Hindawi
Security and Communication Networks
Volume 2018, Article ID 3530123, 18 pages
https://doi.org/10.1155/2018/3530123

Research Article

Differentially Private Recommendation System Based on Community Detection in Social Network Applications

Gesu Li,[1] Zhipeng Cai ,[1,2] Guisheng Yin,[1] Zaobo He,[3] and Madhuri Siddula[2]

[1]*College of Computer Science and Technology, Harbin Engineering University, Heilongjiang, China*
[2]*Department of Computer Science, Georgia State University, Georgia, USA*
[3]*Department of Computer Science and Software Engineering, Miami University, Ohio, USA*

Correspondence should be addressed to Zhipeng Cai; zcai@gsu.edu

Received 17 June 2018; Revised 28 August 2018; Accepted 5 September 2018; Published 3 October 2018

Guest Editor: Liran Ma

The recommender system is mainly used in the e-commerce platform. With the development of the Internet, social networks and e-commerce networks have broken each other's boundaries. Users also post information about their favorite movies or books on social networks. With the enhancement of people's privacy awareness, the personal information of many users released publicly is limited. In the absence of items rating and knowing some user information, we propose a novel recommendation method. This method provides a list of recommendations for target attributes based on community detection and known user attributes and links. Considering the recommendation list and published user information that may be exploited by the attacker to infer other sensitive information of users and threaten users' privacy, we propose the CDAI (Infer Attributes based on Community Detection) method, which finds a balance between utility and privacy and provides users with safer recommendations.

1. Introduction

The recommender system is one of the most popular technologies that is used in any e-commerce websites. This system is mainly developed for the ease of user interaction with the website and to suggest more products to increase the business. Many e-commerce websites such as Amazon, Netflix, and Movielens recommend products based on users' preferences (or interest). The recommender system suggests new products based on various factors including user preferences, item features, user purchase history, and other additional information such as time and space data. By introducing the recommendation system, we solve the search time of a user that in turn reduces the complex and dynamic data processing at the server. However, with the invention of social networking, recommender system has taken a drastic shift from recommending products based on user purchase history to user social activity. Most of social networking websites like Facebook, Google+, and Twitter use recommender system for pop up advertisements that might cater to the users of different tastes. Also, social networking websites like Facebook recommend communities that the user might be interested in based on the profile that he has created while signing up.

The current use of recommender system in online social networks is a content-based or a hybrid system that utilizes collaborative filtering method. This method recommends products based on user's community preferences such as user's ratings and purchase history but ignores the user and item attributes. Content-based filtering or information filtering methods [1, 2] typically match query words or other user data with item attribute information, ignoring data from other users. The hybrid recommender system combines the above two methods but ignores the user's link. As the awareness of user's privacy has increased, most users in online social networks hide personal information. Users tend to opt for selective data publication while registering in a social network. For example, some users might cancel location-based services or do not fill out any personal information. This tendency creates severe lack of information and challenges for the service provider's recommendation service. Additionally, the models mentioned above also have some limitations in dealing with data sparsity and cold start-up problems [3–6].

Although there is a lack of information due to privacy settings, users still disclose at least some of their private information, for example, rating a movie or sharing "like" or "dislike" for a photo. Some users might also reveal sensitive information and hence become vulnerable to the attacker who utilizes background knowledge for retrieving information. These knowledge-based attacks also infer personal identity information. For example, Lewis et al. found correlations of provided favored books, movies, and music [7]. In this paper, we focus on users' inherent data privacy. We assume that an attacker can anonymously collect user's data from a social network. Some users reveal their sensitive information, while others, as privacy-conscious users, have some recommended data privacy in their long-term use. However, attackers can use various methodologies to further infer user's sensitive information. Therefore, it is the main idea of this paper to add noise to the released data to hide sensitive information and to protect users' privacy.

This paper uses the real world data of online social networking websites like Facebook and Google+ for the experiments. Also, we focus on community detection as our output for the recommender system. People, as social animals, like to find groups of people who have something in common with themselves. With the advent of social networks, it has become more accessible for people to connect to those groups of users who share common interests, even if there is no friendship between them in the real world. For example, there are various movie communities in Google+. Users can join any group based on their preferences although there is no link to the group before. Based on this idea, we propose the CDAI (Infer Attributes based on Community Detection) method to provide a list of recommendations for users in different communities. At the same time, a privacy protection strategy is proposed based on the differential privacy to balance the utility and privacy.

In this work, we focus on two main issues: (1) When the user is missing much information, how to provide the recommendation list with high accuracy; (2) How to protect users' inherent data privacy while publishing data. Following is the summary of our contributions and improvements over the previous works:

(i) Two definitions are proposed, one is to define the user, and the other is to classify the attributes according to the privacy type. In this paper, we divide users into two categories, positive and negative (see Definition 8). Additionally, this paper divides the user attributes into two categories, inherent attributes and recommendation attributes (see Definition 9).

(ii) The recommendation framework of the CDAI method is proposed (see Section 5). We have built a recommendation system using community discovery and attribute dependency. The accuracy of the proposed method is superior to the accuracy of the recommender system based on user attributes and connections.

(iii) Based on the differential privacy theory and its properties, we propose the privacy-utility strategy, using differential privacy technology to protect users' sensitive information. Additionally, we also propose methods that prevent attackers from utilizing the results of CDAI to reversely infer privacy.

The remainder of the paper is organized into eight sections as follows. Section 2 provides the related work which consists of three ideologies used in this paper, including community detection, inference attack, and privacy protection. Section 3 talks about preliminaries for community detection. Section 4 defines the CDAI method, including social network model, community, dependency relationship and privacy-utility, input and output, and task definition. We propose our algorithm and the description of inference attribute and privacy protection in Section 5. Sections 6 and 7 are the evaluation and conclusion, respectively.

2. Related Works

2.1. Recommender System. The recommender system is mainly used to evaluate and predict users' preferences for the project. Based on the output type, the recommender system can be divided into the following three categories: rating prediction, ranking prediction (top-n recommendation), and classification. The goal of the first category is to fill the missing user-project scoring matrix. The goal of the second category is to provide a ranking of n items for users. The final category, classification, is to classify the candidates into the correct recommendation category. Social networking platforms vastly differ from e-commerce platforms. The e-commerce platform provides a reference standard for users to purchase a project through the evaluation of the project. However, social networks are more about providing users with "novelty" and finding things they might be interested in. In social network sites such as Facebook, if users' friends have watched a movie, they can share it on their page as well as their friends. Users who see this share on their page serve as a personal advertisement for the movie. So a user's evaluation of a project is only "watched" or "not watched", "like" or "dislike", instead of rating every item. Based on the characteristics of the social network platform itself, this paper selects the last type of task output.

The current recommender system based on the social network is broadly based on the framework of matrix factorization [8] and probabilistic matrix factorization [9]. STE [10] believes that ratings are generated by users and their trusted friends' interest. Sorec [11] uses the user preference vector to decompose the link matrix. TrustMF [12] constructs a link from a specific feature vector of trustees and then integrates it into a user preference vector to predict missing ratings. First of all, most of the above recommender systems are still based on e-commerce platform project recommendation. The results are based on user ratings of the project. However, this recommendation method has serious limitations in cold start and data sparsity; secondly, the recommendation research based on the social network mostly focuses on the recommendation of friends' link relationship. However, with the development of online networks, social networks and e-commerce are gradually breaking down each other's boundaries. The researcher can predict the target item by combining the user's link relationship with their attributes

and item attributes. This paper presents a novel recommendation method, to use community detection and user attributes to provide recommendations to the users. Therefore, it can effectively reduce the inconvenience caused by data sparsity.

2.2. Inference Attack. In the study of attribute inference attack, the target of the attacker is to spread the attribute information of social network users to users with incomplete attribute data. The attacker could be any party (e.g., online social network service, cyber criminal, advertiser, and so on). These attackers may be more interested in users' privacy attributes. They attack users' privacy by collecting public data. In addition to the risk of privacy leaks, it is also possible for an attacker to perform various security sensitive activities using inferred user attributes, such as spear phishing [13] and attacking personal information based backup authentication [14]. Reference [15] proved that an attacker (the provider of the recommender system) can use the machine learning classifier to predict the gender of the user based on the user's movie rating data. In social networks (e.g., Facebook, Twitter, and so on), public data on users include lists of users' preferences (such as movies they like or share) and lists of users' friends. Some researchers [16–23] proved that the attackers use machine learning classifiers to infer target users privacy information based on user's published data.

The current attribute inference attacks are mainly divided into two types, friend-based and behavior-based. Friend-based attacks [18, 19, 24–29] are mainly based on the information that users and friends have disclosed as well as the information of social relationship structure to infer users' sensitive attributes to attack them. Behavior-based attack [15, 17, 30, 31] is based on the public attribute information of the user, to find users with similar attributes, culture, and hobbies. By using this public information, the user's behavior attributes are inferred. In Section 3, the definition of an attacker is provided. By using the prediction results of this paper and combining with the background knowledge such as the user's public information, the attacker could deduce the sensitive attributes (such as location, political view, or sexual orientation) of the user in reverse and thus poses a threat to users' privacy.

2.3. Privacy Protect. At present, the privacy protection researches mainly focus on anonymity and access control. Anonymity is one of the most important methods for protecting privacy in the social network. There are two possible anonymization methods: node anonymity and edge anonymity. The former method hides the node information, while the latter prevents attacker based on the known information and relationship to infer node's real identity through adding or deleting edges [32]. The most popular method is to combine the two methods to hide node information and get the better privacy.

Sweeney [33] proposed the k-anonymity method. This method does not consider sensitive attributes and hence is prone to attacks like homogenous and background knowledge. Hey et al. [34] proposed K-candidate anonymity method based on Sweeney. They aggregate nodes to a different partition and published the number of partitioned nodes and edge degree between inter and outer of partition, using this anonymity graph to study the original feature.

Edge anonymization can be done by randomly removing a certain number of edges and adding the same number of random edges. Vuokko and Terzi [35] studied the reconstruction mechanism of social networks. In this mechanism, the authors have randomized both the structure and the attributes. However, they deemed that the reconstruction can be finished in polynomial time.

Differential privacy is a privacy definition proposed by Dwork in 2006 for the privacy disclosure of statistical databases [36]. They defined the computing result of the data set to be nonsensitive for a single record. Difference privacy can solve two defects in traditional privacy protection model. (1) It does not consider any background knowledge which the attacker has. (2) It has the strict definition and provided a quantitative assessment method. Many scholars begun the research in difference privacy field. For example, McSherry [37] proposed a privacy integration query mechanism (PINQ). PINQ proposed an algebraic method to describe the protection of privacy data analysis, and they ensured that the results satisfied difference privacy. Differential privacy is also frequently studied on the context of index [38–46]. This paper combines matrix manipulation of attributes with differential privacy technology, and we propose a novel privacy protection method to prevent attackers to infer user's sensitive information.

3. Preliminaries

3.1. Community Detection. Community Detection is a study in which nodes are divided into communities. There are various ways in which the nodes can be divided into communities and the splitting method is called "cluster" method. Broadly, all the clustering methods can be divided into overlapping community methods and nonoverlapping community methods. The community detection of this paper is based on Louvian algorithm [47]. We assume that users are only in a community. Because community detection is divided according to the user's target attribute. The target attribute in this paper belongs to recommendation attribute and has multiple values. Therefore, we utilize a method called "CDAI" which is a overlapping community method.

The strength and weakness of intimacy in the community, the quality of division, require a measure. Moreover, that standard is modularity. Newman first proposed the concept of modularity in 2004 [48]. Newman also proposed a large-scale community division approach. However, this approach works very slowly with big data. Therefore, many scholars proposed many improved algorithms based on Newman's research, and Louvian was one of them. Louvian algorithm is proposed based on modularity, and it has been improved so that high accuracy results can be obtained quickly under large-scale community division. At the same time, [47] also proposed the concept of modularity gain as a standard to classify nodes into the community. The formula is as follows.

$$Q = \frac{1}{2b}\sum_{i,j}\left[W_{i,j} - \frac{k_i k_j}{2b}\right]\delta\left(c_i, c_j\right) \quad (1)$$

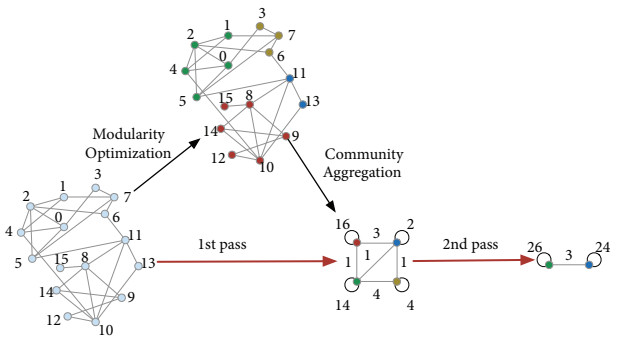

Figure 1: Visualization of the steps of the algorithm.

where $W_{i,j}$ represents the weight between node i and node j. $k_i = \sum_j W_{i,j}$ is the sum of the weights of the edges attached from vertex i, and c_i is the community to which vertex i is assigned. The δ-function $\delta(c_i, c_j)$ decides whether to put i and j in the same community. δ is 1 if u=v and 0 otherwise and $b = (1/2)\sum_{i,j} W_{i,j}$. Simplify formula (1) to formula (2):

$$Q = \sum_c \left[\frac{\sum_{in}}{2b} - \left(\frac{\sum_{tot}}{2b} \right)^2 \right] \tag{2}$$

where $\sum_{in}$ is the sum of the weights of the links inside C and $\sum_{tot}$ is the sum of the weights of the links among nodes in C.

The gain in modularity is ΔQ. If the node itself is a community, compute the gain when the node moved into other communities.

$$\begin{aligned} \Delta Q &= \left[\frac{\sum_{in} + k_{i,in}}{2b} - \left(\frac{\sum_{tot} + k_i}{2b} \right)^2 \right] \\ &\quad - \left[\frac{\sum_{in}}{2b} - \left(\frac{\sum_{tot}}{2b} \right)^2 - \left(\frac{k_i}{2b} \right)^2 \right] \\ &= \frac{1}{2b} \left(k_{i,in} - \frac{\sum_{tot} k_i}{b} \right) \end{aligned} \tag{3}$$

where k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C and b is the sum of the weights of all the links in the network. A similar expression is used in order to evaluate the change of modularity when i is removed from its community. In practice, one therefore evaluates the change of modularity by removing i from its community and then by moving it into a neighbouring community. Visualization of the steps of the algorithm is as follows.

In Figure 1, each part is made of two phases: one where modularity is optimized by allowing only local changes of communities and one where the communities found are aggregated in order to build a new network of communities. The passes are repeated iteratively until no increase of modularity is possible.

3.2. Differential Privacy. Due to the particular attack hypothesis and specific background knowledge, the privacy protection model based on anonymity fails to carry out quantitative analysis on the privacy protection intensity, so it has significant limitations in practical application. However, the differential privacy model makes it impossible for an attacker to identify whether a record is in the original data table, no matter what background knowledge he has. The formal definition of differential privacy is as follows.

Definition 1 (ϵ-differential privacy). A randomized algorithm M satisfies ϵ-differential privacy, if for any two datasets D and D' that differ only in one record, and for any possible output O of M, we have

$$P[M(D) = O] \leq \exp^{\epsilon} \times P\left[M\left(D'\right) = O\right] \tag{4}$$

where the probability of an event is denoted by $P[\cdot]$.

Definition 2 (Laplace mechanism sensitivity). For dataset D, given the function f: $D \longrightarrow R^d$. If the sensitivity of the function is Δf, then the random algorithm is as follows.

$$M(D) = f(D) + X \tag{5}$$

$X \sim Lap(\Delta f/\epsilon)$ is a random noise and follows the Laplace distribution of scale parameter $\Delta f/\epsilon$.

There are many methods of achieving differential privacy, and the most widely used are the Laplace mechanism [36] and the exponential mechanism [49]. Laplace mechanism is only applicable to numerical query results. The index mechanism is applied to query results by type. The exponential mechanism uses random sampling to satisfy specific distribution to realize differential privacy instead of adding noise. The central principle of the exponential mechanism is to define a practical evaluation function q and calculate a practical value for each output scheme. The output scheme with high score is more likely to be published, to ensure the quality of published data. The selection of the evaluation function q must have the lowest sensitivity possible. Sensitivity is the maximum change to the query result caused by deleting one arbitrary tuple in the dataset. It is a crucial parameter to add noise. Its specific definition and formula are as follows.

Definition 3 (exponential mechanism sensitivity). Given a practical evaluation function q, the sensitivity of q is defined as

$$S(q) = \max_{D_1, D_2, r} \|q(D_1, r) - q(D_2, r)\| \tag{6}$$

where D_1 and D_2 are datasets with only one record difference between any pair, and r represents any legitimate output.

According to the definition of sensitivity, Theorem 4 can be obtained.

Theorem 4. *Given dataset D, q is a utility valuation function for all output of dataset D. For dataset D and function q, if algorithm K satisfies the probability of output r and is proportional to* $\exp(\epsilon q(D, r)/2S(q))$, *then algorithm K satisfies* ϵ*-differential privacy.*

The ϵ-differential privacy model has the following properties.

Property 5 (sequence composition property). Assuming random algorithm combination $M = M_1, M_2, \ldots, M_n$. For the same dataset D, M_i provides ϵ_i-differential privacy, and M provides $\sum_{i=1}^{n} \epsilon_i$-differential privacy.

Property 6 (parallel composition property). Assuming random algorithm combination $M = M_1, M_2, \ldots, M_n$. For the disjoint subset of dataset D, ϵ-difference privacy is satisfied, respectively. Then M provides maximum ϵ-difference privacy protection for D.

4. Problem Statement

4.1. Social Network Model

Definition 7 (social network). A social network is an undigraph represented by $G(V, E, \chi)$. It consists of user set, friendship links, and the set of user attributes. The user set includes positive users and negative users represented by V_P and V_N, respectively, and $u_i, u_j \in V(1 \leq i, j \leq |V|)$. Friendship link is represented by E; in this paper all links are unweighted undigraph. But when dealing with similarity (Definition 11), links become weighted undigraph. The set of user attributes is denoted by χ, which consists of recommended attributes and inherent attributes represented by χ_R and χ_I, $(\chi_R, \chi_I) \subseteq \chi$ $\chi = \chi_R \cup \chi_I$. For an arbitrary user $(u_i, u_j) \in V(1 \leq i, j \leq |V|)$, their friendship links $(e_i, e_j) \in E$ also indicate $(e_j, e_i) \in E$.

Definition 8 (positive users and negative users). We divided users into two categories: positive users and negative users. We defined positive users as the people who filled most profile, nonsensitive for information, like sharing the information with other people. Positive users are denoted by V_P. V_N represented negative users, who have strict privacy awareness, blanking much information, inactivity on the online social network. These negative users are our target users. $u_i, u_j \in V(1 \leq i, j \leq |V|)$, $V_P \cup V_N = V$. In this paper, we consider the published information including the friend links and users attributes.

Definition 9 (attribute set). In this paper, we assume that the recommendation attributes are the items in the traditional recommender system, and the attribute value is the specific name of the items. This paper uses known recommendation attributes and some inherent attributes to predict user preferences and provide users with a list of recommendations for relevant content. The target attribute in this paper is a recommendation attribute, and the community is divided according to the target attribute. The inherent attribute is defined as the user's sensitive privacy information, which includes the user's age, gender, political inclination, address, and other personal data. All the inherent information of the user is regarded as an inherent attribute. Generally, such an attribute is a single value attribute. The recommendation attribute is considered as nonsensitive privacy information, which includes books, games, movies, and other information that users like. All items that provide services to users and can be recommended to users can be called recommendation attribute, which usually has multiple attribute values. For the convenience of the experiment, all single value attributes are regarded as inherent attributes in this paper, and all multivalue attributes are considered as recommendation attributes.

For any user u_i, $u_i \in V(1 \leq i \leq |V|))$, attribute set is denoted by $\overrightarrow{X_i} \subset \chi(1 \leq i \leq |V|)$, $x_j \in \overrightarrow{X_i}(1 \leq j \leq |\overrightarrow{X_i}|)$. Attribute category is T, $T_m \subset T(1 \leq m \leq |T|)$. T is the set of attribute categories in the social network, including the inherent category T_I and recommended categories T_R, $(T_I, T_R) \subseteq T$. For a certain attribute category, L is used to represent the attributes value also viewed as the label. $L_m \subset T_m$, $l_n \in L_m(1 \leq n \leq |L_m|)$.

For example, $x_i = \{T_{I_m} : l_1; \ldots; l_n\}$, which means x_i is for category T_{I_m} with value list $l_1; \ldots; l_n$ $n \geq 1$.

Different attribute categories have a differential number of attributes values, and the value can be single or multiple. Attributes such as gender and age need just single value. For categories like "Favorite movies", the input can be "Titanic" and "Darkest Hours". This kind of category has multiple values. Moreover, categories may be with none value for some users, for example, "Religion view". In specific applications, a user can determine which categories are sensitive and hide the categories. For example, Facebook users can directly hide their sensitive attributes in their profile. Following is an example:

$G = (V, E, \chi)$

$V = \{u_1 = Tom, u_2 = Ann\}$

$\chi = \{X_1, X_2\}$

T = {Favorite Movies, Favorite Books, Age, Gender, Political View}

T_R = {Favorite Movies, Favorite Books}

T_I = {Age, Gender, Political View}

$\overrightarrow{X_1}$ = {x_1 = { Favorite Movies: Avatar, Iron Man}; x_2 = {Favorite books: Machine Learning, AI }; x_3 = {Age: 26}; x_4= { Gender: Male }; { Political View: The Liberal Party}}

$\overrightarrow{X_2}$= {x_1 = {Favorite Movies: Iron Man, Bat Man}; x_2 = {Favorite Books: The Moon and Sixpence}}

$e_{1,2} \in E$ $e_{2,1} \in EC$ have five categories: two of them are recommendation attribute categories T_R and the others are inherent attribute categories represented by T_I; two users are u_1 and u_2. In this case, u_1 published two favorite movies, two favorite books, age, gender, and political view. However, u_2 released two favorite movies, one favorite book. From the case we can infer u_1 is a positive user, and u_2 belongs to negative user. We know the relationship between u_1 and u_2; they are friends.

4.2. Community Detection

Definition 10 (community). The community collects the nodes which have the same feature in the network. The vertices within the community are tightly linked, a low density between-group edges [50]. With the growth of the network scale, the community scale becomes bigger, and so is the number of nodes. We merge the original relationship between users and attributes by dividing the community while building a strong link in the community. Users having

same attributes in a community establish a new graph represented by G'.

In this part, to obtain the similarity of attributes and relationship links, getting a new social network graph G', where u_i and u_j have link, return to 1, otherwise return to 0. At the same time, u_i and u_j have a lot of same attributes, which proved they have tight relationship. The similarity is equivalent to the weight of user edges.

$$G' = G \cap SimA = \begin{cases} 0, & e_i \cap e_j = 0 \ or \ SimA = 0 \\ SimA, & e_i \cap e_j = 1 \ and \ SimA \neq 0 \end{cases} \quad (7)$$

Definition 11 (similarity of attributes between users). To obtain the similarity between users, use Jaccard index to measure that. It is defined as the size of the intersection divided by the size of the union of the sample sets.

$$SimA(i, j) = \frac{\left| x_i \cap x_j \right|}{|X|} \quad (8)$$

where x_i and x_j represented the attributes of u_i and u_j and the set of all attributes is denoted by X.

4.3. Attribute Dependence

Definition 12 (the accuracy of dependence). $\vartheta_{m \in M}^{T_R}(G)$ is the accuracy of dependence, classifiers set is M, and m is the specific classifiers in the set. $m_n \in M \ (1 \leq n \leq |M|)$, where n is the number of classifiers. T_R represents the target recommended attribute type.

Definition 13 (highest-dependence). Experiment used the selected classifiers to train. Remove an attribute category in each train, and find the attribute category that causes the maximum gain change, which is the highest-dependence for the target attribute category. If there are several attributes that result in large change, then regard them as highest-dependence.

$$\max \Delta_{m \in M}^{T_R} = \vartheta_{m \in M}^{T_R}(G)\Big|_T - \vartheta_{m \in M}^{T_R}\left(G'\right)\Big|_{T' = T - t_i} \quad (9)$$

When removing an attribute category, the gain of accuracy of recommended attribute category is denoted by $\Delta_{m \in M}^{T_R}$. T' is the set of rest attribute categories, and these attribute categories are the removed low-dependence attribute categories t_i.

4.4. Attacker and Attack Model. In general, an attacker can be anyone who is interested in user attributes. However, the attackers in this paper mainly refer to those cyber criminals who use public background knowledge to infer user attributes and attack users. Cyber criminal uses user attributes to perform certain cyber attacks, for example, spear phishing attacks [13] and personal information attacks based on backup authentication [14]. These behaviors will leak users' privacy and may cause users to lose property or even threaten their security.

In this paper, we assume that data is published from the perspective of social network providers and cyber criminal has some background knowledge. The knowledge of cyber-criminal is $K(V^K, E^K, \chi^K)$, where $V^K = V, E^K = E, \chi^K \subseteq \chi$. χ^K is the set of attributes obtained by machine learning, which is strongly dependent on the target attribute. Publish the set of extracted attributes. Positive users post complete information whereas negative users belong to the user group with strong privacy awareness, and they only release a small amount of information or no information to the outside world. Therefore, the goal of this step is to reduce the exposure of user information. Publish the attributes set after dimension reduction. Hide irrelevant or weakly related attributes to protect some user information.

4.5. Utility Based on Privacy Protection. This paper aims to propose a method to provide more services for negative users and, meanwhile, balance the utility-privacy tradeoff. The existing definition of privacy has difference privacy, K-anonymity [33] and L-diversity [51], which are only for inherent data and are not suitable for inferring attribute. In the current research work, the usual method is adding noise in original data. Rather than the traditional method, we infer attributes for negative users whose data are incomplete. However, consider the attackers use background knowledge to inference attack. Therefore, data-sanitization is necessary, which could protect user privacy, but over data-sanitization will lead to a reduced utility. Based on the above problem, utility is based on privacy protection as follows.

Definition 14 (utility). Given the social network G, stronger protection needs more noise, which leads to less utility. Accordingly, there is the interaction of constraints between privacy and utility. When utility satisfied the condition, get the maximum utility under the good privacy protection. The condition is given as follows.

$$\zeta \geq \frac{\Delta_{m \in M}^{T_R}}{\vartheta_{m \in M}^{T_R}(G)} \quad (10)$$

where ζ approximates to 1, which means utility better. Otherwise, it is close to 0, utility lower.

4.6. Input and Task. Based on the above definitions, given the input and output definitions about this paper. The user-specified thresholds on privacy-utility are given in Section 7.

Input. Social graph is denoted by $G(V, E, \chi, T, L)$, where user set V includes V_P and V_N. Friendship link set is denoted by E, the set of user attributes is denoted by χ, and the set of attribute categories T, $T = T_R \cup T_I$. The set of labels L, $L = L^K \cup L^U$.

The set of known labels for users $u_i \in V^K$ is L^K, where V^K is the set of users with known labels. V^K includes known V_P and V_N. L^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, mainly the negative users. User-specified utility threshold is denoted by ζ.

Table 1: Description of symbols.

Symbol	Description
Q	The modularity, evaluation index of community detection
b	$b = (1/2)\sum_{i,j} W_{i,j}$, b is the sum of the weights of all the links in the network
$W_{i,j}$	The weight between node i and node j
k_i	The sum of the weights of the edges attached from vertex i
$k_{i,in}$	The sum of the weights of the links from i to nodes in C
C	The community. $c_i \in C$
M	An algorithm. $m \in M$
$f(D)$	The result of any query operation f
X	$X \sim Lap(\Delta f/\epsilon)$ is a random noise and follows the Laplace distribution of scale parameter $\Delta f/\epsilon$
$S(q)$	The sensitivity of q
$q(D, r)$	q represents a given practical evaluation function. D is data set, r represents any legitimate output
$G(V, E, \chi)$	G is a social network. V represents the set of users in G. $(V_N^U \cup V_N^K = V_N) \cup (V_P^U \cup V_P^K = V_P) = V$. The friendship link denoted by E. The set of user attributes is denoted by χ. $\chi_R \cup \chi_I = \chi$.
T	The set of attribute categories in G, $T_R \cup T_I = T$
L	The attribute value, equal to label, $L_m \subset T_m, l_n \in L_m$ $(1 \le n \le \lvert L_m \rvert)$
$SimA$	The similarity of attributes between users
$\vartheta_{m\in M}^{T_R}$	The accuracy of dependence
ζ	The parameters for evaluating utility, it is close to 1 that means utility better, otherwise, it is close to 0, utility lower

Output

Task 1: Prediction method can predict L^U for negative users who do not know the label of recommended attribute.

Task 2: Publish a noise-added recommendation list.

For the convenience of readers, the symbols involved in the paper are summarized in Table 1.

5. CDAI Method

The framework of this paper is mainly divided into three parts: the first part is the data processing, the second part is the recommendation, and the third part is the privacy protection. In the first part of data preprocessing, the machine learning algorithm is mainly used to find the dependent attribute of the target attribute and delete the weak dependent attribute, so as to reduce the dimension of high-dimensional original data. In the second part, the recommendation combines the classification algorithm of community detection and machine learning to improve the accuracy of prediction. The third part is privacy protection, the differential privacy based on recommended. Differential privacy is constructed for the naive Bayesian algorithm of community discovery and machine learning classification, respectively. After differential privacy treatment, the recommendation results are published to the public, which makes it impossible for the attacker to infer more information about the user in reverse.

5.1. Data Processing. This section uses machine learning to get the dependent attributes of the target attribute. On the one hand, dimensional reduction processing is carried out for high-dimensional data, so as to improve the overall running speed. On the other hand, after deleting the original data, publishing can reduce the information leakage of users from the source and protect their personal privacy.

Input: $G = (V, E, \chi)$;
Output: D_1
(1) **for** $i \in T$ **do**
(2) $T_{del,i}$ use M $\longrightarrow \vartheta_{m\in M}^{T_R}(G)$
(3) $\Delta_{m\in M}^{T_R} = \vartheta_{m\in M}^{T_R}(G)|_T - \vartheta_{m\in M}^{T_R}(G')|_{T'=T-t_i}$
(4) **if** $\Delta_{m\in M}^{T_R} = \max \Delta_{m\in M}^{T_R}$ **then**
(5) $T_i, T_R \longrightarrow high - dependence$
(6) others are low-dependence $\longrightarrow T_{del}$
(7) **end if**
(8) $T' = T - T_{del}$
(9) $T' useM \longrightarrow T_B^U$
(10) **end for**
(11) Get D_1

Algorithm 1: Data processing.

This section uses the machine learning classifier (such as KNN, NB, SVM) to predict the labels of the target attribute. Remove one weak dependency attribute at a time according to formula (9). And use formula (10) to determine whether the deletion will continue. When the utility is reduced by more than ζ, the deletion stops. The remaining attributes are highly dependent. The recommendation accuracy obtained based on this step will be reduced. But by setting the utility, it is reduced to an acceptable range. See Algorithm 1.

5.2. CDAI Recommendation Method. In social networks, users join specific communities according to their preferences. Users in the community may not have any prior

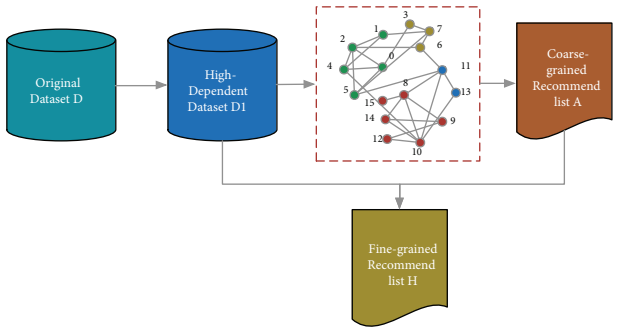

Figure 2: CDAI recommendation structure.

connection to each other. However, these users may have high attribute similarity. From the perspective of sociology, users with high similarity are more likely to have common preferences. Based on this idea, a coarse-grained prediction method based on community detection is proposed. At the same time, even in the same community, there is a high similarity of attributes, but the difference of one of its attribute values still has a specific influence on the final prediction results. For example, in a movie community, both people like comedy movies. However, because of gender differences, a more detailed classification might yield different results. Therefore, in the second stage of the recommendation method, the prediction based on the attribute and machine learning is proposed, to modify the partial prediction results of the first stage. The recommended structure is shown in Figure 2.

This paper is based on the user's friend links and attributes. The community detection algorithm and machine learning classification algorithm are combined to obtain high recommendation accuracy. The recommendation process is divided into two stages. The first stage is a coarse-grained recommendation based on community detection. The Louvain algorithm combined with attribute similarity divides users into communities based on target attribute. The following formula is used to obtain the most probability of target attribute labels in the entire network (see formula (12)), which is assigned to users who are belonging to independent communities and do not have the target attribute label. At the same time, each community has a label that gets the most votes, assigning the label to an empty target attribute in the community. Here, the community weight is added to the probability of getting the labels in each community. The modularity of each community is regarded as the weight of the community (see formula (11)).

$$P_{l_j} = \frac{|l_j|}{|c_i|} Q_{c_i} \tag{11}$$

$$l_{\max j} = \arg\max_{l_j} \sum_{c_i} P_{l_j} \tag{12}$$

In the second stage, the machine learning classification algorithm is used to adjust the prediction results. Thus, higher prediction accuracy can be obtained. The main step of this part is to use the machine learning classification algorithm to obtain the prediction results. In the first step, the predicted results of the independent community remained unchanged. If the predicted results in the community are different from those in the first step, the predicted results of the same community in the first step are replaced by those obtained in the second step. End of recommendation. See Algorithm 2.

5.3. CDAI Privacy Protection and Publishing. It is known from the data preprocessing section that the original data released by social network providers is preprocessed data, while other data is hidden data, which is not visible to the public. Based on this assumption, the maximum range of background knowledge obtained by an attacker is the data published by a social network provider, that is, preprocessed data. For this part of the dataset, this paper uses differential privacy to protect the dataset when the attacker knows all about it. The structure of this part is consistent with that of the recommendation part and consists of two parts. The structure is shown in Figure 3. The first part is a differential privacy based on community detection. The second part is the differential privacy based on Naive Bayes [52].

5.3.1. Differential Privacy Based on Community Detection. This part of differential privacy is based on community discovery. To achieve the effect of privacy protection, it is necessary to hide the link relationship between users in the community detection. Therefore, Laplace noise is added to the community weight to hide the link relationship in the community. Add noise to the number of links to node I within and outside the community to hide the true link relationship. Privacy budget is ϵ. Function sensitivity Δf is related to function. The number of users within the community is the maximum sensitivity of K_{in}. The number of users outside the community is the maximum sensitivity of $\sum_{tot}$. Therefore, the combined sensitivity of the two sequences is the total number of users in the social network. Add noise to formula (2) and change it to the following form:

$$Q^*_{c_i} = \frac{\sum_{in} + lapace\left(k_{i,in}/\epsilon\right)}{2b} - \left(\frac{\sum_{tot} + lapace\left(k_i/\epsilon\right)}{2b}\right)^2 \tag{13}$$

Algorithm 3 achieves the ϵ-differential privacy protection by adding Laplace noise. Therefore, it is necessary to prove that the algorithm strictly abides by the ϵ-difference privacy definition. Each step of the algorithm will be analyzed and proved according to the definition and properties of Section 3.

Differential Privacy Proof. The algorithm adds Laplace noise to each link weight of nodes in Q_{c_i}. We know from the formula for Q that Q is made up of two parts. The first part is c_i internal weight. When only one node u_r is added, Node u_r is connected to $|c_i|$ users in c_i. Node u_r is connected to $d_i n$ users in c_i, and d is at most equal to all $|c_i|$ users in c_i. The second part is c_i external weight. The external maximum number of users is $|V|-|c_i|$. Because it adds Laplace to each link weight of the nodes, when two data sets differ by a tuple, the sensitivity in c_i is $k_{i,in}$. If the maximum weight is 1, the sensitivity is $|V|$. Similarly, the maximum sensitivity outside c_i is also $|V|$. Therefore, ΔQ_{u_r} satisfies $\epsilon/|V|$ -differential privacy. We know from the formula for Q that each user u_r joins is independent of the other. So it is parallel relationship. According to the

Input: $G = (V, E, \chi)$;
$V = Users$;
$E = Links$;
$\chi = Attribute\ set$;
L^K =The known labels of users($L^K = \{l_i \in T_R \cup T_I \mid u_i \in V^K, V^K = V^N \cup V^P\}$);
Output: L^U =The unknown labels of users($L^U = \{l_i \in T_R \mid u_i \in V^U, V^U = V - V^K\}$);
(1) **Phase 1:** Get the coarse-grained $L^U_{coarse-grained}$, List A
(2) Get a new social network graph G'
(3) $SimA(i, j) = |x_i \cap x_j|/|X|$
(4) $G' = G \bigcap SimA$
(5) Initiallize Each Node to a Separated Community
(6) **repeat**
(7) **for** $i \in |V|$ **do**
(8) **for** $j \in |V|$ **do**
(9) Remove i from its community, place to j's community
(10) Compute the composite modularity gain Δ
(11) **end for**
(12) Choose j with maximun positive gain (if exists) and move i to j's community
(13) Otherwise i stays in its community
(14) **end for**
(15) **until** No further improvement in modularity
(16) Gets the label probability in each community, $P_{l_j} = (l_j/c_i)Q_{c_i}$
(17) Find $P_{max} \longrightarrow l_{max}$, Assign l_{max} to $L^{U_{in}}(U_{in} \in c_i)$
(18) $l_{\max j} = \arg\max_{l_j} \sum_{c_i} P_{l_j}$
(19) Assign $l_{\max j}$ to $L^{U_{independent}}(U_{independent} \in c_i)$
(20) Get the coarse-grained L^U_A
(21) **Phase 2:** Get the fine-grained $L^U_{fine-grained}$ for the unknow labels of users in community. Get the list B.
(22) **for** $j \in |V|$ **do**
(23) **if** $A_j \neq B_j$ **then**
(24) $A_j = B_j$
(25) Users of the independent community, the labels remain the same
(26) **end if**
(27) **end for**
(28) Get final L^U

Algorithm 2: CDAI recommendation method.

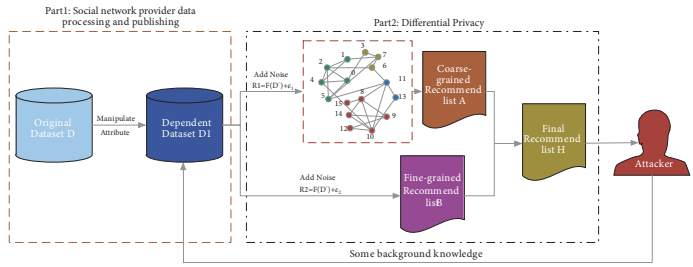

Figure 3: CDAI privacy protection structure.

parallel property of differential privacy, Q_{c_i} satisfies $\epsilon/|V|$ - differential privacy.

5.3.2. Differential Privacy Based on Naive Bayes. Based on the second part of the recommendation, the machine learning classification algorithm is used to predict labels. In this paper, KNN, NB, and SVM are mainly used. This part only constructs differential privacy for naive Bayesian algorithm. Generally, an NB classifier has a good effect on data classification, so it can be considered that it has a good estimate of joint probability density $P(X, Y)$. So, for a randomly generated feature vector x, if the classifier divides it into some class of c_j, we can think of (x, c_j) as a random sample from $P(X, Y)$. The process of selecting the maximum posterior probability class c_{max} is Naive Bayes. It is regarded as the process of voting according to conditional probability and prior probability. According to the exponential mechanism, a decision algorithm in accordance with differential privacy protection is designed. The output of the algorithm is a class variable. To reduce the effect of parameters on the posterior

Input: c_i = Divided communities in G,$c_i \in C$;
L^K = Known labels of users($L^K = \{l_i \in T_R \cup T_I \mid u_i \in V^K, V^K = V^N \cup V^P\}$);
Output: L^{*U} = Unknown noise labels of users($L^U = \{l_i \in T_R \mid u_i \in V^U, V^U = V - V^K\}$);
$L^{*U_{in}=}$ Unknown noise labels of users, users in community i. $U_{in} \in c_i$
$L^{*U_{independent}=}$ Unknown noise labels of users, users in independent community i. $U_{independent} \in c_i$
(1) **for do**$c_i \in |C|$
(2) **for do**$l_j \in |L^U|$
(3) $P^*_{l_j} = (|l_j|/|c_i|)Q^*_{c_i}$
(4) **end for**
(5) **end for**
(6) Find the max probability $P^*_{\max l_j}$ in Community i, $\max l_j \longrightarrow L^{*U_{in}}$
(7) $l_{\max j} = \arg\max_{l_j} \sum_{c_i} P_{l_j} \longrightarrow L^{*U_{independent}}$
(8) Get L^{*U}_A

Algorithm 3: Differential privacy based on community detection.

probability, Laplace smoothing of conditional probability is carried out, and the following definitions of utility function are given:

$$P_\lambda\left(X^{(j)} = a_{jl} \mid Y = c_k\right) = \frac{\sum_{i=1}^{N} I\left(x_i^j = a_{jl}, y_i = c_k\right) + \lambda}{\sum_{i=1}^{N} I\left(y_i = c_k\right) + S_j\lambda} \tag{14}$$

N is the number of samples, $I(x)$ is the indicator function. $x^{(j)_j}$ is the jth feature vector. S_j is the number of values of $x^{(j)_j}$. When $\lambda = 1$, it is called Laplace smoothing.

$$q\left(D, x, c_k\right) = \log\left(p\left(c_k\right)\right) + \sum_{i=0}^{n} \log\left(p\left(x^{(i)} \mid c_k\right)\right) \tag{15}$$

Theorem 15. *For any two random samples x and x' that differ by only one attribute, $q(D, x, C_k)$ has a local sensitivity of* $\log(2N)$. *N is the number of tuples in dataset D.*

Proof. In a given dataset, the conditional probability of $p(x_i^{(j)} \mid c_k)$ after Laplace smoothing is bounded. Its upper bound is 1. $\sum_{i=1}^{N} I(x_i^{(j)} = a_{jl}, y_i = c_k) \geq 0, \sum_{i=1}^{N} I\ (y_i = c_k) \leq N, S_j \leq N, \lambda = 1$. When the inequality equals, the conditional probability is minimized $p(x_i^{(j)} \mid c_k) = 1/2N$.

$$p\left(x_i^j \mid c_k\right) \in \left[\frac{1}{2N}, 1\right) \tag{16}$$

□

Given the data set D, for the n dimensional feature vectors x and x' with only the jth dimension different, the only thing that is not equal is the conditional probability of the jth dimension. Namely, $q(D, x, c_k) - q(D, x', c_k) = \log(p(x^{(j)} \mid c_k)) - \log(p(x'^{(j)} \mid c_k))$. According to the formula above,

$$\left|q\left(D, x, c_k\right) - q\left(D, x', c_k\right)\right| < \log\left(2N\right) \tag{17}$$

By Theorem 15, the sensitivity of the utility function Δq only related to the size of the data set N. Replacing global sensitivity with local sensitivity will leak information related to the size of the dataset. Here $\Delta q = \lceil \log(2N) \rceil$. Because the logarithm function is a slow growing function, therefore, the uplift operation can reduce the possibility of data set size information leakage. Finally, the label c_{max} is selected with probability $p_{c_r} \propto \exp(\epsilon q(D, x, c_r)/2\Delta q)$. Satisfy ϵ-difference privacy.

According to the serial nature of differential privacy. The difference privacy based on community discovery is concatenated with the difference privacy based on Bayesian. It satisfies $(|V| + 1)\epsilon/|V|$ -differential privacy.

6. Evaluation

6.1. Datasets. We will compare Attribute-Link and CDAI using real dataset. In our experiment, we investigate two different datasets. The first one is Facebook dataset (https://snap.stanford.edu/data/egonets-Facebook.html) and the second one is Google+ dataset (https://snap.stanford.edu/data/egonets-Gplus.html). Both of them have the ego-network and profile of each user and anonymized the information. From Facebook we got profile and network data from 10 ego-networks and consist of 88,234 links and 4,039 users with an average circle size of 22 friends. Each user has own profile, which has 22 attribute categories. Profile includes birthday, education classes, education school, education year, and hometown. From Google+ we obtained profile and network data from 132 ego-networks. It consists of 13,673,453 links and 107,614 users. 132 ego-networks represented 132 users. The Google+ dataset is quite different to those from Facebook, in the sense that their creators have chosen to release them publicly. It contains 6 attribute categories which, respectively, are gender, institution, job, last name, city, and university. In experiment, we choose the single value attribute categories as inherent attributes and the multiple-value attribute categories regard as recommendation attributes. Table 2 provides the general statistics of the two datasets. This table shows that all of the two graphs are almost fully connected.

6.2. Experiment Settings. We describe the metrics adopted to evaluate various inference, training and testing, and parameter settings.

Table 2: General statistics about the two datasets.

Network Property	Facebook	Google+
Number of nodes	4039	107614
Number of friendship links	88234	13673453
Number of attribute categories	22	6
Number of recommend attribute categories	13	4
Number of inherent attribute categories	9	2
Diameter (longest shortest path)	8	6
Average clustering coefficient	0.6055	0.4901

Evaluation Metrics. The aim of this paper is to provide recommendation service for users. At the same time, it prevents attackers from using background knowledge and recommendation content to push back other information of users. Since we set the target user is negative user. Due to the evaluation, assume all test users are negative users. Randomizing a multiple-value attribute category is recommended attribute category and predicts the attribute values, for example, we know that Facebook has 22 attribute categories from Table 2. There are 13 recommended attribute categories and 9 inherent attribute categories; therefore, randomly choose one from recommended attribute categories as the target attribute to predict. Use the rest of attribute categories to find the relationship with chosen recommend attribute. We use these models to predict: (1) Attribute-Link inferring attributes model; (2) CDAI model.

In this section, given 3 evaluation metrics, respectively, accuracy, modularity, and utility, the community detection assesses using modularity. The general range of modularity is between -0.5 and -1. When the range is between 0.3 and 0.7, the clustering effect is good. The rest of metrics evaluates the prediction result. Accuracy is the ratio of the number of samples correctly to the total sample size for a given test data set. The high-dependence in Section 4 is based on accuracy. The formula is as follows.

$$Accuracy = \frac{l_{correct}}{L_{total}} \tag{18}$$

where $l_{correct}$ is denoted by the number of correct labels obtained using the classifiers. L_{total} is the number of test samples.

In addition, the utility evaluation metric is discussed in Section 4.5; we will use the definite formulas for evaluation; see formula (10). In this paper, we set ζ as 0.2. The classifiers include KNN, Naive Bayes, and SVM. The ϵ is 0.001, 0.01, 0.05, 0.1, 0.5, 1,1.5, 10, 100, 1000, 10000.

Train Dataset and Test Dataset. Facebook dataset consists of 10 ego-networks, and Google+ dataset includes 133 ego-networks. In this experiment randomly select the train dataset and the test dataset, and the ratio is 8 to 2. We assume all users are target users in test dataset. Each user's attributes consist of recommended attributes and inherent attributes. The target attribute is randomly selected from recommendation attribute categories. Due to the profile obtained in real world, therefore, some information is missing. These links between centre user and his followers are nature weak relationship in network graph, extracting high-dependence links from each ego-network and dividing the community based on the attribute similarity.

Table 3: The modularity of two datasets.

	Link-Att	CDAI
Facebook	0.52896	0.52428
Google+	0.19456	0.30578

6.3. No Noise Labels Predicted. In this section, we compared Attribute-Link method and CDAI method, exploiting modularity, accuracy, and utility to evaluate the two methods. Table 3 shows the comparison results of initial modularity between the two methods.

From Table 3, no matter which datasets the modularity of CDAI is relatively stable and the results are better than Attribute-Link method. Figures 4(a)–4(c) are the test results of Google+ dataset; no matter which classifiers the prediction result of CDAI is significantly better than the other one. In the classifier of KNN in Figure 4(a), the accuracy does not change when using inherent attributes or recommended attributes. In the process of removing the attributes, when removing the sencon attribute category, classification method based on Attribute-Link improved its accuracy; meanwhile using CDAI method the result has no change that means the latter one has better stability than the former. From Figure 4(b) of NB classifier, when only using inherent attributes for prediction, the accuracy improved. However, when deleting the third attribute category, the accuracy of two methods all improved. The SVM classifier has always been smooth in the figure. Figures 4(d)–4(f) are the results of Facebook dataset. There is a larger decline when only using single attribute categories (inherent attribute categories) that means the recommended attribute categories have a great impact for prediction. In NB classifier, no matter which attribute is deleted, the accuracy is stability. But the other classifiers have a sharp decline when the attribute deleted the fourteenth. The results of CDAI in three pictures are always better than Attribute-Link.

Figures 5(a)–5(f) are the utility results of Google + and Facebook datasets. Based on the definition of utility, the values are closer to 0, and the utility is better. From Figure 5, either data or classifier, the two methods on the utility results are below setting values; at the same time the results obtained from CDAI are better than the Attribute-Link. Figures 5(a)–5(c) are the results of Google +. Using KNN classifier, when removing the second attribute category based on Attribute-Link method, the utility is suddenly increased that illustrates that there is bigger influence on the utility. However, based on CDAI, no matter which result is removed, the attribute categories have stabilization. In Figure 5(b) using NB classifier, the two methods have same line trend and when removing the third attribute category there has a big change. For Figure 5(c) based on SVM classifier, the results are smooth and steady use Attribute-Link method and CDAI method.

The results of Facebook dataset are shown in Figures 5(d)–5(f). In the three subfigures, when deleting

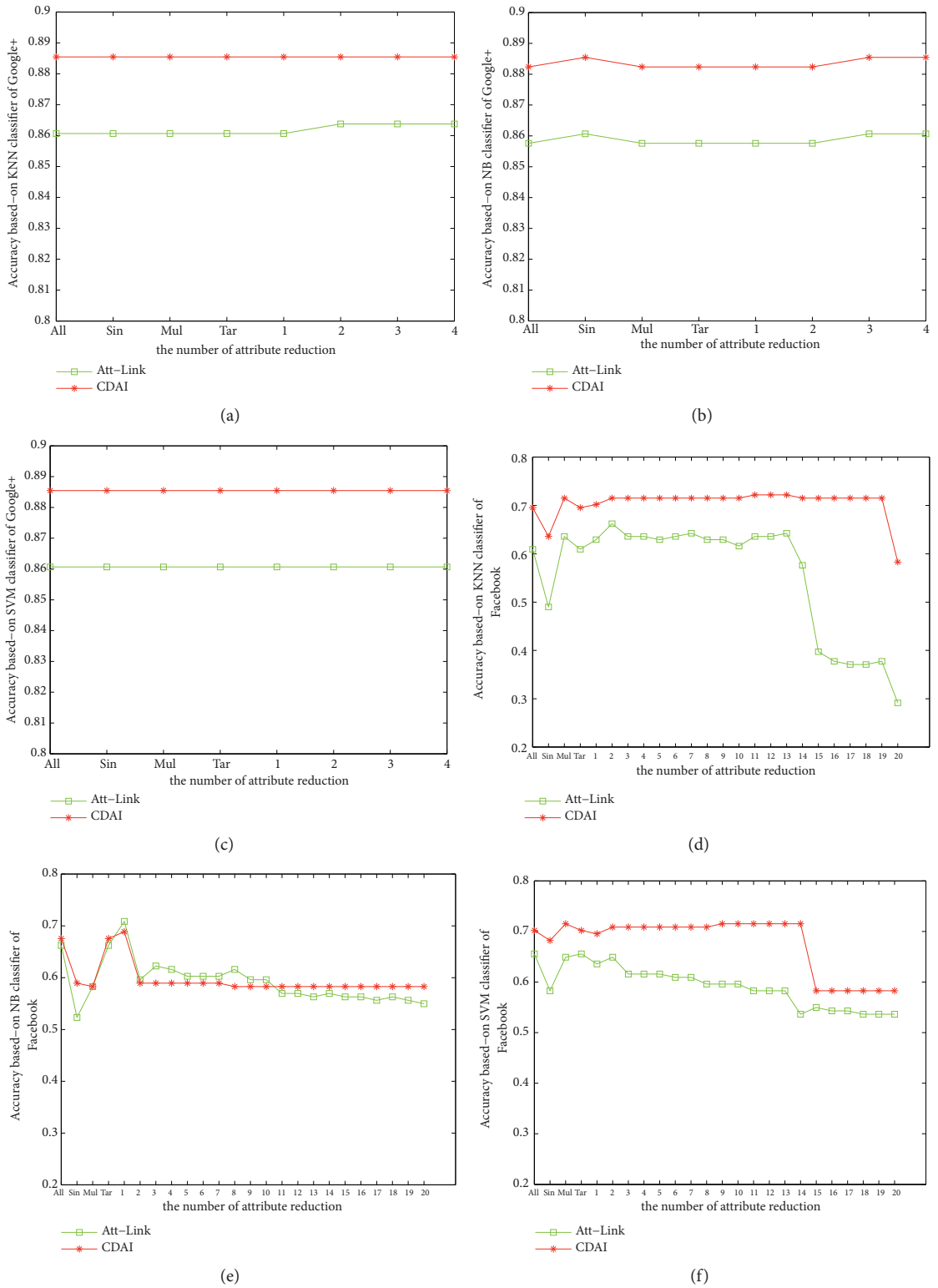

FIGURE 4: The accuracy of predict based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

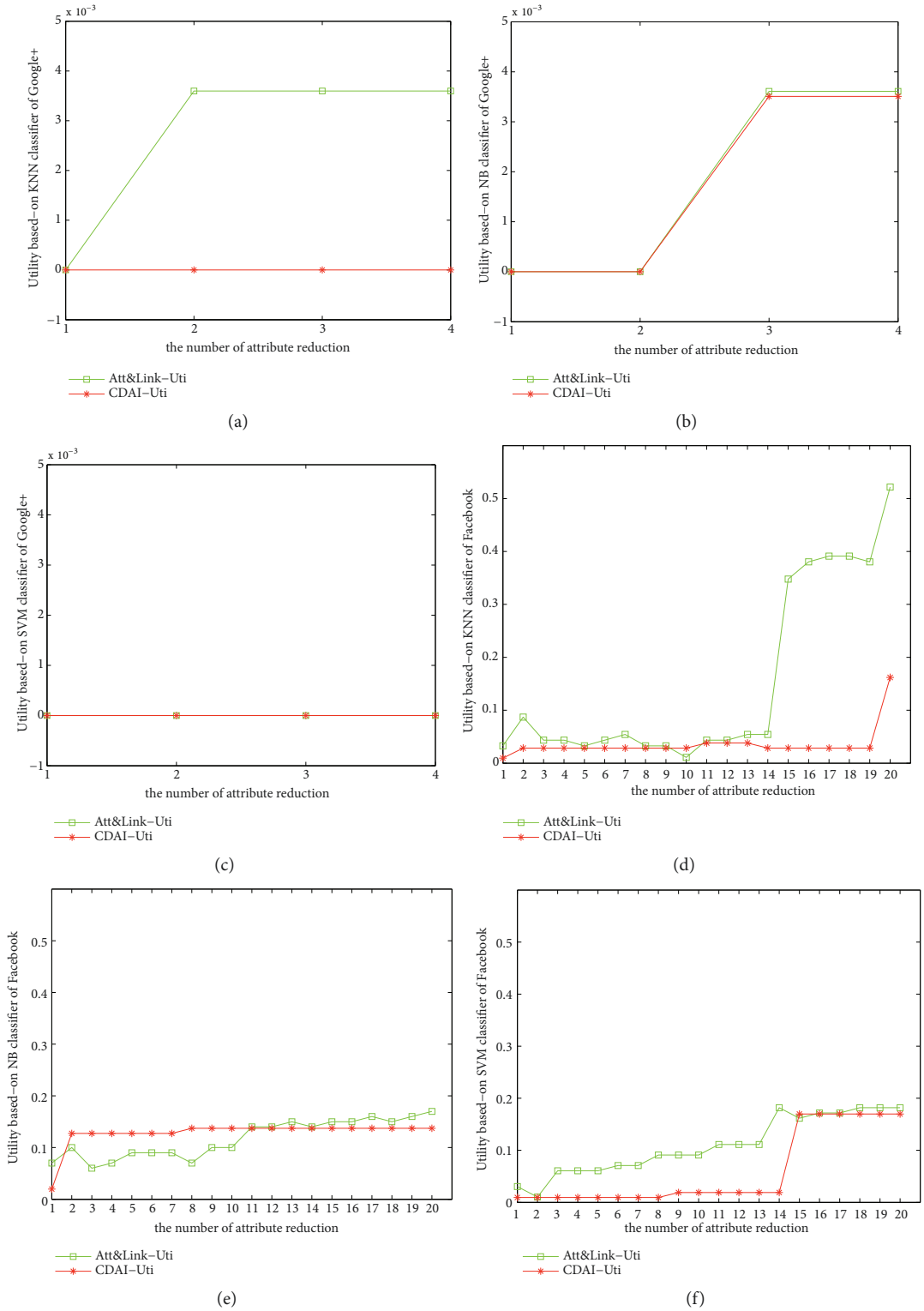

FIGURE 5: The utility based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

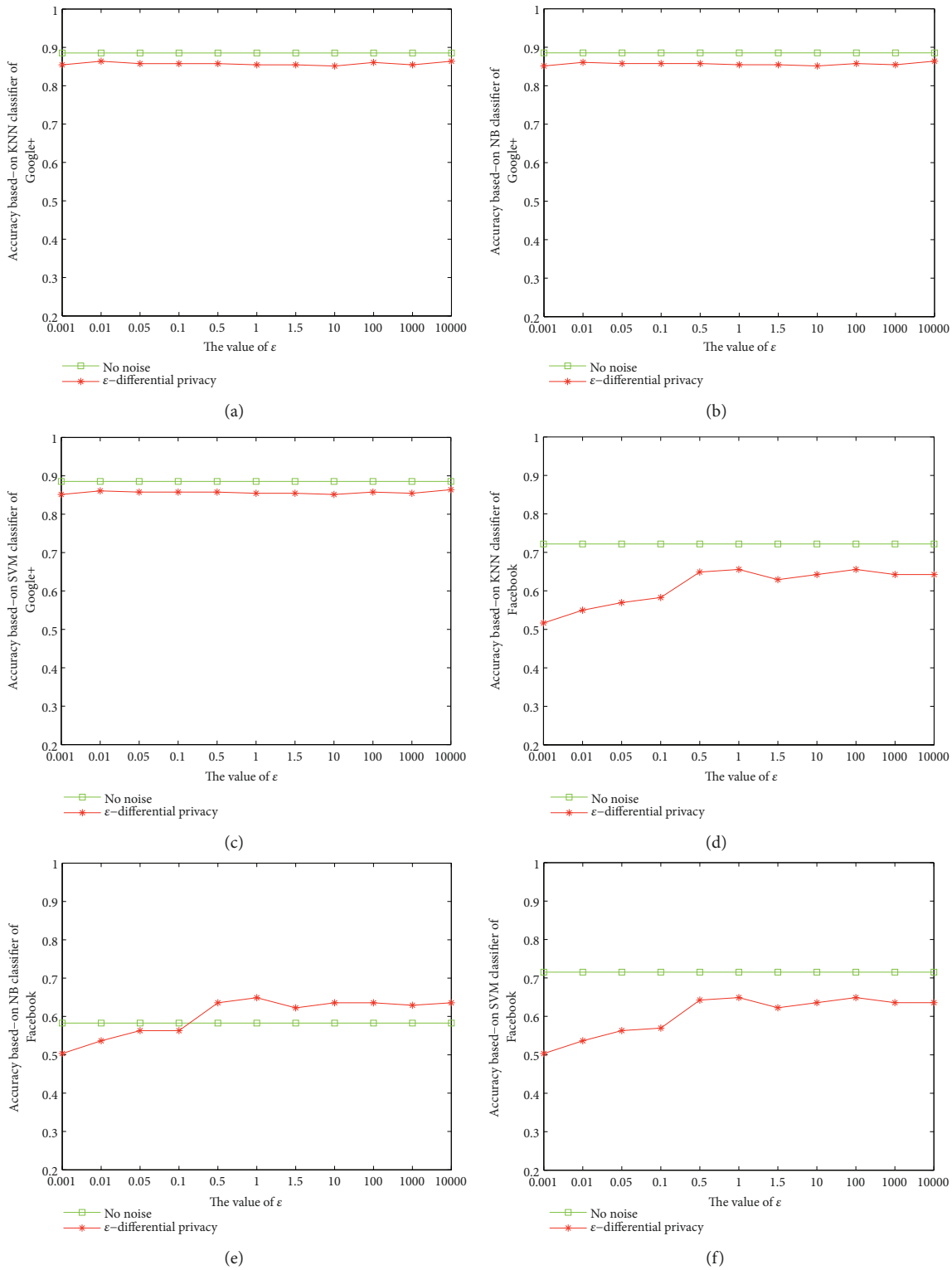

FIGURE 6: The accuracy with noise based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

the second attribute category the utility changed. But based on the definition and set value of utility, this change did not influence the removal. When deleting the fifteenth attribute category, for Figures 5(d) and 5(f) which show huge change, the change range is outside the set value; therefore stop removing the rest of attribute categories.

6.4. Comparison with Noise. Figures 6(a)–6(f) show the predicted results after adding noise. Figures 7(a)–7(f) show

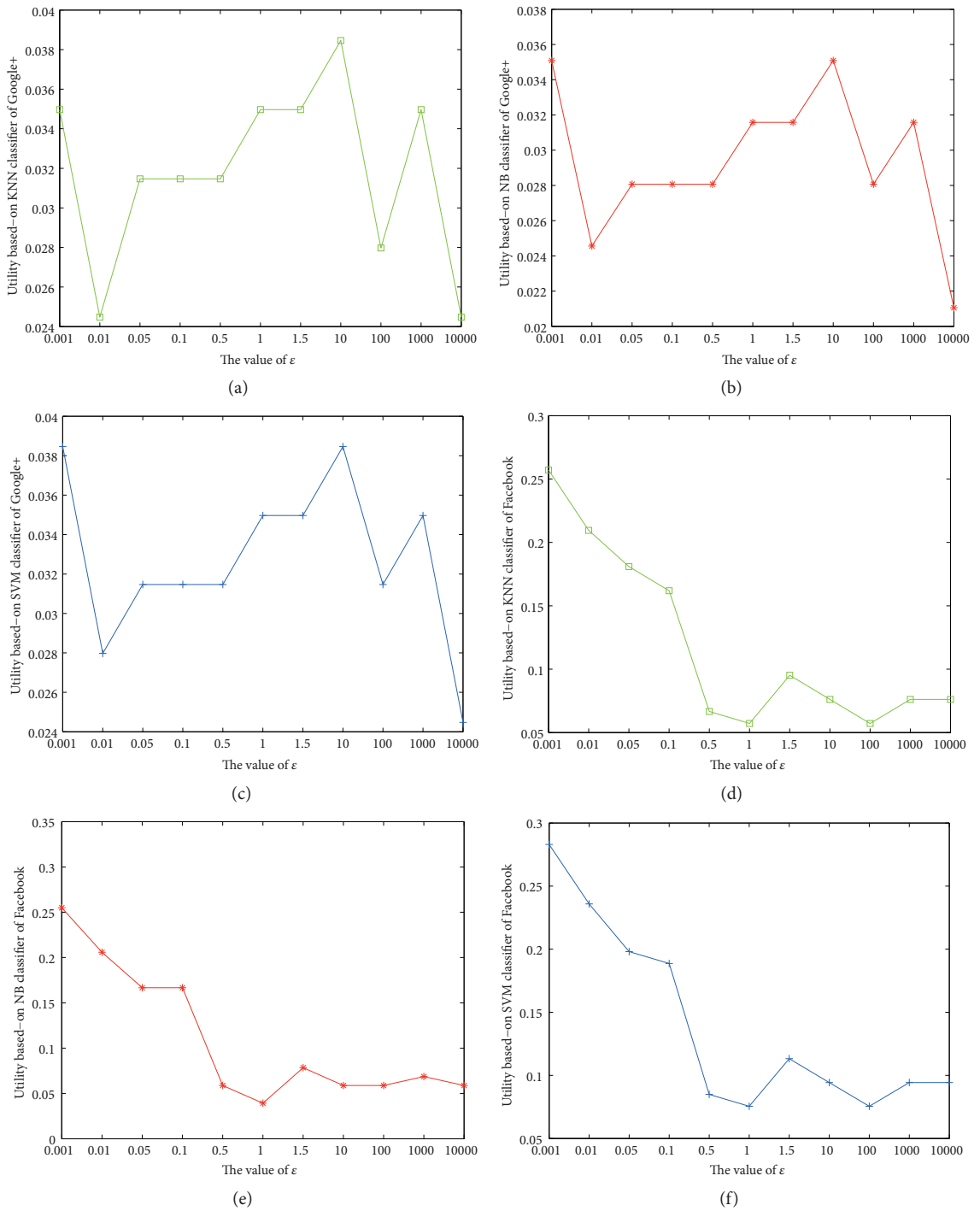

FIGURE 7: The utility with noise based on Google+ dataset and Facebook dataset. Figures (a)-(c) are the results of Google+ dataset. Figures (d)-(f) are the results of Facebook dataset.

the utility after adding noise. The first three graphs in Figures 6 and 7 are based on Google+ dataset. It can be seen from the figures that the noise has certain influence on the prediction and the utility is reduced to a certain extent. However, due to the characteristics of the dataset itself, the effect is not significant. Because every attribute in Google dataset has a very large attribute value, the attribute value of a single attribute remains large even after many attributes have been deleted. Therefore, even if the maximum noise is added, the results are relatively stable.

The last three graphs in Figures 6 and 7 are based on Facebook dataset. As can be seen from the figure, noise has a greater impact on prediction, although the reduced utility is within a given range. However, it can be seen that the prediction results are in accordance with differential privacy. When ϵ increases, the accuracy of the prediction

also increases, and the result is close to that without adding noise.

The reason why Facebook's results are more obvious than Google's is that Facebook's attribute types are diverse and each attribute type has less value, so the effect of adding noise is obvious. However, in general, it still conforms to the protection of differential privacy, and the reduced utility is within the range, reaching the balance between privacy protection and utility.

7. Conclusions

This paper proposes a CDAI recommendation method based on community detection and user attributes. The experiment is based on real social network data. It is found that this method can effectively improve the accuracy of recommendation compared with traditional classification methods. At the same time, in order to prevent the published recommendation content from being used by the attacker to push back users' privacy information, this paper also proposes a novel privacy protection method. First, this method manipulates on nodes and links. Combining with differential privacy and publishing the final recommendation results, this method will lose a few of utility. But it can protect users' privacy. This paper also has some limitations when facing attacks from attackers with complete background knowledge. This is the author's future work.

Data Availability

The data used to support the findings is generally unavailable due to public releasability constraints. However, please contact the corresponding author for special release consideration.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partly supported by the NSF under grant No. 1252292, No. 1829674, No. 1741277, No. 1704287 and the National Natural Science Foundation of China under Grant No. 61632010, No. 61502116, No. U1509216, No. 61370217, No. 61472096, No. 61872105.

References

[1] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016*, pp. 191–198, September 2016.

[2] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 1309–1315, February 2017.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[5] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: how accuracy metrics have hurt recommender systems," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI EA '06)*, pp. 1097–1101, Montreal, Canada, April 2006.

[6] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys '11)*, pp. 109–116, Chicago, Ill, USA, October 2011.

[7] K. Lewis, M. Gonzalez, and J. Kaufman, "Social selection and peer influence in an online social network," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 109, no. 1, pp. 68–72, 2012.

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

[9] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*, pp. 252–260, Vancouver, Canada, December 2007.

[10] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, pp. 203–210, ACM, Boston, MA, USA, July 2009.

[11] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM'08*, pp. 931–940, USA, October 2008.

[12] B. Yang, Y. Lei, J. Liu, and W. Li, "Social Collaborative Filtering by Trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[13] "Spear phishing attacks".

[14] P. Gupta, S. Gottipati, J. Jiang, and D. Gao, "Your love is public now: Questioning the use of personal information in authentication," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013*, pp. 49–59, May 2013.

[15] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "BlurMe: Inferring and obfuscating user gender based on ratings," in *Proceedings of the 6th ACM Conference on Recommender Systems, RecSys 2012*, pp. 195–202, September 2012.

[16] E. Zheleva and L. Getoor, "To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles," in *Proceedings of the 18th International World Wide Web Conference, WWW 2009*, pp. 531–540, April 2009.

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

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

[19] N. Z. Gong, A. Talwalkar, L. MacKey et al., "Joint link prediction and attribute inference using a social-attribute network," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 2, article 27, 2014.

[20] Neil. Zhenqiang Gong and Bin. Liu, "You are who you know and how you behave: Attribute inference attacks via users social friends and behaviors," in *Proceedings of the USENIX Security Symposium*, pp. 979–995, 2016.

[21] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, "Attriinfer: Inferring user attributes in online social networks using markov random fields," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1561–1569, Perth, Australia, April 2017.

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

[23] Y. Lin, X. Wang, F. Hao, L. Wang, L. Zhang, and R. Zhao, "An on-demand coverage based self-deployment algorithm for big data perception in mobile sensing networks," *Future Generation Computer Systems*, vol. 82, pp. 220–234, 2018.

[24] 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. PP, no. 99, pp. 1-1, 2017.

[25] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[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] 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. 99, pp. 1-1, 2017.

[28] M. Humbert, T. Studer, M. Grossglauser, and J.-P. Hubaux, "Nowhere to hide: Navigating around privacy in online social networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8134, pp. 682–699, 2013.

[29] D. Jurgens, T. Finethy, J. McCorriston, Y. T. Xu, and D. Ruths, "Geolocation prediction in twitter using social networks: A critical analysis and review of current practice," in *Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015*, pp. 188–197, May 2015.

[30] A. McCallum, K. Nigam et al., "A comparison of event models for naive bayes text classification," in *Proceedings of the AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41–48, Citeseer.

[31] 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*, 2017.

[32] B. Carminati, E. Ferrari, and A. Perego, "Security and privacy in social networks," in *Social Computing: Concepts, Methodologies, Tools, and Applications*, pp. 1706–1717, IGI Global, 2010.

[33] L. Sweeney, "*k*-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[34] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 102–114, 2008.

[35] N. Vuokko and E. Terzi, "Reconstructing randomized social networks," in *Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010*, pp. 49–59, May 2010.

[36] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4–7, 2006. Proceedings*, vol. 3876 of *Lecture Notes in Computer Science*, pp. 265–284, Springer, Berlin, Germany, 2006.

[37] F. McSherry, "Privacy integrated queries: An extensible platform for privacy-preserving data analysis," in *Proceedings of the International Conference on Management of Data and 28th Symposium on Principles of Database Systems, SIGMOD-PODS'09*, pp. 19–30, July 2009.

[38] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai, and L. Xiong, "Publishing setvalued data via differential privacy," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 1087–1098, 2011.

[39] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, Article ID 7576307, 12 pages, 2017.

[40] 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*, pp. 1–17, 2018.

[41] X. Zheng, G. Luo, and Z. Cai, "A Fair Mechanism for Private Data Publication in Online Social Networks," *IEEE Transactions on Network Science and Engineering*, 2018.

[42] X. Zheng, Z. Cai, and Y. Li, "Data Linkage in Smart Internet of Things Systems: A Consideration from a Privacy Perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[43] 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*, vol. 32, no. 4, pp. 8–14, 2018.

[44] Z. Cai and X. Zheng, "A Private and Efficient Mechanism for Data Uploading in Smart Cyber-Physical Systems," *IEEE Transactions on Network Science and Engineering*, pp. 1-1, 2018.

[45] 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, 2017.

[46] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS '15)*, pp. 205–214, July 2015.

[47] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

[48] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 2, Article ID 026113, 2004.

[49] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS '07)*, pp. 94–103, Providence, RI, USA, October 2007.

[50] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, Article ID 066133, 2004.

[51] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE '06)*, pp. 24-24, Atlanta, Ga, USA, April 2006.

[52] X. Chen, J. Liu, X. Feng, and X. Zhao, "Differential privacy synthetic data set publishing algorithm based on naive bayes," *Computer Science*, vol. 42, pp. 236–238, 2015.