^{N}\Bigg[\sum_{(h,j,k)\in\mathcal{D}_h} -\ln\sigma((\hat{y}_{hj}(\theta_R)-\hat{y}_{hk}(\theta_R)).g(h,j,k)) \\ -\alpha\left[\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{D_P^t}(\hat{p}_{h,t},p_{h,t})\right]\Bigg]+\lambda\Omega(\theta)\Bigg)\end{aligned} \tag{5.10}$$

where $\theta = \{\theta_R, \{\theta_P^t\}_{t=1}^T\}$ is the set of all parameters to be learned, $\Omega(\theta)$ is the L_2 norm regularizer on the parameters, and λ is a scalar to control the contribution of the regularization $\Omega(\theta)$.

5.2.4 Optimization Algorithm

The optimization process is illustrated in Algorithm 4. First, we create a mini-batch sample $\mathcal{U}_b$ of b users from the training data and serve their private attribute

Algorithm 4 The Learning Process of RAP Model

Input: Items set $\mathcal{I}$, training user data $\mathcal{U}$, training user-item matrix data $\mathbf{R}$, batch size b, θ_R, $\{\theta_P^t\}_{t=1}^T$, α, λ and K.

Output: Trained recommendation with protection RAP.

1: **repeat**

2: Create a mini-batch $\mathcal{U}_b$ of b users with their private-attribute and item-rating information from $\mathcal{U}$

3: Train the recommendation with attribute protection via Eq. 10 w.r.t. θ_R

4: For each user h in $\mathcal{U}_b$, calculate the top-K recommended items $\mathcal{R}_h$

5: Train the private-attribute inference attacker D_P (i.e., $\{\theta_P^t\}_{t=1}^T$) via Eq. 5.8 given the users' information including their list of items information, i.e., $\mathcal{S}_h = \{\mathcal{I}_h \cup \mathcal{R}_h\}$

6: **until** Convergence

and item-rating information to the model. Next, we train the Bayesian personalized recommender D_R according to the Eq. 10 w.r.t. θ_R in Line 3. Then, for each user h in $\mathcal{U}_b$ we calculate the top-K recommended items $\mathcal{R}_h$ and accordingly make his list of items information, $\mathcal{S}_h$. The private-attribute inference attacker component is then trained according to the users and item embeddings information using Eq. 5.8 in Line 5. After training RAP, for each user h, a list of top-K items $\mathcal{R}_h$ will be returned as recommendation.

5.3 Experiments

In this section we conduct experiments to evaluate the efficiency of the proposed framework in terms of both privacy and quality of the recommendation. Specifically, we aim to answer the following questions:

- **Q1** - *Privacy*: How does RAP perform in preventing leakage of users' private information?
- **Q2** - *Utility*: How does RAP perform in recommending relevant items to users?
- **Q3** - *Utility-Privacy Relation*: Does the improvement in privacy result in sacrificing the utility of recommendation system?

To answer the first question (**Q1**), we consider different private information, such as age, gender, and occupation. Then, we evaluate the effectiveness of RAP in preventing leakage of users' private information given union of users' previously rated and newly recommended items. Addressing leakage of private attribute information may result in recommendation performance deterioration. Therefore, to answer the second question (**Q2**), we examine the performance of RAP in terms of the quality of the recommendation. Finally, to answer the third question (**Q3**), we investigate the loss in recommendation performance when enhancing privacy of users. Next, we discuss the dataset, experimental setup and results.

5.3.1 Data

We use publicly available data MovieLens (Harper and Konstan, 2016). MovieLens is collected by the GroupLens Research Project at the University of Minnesota (Harper and Konstan, 2016). This dataset includes $100,000$ ratings by 943 users on 1,682 movies. Each user has rated at least 20 movies and the rating scores are between 1 and 5. In the collected dataset, each user is associated with three private attributes, gender (male/female), age, and occupation. For this chapter, we follow the setting of (Hovy and Søgaard, 2015) and categorize age attribute into three groups, over-45, under-35, and between 35 and 45. In total, 21 possible occupations have been considered for this data. The average number of rated items for each user

is 129.

5.3.2 Experimental Setting

Here, we first explain how we design experiments to evaluate utility and privacy. Then, we discuss evaluation metrics and baselines.

Implementation Details: The parameters for recommendation and attacker components are determined through grid search. For the Bayesian personalized ranking recommendation component, we set the dimension of first layer as $d = 70$. Accordingly, size of user and item embedding vectors is $d = 70$. The dimension of hidden layer is also set as 20. For the private-attribute inference attacker component, we use single layer RNN with the dimension of input layer set as $d = 70$. User and item embeddings are then passed from recommendation component to the attacker component. The dimension of hidden layer is set as 100. The parameters α and λ are also determined through cross-validation, $\alpha = 1$ and $\lambda = 0.01$.

We initialize the weight matrices in both components with random values uniformly distributed in $[0, 1]$. The error gradient is back propagated from output to input and parameters in each layer are updated. The optimization algorithm used for gradient update is Adam's algorithm (Kingma and Ba, 2014). The loss generally converges after 20 epochs. The batch size we use in experiments is $b = 32$.

Recommendation Evaluation: We evaluate the performance of recommendation by examining the quality of recommended items for all users. We follow the setting of (Jia and NZhenqiang, 2018) to set-up the experimental settings. To do so, we split the data for train and test as follows. For each user h in the data, we randomly select l rated items for test set and the remaining $n_h - l$ items for training set, where n_h is the number of rated items for user h. We set the item rating for those in the test set as zero. We vary the value of l as $\{35, 40, 45\}$. Then, the top-K items are then

returned to each user as the recommendation. Note that we assume RAP has access to the users' private attribute information during the training process.

Private-Attribute Evaluation: We evaluate privacy of users in terms of their robustness against the malicious attribute inference attacks in which the adversary's goal is to infer users' private attributes. In particular, the malicious attacker learns a multi-class classifier which takes a target user h list of items information, i.e. $\mathcal{S}_h = \{\mathcal{I}_h \cup \mathcal{R}_h\}$, where $\mathcal{I}_h$ is set of h's rated items and $\mathcal{R}_h$ is set of items recommended to h. The adversary then infers the user's private attributes, i.e., gender, age, and occupation.

We use a Neural Network (NN) model as the adversary's classifier. Note that RAP is not aware of the adversary's model. In this attack, the adversary deploys a feed-forward network with a single hidden layer to perform the attack. The input to this model is one-hot encoding of each user, $\mathcal{S}_h = \{\mathcal{I}_h \cup \mathcal{R}_h\}$. Since there are M items in the dataset, the dimension of input vector is M. The input layer is then fully connected to the hidden layer with dimension of hidden state set as 100 and a $softmax$ layer used as the output layer. The dimension of the hidden layer is determined through grid search. We note that Gong et al. (NZhenqiang and Liu, 2016) also proposed an attribute inference attack which leverages both social friends and rating behavior. However, their attack is not applicable to our problem as we focus on leveraging only user-item rating information.

We follow the setting of (Jia and NZhenqiang, 2018) to set-up the experimental settings. We split the data to train and test sets by sampling 80% of the users in the dataset uniformly at random as the training set and use the remaining users as testing set. We assume that the users in the training set has publicly disclosed their private information while the users in the testing set keep those attribute information private. Then, for each user in the test set, we randomly select l rated items and

remove them from the user's rating history by setting the their rating as zero. We keep the user-item ratings for users in the training set intact (i.e., original user-item ratings). Next, the trained RAP model is deployed on the users in the test set and top-l recommended items $\mathcal{R}_h$ are added to the users' previously rated items $\mathcal{I}_h$, in order to make $\mathcal{S}_h = \{\mathcal{I}_h \cup \mathcal{R}_h\}$. We vary the value of l as $\{35, 40, 45\}$.

The adversary's classifier is trained on the training set and evaluated on the users in the test set. Note that we assume that the malicious attacker knows the original intact user-item interactions for those users in the training set and seeks to predict private attribute information of the users in the test set, given their $\mathcal{S}_h$. We evaluate a malicious attack for each private attribute.

Evaluation Metrics: We use the following metrics for evaluating RAP performance w.r.t. malicious private-attribute inference (i.e., privacy) and product recommendation (i.e., utility):

- **Private-Attribute Evaluation:** We report micro-AUC (Fawcett, 2006) of the adversary's classifier. The reason is that the distribution of data for different private attribute values is imbalance, and thus micro-AUC (Fawcett, 2006) gives a more accurate assessment of attribute inference attack. Note that lower value of AUC demonstrates that RAP provides higher privacy for users in terms of obscuring their private attribute information.

- **Recommendation Evaluation:** We use standard metrics that are widely used in other related works (Ziegler *et al.*, 2005), i.e., $P@K$ and $R@K$. $P@K$: $P@K$ represents the ratio of test cases which has been successfully recommended in a top-K position in a ranking list to value of K. For each user, we measure $P@K$ as:

$$P@K = \frac{|\{\text{test items}\} \cap \{\text{top-}K \text{ returned items}\}|}{K} \tag{5.11}$$

  $R@K$: $R@K$ defines the ratio of top-K recommended items which are in the test

set to the number of items to be recommended in the test. For each user in the data, we measure $R@K$ as follows:

$$R@K = \frac{|\{\text{test items}\} \cap \{\text{top-}K \text{ returned items}\}|}{|\{\text{test items}\}|} \tag{5.12}$$

We then report the average of $R@K$ and $P@K$ for all users in the dataset and set the number of returned items as $K = 35$.

Baseline Methods: We compare RAP in terms of both recommendation (utility) and preserving private attributes (privacy) with the following baselines:

- ORIGINAL: This baseline is a variant of RAP which recommends items for each user without incorporating the private-attribute inference attacker component, i.e., $\alpha = 0$.

- LDP-SH (Bassily and Smith, 2015): This method is based on ϵ-differential privacy to protect privacy of an individual user's data record, i.e. user-item ratings. It adds noise to the user's ratings such that two arbitrary users' records have close probabilities to generate the same noisy data. This method requires categorical data which for our case, each user-item rating can be viewed as categorical data taking values $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$. We compare our model against this method to 1) investigate the effect of differential privacy on preventing leakage of private attribute information; and 2) examine the utility loss in recommendation system comparing to our model. Note that this method does not consider quality of recommendation service in practice.

- BLURME (Weinsberg *et al.*, 2012): This method perturbs user-item rating matrix before sending to recommendation system. In particular, for each user, it adds new items to the user's ratings $\mathcal{I}_h$ that are negatively correlated with the user's actual private attribute value and then adds the average rating score to those

items. Note that BlurMe needs to be deployed for each attribute separately. We use this method to see whether or not coarsening user data before sharing it with recommender can prevent leakage of users' private information when they receive new recommendations in future.

To have a fair comparison between our proposed model Rap and two baselines BlurMe and LDP-SH, we anonymze the user-item rating data according to these models. Then, we use the noisy manipulated data to train the recommendation model. We use matrix factorization model as the recommendation framework for both baselines. The procedure discussed in Section 5.3.2 is then used to evaluate the final results.

5.3.3 Privacy Analysis (Q1)

The experimental results against the introduced malicious private-attribute inference attack (Section 5.3.2) for different methods are demonstrated in Table. 5.1. We observe that increasing the number of test items (l) results in decrease of AUC score for all frameworks. This is because for each target user h in the test set, l recommended items $\mathcal{R}_h$ have been added to user's item list $\mathcal{S}_h$. Therefore, increase of l can decrease the malicious attacker's chance for correctly inferring users' private attribute information.

Moreover, Rap has significantly lower AUC score in comparison to Original for all three private attributes and thus outperforms Original in terms of obscuring users' private attribute information. Rap also has significantly better performance in hiding private information in comparison to LDB-SH. The reason is that LDB-SH aims to achieve a privacy goal that is different from preventing leakage of private information. This confirms that although adding noise and satisfying ϵ-differential privacy can indirectly benefit private attribute leakage, it does not directly target this

problem. These results show the importance of private-attribute inference attacker component in obfuscating private information. We also observe that RAP hides more private information rather than BLURME (lower AUC score). This demonstrates that providing obfuscated user-item rating data to the recommendation system, does not necessarily guarantee preventing future private attribute leakage when user receives (and accordingly buy) more recommended products. Moreover, BLURME needs to be deployed for each private attribute separately while RAP considers three private attributes all together.

Table 5.1: Attribute Inference AUC Score for Different Private Attributes. Lower AUC Score Values Indicate Higher Privacy.

Model	# test items (l)								
	35			40			45		
	Gen	Age	Occ	Gen	Age	Occ	Gen	Age	Occ
ORIGINAL	0.7662	0.7050	0.8332	0.7662	0.7050	0.8332	0.7662	0.7050	0.8332
LDP-SH	0.6587	0.6875	0.8076	0.6440	0.6777	0.7954	0.6398	0.6732	0.7817
BLURME	0.6266	0.6177	0.7614	0.6013	0.5949	0.7589	0.5884	0.5901	0.7522
RAP	**0.6039**	**0.5397**	**0.7319**	**0.5714**	**0.5270**	**0.7315**	**0.5278**	**0.5262**	**0.7312**

These results confirm the efficiency of RAP in obscuring users' private attribute information and demonstrate that despite the fact that RAP is not aware of the adversary's inference model, it is prepared against the malicious attacker.

5.3.4 Utility Analysis (Q2)

The results for recommendation task for different methods and different number of test items (l) are shown in Table. 5.2. We observe that increasing the number of

Table 5.2: $P@K$ and $R@K$ Scores for Evaluating Recommendation Systems. Higher $P@K$ and $R@K$ Score Values Show the Higher Quality of Recommendation System (i.e., Utility)

Model	# test items (l)					
	35		40		45	
	$P@K$	$R@K$	$P@K$	$R@K$	$P@K$	$R@K$
Original	0.156	0.156	0.151	0.172	0.145	0.187
LDP-SH	0.071	0.071	0.062	0.078	0.055	0.081
BlurMe	0.118	0.118	0.109	0.134	0.0997	0.150
Rap	**0.152**	**0.152**	**0.147**	**0.168**	**0.142**	**0.183**

test items (l) results in increasing $R@K$ and decreasing $P@K$ for all methods. Note that the higher the $P@K$ and $R@K$ score values are, the higher recommendation quality is.

Another observation is that LDP-SH has the worst performance amongst all methods, i.e., lowest $P@K$ and $R@K$ scores. This is because of the way LDP-SH adds noise to the user data without considering the quality of recommendation service in practice which can result in degraded recommendation results. BlurMe has also lower performance than Rap as it neglects quality of recommendation results. These results confirm the effectiveness of Bayesian personalized recommendation component which helps Rap to take the utility into consideration in practice. Moreover, quality of recommendation results for Rap method is comparable to the Original approach. This means that Rap can accurately capture users' actual preferences and interests (i.e., high utility).

The results confirm the effectiveness of Rap in understanding users' actual pref-

erences and recommending ranked relevant products that are interesting yet safe products to users.

5.3.5 Utility-Privacy Relation (Q3)

To understand the relation between utility and privacy, we compare the malicious private-attribute inference attack AUC score and recommendation performance for all methods, based on the results in Tables 5.1 and 5.2. We observe that LDP-SH has the worst results in terms of both preserving privacy and recommendation performance. Another observation is that BlurMe improves privacy compared to the Original method, but it loses utility in terms of recommendation system performance. This is in contrast with the results of Rap, which has outperformed BlurMe and LDP-SH in terms of recommendation and has comparable results with Original. Rap has also achieved the lowest AUC score and therefore highest privacy among all other methods.

Comparing Rap with other methods confirms that approaching utility loss by minimizing the amount of data changes results in loss of quality of recommendation system in practice. This is reflected as degraded recommendation results for baseline approaches. Moreover, these results confirm the effectiveness of Bayesian personalized recommendation component in our proposed model Rap, which helps us to consider quality of recommendation in practice. Results also demonstrate the complementary roles of both recommendation and private attribute components which guide each other through both privacy and utility issues. This results in a privacy-aware recommendation system which is prepared for private attribute inference attack and understands users' actual preferences as well.

Table 5.3: Impact of Different Private-Attribute Attacker Components on RAP for Private-Attribute Inference Attack. Lower AUC Indicates Higher Privacy.

Model	# test items (l)								
	35			40			45		
	Gen	Age	Occ	Gen	Age	Occ	Gen	Age	Occ
RAP	0.6039	**0.5397**	**0.7319**	0.5714	**0.5270**	**0.7315**	0.5278	**0.5262**	**0.7312**
RAPAGE	0.6450	0.5948	0.7528	0.5489	0.5938	0.7522	0.5475	0.5909	0.7497
RAPGEN	**0.5332**	0.6789	0.7558	**0.5298**	0.6614	0.7556	**0.5211**	0.6415	0.7555
RAPOCC	0.6571	0.6949	0.7468	0.6485	0.6871	0.7466	0.6454	0.6853	0.7438

5.3.6 Impact of Different Components

Here, we investigate the impact of different private attribute components on obscuring users' private information. We define three variants of our proposed framework, i.e., RAPAGE, RAPGEN, and RAPOCC. In each of these variants, the model is trained with the corresponding private-attribute inference attacker component, e.g. RAPAGE is trained solely with age inference attacker component and does not utilize any other private-attribute attackers during training phase. Results for attribute inference attack and recommendation tasks are shown in Table 5.3 and Table 5.4, respectively. We observe that for gender attribute, RAPGEN has the best performance in terms of obscuring gender attribute comparing to the other approaches (i.e., lowest AUC score). This is in contrast to quality of RAPGEN performance for recommendation task which is lower than original proposed model RAP. For other private attributes, RAP still outperforms RAPOCC and RAPAGE in terms of obscuring age and occupation attributes. Moreover, results show that using one private-attribute attacker compromises the effectiveness model for obfuscating other private attributes.

Table 5.4: Impact of Different Private-Attribute Attacker Components on RAP for Recommendation Task. Higher $P@K$ and $R@K$ Values Show Higher Quality of Recommendations.

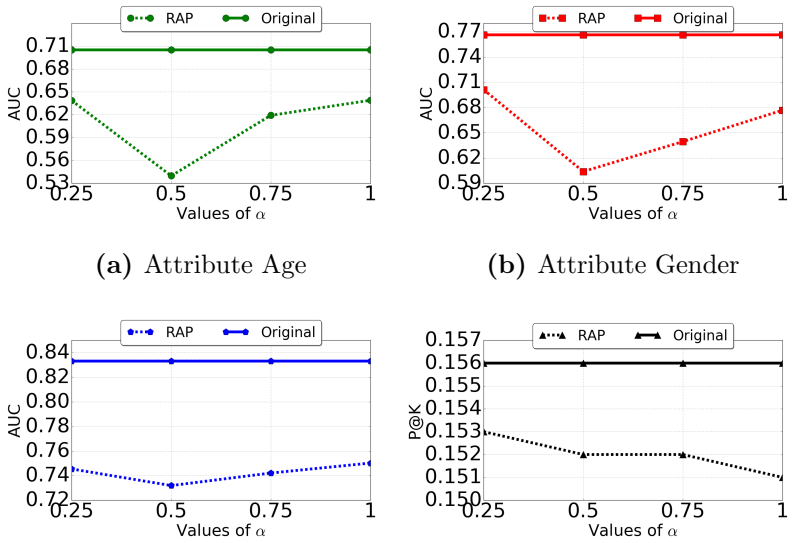
Model	# test items (l)					
	35		40		45	
	$P@K$	$R@K$	$P@K$	$R@K$	$P@K$	$R@K$
RAP **0.152**	**0.152**	**0.147**	**0.142**	**0.183**		
RAPAGE	0.150	0.150	0.146	0.167	0.141	0.182
RAPGEN	0.151	0.151	0.145	0.166	0.141	0.181
RAPOCC	0.147	0.147	0.141	0.161	0.135	0.174

For the recommendation task, we surprisingly observe that using solely one of the private-attribute attackers in training process can result in performance reduction in comparison to RAP in terms of $P@K$ and $R@K$. This means that focusing merely on obscuring one private attribute can result in more recommendation performance degradation.

5.3.7 Probing Further

RAP has one important parameter α which controls the contribution from private-attribute attacker component. In this section, we probe further to investigate the effect of this parameter by varying it as $\{0.25, 0.5, 0.75, 1\}$. For this experiment, we set the number of test items $l = 35$. We also set the number of top-K returned items as $K = 35$ for calculating $P@K$. Note that $P@K$ and $R@K$ are equal in this scenario as $K = l = 35$. Results are shown in the Fig. 5.4.

Although α controls the contribution of private-attribute inference attacker com-



(a) Attribute Age **(b)** Attribute Gender

(c) Attribute Occupation **(d)** Recommendation

Figure 5.4: Performance Results for Private-Attribute Inference Attack and Recommendation Task for Different Values of α

ponent, we surprisingly observe that with the increase of α, the AUC score for attribute inference attack decreases at first up to the point that $\alpha = 0.5$ and then it increases. This means that private information were obscured more accurately at the beginning with the increase of α and less later. Moreover, with the increase of α, the performance of recommendation task decreases, i.e., lower $P@K$. This shows that increasing the contribution of private-attribute attacker component leads to decrease in the quality of recommendation framework. Another observation is that setting $\alpha = 0.25$ leads to improvement in hiding private attribute information in comparison to the results of using ORIGINAL (or when $\alpha = 0$). This result shows the importance of the RAP's private-attribute attacker component in preserving privacy of users. Another observation is that after $\alpha = 0.5$, continuously increasing α increases the AUC for malicious private-attribute inference attack, i.e., degrades the performance of hiding private information. The reason is that the model could overfit by increas-

ing the value of α and therefore leads to an inaccurate estimation in terms of privacy protection.

5.4 Conclusion

In this chapter, we propose an adversarial learning-based recommendation with attribute protection model, RAP, which guards users against private-attribute inference attack while maintaining utility. RAP recommends interesting yet safe products to users such that a malicious attacker cannot infer their private attribute from users' interactions history and recommendations. RAP has two main components, Bayesian personalized recommender, and private-attribute inference attacker. Our empirical results show the effectiveness of RAP in both protecting users against malicious private-attribute inference attacks and preserving quality of recommendation results. RAP also consistently achieves better performance compared to the state-of-the-art related work.

Chapter 6

PROTECTING USER PRIVACY IN TEXTUAL DATA

Textual information is one of the most significant portions of data that users generate by participating in different online activities. On one hand, textual data consists of abundant information about users' behavior which is critical for understanding individuals by profiling them at unprecedented scales. It has also been used in many tasks such as sentiment analysis (Zafarani *et al.*, 2014; Beigi *et al.*, 2016a), part-of-speech tagging (Hovy *et al.*, 2015) and information extraction and retrieval (Zafarani *et al.*, 2014). On the other hand, the textual data itself contains sufficient information that allows people in the textual database to be re-identified (Zhang *et al.*, 2018) and leaks their private attribute information (Mukherjee and Liu, 2010; Beretta *et al.*, 2015; Volkova *et al.*, 2015). Thus, "you are what you write" as the saying goes. Take the following tweet as an example:

Dr.appt Tuesday morning was told I need to lose 30 pounds by X-Mas, have ***high cholesterol****, and* ***high blood pressure****. Today starting counting calories #myfitnesspal and juicing for dinner* [1]

This user may not be aware that the sensitive medical condition information can be easily inferred from this post– exposing symptoms of Diabetes. Users' sensitive and private information that they do not wish to disclose such as vacation plans, medical conditions, age and location can be thus easily inferred from text (Beretta *et al.*, 2015; Hovy *et al.*, 2015). Private attribute information are usually implicitly hidden in the textual information. Take the following reviews from TrustPilot product review website as examples:

[1]The tweet is real, however, we altered it to preserve the privacy of the user.

Review 1: *It was recently **my daughter's** birthday and after speaking with her and getting the usual " I don't want or need anything **Dad** ", I decided to send her a vase full of roses. Well..WOW !! , was she surprised and so delighted!*

Review 2:*I ordered heely's for xmas and when they came they were too small so i sent them back for an exchange ... I have actually recommended Rawk to **one of the mums at my childs swimming class** and would recommend in the future. Rawk your service is the best i have ever had before, keep up the good work.*

The gender and age of the user who has written the first review could be easily inferred from keywords such as *daughter's birthday*, *Dad*– user is male and is over 45 years old. Second review also reveal the gender of the user which is probably female as the user communicates with other *mums*. The review also indicates the user's age which is probably less than 40 since the user has young kids whom going to swimming classes.

Another privacy issue arises when a malicious data consumer (or any potential adversary) attempts to re-identify the identity of an individual in the database by investigating whether a targeted user's textual data is in the database or inferring which record is associated with it. Therefore, publishing complete and intact users' textual data risks exposing their privacy by allowing an adversary to figure out *what* they are.

These users' privacy concerns, therefore, mandate data publishers to protect privacy by anonymizing the data. The ultimate goal of an anonymization approach is to preserve user privacy while ensuring the utility of the published data for future tasks. One straightforward technique is to remove "Personally Identifiable Information" (a.k.a. PII) such as names, age and location information. This solution has shown to be insufficient to protect people's privacy. The reason is that private attributes are usually hidden in the textual information as we see in aforementioned reviews

and tweet examples and it is challenging to find exact pieces of textual information which implicitly reveal these sensitive information. Other examples of insufficiencies are the anonymized dataset published for the Netflix prize challenge (Narayanan and Shmatikov, 2008) and the AOL search data leak (Barbaro *et al.*, 2006) in which users were re-identified according to their reviews and search queries, respectively. Various protection techniques for structured data have been developed over the years such as k-anonymity (Sweeney, 2002) and differential privacy (Dwork, 2008). However, traditional privacy preserving techniques are inefficient for user-generated textual data because this data is highly unstructured, noisy and unlike traditional documental content, consists of large numbers of short and informal posts (Fung *et al.*, 2010). Moreover, these works may impose a significant utility loss for protecting textual data as they may not explicitly include utility into the design objective of the privacy preserving model. It is thus challenging to design effective anonymization techniques for user-generated textual data.

To address the aforementioned challenges, we propose a double privacy preserving text representation learning framework, called DPTEXT (Beigi *et al.*, 2019c,d). The proposed framework seeks to learn a privacy preserved text representation so that 1) a potential adversary cannot infer whether or not a target text representation is in the dataset, 2) the adversary cannot deduce users' private attribute from the learned representation, and 3) the semantic meaning of the original textual information is still preserved in the learned representation. Our double privacy preserving framework protects individuals' privacy against identity re-identification and leakage of private information. Inspired by the recent success in adversarial learning (Goodfellow *et al.*, 2014), we build DPTEXT through an integrated process which consists of an auto-encoder, a differential-privacy-based noise adder and two discriminator-learning components (illustrated in Figure 6.1). We deploy a document auto-encoder

to extract latent representation of the original text's content. The noise adder then adds noise to the text representation by adopting a Laplacian mechanism in order to guarantee differential privacy. Although guaranteeing differential privacy minimizes the chances of revealing whether or not a target text representation is in the database, it cannot prevent the adversary from learning user's private information. Moreover, adding too much noise can destroy the semantic meaning of the textual information. To infer the amount of added noise w.r.t. these constraints, we utilize two discriminators that regularize the noise adding process by incorporating necessary constraints. First, we incorporate a *semantic discriminator* to ensure that the semantic meaning of the perturbed text representation is preserved w.r.t. the given task (e.g., classification). Second, we introduce a *private attribute discriminator* to ensure that the perturbed representation does not contain private attributes.

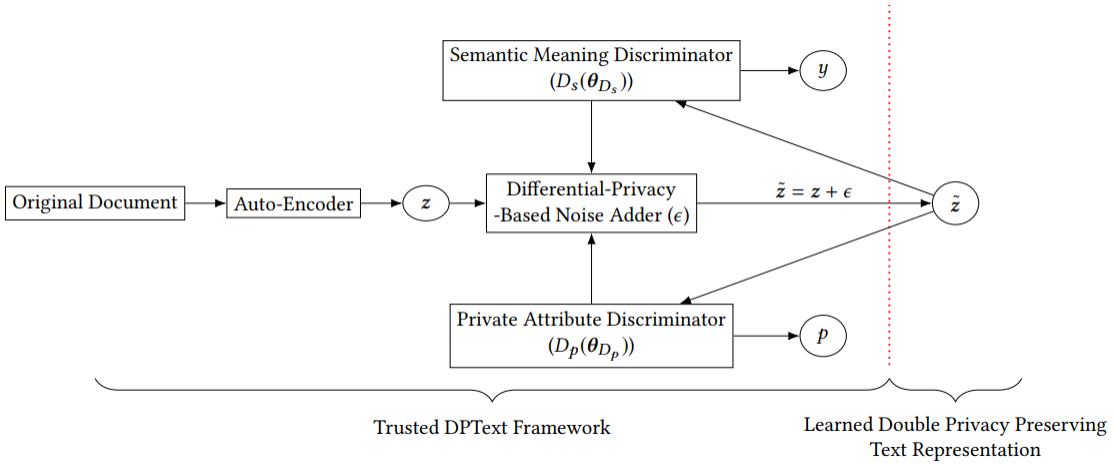


Figure 6.1: The Framework of DPTEXT Architecture. Red Dashed Line Shows the Privacy Barrier and Everything to the Left of it (i.e., the Original Data and Intermediate Results) are Kept Private.

In essence, we investigate the following challenges: 1) How should textual representation be perturbed to ensure that differential privacy is preserved?, 2) How could

we control the amount of the added noise so that the semantic meaning of the text is preserved with respect to the given task? and 3) How could we handle the amount of the added noise so that the user's private attributes are obscured? Our solution to these challenges results in a novel framework DPTEXT. Our main contributions are summarized as:

- We study the problem of text annonymization by learning a differentially private representation that prevents text reconstruction and re-identification by minimizing the chance of attacker to infer whether target text representation is in the database;
- We provide a principled way to learn a textual representation that does not contain users' private attribute information while retaining the utility for a given task; and
- We theoretically show that the learned representation is differentially private which confirms DPTEXT minimizes the re-identification chance. We also conduct experiments on real-world datasets to demonstrate the effectiveness of DPTEXT in two important natural language processing tasks, i.e., sentiment prediction and part-of-speech (POS) tagging. Our empirical results show that DPTEXT is able to keep the semantic meaning while obscuring private attribute information.

6.1 Problem Statement

Let $\mathcal{X} = \{x_1, ..., x_N\}$ denotes a set of N documents and $\mathcal{P} = \{p_1, ..., p_T\}$ denotes a set of T private and sensitive attributes. Each document x_i is composed of a sequence of words, i.e., $x_i = \{x_i^1, ..., x_i^m\}$. We denote $\mathbf{z}_i \in \mathbb{R}^{d \times 1}$ as the latent representation of the original document x_i. We would like to use x_i in the given task $\mathcal{T}$ (e.g., classification). However, we want to preserve users' privacy by preventing a potential adversary from inferring whether a target text representation is in the dataset or which record is associated with it or being able to learn the target users' private

attribute information. Thus, in this chapter, we study the following problem:

Problem 4. *Given a set of documents $\mathcal{X}$, set of sensitive attributes $\mathcal{P}$, and given task $\mathcal{T}$, learn a function f that can generate and release a manipulated latent representation $\tilde{\mathbf{z}}_i$, for each document x_i so that, 1) the adversary cannot re-identify a targeted text representation and infer whether or not this latent representation is in the database, 2) the adversary cannot infer the targeted user's private attributes $\mathcal{P}$ from the generated representation $\tilde{\mathbf{z}}_i$, and 3) the generated representation $\tilde{\mathbf{z}}_i$ is good for the given task $\mathcal{T}$, i.e., $\tilde{\mathbf{z}}_i = f(x_i, \mathcal{P}, \mathcal{T})$.*

Note that in our work, the goal is to achieve a protection against possible attacks of malicious data consumers who have access to the released textual information, but not against the system (i.e., text representation learner) which we assume is trusted.

6.2 The Proposed Framework

Here, we discuss the details of double privacy preserving text representation learning framework. We illustrate the entire model in Figure 6.1. This framework consists of four major components: 1) an auto-encoder for text representation, 2) differential-privacy-based noise adder, 3) a semantic meaning discriminator, and 4) a private attribute discriminator. The auto-encoder A aims to learn the content representation of a document by minimizing the reconstruction error. Then, the differential-privacy-based noise adder adds a random noise, i.e., Laplacian noise, to the original text representation w.r.t. a given privacy budget to further satisfy the differential privacy guarantee. Since adding noise neither preserves semantic meaning nor necessarily prevents leakage of private attributes, semantic meaning and private attributes discriminators are utilized to infer the amount of the added noise. The semantic meaning discriminator D_S ensures that the added noise does not destroy the semantic meaning

w.r.t. a given task. The private attribute discriminator D_P also guides the amount of added noise by ensuring that the manipulated representation does not include users' private information. Note that we assume that the framework is trusted and therefore everything to the left of the privacy barrier (the red dashed line in Figure 6.1) including the original textual information and intermediate results, are kept private. The final learned representation which is to the right of the privacy barrier is released to the public. The final output 1) is differentially private, 2) obscures private attribute information, and 3) preserves semantic meaning.

6.2.1 Extracting Textual Representation

Here, we demonstrate how to extract the content representation for a given document. Let $x = \{x^1, ..., x^m\}$ be a textual document with m words. Auto-encoder has been widely utilized for text generation and has shown to be effective recently (Bowman *et al.*, 2015; Cho *et al.*, 2014). We therefore use an auto-encoder A to extract content representation $\mathbf{z}$ from document x. Let $E_A : \mathcal{X} \rightarrow \mathcal{Z}$ be an encoder that can infer the content representation $\mathbf{z}$ for a given document x, and $D_A : \mathcal{Z} \rightarrow \mathcal{X}$ be a decoder that reconstruct the document from its learned representation.

Recurrent neural networks (RNN) has been shown to be effective for summarizing and learning semantic of unstructured noisy short texts (Cho *et al.*, 2014; Shang *et al.*, 2015). In this work, we apply RNN as the encoder to learn the latent representation of texts. RNN can learn a probability distribution over a sequence by being trained to predict the next symbol in a sequence. The RNN consists of a hidden state S and an optional output which operates on a word sequence $x = \{x^1, ..., x^m\}$. At each time step t, the hidden state s_t of RNN is updated by,

$$s_t = f_{enc}(s_{t-1}, x^t) \tag{6.1}$$

After reading the end of the given document, we use the last hidden state of the RNN as the representation vector $\mathbf{z} \in \mathbb{R}^{d \times 1}$ of the document x. We employ the gated recurrent unit (GRU) as the cell type to build the RNN, which is designed in a manner to have a more persisted memory (Cho *et al.*, 2014). Let θ_e denotes the parameters for the encoder E_A. Then we will have:

$$\mathbf{z} = E_A(x, \theta_e) \tag{6.2}$$

Decoder $\hat{x} = D_A(\mathbf{z}, \theta_d)$ takes $\mathbf{z}$ as the input to start the generation process and θ_d denotes the parameters for the decoder D_A. We use another RNN to build the decoder D_A to generate the output word sequence $\hat{x} = \{\hat{x}^1, .., \hat{x}^m\}$. At each time step t, the hidden state of the decoder is computed as:

$$s_t = f_{dec}(s_{t-1}, \hat{x}^t) \tag{6.3}$$

where $s_0 = \mathbf{z}$. The word at step t is predicted using a softmax classifier:

$$\hat{x}^t = softmax(\mathbf{W}^{(S)} s_t) \tag{6.4}$$

where $softmax(.)$ is a softmax activation function, $\mathbf{W}^{(S)} \in \mathbb{R}^{|\mathcal{V}| \times (d+k)}$ with $d+k$ as the dimension of the hidden state in each layer, and $\hat{x}^t \in \mathbb{R}^{|\mathcal{V}|}$ is a probability distribution over the vocabulary. Here $\mathcal{V}$ denotes a fixed vocabulary set with size $|\mathcal{V}| = K$. We define $\hat{x}^{t,j}$ as the probability of choosing j-th word $v_j \in \mathcal{V}$ as:

$$\hat{x}^{t,j} = p(\hat{x}^t = v_j | \hat{x}^{t-1}, \hat{x}^{t-2}, ..., \hat{x}^1) \tag{6.5}$$

We can thus define the probability of generating an output sequence $\hat{x} = \{\hat{x}^1, .., \hat{x}^m\}$ given the input document x as:

$$p(\hat{x}|x, \theta_d) = \prod_{t=1}^{t=m} p(\hat{x}^t \hat{x}^{t-1}, \hat{x}^{t-2}, ..., \hat{x}^1, \mathbf{z}, \theta_d) \tag{6.6}$$

The two components of the proposed auto-encoder are jointly trained to minimize the negative conditional log-likelihood for all documents. The loss function is defined as:

$$\mathcal{L}_{\text{auto}} = -\sum_{i=1}^{m} \log p(\hat{x}^i | x^i, \theta_d, \theta_e) \tag{6.7}$$

where θ_e and θ_d are the set of model parameters for the encoder and decoder, respectively. We use the trained auto-encoder E_A to obtain the content representation $\mathbf{z} \in \mathbb{R}^{d \times 1}$ according to Eq. 6.2 where d is the size of textual representation.

6.2.2 Preventing Text Re-identification and Reconstruction by Adding Noise

Textual information is rich in content and publishing this data without proper anonymization lead to privacy breach and revealing the identity of an individual. This can let the adversary infer if a targeted user's latent textual representation is in the database or which record is associated with it. Moreover, publishing a document's latent representation could result in leakage of the original text. In fact, recent advancement in adversarial machine learning shows that it is possible to recover the input textual information from its latent representation (Hitaj *et al.*, 2017). In this case, if an adversary has preliminary knowledge of the training model, they can readily reverse engineer the input, for example, by a GAN attack algorithm (Hitaj *et al.*, 2017). It is thus essential to protect the textual information before publishing it.

Differential privacy is a powerful technique for preserving privacy of users' data included in a database and provides a privacy guarantee. Our method is inspired by Chaudhuri et al. (Chaudhuri *et al.*, 2011), where the differential privacy is achieved through adding a random noise, i.e., Laplacian noise, to the output of an algorithm $\mathcal{A}$. This mechanism is known as *output perturbation* and it has been proved that under certain conditions this output perturbation mechanism will guarantee differential

privacy (Chaudhuri *et al.*, 2011).

The main idea of the output perturbation mechanism is to add noise to the output of an algorithm to preserve its privacy. In our problem, the output is the original document latent representation $\mathbf{z}$. The benefit of adding noise to this latent representation is two fold. First, it minimizes the chance of the re-identification of learned text representation by preventing the adversary to infer whether or not a target representation is in the database, and second, it makes it difficult for the adversary to recover the raw textual data. The goal here is thus to add noise to the output such that the differential privacy condition is satisfied. Laplacian mechanism is a popular way to add noise to preserve differential privacy. In particular, with Laplacian mechanism, we perturb the output $\mathbf{z}$ by adding Laplacian noise to it as follows:

$$\tilde{\mathbf{z}}(i) = \mathbf{z}(i) + \mathbf{s}(i), \quad \mathbf{s}(i) \sim Lap(b), \quad b = \frac{\Delta}{\epsilon}, \quad i = 1, .., d \tag{6.8}$$

where ϵ is the privacy budget, Δ is the L_1-sensitivity of the latent representation $\mathbf{z}$, d the dimension of $\mathbf{z}$, $\mathbf{s}$ the noise vector, $\mathbf{s}(i)$ and $\mathbf{z}(i)$ are the i-th element for vectors $\mathbf{s}$ and $\mathbf{z}$, respectively. $\Delta = 2d$ (see details in Section 6.3). Note that each element of the noise vector is drawn from Laplacian distribution.

6.2.3 Preserving Semantic Meaning

Perturbing the latent representation of the given text by adding noise to it (Eq. 6.8) prevents the adversary from re-constructing the text from its latent representation and guarantees differential privacy. However, this approach may destroy the semantic meaning of the text data. Semantic meaning is task-dependant, e.g., classification is one of the common tasks. In the case of sentiment analysis, sentiment is of semantic meaning in the given text and sentiment prediction is a classification task. In order to preserve the semantic meaning of the textual representation, we need to add an

optimal amount of noise to the text latent representation which does not destroy the semantic meaning of the text data while ensuring data privacy. We approach this challenge by *learning* the amount of the added noise with the privacy budget ϵ in terms of training a classifier:

$$\hat{y} = softmax(\tilde{\mathbf{z}}; \theta_{D_S}) \tag{6.9}$$

where θ_{D_S} are the weights associated with the softmax function and $\hat{y}$ represents the inferred label for the classification.

To preserve the semantic meaning of the text representation, we seek a noisy latent representation which retains high utility and accordingly contains enough information for a downstream task, e.g., classification. We define a *semantic discriminator* D_S that aims to assign a correct class label to the perturbed representation, whose loss function is minimized as follows,

$$\min_{\theta_{D_S}, \epsilon} \mathcal{L}(\hat{y}, y) = \min_{\theta_{D_S}, \epsilon} \sum_{i=1}^{C} -y(i) \log \hat{y}(i) \tag{6.10}$$

where C is the number of classes, and $\mathcal{L}$ denotes the cross entropy loss function. The one-hot encoding of the ground truth label for the classification task is also denoted by y and $y(i)$ represents the i-th element of y, i.e., the ground truth label for i-th class.

To learn the value of the privacy budget ϵ, we employ the commonly used reparameterization trick (Kingma and Welling, 2013). Instead of directly sampling noise $\mathbf{s}(i)$ from Laplacian distribution (i.e., Eq. 6.8), this trick first samples a value r from a uniform distribution, i.e. $r \sim [0, 1]$, and then rewrites the amount of added noise $\mathbf{s}(i)$ as follows:

$$\mathbf{s}(i) = -\frac{\Delta}{\epsilon} \times sgn(r) \ln(1 - 2|r|), \quad i = 1, 2, .., d \tag{6.11}$$

This is equivalent to sampling noise s from $Lap(\frac{\Delta}{\epsilon})$. The advantage of doing so is that the parameter ϵ is now explicitly involved in the representation of the added

noise, $\mathbf{s}$, which makes it possible to use back-propagation to find the optimal value of ϵ. Large privacy budget ϵ could result in large privacy bounds. Hence, we add a constraint, $\epsilon < c_1$ where c_1 is a predefined constraint.

Another challenge here is that, $\hat{y}$ is inferred from $\tilde{\mathbf{z}}$ after introducing noise to the original latent representation $\mathbf{z}$. The noise is also sampled from the Laplacian distribution which results in large variance in the training process. To solve this issue and make the model more robust, we sample K copies of noise for each given document. In other words, we can rewrite Eq. 6.10 as follows:

$$\min_{\theta_{D_S},\epsilon} \mathcal{L}_{D_S}(\hat{y}, y) = \min_{\theta_{D_S},\epsilon} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(\hat{y}^k, y) =$$
$$\min_{\theta_{D_S},\epsilon} \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{C} -y(i) \log \hat{y}^k(i) \quad s.t. \quad \epsilon \leq c_1 \tag{6.12}$$

where the goal is to minimize loss function $\mathcal{L}_{D_S}$ w.r.t. the parameters $\{\theta_{D_S}, \epsilon\}$, and $\hat{y}^k = softmax(\tilde{\mathbf{z}}^k; \theta_{D_S})$. Note that $\tilde{\mathbf{z}}^k = \mathbf{z} + \mathbf{s}^k$ in which $\mathbf{s}^k$ is the k-th sample of the noise calculated with Eq. 6.11.

6.2.4 Protecting Private Information

We discuss how adding noise to the latent representation of the text can prevent adversary from learning the input textual information and guarantee differential privacy. Another important aspect of learning privacy preserving text representation is to ensure that sensitive and private information of the users such as age, gender, and location is not captured in the latent representation.

An adversary cannot design a private attribute inference attack better than what it has already anticipated. In this spirit, we leverage the idea of adversarial learning. In particular, we seek to train a *private attribute discriminator* D_P that can accurately identify the private information from the given representation, while learning a representation that can fool the discriminator and minimize leakage of private at-

tribute w.r.t. the determined adversary, which results in a representation that does not contain sensitive information. Assume that there are T private attributes (e.g., age, gender, location). Let p_t represents the ground truth (i.e., correct label) for the t-th sensitive attribute and $\theta_{D_P^t}$ demonstrates the parameters of discriminator model D_P for the t-th sensitive attribute. The adversarial learning can be formally written as:

$$\min_{\{\theta_{D_P^t}\}_{t=1}^T} \max_{\epsilon} \mathcal{L}_{D_P} = \min_{\{\theta_{D_P^t}\}_{t=1}^T} \max_{\epsilon} \frac{1}{K.T} \sum_{t=1}^{T} \sum_{k=1}^{K} \mathcal{L}_{D_P^t}(\hat{p}_t^k, p_t), \quad s.t. \;\; \epsilon \leq c_1 \tag{6.13}$$

where $\mathcal{L}_{D_P^t}$ denotes the cross entropy loss function and $\hat{p}_t^k = softmax(\tilde{\mathbf{z}}^k, \theta_{D_P^t})$ is the predicted t-th sensitive attribute using the k-th sample. The outer minimization finds the strongest private attribute inference attack and the inner maximization seeks to fool the discriminator by obscuring private information.

6.2.5 DPText - Learning the Text Representation

In the previous sections, we discuss how we can (1) add noise to prevent the adversary from reconstructing the original text from the latent representation and minimize the chance of privacy breach by satisfying differential privacy (Eq. 6.8), (2) control the amount of the added noise to preserve the semantic meaning of the textual information for a given task (Eq. 6.12), and (3) control the amount of the added noise so that user's private information is masked (Eq. 6.13). Inspired by the idea of adversarial learning, we achieve all three by modeling the objective function as a minmax game among the two introduced discriminators as follows:

$$\begin{aligned} &\min_{\theta_{D_S}, \epsilon} \max_{\{\theta_{D_P^t}\}_{t=1}^T} \mathcal{L}_{D_S} - \alpha \mathcal{L}_{D_P} = \\ &\min_{\theta_{D_S}, \epsilon} \max_{\{\theta_{D_P^t}\}_{t=1}^T} \frac{1}{K} \sum_{k=1}^{K} \left[\mathcal{L}(\hat{y}^k, y) - \alpha \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{D_P^t}(\hat{p}_t^k, p_t) \right], \quad s.t. \;\; \epsilon \leq c_1 \end{aligned} \tag{6.14}$$

where α controls the contribution of the private attribute discriminator in the learning process. This objective function seeks to minimize privacy leakage w.r.t. the attack, minimize loss in the semantic meaning of the textual representation, and protect private information. With N documents, Eq. 6.14 is written as follows:

$$\min_{\theta_{D_S},\epsilon} \max_{\{\theta_{D_P^t}\}_{t=1}^T} \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{K}\sum_{k=1}^{K}\left[\mathcal{L}(\hat{y}_n^k, y_n) - \alpha\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{D_P^t}(\hat{p}_{n,t}^k, p_{n,t})\right]\right] + \lambda\Omega(\theta),$$
$$s.t. \qquad \epsilon \leq c_1 \tag{6.15}$$

where $\theta = \{\theta_{D_S}, \epsilon, \{\theta_{D_P^t}\}_{t=1}^T\}$ is the set of all parameters to be learned, $\Omega(\theta)$ is the regularizer on the parameters such as Frobenious norm and λ is a scalar to control the amount of contribution of the regularization $\Omega(\theta)$.

The aim of this objective function is to perturb the original text representation by adding a proper amount of noise to it in order to prevent an adversary from inferring existence of the target textual representation in the database, reconstructing the user's original text and learning user's sensitive information from the latent representation, while preserving the semantic meaning of the perturbed representation for a given specific task. We stress that the resultant text representation satisfies $\tilde{\epsilon}$-differential privacy, where $\tilde{\epsilon} \leq c_1$ is the optimal learned privacy budget. This is further discussed in Section. 6.3.

6.2.6 Optimization Algorithm

The optimization process is illustrated in Algorithm 5. First, we compute the latent representation of all documents $\mathcal{Z} = \{\mathbf{z}_i, ..., \mathbf{z}_N\}$ in Line 1. We then sample a mini-batch of b samples from the training data. Next, we train the semantic discriminator D_S in Line 5 and private attribute discriminator using Eq. 6.13 in Line 6. Recall that we have a constraint on the variable ϵ, i.e., $\epsilon < c_1$. To satisfy this constraint, we use the idea of the projected gradient descent (Boyd and Vandenberghe,

Algorithm 5 The Learning Process of DPTEXT Model

Input: Training data $\mathcal{X}$, θ_{D_S}, ϵ, $\{\theta_{D_P^t}\}_{t=1}^{T}$, batch size b, c_1 and α.

Output: The privacy preserving learned text representation $\tilde{\mathbf{z}}$

1: Pre-train the document auto-encoder E_A to obtain the content representations according to Eq. 6.2 as $\mathbf{z} = E_A(x, \theta_e)$

2: **repeat**

3: Sample a mini-batch of b samples $\{x^i\}_{i=1}^{b}$ from $\mathcal{X}$

4: Add noise $\mathbf{s}$ to initial document representation $\mathbf{z}_i$, and get the new document representation $\tilde{\mathbf{z}}_i$, $i = 1, 2, ..., b$

5: Train semantic discriminator D_S by gradient descent via Eq. 6.12

6: Train the private attribute discriminator D_P via Eq. 6.13.

7: **until** Convergence

2004) wherein the gradient descent is performed one step, i.e. $\epsilon - \gamma \times \epsilon$ where γ is the learning rate. Then, the parameter ϵ is projected back to the constraint. This means that if $\epsilon > c_1$, then we set $\epsilon = c_1$, otherwise, keep the value of ϵ. The final noisy representation $\tilde{\mathbf{z}}$ can be then calculated for each given document according to the value of optimal learned privacy budget $\tilde{\epsilon} \leq c_1$ using Eq. 6.8.

6.3 Theoretical Analysis

Here, we show that the learned text representation using DPTEXT is $\tilde{\epsilon}$-differential privacy where $\tilde{\epsilon} \leq c_1$ is the learned optimal privacy budget. In particular, we prove the privacy guarantee for the final noisy latent representation $\tilde{\mathbf{z}}$ for each given document. The theoretical findings confirm the fact that DPTEXT minimizes the chance of revealing existence of textual representations in the database.

Theorem 2. *Let $\tilde{\epsilon} \leq c_1$ be the optimal value learned for the privacy budget variable*

ϵ w.r.t the semantic meaning and private attribute discriminators. Let $\mathbf{z}_i$ represents the original latent representation for document $\mathbf{x}_i$, $i = 1, ..., N$ inferred using Eq. 6.2 and. Moreover, let Δ denotes the L_1-sensitivity of the textual latent representation extractor function discussed in Section. 6.2.1. If each element $\mathbf{s}_i(l)$, $l = 1, ..., d$ in noise vector $\mathbf{s}_i$ is selected randomly from $Lap(\frac{\Delta}{\tilde{\epsilon}})$ ($\Delta = 2d$), the final noisy latent representation $\tilde{\mathbf{z}}_i = \mathbf{z}_i + \mathbf{s}_i$ satisfies $\tilde{\epsilon}$-differential privacy.

Proof. First we bound the change of $\mathbf{z}$ when one data point in the database changes. This gives the L_1-sensitivity of the textual latent representation extractor function discussed in Section. 6.2.1.

Recall the way $\mathbf{z}$ is calculated using Eq. 6.2. Function *tanh* is used in GRU to build the RNN which is used in Section. 6.2.1 to find the latent representation of a given document. The output of *tanh* function is within range $[-1, 1]$. This indicates that value of each element $\mathbf{z}(l)$, $l = 1, ..., d$ in the latent representation vector $\mathbf{z}$ is within range $[-1, 1]$. If one data point changes (i.e., removed from the database), the maximum change in value of each element $\mathbf{z}(l)$ is 2. Since the dimension of $\mathbf{z}$ is d, the maximum change in the L_1 norm of $\mathbf{z}$ happens when all of its elements, $\mathbf{z}(l)$, have the maximum change. According to Definition. 2.2, the L_1-sensitivity of $\mathbf{z}$ is $\Delta = 2 \times d$.

Now, assume that $\tilde{\epsilon} \leq c_1$ is the optimal value for the learned privacy budget. Then each element in $\mathbf{s}$ (i.e., $\mathbf{s}(l)$, $l = 1, 2, ...d$) is distributed as $Lap(\frac{\Delta}{\tilde{\epsilon}})$ based on Eq. 6.8 which is equal to randomly picking each $\mathbf{s}(l)$ from the $Lap(\frac{\Delta}{\tilde{\epsilon}})$ distribution, whose probability density function is $Pr(\mathbf{s}(l)) = \frac{\tilde{\epsilon}}{2\Delta} e^{-\frac{\tilde{\epsilon}|\mathbf{s}(l)|}{\Delta}}$.

Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be any two datasets only differ in the value of one record. Without loss of generality we assume that the representation of the last document is changed from $\mathbf{z}_n$ to $\mathbf{z}'_n$. Since the L_1-sensitivity of $\mathbf{z}$ is $\Delta = 2d$, then $\|\mathbf{z}_n - \mathbf{z}'_n\|_1 \leq \Delta$. Then

we have:

$$\frac{Pr[\mathbf{z}_n + \mathbf{s}_n = r|\mathcal{D}_1]}{Pr[\mathbf{z}'_n + \mathbf{s}'_n = r|\mathcal{D}_2]} = \frac{\prod_{l\in\{1,2,\ldots,d\}} Pr(r - \mathbf{z}_n(l))}{\prod_{l\in\{1,2,\ldots,d\}} Pr(r - \mathbf{z}'_n(l))}$$
$$= \frac{\prod_{l\in\{1,2,\ldots,d\}} Pr(\mathbf{s}_n(l))}{\prod_{l\in\{1,2,\ldots,d\}} Pr(\mathbf{s}'_n(l))} = e^{-\frac{\tilde{\epsilon}\sum_l |\mathbf{s}_n(l)|}{\Delta}} / e^{-\frac{\tilde{\epsilon}\sum_l |\mathbf{s}'_n(l)|}{\Delta}}$$
$$= e^{\frac{\tilde{\epsilon}\sum_l (|\mathbf{s}'_n(l)| - |\mathbf{s}_n(l)|)}{\Delta}} \leq e^{\frac{\tilde{\epsilon}\sum_l |\mathbf{s}'_n(l) - \mathbf{s}_n(l)|}{\Delta}} = e^{\frac{\tilde{\epsilon}\|\mathbf{s}'_n - \mathbf{s}_n\|_1}{\Delta}} \tag{6.16}$$

where $\mathbf{s}_n$ and $\mathbf{s}'_n$ are the corresponding noise vectors with respect to the learned $\tilde{\epsilon}$ when the input are $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. The first inequality also follows from the triangle inequality, i.e. $|a| - |b| \leq |a - b|$. The last equality follows from the definition of L_1-norm.

Since we have $\mathbf{s}_n = r - \mathbf{z}_n$ and $\mathbf{s}'_n = r - \mathbf{z}'_n$, we can write:

$$\|\mathbf{s}'_n - \mathbf{s}_n\|_1 = \|(r - \mathbf{z}'_n) - (r - \mathbf{z}_n)\|_1 = \|\mathbf{z}'_n - \mathbf{z}_n\|_1 \leq \Delta \tag{6.17}$$

This follows from the definition of L_1-sensitivity. We rewrite Eq. 6.16:

$$\frac{Pr[\mathbf{z}_n + \mathbf{s}_n = r|\mathcal{D}_1]}{Pr[\mathbf{z}'_n + \mathbf{s}'_n = r|\mathcal{D}_2]} \leq e^{\frac{\tilde{\epsilon}\|\mathbf{s}'_n - \mathbf{s}_n\|_1}{\Delta}} \leq e^{\frac{\tilde{\epsilon}\Delta}{\Delta}} = e^{\tilde{\epsilon}} \tag{6.18}$$

So, the theorem follows and the final noisy latent representation is $\tilde{\epsilon}$-differentially private. □

6.4 Experiments

In this section, we conduct experiments on real-world data to demonstrate the effectiveness of DPTEXT in terms of preserving both privacy of users and utility of the resultant representation for a given task. Specifically, we aim to answer the following questions:

- **Q1** - *Utility*: Does the learned text representation preserve the semantic meaning of the original text for a given task?

- **Q2** - *Privacy*: Does the learned text representation obscure users' private information?

- **Q3** - *Utility-Privacy Relation*: Does the improvement in privacy of learned text representation result in sacrificing the utility?

To answer the first question (**Q1**), we report experimental results for DPTEXT w.r.t. two well known text-related tasks, i.e., sentiment analysis and part-of-speech (POS) tagging. Sentiment analysis and POS tagging have many applications in Web and user-behavioral modeling (Zafarani *et al.*, 2014; Hovy and Søgaard, 2015; Jørgensen *et al.*, 2016). A recent research has shown how linguistic features such as sentiment are highly correlated with users demographic information (Hovy *et al.*, 2015; Potthast *et al.*, 2017). Another group of research shows the effectiveness of POS tags in predicting users' age and gender information (Nguyen *et al.*, 2011; Mukherjee and Liu, 2010). This makes users vulnerable against inference of their private information. Therefore, to answer the second question (**Q2**), we consider different private information, i.e., age, location, and gender, and report results for private attribute prediction task. To answer the third question (**Q3**), we investigate the utility loss against privacy improvement of the learned text representation. Next, we discuss each task and the experimental settings.

6.4.1 Task 1: Sentiment Analysis

Sentiment analysis is one of the important language processing applications(Zafarani *et al.*, 2014). Next, we describe the used dataset and model.

Data

We use a dataset from TrustPilot [2] from Hovy et al. (Hovy *et al.*, 2015). On

[2]http://trustpilot.com

their website, users can write reviews and leave a one to five star rating. Users can also provide some demographic information. In the collected dataset, each review is associated with three attributes, gender (male/female), age, and location (Denmark, France, United Kingdom, and United States). We follow the same approach as in (Li *et al.*, 2018) and discard all non-English reviews based on LANGID.PY [3] (Lui and Baldwin, 2012), and only keep reviews classified as English with a confidence greater than 0.9. We follow the setting of (Hovy and Søgaard, 2015) and categorize age attribute into three groups, over-45, under-35, and between 35 and 45. We follow the setting of (Lui and Baldwin, 2012) and subsample 10k reviews for each location to balance the five locations. We consider each review's rating score as the target sentiment class.

Model and Parameter Settings

For the document auto-encoder A, we use single-layer RNN with GRU cell of input/hidden dimension with d=64. For semantic and private attribute discriminators, we use feed-forward networks with single hidden layer with the dimension of hidden state set as 200, and a sigmoid output layer, which is determined through grid search. The parameters α and λ are determined through cross-validation, and are set as $\alpha = 1$ and $\lambda = 0.01$. The upper-bound constraint c_1 for the value of parameter ϵ is also set as $c_1 = 0.1$ to ensure the ϵ-differential privacy, $\epsilon = 0.1$ for the learned representation.

6.4.2 Task 2: Part-of-speech (POS) Tagging

POS tagging is another language processing application which is framed as a sequence tagging problem (Hovy *et al.*, 2015).

Data

For this task we use a manually POS tagged version of TrustPilot dataset in

[3]https://github.com/saffsd/langid.py

English. This data is obtained from Hovey et al. (Hovy and Søgaard, 2015) and consists of 600 sentences, each tagged with POS information based on the Google Universal POS tagset (Petrov *et al.*, 2012) and also labeled with both gender and age of the users. The gender attribute is categorized into male and female, and age attribute is categorized into two groups over-45, under-35. We follow the setting of (Li *et al.*, 2018) and use Web English Tree-bank (WebEng) (Bies *et al.*, 2012) as a pre-training tagging model because of the small quantity of text available for this task. WebEng is similar to TrustPilot datasets w.r.t. the domain as both contains unedited user generated textual data.

Model and Parameter Settings

Similar to the sentiment analysis task, we use single-layer RNN with GRU cell of input/hidden dimension with d=64 for document auto-encoder A. For semantic discriminator (i.e., POS tag predictor), we use bi-directional LSTM:

$$\begin{aligned} &\mathbf{h}_i = LSTM(x_i, \mathbf{h}_{i-1}; \theta_h), \quad \mathbf{h}'_i = LSTM(x_i, \mathbf{h}'_{i+1}; \theta'_h) \\ &y_i = Categorical(\phi([\mathbf{h}_i; \mathbf{h}'_i]); \theta_0) \end{aligned} \tag{6.19}$$

where $[.;.]$ denotes vectors concatenation, x_i, $i = 1, 2, ..N$ the input sequence, $\mathbf{h}_i$, the i-th hidden state and h_0 and h'_{N+1} are the terminal hidden states set to zero, and ϕ a linear transformation. The dimension of the hidden layer is set as 200.

For the private attribute discriminator, we use feed-forward networks with single hidden layer with the dimension of hidden state set as 200, and a sigmoid output layer (determined via grid search). For hyperparameters, we set values of α and λ as $\alpha = 1$ and $\lambda = 0.01$ which are determined through cross-validation. The upper-bound constraint for the value of ϵ is also set as $c_1 = 0.1$.

6.4.3 Experimental Design

We perform 10-fold cross validation for both POS tagging and sentiment analysis tasks. We follow state-of-the-art research and report accuracy score to evaluate the utility of the generated data for the given POS tagging (Brants, 2000; Hovy and Søgaard, 2015) or sentiment analysis task (dos Santos and Gatti, 2014). In particular, for the sentiment prediction task, we report accuracy for correctly predicting rating of reviews. We also report tagging accuracy for sentences for the POS tagging task. To examine the text representation in terms of obscuring private attributes, we report test performance in terms of $F1$ score for predicting private attributes. Note that the private attributes for sentiment task include age, gender and location while private attributes for tagging task only include gender and age.

We compare DPTEXT in both tasks with the following baselines:

- ORIGINAL: This is a variant of DPTEXT and publishes the original representation $\mathbf{z}$ without adding noise or utilizing D_S and D_P discriminators.

- DIFPRIV: This baseline adds Laplacian noise to the original representation $\mathbf{z}$ according to Eq. 6.8 (i.e., $Lap(\frac{\Delta}{\epsilon})$, $\epsilon = 0.1$, $\Delta = 2d$) without utilizing D_S and D_P discriminators. Note that this method makes the final representation ϵ-differentially private. We compare our model against this method to investigate the effectiveness of semantic and private attribute discriminators.

- ADV-ALL (Li *et al.*, 2018): This method utilizes the idea of adversarial learning and has two components, generator, discriminator. It generates a text representation that has high quality for the given task but has poor quality for inference of private attributes.

In both tasks, semantic discriminator D_S is trained on the train data and applied

to test data for predicting sentiment and POS tags. Similarly, we can apply private attribute discriminator D_P where it plays the role of an adversary trying to infer the private attributes of the user based on the latent textual representation. Private attribute discriminator D_P is also trained on the train data and applied to test data for evaluation. Higher accuracy score for semantic discriminator D_S indicates that representation has high utility for the given task, while lower $F1$ score for private attribute discriminator D_P demonstrates that the textual representation has higher privacy for individuals due to obscuring their private information.

6.4.4 Performance Comparison

For evaluating the quality of the learned text representation, we answer questions **Q1**, **Q2** and **Q3** for two different natural language processing tasks, i.e., sentiment prediction and POS tagging. The experimental results for different methods are demonstrated in Table 6.1.

Utility (Q1). The results of sentiment prediction for DPTEXT is comparable to the ORIGINAL approach. This means that the representation by DPTEXT preserves the semantic meaning of the textual representation according to the given task (i.e., high utility). DIFPRIV performs significantly better than DPTEXT and the reason is that DPTEXT applies noise at least as strong as DIFPRIV (or even more). Therefore, adding more noise results in bigger utility loss. We also observe that DPTEXT has better performance in terms of predicting sentiment in comparison to ADV-ALL.

The accuracy of POS tagging task is higher when DPTEXT is utilized rather than when ORIGINAL is used. This is because POS tagging results are biased toward gender, age and location (Hovy and Søgaard, 2015; Jørgensen *et al.*, 2016). In other words, this information affects the performance of tagging task. Removing private information from the latent representation results in removing this type of bias for

Table 6.1: Accuracy Score for Two Different Natural Language Processing Tasks, i.e., Sentiment Prediction and POS Tagging. $F1$ Score is Used to Evaluate Private Attribute Prediction task. Higher Accuracy Values Show Higher Utility, While Lower $F1$ Score Values Indicate Higher Privacy.

(a) Sentiment Prediction Task

Model	Sentiment	Private Attribute (F1)		
	(Acc)	Age	Loc	Gen
ORIGINAL	0.7493	0.3449	0.1539	0.5301
DIFPRIV	0.7397	0.3177	0.1411	0.5118
ADV-ALL	0.7165	0.3076	0.1080	0.4716
DPTEXT	**0.7318**	**0.1994**	**0.0581**	**0.3911**

(b) POS Tagging Task

Model	POS Tagging	Private Attribute (F1)	
	(Acc)	Age	Gen
ORIGINAL	0.8913	0.4018	0.5627
DIFPRIV	0.8982	0.3911	0.5417
ADV-ALL	0.8901	0.3514	0.5008
DPTEXT	**0.9257**	**0.2218**	**0.3865**

tagging task. Therefore, the learned representation is more robust and results in a more accurate tagging. DPTEXT also has better performance than DIFPRIV due to removal of private information and thus bias. Besides, results demonstrate that DP-TEXT outperforms ADV-ALL. These results indicate the effectiveness of DPTEXT in preserving semantic meaning of the learned text representation.

Privacy (Q2). In the sentiment prediction task, DPTEXT has significantly lower $F1$ score in comparison to ORIGINAL and thus outperforms ORIGINAL in terms of obscuring private information. DPTEXT has significantly better performance in hiding private information than DIFPRIV. This indicates that solely adding noise and satisfying ϵ-differential privacy does not protect textual information against leakage of private attributes. This further demonstrates the importance of private attribute discriminator D_P in obscuring users' private information. We also observe that the learned textual representation via DPTEXT hides more private information than ADV-ALL (lower $F1$ score). These results indicate that DPTEXT can successfully obscure private information.

In the POS tagging task, $F1$ scores of DPTEXT are significantly lower than ORIGINAL approach. These results demonstrate the effectiveness of DPTEXT in obscuring users' private attribute. Similarly, comparing $F1$ scores of DPTEXT and DIFPRIV shows that DPTEXT contains less private attribute information. This confirms the incapability of DIFPRIV in obscuring users' private information, and clearly shows the effectiveness of private attribute discriminator D_P. Moreover, DPTEXT outperforms ADV-ALL method in terms of hiding user's age and gender information. It confirms that the learned textual latent representation by DPTEXT preserves privacy by eliminating their sensitive information w.r.t. POS tagging task, i.e., high privacy.

Utility-Privacy Relation (Q3). For the sentiment prediction task, DPTEXT has achieved the highest accuracy and thus reached the highest utility in comparison to other methods. It also has comparable utility results to ORIGINAL. However, ORIGINAL utility is preserved at the expense of significant privacy loss. Moreover, although DIFPRIV satisfies differential privacy and its performance is comparable with DPTEXT in sentiment prediction task, it performs poorly in obscuring private information. DIFPRIV may provide weaker privacy guaranty comparing with DPTEXT

Table 6.2: Impact of Different Private Attribute Discriminators on DPTEXT for Sentiment Prediction and POS Tagging Tasks. Higher Accuracy Values Show Higher Utility, While Lower $F1$ Score Values Indicate Higher Privacy.

(a) Sentiment Prediction Task

Model	Sentiment	Private Attribute (F1)		
	(Acc)	Age	Loc	Gen
DPTEXT	0.7318	**0.1994**	0.0581	0.3911
DPTEXTAGE	**0.7573**	0.2248	0.1012	**0.3982**
DPTEXTLOC	0.7360	0.2861	0.0731	0.4100
DPTEXTGEN	0.7347	0.2997	**0.0623**	0.4053

(b) POS Tagging Task

Model	POS Tagging	Private Attribute (F1)	
	(Acc)	Age	Gen
DPTEXT	**0.9257**	0.2218	**0.3865**
DPTEXTAGE	0.9218	**0.2111**	0.4179
DPTEXTGEN	0.9361	0.2412	0.3916

since learned ϵ in DPTEXT can be smaller than $\epsilon = 0.1$ in DIFPRIV. In contrast, DPTEXT has significantly better (best) results in terms of privacy compared to the other approaches and also achieves the least utility loss in comparison to ADV-ALL. For the POS tagging task, the resultant representation from DPTEXT achieves the highest utility and privacy amongst all approaches. This shows the effectiveness of DPTEXT in preserving semantic meaning and obscuring private information for more accurate tagging.

The results for two natural language processing tasks indicate that DPTEXT learns a textual representation that (1) does not contain private information, and (2) preserves the semantic meaning of the representation for the given task.

6.4.5 Impact of Different Components

In this subsection, we investigate the impact of different private attribute discriminators on obscuring users' private information. To achieve this goal, we define three variants of the proposed framework, i.e., DPTEXT{AGE/GEN/LOC}. In each of these variants, the model is trained with discriminator of just one of the private attributes. For example, DPTEXTAGE is trained solely with age discriminator and does not use any other private attribute discriminators during training phase. The performance comparison is shown in Table 6.2.

In sentiment prediction task, we observe that using solely one of the private attribute discriminators can result in a representation which performs better in terms of sentiment prediction, in comparison to DPTEXT in which we use all three private attributes discriminators (i.e., higher utility). However, these variants perform poorly in terms of obscuring private attributes in comparison to the original DPTEXT model. These results indicate that although using one discriminator in the training process can help in preserving more semantic, it can compromise the effectiveness of learned representation in obscuring attributes.

In the POS tagging task, results show that DPTEXT achieves the best performance in tagging task (i.e., higher utility) in comparison to other methods that solely use one of the private attribute discriminators. The reason is that presence of age and gender related information in the text can negatively affect the tagging performance due to existing bias (Hovy and Søgaard, 2015; Jørgensen *et al.*, 2016). DPTEXT is thus more effective in removing this bias and leads to more accurate tagging in com-

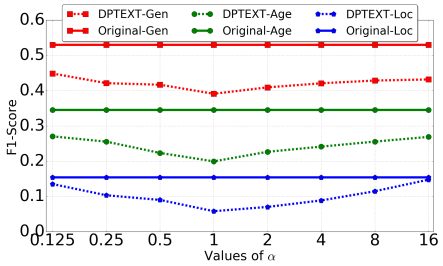
parison to DPTextAge and DPTextGen. Similar to sentiment prediction task, we observe that DPTextGen with only gender attribute discriminator is less effective than DPText in terms of hiding private attributes information. DPTextAge however, has the best results in terms of obscuring age attribute information.

6.4.6 Parameter Analysis

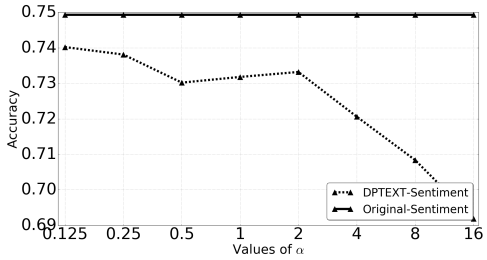
DPText has one important parameter α which controls the contribution from private attribute discriminator D_P. We investigate the effect of this parameter by varying it as $\{0.125, 0.25, 0.5, 1, 2, 4, 8, 16\}$. Original-{Age/Gen/Loc} shows the results for the corresponding task when the original text representation has been utilized. Results are shown in the Fig. 6.2.(a-b) and Fig. 6.2.(c-d) for sentiment prediction and POS tagging, respectively.

Although α controls the contribution of private attribute discriminator, we surprisingly observe that in both sentiment prediction and POS tagging task with the increase of α, the $F1$ scores for prediction of different private attributes decrease at first up to the point that $\alpha = 1$ and then it increases. This means that the private attributes were obscured more accurately at the beginning with the increase of α and less later. Moreover, with the increase of α, the accuracy of sentiment prediction task decreases. This shows that increasing the contribution of private attribute discriminator lead to decrease in the utility of resultant text representation. In case of POS tagging, the accuracy first increases and then decreases after $\alpha = 1$. This shows that removing the age and gender attributes related information results in removing the bias from learned text representation and improve the tagging task. However, after $\alpha = 1$ the utility of resultant representation decreases. Those patterns are useful for selecting the value of parameter α in practice.

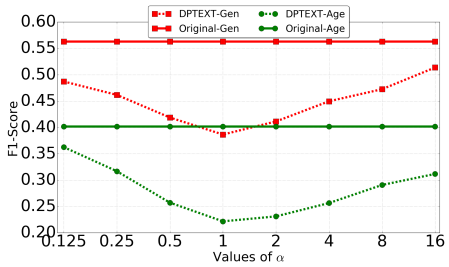
Moreover, in both tasks, setting $\alpha = 0.125$ results in an improvement in terms



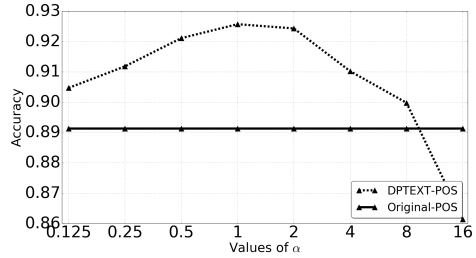
(a) Private Attribute Prediction w.r.t. Sentiment Task (F1)



(b) Sentiment Prediction (Acc)



(c) Private Attribute Prediction w.r.t. POS Tagging (F1)



(d) POS Tagging Prediction (Acc)

Figure 6.2: Performance Results for Private Attribute and Sentiment Prediction Tasks for Different Values of α

of the amount of hidden private information in comparison to the results of using ORIGINAL representation. This observation supports the importance of the private attribute discriminator. Another observation is that, after $\alpha = 1$, continuously increasing α degrades the performance of hiding private attributes (i.e., increasing $F1$ scores) in both sentiment prediction and POS tagging tasks. This is because the model could overfit by increasing α which lead to an inaccurate learned text representation in terms of preserving private attributes and semantic meaning of the text.

6.5 Generating Privacy Protected Text

Textual data is a very important source of users' privacy breach. In this work we propose a double privacy preserving text representation learning framework which extracts a privacy preserved *latent vector representation* from the given original textual document. We may sometimes need interpretable results and textual information for different applications. One important future direction is to generate privacy preserved text such as sentences and paragraphs rather than latent representation which is not interpretable. This is a very challenging task as removing personally identifiable information is not sufficient for protecting privacy of users. The reason is that private and sensitive information are not explicitly available in the textual information and are usually implicitly inferred from the given text. This makes existing solutions such as name entity recognition to be impractical. However, it is still important to publish interpretable privacy protected textual information.

One solution is to generate privacy preserved textual data rather than sharing intact original textual data or privacy preserved text representation. In order to do that, one can replace the semantic meaning discriminator component in DPTEXT with a text generator component which gets the original document and seeks to generate a new text which has similar semantic to the original document. Similar to DPTEXT, private-attribute discriminator can ensure that the newly generated textual information does not contain any private-attribute information. The final generated text will not contain the user's private attributes while it has same semantic as the original document. In future, we will investigate to extend DPTEXT to generate privacy preserving text which is critical for having interpretable results.

6.6 Conclusion

In this chapter, we propose a double privacy preserving text representation learning framework, DPTEXT, which learns a text representation that (1) is differentially private and protects users against identity disclosure attack, (2) guards users against private-attribute inference attack, and (3) retains utility of the textual information for a given task. DPTEXT is adversarial learning-based and has four main components, 1) an auto-encoder, 2) differential-privacy-based noise adder, 3) a semantic meaning discriminator, and 4) a private-attribute discriminator. Our theoretical and empirical results shows the effectiveness of DPTEXT in minimizing chances of learned textual representation re-identification, obscuring private-attribute information and preserving semantic meaning of the text.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Summary

The pervasive use of the Web has connected billions of people all around the globe and enabled them to obtain information at their fingertips. This results in tremendous amounts of user-generated data. This user-generated data is rich in content and contains sensitive information about users which risks exposing individuals' privacy and makes users traceable. Such rich data makes users vulnerable against two general types of attacks, identity disclosure and private-attribute information disclosure. People's privacy leakage leads to potential risks ranging from persecution by government to targeted frauds. Therefore, it is necessary to protect users' privacy without leaving their unnecessary traces of online activities. Preserving privacy of user-generated data is more challenging than structured one as it is heterogeneous, highly unstructured, and inherently different from relational and tabular data. Moreover, these information is crucial for online vendors to provide personalized services for users. However, protecting privacy comes at the cost of sacrificing utility of the user-generated data. Lack of users' information quality would result in low quality personalized services such as low quality search results and recommendations. This leads to a dilemma of privacy and utility.

In this dissertation, we investigate if users' privacy could be protected with respect to different types of attacks considering the aforementioned dilemma between privacy and utility. We study protecting user privacy problem from different aspects for different types of user-generated data. We propose four innovative research tasks - (1)

protecting user privacy in heterogeneous social media data, (2) protecting user privacy in Web browsing history data, (3) protecting user privacy in user-item interactions data, and (4) protecting user privacy in textual data

For protecting user privacy in heterogeneous social media data, we follow an adversarial approach and introduce a new identity disclosure attack specialized for heterogeneous social media data, namely ATHD. Using this attack, we evaluate the strengths of anonymization techniques in the context of heterogeneous social media data with multiple aspects and verify if it is sufficient. Our results illustrate that anonymizing even all aspects of data is not sufficient for protecting user privacy against identity disclosure attacks due to hidden relations between different aspects of the heterogeneous data.

For protecting user privacy in Web browsing history data, we propose an efficient framework PBOOSTER which protects users against identity disclosure attacks. The proposed solution takes advantage of user behavioral patterns from social media to infer what and how much additional data (in this case URLs) is required to improve user privacy while keeping the utility of the resultant data for future tasks. In particular, we first introduce two metrics to quantify privacy and utility and the trade-off between user privacy and utility. Then, we leverage these metrics in the proposed PBOOSTER framework to address the problem of anonymizing web browsing histories while retaining the utility. This framework first calculates how many links should be added to each user's browsing history. Then, it finds proper corresponding links according to a non-friend user's browsing behavior on social media platform. Our experiments demonstrate the efficiency of the proposed model by increasing user privacy and preserving utility of browsing history for future applications.

For protecting user privacy in user-item interactions data, we propose an adversarial learning-based recommendation with attribute protection model, RAP.

RAP guards users against private-attribute inference attack while maintaining utility. RAP recommends interesting yet safe products to users such that a malicious attacker cannot infer their private attribute from users' interactions history and recommendations. RAP has two main components, Bayesian personalized recommender, and private-attribute inference attacker. Our empirical results show the effectiveness of RAP in both protecting users against malicious private-attribute inference attacks and preserving quality of recommendation results.

For protecting user privacy in textual information, we propose a double privacy preserving text representation learning framework, DPTEXT, which protect users' privacy against both identity disclosure and private-attribute inference attacks. DPTEXT learns a text representation that (1) is differentially private and protects users against identity disclosure attack, (2) guards users against private-attribute inference attack, and (3) retains utility of the textual information for a given task. DPTEXT is adversarial learning-based and has four main components, 1) an auto-encoder, 2) differential-privacy-based noise adder, 3) a semantic meaning discriminator, and 4) a private-attribute discriminator. Our theoretical and empirical results show the effectiveness of DPTEXT in minimizing chances of learned textual representation re-identification, obscuring private-attribute information and preserving semantic meaning of the text.

Table 7.1 represents a summary of the existing state-of-the-art work and novel research problems we study in this dissertation with respect to the types of the privacy leakage attacks and different types of user-generated data.

7.2 Future Work

In this dissertation, we study the research problem of protecting user privacy with social media data and mining for different types of user-generated data. We show its

Table 7.1: An Overview of Privacy Attacks w.r.t. the Type of User-Generated Data.

Data Type / Attacks	Heterogeneous	Web Browsing History	User-Item Interactions	Textual
Identity Disclosure	✓	✓	Previous work	✓
Private-Attribute Inference	Previous work	Future opportunity	✓	✓

potential and significance, but only touch upon the tip of the iceberg of this fertile research area. Table 7.1 represents a summary of existing state-of-the-art work and novel research problems we study in this dissertation. There are many extensions and work that are worth further explorations. Below we present some promising research directions:

- **Anonymization of Heterogeneous Social Media Data**: User-generated social media data is heterogeneous and consists of different aspects. Existing research illustrates the vulnerability of each aspect against identity and private-attribute disclosure attacks. Existing anonymization techniques also assume that it is enough to anonymize each aspect of heterogeneous social media data independently. In our previous study, we evaluate this assumption showing that it is not correct in practice due to the hidden relations between different aspects of the heterogeneous data. In future, we will examine how different combinations of heterogeneous data (e.g., a combination of location and textual information) are vulnerable to de-anonymization attacks. Another research direction is to improve anonymization techniques to preserve privacy of users by considering hidden relations between different components of the heterogenous user-generated data.

- **Web Browsing History Data and Private-Attribute Inference Attacks**: Web browsing history data contains users' traces and private information. In our previous study, we propose a web browsing history anonymization framework which protects users against identity disclosure attacks while preserving the utility of web browsing history data for future personalized services. To the best of our knowledge, no research has been done on identifying potential privacy risks of web browsing history data against private-attribute disclosure attack. In future, we will examine vulnerabilities of such data. Moreover, we plan to investigate possible solutions for protecting users' privacy against private-attribute inference attack and web browsing history data anonymization.

- **Generating Privacy Protected Text**: Textual data is rich in content and is a very important source of information for adversaries and could be exploited in privacy breach attacks. It is thus important to properly anonymize such data. Few works focus on generating privacy preserving textual embeddings which protect users against different types of privacy attacks while retaining textual data utility for future tasks (Mosallanezhad *et al.*, 2019; Beigi *et al.*, 2020, 2019c). We may need interpretable results and textual information for some tasks. We will investigate to generate privacy preserving text (e.g., sentences, paragraphs) rather than latent representation which is critical for having interpretable results.

- **Privacy of Spatiotemporal User-generated Data**: Most of the online platforms support space-time indexed data which allows users to create a large volume of time-stamped, geo-located data. Such spatiotemporal data has an immense value for understanding users behavior better. Research has shown vulnerability of such data for breaching privacy of users due to intertwined re-

lation between time and geo-located data Jurgens *et al.* (2015); Beigi and Liu (2018a). This information may be used to infer users' location as well as their preferences and interests in case of recommendation systems. One future research direction could be investigating the role of spatiotemporal information in privacy of online users. Hence, we will investigate how to build anonymization frameworks for protecting users spatiotemporal information.

- **Adaptive Privacy Protection Techniques**: Attackers always seek to accurately infer users' identities and private-attribute information. Therefore, they can have the opportunity to iteratively adapt their attack model with respect to the existing defenses and privacy protection techniques. Privacy preserving techniques thus need to be updated accordingly considering the strength and knowledge of the malicious adversary. In our previous study, we show how adversarial learning could be leveraged to update the privacy protection frameworks by minimizing the attacker's gain. In particular, we study this problem from two different aspects, i.e., user-item interaction data and textual data. In this dissertation, we assume that the environment setting is static. In future, we will study adaptive privacy protection techniques in dynamic settings in which both attacker and defender can update their models over time with respect to each other. Techniques such as reinforcement learning (Sutton and Barto, 2018) could be also used for developing dynamic defense in which the agent observes the environment and updates its strategy gradually w.r.t. the conditions overtime.

- **Privacy-Preserving Training of Machine Learning Models:** In this dissertation, we study identifying and mitigating privacy risks originating from different types of online user-generated data. User-generated data is rich in con-

tent and therefore used for training various machine learning models in many applications. Recent research demonstrates leakage of machine learning algorithms about individuals whose data is used as a part of the training set for these models (Nasr *et al.*, 2018; Shokri *et al.*, 2017). This shows not only the user-generated data can leak sensitive and private information about individuals on its own but also machine learning models are vulnerable against privacy leakage attacks when trained on such data. The crucial question to ask is: How privacy could be protected for those users whose information have been used for the training process while maintaining the accuracy of the machine learning algorithm? In future, we will answer this question by investigating solutions to protect users against private-attribute and identity disclosure privacy attacks in various machine learning algorithms.

BIBLIOGRAPHY

Abawajy, J. H., M. I. H. Ninggal and T. Herawan, "Privacy preserving social network data publication", IEEE communications surveys & tutorials **18**, 3, 1974–1997 (2016).

Abbasi, A. and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace", ACM Transactions on Information Systems (TOIS) **26**, 2, 7 (2008).

Afroz, S., M. Brennan and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online", in "Security and Privacy (SP), 2012 IEEE Symposium on", pp. 461–475 (IEEE, 2012).

Aggarwal, G., T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas and A. Zhu, "Approximation algorithms for k-anonymity", Journal of Privacy Technology (JOPT) (2005).

Agrawal, R. and R. Srikant, "Privacy-preserving data mining", in "ACM Sigmod Record", vol. 29 (2000).

Al-Qurishi, M., M. Al-Rakhami, A. Alamri, M. Alrubaian, S. M. M. Rahman and M. S. Hossain, "Sybil defense techniques in online social networks: a survey", IEEE Access **5**, 1200–1219 (2017).

Almishari, M. and G. Tsudik, "Exploring linkability of user reviews", in "European Symposium on Research in Computer Security", pp. 307–324 (Springer, 2012).

Alvari, H., "Twitter hashtag recommendation using matrix factorization", arXiv preprint arXiv:1705.10453 (2017).

Alvari, H., G. Beigi, S. Sarkar, S. W. Ruston, S. R. Corman, H. Davulcu and P. Shakarian, "A feature-driven approach for identifying pathogenic social media accounts", arXiv preprint arXiv:2001.04624 (2020).

Alvari, H., A. Hajibagheri and G. Sukthankar, "Community detection in dynamic social networks: A game-theoretic approach", in "2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)", pp. 101–107 (IEEE, 2014a).

Alvari, H., A. Hajibagheri, G. Sukthankar and K. Lakkaraju, "Identifying community structures in dynamic networks", Social Network Analysis and Mining **6**, 1, 77 (2016a).

Alvari, H., S. Hashemi and A. Hamzeh, "Discovering overlapping communities in social networks: A novel game-theoretic approach", AI Communications **26**, 2, 161–177 (2013).

Alvari, H., K. Lakkaraju, G. Sukthankar and J. Whetzel, "Predicting guild membership in massively multiplayer online games", in "International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction", pp. 215–222 (Springer, 2014b).

Alvari, H., S. Sarkar and P. Shakarian, "Detection of violent extremists in social media", in "2nd International Conference on Data Intelligence and Security (ICDIS).", pp. 43–47 (IEEE, 2019a).

Alvari, H., E. Shaabani, S. Sarkar, G. Beigi and P. Shakarian, "Less is more: Semi-supervised causal inference for detecting pathogenic users in social media", in "Companion Proceedings of The 2019 World Wide Web Conference", pp. 154–161 (ACM, 2019b).

Alvari, H., E. Shaabani and P. Shakarian, "Early identification of pathogenic social media accounts", in "IEEE Intelligence and Security Informatics (ISI)", (IEEE, 2018).

Alvari, H. and P. Shakarian, "Causal inference for early detection of pathogenic social media accounts", arXiv preprint arXiv:1806.09787 (2018).

Alvari, H. and P. Shakarian, "Hawkes process for understanding the influence of pathogenic social media accounts", in "2nd International Conference on Data Intelligence and Security (ICDIS).", pp. 36–42 (IEEE, 2019).

Alvari, H., P. Shakarian and J. K. Snyder, "A non-parametric learning approach to identify online human trafficking", in "2016 IEEE Conference on Intelligence and Security Informatics (ISI)", pp. 133–138 (IEEE, 2016b).

Alvari, H., P. Shakarian and J. K. Snyder, "Semi-supervised learning for detecting human trafficking", Security Informatics **6**, 1, 1 (2017).

Anandan, B., C. Clifton, W. Jiang, M. Murugesan, P. Pastrana-Camacho and L. Si, "t-plausibility: Generalizing words to desensitize text", Transactions on Data Privacy **5**, 3, 505–534 (2012).

Andreou, A., O. Goga and P. Loiseau, "Identity vs. attribute disclosure risks for users with multiple social profiles", in "Proceedings of the 2017 IEEE/ACM ASONAM", pp. 163–170 (ACM, 2017).

Backes, M., P. Berrang, O. Goga, K. P. Gummadi and P. Manoharan, "On profile linkability despite anonymity in social media systems", in "Proceedings of ACM on Workshop on Privacy in the Electronic Society", (2016).

Backstrom, L., C. Dwork and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography", in "Proceedings of international conference on World Wide Web", (ACM, 2007).

Baeza-Yates, R. and A. Tiberi, "Extracting semantic relations from query logs", in "Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 76–85 (ACM, 2007).

Balsa, E., C. Troncoso and C. Diaz, "Ob-pws: Obfuscation-based private web search", in "Security and Privacy (SP), 2012 IEEE Symposium on", pp. 491–505 (IEEE, 2012).

Barbaro, M., T. Zeller and S. Hansell, "A face is exposed for aol searcher no. 4417749", New York Times **9**, 2008, 8 (2006).

Bassily, R. and A. Smith, "Local, private, efficient protocols for succinct histograms", in "Proceedings of the forty-seventh annual ACM symposium on Theory of computing", pp. 127–135 (ACM, 2015).

Beigi, G., "Social media and user privacy", arXiv preprint arXiv:1806.09786 (2018).

Beigi, G., R. Guo, A. Nou, Y. Zhang and H. Liu, "Protecting user privacy: An approach for untraceable web browsing history and unambiguous user profiles", in "Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining", pp. 213–221 (ACM, 2019a).

Beigi, G., X. Hu, R. Maciejewski and H. Liu, "An overview of sentiment analysis in social media and its applications in disaster relief", in "Sentiment analysis and ontology engineering", pp. 313–340 (Springer, 2016a).

Beigi, G., M. Jalili, H. Alvari and G. Sukthankar, "Leveraging community detection for accurate trust prediction", in "ASE International Conference on Social Computing, Palo Alto, CA, May 2014", (2014).

Beigi, G. and H. Liu, "Privacy in social media: Identification, mitigation and applications", arXiv preprint arXiv:1808.02191 (2018a).

Beigi, G. and H. Liu, "Similar but different: Exploiting users' congruity for recommendation systems", in "International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction", (Springer, 2018b).

Beigi, G. and H. Liu, "Identifying novel privacy issues of online users on social media platforms by ghazaleh beigi and huan liu with martin vesely as coordinator", ACM SIGWEB Newsletter , Winter, 4 (2019).

Beigi, G. and H. Liu, "A survey on privacy in social media: Identification, mitigation and applications", ACM Trans. Data. Science. (2020).

Beigi, G., 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 Thirteenth ACM International Conference on Web Search and Data Mining", (ACM, 2020).

Beigi, G., S. Ranganath and H. Liu, "Signed link prediction with sparse data: The role of personality information", in "Companion Proceedings of the The Web Conference 2019", (International World Wide Web Conferences Steering Committee, 2019b).

Beigi, G., K. Shu, R. Guo, S. Wang and H. Liu, "I am not what i write: Privacy preserving text representation learning", arXiv preprint arXiv:1907.03189 (2019c).

Beigi, G., K. Shu, R. Guo, S. Wang and H. Liu, "Privacy preserving text representation learning", in "Proceedings of the 30th ACM Conference on Hypertext and Social Media", pp. 275–276 (ACM, 2019d).

Beigi, G., K. Shu, Y. Zhang and H. Liu, "Securing social media user data: An adversarial approach", in "Proceedings of the 29th on Hypertext and Social Media", pp. 165–173 (ACM, 2018).

Beigi, G., J. Tang and H. Liu, "Signed link analysis in social media networks", in "10th International Conference on Web and Social Media, ICWSM 2016", (AAAI Press, 2016b).

Beigi, G., J. Tang and H. Liu, "Social science guided feature engineering: A novel approach to signed link analysis", ACM Trans. Intell. Syst. Technol. **11**, 1 (2019e).

Beigi, G., J. Tang, S. Wang and H. Liu, "Exploiting emotional information for trust/distrust prediction", in "Proceedings of the 2016 SIAM International Conference on Data Mining", pp. 81–89 (SIAM, 2016c).

Beretta, V., D. Maccagnola, T. Cribbin and E. Messina, "An interactive method for inferring demographic attributes in twitter", in "Proceedings of the 26th ACM Conference on Hypertext & Social Media", (ACM, 2015).

Bhagat, S., U. Weinsberg, S. Ioannidis and N. Taft, "Recommending with an agenda: Active learning of private attributes using matrix factorization", in "Proceedings of RecSys", (ACM, 2014).

Bies, A., J. Mott, C. Warner and S. Kulick, "English web treebank", Linguistic Data Consortium, Philadelphia, PA (2012).

Blei, D. M., A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation", Journal of machine Learning research **3**, Jan, 993–1022 (2003).

Bonneau, J., J. Anderson and G. Danezis, "Prying data out of a social network", in "Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in", pp. 249–254 (IEEE, 2009).

Bowers, J., H. Williams, G. Dozier and R. Williams, "Mitigation deanonymization attacks via language translation for anonymous social networks", Proceedings of ICML (2015).

Bowman, S. R., L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz and S. Bengio, "Generating sentences from a continuous space", arXiv preprint arXiv:1511.06349 (2015).

Boyd, S. and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).

Bradley, J. K., P. G. Kelley and A. Roth, "Author identification from citations", (2008).

Brants, T., "Tnt: a statistical part-of-speech tagger", in "Proceedings of the sixth conference on Applied natural language processing", pp. 224–231 (ACL, 2000).

Buades, A., B. Coll and J.-M. Morel, "A non-local algorithm for image denoising", in "Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on", vol. 2, pp. 60–65 (IEEE, 2005).

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

Chaabane, A., 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)", (2012).

Chaski, C. E., "Who is at the keyboard? authorship attribution in digital evidence investigations", International journal of digital evidence **4**, 1, 1–13 (2005).

Chaudhuri, K., C. Monteleoni and A. D. Sarwate, "Differentially private empirical risk minimization", in "JMLR", vol. 12 (2011).

Cheng, J., A. W.-c. Fu and J. Liu, "K-isomorphism: privacy preserving network publication against structural attacks", in "Proceedings of ACM SIGMOD International Conference on Management of data", (2010).

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation", arXiv preprint arXiv:1406.1078 (2014).

Christin, N., S. S. Yanagihara and K. Kamataki, "Dissecting one click frauds", in "Proceedings of ACM conference on Computer and communications security", (2010).

Cooper, A., "A survey of query log privacy-enhancing techniques from a policy perspective", ACM Transactions on the Web (TWEB) **2**, 4, 19 (2008).

Crandall, D., D. Cosley, D. Huttenlocher, J. Kleinberg and S. Suri, "Feedback effects between similarity and social influence in online communities", in "Proceedings of ACM SIGKDD", pp. 160–168 (2008).

Craswell, N. and M. Szummer, "Random walks on the click graph", in "Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval", pp. 239–246 (ACM, 2007).

Davies, M., "N-grams data from the corpus of contemporary american english (coca)", Downloaded from http://www.ngrams.info (2011).

Ding, D., M. Zhang, S.-Y. Li, J. Tang, X. Chen and Z.-H. Zhou, "Baydnn: Friend recommendation with bayesian personalized ranking deep neural network", in "Proceedings of the ACM CIKM", (2017).

Dingledine, R., N. Mathewson and P. Syverson, "Tor: The second-generation onion router", Tech. rep., Naval Research Lab Washington DC (2004).

dos Santos, C. and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts", in "Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers", pp. 69–78 (2014).

Duncan, G. T. and D. Lambert, "Disclosure-limited data dissemination", Journal of the American statistical association **81**, 393, 10–18 (1986).

Dwork, C., "Differential privacy: A survey of results", in "International Conference on Theory and Applications of Models of Computation", pp. 1–19 (Springer, 2008).

Dwork, C., F. McSherry, K. Nissim and A. Smith, "Calibrating noise to sensitivity in private data analysis", in "Theory of cryptography conference", pp. 265–284 (Springer, 2006).

Dwork, C., A. Roth *et al.*, "The algorithmic foundations of differential privacy", Foundations and Trends® in Theoretical Computer Science **9**, 3–4, 211–407 (2014).

Fawcett, T., "An introduction to roc analysis", Pattern recognition letters **27**, 8, 861–874 (2006).

Feige, U., V. S. Mirrokni and J. Vondrak, "Maximizing non-monotone submodular functions", SIAM Journal on Computing **40**, 4, 1133–1153 (2011).

Fu, H., A. Zhang and X. Xie, *Effective social graph deanonymization based on graph structure and descriptive information*, vol. 6 (ACM, 2015).

Fung, B. C., K. Wang, R. Chen and S. Y. Philip, "Privacy-preserving data publishing: A survey of recent developments", ACM Computing Surveys **42**, 4 (2010).

Gervais, A., R. Shokri, A. Singla, S. Capkun and V. Lenders, "Quantifying web-search privacy", in "Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security", pp. 966–977 (ACM, 2014).

Goga, O., H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer and R. Teixeira, "Exploiting innocuous activity for correlating users across sites", in "Proceedings of WWW", (2013).

Gong, N. Z. and B. Liu, "You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors.", in "USENIX Security Symposium", pp. 979–995 (2016).

Gong, N. Z. and B. Liu, "Attribute inference attacks in online social networks", ACM Transactions on Privacy and Security (TOPS) **21**, 1 (2018).

Gong, N. Z., A. Talwalkar, L. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. R. Shi and D. Song, "Joint link prediction and attribute inference using a social-attribute network", ACM Transactions on Intelligent Systems and Technology (TIST) **5**, 2 (2014).

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets", in "Advances in neural information processing systems", pp. 2672–2680 (2014).

Gotz, M., A. Machanavajjhala, G. Wang, X. Xiao and J. Gehrke, "Publishing search logs—a comparative study of privacy guarantees", IEEE Transactions on Knowledge and Data Engineering **24**, 3, 520–532 (2012).

Guerraoui, R., A.-M. Kermarrec, R. Patra and M. Taziki, "D 2 p: distance-based differential privacy in recommenders", Proceedings of the VLDB Endowment **8**, 8, 862–873 (2015).

Guerraoui, R., A.-M. Kermarrec and M. Taziki, "The utility and privacy effects of a click", in "Proceedings of ACM", (2017).

Gupta, P., S. Gottipati, J. Jiang and D. Gao, "Your love is public now: Questioning the use of personal information in authentication", in "Proceedings of ACM SIGSAC", (ACM, 2013).

Hajibagheri, A., G. Sukthankar, K. Lakkaraju, H. Alvari, R. T. Wigand and N. Agarwal, "Using massively multiplayer online game data to analyze the dynamics of social interactions", Social Interactions in Virtual Worlds: An Interdisciplinary Perspective (2018).

Hakkini-Tur, D., G. Tur *et al.*, "Sanitization and anonymization of document repositories", in "Web and information security", pp. 133–148 (IGI Global, 2006).

Harper, F. M. and J. A. Konstan, "The movielens datasets: History and context", Acm transactions on interactive intelligent systems (tiis) **5**, 4 (2016).

Hay, M., G. Miklau, D. Jensen, P. Weis and S. Srivastava, "Anonymizing social networks", Computer science department faculty publication series p. 180 (2007).

He, J., W. W. Chu and Z. V. Liu, "Inferring privacy information from social networks", in "International Conference on Intelligence and Security Informatics", pp. 154–165 (Springer, 2006).

Hill, S. and F. Provost, "The myth of the double-blind review?: author identification using only citations", Acm Sigkdd Explorations Newsletter **5**, 2, 179–184 (2003).

Hitaj, B., G. Ateniese and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning", in "Proceedings of ACM SIGSAC Conference on Computer and Communications Security", (2017).

Hovy, D., A. Johannsen and A. Søgaard, "User review sites as a resource for large-scale sociolinguistic studies", in "Proceedings of the 24th International Conference on World Wide Web", (2015).

Hovy, D. and A. Søgaard, "Tagging performance correlates with author age", in "Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics", vol. 2, pp. 483–488 (2015).

Howe, D. C. and H. Nissenbaum, "Trackmenot: Resisting surveillance in web search", Lessons from the Identity trail: Anonymity, privacy, and identity in a networked society **23**, 417–436 (2009).

Hua, J., C. Xia and S. Zhong, "Differentially private matrix factorization.", in "IJCAI", pp. 1763–1770 (2015).

Ji, S., W. Li and P. Mittal, "Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization.", in "24th USENIX Security Symposium (USENIX Security 15)", (2015).

Ji, S., W. Li, M. Srivatsa and R. Beyah, "Structural data de-anonymization: Quantification, practice, and implications", in "Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security", (2014).

Ji, S., W. Li, M. Srivatsa, J. S. He and R. Beyah, "General graph data de-anonymization: From mobility traces to social networks", ACM Transactions on Information and System Security (TISSEC) (2016a).

Ji, S., P. Mittal and R. Beyah, "Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey", IEEE Communications Surveys & Tutorials (2016b).

Jia, J. and G. NZhenqiang, "Attriguard: A practical defense against attribute inference attacks via adversarial machine learning", in "27th {USENIX} Security Symposium ({USENIX} Security 18)", (USENIX Association, 2018).

Jia, J., B. Wang, L. Zhang and N. Z. Gong, "Attriinfer: Inferring user attributes in online social networks using markov random fields", in "Proceedings of the WWW", pp. 1561–1569 (2017).

Jones, R., R. Kumar, B. Pang and A. Tomkins, "I know what you did last summer: query logs and user privacy", in "Proceedings of the sixteenth ACM conference on Conference on information and knowledge management", (ACM, 2007).

Jørgensen, A., D. Hovy and A. Søgaard, "Learning a pos tagger for aave-like language", in "Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies", pp. 1115–1120 (2016).

Jorgensen, Z. and T. Yu, "A privacy-preserving framework for personalized, social recommendations.", vol. 582 (2014).

Jurgens, D., T. Finethy, J. McCorriston, Y. T. Xu and D. Ruths, "Geolocation prediction in twitter using social networks: A critical analysis and review of current practice", in "Ninth International AAAI Conference on Web and Social Media", (2015).

Kifer, D. and A. Machanavajjhala, "No free lunch in data privacy", in "Proceedings of ACM SIGMOD International Conference on Management of data", pp. 193–204 (ACM, 2011).

Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).

Kingma, D. P. and M. Welling, "Auto-encoding variational bayes", arXiv preprint arXiv:1312.6114 (2013).

Klein, D. and C. D. Manning, "Accurate unlexicalized parsing", in "Proceedings of the 41st annual meeting of the association for computational linguistics", (2003).

Konstan, J. A. and J. Riedl, "Recommender systems: from algorithms to user experience", User modeling and user-adapted interaction **22**, 1-2 (2012).

Koppel, M., J. Schler and S. Argamon, "Computational methods in authorship attribution", Journal of the Association for Information Science and Technology **60**, 1, 9–26 (2009).

Koppel, M., J. Schler and S. Argamon, "Authorship attribution in the wild", Language Resources and Evaluation **45**, 1, 83–94 (2011).

Koppel, M., J. Schler, S. Argamon and E. Messeri, "Authorship attribution with thousands of candidate authors", in "Proceedings of ACM SIGIR", pp. 659–660 (ACM, 2006).

Koren, Y., "Collaborative filtering with temporal dynamics", in "Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 447–456 (ACM, 2009).

Korolova, A., K. Kenthapadi, N. Mishra and A. Ntoulas, "Releasing search queries and clicks privately", in "Proceedings of the 18th international conference on World wide web", pp. 171–180 (ACM, 2009).

Kosinski, M., D. Stillwell and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior", Proceedings of the National Academy of Sciences **110**, 15, 5802–5805 (2013).

Krause, A., A. Singh and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies", Journal of Machine Learning Research **9**, Feb, 235–284 (2008).

Lambert, D., "Measures of disclosure risk and harm", Journal of Official Statistics **9**, 2, 313 (1993).

Li, L., D. Wang, T. Li, D. Knox and B. Padmanabhan, "Scene: a scalable two-stage personalized news recommendation system", in "Proceedings ACM SIGIR", (2011).

Li, N., T. Li and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity", in "Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on", pp. 106–115 (IEEE, 2007).

Li, R., S. Wang, H. Deng, R. Wang and K. C.-C. Chang, "Towards social user profiling: unified and discriminative influence model for inferring home locations", in "Proceedings of ACM SIGKDD", (2012).

Li, Y., T. Baldwin and T. Cohn, "Towards robust and privacy-preserving text representations", (2018).

Lindamood, J., R. Heatherly, M. Kantarcioglu and B. Thuraisingham, "Inferring private information using social network data", in "Proceedings of WWW", pp. 1145–1146 (ACM, 2009).

Liu, C., S. Chakraborty and P. Mittal, "Dependence makes you vulnberable: Differential privacy under dependent tuples.", in "NDSS", vol. 16, pp. 21–24 (2016).

Liu, K. and E. Terzi, "Towards identity anonymization on graphs", in "Proceedings of international conference on Management of data", (ACM, 2008).

Lui, M. and T. Baldwin, "langid. py: An off-the-shelf language identification tool", in "Proceedings of the ACL 2012 system demonstrations", pp. 25–30 (Association for Computational Linguistics, 2012).

Luo, D., H. Xu, H. Zha, J. Du, R. Xie, X. Yang and W. Zhang, "You are what you watch and when you watch: Inferring household structures from iptv viewing data", IEEE Transactions on Broadcasting **60**, 1, 61–72 (2014).

Luo, Z. and Z. Chen, "A privacy preserving group recommender based on cooperative perturbation", in "International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery", (IEEE, 2014).

Machanavajjhala, A., J. Gehrke, D. Kifer and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity", in "Proceedings of ICDE", pp. 24–24 (IEEE, 2006).

Machanavajjhala, A., A. Korolova and A. D. Sarma, "Personalized social recommendations: accurate or private", Proceedings of the VLDB Endowment (2011).

Mack, N., J. Bowers, H. Williams, G. Dozier and J. Shelton, "The best way to a strong defense is a strong offense: Mitigating deanonymization attacks via iterative language translation", International Journal of Machine Learning and Computing **5**, 5, 409 (2015).

Mahmud, J., J. Nichols and C. Drews, "Home location identification of twitter users", ACM Transactions on Intelligent Systems and Technology (TIST) **5**, 3, 47 (2014).

Mao, H., X. Shuai and A. Kapadia, "Loose tweets: an analysis of privacy leaks on twitter", in "Proceedings of the 10th annual ACM workshop on Privacy in the electronic society", pp. 1–12 (ACM, 2011).

McPherson, M., L. Smith-Lovin and J. M. Cook, "Birds of a feather: Homophily in social networks", Annual review of sociology **27**, 1, 415–444 (2001).

McSherry, F. and I. Mironov, "Differentially private recommender systems: building privacy into the net", in "Proceedings of SIGKDD", (ACM, 2009).

Mendenhall, T. C., "The characteristic curves of composition", Science **9**, 214, 237–249 (1887).

Meng, X., S. Wang, K. Shu, J. Li, B. Chen, H. Liu and Y. Zhang, "Personalized privacy-preserving social recommendation", (2018).

Minkus, T., Y. Ding, R. Dey and K. W. Ross, "The city privacy attack: Combining social media and public records for detailed profiles of adults and children", in "ACM Conference on Online Social Networks", (2015).

Mislove, A., B. Viswanath, K. P. Gummadi and P. Druschel, "You are who you know: inferring user profiles in online social networks", in "Proceedings of WSDM", pp. 251–260 (ACM, 2010).

Mosallanezhad, A., G. Beigi and H. Liu, "Deep reinforcement learning-based text anonymization against private-attribute inference", in "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)", pp. 2360–2369 (2019).

Mosteller, F. and D. Wallace, "Inference and disputed authorship: The federalist", (1964).

Mukherjee, A. and B. Liu, "Improving gender classification of blog authors", in "Proceedings of the 2010 conference on Empirical Methods in natural Language Processing", pp. 207–217 (Association for Computational Linguistics, 2010).

Nanavati, M., N. Taylor, W. Aiello and A. Warfield, "Herbert west-deanonymizer.", (HotSec, 2011).

Narayanan, A., H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin and D. Song, "On the feasibility of internet-scale author identification", in "Security and Privacy (SP)", (IEEE, 2012).

Narayanan, A., E. Shi and B. I. Rubinstein, "Link prediction by de-anonymization: How we won the kaggle social network challenge", in "Neural Networks (IJCNN), The 2011 International Joint Conference on", pp. 1825–1834 (IEEE, 2011).

Narayanan, A. and V. Shmatikov, "Robust de-anonymization of large sparse datasets", in "IEEE Symposium on Security and Privacy", (IEEE, 2008).

Narayanan, A. and V. Shmatikov, "De-anonymizing social networks", in "Security and Privacy, 2009 30th IEEE Symposium on", pp. 173–187 (IEEE, 2009).

Nasr, M., R. Shokri and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks", arXiv preprint arXiv:1812.00910 (2018).

Nemhauser, G. L., L. A. Wolsey and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i", Mathematical Programming **14**, 1, 265–294 (1978).

Nguyen, D., N. A. Smith and C. P. Rosé, "Author age prediction from text using linear regression", in "Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities", pp. 115–123 (Association for Computational Linguistics, 2011).

Nilizadeh, S., A. Kapadia and Y.-Y. Ahn, "Community-enhanced de-anonymization of online social networks", in "Proceedings of the 2014 acm sigsac conference on computer and communications security", pp. 537–548 (ACM, 2014).

Niu, W., J. Caverlee and H. Lu, "Neural personalized ranking for image recommendation", in "Proceedings of the 11th ACM WSDM", (2018).

NZhenqiang, G. and B. Liu, "You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors", in "25th {USENIX} Security Symposium ({USENIX} Security 16)", (USENIX Association, 2016).

Parra-Arnau, J., D. Rebollo-Monedero and J. Forné, "Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems", Entropy **16**, 3, 1586–1631 (2014).

Pedarsani, P., D. R. Figueiredo and M. Grossglauser, "A bayesian method for matching two similar graphs without seeds", in "Communication, Control, and Computing (Allerton)", (IEEE, 2013).

Peddinti, S. and N. Saxena, "On the privacy of web search based on query obfuscation: a case study of trackmenot", in "Privacy Enhancing Technologies", (Springer, 2010).

Petrov, S., D. Das and R. McDonald, "A universal part-of-speech tagset", in "Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)", (2012).

Phuong, T. M. *et al.*, "Gender prediction using browsing history", in "Knowledge and Systems Engineering", (2014).

Polat, H. and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques", in "International Conference on Data Mining", (IEEE, 2003).

Potthast, M., F. Rangel, M. Tschuggnall, E. Stamatatos, P. Rosso and B. Stein, "Overview of pan'17", in "International Conference of the Cross-Language Evaluation Forum for European Languages", (2017).

Proserpio, D., S. Goldberg and F. McSherry, "Calibrating data to sensitivity in private data analysis: a platform for differentially-private analysis of weighted datasets", Proceedings of the VLDB Endowment **7**, 8 (2014).

Qian, J., X.-Y. Li, C. Zhang and L. Chen, "De-anonymizing social networks and inferring private attributes using knowledge graphs", in "INFOCOM International Conference on Computer Communications", (2016).

Ramakrishnan, N., B. J. Keller, B. J. Mirza, A. Y. Grama and G. Karypis, "Privacy risks in recommender systems", IEEE Internet Computing , 6, 54–62 (2001).

Rao, J. R., P. Rohatgi *et al.*, "Can pseudonymity really guarantee privacy?", in "USENIX Security", (2000).

Rashid, A. M., I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan and J. Riedl, "Getting to know you: learning new user preferences in recommender systems", in "Proceedings of the 7th international conference on Intelligent user interfaces", pp. 127–134 (ACM, 2002).

Rebollo-Monedero, D., J. Parra-Arnau and J. Forné, "An information-theoretic privacy criterion for query forgery in information retrieval", in "International Conference on Security Technology", pp. 146–154 (Springer, 2011).

Rehurek, R. and P. Sojka, "Software framework for topic modelling with large corpora", in "In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks", (Citeseer, 2010).

Rendle, S., C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback", in "Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence", (AUAI Press, 2009).

Sala, A., X. Zhao, C. Wilson, H. Zheng and B. Y. Zhao, "Sharing graphs using differentially private graph models", in "Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference", (2011).

Sarwar, B. M., G. Karypis, J. Konstan and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering", in "International conference on computer and information technology", (2002).

Shang, L., Z. Lu and H. Li, "Neural responding machine for short-text conversation", arXiv preprint arXiv:1503.02364 (2015).

Sharad, K., "True friends let you down: Benchmarking social graph anonymization schemes", in "Proceedings of Workshop on Artificial Intelligence and Security", (ACM, 2016).

Shen, Y. and H. Jin, "Privacy-preserving personalized recommendation: An instance-based approach via differential privacy", in "Proceedings of ICDM", (IEEE, 2014).

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

Shu, K., S. Wang, J. Tang, R. Zafarani and H. Liu, "User identity linkage across online social networks: A review", ACM SIGKDD Explorations Newsletter **18**, 2, 5–17 (2017).

Stamatatos, E., "A survey of modern authorship attribution methods", Journal of the Association for Information Science and Technology **60**, 3, 538–556 (2009).

Su, J., A. Sharma and S. Goel, "The effect of recommendations on network structure", in "Proceedings of WWW", (2016).

Su, J., A. Shukla, S. Goel and A. Narayanan, "De-anonymizing web browsing data with social networks", in "Proceedings of the 26th International Conference on World Wide Web", pp. 1261–1269 (2017).

Sutton, R. S. and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).

Sweeney, L., "k-anonymity: A model for protecting privacy", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (2002).

Tassa, T. and D. J. Cohen, "Anonymization of centralized and distributed social networks by sequential clustering", IEEE Transactions on Knowledge and Data Engineering **25**, 2, 311–324 (2013).

Ungar, L. H. and D. P. Foster, "Clustering methods for collaborative filtering", in "AAAI workshop on recommendation systems", vol. 1, pp. 114–129 (1998).

Volkova, S., Y. Bachrach, M. Armstrong and V. Sharma, "Inferring latent user properties from texts published in social media.", in "Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence.", (2015).

Wang, L., D. Yang, X. Han, T. Wang, D. Zhang and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation", in "Proceedings of the 26th International Conference on World Wide Web", pp. 627–636 (International World Wide Web Conferences Steering Committee, 2017).

Wang, P., J. Guo, Y. Lan, J. Xu and X. Cheng, "Your cart tells you: Inferring demographic attributes from purchase data", in "Proceedings of WSDM", (ACM, 2016).

Wang, Y. and X. Wu, "Preserving differential privacy in degree-correlation based graph generation", Transactions on data privacy **6**, 2, 127 (2013).

Weinsberg, U., S. Bhagat, S. Ioannidis and N. Taft, "Blurme: Inferring and obfuscating user gender based on ratings", in "Proceedings of the sixth ACM conference on Recommender systems", pp. 195–202 (ACM, 2012).

Wu, X., X. Ying, K. Liu and L. Chen, "A survey of privacy-preservation of graphs and social networks", Managing and mining graph data pp. 421–453 (2010).

Xiao, Q., R. Chen and K.-L. Tan, "Differentially private network data release via structural inference", in "Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 911–920 (ACM, 2014).

Yang, H., I. Soboroff, L. Xiong, C. L. Clarke and S. L. Garfinkel, "Privacy-preserving ir 2016: Differential privacy, search, and social media", in "ACM SIGIR", (2016).

Yang, J. and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach", in "Proceedings of the sixth ACM international conference on Web search and data mining", pp. 587–596 (ACM, 2013).

Yartseva, L. and M. Grossglauser, "On the performance of percolation graph matching", in "Proceedings of the first ACM conference on Online social networks", pp. 119–130 (ACM, 2013).

Ye, S., F. Wu, R. Pandey and H. Chen, "Noise injection for search privacy protection", in "Computational Science and Engineering", vol. 3, pp. 1–8 (IEEE, 2009).

Yin, Z., M. Gupta, T. Weninger and J. Han, "Linkrec: a unified framework for link recommendation with user attributes and graph structure", in "Proceedings of WWW", pp. 1211–1212 (ACM, 2010a).

Yin, Z., M. Gupta, T. Weninger and J. Han, "A unified framework for link recommendation using random walks", in "Proceedings of ASONAM", pp. 152–159 (IEEE, 2010b).

Ying, X. and X. Wu, "Graph generation with prescribed feature constraints", in "Proceedings of SDM", (SIAM, 2009).

Yuan, M., L. Chen and P. S. Yu, "Personalized privacy protection in social networks", Proceedings of the VLDB Endowment **4**, 2, 141–150 (2010).

Zafarani, R., M. A. Abbasi and H. Liu, *Social media mining: an introduction* (Cambridge University Press, 2014).

Zhang, J., J. Sun, R. Zhang and Y. Zhang, "Privacy-preserving social media data outsourcing", in "Proceedings of IEEE International Conference on Computer Communications (INFOCOM)", (2018).

Zhang, S., H. Yang and L. Singh, "Anonymizing query logs by differential privacy", in "Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval", pp. 753–756 (ACM, 2016).

Zheleva, E. and L. Getoor, "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles", in "Proceedings of the 18th international conference on World wide web", pp. 531–540 (ACM, 2009).

Zhou, B. and J. Pei, "Preserving privacy in social networks against neighborhood attacks", in "Proceedings of International Conference on Data Engineering", (2008).

Zhu, T., G. Li, Y. Ren, W. Zhou and P. Xiong, "Differential privacy for neighborhood-based collaborative filtering", in "Proceedings of ASONAM", pp. 752–759 (ACM, 2013).

Zhu, X. and Y. Sun, "Differential privacy for collaborative filtering recommender algorithm", in "Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics", pp. 9–16 (ACM, 2016).

Zhu, Y., L. Xiong and C. Verdery, "Anonymizing user profiles for personalized web search", in "WWW", (ACM, 2010).

Ziegler, C.-N., S. M. McNee, J. A. Konstan and G. Lausen, "Improving recommendation lists through topic diversification", in "Proceedings of the 14th international conference on World Wide Web", pp. 22–32 (ACM, 2005).

Zou, L., L. Chen and M. T. Özsu, "K-automorphism: A general framework for privacy preserving network publication", Proceedings of the VLDB Endowment **2**, 1, 946–957 (2009).

BIOGRAPHICAL SKETCH

Ghazaleh Beigi is a Ph.D. candidate in Computer Science at Arizona State University. Ghazaleh obtained her master's degree in Artificial Intelligence in 2014 and bachelor's degrees in Software Engineering in 2013 from Sharif University of Technology, Iran. The focus of her research is user behavioral modeling, privacy protection, trust/distrust prediction, signed network analysis and recommendation systems. As a result of her research work, she has published over 25 innovative works in major high-impact venues (including WSDM, SDM, TheWeb (former WWW), EMNLP, HyperText, ICWSM, ASONAM, SBP, ACM TIST, ACM TDS, and ACM SigWeb Newsletter), with over 300 citation count. She is also an author of 2 book chapters and 3 patents. Her research has been featured multiple times in the news including KDNugget, The Morning Paper, and ASU Now. Ghazaleh regularly serves on program committees for major international conferences and reviews for multiple journals and conferences. She has also interned as a Software Engineer at Google Inc., Sunnyvale, in 2019. Ghazaleh will join Google Inc., as a full-time Machine Learning Engineer in 2020. More information can be found at http://www.public.asu.edu/∼gbeigi/.