8-7-2018

Privacy Preserving Data Publishing

Zaobo He

PRIVACY PRESERVING DATA PUBLISHING

by

ZAOBO HE

Under the Direction of Zhipeng Cai, PhD and Yingshu Li, PhD

ABSTRACT

Recent years have witnessed increasing interest among researchers in protecting individual privacy in the big data era, involving social media, genomics, and Internet of Things. Recent studies have revealed numerous privacy threats and privacy protection methodologies, that vary across a broad range of applications. To date, however, there exists no powerful methodologies in addressing challenges from: high-dimension data, high-correlation data and powerful attackers.

In this dissertation, two critical problems will be investigated: the prospects and some challenges for elucidating the attack capabilities of attackers in mining individuals private information; and methodologies that can be used to protect against such inference attacks, while guaranteeing significant data utility.

First, this dissertation has proposed a series of works regarding inference attacks laying emphasis on protecting against powerful adversaries with auxiliary information. In the context of genomic data, data dimensions and computation feasibility is highly challenging in conducting data analysis. This dissertation proved that the proposed attack can effectively infer the values of the unknown SNPs and traits in linear complexity, which dramatically

improve the computation cost compared with traditional methods with exponential computation cost.

Second, putting differential privacy guarantee into high-dimension and high-correlation data remains a challenging problem, due to high-sensitivity, output scalability and signal-to-noise ratio. Consider there are tens-of-millions of genomes in a human DNA, it is infeasible for traditional methods to introduce noise to sanitize genomic data. This dissertation has proposed a series of works and demonstrated that the proposed differentially private method satisfies differential privacy; moreover, data utility is improved compared with the states of the arts by largely lowering data sensitivity.

Third, putting privacy guarantee into social data publishing remains a challenging problem, due to tradeoff requirements between data privacy and utility. This dissertation has proposed a series of works and demonstrated that the proposed methods can effectively realize privacy-utility tradeoff in data publishing.

Finally, two future research topics are proposed. The first topic is about Privacy Preserving Data Collection and Processing for Internet of Things. The second topic is to study Privacy Preserving Big Data Aggregation. They are motivated by the newly proposed data mining, artificial intelligence and cybersecurity methods.

INDEX WORDS: Inference Attack, Data Sanitization, Differential Privacy, SNP-Trait Association, Belief Propagation, Probabilistic Graphical Model

PRIVACY PRESERVING DATA PUBLISHING

by

Zaobo He

A Dissertation Submitted in Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2018

PRIVACY PRESERVING DATA PUBLISHING

by

ZAOBO HE

Committee Chair: Zhipeng Cai

Committee: Yingshu Li

Guantao Chen

Wei Li

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2018

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background and Motivations

Consider a social network application that collects user data from social media platforms, for performing business analysis or providing services. Due to privacy concern, the users do not want to release their sensitive data to third-party social applications, so that sensitive data is generally sanitized prior to releasing. However, it is possible to mine sensitive information carried in collected data by data mining techniques, to contribute to more commercial benefit. We proposed that well-designed data *sanitization* can be developed for realizing privacy-utility tradeoff in social network data publishing, although powerful attackers are presented with a broad range of auxiliary information to launch inference attacks. Such sanitization helps users sanitize their data by deleting some attributes, inserting other attributes, and perturbing some attributes, thereby hiding private information within randomness. Meanwhile, such sanitization should enable applications effectively recover useful information from sanitized data for data utility concern. Searching a well-designed sanitization method is highly non-trivial as data utility is restricted in sanitization process. Our works show that this issue can be alleviated by identifying the implicit dependency relationship encoded in data, and incorporating social attribute sanitization and link sanitization simultaneously, etc.

We looked more generally at using sanitization for preserving privacy. Consider a set of multi-modal sensory data is collected by mobile devices, which offers great potentials to promote meaningful services. However, privacy concerns arises from multiple situations as well: it is possible to collect private information from released data without any permission directly; furthermore, third party applications can also infer sensitive information contained in released data using data mining techniques. Given a certain data sanitization method for

user's data, what can we claim about the privacy protection it achieves? and what can we claim about the utility guarantee under certain level of privacy protection? New methodologies have been developed that answer these two questions in terms of optimizing utility with customized privacy. We proposed that these two types of privacy threats should be identified separately, defined as inherent data privacy and latent data privacy, and construct data sanitization methodology to optimize the tradeoff between data utility and customized two types of privacy. Moreover, we proposed to design such sanitization methodologies to combat against powerful third-party application with broad knowledge and launching optimal inference attacks. The new methodology has been applied to preserve the privacy of the users of Internet of Things and shown to yield practically useful results. The generality of this methodology has allowed us to extend privacy guarantees to multi-modal data in cyber-physical systems.

Consider individuals are using their genomes to learn about their (genetic) predispositions to diseases. However, once the owner of a genome is identified, he not only damages his own genomic privacy, but also puts his relatives privacy at risk (for example, form insurance companies). How do launch inference attacks to predict target phenotypes and genotypes with known genomes of an owner? How do release individual genomes privately with guaranteed genetic service quality? New methodologies have been developed that answer these two questions.

For the first research issue, we formalize the problem and detail an efficient reconstruction attack based on graphical models and belief propagation. We proposed that an effective reconstruction methodology can be built by incorporating SNPs, traits and SNP/trait associations (released by GWAS Catalog) on a probability graphical model and running belief propagation for inference. Our work does consider the magnitude property of SNPs and empower the inference method on target traits and genotypes in linear complexity. To protect against such inference attacks, we formalize the genomic privacy and utility metrics of individuals and develop a data-sanitization method to realize privacy/utility tradeoff.

For the second research issue, the state-of-the-art approach for privacy preserving data

publishing is *differential privacy*, which offers powerful privacy guarantee without confining assumptions about the background knowledge about attackers. However, high-dimensional data releasing with differential privacy guarantee is highly non-trivial as it requires injecting huge amount of noise, which would significantly degrade data utility. For genomic data with tens of million of SNPs (Single Nucleotide Polymorphism), current approaches based on differential privacy are not effective to handle. To address this problem, we propose a methodology to approximate the high-dimensional distribution of the original genomic data with a set of well-chosen low-dimensional distributions; then, noise with differential privacy guarantee can be injected into them. Finally, synthetic genomes are sampled from the approximate distribution, which can be proved satisfying differential privacy.

The above problems are briefly introduced in the following three sections. For the detailed information, please refer to Chapter 3, 4 and 5.

Finally, future research topics are proposed to complete the dissertation. The first topic is about privacy preserving data collection and processing for IoTs. The second future work is to study privacy preserving big data aggregation.

1.2 Collective Data-Sanitization for Preventing Sensitive Information Inference Attacks in Social Networks

Releasing social network data could seriously breach user privacy. User profile and friendship relationships are inherently private and generally are protected. Unfortunately, it is possible to predict the sensitive information carried in the released data latently by utilizing data mining techniques. Therefore, sanitizing network data prior to release is necessary. This study explore how to lunch an inference attack exploiting social networks with a mixture of non-sensitive attributes and social relationships. This issue is mapped to a collective classification problem and a collective inference model is proposed. In this model, an attacker utilizes user profile and social relationships in a collective manner to predict the sensitive information of related victims in a released social network dataset. To protect against such attacks, this study proposes a novel data sanitization method that collectively manipulates

user profile and friendship relations. The key novel idea lies that in addition to sanitize friendship relations, the proposed method can take advantages of various data-manipulating methods. It is shown that on various characteristics social communities, the proposed method can easily reduce adversary's prediction accuracy on sensitive information, while resulting in less accuracy decrease on non-sensitive information.

1.3 Latent-Data Privacy Preserving With Customized Data Utility for Social Network Data

Social network data can help with obtaining valuable insight into social behaviors and revealing the underlying benefits. New big data technologies are emerging to make it easier to discover meaningful social information from market analysis to counterterrorism. Unfortunately, both diverse social datasets and big data technologies raise stringent privacy concerns. Adversaries can launch inference attacks to predict sensitive latent information, which is unwilling to be published by social users. Therefore, there is a tradeoff between data benefits and privacy concerns. This study investigates how to optimize the tradeoff between latent-data privacy and customized data utility. In this study, a data sanitization strategy is proposed that does not greatly reduce the benefits brought by social network data, while sensitive latent information can still be protected. Even considering powerful adversaries with optimal inference attacks, the proposed data sanitization strategy can still preserve both data benefits and social structure, while guaranteeing optimal latent-data privacy. This is the first work that preserves both data benefits and social structure simultaneously and combats against powerful adversaries.

1.4 Inference Attacks and Controls on Genotypes and Phenotypes for Individual Genomic Data

The rapid growth of DNA-sequencing technologies motivates more personalized and predictive genetic-oriented services, which further attract individuals to increasingly release their genome information to learn about personalized medicines, disease predispositions,

genetic compatibilities, etc. Individual genome information is notoriously privacy-sensitive and highly associated with relatives. In this study, an inference attack algorithm is proposed to predict target genotypes and phenotypes based on belief propagation in factor graphs. With this algorithm, an attacker can effectively predict the target genotypes and phenotypes of target individuals based on genome information shared by individuals or their relatives, and genotype and phenotype association from genome-wide association study (GWAS). To address the privacy threats resulted from such inference attacks, this work elaborates the metrics to evaluate data utility and privacy and then presents a data sanitization method. The inference attack algorithm and data sanitization method are evaluated based on real GWAS dataset: Age-related macular degeneration (AMD) case/control dataset. The evaluation results show that the proposed method can effectively defense against genome threats while guaranteeing data utility.

1.5 Organization

The rest of this dissertation is organized as follows: Chapter 2 summarized the related literatures. Chapter 3 presents a data-sanitization method to prevent against sensitive information inference attacks in social networks. Chapter 4 studies latent-data privacy preserving with customized data utility for social network data. Chapter 5 solves the problem of inference attacks and controls on genotypes and phenotypes for individual genomic data. Chapter 6 proposes the future works.

Chapter 2

RELATED WORK

2.1 Privacy Threats in Social Network Data Publishing and Protection

Anonymization and De-anonymization. Privacy is typically protected by anonymization methods, *i.e.*, removing information regarding name, religion, political view, *etc.* However, such network could be de-anonymized by utilizing background knowledge such as reference network. For example, de-anonymization approaches utilize 'network mapping' to map social nodes from reference networks to anonymized networks.

In [1], the authors propose a community-enhanced de-anonymization approach to re-identify users, which first partitions the network into communities and then carries out a two-stage mapping: first mapping communities then the entire network. In [2], the authors consider a de-anonymization algorithm to re-identify the users in an anonymized social network based on network topology, namely, mapping the anonymous target graph and the aggregated graph from multiple social networks. Comparatively, our works attempt to protect against inference attacks on sensitive information of users, rather than solely re-identifying users in an anonymized network. In [3], the authors propose a family of anonymization algorithms and consider the corresponding de-anonymization algorithms. However, their network model only consists of users and friendship links and the attackers are assumed to re-identify the users. Clearly, their studied problem is quite different from our works because they do not consider how to anonymize a network in order to protect against inference attacks on sensitive attributes. The work in [4] presents a systematic survey for the anonymization techniques for social network data. The anonymization techniques are mainly categorized into the clustering-based approaches and the graph modification approaches. Comparatively, our works take advantage of various techniques to balance privacy and data utility.

Inference Attacks and Protection Methods. There are many works investigating how to infer sensitive information of users. In [5], the authors demonstrate that users' sensitive information can be inferred via detecting communities based on the assumption that users in a community are more likely share common attributes. Similarly, the work in [6] indicates that users' sensitive information can be inferred based on friendship information and group memberships, and it also shows that disclosure of one user's hidden attribute would breach her friends privacy. In [7], the authors develop a Bayes network model to infer sensitive information based on friendship links. Meanwhile, [7] takes a protection method that randomly hides friendship links and friends' attributes. In [8], several link-prediction and attribute-prediction algorithms are proposed in social-attribute networks. In [9], the authors employ the big data technologies to predict demographic information of users such as age and location based on users' mobile communication patterns. The work in [10] designs a method to predict sensitive latent information from texts published in social media. The work in [11] develops a data-sanitization strategy to predict sensitive information which can harness link and attribute information simultaneously. The work in [11] also evaluates the effect of removing links, removing attributes and perturbing attributes on protecting sensitive latent information. Our previous work [12] also studies how to customize the tradeoff data utility and customized latent-data privacy in classification based applications.

Comparatively, our works study which friendship link(s) and user' attribute(s) should be manipulated to protect privacy. Close to our works, [9] studies how to infer users' demographics (gender and age) depending on users' daily communication patterns. It novelly harness both the interaction between sensitive attributes and non-sensitive attributes, and the interaction among sensitive attributes (such as gender and age). Clearly, their method is quite different from our works because they do not consider the information from friendship relations that can be utilized in order to infer sensitive information. Moreover, our works further study how to protect against such inference attack deriving from collective information. Moreover, in [13], the authors consider the inference attacks to infer which shortened URLs clicked by a user in Twitter, only based on two public available information, twitter

metadata and the click analytics information.

Note that sanitizing data prior to release is a popular method to realize privacy protection and utility guarantee [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [28] [39] [40] [41] [42]

Both the work [14] and [15] sanitize data by synthesizing sampled data so that synthesized data satisfy differential privacy. In addition to sensitive latent information, protecting social network property privacy, like link privacy [16], degree distribution [17], graph privacy [18] and applications such as influence maximization [19] and privacy preserving content sharing [43], also attracts much attention. [44] explored how to sanitize data to optimize the tradeoff between three parties: utility, inherent-data privacy and latent-data privacy. To protect against inference attacks on social data, [45] proposed a data-sanitization method that can sanitize social attributes and links collectively with different sanitization methods. [20] explored the inference attacks on personal traits and genotypes based on belief propagation. Furthermore, a genomic data sanitization method is proposed in [20], by removing most indicative genomes to traits.

Existing privacy preserving techniques, like differential privacy [46], k-anonymity [47], l-diversity [48], are generally proposed for preserving inherent-data privacy; however, they are not competent for protecting latent-data privacy being subject to inference attacks. Inherent-data privacy is related to sensitive attribute contained in the attribute set released by users in order to receive data-related services. For example, age and gender are unavoidable data for health related services yet unwilling to be released by most consumers.

2.2 Genomic Data Privacy Threats and Protection

Probability graph models are widely used in predicting haplotype, genotypes or phenotypes in the context of genomic data releasing. Especially, Bayesian networks attract much attention in mapping the association between phenotypes and genes [49] [50] [51], or genetic linkage analysis [52] [53]. Factor graphs are also employed for inference attacks on S-NPs through incorporating linkage disequilibrium to preserve kin-genomic privacy [54]. The

work in [53] elaborates the applications of gene expression studies and genetic architecture of disease linkage analysis based on probability graph models. The authors in [54] proposed a reconstruction model to infer the genotypes of target individuals with released SNPs from individuals or their relatives, by mapping the linkage disequilibrium of SNPs. The work in [55] aims to predict the haplotype of individuals with publicly available genotypes and phenotypes, as well as lifestyle knowledge of individuals, based on Markov chain Monte Carlo (MCMC) sampling. To lower the large computational burden, the work in [56] introduces a "pre-phasing" strategy to balance the linkage analysis and computational cost, by estimating the haplotype statistically first and then impute unknown genotypes into the estimated haplotype in prior stage. The work in [57] reviews differential statistical techniques for genotype imputation, and explores the aspects that result in diverse imputation performance.

A crucial challenge presented in inference or linkage analysis for genomic data is high computation complexity. Although previous works have made significant efforts to address this issue, none of them has incorporated known and unknown genotypes, phenotypes and background knowledge of attackers together to launch inference attacks. Moreover, none of them presents effective methods to defense against such inference attacks in the context of high-dimensional data and complex associations.

In the recent years, inference attacks based on data mining, machina learning and statistical prediction methods have been investigated in several areas, such as location tracking, social networks [45] [44], and mobile networks [58] [43]. Differential privacy [59] is widely adopted in providing formal privacy guarantee through enabling distinguishability for query results over released data. However, applying differential privacy for protecting genomic data privacy is non-trivial since massive noises are required due to the high dimension of genomic data. The work in [60] proposes a privacy preserving data mining technique which supports analysts to conduct data analysis with accurate results while guarantees analysts cannot learn which and how many SNPs to consider.

Moreover, some recent works have proved that anonymization is not sufficient to preserve privacy [61] [62] [63]. The work in [61] proves that the released genotypes can be

de-anonymized with the help of auxiliary information such as known phenotypes. It is shown in [62] that individual surnames can be identified from individual genome data. Moreover, the works in [62] claims combining a surname with auxiliary information such as gender, age or state, the identity of target individual can be triangulated. Moreover, some cryptography techniques are employed to achieve tradeoff between genomic data privacy and utility [63] [64]. The work in [54] presents two metrics to evaluate privacy, attacker uncertainty and incorrectness.

Compared with the previous works, our work proposes an efficient inference model with low computation complexity by incorporating target unknown variables, known variables and auxiliary information into a probability graph model. Furthermore, we achieve the tradeoff between genomic data privacy and utility by introducing the utility and privacy metrics first and proposing an effective SNP-sanitization method, which can maximize data utility while protecting genome privacy.

Chapter 3

INFERENCE ATTACKS AND COLLECTIVE DATA-SANITIZATION FOR SOCIAL DATA PUBLISHING

3.1 Introduction

Social networks provide a virtual stage for users to reveal themselves to their own societies or to the public. For example, Facebook users publish information regarding favorite books, popular songs, interesting movies, political views, *etc.* Users of ResearchGate [65], a professional network for scientists and researchers, publish information regarding research experiences, publications, academic activities and so on. Besides users, third party users such as researchers, merchants, advertisers, and even adversaries may benefit from the huge amount of published data that can be easily and deliberately obtained from social networks for scientific/commercial purpose or malicious intention. For instance, IMDb [66] may make use of the data released by Facebook to suggest proper movies and TV programs to target users. However, the rising privacy concerns restrain the data release scale. Facebook Beacon [67] is an unsuccessful example that reminds people to release anonymous and incomplete user data. Therefore, the contradiction between the benefit rendered by data and privacy concerns drive third party users to mine sensitive information hidden in the released data in addition to non-sensitive information.

Privacy concerns in social networks can be mainly categorized into two types: inherent-data privacy and latent-data privacy. Inherent-data privacy is related to sensitive data contained in the data profile submitted by users in order to receive data-related services. For example, age and gender are unavoidable data for health related services yet unwilling to be released by most users. De-anonymization towards anonymous data is an inherent-data privacy instance. For example, two New York Time journalists used to successfully identify personal information from the published search logs involving 650,000 users made

available by AOL. The logs include the information of name, age, sex, location, *etc.*, and such information is associated with a specific individual. Another well-known example is that individuals' medical visits were successfully identified based on the anonymized data made available by the Group Insurance Commission, and the former governor of Massachusetts was one of the victims. On the other hand, latent-data privacy is related to unreleased sensitive information, yet such sensitive information can be inferred from released data or users' social relationships. For instance, Jenny does not publish her political opinions online, yet such information could be inferred by mining her friends' data as Jenny's social relationships may be public. Another illustrative example comes from ABCNews.com [68] and Boston Globe [69]. They reported that it is possible to determine the sexual orientations of some users by analyzing a subgraph from Facebook.

In this paper, we focus on latent-data privacy. We assume third party users may collect anonymous user data from social networks. Some users disclose their sensitive information, while others do not [70]. However, third party users can carry out de-anonymization actions and further infer sensitive information of users. We first investigate how to infer sensitive information hidden in the released data. Then, we propose some effective data sanitization strategies to prevent information inference attacks. On the other hand, the sanitized data obtained by these strategies should not reduce the valuable benefit brought by the abundant data resources, so that non-sensitive information can still be inferred and utilized by third party users.

To explore how to launch an inference attack by third party users, we employ a typical inference attack, called collective inference, as a case study. We present a novel implementation method for collective inference. Collective inference mainly rely on iteratively propagating current predicting results throughout a network to improve prediction accuracy, thus we need to consider how to best predict sensitive information in each iteration. Previous works primarily utilize the Naive Bayes classifier to infer sensitive information in each iteration. However, social network data are generally incomplete, inaccurate and uncertain. Hence, the existing approaches may not obtain a precise learned model and may

degrade inference performance. Our work does consider the special features of social network data to investigate collective attacks in diverse large scale social networks.

The previous works for preventing inference attacks mainly have three deficiencies. First, users' released data and their friendship information are separately considered, degrading prediction accuracy possibly. Second, only a single type of manipulation method, such as filtering, perturbing, and adding, is considered at a time, incurring poor effectiveness performance. Third, data utility is not taken into full consideration, reducing the benefit brought by the abundant amount of data. Therefore, the previous works cannot reasonably balance privacy and data utility. In this work, we propose two strategies to prevent inference attacks. Our strategies can ensure that third party users cannot obtain necessary information to accurately predict sensitive information. On the other hand, our strategies can still promote data utility.

In this work, we focus on two concrete issues: (a) how exactly third party users launch an inference attack to predict sensitive information of users, and (b) are there effective strategies to protect against such an attack to achieve a desired privacy-utility tradeoff. Following is the summary of our contributions and improvements over the previous works:

1. Rather than considering users' attribute sets and friendship information separately, we present a novel implementation method for collective inference that can effectively predict users' sensitive information, with both attribute sets and friendship information comprehensively taken into account.

2. To hide sensitive information through manipulating attribute sets, rather than simply implementing perturbing methods through introducing various types of noises, we rationally identify the dependency relationship between sensitive information and non-sensitive information.

3. To hide sensitive information through manipulating friendship information, rather than simply adding or removing friendship links, we propose a novel concept that enables us to easily find the most representative links.

4. We further analyze the relationships between data utility and non-sensitive information. The identified relationships then support us to design a collective strategy to achieve a desired privacy-utility tradeoff. Rather than relying on a signal type of manipulating method, our collective strategy is able to take advantages of various manipulating methods.

The remainder of the paper is organized as follows. The investigated problem is formalized in Section 4.2. Section 4.3 introduces some preliminary knowledge. In Section 3.4, we investigate the working scenario of inference. Some data sanitization strategies are then proposed in Section 3.5 and Section 3.6. The evaluation results are presented in Section 4.6. Section 4.7 concludes the paper.

3.2 Problem Statement

3.2.1 Social Network Model

We now present our network model.

Definition 3.2.1. ***Social network.*** *A social network is a graph $G(V, E, \mathcal{X})$ consisting of user set V, friendship link set E and the set of user attribute sets denoted by $\mathcal{X}$. For any user $u_i, u_j \in V$ $(1 \leq i, j \leq |V|)$, their friendship link $e_{i,j} \in E$ also indicates $e_{j,i} \in E$.*

Definition 3.2.2. ***Attribute set.*** *For an arbitrary user u_i, its attribute set is denoted by $\vec{X}_i \in \mathcal{X}$ $(1 \leq i \leq |V|)$. Each attribute $x_j \in \vec{X}_i$ $(1 \leq j \leq |\vec{X}_i|)$ is for a certain attribute category $h_r \in H$ $(1 \leq r \leq |H|)$, where H is the set of all the categories for a social network. We denote an attribute x_j as $x_j = \{h_r : l_1; ...; l_t\}$, which means x_j is for category h_r with value list $l_1; ...; l_t$ where $t \geq 1$.*

It is worth mentioning that for a particular category, the user input can be a single value or multiple values. For example, for category "Favorite movies", the input can be "The Terminator", "Titanic" and "Pianist". For category "Birthday", the input should be a single value. Moreover, there may be categories with no input values for some users, such as

"Political view" and "Religion view". In specific applications, certain categories are regarded as sensitive categories. We use $H_s \subseteq H$ to denote the set of the sensitive categories for a particular user. Any $x_j \in \vec{X}_i$ is a sensitive attribute of user u_i if x_j is for $h_r \in H_s$. Following is an example.

$$
\begin{aligned}
H &= \{\text{Favorite movies, Favorite books, Religion view,} \\
&\quad \text{Political view}\} \\
V &= \{u_1 = \text{Jack}, u_2 = \text{Emily}\} \\
\vec{X_1} &= \{x_1 = \{\text{Favorite movies:Titanic}\}, x_2 = \{\text{Favorite books:} \\
&\quad \text{Automata; Machine learning}\}\} \\
\vec{X_2} &= \{x_1 = \{\text{Favorite movies:Pianist}\}, x_2 = \{\text{Political view:} \\
&\quad \text{Conservative}\}\} \\
e_{1,2} &\in E, e_{2,1} \in E
\end{aligned}
$$

In this example, there are four categories as shown in H. There are two users u_1 and u_2. u_1 publishes one favorite movie and two favorite books. Thus, for u_1, $H_s = \{\text{Religion view, Political view}\}$. u_2 publishes her political view, thus for u_2, $H_s = \{\text{Religion view}\}$. u_1 and u_2 are friends in the social network.

Each possible attribute value for an arbitrary attribute category $h_r \in H_s$ can be viewed as a class label when third party users predict sensitive attribute x_j for category h_r. For example, if h_r is category "Political view", we can consider two possible attribute values as our class labels: "Conservative" and "Liberal". Class label is formally defined as follows.

Definition 3.2.3. ***Class label.*** *We say that y_i $(i \geq 1)$ is one of the class labels for $h_r \in H_s$ if y_i is one of the attribute values for attribute category h_r.*

3.2.2 Utility and Privacy

We now formally define privacy and utility. The existing privacy definitions, such as differential privacy [46], k-anonymity [47], l-diversity [48], are only for inherent-data, and

are not suitable for inference attacks. Meanwhile, most of the existing works evaluate data utility by only considering how much noise is added to the initial data. In this paper, we present a finer-grained utility definition.

Intuitively, we expect released data do not help with significantly improving prediction accuracy compared with the prediction accuracy based on prior knowledge.

Definition 3.2.4. ***Prior knowledge.*** *Prior knowledge is the information related to a data set but not necessarily obtained from the data set.*

For instance, prior knowledge can be users' movie viewing records, phone numbers, zip codes or the publicly available Voter Registration List. Such knowledge can be obtained from many ways rather than the data set itself. Then, privacy is formally defined as follows.

Definition 3.2.5. ***Classifier accuracy.*** *Classifier accuracy, denoted as $\Lambda_c^{h_r}(G)$, is the accuracy of classifier c trained on the available information of social graph G, and it is used to classify G to predict attributes for category $h_r \in H$.*

Definition 3.2.6. ***Privacy.*** *Given a social network G, prior knowledge $\mathcal{K}$ held by third party users, a set of classifiers denoted by $\mathcal{C}$, and a set of sensitive categories H_s, G is $(\Delta, \mathcal{C})$-private if for each attribute category $h_r \in H_s$, G satisfies*

$$\max_{c \in \mathcal{C}} \Lambda_c^{h_r}(G, \mathcal{K}) - \max_{c' \in \mathcal{C}} \Lambda_{c'}^{h_r}(\mathcal{K}) \leq \Delta$$

Δ denotes the additional prediction accuracy gained by third party users by utilizing G. Clearly, $\Delta \geq 0$, which is specified by data publisher. If $\Delta = 0$, it indicates that third party users do not gain additional prediction accuracy in predicting sensitive attributes for category $h_r \in H_s$.

With respect to data utility, there are two factors to consider. First, the sanitized social graph should not deviate from the initial one by too much. Second, the sanitized social

graph should guarantee a beneficiary can effectively infer the non-sensitive information of users. Then, we formally define it as follows:

Definition 3.2.7. ***Utility.*** *Given social graph G, data dissimilarity measurer $\mathcal{M}$, prior knowledge known to third party users $\mathcal{K}$, classifier set $\mathcal{C}$, and non-sensitive category set $H - H_s$, the sanitized graph of G, denoted as G', satisfies (ϵ, δ)-utility if for each attribute category $h_r \in H - H_s$, the following conditions are satisfied:*

$$\text{(i)}.\ \mathcal{M}(G, G') \leq \epsilon;$$
$$\text{(ii)}.\ \max_{c \in \mathcal{C}} \Lambda_c^{h_r}(G', \mathcal{K}) - \max_{c' \in \mathcal{C}} \Lambda_{c'}^{h_r}(\mathcal{K}) \geq \delta.$$

δ denotes the additional prediction accuracy gained by third party users by utilizing G'. Clearly, $\delta \geq 0$. If $\delta = 0$, it indicates that the classifier does not gain additional classification accuracy by utilizing G' in predicting non-sensitive attributes for category $h_r \in H - H_s$. As well, both ϵ and δ are specified by data publisher.

Compared with the existing definitions, Definition 4.4.1 takes the inferred non-sensitive attributes into consideration (condition (ii)). That is, any sanitization strategy should guarantee a beneficiary of the sanitized data and could effectively infer the non-sensitive attributes.

3.2.3 Problem Definition

Based on the above privacy and utility definitions, given user-specified thresholds on privacy and utility, the sanitization social graph is expected to achieve the desired privacy-utility tradeoff:

Input:

(1) Social graph $G(V = V^k \cup V^U,\ E,\ \mathcal{X},\ Y = Y^K \cup Y^U,\ H_s)$ with user set V, friendship link set E, the set of user attribute sets $\mathcal{X}$, and the set of sensitive categories H_s. $y^i \in Y$ is a class label of u_i for an arbitrary category $h_r \in H_s$.

(2) Y^K is the set of known labels for users $u_i \in V^K$, where V^K is the set of users with known labels. Y^U is the set of unknown labels for users $u_i \in V^U$, where V^U is the set of users with unknown labels.

(3) User-specified privacy threshold Δ, and utility thresholds ϵ and δ.

Output:

Task 1: Prediction method that can predict Y^U for users $u_i \in V^U$, where $V^U = V - V^K$.

Task 2: Data publishing method with optimized tradeoff between privacy and utility.

The first task investigates how third party users launch an inference attack to predict sensitive attributes. A powerful inference method is expected. Since users have the option to publish no attributes for some categories, the attribute data are usually incomplete. Meanwhile, there are always dishonest users, so the attribute data may be inaccurate or uncertain. Therefore, we employ the Rough Set Theory (RST) as a building block to develop our inference method. RST is a mathematical tool that can be used to extract knowledge from incomplete, inaccurate and uncertain data sets. It allows us to easily analyze the large scale and diverse social network data. For the second task, RST helps us to easily distinguish the objective attributes to be manipulated to protect against inference attacks.

3.3 Preliminaries

In this section, several concepts of RST are introduced and some illustrative examples are given. We then describe how to use RST to extract decision rules from the attribute data. Last, we present how to determine the class label of an user based on friendship information.

3.3.1 Rough Set Theory

We only introduce several basic concepts of RST and more details can be found in [71]. Knowledge representation in RST is through an information system. Based on the information system, the decision rules can be extracted.

Table 3.1. An example information system for a Facebook data set.

V	h_1: Favorite musical	h_2: Favorite movies	h_3: Favorite books	d: Political view
u_1	Taylor Swift	God's Not Dead	Heaven Is For Real	Conservative
u_2	Carrie Underwood	Son of God	I Declare	Conservative
u_3	Carrie Underwood	God's Not Dead	Heaven Is For Real	Liberal
u_4	George Strait	The Fast and the Furious	Heaven Is For Real	Green
u_5	George Strait	Son of God	I Declare	Liberal
u_6	Taylor Swift	Transformers	The Hunger Games	Conservative
u_7	George Strait	Son of God	The Hunger Games	Liberal
u_8	Taylor Swift	Transformers	I Declare	Conservative

Definition 3.3.1. ***Information system.*** *An information system is a pair* $\Gamma = (V,\ H = C \cup D)$*, where* V *is a finite set of users, and* H *is a nonempty finite set of attribute categories.* H *includes two subsets: the set of condition attribute categories* C *and the set of decision attribute categories* D*. For each attribute* x_j *for category* $h_r \in H$*, function* $f_{x_j}(u) : V \xrightarrow{x_j} \Omega_{h_r}$ *assigns an attribute value to* x_j *for user* u*, where* Ω_{h_r} *is the attribute value set for* h_r*.*

Example 3.3.1. *A simple example of information system for a Facebook data set is presented in Table 3.1. As shown in Table 3.1,* $V = \{u_1, u_2, \ldots u_8\}$*,* $C = \{h_1, h_2, h_3\}$*, and* $D = \{d\}$*. Attribute "Favorite movies" of* u_1 *is assigned value "God's Not Dead".*

Definition 3.3.2. ***Indiscernibility relation.*** *Given* $H' \subseteq H$*, any two users* u_i *and* u_j *having* H'*-indiscernibility relation is denoted by* $IND_{H'}(u_i, u_j)$ *where*

$$\begin{aligned} &IND_{H'}(u_i, u_j) \\ =&\{(u_i, u_j) \in V^2 \mid \forall x_j \text{ for } H',\ f_{x_j}(u_i) = f_{x_j}(u_j)\} \end{aligned}$$

We denote the users whose attributes have the same values for H' as $[u]_{H'}$, called the equivalence class of H'-indiscernibility relation.

Example 3.3.2. *Suppose* $H' = \{h_2, h_3\}$ *which is extracted from Table 3.1. Hence, both* (u_1, u_3) *and* (u_2, u_5) *have* H'*-indiscernibility relation. Table 3.1 also indicates* $[u]_{H'} = \{\{u_1, u_3\},\ \{u_2, u_5\},\ \{u_4\},\ \{u_6\},\ \{u_7\},\ \{u_8\}\}$*.*

Definition 3.3.3. ***H'-lower and H'-upper approximation of V'.*** *Given $V' \subseteq V$ and $H' \subseteq H$, V' can be approximated using only the information contained in H' by constructing H'-lower approximation and H'-upper approximation of V':*

$$\underline{H'}V' = \{u \mid [u]_{H'} \subseteq V'\}$$
$$\overline{H'}V' = \{u \mid [u]_{H'} \cap V' \neq \Phi\}$$

Example 3.3.3. *For the information system shown in Table 3.1, let $H' = \{h_2, h_3\}$ and $V' = \{u_1, u_2, u_6, u_8\}$. Hence, $\overline{H'}V' = \{u_1, u_2, u_3, u_5, u_6, u_8\}$ and $\underline{H'}V' = \{u_6, u_8\}$.*

Definition 3.3.4. ***Attribute dependency.*** *Let $H' \subseteq H$ and $H'' \subseteq H$. We say that H'' depends on H' with degree k $(0 \leq k \leq 1)$, denoted by $H' \rightarrow^k H''$, if*

$$k = \gamma(H',\ H'') = \frac{|POS_{H'}(H'')|}{|V|} \tag{3.1}$$

where $POS_{H'}(H'') = \bigcup_{X \in [x]_{H''}} \underline{H'}(X)$, called H'-positive region of H''.

In particular, if $k = 1$, we say that A'' totally depends on A'.

Example 3.3.4. *For the information system shown in Table 3.1, let $H' = \{h_2, h_3\}$ and $H'' = d$. Since*

$$[x]_{H'} = \{\{u_1, u_3\}, \{u_2, u_5\}, \{u_4\}, \{u_6\}, \{u_7\}, \{u_8\}\}$$
$$[x]_{H''} = \{\{u_1, u_2, u_6, u_8\}, \{u_4\}, \{u_3, u_5, u_7\}\}$$
$$\underline{H'}(\{u_1, u_2, u_6, u_8\}) = \{u_6, u_8\}$$
$$\underline{H'}(\{u_4\}) = \{u_4\}$$
$$\underline{H'}(\{u_3, u_5, u_7\}) = \{u_7\}$$

we can compute

$$POS_{H'}(H'') = \{u_6, u_8, u_4, u_7\}.$$

Hence,

$$k = \gamma(H',\ H'') = \frac{|POS_{H'}(H'')|}{|V|} = 4/8 = 1/2.$$

For an information system, there usually exist some redundant condition attributes that do not provide any additional knowledge for prediiction. Hence, RST defines a *reduct* for an information system as a minimum attribute set that keeps the indiscernibility relation. Furthermore, as would be discussed in Section 3.6, reduct can help us to find the privacy-dependent attributes and utility-dependent attributes, which is the foundation to balance the privacy-utility tradeoff.

Definition 3.3.5. ***Reduct.*** *Given an information system* $\Gamma = (V, H = C \cup D)$*, any* $R \subseteq C$ *is a reduct of* C *if*

(i). $POS_R(D) = POS_C(D)$;

(ii). for any $h_r \in C, IND(R - h_r) \neq IND(C)$.

After removing the repetitive row, $(V, R \cup D)$ *is called a reduct system.*

Example 3.3.5. *For the information system shown in Table 3.1, let* $R_1 = \{h_1, h_2\}$*,* $R_2 = \{h_1, h_3\}$ *and* $R_3 = \{h_2, h_3\}$*. We have*

$$POS_C(D) = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$$
$$POS_{h_1}(D) = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$$
$$POS_{h_2}(D) = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$$
$$POS_{h_3}(D) = \{u_4, u_6, u_7, u_8\}$$

Hence, we can conclude R_1 *and* R_2 *are reducts of* C *since they also satisfy the second condition according to Definition 3.3.2. However,* R_3 *is not a reduct of* C*.*

The first condition of Definition 3.3.5 indicates that the reduct retains the indiscernibility relation of the original attribute set. That is, any indiscernible pair of objects based on R is also indiscernible in A and vice versa. The second condition indicates that R is the minimum subset of A that keeps its indiscernibility.

3.3.2 Generating Decision Rules Based on an Attribute Set

We now introduce how the decision rules are generated based on the reduct system $(V, R \cup D)$. Suppose the equivalence class of the R-indiscernibility relation and the D-indiscernibility relation are $[u]_R = \{P_1, P_2, \ldots, P_m\}$ and $[u]_D = \{Q_1, Q_2, \ldots, Q_n\}$, respectively. Each P_i $(1 \leq i \leq m)$ and Q_j $(1 \leq j \leq n)$ is a user or a set of users. For example, for the information system shown in Table 3.2, $[u]_R = \{P_1 = \{u_1, u_3, u_9\}, P_2 = \{u_2, u_4\}, P_3 = \{u_5, u_6\}, P_4 = \{u_7, u_8\}\}$ if $R = \{h_1, h_2\}$, and $[u]_D = \{Q_1 = \{u_1, u_2, u_3, u_4, u_7, u_9\}, Q_2 = \{u_5, u_6, u_8\}\}$ if $D = \{d\}$.

Since both $[u]_R$ and $[u]_D$ partition V, each P_i is associated with a set $M_i = \{Q_j \mid P_i \cap Q_j \neq \Phi\}$. For example, P_4 is associated with $M_4 = \{Q_1, Q_2\}$.

Hence, for an arbitrary user u, we have:

If $u \in P_i$, then $u \in Q_{j_1}$ or $u \in Q_{j_2}$ ... or $u \in Q_{j_{|M_i|}}$.

According to Definition 3.3.1, we know that each P_i of $[u]_R$ corresponds to an attribute vector $\vec{X}(P_i) = \{x_1^i, x_2^i \ldots x_{|R|}^i\}$, where an arbitrary user $u \in P_i$ if and only if $f_{x_1^i}(u) = v_{x_1^i}$ and ... and $f_{x_{|R|}^i}(u) = v_{x_{|R|}^i}$, where $v_{x_k^i}$ $(1 \leq k \leq |R|)$ is the attribute value of attribute x_k for the users in P_i. For example, P_1 corresponds to $\vec{X}(P_1) = \{$"Taylor Swift", "Gods Not Dead"$\}$.

Similarly, suppose there is a signal decision attribute d, *i.e.*, $|D = 1|$, and each Q_j of $[u]_D$ corresponds to a decision attribute value v_{d_j}, where an arbitrary user $u \in Y_j$ if and only if $f_d(u) = v_{d_j}$. For example, any $u \in Y_j$ if and only if $f_d(u) =$ "Conservative".

Hence, the above rule can be rewritten as

if $f_{x_1^i}(u) = v_{x_1^i}$ and ... and $f_{x_{|R|}^i}(u) = v_{x_{|R|}^i}$, then $f_d(u) = v_{d_1}$ or $f_d(u) = v_{d_2}$, or ..., or

Table 3.2. Information system for generating decision rules.

V	h_1: Favorite musical	h_2: Favorite movies	d: Political view
u_1	Taylor Swift	God's Not Dead	Conservative
u_2	Carrie Underwood	Son of God	Conservative
u_3	Taylor Swift	God's Not Dead	Conservative
u_4	Carrie Underwood	Son of God	Conservative
u_5	George Strait	Son of God	Liberal
u_6	George Strait	Son of God	Liberal
u_7	Taylor Swift	Transformers	Conservative
u_8	Taylor Swift	Transformers	Liberal
u_9	Taylor Swift	God's Not Dead	Conservative

$f_d(u) = v_{d_{|M_i|}}$.

If $P_i \subseteq Q_j$, which indicates the class label of any user $u \in P_i$ is uniquely determined by d_j, we say P_i is a deterministic class. Otherwise, we call P_i as an indeterministic class.

Example 3.3.6. *We extract decision rules from the reduct system $(V, R \cup D)$ shown in Table 3.2, where $R = \{h_1, h_2\}$ and $D = \{d\}$. Let $P_1 = \{u_1, u_3, u_9\}$, $P_2 = \{u_2, u_4\}$, $P_3 = \{u_5, u_6\}$, $P_4 = \{u_7, u_8\}$, $Q_1 = \{x_1, x_2, x_3, x_4, x_7, x_9\}$ and $Q_2 = \{x_5, x_6, x_8\}$. Based on the prior analysis, P_1, P_2 and P_3 are deterministic classes. Hence, the following decision rules are extracted:*

if A_1 = "Taylor Swift" and A_2 = "God's Not Dead",

then, D = "Conservative";

if A_1 = "Carrie Underwood" and A_2 = "Son of God",

then, D = "Conservative";

if A_1 = "George Strait" and A_2 = "Son of God",

then, D = "Liberal".

3.3.3 Prediction Based on Friendship Information

Another significant knowledge that can be utilized to infer sensitive attributes is friendship information in social networks. However, it is inaccurate to extract decision rules based on friendship information directly, since there are relatively few links from users with known labels that connect to an arbitrary user u_i. Therefore, rather than directly extracting decision rules from the friendship links of u_i, we consider u_j's class, where $u_j \in N_i$ and N_i is the neighbor set of u_i. For clarity, u_i in class y_t is denoted by y_t^i.

For simplicity, the probability of u_i to be in class y_t, denoted as $P(y_t^i)$, is the average probability of its neighbors being in y_t:

$$P(y_t^i|N_i) = \frac{1}{|N_i|} \sum_{u_j \in N_i} P(y_t^j)$$

However, purely calculating the average probability of neighbors would incur overfitting. To prevent this, the weighted-vote Relational Neighbor algorithm (wvRN) [72] suggests to add a weight to each friendship link. There are many such methods and we adopt the ones with the assumption that the more public attributes shared by two friends, the more is the sensitive attributes that are shared by two friends. Then we introduce weight $W_{i,j}$ between u_i and u_j as follows:

$$W_{i,j} = \frac{|(A_i^1, \ldots, A_i^m) \cap (A_j^1, \ldots, A_j^n)|}{|A_i|} \tag{3.2}$$

Equation (4.2) calculates the total number of attributes shared by u_i and u_j divided by the number of u_i's attributes. Obviously, $W_{i,j} \neq W_{i,j}$. Then to determine y^i based on N_i becomes the following, where Z is a normalization factor:

$$P(y_t^i|N_i) = \frac{1}{Z} \sum_{n_j \in N_i} P(y_t^j) \times W_{i,j} \tag{3.3}$$

3.4 Collective inference

Unfortunately, the prediction methods described in the previous section have several problems. The attribute-based classifier (Section 3.3.2) just considers the attribute sets of the users it is classifying. Conversely, relation-based classifier (Section 3.3.3) only considers the friendship information of a user. However, third party users may launch an inference attack by exploiting all the publicly available information. Moreover, a major problem of relation-based classifier is that it requires that at least one of the neighbors of each unlabeled user to be located in the training set (*i.e.*, the set of users with known labels, as shown in Equation (4.3)). Obviously, this strict requirement is hard to be satisfied by real-world data. Collective inference attempts to tackle the above two issues by considering both attribute-based classifier and relation-based classifier in a collaborative manner to improve prediction accuracy. Formally, we consider the following network prediction problem.

Definition 3.4.1. ***Collective inference.*** *Given social graph* $G(V = V^k \cup V^U,\ E,\ \mathcal{X},\ Y = Y^K \cup Y^U,\ H_s)$ *with user set* V*, friendship link set* E*, the set of user attribute sets* $\mathcal{X}$*, and the set of sensitive categories* H_s*.* $y^i \in Y$ *is a class label of* u_i *for an arbitrary category* $h_r \in H_s$*.* L^K *is the known labels for users* $u_i \in V^K$*. Collective inference is to predict* Y^U *for users* $u_i \in V^U$*, where* $V^U = V - V^K$*.*

This problem is challenging as some of the user labels are unknown. A fundamental idea is to first predict a class label approximately and then refine the predicted result iteratively. Several collective classification algorithms have been proposed to increase accuracy when the network users are interrelated, such as the Iterative Classification Algorithm (ICA) [73] and Gibbs sampling (Gibbs) [74]. Many collective classification algorithms and variants, including ICA, use an attribute-based classifier M_A to predict the approximate class label at the bootstrap stage; then, they use both attribute and link based classifier, M_{AR}, to refine the results. The algorithms repeat these two operations until the class labels converge. We present an algorithm under the framework of ICA that takes RST as a local classifier (one that uses local information, *e.g.*, attribute sets of users), denoted by ICA-RST.

ICA-RST is shown in Algorithm 1. It first learns an attribute-based classifier M_A based on the known labels Y^K (step 1), which is a set of RST decision rules. Then, by M_A, it predicts the labels of the users with unknown labels, V^U (steps 2-3). Step 5 stores the known labels Y^K and the predicted labels $\{y^i|u_i \in V^U\}$ in set Y^R. The known labels and the predicted labels are utilized to calculate link features for each user in V^U (step 7). Step 8 then learns a classifier M_{AR} based on all of the attributes and labels. Step 10 utilizes M_{AR} to predict unknown labels. Finally, Step 11 returns the predicted results.

Algorithm 1: ICA-RST

Input: V = users, E = links, $\mathcal{X}$ = attribute set, Y^K = labels of known users ($Y^K = \{y_i|u_i \in V^K\}$)
Output: Y^U = labels of unknown users ($Y^U = \{y_i|u_i \in V^U, V^U = V - V^K\}$)

1 $M_A = learn\_RST\_Rule(V^K, Y^K)$; // learn classifier M_A utilizing only attributes
2 **for** *each user* $u_i \in V^U$ **do**
3 $y_i \leftarrow M_A(\vec{X}_i)$; // predict the labels of the unknown users utilizing M_A
4 **for** t = *1 to* n **do**
5 $Y^R \leftarrow Y^K \cup \{y_i|u_i \in V^U\}$; // store the known labels and the predicted labels in set Y^R
6 **for** *each user* $u_i \in V^U$ **do**
7 $\vec{f_i} = calReFeats(V, E, Y^R)$; // calculate link features utilizing known labels and the predicted labels
8 $M_{AR} = learn\_RST\_Rule(V, Y^R)$; // learn classifier M_{AR} utilizing all of the attributes and labels
9 **for** *each user* $u_i \in V^U$ **do**
10 $y_i = M_{AR}(\vec{X}_i, \vec{f_i})$; // re-predict the unknown labels utilizing M_{AR}
11 return Y^U

Fig.3.1 shows an example for ICA-RST, which is applied to political view inference attacks. Each step in Fig.3.1 displays a social graph consisting of five users with the corresponding friendship links. The class label of each node is y_i, which takes value from label set $Y = \{Con, Lib\}$, representing *conservative party* and *liberal party*, respectively. Four users have unknown labels ($V^U = \{u_2, u_3, u_4, u_5\}$) and only one user has known labels

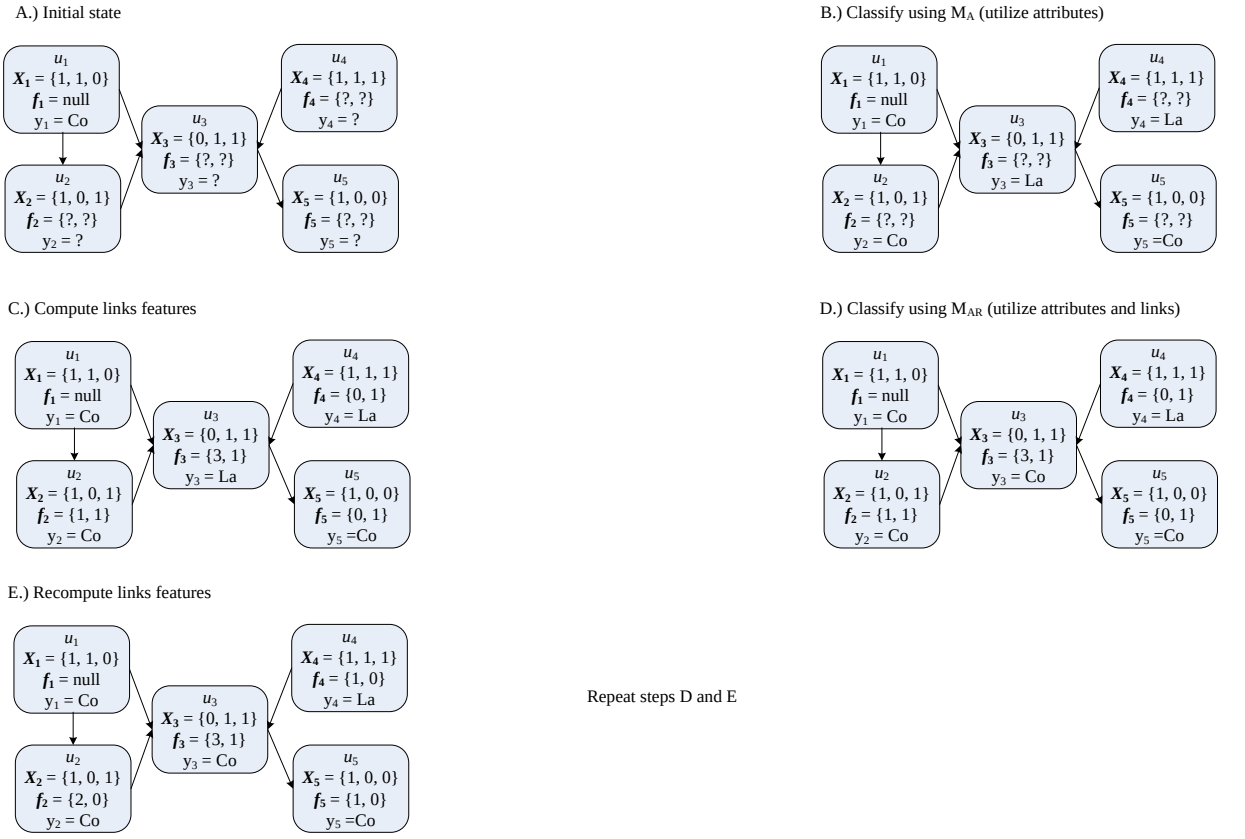


Figure 3.1. An example for ICA-RST.

($V^K = \{u_1\}$). In step A, no labels y_i and link features $\vec{f_i}$ in V^U have been predicted, so they are marked with a question mark. In step B, attribute-based classifier M_A assigns a label to u_i in V^U using only attribute $\vec{X}_i$. Based on the predicted labels in step B, step C then computes the link features of each u_i. For instance, $\vec{f_3} = \{3, 1\}$ in step C since u_3 has three links with label *Co* (*i.e.*, u_1, u_2, u_5) and one link with label *La* (*i.e.*, u_4). In step D, classifier M_{AR} reclassifies users in V^U using the attributes and link features, and it recomputes the link features. Repeat step D and step E until the labels of u_i in V^U converge to a stable state.

3.5 Hiding Sensitive Information

The existing privacy preservation techniques, such as differential privacy [46], k-anonymity [47], l-diversity [48] and so forth, are designed for inherent-data privacy only, and

do not protect against inference attacks directly. For instance, differential privacy ensures that the aggregation results of a data set that operates the differential privacy algorithms are the same with or without one row. k-anonymity guarantees that third party users cannot distinguish real data from at least their nearest $k-1$ neighbors. Since our goal is to release social network data while preserving data utility and protecting against inference attacks, the above techniques are not competent.

To develop a sanitization strategy, there are three issues to be addressed concerning inference attacks. First, we should understand the relationship between sensitive attributes and the released data set. For instance, *Bryden* made Facebook analysis and found that conservatives with distinguished cultural tastes than other partisans [75]. Second, it is necessary to figure out which attribute or link manipulating method(s) should be carried out to achieve the desired privacy-utility tradeoff. For example, we can add or modify an attribute or a link to add noises to the released data. Also, we can remove some attributes and links to anonymize the released data. However, which one of the above methods are better? Last, for a specific manipulating method, how to effectively carry it out to achieve the desired privacy-utility tradeoff? For example, which attributes and links should be removed to markedly decrease the prediction accuracy on sensitive attributes while resulting in less accuracy decrease on non-sensitive attributes. In the following, we address the three issues.

3.5.1 Choosing Attributes to Manipulate

One of the most significant aspects is the dependency relationships between non-sensitive attributes and sensitive attributes. Through analyzing dependence relationships, we can reveal which publicly available attributes dominate the prediction results on sensitive attributes. Namely, dependency relationship provides the theoretical basis to determine which attributes should be chosen to manipulate. For example, suppose *political view* depends on *activity* and *favorite movies*, which indicates that we can manipulate these two attributes to reduce the prediction accuracy on *political view*. We denote the attributes that dominate the classification results on sensitive attributes as privacy-dependent attributes.

As shown in Definition 3.3.4 in Section 3.3.1, given an arbitrary information system $\Gamma = (V, A = C \cup D)$, any decision attribute set $D' \subseteq D$ depending on condition attribute set $C' \in C$ with degree k can be calculated as $C' \rightarrow^k D'$. Here, C and D can be viewed as publicly available attributes and sensitive attributes, respectively.

To hide sensitive attributes, our idea is to manipulate the most dependent attributes with respect to each sensitive attribute: for an arbitrary user u_i with attribute set $\vec{X}_i$, and a sensitive attribute $x_j = \{h_r : l_t\}$, we can find the most dependent attribute $x_s \in C$ ($1 \leq s \leq |C|$) for sensitive attribute x_j based on the following:

$$\text{argmax}_s\{k \mid x_s \rightarrow^k l_t\}$$

In practice, we can find any n_t-most dependent attributes for sensitive attribute with attribute value l_t, after extracting the attributes with the largest n_t dependence degree.

However, simply manipulating privacy-dependent attributes could incur utility reduction if we do not take utility into consideration. Consider the scenario that IMDb makes use of the data released by Facebook to suggest proper movies and TV programs to target users. It may classify users considering different movie types to make recommendations, depending on users' attribute sets. However, movie types could also depend on a privacy-dependent attribute. For example, the possible movie types are closely related to the attribute of "favorite movies".

We denote the attributes that dominate the classification results on non-sensitive attributes as utility-dependent attributes. Hence, the following statement determines our choice:

Problem 3.5.1. *Given social graph $G(V,\ E,\ \mathcal{X} = C \cup D)$ with publicly available attribute set C and sensitive attribute set D, determine the set of attributes $C' \subseteq C$ so that $G'(V, E, C' \cup D)$ has the most decrease in prediction accuracy in D, while preserving the utility of C.*

Hence, the double dependency relationships become a challenge for the attribute manipulating method.

3.5.2 Attribute Manipulating Method

Obviously, attributes can be manipulated in three manners: *adding* new attributes, *removing* existing attributes, and *perturbing* (substitute one attribute to another). Since both *adding* and *perturbing* decrease prediction accuracy on sensitive information by introducing different types of noises, they are collectively called the obfuscation method. *Removing*, however, can be viewed as an anonymization method. Taking which manipulating method(s) depends on data semantics, privacy and utility metrics and so on. For example, if users specify a set of attributes as sensitive and quantify utility as the expected number of released attributes, the *removing* method could be advisable.

Suppose we just release our data to the public and do not announce what the data is used for. For example, social graph G is released online for research purpose and x_p is a privacy-dependent attribute of G. In this case, we have no direct measurement to determine how to perturb a_p; namely, use what attribute to substitute x_p, since no applications are specified. In this case, the removing method may be a proper choice. We just need to remove the privacy-dependent attributes.

For example, consider two social graphs G_1 and G_2, which are sanitized graphs of G after applying the obfuscation and anonymization methods, respectively. When we consider G_1 in which there is an attribute "favorite movies: Titanic", based on the employed obfuscation method, the original attribute set may not have this attribute or have an obsoletely distinct one. Hence, utility cannot be guaranteed by an obfuscation method when the application is not specified.

However, if the data are released for a special purpose such as movie recommendation, we could evaluate the changing utility when manipulating the attributes. In this case, the perturbing method could be a proper choice since properly perturbing can guarantee the desired privacy-utility tradeoff. For example, when we consider G_2, it may sacrifice much utility if there exists intersection between privacy-dependent attributes and utility-dependent attributes. Due to these observations, we consider *removing* and *perturbing* separately.

3.5.3 Link Manipulating Method

Another option for protecting against inference attacks is to manipulate links. Unlike attribute, link manipulating methods only add new links and remove existing links. With the same reason, we only consider the link anonymization method in the case of releasing the data set to the public and without announcing what the data are used for. With the same goal, the manipulated links should reduce the prediction accuracy on sensitive attributes. Suppose that adding or removing a link renders the prediction results on sensitive attributes locating in each class with a same probability, and we call this link as *indistinguishable link*, which is formally defined as follows:

Definition 3.5.1. ***Δ'-Indistinguishable link.*** *Given social graph $G(V, E, \mathcal{X})$ and an arbitrary $u_i \in V$ with possible class labels $Y = \{y_1, y_2, \ldots\}$, and $P\{y_t^i\}$ is the probability of u_i with label y_t. Any link $f_j \in F_{i,j}$ is an indistinguishable link of u_i if removing f_j results that*

$$Var\{P\{y_1^i\}, P\{y_2^i\}, \ldots, P\{y_{|Y|}^i\}\} \leq \Delta' \tag{3.4}$$

where $Var(S)$ is for valuating the variance of set S.

To hide sensitive attributes through removing links, our idea is to manipulate the most indistinguishable link with respect to each user. We can find the most indistinguishable link f_j for u_i based on the following:

$$\text{argmin}_j\{Var\{P\{y_1^i\}, P\{y_2^i\}, \ldots, P\{y_{|Y|}^i\}\} \mid \text{removing } f_j\}$$

3.6 Collective Method

To protect against inference attacks, we attempt to manipulate attributes by perturbing and removing separately in the respective situations. As mentioned in Section 3.5.2, these two methods must be restricted by the utility requirements. In this section, in order to achieve

the desired privacy-utility tradeoff, we present how to utilize removing and perturbing in a collective manner.

Clearly, simply removing or perturbing Privacy-Dependent Attributes (PDAs) could reduce prediction accuracy on non-sensitive attributes. Hence, there should exist a compromise strategy for manipulating the PDAs to achieve the privacy-utility tradeoff. Therefore, rather than removing or perturbing PDAs directly, we analyze the relationship between PDAs and Utility-Dependent Attributes (UDAs) first.

For simplicity, we have the following collective method:

Algorithm 2: Collective method

Input: G, PDAs, UDAs
Output: collective method
1 **if** PDAs $\cap$ UDAs $= \Phi$ **then**
2 **removing** PDAs;
3 **else then**
4 **removing** PDAs - *Core*;
5 **perturbing** *Core*

Algorithm 2 shows that if there are no shared attributes between PDAs and UDAs, we just need to remove the PDAs since they have no contributions on utility (Step 2). Conversely, with the same reason, we remove the difference set between PDAs and the shared attribute set *Core* (step 4). For the shared attributes, perturbing them to optimize the privacy-utility tradeoff (step 5).

The details of the perturbing method on *Core* in Algorithm 2 are presented in Subsection 3.6.1.

3.6.1 Perturbing

We formally define the shared attributes as a *Core*.

Definition 3.6.1. ***Core.*** *Given an information system* $\Gamma = (V,\ A = C \cup D), D = D_u \cup D_p$, *where* D_u *and* D_p *are two decision attribute sets for utility and privacy, respectively. We*

say that $C' \subseteq C$ is a core of D_u and D_p if $C' \subseteq R_u$ and $C' \subseteq R_p$, where R_u and R_p are the reduct of C for $\Gamma = (V, A = C \cup D_u)$ and $\Gamma = (V, A = C \cup D_p)$, respectively.

Our idea is to substitute each attribute in the *Core* with a generic attribute, which ensures that third party users cannot get specific information to increase prediction accuracy on sensitive attributes, while guarantees no significant accuracy reduction on data utility. Moreover, the higher level of generalization, the more preference to privacy for the utility-privacy tradeoff. Since there are different levels of generalization, the generic attributes can be organized into a hierarchy, which is formally defined as follows:

Definition 3.6.2. ***Generic Attribute Hierarchy.*** *A Generic Attribute Hierarchy (GAH) is a finite hierarchical ordering. The first layer of the ordering is one of the privacy-dependent attributes, and each parent layer is a generic of the sublayer.*

Definition 3.6.2 indicates that the ancestor of the GAH is the highest level of generalization of initial attributes. Substituting one privacy-dependent attribute with the ancestor of the GAH would render the highest level of privacy. For example, if one attribute value in core is for category *favorite movies*, the corresponding GAH can be

$$\text{Star Wars} \rightarrow \text{ Fantasy} \rightarrow \text{American film}$$

This indicates that we can substitute original attribute "Star Wars" with "American film", in order to get the highest level of generalization. We could also substitute it with "Fantasy" to give more preference to utility for the utility-privacy tradeoff since "Fantasy" is more specific than "American film". Hence, GAH guarantees that we can programmatically determine which level of generic value should be chosen to optimize the privacy-utility tradeoff.

Algorithm 3 presents the generation process of the generic values for guaranteeing optimal ϵ-utility.

Algorithm 3: Generate generic value

Input: Core, ϵ = utility threshold
Output: GAH
while $\max_{c \in \mathcal{C}} \Lambda(G', \mathcal{K}, X_{non}) - \max_{c' \in \mathcal{C}} \Lambda(\mathcal{K}, X_{non}) \geq \epsilon$ **do**
    further generate all the current attributes;
return *Perturbed Core*

3.7 Evaluation

3.7.1 Datasets

In our experiments, we investigate three different Facebook datasets. The first one is the SNAP Facebook dataset[1] which contains user friendships and a number of node attributes such as gender, birthday, position, employer, location, *etc.* The other two are the Facebook dataset containing all the Facebook friendships at Caltech and MIT in 2005, as well as a number of node attributes such as student/faculty status flag, gender, graduation year, academic major, *etc.* [2] For convenience, we denote these three datasets as SNAP, Caltech, and MIT, respectively. In Caltech and MIT, each attribute is specified by a numeric value and each of which indicates a corresponding attribute. However, in SNAP, each attribute is specified by a 0/1 value and each of which indicates the absence/presence of the corresponding attribute. For example, attribute "EducationDegree: undergraduate; master; PHD" with attribute value 010 means that the attribute value is master. For convenience, we map each attribute in SNAP into an unique numeric value in each attribute category. For example, the above attribute value 010 in Education degree is mapped to 2.

In Table 3.3, some general statistics about the three datasets are provided. It shows that all of the three graphs are almost fully connected.

[1] https://snap.stanford.edu/data/egonets-Facebook.html
[2] http://www.michaelzimmer.org/2011/02/15/facebook-data-of-1-2-million-users-from-2005-released/

Table 3.3. General statistics about the three datasets

Network property	SNAP	Caltech	MIT
Number of nodes	792	769	6440
Number of friendship links	14024	16656	251252
Number of attributes for each user	20	7	7
Number of values for decision attribute	2	4	7
Number of components in the graph	10	4	18
Nodes in largest connected component	775	762	6402
Edges in largest connected component	14006	16651	251230
Diameter longest shortest path	10	6	8

3.7.2 Experiment Settings

In our experiments, we regard *gender* in SANP and *student/faculty status flag* (*flag* for short) in Caltech and MIT as sensitive attributes.

Table 3.3 shows that there are 2, 4, and 7 attribute values in SNAP, Caltech and MIT, respectively, which are regarded as class labels here.

We predict a sensitive attribute with the following attack models: 1) the attack model with absence of link information (AttrOnly), 2) the attack model with absence of attribute information (LinkOnly), and 3) the attack model based on collective inference (CC).

As mentioned in Section 3.4, a major issue is raised if directly executing LinkOnly requires that at least one of the neighbors of each unlabeled user locates in the training set (as shown in Equation (4.3)). Hence, in our experiments, we first predict the class label of those unlabeled nodes by classifying their attribute sets. Next, we predict the class label of any user u_i by calculating the weighted average probability of its neighbors with one class label (as calculated in Equation (4.3)).

Moreover, CC employs attribute based classifier to predict the approximate class label at the bootstrap stage. Then, it uses classifier that based on both attribute and link, M_{AR}, to refine the results. In our experiments, we employ the following M_{AR}

$$\alpha P_A\{y_t^i\} + \beta P_L\{y_t^i\} \tag{3.5}$$

where $P\{y_t^i\}$ and $P_L\{y_t^i\}$ are the probabilities of u_i with label y_t, assigned by AttrOnly

Table 3.4. Information of the Reduct Systems for SNAP, Caltech and MIT

Decision attribute	No. of condition attributes
Gender in SNAP	$19 \rightarrow 13$
Flag in Caltech	$6 \rightarrow 5$
Flag in MIT	$6 \rightarrow 5$

and LinkOnly, respectively. $\alpha + \beta = 1$, where α and β represent the ratio of AttrOnly and LinkOnly, respectively. The values of α and β are determined by dataset features. Specifically, α is larger than β iff the node attributes are more indicative than node relations. To determine α and β, we study a set of experiments with multiple combinations and find the optimal one that renders the best prediction accuracy for CC. In Section 3.7.3, we set both α and β as 0.5; namely, an average prediction result assigned by AttrOnly and LinkOnly is expected. In Section 3.7.4, the utility and privacy under several pairs of α and β would be discussed.

For the attribute-based classifier utilized in AttrOnly, LinkOnly and CC, we carry it out with three techniques: RST, Navie Bayes and KNN [76]. Hence, with different attribute-based classifiers, AttrOnly can be further specified as: 1) RST, 2) Navie Bayes, 3) KNN; LinkOnly can be further specified as: 4) LinkOnly-RST, 5) LinkOnly-Bayes, 6) LinkOnly-KNN; and CC can be further specified as: 7) ICA-RST, 8) ICA-Bayes, 9) ICA-KNN.

3.7.3 Effect of Attribute-removal and Link-removal Methods on Inference Attacks

In this part, we aim to protect against inference attacks with the following sanitization methods: 1) Attribute removal: remove the most privacy dependent attributes, namely, the attributes in the reduct system (Section 3.5.1), and 2) Link removal: remove the distinguishable links (Section 3.5.3).

Table 3.4 lists the information of the reduct systems for these three datasets. We can see that in Table 3.4, the number of condition attributes is reduced from 19 to 13 in SNAP and from 6 to 5 in Caltech and MIT, respectively.

SNAP Fig.3.2(a), Fig.3.2(b) and Fig.3.2(c) show the prediction accuracy of different attack models on SNAP with the removal of the most privacy dependent attributes. Fig.3.2(d), Fig.3.2(e) and Fig.3.2(f) show the prediction accuracy of different attack models on SNAP with the removal of the indistinguishable links. As we can see from Fig.3.2(a), Fig.3.2(b) and Fig.3.2(c), removing the most privacy dependent attributes is generally successful in reducing the prediction accuracy on sensitive attributes.

It is shown that there is a decrease in the prediction accuracy with more and more attributes being removed. Surprisingly, however, the accuracy of LinkOnly does not decrease significantly while we remove attributes. For LinkOnly, as discussed in Section 3.7.2, we first predict the class labels of those unlabeled nodes by classifying their attribute sets; hence, the accuracy decrease of attribute-based classifier should also render the accuracy decrease for LinkOnly. A possible explanation is that just a small part of the nodes need labels in the first step of LinkOnly through classifying attribute set, since most of the labeled nodes are in the training set. Hence, removing attributes do not have a significant influence. Clearly, we can see that CC generally outperforms AttrOnly and LinkOnly.

The results in Fig.3.2(d), Fig.3.2(e) and Fig.3.2(f) show that removing the indistinguishable links is generally successful in reducing the prediction accuracy on sensitive attributes. However, we find a surprising phenomena in Fig.2: a volatile prediction accuracy after the removal of a single attribute or link. Especially, a much more volatile prediction accuracy after the removal of a single link. For the volatility related to attribute, it is a result of large class size difference in the SNAP dataset. Since approximately 65% of the nodes in SNAP are "male" and there are no attributes that are highly dependent on gender, a small change of attributes can affect the prediction accuracy in uncontrollable ways. For the volatility related to links, it is a result of the local optimal link-removal strategy. Since the link-removal strategy always manipulates the most indistinguishable link with respect to each user, it cannot guarantee the removed link is globally optimal. Therefore, a small change of links can also affect the prediction accuracy in uncontrollable ways.

Caltech Fig.3.3(a), Fig.3.3(b) and Fig.3.3(c) show the prediction accuracy of different attack models on Caltech with the most privacy dependent attributes removed. Fig.3.3(d), Fig.3.3(e) and Fig.3.3(f) show the prediction accuracy of different attack models on Caltech with the removal of the indistinguishable links. As we can see from Fig.3.3(a), Fig.3.3(b) and Fig.3.3(c), compared with the results on SNAP, there is a much more volatile prediction accuracy after the removal of a single attribute. This is a result of larger class size difference in Caltech than that of SNAP. Since approximately 72% of the nodes in Caltech have a same class label and there are no attributes that are highly dependent on flag, a small change of attributes can affect the prediction accuracy in uncontrollable ways.

MIT Fig.3.4(a), Fig.3.4(b) and Fig.3.4(c) show the prediction accuracy of different attack models on MIT with the most privacy dependent attributes removed. Fig.3.4(d), Fig.3.4(e) and Fig.3.4(f) show the prediction accuracy of different attack models on MIT with the removal of the indistinguishable links. Fig.4 shows that removing the most privacy dependent attributes or indistinguishable links is generally successful in reducing the prediction accuracy on sensitive attributes. As we can see from Fig.3.4(a), Fig.3.4(b) and Fig.3.4(c), compared with the results on Caltech, there is a less volatile prediction accuracy after the removal of a single attribute. This appears to be a result of larger class size difference in the Caltech dataset than that of the MIT. Approximately 67% of the nodes in MIT have a same class label and there are no attributes that are highly dependent of flag.

Fig.3.5(a) and Fig.3.5(b) show the prediction accuracy of different attack models on MIT with the most privacy dependent attributes and indistinguishable links removed simultaneously. As shown in Fig.3.5, the prediction accuracy is more sensitive to the removal of attribute than the removal of link.

3.7.4 Effect of Collective Method on Inference Attacks

We further test the collective method to evaluate the effectiveness of our data sanitization. Since there are no utility and privacy expectation specified for each dataset, we select

Table 3.5. Setting of utility attribute and privacy attribute

	Utility attribute	Privacy attribute
SNAP	education type	gender
Caltech	gender	flag
MIT	gender	flag

two attributes as privacy attribute and utility attribute, respectively. The selection of the above two attributes is listed in Table 3.5. We attempt to evaluate the effectiveness of our method in achieving a desired privacy/utility tradeoff: reducing the prediction accuracy on sensitive attribute while ensuring the prediction accuracy on utility atteibute.

Since each attribute has a numeric value, we cannot generate a generic value from the semantic view directly. However, we can map several attribute values to an interval and generalize them with an unique value in this interval. Algorithm 4 is used to generate generic attribute values. For each attribute category h_r in *Core*, Algorithm 4 first calculates the maximum and minimum attribute values of all the users for h_r (steps 2-3). Then, it calculates the range between MAX and MIN under generic level L (step 4). Finally, for each user i, Algorithm 4 maps its original attribute value $x_{i,r}$ to $\lfloor (X_{i,r} - MIN)/Range \rfloor$ (steps 5-7). In Algorithm 4, perturbing degree decreases with the increase of generalization level L.

Algorithm 4: Generate generic value

Input: *Core*, L = generalization level
Output: Generic attribute set with level L
1 **for** *each attribute category* $h_r \in$ Core **do**
2 $\quad MAX_r = \max(x_{1,r}, x_{1,r}, \ldots, x_{|V|,r})$;
3 $\quad MIN_r = \min(x_{1,r}, x_{1,r}, \ldots, x_{|V|,r})$;
4 $\quad Range_r = \lfloor (MAX_r - MIN_r)/L + 1 \rfloor$;
5 $\quad$ **for** $i = 1$ *to* $|V|$ **do**
6 $\quad\quad x_{i,r} = \lfloor (x_{i,r} - MIN)/Range \rfloor$;
7 return $x_{i,r}$

According to Algorithm 2, the information for PDAs, UDAs and Core for SNAP, Caltech

Table 3.6. Information for PDAs, UDAs and Core

Dataset	No. of UDAs	No. of PDAs - Core	No. of Core
SNAP	7	6	6
Catech	3	2	1
MIT	3	2	1

Table 3.7. Maximum utility/privacy under collective, attribute removal and link removal methods with $\alpha = 0.5$, $\beta = 0.5$

Dataset	Collective	Attribute removal	Link removal
SNAP	1.1967	1.1639	1.1639
Caltech	1.5273	1.3433	1.3433
MIT	1.2636	1.1881	1.1931

and MIT are shown in Table 3.6.

We test multiple levels of generalization (set generalization level as $L = 5, 6, 7, 8$) and compare the collective method with the data removal and link removal sanitization methods. We use utility/privacy as privacy-utility tradeoff criteria to evaluate the performance of these three data-sanitization methods.

Table 3.7 shows the maximum utility/privacy under these three methods, with $\alpha = 0.5$ and $\beta = 0.5$. From Table 3.7, we observe that the collective method achieves the best privacy/utility tradeoff with ratio 1.1967, 1.5273 and 1.2636 in SNAP, Caltech and MIT, respectively. Table 3.8, Table 3.9 and Table 3.10 show the utility/privacy under different generalization levels, and different numbers of removed attributes and links. In Table 3.8, Table 3.9 and Table 3.10, "R-Attr", "R-Link" and "Uti/pri" represent "Number of Removed attribute", "Number of Removed link" and "Utility/privacy", respectively. As shown in Table 3.8, Table 3.9 and Table 3.10, utility to privacy ratio decreases with the increase of perturbing degree (L from 5 to 8). Moreover, utility to privacy ratio decreases as well with more and more attributes and links being removed. Additionally, we observe that our proposed collective method generally outperforms attribute removal and link removal method.

Table 3.8. General statistics about priacy/utility on SNAP with $\alpha = 0.5$, $\beta = 0.5$

L	Uti/pri	No. of R-Attr	Uti/pri	No. of R-Link	Uti/pri
5	1.1613	0	1.1639	0	1.1639
6	1.1803	3	1.0862	200	1.1500
7	1.1967	6	0.9524	400	1.1333
8	1.1967	9	0.9375	600	1.1148

Table 3.9. General statistics about priacy/utility on Caltech with $\alpha = 0.5$, $\beta = 0.5$

L	Uti/pri	No. of R-Attr	Uti/pri	No. of R-Link	Uti/pri
5	1.4839	0	1.3433	0	1.3433
6	1.4918	1	1.1970	400	1.2464
7	1.5112	2	1.0274	800	1.2206
8	1.5273	3	0.9865	1200	1.1690

Table 3.10. General statistics about priacy/utility on MIT with $\alpha = 0.5$, $\beta = 0.5$

L	Uti/pri	No. of R-Attr	Uti/pri	No. of R-link	Uti/pri
5	1.2313	0	1.1881	300	1.1931
6	1.2425	1	1.0469	600	1.1901
7	1.2580	2	1.0342	900	1.1897
8	1.2636	3	0.9698	1200	1.1798

Table 3.11. Maximum utility/privacy under collective, attribute removal and link removal methods with $\alpha = 0.1$, $\beta = 0.9$

Dataset	Collective	Attribute removal	Link removal
SNAP	1.3019	1.3148	1.4800
CalTech	1.4032	1.3770	1.3770
MIT	1.2274	1.2121	1.2239

Table 3.12. Maximum utility/privacy under collective, attribute removal and link removal methods with $\alpha = 0.9$, $\beta = 0.1$

Dataset	Collective	Attribute removal	Link removal
SNAP	1.1356	1.1754	1.1930
CalTech	1.3968	1.2985	1.2985
MIT	1.2674	1.2101	1.2132

Furthermore, we evaluate the maximum utility/privacy under different combinations of α and β: $\alpha = 0.1$, $\beta = 0.9$ and $\alpha = 0.9$, $\beta = 0.1$. The results are shown in Table 3.11 and Table 3.12. Table 3.7, Table 3.11 and Table 3.12 show that utility/privacy value of collective method is always better than that of attribute removal and link removal method, when an average prediction result are assigned by AttrOnly and LinkOnly, *i.e.*, $\alpha = 0.5$ and $\beta = 0.5$.

3.8 Conclusions

We address two issues in this paper: (a) how exactly third party users launch an inference attack to predict sensitive information of users, and (b) are there effective strategies to protect against such an attack to achieve a desired privacy-utility tradeoff. For the first issue, we show that collectively utilizing both attribute and link information can significantly increase prediction accuracy for sensitive information. For the second issue, we explore the dependence relationships for utility/public attributes, and privacy/public attributes. Based on these results, we propose a Collective Method that take advantages of various data manipulating methods to guarantee sanitizing user data does not incur a bad impact on data utility. Using Collective Method, we are able to effectively sanitize social network data prior

to release. The solutions for the two addressed issues are proven to be effective towards three real social datasets.

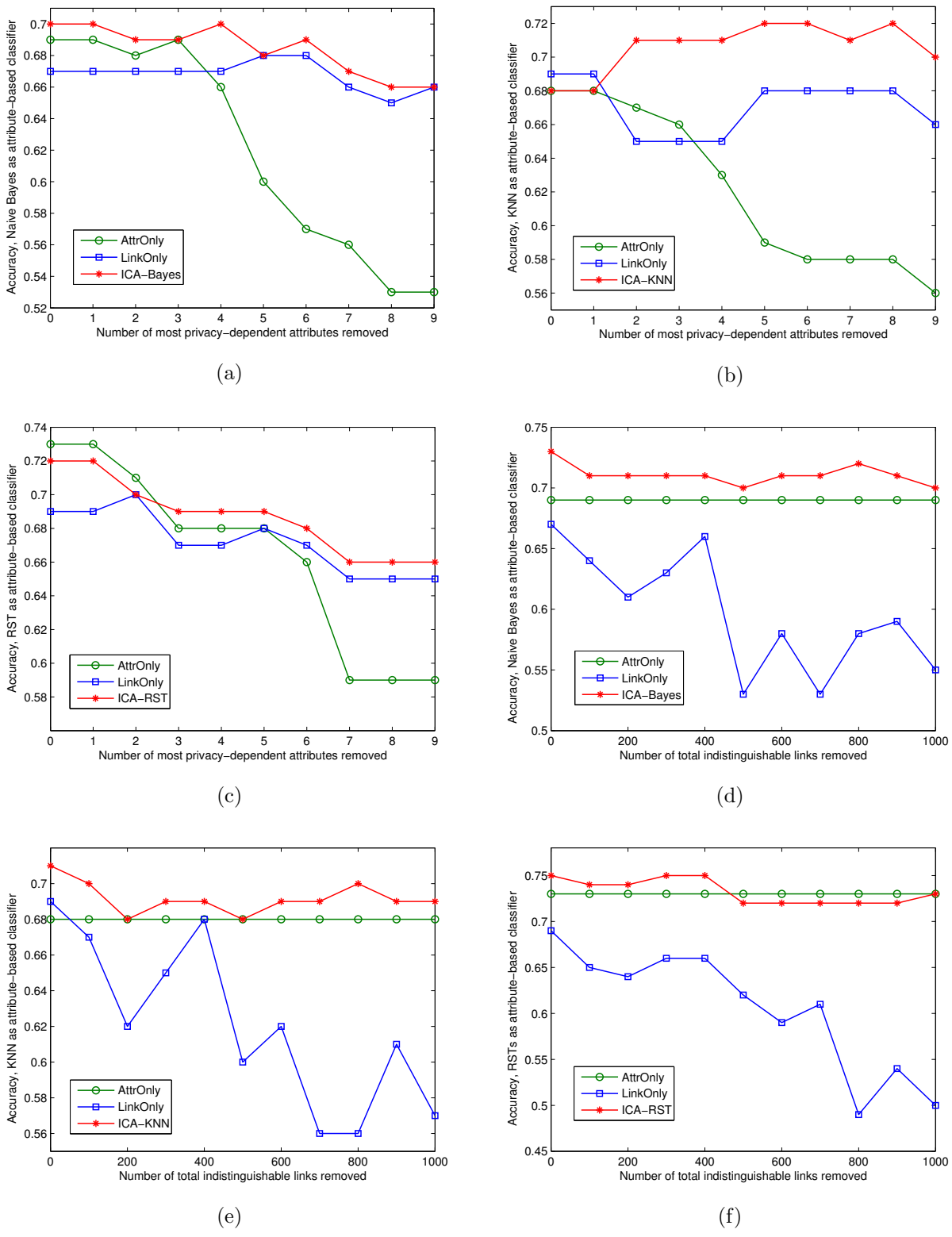


Figure 3.2. Sensitive attribute prediction accuracy on SNAP with different attack models. With most privacy-dependent attributes removed, and (a) Bayes, (b) KNN, (c) RST as attribute-based classifier; With indistinguishable links removed, and (d) Bayes, (e) KNN, (f) RST as attribute-based classifier.

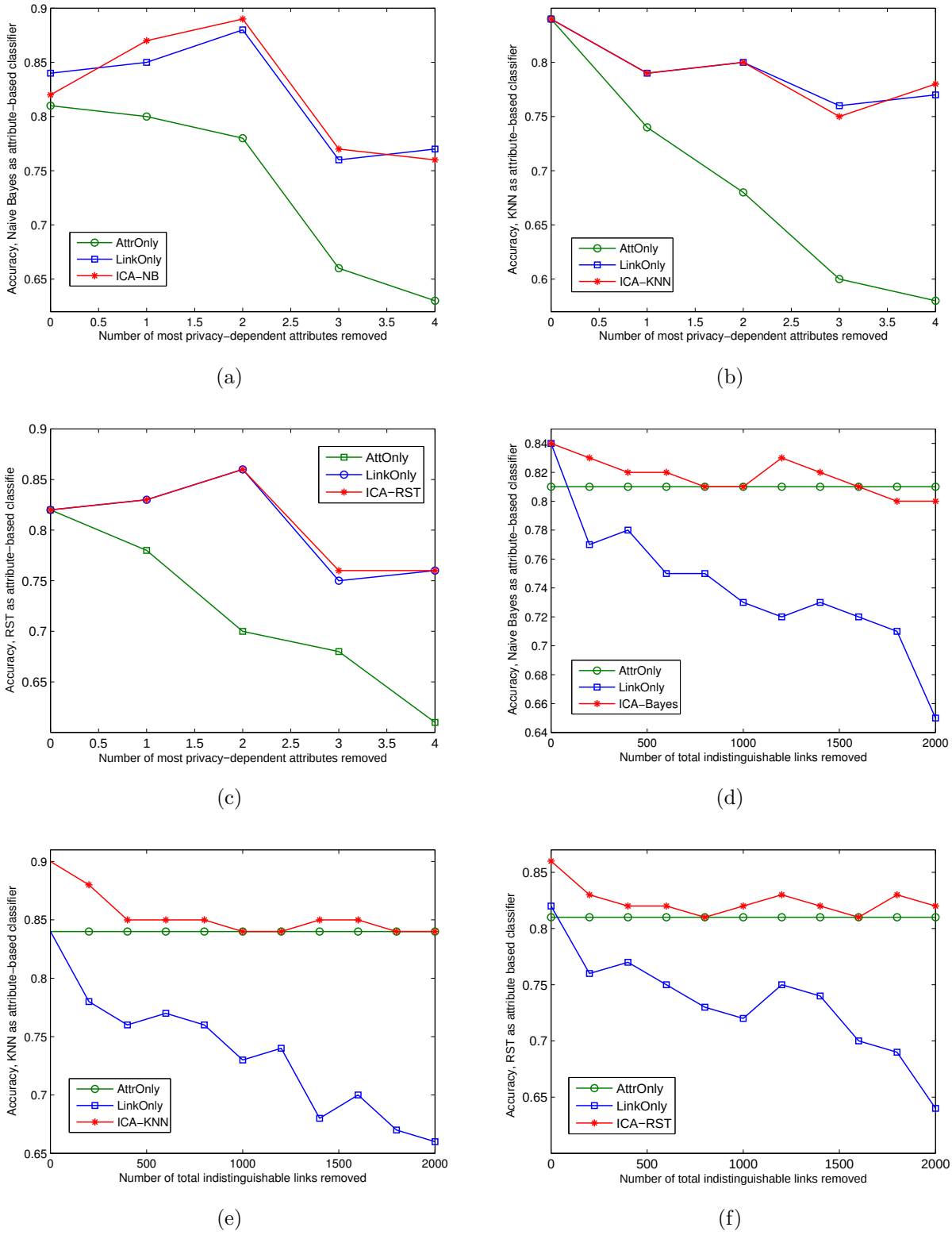


Figure 3.3. Sensitive attribute prediction accuracy on Caltech with different attack models. With most privacy-dependent attributes removed, and (a) Bayes, (b) KNN, (c) RST as attribute-based classifier; With indistinguishable links removed, and (d) Bayes, (e) KNN, (f) RST as attribute-based classifier.

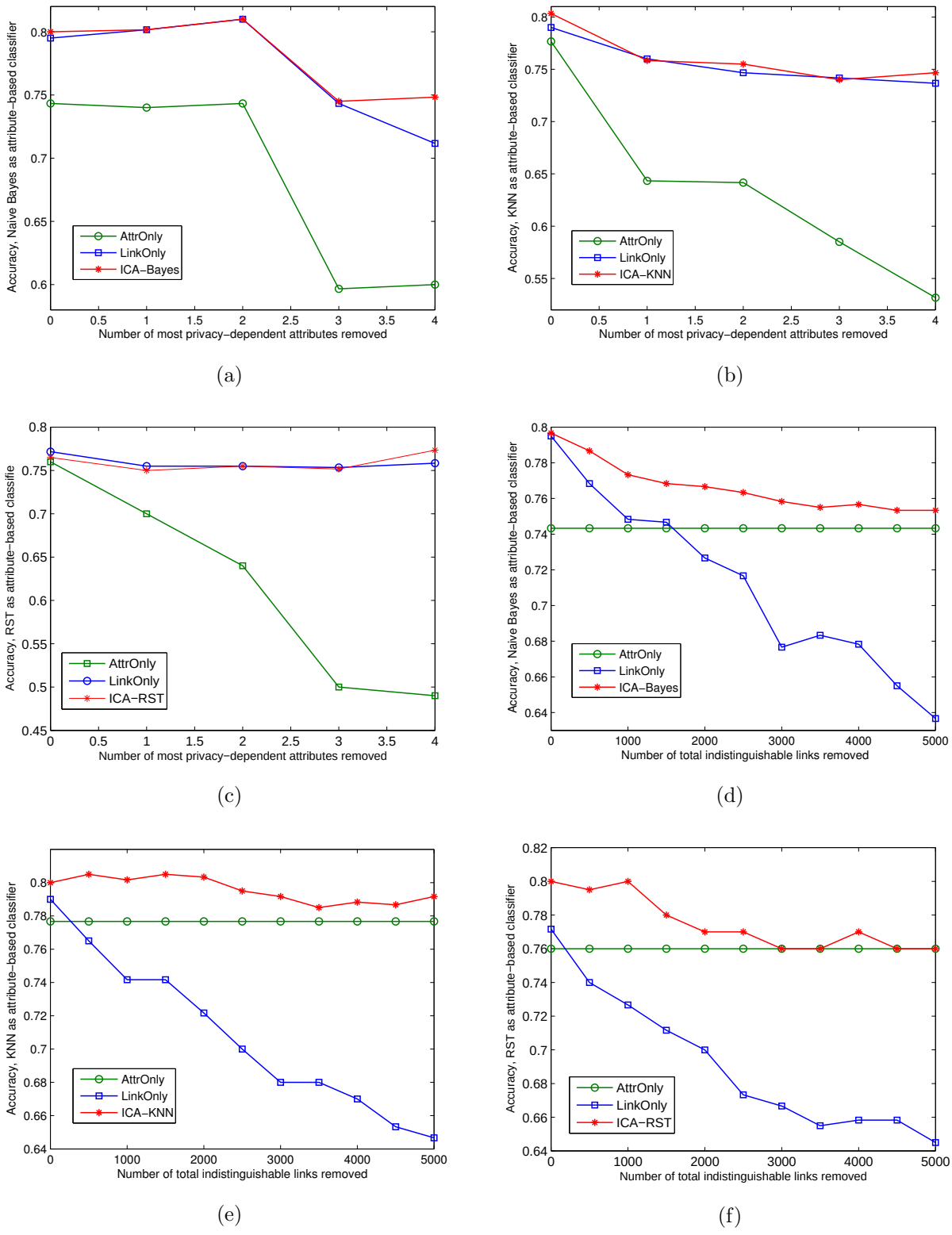
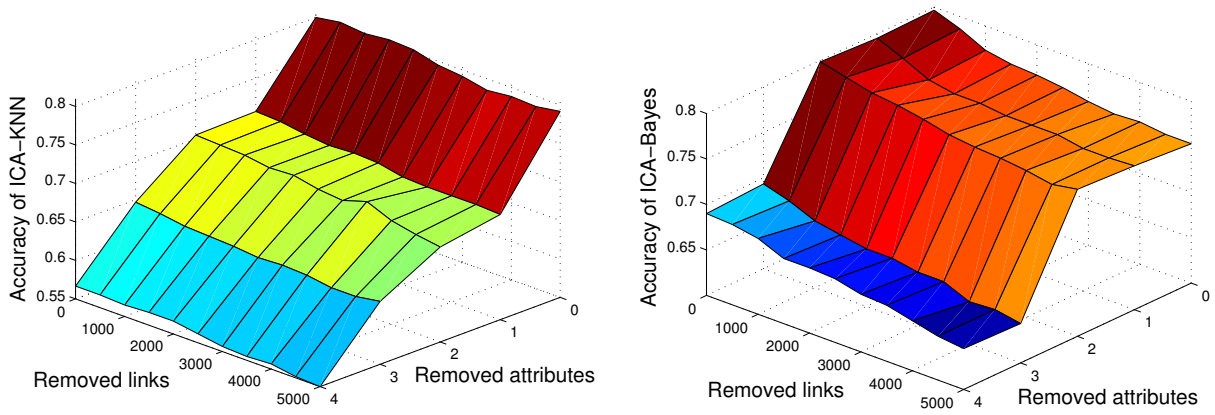


Figure 3.4. Sensitive attribute prediction accuracy on MIT with different attack models. With most privacy-dependent attributes removed, and (a) Bayes, (b) KNN, (c) RST as attribute-based classifier; With indistinguishable links removed, and (d) Bayes, (e) KNN, (f) RST as attribute-based classifier.



(a) (b)

Figure 3.5. Predicting accuracy on MIT with the most privacy dependent attributes and indistinguishable links removed simultaneously: (a) ICA-KNN as attack model; (b) ICA-Bayes as attack model.

Chapter 4

TRADEOFF BETWEEN PRIVACY AND CUSTOMIZED DATA UTILITY FOR SOCIAL DATA PUBLISHING

4.1 Introduction

Among the many big data resources, social networks contribute considerable amount of data covering all the aspects of frontend and backend. Facebook has 1.65 billion users with 1 billion active users per month, Twitter has 600 million users with 0.5 billion tweets published per day, Amazon has 304 million users with 9.65 billion items traded per year, Tencent QQ has 829 million active users with up to 210 million simultaneous online users, WeChat has over a billion users with 700 million active users, *etc.* With such large scale of and variety of data, Social Network Analysis (SNA) becomes increasingly important for classifying end users, predicting buying interests, foretelling event occurrence, *etc.* Recent years have witnessed the boom of social networks, offering a great opportunity for SNA to prompt more novel applications.

Although the abundant social data bring valuable benefits, they unfortunately raise stringent privacy concerns as well. Each social network user is generally associated with an attribute set which may contain sensitive attributes like location, gender, sexual orientation, *etc.* Such personal information could be exploited by third parties like data analysts, marketer, or social media itself. Any third parties with malicious intentions on sensitive information of users can be viewed as adversaries and they breach user privacy by collecting sensitive data first. People now begin to concern about the privacy issue and become more conservative in publishing personal and sensitive data, which may degrade data publishing scale and drive users to publish anonymized data. Therefore, the conflict between privacy concerns and data utility promotes adversaries to exploit sensitive information contained in the published data.

Concerns derived from inference attacks towards sensitive information contained in user data is represented as *latent-data privacy*, where the inference attacks usually employ statistical analysis, machine learning or data mining techniques to infer sensitive information. For instance, suppose a user does not disclose her opinions and interests online. Unfortunately, it is easy to predict some of her opinions and interests if it is publicly known that she is affiliated with any particular organization or club. ABCNews.com and Boston Globe [69] shown it is achievable to infer the sexual orientation of a user through mining a Facebook subnetwork involving the user's friendship relations, gender, and other attributes. Latent-data privacy breaches could incur serious negative repercussions.

Publishing sanitized data is generally adopted to protect latent-data privacy. Data sanitization methods introduce noises by sanitizing attribute sets or social links. Although sanitizing publicly available data can help with protecting latent-data privacy, such simple methods could also reduce data utility for SNA. On the one hand, some user attributes are indicative for specific social analysis which is expected to be accurately predicted. For instance, a SNA server utilizes published Facebook data to make movie recommendation for target users. Unfortunately, some dominant attributes, such as "gender", may have been sanitized to protect latent-data privacy, degrading recommendation performance. On the other hand, in addition to sanitizing attributes, sanitizing social network links can distort friendship relations among users and change one's social status, which is another reason of reducing data utility for SNA. For example, social link sanitization can turn an influential user to an unsocial one. Therefore, effective privacy preserving SNA strategies are crucial for big social network data.

In this work, we explore how to balance the tradeoff between latent-data privacy and data utility. We assume adversaries collect user data, and some privacy-unconscious users publish their sensitive latent information. We first formalize the metrics to measure data utility loss and latent-data privacy. Then, we propose two data sanitization methods that sanitize social attributes and links, respectively. Finally, data-sanitization strategies are proposed, which should not degrade the benefits brought by social network data, while

sensitive latent information can still be protected.

To measure data utility loss, we introduce prediction accuracy deviation and network structure disparity. Both of them cause utility loss because of the employed data sanitization strategy. We investigate how to measure them and their relationship. Previous works usually consider them separately. Network structure disparity not only affects prediction accuracy, but also limits social interaction among users. The current metrics do not comprehensively measure data utility and could sacrifice more utility in realizing privacy-utility tradeoff. Our work does consider both prediction accuracy deviation and network structure disparity. For latent-data privacy, we expect our data sanitization strategy can combat against powerful adversaries with abundant prior knowledge who launch inference attacks. Thus, it is necessary to figure out how adversaries launch inference attacks. Previous works primarily assume relatively weak adversaries such that the proposed data sanitization strategy is not effective. Our work does consider this problem and quantify the capabilities of adversaries.

The previous studies for privacy-utility tradeoff have several deficiencies. First, attribute-sanitization and link-sanitization are separately considered, degrading the privacy preserving effect. Second, relatively weak adversaries are assumed so that the proposed data sanitization strategies are not sufficient to combat against powerful adversaries. Third, structure utility loss caused by social structure disparity is ignored so that preserved utility is overestimated. Therefore, the previous studies cannot effectively optimize the tradeoff between latent-data privacy and data utility. In this paper, we identify an optimization problem seeking a data sanitization strategy to realize the maximum latent-data privacy with customized data utility. Our main contributions are summarized as follows:

1. We consider prediction utility loss and structure utility loss simultaneously rather than considering them separately.

2. We assume powerful adversaries who can launch optimal inference attacks instead of weak adversaries.

3. Rather than separately considering attribute-sanitization and link-sanitization, we col-

lectively sanitize social links and attributes.

We organize the paper as follows. Section 4.2 introduces Network model and problem definition. Section 4.3 introduces the prediction method for latent attributes and data-sanitization method. In Section 4.4, privacy and utility metrics are introduced. The data sanitization strategy to optimize the privacy-utility tradeoff is presented in Section 4.5. The performance evaluation are shown in Section 4.6. Section 4.7 concludes the paper.

4.2 Problem Statement

4.2.1 Social Network Model

Definition 4.2.1. ***Social network.*** *Social network is represented by graph model $G(V, E, \mathcal{X})$, with user set V, link set E, and the set of attribute sets, $\mathcal{X}$. For any link $e_{ij} \in E$ between users u_i and u_j, $e_{ij} \in E$ also indicates $e_{ji} \in E$.*

Definition 4.2.2. ***Attribute set.*** *For user $u_i \in V$, its attribute set is represented by an attribute vector $X_i \in \mathcal{X}$. Each attribute $x_j \in X_i$ $(1 \leq j \leq |X_i|)$ takes value(s) from the j-th dimension.*

For social network data, a SNA server performs analysis to predict users' latent information such as preferences. Then, according to the predicted results, the corresponding services are provided. For example, a SNA server can predict movie preference of users by classifying the users into different classes such as *action*, *adventure*, *comedy*, *etc.* However, adversaries also attempt to gain benefit from users' social relationships and attribute set to infer sensitive latent information. These two types of latent information related to data utility and latent-data privacy are denoted as Sensitive Latent Attributes (SLA) and Non-Sensitive Latent Attributes (NSLA), respectively.

Definition 4.2.3. ***SLA.*** *SLA is a set of unpublished sensitive attributes, yet such attributes could be predicted from published social network data combined with prior knowledge.*

Definition 4.2.4. ***NSLA.*** *NSLA is a set of unpublished non-sensitive attributes, yet such attributes can be predicted from published social network data combined with prior knowledge.*

We expect NSLA can be accurately predicted so that satisfactory services can be guaranteed. Conversely, to protect the privacy of SLA, we expect SLA does not being predicted accurately. Furthermore, social network structure should be preserved such as node degree, centrality, betweenness, *etc.* Thus, there exists a tradeoff between latent-data privacy and data utility. Utility and latent-data privacy are formally defined as follows.

Definition 4.2.5. ***Latent-data privacy.*** *Latent-data privacy preserving is to protect the SLA of each user.*

Definition 4.2.6. ***Utility.*** *The utility of a social network dataset is high iff 1) a SNA server has a high prediction accuracy for NSLA; and 2) the social network structure is effectively preserved.*

For the sake of brevity, we omit the subscript and use X and X' to denote an original and sanitized attribute set of a user, respectively, in the rest of the paper without confusion.

4.2.2 Model of Adversaries

We assume powerful adversaries with abundant prior knowledge about users, and they can launch optimal inference attacks to infer the SLA of each user. This assumption allows the constructed data-sanitation method can combat against adversaries with a larger range of capability.

There exists a prior probability for a user's attribute vector X, denoted as $\psi(X)$, which represents the probability of a user with attribute set X. For a user, all her possible attribute sets satisfy $\sum \psi(X) = 1$. We call the set of $\psi(X)$ as a user's profile.

Definition 4.2.7. ***Profile.*** *The profile of a user is a set of probabilities* $\Psi = \{\psi(X_1), \psi(X_2), \ldots, \psi(X_k)\}$, $\sum_{1 \le i \le k} \psi(X_i) = 1$*, where each* $\psi(X_i)$ *is the probability of a user with attribute set* X_i *and* k *is the number of possible attribute sets.*

Table 4.1. Major symbols

Parameter	Definition
$\mathcal{X}$	Set of attribute sets
X_i	Attribute set of user u_i
x_j	j-th attribute
$\psi(X)$	Prior probability of attribute set X
l_t^i	t-th latent attribute, l_t, of u_i
$P(l_t^i)$	Probability of u_i with latent attribute l_t
$W_{i,j}$	Weight between u_i and u_j
$f(X'\|X)$	Attribute sanitization strategy
$\mathcal{L}(X'\|X)$	link sanitization strategy
ϵ	Structure-utility loss threshold
δ	Prediction-utility loss threshold

First, we assume adversaries know each user's profile. Second, adversaries are assumed to know the data-sanitization strategy employed to realize the tradeoff between utility and privacy. Based on the above knowledge, optimal inference attacks are launched by adversaries.

4.2.3 Problem Definition

In this paper, we study the following problem.

Input:

(1) Social graph G, SLA and NSLA of users.

(2) Utility thresholds ϵ and δ.

Output:

The data sanitization strategy that minimizes the predication accuracy for unpublished SLA and satisfies utility threshold ϵ and δ.

For clarity, the meanings of the symbols are summarized in Table 4.1.

4.3 Preliminaries

In this section, the prediction method is presented to predict both SLA and NSLA of a user based on published social data.

4.3.1 Prediction Method for Latent Attributes

We assume powerful adversaries that launch inference attacks by utilizing all publicly available knowledge including social links and attribute sets. Therefore, the prediction method predicts latent information considering social links and attribute sets collectively to increase prediction accuracy.

Link knowledge is important for predicting latent information in social networks. Therefore, we consider u_j' latent information when predicting u_i' latent information, where $u_j \in N_i$ and N_i denotes the neighbor set of u_i. For clarity, u_i with latent attribute l_t is denoted as l_t^i.

For brevity, the probability of u_i to have latent attribute l_t is denoted as $P(l_t^i)$. The average probability of u_i' neighbors with latent attribute l_t is calculated as:

$$P(l_t^i|N_i) = \frac{1}{|N_i|} \sum_{u_j \in N_i} P(l_t^j) \tag{4.1}$$

However, directly computing the average probability may incur overfitting. In practice, close neighbors should have larger impact for each other on the determination of latent information. To avoid overfitting, we introduce a weight to evaluate the impact of one neighbor for target user. We assume that if more published attributes are shared by two friends, they tend to share more latent attributes. Then the weight $W_{i,j}$ between u_i and u_j is calculated as

$$W_{i,j} = \frac{|(x_1^i, \ldots, x_m^i) \cap (x_1^j, \ldots, x_n^j)|}{|X_i|} \tag{4.2}$$

Equation (4.2) computes the proportion of the shared attributes between u_i and u_j among u_i's attributes. Clearly, $W_{i,j} \neq W_{j,i}$. To determine l^i based on N_i, we combine Equation (4.1) and Equation (4.2) as follows,

$$P(l_t^i|N_i) = \frac{1}{|N_i|} \sum_{u_j \in N_i} P(l_t^j) \frac{W_{i,j}}{\sum_{u_k \in N_i} W_{i,k}} \tag{4.3}$$

It is easy to find that Equation (4.3) requires that at least one of the neighbors of each user to publish her latent attributes. Obviously, this strict condition is hard to be satisfied in real social networks. Therefore, it is inaccurate to predict the latent attributes of user u_i based on link information directly, since it is possible that few neighbors publish their latent attributes. To solve this problem, we first predict the latent attributes of those unpublished users through analyzing their attribute sets. Then, we predict the latent attributes of unpublished users through utilizing weighted link knowledge calculated by Equation (4.3).

Next, we present how to predict the latent attributes of a user through analyzing her attribute set. Given a user u_i with attribute set $X_i = \{x_1, \ldots, x_n\}$ and p potential latent attributes $l_1, \ldots, l_p$, the probability of u_i with latent attribute l_t is $\arg\max_{1 \le t \le p}[P(l_t^i | x_1, \ldots, x_n)]$.

To calculate the above value, based on Bayes Theorem, assuming that all attributes are independent, we have

$$\arg\max_{1 \le t \le p} \left[\frac{P(l_t^i) \times P(x_1|l_t^i) \times \ldots \times P(x_n|l_t^i)}{P(x_1, \ldots, x_n)} \right].$$

We find that $P(x_1, \ldots, x_n)$ is the same for any value of $P(l_t^i)$. Therefore, we only need to calculate

$$\arg\max_{1 \le t \le p} \left[P(l_t^i) \times P(x_1|l_t^i) \times \ldots \times P(x_n|l_t^i) \right].$$

4.3.2 Data Sanitization Method

In Section 4.3.1, we assume powerful adversaries that launch inference attacks by exploiting social links and attribute sets simultaneously. Therefore, in order to realize the tradeoff between privacy and utility, our objective is to sanitize both social links and attribute sets.

Attribute-sanitization method An attribute set could be sanitized in three ways, *adding* attributes, *removing* attributes, and *perturbing* attributes (replace one attribute with

another). Which methods should be employed to sanitize social data depends on data utility and privacy metrics and data semantics.

To prevent inference attacks on SLA, we can sanitize the most indicative attributes for each SLA which is publicly available to adversaries. With this objective, for a user with attribute set X, it is easy to determine the most indicative attribute x_j for any SLA $z_i \in Z$ by $\text{argmax}_j[\forall z_i \in Z : P(x_j|z_i)]$.

This allows us to determine a single attribute which is the most indicative for a SLA and sanitize it. Unfortunately, directly sanitizing the most indicative attributes for SLA can reduce utility if we don't consider the most indicative attributes for NSLA. For instance, consider the case to predict health conditions of users which could be viewed as NSLA. Health conditions and SLA such as sexual orientation share indicative attribute "gender". Therefore, although sanitizing "gender" reduces the prediction accuracy for SLA, it also reduces the prediction accuracy for NSLA.

To resolve the above conflict, we propose the following data sanitization method: (1) If there exist indicative attributes shared by SLA and NSLA, we *perturb* the shared indicative attributes; and *remove* the SLA except the shared indicative attributes; (2) If there does not exist any indicative attribute shared by SLA and NSLA, we *remove* the indicative attributes for SLA.

The next challenge is how to perturb the indicative attributes shared by SLA and NSLA. Our idea is to generalize each shared indicative attribute. For example, if a shared attribute is *idol: Jodon*, it can be generalized to *basketball star*. For each shared indicative attribute, we can organize potential generalized attributes into a hierarchy.

Link-sanitization method Unlike attributes, social links can only be sanitized by *adding* links and *removing* existing links. Similar with the attribute-sanitization method, a link-sanitization method should reduce the prediction accuracy for SLA and do not greatly reduce the prediction accuracy for NSLA. Unfortunately, unlike attributes, it is nontrivial to find the indicative links shared by SLA and NSLA, thus we focus on reducing the prediction

accuracy for SLA firstly when sanitize links and more constraints will be given later to guarantee utility.

For this goal, the concept of Vulnerable Link is introduced as follows:

Definition 4.3.1. ***Vulnerable link.*** *A vulnerable link of one user is the link whose removal will lower the prediction accuracy for the SLA of the user. The prediction accuracy for the SLA of u_i upon removing the vulnerable link e_{ij} is $\Lambda(E_i - e_{ij})$.*

From the above definition, it shows that $\Lambda(E_i - e_{ij}) \leq \Lambda(E_i)$. To protect SLA of u_i through removing links, we first identify a set of vulnerable links denoted as A_i. Second, for any $e_{ij} \in A_i$, we calculate the reduction of prediction accuracy for SLA upon removing e_{ij}. Then, we order the links in A_i according to the calculated prediction accuracy reduction. We next remove those links with the largest prediction accuracy reduction in A_i.

4.4 Metrics

Now we discuss how to measure utility and latent-data privacy. Our data sanitization strategy includes two parts: attribute sanitization strategy $f(X'|X)$ and link sanitization strategy $\mathcal{L}(X'|X)$. $f(X'|X)$ likes a transfer function that takes a user's attribute set X as input and outputs the sanitized one X'. Meanwhile, for an arbitrary user u_i, $\mathcal{L}(E_i'|E_i)$ can be viewed as a transfer function that takes u_i's link set E_i as input and outputs the sanitized one E_i'.

4.4.1 Utility metric

For data utility, two aspects need to be considered. First, the sanitized attribute set and social links should guarantee a SNA server can effectively infer the NSLA of users. Second, the sanitized network structure should not deviate from the original one very much. Worth to note that the second aspect expects that sanitizing social links does not distort friendship relations among users and does not change one's social status too much. We introduce two

parameters ϵ and δ to scale the above two aspects. Then, (ϵ, δ)-data utility can be defined as follows.

Definition 4.4.1. ***(ϵ, δ)-Utility.*** *Given social graph G, network disparity measurer $\mathcal{M}$, collective prediction method $\mathcal{C}$, NSLA set Y, accessible prior knowledge known to third party users $\mathcal{K}$, we say that G's sanitized graph G' satisfies (ϵ, δ)-utility if for any NSLA $y_i \in Y$,*

$$\text{(i). } \mathcal{M}(G, G') \leq \epsilon;$$
$$\text{(ii). } \Lambda_{\mathcal{C}}^{y_i}(G', \mathcal{K}) - \Lambda_{\mathcal{C}}^{y_i}(\mathcal{K}) \geq \delta,$$

where $\Lambda_{\mathcal{C}}^{y_i}(G)$ represents the prediction accuracy of collective prediction method $\mathcal{C}$ for NSLA y_i. ϵ is the super-threshold of social structure changes. δ measures how much added prediction accuracy is earned by adversaries through predicting with the published G'. Clearly, $\epsilon, \delta \geq 0$. To preserve data utility, both ϵ and δ are given by the data publisher.

Next, we define utility loss due to the data-sanitization strategy carried out on published data. Definition 4.4.1 shows that utility loss comes from two aspects: network structure disparity and prediction accuracy deviation for NSLA. Therefore, utility loss is defined based on the above two aspects: structure utility loss and prediction utility loss.

Definition 4.4.2. ***ϵ-Structure utility loss.*** *Structure utility loss estimates how much an arbitrary user u_i loses regarding network structure after sanitizing its social links. Structure utility loss of u_i is determined by the structure utility values of u_i's neighbors. For a given structure utility value metric, the ϵ-structure utility loss for u_i after sanitizing u_i' vulnerable link set $A_i \subseteq N_i$ is given by $SUL_i = \zeta(\mathcal{S}_{A_i}) \leq \epsilon$, where $\mathcal{S}_{A_i} = \{S_j | u_j \in A_i \subseteq N_i, S_j \in \mathbb{R}^*\}$, and S_j represents the structure utility value of user u_j.*

The structure utility value of a user reflects social structure properties, which can be measured by different metrics. In this paper, we use number of shared friends as structure utility metric. Unfriending a friend that shares a large of friends of one user has a bad effect

on the clustering coefficient of the user. Furthermore, we assume $\zeta(.)$ is an additive function, then $SUL_i = \sum_{u_j \in A \subseteq N_i} S_j \leq \epsilon$.

Since both attribute set and social links of a user are sanitized and we assume powerful adversaries predict SLA based on them simultaneously as shown in Section 4.3.1, prediction utility loss is derived from both of the disparity sources. Since social structure disparity is measured by ϵ-structure utility loss, prediction utility loss only needs to measure the prediction accuracy deviation derived from attribute sanitization.

To evaluate prediction utility loss due to sanitized attribute set, we introduce an attribute set disparity measurer d_u, such that $d_u(X, X')$ measures how much prediction utility loss there is if a SNA server performs analysis depending on X' rather than X. Thus, given $\psi(X)$, $f(X'|X)$, and $d_u(X, X')$, prediction utility loss can be calculated as the expectation of $d_u(X, X')$ over all X and X' for a user.

Definition 4.4.3. δ***-Prediction utility loss.*** *Prediction utility loss estimates the amount of prediction accuracy deviation for the NSLA of an arbitrary user u_i. For a given attribute set disparity measurer d_u, the δ-prediction utility loss for u_i after carrying out a data sanitization method on its attribute set X and social links, is given by $PUL_i = \sum_{X,X'} \psi(X) f(X'|X) d_u(X, X') \leq \delta$.*

Attribute set disparity measurer d_u is determined by data semantics. In different applications, d_u can be defined as Euclidean, Hamming, or Mahalanobis distance, *etc.*

4.4.2 Latent-data privacy metric

We assume powerful adversaries have the knowledge of user's profile $\psi(X)$ and our data-sanitization strategy. After obtaining the sanitized attribute set, adversaries calculate the posterior probability of X, conditional on X' with prior knowledge $\psi(X)$ and $f(X'|X)$:

$$Pr(X|X') = \frac{Pr(X, X')}{Pr(X')} = \frac{f(X'|X)\psi(X)}{\sum_X f(X'|X)\psi(X)}$$

Then, for each X with posterior probability $Pr(X|X')$, adversaries can predict the

user's SLA based on X and sanitized social links. We represent the SLA predicted from $Pr(X|X')$ as Z_X. Obviously, Z_X is related to the sanitized link set A such that we denote Z_X as the function of A, *i.e.*, $Z_X(A)$. Adversaries' goal is then to choose $\hat{Z}$ to minimize the user's conditional expected latent-data privacy, conditional on $Pr(X|X')$. For an arbitrary $\hat{Z}$, the user's conditional expected latent-data privacy is $\sum_X Pr(X|X')d_p(Z_X(A), \hat{Z})$, where $d_p(Z_X(A), \hat{Z})$ is the privacy disparity between $Z_X(A)$ and $\hat{Z}$.

For the minimized $\hat{Z}$, it is

$$\min_{\hat{Z}} \sum_X Pr(X|X')d_p(Z_X(A), \hat{Z}) \tag{4.4}$$

The latent-data privacy conditional on a given X' is given by Equation (4.4). Meanwhile, the probability of X' output by the sanitization method is $P(X') = \sum_X f(X'|X)\psi(X)$. Thus, unconditional expected privacy of the user's is

$$\begin{aligned} &\sum_{X'} \psi(X') \min_{\hat{Z}} \sum_X Pr(X|X')d_p(Z_X(A), \hat{Z}) \\ =&\sum_{X'} \min_{\hat{Z}} \sum_X \psi(X)f(X'|X)d_p(Z_X(A), \hat{Z}) \end{aligned} \tag{4.5}$$

We define

$$P_{X'} = \min_{\hat{Z}} \sum_X \psi(X)f(X'|X)d_p(Z_X(A), \hat{Z}). \tag{4.6}$$

Incorporating $P_{X'}$ into Equation (4.5), the users unconditional expected privacy is rewritten as

$$\sum_{X'} P_{X'}, \tag{4.7}$$

which is the user attempts to maximize by finding the optimal $f(X'|X)$.

Unfortunately, the minimum operator in Equation (4.6) makes the computation problem

nonlinear. Therefore, we can transform (4.6) into a series of linear constraints by

$$P_{X'} \leq \sum_{X} \psi(X) f(X'|X) d_p(Z_X(A), \hat{Z}) \qquad \forall \hat{Z} \tag{4.8}$$

Therefore, maximizing Formula (4.7) under constraint (4.6) is equal to optimizing Formula (4.7) under constraint (4.8).

4.5 Privacy-Utility Tradeoff

In this section, we first formalize optimal problem that can produce optimized data sanitization strategy. Then, we discuss how to solve the proposed optimal problem. Here, we introduce function $LaPri(.)$ to measure latent-data privacy with current sanitized attribute set and social links.

4.5.1 Optimal Problem Formulation

The problem of (ϵ, δ)-utility with maximize latent-data privacy can be formulated as follows.

Definition 4.5.1. (ϵ, δ)-***UtiOptPri*** $(\psi(.), d_u(.), d_p(.), \mathcal{S}, \epsilon, \delta,)$**.** *Given user's profile $\psi(.)$, attribute set disparity measurer $d_u(.)$, privacy disparity measure $d_p(.)$, structure utility value metric $\mathcal{S}$, structure utility loss threshold ϵ, and prediction utility loss threshold δ, the question is to find data-sanitization strategy $f(.)$ and link-sanitization strategy $\mathcal{L}(.)$, and latent-data privacy function LaPri(.) such that*

1. $\mathcal{L}(.)$ satisfies ϵ-structure utility loss and $f(.)$ satisfies δ-prediction utility loss;

2. For any $\mathcal{L}'(.)$ that satisfies ϵ-structure utility loss and $f'(.)$ that satisfies δ-prediction utility loss, $LaPri(\mathcal{L}'(.), f'(.), \psi(.), d_p(.)) \geq LaPri(\mathcal{L}(.), f(.), \psi(.), d_p(.))$.

The linear optimization program for an arbitrary user u_i to find the optimal data sanitization strategy is as following: choose $f(X'|X)$, $\hat{Z}$, $\forall X, X'$, in order to

$$\textbf{Maximize:} \sum_{X'} P_{X'}$$

Subject to:

$$P_{X'} \leq \sum_{X} \psi(X) f(X'|X) d_p(Z_X(A), \hat{Z}) \qquad \forall \hat{Z}$$

$$\sum_{u_j \in A_i \subseteq N_i} S_j \leq \epsilon$$

$$\sum_{X} \psi(X) \sum_{X'} f(X'|X) d_u(X, X') \leq \delta$$

$$f(X'|X) \geq 0 \qquad \forall X, X'$$

$$\sum_{X'} f(X'|X) = 1, \quad \forall X$$

4.5.2 Solve the optimal problem

We now solve the optimal problem to find attribute sanitization strategy $f(.)$ and link sanitization strategy $\mathcal{L}(.)$.

Find Link-sanitization Strategy First, we prove the link sanitization method introduced in Section 4.3.2 has monotonicity property. The monotonicity property indicates that if we increase the number of removed links of a user, we can only improve this user's latent-data privacy.

Theorem 4.5.1. ***Monotonicity.*** *Function $LaPri : A_i \to \mathbb{R}^*$ is monotonically nondecreasing, namely, $LaPri(A_i \cup e_{ij}) \leq LaPri(A_i)$, where $e_{ij} \in A_i$, $A_i \in N_i$, and A_i is the vulnerable link set of u_i.*

Proof. As discussed in Definition 4.3.1, the prediction accuracy for user u_i' SLA decreases upon removing the vulnerable link between u_i and u_j; namely, for any vulnerable link e_{ij}, $\Lambda(A_i) \leq \Lambda(A_i \cup e_{ij})$. This accuracy relation indicates that for user u_i, the latent-data privacy with vulnerable link set A_i is definitely larger than the latent-data privacy with vulnerable link set $A_i \cup e_{ij}$. Hence, $LaPri(A_i \cup e_{ij}) \leq LaPri(A_i)$. □

Theorem 4.5.2. ***Submodularity.*** *Function* $LaPri : A_i \rightarrow \mathbb{R}^*$ *is submodular, namely,* $LaPri(B_i \cup e_{ij}) - LaPri(B_i) \leq LaPri(A_i \cup e_{ij}) - LaPri(A_i)$*, where* $A_i \subseteq B_i \subseteq N_i$*,* $e_{ij} \in N_i$*, and* A_i *and* B_i *are vulnerable link sets of* u_i*.*

Proof. For the prediction accuracy for SLA, the maximum decrease in prediction accuracy of user u_i, by removing a vulnerable link e_{ij} from vulnerable link set A_i is at least more than the maximum decrease by removing e_{ij} from another set B_i, namely, $\Lambda(A_i \cup e_{ij}) - \Lambda(A_i) \leq \Lambda(B_i \cup e_{ij}) - \Lambda(B_i)$, where $A_i \subseteq B_i \subseteq N_i$, and $e \in N_i$. The accuracy relation indicates that for user u_i, the maximum gain in latent-data privacy after removing vulnerable link e_{ij} from vulnerable link set A_i is at least more than the maximum gain by removing e_{ij} from B_i. Hence, $LaPri(B_i \cup e_{ij}) - LaPri(B_i) \leq LaPri(A_i \cup e_{ij}) - LaPri(A_i)$. □

With Theorem 4.5.1 and Theorem 4.5.2, the problem of finding a link-sanitization strategy is equivalent to the minimization of submodular, nondecreasing, nonnegative function with constraints that is knapsack-like. The greedy algorithm proposed in [77] could be exploited to solve this problem with nondecreasing, submodular, nonnegative objective function constrained by structure utility loss.

Find Attribute-sanitization Strategy To find an attribute-sanitization strategy, the optimization problem can be solved by iterating over all possible $f(X'|X)$, all X and all sanitized X' to make sure the prediction accuracy loss of latent-data privacy is less than δ. Furthermore, find the optimal set of $f(X'|X)$ that produce minimum value of objective function $\sum_{X'} P_{X'}$. However, this approach is impractical since there is an infinite number of $f(X'|X)$. For example, X has three possible sanitized attribute vectors X_1, X_2 and X_3, and the probabilities that satisfy $\sum_{i=1:3} f(X_i|X) = 1$, $f(X_i|X) \geq 0, \forall X, X_i$ are infinite. To solve this problem, we discrete the probability space, *i.e.*, $[0, \ldots, 1] \rightarrow [0, 1/d, 2/d, \ldots, 1]$ to get a suboptimal solution. Furthermore, to shrink search space of X, the set of X' can be derived through substituting each attribute in the shared attributes between SLA and NSLA with a generic attribute, which ensures that adversaries cannot get specific information to increase prediction accuracy on sensitive attributes, while guarantees no significant accuracy

Table 4.2. General information about Caltech

Network property	**Value**
Number of users	769
Number of social links	16656
Number of attributes of each user	7
Number of possible attribute values for SLA	4
Number of possible attribute values for NSLA	2

reduction on data utility. Moreover, since there are different levels of generalization, we organize the generic attributes as a hierarchy.

4.6 Evaluation

4.6.1 Dataset

In our evaluation, we study a large Facebook dataset that contains all the Facebook "friendship" links among the users at California Institute of Technology at a certain time in September 2005. It also includes some demographic information like student/faculty status flag, gender and some other attributes, which are published by users on their Facebook pages. Each attribute is assigned a numeric value and user identity is ignored. For convenience, the dataset is named as *Caltech*. Some general information about Caltech are listed in Table 4.2.

4.6.2 Experimental Settings

As shown in Table 4.2, there are 7 attributes for each user. We choose attribute *student/faculty status flag* (represented by *flag*) and *gender* as SLA and NSLA, respectively. Table 4.2 shows that SLA and NSLA have 4 and 2 possible attribute values, respectively. The remaining 5 attributes are assumed to be publicly available attributes, among which 3 attributes are for SLA, 3 attributes are for NSLA, and 1 attribute is common.

We compare our data-sanitization strategy with different strategies to satisfy the (ϵ, δ)-UtiOptPri problem defined in Definition 4.5.1: 1) Attribute Removal: remove the most indicative attributes for SLA; 2) Attribute Perturbing: perturb the most indicative attributes

for SLA; 3) Link Removal: remove vulnerable links; 4) Random Link Removal: randomly remove links. We denote our data sanitization strategy as *Collective Sanitization* since it collectively harnesses different data sanitization methods.

4.6.3 Privacy-Utility Tradeoff with Different Data-Sanitization Strategies

We evaluate the effectiveness of our Collective Sanitization to realize the privacy-utility tradeoff. To make a fair comparison, we first evaluate latent-data privacy when the above five strategies satisfy the same data utility thresholds. We choose an arbitrary pair of ϵ and δ such as $\epsilon = 180$, $\delta = 0.4$, and then calculate the latent-data privacy under different strategies with increasing number of attributes and links being sanitized. As stated in Section 4.3.2, Collective Sanitization sanitizes user attributes by employing removing and perturbing collectively. The horizontal axis of Fig.4.1(a) for Collective Sanitization represents the number of the removed attributes (indicative for SLA) and the number of attributes (common indicative attributes for SLA and NSLA) being perturbed. Similarly, the horizontal axis of Fig.4.1(b) for Collective Sanitization represents the number of the removed vulnerable links (as presented in Section 4.3.2).

As shown in Fig.4.1(a), four strategies are generally effective in protecting latent-data privacy while realizing customized (ϵ, δ)-utility. With increasing number of attributes being sanitized, latent-data privacy monotonically increases as well. However, compared with the remaining three strategies, *Collective Sanitization* can realize a larger level of latent-data privacy with the same number of attributes being sanitized and same utility thresholds. Meanwhile, as expected, *Attribute Removal* is better than *Attribute Perturbing* in protecting latent-data privacy. With more and more attributes removed and perturbed, this advantage of *Attribute Removal* becomes more and more obvious. Furthermore, in protecting latent-data privacy, *Link Removal* is better than both *Attribute Removal* and *Attribute Perturbing*. To explain this observation, we find that the latent-data privacy under *Link Removal* and *Collective Sanitization* are close, indicating that removing vulnerable links contributes more effectiveness than attribute sanitization in protecting latent-data privacy.

The same observation can be found in Fig.4.1(b), where latent-data privacy monotonically increases with more and more links removed. However, compared with the remaining two strategies, *Collective Sanitization* can also achieve a larger level of latent-data privacy with the same number of links removed and same utility threshold. Meanwhile, *Link Removal* is better than *Random Link Removal* in protecting latent-data privacy. With more and more links removed, this advantage of *Link Removal* becomes more and more obvious.

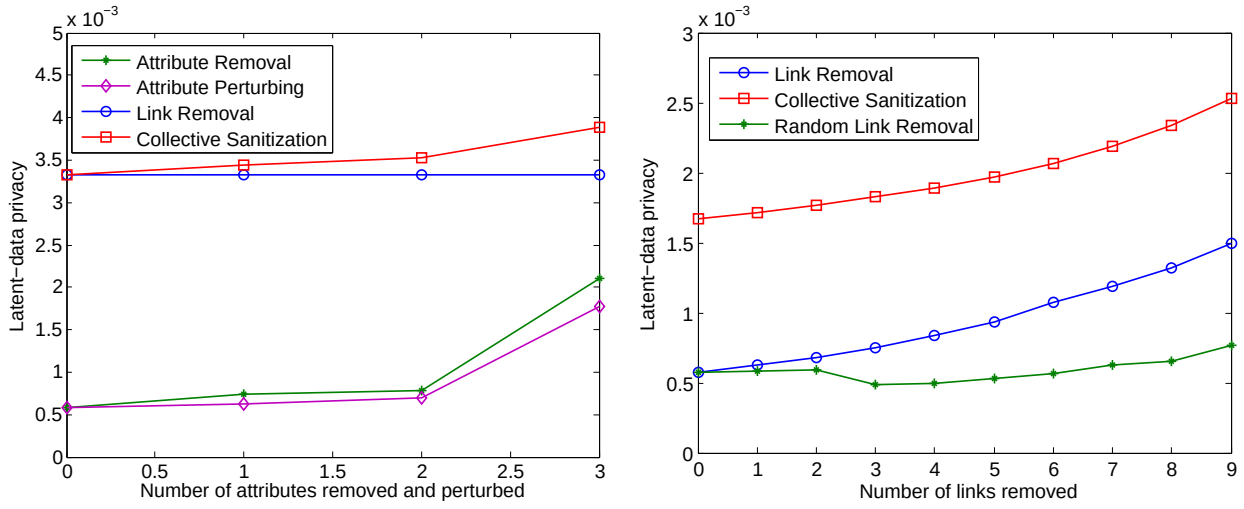
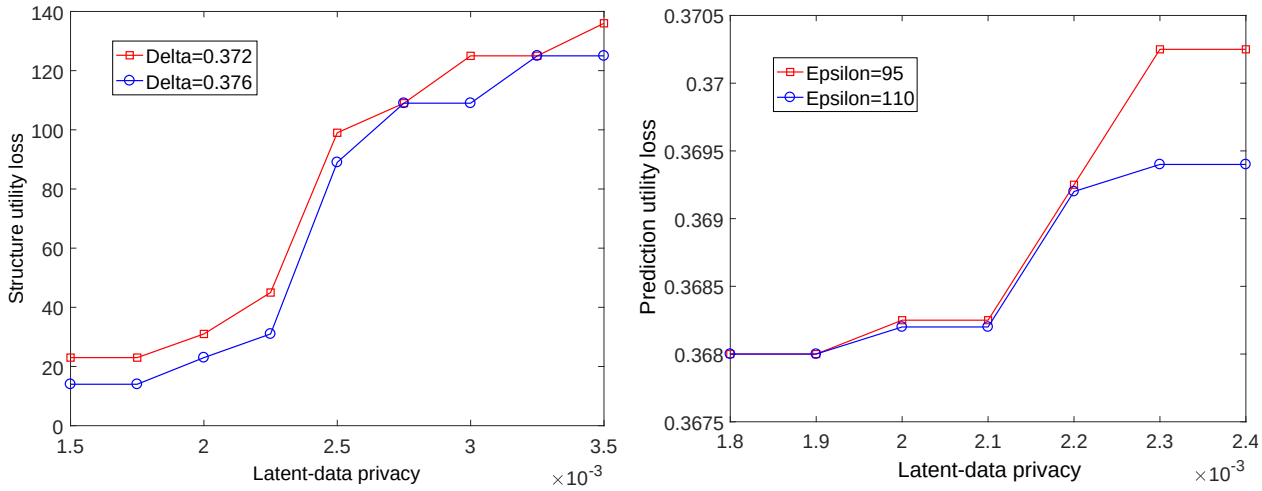


Figure 4.1. Latent-data privacy under different data-sanitization strategies with increasing number of (a) attributes; (b) sanitized links, $\epsilon = 180$, and $\delta = 0.4$.

We further discuss the effectiveness of Collective Sanitization in guaranteeing utility under different levels of latent-data privacy. The results are shown in Fig.4.2, which shows that utility loss increases with the increasing of latent-data privacy level. Furthermore, utility loss converges to a stable level with the increasing of latent-data privacy level. The reason lies that the marginal gain of latent-data privacy is obtained with the maximum number of sanitized attributes and links, and minimized utility.

4.6.4 Privacy-Utility Tradeoff with Different Prior Knowledge

We evaluate the privacy-utility tradeoff with different cases of prior knowledge for adversaries. We compare our Collective Sanitization assuming most powerful adversaries with



(a) (b)

Figure 4.2. Utility loss under different levels of latent-data privacy: (a) structure utility loss with different prediction utility loss thresholds and $\epsilon = 180$; (b) prediction utility loss with different structure utility loss thresholds and $\delta = 0.4$.

the knowledge of user profile $\psi(X)$ and data-sanitization strategy, where different types of prior knowledge are assumed: 1) Profile Only: only profile is known to adversaries; 2) Strategy Only: only data-sanitization strategy is known to adversaries; 3) Unknown Both: neither profile nor strategy is known to adversaries.

To make a fair comparison, we first compare the latent-data privacy when the above four cases has same utility thresholds. With the same utility thresholds $\epsilon = 500$ and $\delta = 0.4$, we calculate the latent-data privacy under different cases with increasing number of sanitized attributes and links. The results are shown in Fig.4.3(a) and Fig.4.3(b), where the horizontal axis for Collective Sanitization represents the number of removed/perturbed attributes and the number of removed vulnerable links, respectively.

Fig.4.3 shows that compared with different cases, Collective Sanitization assuming powerful adversaries is the most effective one in protecting latent-data privacy while guaranteeing customized (ϵ, δ)-utility. As shown in Fig.4.3(a) and Fig.4.3(b), the latent-data privacy under Profile Only and Strategy Only lies somewhere in between Collective Sanitization and Unknown Both, and profile information is more valuable than strategy information in some cases. The similar observation can be obtained in Fig.4.3(c) and Fig.4.3(d), where it is al-

so shown that latent-data privacy converges to a stable level with the increasing of utility thresholds. The reason lies that the marginal gain of latent-data privacy is obtained with the most sacrifice in utility.

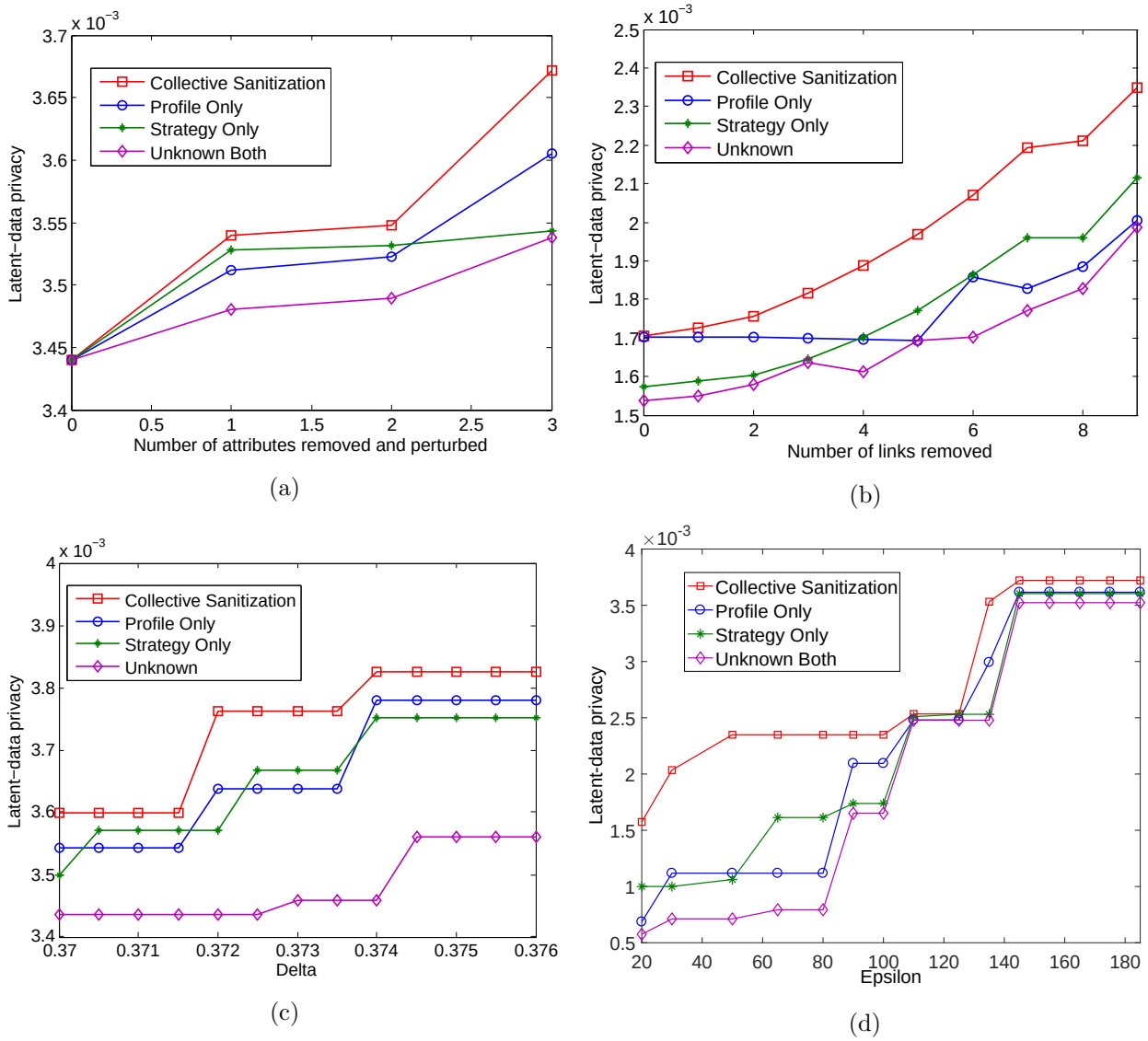


Figure 4.3. Latent privacy-utility tradeoff with different cases of prior knowledge for adversaries, with increasing number of (a) attributes; (b) sanitized links; and the increasing of (c) prediction utility threshold; (d) structure utility threshold.

Finally, the latent-data privacy with different utility thresholds is shown in Fig.4.4. Fig.4.4 shows that with the increasing of ϵ and δ, latent-data privacy increases as well. The reason lies that it is possible to determine a better data-sanitization strategy with fewer

utility requirements. Furthermore, latent-data privacy converges to a stable value with continuously increase of ϵ and δ, which indicates the optimal data-sanitization strategy is found.

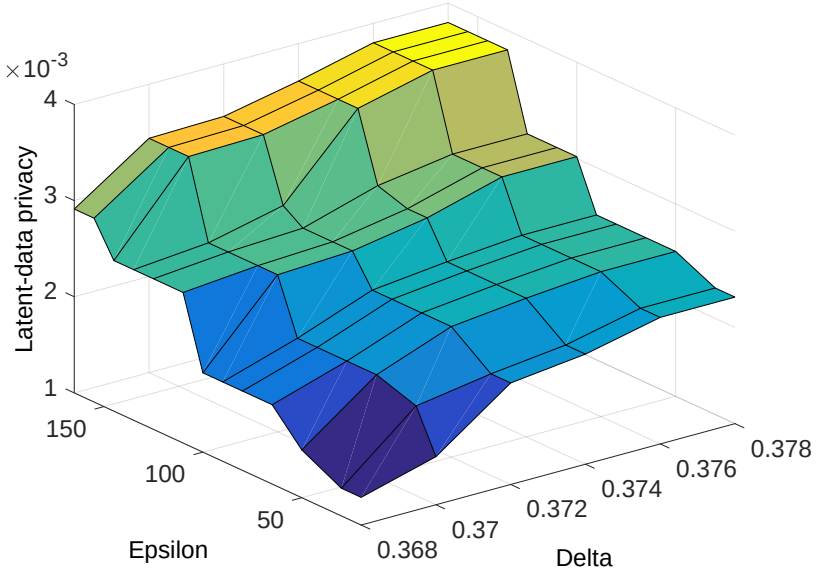


Figure 4.4. Latent-data privacy with different utility thresholds.

4.7 Conclusions

In this paper, we study how to optimize the tradeoff between latent-data privacy and customized data utility when combating against powerful adversaries with optimal inference attacks. To address this issue, we first propose two sanitization methods for links and attributes, based on which we formalize prediction utility loss matric, structure utility loss matric and latent-data privacy. Then we formulate an optimization problem that can maximize latent-data privacy while guaranteeing customized data utility. Finally, we evaluate our data-sanitization strategy towards real big social network data and the results show that the proposed data-sanitization strategy can effectively achieve a meaningful privacy-utility

tradeoff. Our future work is to explore formal privacy models, such as differential privacy or k-anonimity to balance latent-data privacy and customized data utility.

Chapter 5

PRIVACY PRESERVING GENOMIC DATA PUBLISHING

5.1 Introduction

The rapid growth of genetic techniques motivates numerous genetic-testing services, in which DNA-sequencing becomes more and more popular with decreasing cost. Consequently, an increasing number of individuals (or families) release their genomic data to genetic service providers, such as 23andMe [78], a DNA genetic testing & analysis provider; OpenSNP [79], a test result sharing platform; and PatientsLikeMe [80], a disease sharing and research platform.

Meanwhile, high availability of human genomes have accelerated genomic research in heralding the diagnosis of hereditary diseases, personalized medicine, or genetic identification. Furthermore, individuals can benefit from the research to learn about their genetic disease predispositions, genetic characteristics of ancestry, and even paternity test results, using their sequenced genome data. As a consequence, researchers expect more and more genome data could be collected to pave the way for genomic-orientated services. Individuals are also inclined to release their genome data to gain benefits from these services.

Although the released genome data bring significant benefits, they also present serious privacy threats. Single nucleotide polymorphisms (SNPs), the most important variants of DNA among human beings, can provide key information to compute the diseases susceptibility of an individual. For example, GWAS has reported that 3 particular SNPs (rs8034191, rs2808630 and rs7626795 on chromosomes 15, 1 and 3, respectively) indicate an increasing susceptibility for lung cancer. Even though genome data are generally anonymized prior to release, studies have shown that anonymization is not sufficient to preserve privacy [62] [81]. An individual may incur discrimination risks from, for example, insurance providers or employers [82].

Relatives's genomes are highly correlated. Currently, an individual can release her genome data through a simple click on a personal computer, without any consent from her relatives in advance. Consequently, once an individual is identified, even anonymized genome data would also threaten the privacy of this individual's relatives. For example, a controversy is reported regarding the publishing and sequencing of Lacks's genome data without any consent from her relatives [83]. The relatives think that their privacy is being threatened. However, some researchers think that the genetic information has been diluted because of gene mixing in the reproduction process. In this work, we intend to show that kin genomic privacy can actually be threatened. We also investigate the necessary effective tradeoff between data utility and *kin genome privacy* in order to take full advantage of genome data.

Publicly available genome association studies help with identifying sensitive information from the released genome data. For example, GWAS Catalog provides a publicly available quality controlled collection of GWAS assaying including at least 100,000 SNPs and all the SNP-trait associations [84]. An SNP-trait association indicates some SNPs (Genotypes) are associated with some human traits (Phenotypes). Once the genome data releaser is identified, an attacker can predict possible traits of this releaser and the releaser's relatives, through some data mining and machine learning techniques. As a consequence, some individuals choose to never release any genome data or only release partial genome data. However, those individuals may still face privacy threats because their relatives may choose to release genome data. Releasing partial genome data cannot completely protect against inference attacks. A pertinent example is that, James Watson, co-discover of DNA, shared his whole DNA sequence to the public, excepting Apolipoprotein E, the significant predictor of Alzheimer's disease. However, a recent article [85] shows that, although this sensitive gene is removed, it can be inferred with the publicly available statistical correlation among SNPs (*i.e.*, *linkage disequilibrium*).

In this paper, we first propose an inference attack algorithm to predict target SNPs and traits based on genome data shared by individuals and SNP-trait associations from

GWAS Catalog [84]. We then develop a data sanitization method to protect privacy by sanitizing SNPs and traits prior to releasing while guaranteeing data utility. In our inference attack algorithm, we incorporate SNPs (known and unknown), traits (known and unknown) and SNP-trait associations in a factor graph, and then apply belief propagation in this factor graph to compute the marginal probability of target unknown SNPs and traits. The previous algorithms generally incur very high computation cost which is proportional to the number of SNPs of individuals. Considering the number of human's SNPs is of tens of millions, it is a big limitation for the existing methods to obtain precise inference results. Our method achieves linear computation complexity and is more practical for inference attacks. For our data sanitization method, we first formally define the metrics to evaluate genomic privacy and data utility. We then introduce noises into SNPs prior to releasing to achieve a reasonable tradeoff between privacy and utility. Compared with the previous works, our data-sanitization method can guarantee better privacy-utility tradeoff.

5.2 Preliminaries

In this section, three fundamental concepts are briefly introduced.

5.2.1 Single Nucleotide Polymorphism

A single SNP is a DNA variation between sets of individuals of a species. Such variation means the difference in a single nucleotide (A, T, C, or G) in the genome between sets of individuals. For example, the following two DNA fragments from two individuals, CAGGTCA to CAAGTCA, have a different single nucleotide: G and A. For such a pair of nucleotides, G and A, or C and T, we say that they are alleles.

Recent studies show that SNPs carry significant information involving the susceptibility to diseases for human beings. As aforementioned, it is shown that 3 particular SNPs (rs8034191, rs2808630 and rs7626795 on the chromosomes 15, 1 and 3, respectively) indicate an increasing susceptibility for lung cancer.

Within a population, two nucleotides (*i.e.*, alleles) on a SNP locus could be distinguished

as a major allele and a minor allele. A major allele refers to the most common nucleotides of a given population. A minor allele refers to the rare nucleotides of a given population. We denote a major allele and minor allele as B and b, respectively.

In alleles, one of the nucleotides on a SNP locus is inherited from father and the other one is inherited from mother. Therefore, alleles can be denoted as BB (both nucleotides are major alleles), Bb (a major allele and a minor allele) or bb (both nucleotides are minor alleles).

5.2.2 Belief Propagation

Belief propagation is a statistical inference algorithm which passes messages in probabilistic graph models, including Bayesian networks, factor graphs and Markov random fields. It is generally used to calculate Marginal Probability Distribution (MPD) for target unknown variables, conditional on known ones. Belief propagation is generally described by the operations in factor graphs (a Bayesian network and Markov random field can be transformed to a factor graph). A factor graph is undirected, which contains two disjoint types of nodes: variable nodes and factor nodes. There is an edge between a factor node and a variable node iff this variable node is an argument of the factor node. In belief propagation, each variable node (factor node) passes messages to its neighbor factor nodes (variable nodes). The propagated message is the probability (belief) of a variable node being a value (such as 1 (0) representing the presence (absence) of a trait). Given certain initial states and boundary conditions, belief propagation is to propagate messages until the unknown variables converge to the boundary conditions.

5.2.3 GWAS Catalog

GWAS can be used to identify the SNPs associated with human's traits, by splitting individuals in a given population into case groups and control groups. GWAS has identified the SNPs associated with many human traits and diseases, including lung cancer, Chronic kidney disease, height, cervical cancer, type-2 diabetes, etc.

GWAS Catalog is a publicly available report of GWAS which presents all the SNP-trait associations. The analysis of massive variation across human genomes in case-control studies can also distinguish two nucleotides in a SNP locus as: risk allele and non-risk allele. A risk allele refers to the most common nucleotide of the individuals in a case group (*i.e.*, individuals present a target trait). Accordingly, the other nucleotide in a SNP locus is referred as a non-risk allele.

In the context of GWAS studies, another parameter reported by GWAS Catalog is *odds ratio* of a nucleotide K, which refers to the ratio of the odds of traits for individuals having K and the odds of traits for individuals who do not have K.

5.3 Problem Formulation

5.3.1 Genomic Data Model

All SNPs in the DNA sequence of an individual is denoted by S ($|S| = n$). The genotype of SNP S is denoted by s_i with $s_i \in S$ and $s_i \in \{BB, Bb, bb\}$ (as defined in Section 5.2.1). We assume target individuals or familial members release complete or partial genome data to the public, and the target sensitive part is not released for privacy concerns. The publicly available SNPs of target individual is defined as S_K, while the unknown part is defined as S_U.

In an SNP-trait association, we represent the collected traits for target family by T. For each trait $t_j \in T$, the set of associated SNPs is denoted by S_{t_j}. For each $s_i \in S_{t_j}$, the risk allele of s_i is denoted by r_i^j, and the odds ratio of r_i^j is denoted by O_i^j. GWAS also reports the risk-allele frequency (RAF) in a control group which is expressed by $f_i^{j^o}$. Although GWAS does not report the RAF in the case group (denoted by $f_i^{j^a}$), we can obtain it from $f_i^{j^o}$ and O_i^j [49]. Similarly, the released traits by familial members are defined by T_K and the unreleased ones are denoted as T_U.

5.3.2 Attacker Model

An attacker intends to predict target traits and SNPs of target individuals, *i.e.*, $X_U = T_U \cup S_U$. We assume the attacker is relatively powerful with broad background knowledge: (i) the released SNPs from the target individual and the relatives (if any) (*i.e.*, S_K), (ii) the traits shared by the individual and her relatives (if any) (*i.e.*, T_K), and (iii) SNP-trait association and auxiliary information (*i.e.*, $\mathcal{C}(T, s_i, r_i^j, O_i^j, f_i^{j^o})$).

5.3.3 Problem Definition

The studied problem can be formally defined as follows:

Input:

(1) individual released SNPs S_K, released traits T_K, and SNP/trait association $\mathcal{C}(T, s_i, r_i^j, O_i^j, f_i^{j^o})$.

(2) Privacy threshold δ.

Output:

Inference algorithm for predicting X_u.

Privacy preserving genome data releasing method achieving tradeoff between privacy and data utility.

5.4 Inference Attacks

For the above problem, the inference attacks on target SNPs and traits could be formulated as calculating their Marginal Probability Distribution (MPD).

With this objective, we first calculate the joint probability distribution of the target unknown variables, *i.e.*, $p(X_U|S_K, T_K, \mathcal{C})$, where $X_U = T_U \cup S_U$.

Then, the marginal probability distribution of a target variable $x_i \in X_U$ can be obtained:

$$p(x_i|S_K, T_K, \mathcal{C}) = \sum_{X_U \backslash x_i} p(X_U|S_K, T_K, \mathcal{C}) \tag{5.1}$$

where $X_U \backslash x_i$ is to sum over all the variables in X_U except x_i.

In Equation (5.1), the number of the terms exponentially increases with the number of the variables in X_U. Considering human's DNA sequences contain tens of million of SNPs as well as massive potential traits, it is impossible to predict target variables by computing marginal probability distribution directly. Our solution is to factorize the joint probability distribution of target variables into sets of local distributions, with each one taking a subset of variables as arguments. Conducting such a transformation is challenging since we need to identify proper local functions and their arguments from massive SNPs and traits.

To address this issue, we model known variables, unknown variables, and SNP-trait associations as a probability graph, and then apply belief propagation to calculate the marginal probability distribution of target variables. In this way, the calculation of marginal probability distribution is transformed from an exponential complexity problem into a linear complexity problem.

A factor graph is a probability graph model containing two types of nodes: variable nodes and factor nodes. A *SNP variable node* represents a known or unknown SNP, and a *trait variable node* represents a potential known or unknown trait. A *factor node* represents an SNP-trait association.

Variable nodes and factor nodes are connected in the following way: SNP variable node s_i and trait variable node t_j are connected to factor node f_{ji} if s_i is associated with trait t_j.

Fig.5.1 shows a simple example with 3 trait variables $T = \{t_1, t_2, t_3\}$ and 5 SNP variables $S = \{s_1, s_2, s_3, s_4, s_5\}$. From Fig. 5.1, we observe that $\{s_1, s_2\}$, $\{s_2, s_3, s_4\}$, $\{s_5\}$ are associated with t_1, t_2 and t_3, respectively.

By applying belief propagation in a factor graph, $p(X_U|S_K, T_K, \mathcal{C})$ could be factorized into sets of local distributions and each one takes a subset of variables (SNPs and traits) as arguments:

$$p(X_U|S_K, T_K, \mathcal{C}) = \frac{1}{Z} \prod_{i \in S} \prod_{j \in T} f_{ji}(s_i, t_j, \mathcal{C}) \tag{5.2}$$

where Z ia a normalization factor.

We now investigate the rationality of Equation (5.2). As introduced in Section 5.2.2,

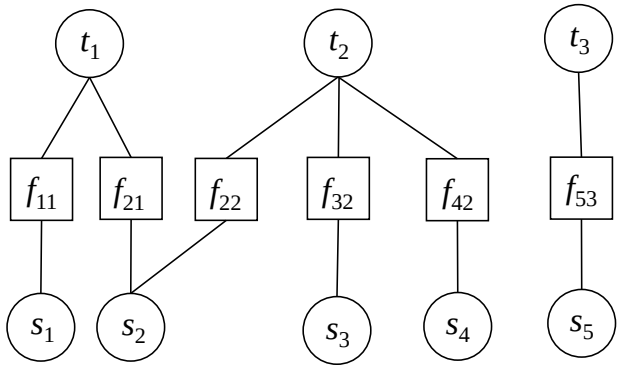


Figure 5.1. A factor graph with 3 traits $T = \{t_1, t_2, t_3\}$ and 5 SNPs $S = \{s_1, s_2, s_3, s_4, s_5\}$.

belief propagation performs inference on probability graphical models by passing messages from variable nodes and factor nodes, and from factor nodes to variable nodes. Two parameters are introduced, μ and λ. μ represents the massages from a variable node (s_i or t_j) to a factor node. λ represents the massages from a factor node to a variable node. To describe the message-passing process, we take nodes t_2 and s_1, factor node f_{21} in Fig.5.1 as an example. Massage $\mu^{(n)}_{v\to f}({s_1}^{(n)})$ from s_1 to f_{21} represents the probability of $s_1 = \kappa$ ($\kappa = BB, Bb, bb$) in the n-th iteration. Message $\lambda^{(n)}_{f\to v}({s_1}^{(n)})$ from f_{21} to s_1 represents the probability of $s_1 = \kappa$ ($\kappa = BB, Bb, bb$) in the n-th iteration, given the trait/SNP associations.

A variable node v passes massage to its neighbor factor node f by multiplying all messages passed from its neighbor factor nodes except f. Taking the factor graph in Fig.5.1 as an example, the message from s_1 to f_{21} (denoted as $s \to f$) is:

$$\mu^{(n)}_{s\to f}({s_1}^{(n)}) = \frac{1}{Z} \times \prod_{f^* \in N(s_1)\backslash f_{21}} \lambda^{(n-1)}_{f^*\to s}({s_1}^{(n-1)}) \tag{5.3}$$

where $N(s_1)\backslash f_{21}$ includes all the neighbor factor nodes of s_1 except f_{21} (in Fig.5.1, $N(s_1)\backslash f_{21} = \{f_{11}\}$).

Similarly, variable node t_2 sends message to factor node f_{21} (denoted as $t \to f$):

$$\mu_{t\to f}^{(n)}({t_2}^{(n)}) = \frac{1}{Z} \times \prod_{f^* \in N(t_2)\backslash f_{21}} \lambda_{f^*\to t}({t_2}^{(n-1)}) \tag{5.4}$$

where $N(t_2)\backslash f_{21} = \{f_{22}, f_{23}\}$.

Then, from belief propagation, factor node f sends message to neighbor variable node v by multiplying all the massages from the neighbors of f except v, and multiplies the obtained product with the factor. It then sums all the neighbor variable nodes of f except v. The massage from f_{21} to variable node s_1 (denoted as $f \to s$) is

$$\lambda_{f\to s}^{(n)}({s_1}^{(n)}) = \sum_{t_2} f_{21}(s_1, t_2) \prod_{v^* \in N(f_{21})\backslash s_1} \mu_{v^*\to f}^{(n)}(v^*) \tag{5.5}$$

where $f_{21}(s_1, t_2) \propto p(s_1|t_2)$ and we will discuss its computation in the following part.

Similarly, the message passing from f_{21} to variable node t_2 (denoted as $f \to t$) is

$$\lambda_{f\to t}^{(n)}({t_2}^{(n)}) = \sum_{s_1} f_{21}(s_1, t_2) \prod_{v^* \in N(f_{21})\backslash t_2} \mu_{v^*\to f}^{(n)}(v^*) \tag{5.6}$$

where $f_{21}(s_1, t_2) \propto p(t_2|s_1)$ and we will discuss its computation in the following part.

We now show the initial state and termination condition in a massage-passing iteration. In the initial iteration (*i.e.*, $n = 1$), variable nodes first pass massage to factor nodes. Variable node $s_i \in S_U$ begins passing massage to its neighbor factor nodes. We set $\mu_{s\to f}^{(1)}({s_i}^{(1)}) = 1$ for each potential value of s_i (*i.e.*, $\mu_{s\to f}^{(1)}({s_i}^{(1)} = BB) = 1$, $\mu_{s\to f}^{(1)}({s_i}^{(1)} = Bb) = 1$, $\mu_{s\to f}^{(1)}({s_i}^{(1)} = bb) = 1$). On the other hand, for any SNP variable node $s_i \in S_K$ with known value $s_i = \kappa$, we set $\mu_{s\to f}^{(1)}({s_i}^{(1)} = \kappa) = 1$ and $\mu_{s\to f}^{(1)}({x_j^i}^{(1)} = \kappa') = 0$ for other potential SNP values κ', $\kappa' \in \{\{BB, Bb, bb\}\backslash\kappa\}$.

The massages passing from $t_j \in T_U$ to its neighbor factor nodes follows the same rule. Until all the massages are converged (*i.e.*, the values of μ and λ never change), the iteration is finished. Finally, the MPD of each unknown variable $x_i \in X_U$ is obtained by multiplying

Table 5.1. Conditional probability of risk allele r_i^j and non-risk allele ρ_i^j, given one of neighbor factor nodes t_j of s_i

	t_j	$\bar{t_j}$
r_i^j	$f_i^{j^a}$	$f_i^{j^o}$
ρ_i^j	$1 - f_i^{j^a}$	$1 - f_i^{j^o}$

Table 5.2. Genotype probability of $r_i^j r_i^j$, $r_i^j \rho_i^j$ and $\rho_i^j \rho_i^j$, given one of s_i' neighbor factor nodes t_j

	t_j	$\bar{t_j}$
$r_i^j r_i^j$	$\sqrt{f_i^{j^a}}$	$\sqrt{f_i^{j^o}}$
$r_i^j \rho_i^j$	$f_i^{j^a}(1 - f_i^{j^a})$	$f_i^{j^a}(1 - f_i^{j^o})$
$\rho_i^j \rho_i^j$	$\sqrt{1 - f_i^{j^a}}$	$\sqrt{1 - f_i^{j^o}}$

all the massages passed to x_i.

As indicated in Equations (5.5) and (5.6), to formulate the message content in each iteration, we need to calculate the conditional probability of traits and SNPs. Firstly, the prevalence rate of each trait $p(t_j)$ can be viewed as prior knowledge, which can be collected from public organizations such as CDC [86]. Then, since it is non-trivial to deduce the probability of SNP s_i conditioned on an associated trait, we calculate the conditional probability of the nucleotide of a SNP locus. As introduced in Section 5.2.3, two nucleotides are distinguished in a SNP locus as: risk allele and non-risk allele. Table 5.1 shows the probability of RAF and nRAF conditioned on an associated trait, respectively.

Based on the conditional probability of RAF and nRAF, we go back to calculate the probability of SNP conditioned on an associated trait. Given allele r_i^j and allele ρ_i^j, the genotype of s_i associated by trait t_j can be one of the following: $r_i^j r_i^j$, $r_i^j \rho_i^j$ and $\rho_i^j \rho_i^j$. Therefore, the genotype frequency can be easily obtained by simply transforming Table 5.1. The resultant table is shown in Table 5.2. Similarly, the trait probability conditioned on an SNP to which it associates can be easily deduced from Table 5.2 based on Bayesian posterior probability.

5.5 Tradeoff between Privacy and Utility

In this section, we present a data-sanitization method to achieve a reasonable tradeoff between privacy and utility by introducing noises to SNPs prior to releasing. The data sanitization method is expected to effectively defense against inference attacks on target sensitive SNPs and traits, as well as to guarantee data utility. With this goal, we first define the metrics for evaluating privacy and utility loss with noises introduced into SNPs.

5.5.1 Metrics for Privacy and Utility

If the sanitized data can prevent attackers to learn sensitive information, generally, such sanitization can effectively protect privacy. We define privacy in terms of the ambiguity level of inference results. Specifically, we expect that the larger uncertainty of an attacker, the higher the privacy preservation level.

We use the entropy of $p(x_i|S_K, T_K, \mathcal{C})$ to evaluate the uncertainty of inference results from an attacker:

$$H_i = \frac{-\sum_{x_i} p(x_i|S_K, T_K, \mathcal{C}) \log p(x_i|S_K, T_K, \mathcal{C})}{\log(3)} \tag{5.7}$$

where x_i is either a target SNP ($x_i \in \{BB, Bb, bb\}$) or a trait ($x_i \in \{0, 1\}$). The larger the entropy, the larger the ambiguity of $p(x_i|S_K, T_K, \mathcal{C})$. Then, parameter δ is introduced to bound H_i as a privacy metric:

Definition 5.5.1. ***δ-privacy.*** *The released SNPs satisfy δ-privacy if $H_i \geq \delta$ for each SNP s_i.*

For data utility, it is expected that as many actual SNPs as possible are released, while guaranteeing δ-privacy.

Definition 5.5.2. ***Utility.*** *The utility of a set of SNPs is measured by the expected number of released SNPs.*

5.5.2 Data-Sanitization Method

To defense against inference attacks on x_i, we propose to sanitize the neighbor SNPs of x_i. The *neighbor SNPs* of a trait and an SNP are defined as follows:

Definition 5.5.3. ***Neighbor SNPs of a trait.*** *The neighbor SNPs of trait t_j are those SNPs which:*

1. *are directly associated with t_j.*
2. *are associated with the traits sharing common SNPs with t_j.*
3. *share common traits with the SNP in Case 2.*

For example, all three SNPs s_1, s_2 and s_3 are neighbor SNPs of t_1 in Fig.5.1, as s_2 and s_3 are associated with t_2 that shares s_1 with t_1 (satisfying Case 2). Furthermore, if s_3 and another SNP s_4 are associated with another trait t_3, s_4 is also the neighbor SNP of t_1 (satisfying Case 3).

Similarly, the neighbor SNPs of an SNP is defined as follows:

Definition 5.5.4. ***Neighbor SNPs of an SNP.*** *The neighbor SNPs of SNP s_i are those SNPs which:*

1. *are associated with a same trait with s_i.*
2. *are associated with the traits associated with the SNPs in Case 1.*
3. *share common traits with the SNP in Case 2.*

For example, s_2 and s_3 are neighbor SNPs of s_1, as they are associated with same trait t_2 with s_1. Furthermore, if s_3 and another SNP s_4 are associated with another trait t_3, s_4 is also the neighbor SNP of s_1 (satisfying Case 2).

To achieve the tradeoff between privacy and utility, we expect the set of neighbor SNPs of each x_i can be identified so that sanitizing them can maximize data utility while satisfying

the privacy preservation constraint. For this purpose, the concept of *vulnerable neighbor SNP* is introduced:

Definition 5.5.5. ***Vulnerable neighbor SNP.*** *The vulnerable neighbor SNP of x_i is a neighbor SNP of x_i, whose sanitizing will decrease the prediction accuracy on x_i.*

Since sanitized released SNPs through the perturbing method (*i.e.*, replace an actual SNP with another one) may generate uncontrollable results when making genetic analysis, we choose to sanitize SNPs through the *removing* method. The privacy of x_i upon removing its vulnerable neighbor SNP x_k is $H_i(N_i - x_k)$, where N_i is the neighbor SNPs of x_i.

With Definition 5.5.5, the problem of achieving Genome Privacy-Utility Tradeoff (G-PUT) can be formally stated as follows:

Definition 5.5.6. ***GPUT($S_K, T_K, S_U, T_U, \mathcal{C}, \delta$).*** *Given known SNPs X_K, known traits T_K, statistical information from GWAS Catalog $\mathcal{C}$, and privacy threshold δ, how to identify the minimum number of SNPs to sanitize so that the sanitized SNPs guarantee each trait in T_U and each SNP in S_K satisfy δ-privacy.*

To solve the problem, we first prove the ambiguity of inference results, *i.e.*, Equation (5.7) has the *monotonicity* and *submodularity* properties, when the increasing number of SNPs are sanitized. The monotonicity property means that if we sanitize more SNPs, we can only improve privacy.

Theorem 5.5.1. ***Monotonicity.*** *The privacy function of an arbitrary variable $x_i \in X_U$, $H_i : N_i \to \mathbb{R}^*$ is monotonically nondecreasing,* i.e., *$H_i(N_i \cup s_k) \leq H_i(N_i)$, where $s_k \in N_i$ and N_i is the set of vulnerable neighbor SNPs of x_i.*

Proof: As mentioned in Definition 5.5.5, the prediction accuracy on x_i decreases upon sanitizing vulnerable neighbor SNPs, which implies that $\Lambda(N_i) \leq \Lambda(N_i \cup s_k)$ for any vulnerable neighbor SNP x_k. Obviously, the above inequation indicates that for any x_i, the privacy with neighbor SNPs N_i is definitely larger than the privacy with neighbor SNPs $N_i \cup x_k$, namely, $H_i(N_i \cup s_k) \leq H_i(N_i)$.

Theorem 5.5.2. ***Submodularity.*** *The privacy function of an arbitrary variable $x_i \in X_U$, $H_i : N_i \to \mathbb{R}^*$ has the submodularity property,* i.e., *$H_i(U_i \cup s_k) - H_i(U_i) \leq H_i(V_i \cup s_k) - H_i(V_i)$, where $U_i \subseteq V_i \subseteq N_i$, $s_k \in N_i$, and U_i and V_i are the sets of vulnerable neighbor SNPs of x_i.*

Proof For an arbitrary variable $x_i \in X_U$, the maximum decrease in prediction accuracy on x_i, by sanitizing a vulnerable neighbor SNP s_k from vulnerable neighbor SNPs V_i is at least more than the maximum decrease by removing s_k from another set U_i, namely, $\Lambda(V_i \cup s_k) - \Lambda(V_i) \leq \Lambda(U_i \cup s_k) - \Lambda(U_i)$, where $V_i \subseteq U_i \subseteq N_i$, and $s_k \in N_i$. The accuracy relation indicates that for x_i, the maximum gain in privacy after removing vulnerable neighbor SNP s_k from vulnerable neighbor SNPs V_i is at least more than the maximum gain by removing s_k from U_i. Hence, $H_i(U_i \cup s_k) - H_i(U_i) \leq H_i(V_i \cup s_k) - H_i(V_i)$.

Theorem 5.5.1 and Theorem 5.5.2 show that the problem of finding an SNP sanitization method is transformed to the minimization of submodular, nondecreasing, nonnegative function with constraints that is knapsack-like. Then, we can utilize the greedy algorithm proposed in [77] to solve this problem.

5.6 Evaluation

5.6.1 Datasets

In our evaluation, we construct a factor graph relying on the trait/SNP associations provided by GWAS Catalog, as discussed in Section 5.2.3 and Section 5.4. Then, we evaluate our inference method on trait and SNPs and the data-sanitization method towards the Age-related macular degeneration (AMD) dataset. AMD is a degeneration of eye's macula, which generally leads to vision loss for elder people. As a chronic disease, AMD is caused by a combination of genetic defect and environmental factors. The AMD dataset contains genotypes of 90449 SNPs from 96 cases and 50 controls [**?**].

Table 5.3. Seven other popular diseases and the corresponding prevalence rates

Disease	Prevalence rate
Alzheimer's Disease	0.0167
Celiac Disease	0.0075
Heart Diseases	0.115
Hypertensive disease	0.29
Liver carcinoma	0.000017
Osteoporosis	0.103
Stomach Carcinoma	0.00025

5.6.2 Experiment Setting

Since the AMD dataset only contains the case/control groups involving the AMD disease, for our evaluation, we choose 7 other popular diseases and assume each individual has each disease with disease prevalence rate. The chosen diseases and the corresponding prevalence rates are shown in Table 5.3.

By searching from GWAS Catalog, the corresponding associated SNPs and parameters can be identified for each disease. Then, the factor graph involving these diseases and associated SNPs can be constructed.

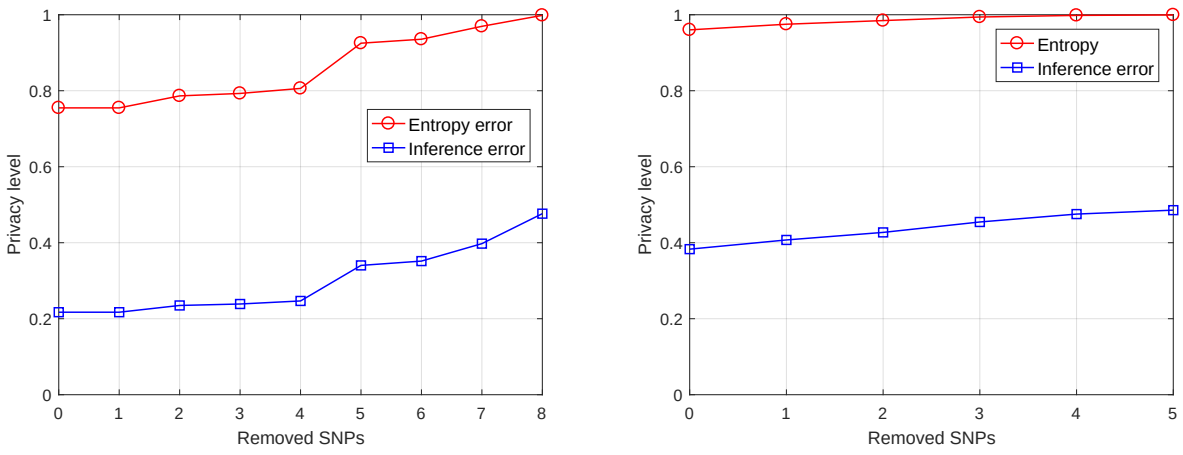
As a comparison, we introduce the estimation error of an attacker for target traits and SNPs as another privacy metric, and the estimation error of an attacker in predicting x_i is defined as:

$$Er = \sum_{x_i} p(x_i|S_K, T_K, \mathcal{C})\|x_i - \hat{x_i}\| \tag{5.8}$$

where $\hat{x_i}$ is the predicted result by an attacker.

5.6.3 Experiment Results

We show the evolution of trait privacy level with the increasing number of sanitized SNPs in Fig.5.2. It shows that our prediction method has a better accuracy performance. When no SNPs are removed, our inference method has larger entropy (less attacker uncertainty) and lower estimation error compared with that of naive Bayes. Furthermore, to maximize attacker uncertainty (*i.e.*, entropy error value approximates to 1), our inference method requires removing more SNPs.



(a) (b)

Figure 5.2. Privacy level with increasing number of sanitized SNPs: (a) belief propagation; (b) Naive Bayes, as a prediction method.

5.7 Conclusions

In this paper, we propose an inference attack algorithm which can predict the genotypes and traits of individuals with linear computation complexity, based on publicly available genome data and traits released by individuals or their relatives. We also propose an SNP-sanitization method to achieve the tradeoff between genomic data privacy and utility, by introducing noises to genome data to be released. The proposed reconstruction method can efficiently launch inference attacks for high-dimensional genomic data, relying on factor graphs and belief propagation. To develop such a method, we first introduce the metrics to evaluate utility and privacy based on data availability and attacker uncertainty. With the defined metrics, proper SNPs can be sanitized to satisfy the privacy protection budget with less utility loss.

Chapter 6

FUTURE RESEARCH DIRECTIONS

6.1 Privacy-Preserving Data Collection and Processing for the Internet of Things

The privacy challenges raised by IoT are critical to address as they have implications on basic rights and our collective ability to trust the Internet and the devices that connect to it. Generally, privacy concerns are amplified by the way in which the IoT expands the feasibility and reach of surveillance and tracking.

The Internet of Things (IoT) is becoming more and more widespread, which has led to increasing volume of sensory data. As estimated by the IDC, by 2020, more than 212 billion sensors will be connected worldwide and 44 zettabytes of data will be generated. With the development of big data techniques, certainly, clients daily life will also benefit from such incomprehensible sensory data. Therefore, emerging data techniques are expected to extract valuable information from such big multi-modal sensory data.

However, the emerging privacy scandals reminder clients must demand better privacy and security preservations that protect them against data breaches, inference attacks, corporate surveillance, etc. Therefore, how to conduct privacy preserving data collection and processing for the IoT with significant data utility, is becoming increasingly stringent.

Although several methodologies have been developed for addressing this issue; however, three key challenges are still demanding new techniques: 1) Analyzing how the potential strategies taken by the IoT server and the clients to guarantee better tradeoff between IoT privacy and data utility 2) Theory and practice of designing privacy proxy to outsource the management of privacy preference expressing, regulating and enforcing. 3) Privacy preserving aggregation on big IoT data.

The first challenge mainly derives from the high dimensional property of IoT data, the

complex data correlation, and the potential auxiliary knowledge of attacker. In collecting high-dimensional data with privacy guarantee, large scale of noise is generally required to be injected, because of output salability and signal-to-noise ratio [1]. Furthermore, the complex data correlation and auxiliary knowledge among multi-modal data presents significant challenges in protecting against inference attacks. To address issue, I propose to incorporate such high-dimensional data, correlation and auxiliary information into a probabilistic graphical model (such as factor graph), and then approximate the high-dimensional distribution of the IoT data with a set of well-chosen low-dimensional distributions; then, noise for specific privacy guarantee can be injected into them.

The second challenge mainly derived from large number of devices in modern IoT. To rescue clients from such a heavy burden of expressing, regulating and enforcing privacy preferences, I propose to develop a privacy proxy based on game theory. Since such a proxy is honest-and-curious so that we propose to identify a privacy proxy from the game playing among attacker, clients and proxy.

The third challenge mainly derived from the big data property of IoT data. How to effectively obtain complex aggregation results with specific privacy guarantee from big IoT data is challenging, such as range counting, quantiles, etc. For example, to guarantee the differential privacy (viewed as the formal privacy definition), the sensitivity of aggregation functions is generally very high in IoT. To address this issues, we propose to lower the sensitivity of functions with sampling and data combinations.

It is my belief and a key motivation for the future research interests that, to properly protect privacy in an IoT application, one must make available two different toolsets:

6.1.1 Toolset 1: Enable Users to Express, Regulate and Enforce Their Privacy Preferences

For a client, there must be a toolset for him/her to properly *express*, *regulate*, and *enforce* their privacy preferences involving a large number of IoT devices. It is important for the client to determine how much he/she values his/her private data and, in turn, whether

the benefit of an IoT application outweighs the sacrifice on privacy. Given many clients' lack of expertise on understanding the implications of disclosing private information, it is imperative to have a toolset that helps a client with the proper valuation of private data and determining whether to share it to enable an IoT application. Moreover, since privacy preferences are generally evolving dynamically, it is imperative to have a toolset that helps clients to regulate their preferences autonomously. Most importantly, such toolset can help clients to put their preferences into effect.

6.1.2 Toolset 2: Understand the Tradeoff between Service Quality and Privacy guarantees

For a server, there must be a toolset for it to understand the tradeoff between service quality and privacy guarantees. After all, it is the job of the server to return to clients the benefit of an IoT application, so as to justify the collection of private information. A reputable company may be willing to provide a consumer-friendly privacy policy - but this cannot come at a significant expense of the service quality offered by the IoT application. Thus, there must be a toolset available for the server to properly evaluate the tradeoff between privacy guarantees and the resulting loss of quality of service; and to devise optimal strategies that preserve service quality given potentially wide ranges of privacy preferences of different clients.

6.1.3 Interesting Problems

Here are some examples of problems I find interesting:

- Theory and practice of designing privacy proxy to outsource the management of privacy preference expressing, regulating and enforcing.
- Big data mining methodology over multi-modal IoT data to reconstruct detailed profiles of clients.

- Analyzing how the potential strategies taken by the server and the clients play out with each other - specifically, the implications of such strategies on the effectiveness of IoT applications and the client privacy.
- Dynamic and distributed IoT data publishing with privacy and utility guarantee.
- Privacy preserving IoT data mining.

6.2 Differentially Private Algorithms for Big Data Aggregation

The proliferation and ever-increasing capabilities of mobile devices such as smart phones give rise to a variety of mobile sensing applications, and also produce a large amount of sensory data. How to effectively extract useful information from such mass data, such as performing business analysis, identifying frequent patterns, releasing data statistics, etc, is becoming more and more valuable and imperative, with sensory data being collected, analyzed, and disseminated in a massive scale.

Although aggregation statistics computed from sensory data are very useful, in many scenarios, the data from users are privacy-sensitive, and users do not trust any single third-party aggregator to see their data values. Fortunately, differential privacy, the state-of-the-art paradigm can be used to address the balance between data utility and privacy in data aggregation, which requires that the data released reveals little information about whether any particular individual is present or absent from the data. However, the most significant challenges is, to implement differential privacy over big data (general high-dimensional), the amount of noise injected in data has to be very high.

Therefore, it is my belief and a key motivation for the future research interests that, to properly protect privacy in big data aggregation, one must address the following issues first:

- Differentially private algorithms for constructing data aggregation over high-dimensional domain.
- Approaches to reduce the sensitivity in data aggregation.

- Privacy preserving deep learning method to assist data aggregation.

6.3 Privacy Preserving Genomic Data Publishing

A key challenge for developing privacy preserving genomic data publishing is how to deal with high computational complexity brought by the massive genomes and human traits, as well as complex association in human's genetic information. As shown by the dbSNP database, the largest public SNP repository [87], includes over 50 million human SNPs, encoding the most common type of genetic variation among people. Meanwhile, statistical data form NIH in 2010 shows that there are more than 6,000 genetic disorders known. Genome-Wide Association Study of different type of human traits (i.e., case-control study) can be publicly accessed in dbGaP [88], which offers the genetic information of case group population and control group population. For example, we can collect Age-related macular degeneration (AMD) dataset from dbGaP. AMD is a degeneration of eye's macula, which generally leads to vision loss for elder people. As a chronic disease, AMD is caused by a combination of genetic defect and environmental factors. The AMD dataset reported genotypes of 90449 SNPs from 96 cases and 50 controls. For such massive SNPs and traits, SNP-trait association shows the susceptibility of an individual to several diseases can be computed from his SNPs. The GWAS Catalog published a vast amount of data, encompassing over 38,000 SNP-trait associations from more than 2,800 publications as of May 2017 [89]. Moreover, DNA sequences are highly correlated, leading to interdependent privacy risks. Linkage disequilibrium is a correlation that appears between any pair of SNP positions in the whole genome due to the population's genetic history. Such genetic information can be accessed in [87]. For example, the CEPH/Utah Pedigree 1463 that contains the partial DNA sequences of 17 family members. Therefore, protecting individual privacy with privacy guarantee is challenging.

REFERENCES

[1] S. Nilizadeh, 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*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 537–548.

[2] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, ser. SP '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 173–187.

[3] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 181–190.

[4] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," *SIGKDD Explor. Newsl.*, vol. 10, no. 2, pp. 12–22, Dec. 2008.

[5] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: Inferring user profiles in online social networks," ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 251–260.

[6] E. Ryu, Y. Rong, J. Li, and A. Machanavajjhala, "Curso: Protect yourself from curse of attribute inference: A social network privacy-analyzer," in *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*, ser. DBSocial '13. New York, NY, USA: ACM, 2013, pp. 13–18.

[7] J. He, W. W. Chu, and Z. V. Liu, "Inferring privacy information from social networks,"

in *Proceedings of the 4th IEEE International Conference on Intelligence and Security Informatics*, ser. ISI'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 154–165.

[8] N. Z. Gong, 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 Trans. Intell. Syst. Technol.*, vol. 5, no. 2, pp. 27:1–27:20, Apr. 2014.

[9] Y. Dong, Y. Yang, J. Tang, Y. Yang, and N. V. Chawla, "Inferring user demographics and social strategies in mobile social networks," ser. KDD '14. New York, NY, USA: ACM, 2014, pp. 15–24.

[10] S. Volkova, Y. Bachrach, M. Armstrong, and V. Sharma, "Inferring latent user properties from texts published in social media." in *AAAI*, 2015, pp. 4296–4297.

[11] R. Heatherly, M. Kantarcioglu, and B. M. Thuraisingham, "Preventing private information inference attacks on social networks," *IEEE Trans. on Knowl. and Data Eng.*, vol. 25, no. 8, pp. 1849–1862, Aug. 2013.

[12] Z. He, Z. Cai, and Y. Li, "Customized privacy preserving for classification based applications," in *Proceedings of the 1st ACM Workshop on Privacy-Aware Mobile Computing*, ser. PAMCO '16. New York, NY, USA: ACM, 2016, pp. 37–42.

[13] J. K. Jonghyuk Song, Jonghyuk Song, "Inference attack on browsing history of twitter users using public click analytics and twitter metadata," *IEEE Transactions on Dependable and Secure Computing*, 2014.

[14] Z. Jorgensen, T. Yu, and G. Cormode, "Publishing attributed social graphs with formal privacy guarantees," ser. SIGMOD '16, 2016, pp. 107–122.

[15] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: Private data release via bayesian networks," ser. SIGMOD '14, 2014, pp. 1423–1434.

[16] C. Liu and P. Mittal, "Linkmirage: Enabling privacy-preserving analytics on social relationships," ser. NDSS '16, 2016.

[17] W.-Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," ser. SIGMOD '16, 2016, pp. 123–138.

[18] P. Gundecha, G. Barbier, J. Tang, and H. Liu, "User vulnerability and its reduction on a social networking site," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 2, pp. 12:1–12:25, Sep. 2014.

[19] M. Han, M. Yan, Z. Cai, and Y. Li, "An exploration of broader influence maximization in timeliness networks with opportunistic selection," *Journal of Network and Computer Applications*, vol. 63, pp. 39 – 49, 2016.

[20] Z. He, Y. Li, J. Li, J. Yu, H. Gao, and J. Wang, "Addressing the threats of inference attacks on traits and genotypes from individual genomic data," in *International Symposium on Bioinformatics Research and Applications*. Springer, 2017, pp. 223–233.

[21] X. Zheng, Z. Cai, J. Li, and H. Gao, "A study on application-aware scheduling in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1787–1801, July 2017.

[22] X. Zheng and Z. Cai, "Real-time big data delivery in wireless networks: A case study on video delivery," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2017.

[23] X. Zheng, Z. Cai, J. L, and H. G, "Location-privacy-aware review publication mechanism for local business service systems," *In The 36th Annual IEEE International Conference on Computer Communications (INFOCOM).*, 2017.

[24] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: Privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1868–1878, 2017.

[25] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart iot systems: A consideration from privacy perspective," *IEEE Communications Magazine*, 2018.

[26] J. Lu, Z. Cai, X. Wang, L. Zhang, P. Li, and Z. He, "User social activity-based routing for cognitive radio networks," *Personal and Ubiquitous Computing*, pp. 1–17, 2018.

[27] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, 2017.

[28] L. Zhang, X. Wang, J. Lu, P. Li, and Z. Cai, "An efficient privacy preserving data aggregation approach for mobile sensing," *Security and Communication Networks*, vol. 9, no. 16, pp. 3844–3853, 2016.

[29] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 665–673, 2018.

[30] Y. Huang, Z. Cai, and A. G. Bourgeois, "Location privacy protection with accurate service," *Journal of Network and Computer Applications*, vol. 103, p. 146C156, 2018.

[31] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, March 2017.

[32] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*, April 2016, pp. 1–9.

[33] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *The 35th IEEE International Conference on Distributed Computing Systems (ICDCS 2015)*, June 2015, pp. 205–214.

[34] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.

[35] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices." *IEEE Network Magazine*, 2018.

[36] J. Li, S. Cheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Approximate holistic aggregation in wireless sensor networks." *ACM Transactions on Sensor Networks*, vol. 13, no. 2, pp. 1–11, 2017.

[37] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Comput. Netw.*

[38] J. Wang, Z. Cai, Y. Li, D. Yang, J. Li, and H. Gao, "Protecting query privacy with differentially private k-anonymity in location-based services," *Personal and Ubiquitous Computing*.

[39] K. Zhang, Q. Han, Z. Cai, and G. Yin, "Rippas: A ring-based privacy-preserving aggregation scheme in wireless sensor networks," in *Sensors*, 2017.

[40] M. Han, J. Li, Z. Cai, and Q. Han, "Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks," in *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on*. IEEE, 2016, pp. 284–292.

[41] J. Wang, Z. Cai, C. Ai, D. Yang, H. Gao, and X. Cheng, "Differentially private k-anonymity: Achieving query privacy in location-based services," in *Identification, Information and Knowledge in the Internet of Things (IIKI), 2016 International Conference on*. IEEE, 2016, pp. 475–480.

[42] "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," *Computer Networks*, vol. 135, pp. 32 – 43, 2018.

[43] Z. He, Z. Cai, Q. Han, W. Tong, L. Sun, and Y. Li, "An energy efficient privacy-

preserving content sharing scheme in mobile social networks," *Personal Ubiquitous Comput.*, vol. 20, no. 5, Oct. 2016.

[44] Z. He, Z. Cai, Y. Sun, Y. Li, and X. Cheng, "Customized privacy preserving for inherent data and latent data," *Personal Ubiquitous Comput.*, vol. 21, no. 1, Feb. 2017.

[45] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[46] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Springer Berlin Heidelberg, 2006, vol. 4052, pp. 1–12.

[47] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[48] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, Mar. 2007.

[49] Y. Wang, X. Wu, and X. Shi, "Using aggregate human genome data for individual identification," in *2013 IEEE International Conference on Bioinformatics and Biomedicine*, Dec 2013, pp. 410–415.

[50] L. Zhang, Q. Pan, X. Wu, and X. Shi, "Building bayesian networks from gwas statistics based on independence of causal influence," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec 2016, pp. 529–532.

[51] X. Guo, J. Zhang, Z. Cai, D.-Z. Du, and Y. Pan, "Dam: A bayesian method for detecting genome-wide associations on multiple diseases," in *International Symposium on Bioinformatics Research and Applications*. Springer, 2015, pp. 96–107.

[52] M. Fishelson and D. Geiger, "Exact genetic linkage computations for general pedigrees," *Bioinformatics*, vol. 18, p. S189, 2002.

[53] R. Mourad, C. Sinoquet, and P. Leray, "Probabilistic graphical models for genetic association studies," *Briefings in bioinformatics*, vol. 13, no. 1, pp. 20–33, 2011.

[54] M. Humbert, E. Ayday, J.-P. Hubaux, and A. Telenti, "Addressing the concerns of the lacks family: quantification of kin genomic privacy," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 1141–1152.

[55] J. O'Connell, K. Sharp, N. Shrine, L. Wain, I. Hall, M. Tobin, J.-F. Zagury, O. Delaneau, and J. Marchini, "Haplotype estimation for biobank-scale data sets," Nature Publishing Group, Tech. Rep., 2016.

[56] B. Howie, C. Fuchsberger, M. Stephens, J. Marchini, and G. R. Abecasis, "Fast and accurate genotype imputation in genome-wide association studies through pre-phasing," *Nature genetics*, vol. 44, no. 8, pp. 955–959, 2012.

[57] J. Marchini and B. Howie, "Genotype imputation for genome-wide association studies," *Nature Reviews Genetics*, vol. 11, no. 7, pp. 499–511, 2010.

[58] L. Zhang, Z. Cai, and X. Wang, "Fakemask: A novel privacy preserving approach for smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[59] C. Dwork, *Differential Privacy*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.

[60] A. Johnson and V. Shmatikov, "Privacy-preserving data exploration in genome-wide association studies," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 1079–1087.

[61] M. Humbert, K. Huguenin, J. Hugonot, E. Ayday, and J.-P. Hubaux, "De-anonymizing genomic databases using phenotypic traits," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 99–114, 2015.

[62] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.

[63] Y. Erlich and A. Narayanan, "Routes for breaching and protecting genetic privacy," *Nature Reviews Genetics*, vol. 15, no. 6, pp. 409–421, 2014.

[64] M. Humbert, E. Ayday, J.-P. Hubaux, and A. Telenti, "Reconciling utility with privacy in genomics," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, ser. WPES '14. ACM, pp. 11–20.

[65] https://www.researchgate.net/.

[66] http://www.imdb.com/.

[67] "Facebook beacon," 2007.

[68] K. Heussner, "'gaydar' on facebook: Can your friends reveal sexual orientation?" *ABC News.*, 2009.

[69] C. Johnson, "Gaydar," *The Boston Blobe.*, 2009.

[70] http://www.pewinternet.org/2013/05/21/teens-social-media-and-privacy/.

[71] Z. Pawlak, "Rough set theory and its applications to data analysis," *Cybernetics and Systems*, vol. 29, no. 7, pp. 661–688, 1998.

[72] S. A. Macskassy and F. Provost, "Classification in networked data: A toolkit and a univariate case study," *J. Mach. Learn. Res.*, vol. 8, pp. 935–983, May 2007.

[73] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[74] D. Jensen, J. Neville, and B. Gallagher, "Why collective inference improves relational classification," ser. KDD '04. ACM, pp. 593–598.

[75] http://www.cbc.ca/1.3154617.

[76] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[77] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Research Letters*, vol. 32, no. 1, pp. 41 – 43, 2004.

[78] https://www.23andme.com/.

[79] https://opensnp.org/.

[80] https://www.patientslikeme.com/.

[81] L. Sweeney, A. Abu, and J. Winn, "Identifying participants in the personal genome project by name (A re-identification experiment)," *CoRR*, vol. abs/1304.7605, 2013.

[82] E. Ayday, E. D. Cristofaro, J. Hubaux, and G. Tsudik, "The chills and thrills of whole genome sequencing," *CoRR*, vol. abs/1306.1264, 2013.

[83] https://www.nature.com/news/hela-publication-brews-bioethical-storm-1.12689.

[84] "The nhgri-ebi catalog of published genome-wide association studies," https://www.ebi.ac.uk/gwas/docs/about.

[85] Y. C.-E. V. P. M. Nyholt, D. R., "On jim watsons apoe status: genetic information is hard to hide," *European Journal of Human Genetics*, vol. 17(2), no. 3, pp. 147–149, 209.

[86] https://www.cdc.gov/nchs/fastats/hypertension.htm.

[87] https://www.ncbi.nlm.nih.gov/projects/SNP/.

[88] https://www.ncbi.nlm.nih.gov/gap.

[89] https://www.ebi.ac.uk/gwas/.