Satisfying Privacy Requirements Before Data Anonymization

Xiaoxun Sun[1], Hua Wang[2], Jiuyong Li[3] and Yanchun Zhang[4]

[1]*Australian Council for Educational Research, Australia*
[2]*Department of Mathematics & Computing, University of Southern Queensland, Australia*
[3]*School of Computer and Information Science, University of South Australia, Australia*
[4]*School of Engineering and Science, Victoria University, Australia*
Email: sun@acer.edu.au; Hua.Wang@usq.edu.au; Jiuyong.Li@unisa.edu.au; Yanchun.Zhang@vu.edu.au

In this paper, we study a problem of protecting privacy of individuals in large public survey rating data. We propose a novel (k, ϵ, l)-anonymity model to protect privacy in large survey rating data, in which each survey record is required to be similar with at least $k-1$ others based on the non-sensitive ratings, where the similarity is controlled by ϵ, and the standard deviation of sensitive ratings is at least l. We study an interesting yet nontrivial satisfaction problem of the proposed model, which is to decide whether a survey rating data set satisfies the privacy requirements given by the user. For this problem, we investigate its inherent properties theoretically, and devise a novel slicing technique to solve it. We analyze the computation complexity of the proposed slicing technique, and conduct extensive experiments on two real-life data sets, and the results show that the slicing technique is fast and scalable with data size and much more efficient in terms of execution time and space overhead than the heuristic pairwise method.

1. INTRODUCTION

The problem of privacy-preserving data publishing has received a lot of attention in recent years. Privacy preservation on relational data has been studied extensively. A major category of privacy attacks on relational data is to re-identify individuals by joining a published table containing sensitive information with some external tables. Most of existing work can be formulated in the following context: several organizations, such as hospitals, publish detailed data (called microdata) about individuals (e.g. medical records) for research or statistical purposes [1, 2, 3, 4].

Privacy risks of publishing microdata are well-known. Famous attacks include de-anonymisation of the Massachusetts hospital discharge database by joining it with a public voter database [1] and privacy breaches caused by AOL search data [5]. Even if identifiers such as names and social security numbers have been removed, the adversary can use linking [1], homogeneity and background attacks [2] to re-identify individual data records or sensitive information of individuals. To overcome the re-identification attacks, k-anonymity was proposed [1, 6, 7, 8]. Specifically, a data set is said to be k-anonymous ($k \geq 1$) if, on the quasi-identifier (QID) attributes (that is, the maximal set of join attributes to re-identify individual records), each record is identical with at least $k-1$ other records. The larger the value of k, the better the privacy is protected. Several algorithms are proposed to enforce this principle [9, 10, 11, 12, 13, 14, 15]. Machanavajjhala et al. [2] showed that a k-anonymous table may lack of diversity in the sensitive attributes. To overcome this weakness, they propose the l-diversity [2]. However, even l-diversity is insufficient to prevent attribute disclosure due to the skewness and the similarity attack. To amend this problem, t-closeness [3] was proposed to solve the attribute disclosure vulnerabilities inherent to previous models.

Recently, a new privacy concern has emerged in privacy preservation research: how to protect individuals' privacy in large survey rating data. Though several models and many algorithms have been proposed to preserve privacy in relational data (e.g., k-anonymity [1], l-diversity [2], t-closeness [3], etc.), most of the existing studies are incapable of handling rating data, since the survey rating data normally does not have a fixed set of personal identifiable attributes as relational data, and it is characterized by high dimensionality and sparseness. The survey rating data shares the similar format with transactional data. The privacy preserving research of transactional data has recently been acknowledged as an important problem in the data mining literature [16, 17]. To our best knowledge, there is no current research addressing the issue of how to efficiently determine whether the survey rating data satisfies the privacy requirement. In this paper, we propose a (k, ϵ, l)-anonymity model to protect privacy in the large survey rating data and study the *Satisfaction Problem* (Section 5) of the proposed model, which is to decide whether a survey rating data set satisfies the given privacy requirements. By utilizing the largeness and sparseness properties, we develop a novel slicing technique solving the satisfaction problem. Our extensive experiments confirm that our new slicing algorithm is fast and scalable in practical compared with the heuristic pairwise algorithm. The main contributions of the paper are summarized as follows:

(1) Propose a novel (k, ϵ, l)-anonymity model to protect individual's privacy in large survey rating data. The principle demands that each transaction be similar with $k-1$ others, where the similarity is measured by ϵ metric, and it further requires the standard deviation of the sensitive ratings be at least l. ϵ captures the protection range of each individual, whereas k is to lower an adversary's chance of beating that protection, and l reflects diversity of the sensitive ratings.

(2) Investigate the theoretical properties of (k, ϵ, l)-anonymity model. Specifically, we prove a sufficient condition of the existence of at least one (k, ϵ, l)-anonymity solution in large survey rating data, and we prove the lower and upper bound of the parameter l.

(3) Apply the flag matrix to index the rating data and devise a novel slicing technique by searching closest neighbors in large, sparse and high dimensional rating data to determine the satisfaction problem, which is to decide if the given rating data satisfies privacy requirements.

(4) Analyze the computational complexity of the slicing algorithm in a theoretical way and examine one special case when the survey rating data set follows uniform distribution.

(5) Conduct extensive experiments to show that the slicing approach is scalable, time efficient and space efficient compared with the heuristic pairwise method.

The rest of the paper is organized as follows. The motivation of the paper and its rationality are introduce in Section 2. We survey the related work in Section 3. We formally defined the (k, ϵ, l)-anonymity model and investigate its theoretical properties in Section 4. The novel slicing algorithm is presented in Section 5. The analysis of the algorithm complexity is detailed in Section 6. The extensive experiments are included in Section 7. Finally, we conclude the paper in Section 8.

2. MOTIVATION

On October 2, 2006, Netflix, the world's largest online DVD rental service, announced the $1-million Netflix Prize to improve their movie recommendation service [18]. To aid contestants, Netflix publicly released a data set containing 100,480,507 movie ratings, created by 480,189 Netflix subscribers between December 1999 and December 2005. Narayanan and Shmatikov shown in their recent work [19] that an attacker only needs a little information to identify the anonymized movie rating transaction of the individual. They re-identified Netflix movie ratings using the Internet Movie Database (IMDb) as a source of auxiliary information and successfully identified the Netflix records of known users, uncovering their political preferences and other potentially sensitive information.

We consider the privacy risk in publishing anonymous survey rating data. For example, in a life style survey, ratings to some issues are non-sensitive, such as the likeness of book "Harry Potter", movie "Star Wars" and food "Sushi". Ratings to some issues are sensitive, such as the income level and sexuality frequency. Assume that each survey participant is cautious about his/her privacy and does not reveal his/her ratings. However, it is easy to find his/her preferences on non-sensitive issues from publicly available information sources, such as personal weblog or social networks. An attacker can use these preferences to re-identify an individual in the anonymous published survey rating data and consequently find sensitive ratings of a victim.

Based on the public preferences, person's ratings on sensitive issues may be revealed in a supposedly anonymized survey rating data set. An example is given in the Table 1. In a social network, people make comments on various issues, which are not considered sensitive. Some comments can be summarized as in Table 1(b). People rate many issues in a survey. Some issues are non-sensitive while some are sensitive. We assume that people are aware of their privacy and do not reveal their ratings, either non-sensitive or sensitive ones. However, individuals in the anonymoized survey rating data are potentially identifiable based on their public comments from other sources. For example, Alice is at risk of being identified, since the attacker knows Alice's preference on issue 1 is 'excellent', by cross-checking Table 1(a) and (b), s/he will deduce that t_1 in Table 1(a) is linked to Alice, the sensitive rating on issue 4 of Alice will be disclosed. This example motivates us the following research question:

(Satisfaction Problem): Given a large survey rating data set T with the privacy requirements, how to efficiently determine whether T satisfies the given privacy requirements?

Although the satisfaction problem is easy and straightforward to be determined in the relational databases, it is nontrivial in the large survey rating data set. The research of the privacy protection initiated in the relational databases, in which several state-of-art privacy paradigms [1, 2, 3] are proposed and many greedy or heuristic algorithms [4, 11, 13, 14]

	non-sensitive			sensitive
ID	issue 1	issue 2	issue 3	issue 4
t_1	6	1	*null*	6
t_2	1	6	*null*	1
t_3	2	5	*null*	1
t_4	1	*null*	5	1
t_5	2	*null*	6	5

(a)

	non-sensitive issues		
name	issue 1	issue 2	issue 3
Alice	excellent	so bad	-
Bob	awful	top	-
Jack	bad	-	good

(b)

TABLE 1: (a) A published survey rating data set containing ratings of survey participants on both sensitive and non-sensitive issues. (b) Public comments on some non-sensitive issues of some participants of the survey. By matching the ratings on non-sensitive issues with public available preferences, t_1 is linked to Alice, and her sensitive rating is revealed.

are developed to enforce the privacy principles. In the relational database, taking k-anonymity as an example [1, 7], it requires each record be identical with at least $k-1$ others with respect to a set of quasi-identifier attributes. Given an integer k and a relational data set T, it is easy to determine if T satisfies k-anonymity requirement since the equality has the transitive property, whenever a transaction a is identical with b, and b is in turn indistinguishable with c, then a is the same as c. With this property, each transaction in T only needs to be check once and the time complexity is at most $O(n^2d)$, where n is the number of transactions in T and d is the size of the quasi-identifier attributes. So the satisfaction problem is trivial in relational data sets. While, the situation is different for the large rating data. First of all, the survey rating data normally does not have a fixed set of personal identifiable attributes as relational data. In addition, the survey rating data is characterized by high dimensionality and sparseness. The lack of a clear set of personal identifiable attributes together with its high dimensionality and sparseness make the determination of satisfaction problem challenging. Second, the defined dissimilarity distance between two transactions (ϵ-proximate) does not possess the transitive property. When a transaction a is ϵ-proximate with b, and b is ϵ-proximate with c, then usually a is not ϵ-proximate with c. Each transaction in T has to be checked for as many as n times in the extreme case, which makes it highly inefficient to determine the satisfaction problem. It calls for smarter technique to efficiently determine the satisfaction problem before anonymizaing the survey rating data. To our best knowledge, this research is the first touch of the satisfaction of privacy requirements in the survey rating data. In order to solve the *Satisfaction Problem*, in this paper, we utilize the largeness and sparseness properties to develop a novel slicing technique.

3. RELATED WORK

Privacy preserving data publishing has received considerable attention in recent years. especially in the context of relational data [2, 6, 9, 10, 11, 12, 13, 14, 20]. All these works assume a given set of attributes QID on which an individual is identified, and anonymize data records on the QID. Their main difference consist in the selected privacy model and in various approaches employed to anonymize the data. The author of [9] presents a study on the relationship between the dimensionality of QID and information loss, and concludes that, as the dimensionality of QID increases, information loss increases quickly. Transactional databases present exactly the worst case scenario for existing anonymisation approaches because of high dimension of QID. To our best knowledge, all existing solutions in the context of k-anonymity [7, 8], l-diversity [2] and t-closeness [3] assume a relational table, which typically has a low dimensional QID. As we have illustrated in Section 2, the determination of whether the relational databases satisfy the privacy requirements is easy and straightforward. However, it is non-trivia for large survey rating data characterized by high dimensionality and sparseness.

There are few previous work considering the privacy of large rating data. In collaboration with MovieLens recommendation service, [21] correlated public mentions of movies in the MovieLens discussion forum with the users' movie rating histories in the internal MovieLens data set. Recent study reveals a new type of attack on anonymized data for transactional data [19]. Movie rating data supposedly to be anonymized is re-identified by linking non-anonymized data from other source. In our recent work [22], we assumed that the survey rating data sets have violated the privacy requirements, and we discussed how to publish anonymous survey rating data by using graph modification methods. No solution exists for how to determine whether the high dimensional large survey rating databases satisfy the underlying privacy requirements.

Though we consider data publishing for data mining purposes, we assume that the data publisher has no capability or interests in data mining. Therefore, it is not realistic to expect such data publishers to perform privacy-preserving data mining on behalf of the recipient. In fact, the data may be published on the Internet without a specific recipient. For this reason, techniques for privacy-preserving data mining [23, 24, 25] cannot be applied to data publishing.

Privacy-preservation of transactional data has been acknowledged as an important problem in the data mining literature. There us a family of literature [26, 27] addressing the privacy threats caused by publishing data mining results such as frequent item sets and association rules. Existing works on topic [28, 29] focus on publishing patterns, The patterns are mined from the original data, and the resulting set of rules is sanitized to present privacy breaches. In contrast, our work addresses the privacy threats caused by publishing data for data mining. As discussed above, we do not assume that the data publisher can perform data mining tasks, and we assume that the data must be made available to the recipient. The two scenarios have different assumptions on the capability of the data publisher and the information requirement of the data recipient. The recent work on topic [16, 17] focus on high dimensional transaction data, while our focus is to validate the privacy requirements in an efficient way before the data anonymization.

This paper is loosely related to the work on anonymizing social networks [30]. A social network is a graph in which a node represents a social entity (e.g., a person) and an edge represents a relationship between the social entities. Although the data is very different from transaction data, the model of attacks is similar to ours: An attacker constructs a small subgraph connected to a target individual and then matches the subgraph to the whole social network, attempting to re-identify the target individual's node, and therefore, other unknown connection to the node. [30] demonstrates the severity of privacy threats in nowadays social networks, but does not provide a solution to prevent such attacks. In this paper, we study the *Satisfaction Problem*, which is to decide whether a survey rating data set satisfies the given privacy requirements and it is an important step before data anonymization.

In [31], the authors investigated a systematic approach for authenticating clients by three factors, namely password, smart-card and biometrics. A generic and secure framework was proposed to upgrade two-factor authentication to three-factor authentication. The conversion could not only significantly improves the information assurance at low-cost but also protects client privacy in distributed systems. The research work in [31] focus on maximizing user's privacy through authentication, while our proposed method is through database modification.

4. PROBLEM FORMALIZATION

We assume that a survey rating data set publishes people's ratings on a range of issues. In a lifestyle survey, some issues are sensitive, such as income level and sexuality frequency, while some are non-sensitive, such as the likeness of a book, a movie or a kind of food. Each survey participant is cautious about his/her privacy and does not reveal his/her ratings. However, an attacker can use the public available information to identify an individual's sensitive ratings in the supposedly anonymous survey rating data. Our objective is to design effective models to protect privacy of people's sensitive ratings in the published survey rating data.

Given a survey rating data set T, each transaction contains a set of numbers indicate the ratings on some issues. Let $(o_1, o_2, \cdots, o_p, s_1, s_2, \cdots, s_q)$ be a transaction, $o_i \in \{1 : r, null\}$, $i = 1, 2, \cdots, p$ and $s_j \in \{1 : r, null\}$, $j = 1, 2, \cdots, q$, where r is the maximum rating and $null$ indicates that a survey participant did not rate. $o_1, \cdots, o_p$ stand for non-sensitive ratings and $s_1, \cdots, s_q$ denote sensitive ratings. Each transaction belongs to a survey participant.

Although each survey participant is wary about their privacy and does not disclose his/her ratings, an attacker may find a victim's preference (not exact rating scores) by personal familiarity or by reading the victim's comments on some issues from personal Weblog or social networks. We consider that attackers know preferences of non-sensitive issues of a victim but do not know exact ratings and want to find out the victim's ratings on some sensitive issues.

4.1. Background knowledge

The auxiliary information of an attacker includes: (i) the knowledge that a victim is in the survey rating data; (ii) preferences of the victims on some non-sensitive issues. The attacker wants to find ratings on sensitive issues of the victim.

In practice, knowledge of Types (i) and (ii) can be gleaned from an external database [19]. For example, in the context of Table 1(b), an external database may be the IMDb. By examining the anonymous data set in Table 1(a), the adversary can identify a small number of candidate groups that contain the record of the victim. It will be the unfortunate scenario where there is only one record in the candidate group. For example, since t_1 is unique in Table 1(a), Alice is at risk of being identified. If the candidate group contains not only the victim but other records, an adversary may use this group to infer the sensitive value of the victim individual. For example, although it is difficult to identify whether t_2 or t_3 in Table 1(a) belongs to Bob, since both records have the same sensitive value, Bob's private information is identified.

In order to avoid such attack, we propose a two-step protection model. Our first step is to protect individual's identity. In the released data set, every transaction should be "similar" to at least to $(k-1)$ other records based on the non-sensitive ratings so that no survey participants are identifiable. For example, t_1 in Table 1(a) is unique, and based on the preference of Alice in Table 1(b), her sensitive issues can be re-identified in the supposed anonymized data set. Jack's sensitive issues, on the other hand, is much safer. Since t_4 and t_5 in Table 1(a) form a similar group based on their non-sensitive rating.

The second step is to prevent the sensitive rating from being inferred in an anonymized data set. The idea is to require that the sensitive ratings in a similar group should be diverse. For example, although t_2 and t_3 in Table 1(a) form a similar group based on their non-sensitive rating, their sensitive ratings are identical. Therefore, an attacker can immediately infer Bob's preference on the sensitive issue without identifying which transaction belongs to Bob. In contrast, Jack's preference on the sensitive issue is much safer than both Alice and Bob.

4.2. (k, ϵ, l)-anonymity

Let $T_A = \{o_{A_1}, o_{A_2}, \cdots, o_{A_p}, s_{A_1}, s_{A_2}, \cdots, s_{A_q}\}$ be the ratings for a survey participant A and $T_B = \{o_{B_1}, o_{B_2}, \cdots, o_{B_p}, s_{B_1}, s_{B_2}, \cdots, s_{B_q}\}$ be the ratings for a participant B. We define the dissimilarity between two non-sensitive ratings as follows.

$$Dis(o_{A_i}, o_{B_i}) = \begin{cases} |o_{A_i} - o_{B_i}| & \text{if } o_{A_i}, o_{B_i} \in \{1 : r\} \\ 0 & \text{if } o_{A_i} = o_{B_i} = null \\ r & \text{otherwise} \end{cases} \tag{1}$$

DEFINITION 4.1 (ϵ-PROXIMATE). *Given a survey rating data set T with a small positive number ϵ, two transactions T_A, $T_B \in T$, where $T_A = \{o_{A_1}, o_{A_2}, \cdots, o_{A_p}, s_{A_1}, s_{A_2}, \cdots, s_{A_q}\}$ and $T_B = \{o_{B_1}, o_{B_2}, \cdots, o_{B_p}, s_{B_1}, s_{B_2}, \cdots, s_{B_q}\}$. We say T_A and T_B are ϵ-proximate, if $\forall\ 1 \le i \le p$, $Dis(o_{A_i}, o_{B_i}) \le \epsilon$. We say T is ϵ-proximate, if every two transactions in T are ϵ-proximate.*

If two transactions are ϵ-proximate, the dissimilarity between their non-sensitive ratings is bounded by ϵ. In our running example, suppose $\epsilon = 1$, ratings 5 and 6 may have no difference in interpretation, so t_4 and t_5 in Table 1(a) are 1-proximate based on their non-sensitive rating. If a group of transactions are in ϵ-proximate, then the dissimilarity between each pair of their non-sensitive ratings is bounded by ϵ. For example, if $T = \{t_1, t_2, t_3\}$, then it is easy to verify that T is 5-proximate.

DEFINITION 4.2 ((k, ϵ)-ANONYMITY). *A survey rating data set T is said to be (k, ϵ)-anonymous if every transaction is ϵ-proximate with at least $(k - 1)$ other transactions. The transaction $t \in T$ with all the other transactions that ϵ-proximate with t form a (k, ϵ)-anonymous group.*

For instance, there are two (2,5)-anonymous groups in Table 1(a). The first one is formed by $\{t_1, t_2, t_3\}$ and the second one is formed by $\{t_4, t_5\}$. The idea behind this privacy principle is to make each transaction contains non-sensitive attributes are similar with other transactions in order to avoid linking to personal identity. (k, ϵ)-anonymity well preserves identity privacy. It guarantees that no individual is identifiable with the probability greater than the probability of $1/k$. Both parameters k and ϵ are intuitive and operable in real-world applications. The parameter ϵ captures the protection range of each identity, whereas the parameter k is to lower an adversary's chance of beating that protection. The larger the k and ϵ are, the better protection it will provide.

Although (k, ϵ)-anonymity privacy principle can protect people's identity, it fails to protect individuals' private information. Let us consider one (k, ϵ)-anonymous group. If the transactions of the group have the same rating on a number of sensitive issues, an attacker can know the preference on the sensitive issues of each individual without knowing which transaction belongs to whom. For example, in Table 1(a), t_2 and t_3 are in a $(2, 1)$-anonymous group, but they have the same rating on the sensitive issue, and thus Bob's private information is breaching.

This example illustrates the limitation of the (k, ϵ)-anonymity model. To mitigate the limitation, we require more diversity of sensitive ratings in the anonymous groups. In the following, we define the distance between two sensitive ratings, which leads to the metric for measuring the diversity of sensitive ratings in the anonymous groups.

First, we define dissimilarity between two sensitive rating scores as follows.

$$Dis(s_{A_i}, s_{B_i}) = \begin{cases} |s_{A_i} - s_{B_i}| & \text{if } s_{A_i}, s_{B_i} \in \{1 : r\} \\ r & \text{if } s_{A_i} = s_{B_i} = null \\ r & \text{otherwise} \end{cases} \tag{2}$$

Note that there is only one difference between dissimilarities of sensitive ratings $Dis(s_{A_i}, s_{B_j})$ and dissimilarities of non-sensitive ratings $Dis(o_{A_i}, o_{B_j})$, that is, in the definition of $Dis(o_{o_i}, o_{o_j})$, $null - null = 0$, and for the definition of $Dis(s_{A_i}, s_{B_j})$, $null - null = r$. This is because for sensitive issues, two $null$ ratings mean that an attacker will not get information from two survey participants, and hence are good for the diversity of the group.

Next, we introduce the metric to measure the diversity of sensitive ratings. For a sensitive issue s, let the vector of ratings of the group be $[s_1, s_2, \cdots, s_g]$, where $s_i \in \{1 : r, null\}$. The means of the ratings is defined as follows:

$$\bar{s} = \frac{1}{Q} \sum_{i=1}^{g} s_i$$

where Q is the number of non-$null$ values, and $s_i \pm null = s_i$. The standard deviation of the rating is then defined as:

$$SD(s) = \sqrt{\frac{1}{g} \sum_{i=1}^{g} (s_i - \bar{s})^2} \tag{3}$$

For instance in Table 1(a), for the sensitive issue 4, the means of the ratings is $(6+1+1+1+5)/5 = 2.8$ and the standard deviation of the rating is 2.23 according to Equation (3).

Definition 4.3 ((k,ϵ,l)-anonymity). *A survey rating data set is said to be (k,ϵ,l)-anonymous if and only if the standard deviation of ratings for each sensitive issue is at least l in each (k,ϵ)-anonymous group.*

Still consider Table 1(a) as an example. t_4 and t_5 is 1-proximate with the standard deviation of 2. If we set $k = 2, l = 2$, then this group satisfies (2,1,2)-anonymity requirement. The (k,ϵ,l)-anonymity requirement allows sufficient diversity of sensitive issues in T, therefore it could prevent the inference from the (k,ϵ)-anonymous groups to a sensitive issue with a high probability.

4.3. Characteristics of (k,ϵ,l)-anonymity

In this section, we investigate the properties of (k,ϵ,l)-anonymity model.

Definition 4.4. *Given a subset G of T, $neighbor(t,G)$ is the set of tuples whose non-sensitive values are ϵ-proximate with t and $|neighbor(t,G)|$ indicates its cardinality. $maxsize(G)$ is the largest size $neighbor(t,G)$ of every $t \in G$. Formally, $maxsize(G) = \max_{\forall t \in G} |neighbor(t,G)|$.*

For example, let T be the data in Table 1(a), consisting of $t_1, \cdots, t_5$, and $G = T$. Assume $\epsilon = 1$, then $|neighbor(t_1,G)| = \{t_1\}$ since no other transaction in G is 1-proximate with t_1 and $|neighbor(t_1,G)| = 1$. Similarly, $neighbor(t_2,G) = \{t_2,t_3\}$ with $|neighbor(t_2,G)| = 2$ because t_2 and t_3 are 1-proximate with t_1. $maxsize(G) = 2$, because no other transaction $t \in G$ has a $neighbor(t,G)$ higher than 2. $maxsize(G)$ has the following property:

Lemma 4.1. *Let G_1, G_2 be two partition of G and $G_1 \cup G_2 = G$. Then,*

$$\frac{maxsize(G)}{|G|} \leq max\{\frac{maxsize(G_1)}{|G_1|}, \frac{maxsize(G_2)}{|G_2|}\}$$

Proof: We first show $maxsize(G) \leq maxsize(G_1) + maxsize(G_2)$. Due to symmetry, assume $t \in G_1$, and that $maxsize(G)$ is the size of the neighbor covering set $neighbor(t,G)$ of a tuple $t \in G$. Use S_1 (S_2) to denote the set of tuples in $neighbor(t,G)$ that also belong to G_1 (G_2). Obviously, $neighbor(t,G) = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$. Let t' be the tuple in S_2 with the largest range. Notice that $S_1 \subseteq neighbor(t,G_1)$ and $S_2 \subseteq neighbor(t',G_2)$. Therefore, $maxsize(G) = |S_1| + |S_2| \leq |neighbor(t,G_1)| + |neighbor(t',G_2)| \leq maxsize(G_1) + maxsize(G_2)$.

Given any subset G of T, we define $\alpha(G) = maxsize(G)/|G|$, and $\alpha(G_1)$, $\alpha(G2)$ in the same manner. As $maxsize(G) \leq maxsize(G_1) + maxsize(G_2)$, we have $(|G_1| + |G_2|) \cdot \alpha(G) = |G_1| \cdot \alpha(G_1) + |G2| \cdot \alpha(G2)$, leading to $\frac{|G_1|}{|G_2|} \cdot (\alpha(G) - \alpha(G_1)) + \alpha(G) \leq \alpha(G_2)$. If $\alpha(G) \leq \alpha(G_1)$, lemma holds. If $\alpha(G) \geq \alpha(G_1)$, the term $\frac{|G_1|}{|G_2|} \cdot (\alpha(G) - \alpha(G_1)) > 0$; hence, $\alpha(G) \leq \alpha(G_2)$. No matter in which case, lemma holds. ■

Note that if $G = \cup_{i=1}^{n} G_i$, the result of the lemma can be extended to $\frac{maxsize(G)}{|G|} \leq \max_{i=1}^{n}\{\frac{maxsize(G_i)}{|G_i|}\}$. In our example with $\epsilon = 5$, $G_1 = \{t_1,t_2,t_3\}$ and $G_2 = \{t_4,t_5\}$. Clearly, $G_1 \cup G_2 = T$. It is easy to verify that $maxsize(G_1) = neighbor(t_2,G_1) = 2$ and $maxsize(G_2) = neighbor(t_4,G_2) = 2$. Hence, $\frac{2}{5} < \max\{\frac{2}{3}, \frac{2}{2}\} = 1$, the inequality in Lemma holds.

Theorem 4.1. *Given ϵ and a partition of $T = \cup_{i=1}^{n} G_i$, if T has at least one (k,ϵ)-anonymity solution, then $k \leq \lceil \frac{maxsize(T) \cdot |G_j|}{|T|} \rceil$, where $\frac{maxsize(G_j)}{|G_j|} = max_{i=1}^{n}\{\frac{maxsize(G_i)}{|G_i|}\}$.*

Proof: Suppose $|neighbor(t,G_j)| = maxsizeG_j$ and $k > \lceil \frac{maxsize(G) \cdot |G_j|}{|T|} \rceil$. If T has a (k,ϵ)-anonymous solution, then the possibility of t being identified is at least $\frac{1}{neighbor(t,G_j)}$, which is greater than $\frac{|T|}{maxsize(T) \cdot |G_j|}$ due to the fact that $\frac{maxsize(T)}{|T|} \leq \frac{maxsize(G_j)}{|G_j|}$. With our assumption, we get that the possibility of t being identified is greater than $\frac{1}{k}$, which contradicts with the fact that T has a (k,ϵ)-anonymous solution. ■

Theorem 4.1 provides a sufficient condition for the existence of a (k,ϵ)-anonymity solution. In our running example with $\epsilon = 1$, we already know that $maxsize(G) = 2$, then according to Theorem 4.1, if a (k,ϵ)-anonymity exists, then $k \leq \lceil \frac{2 \times 3}{5} \rceil = 2$.

Lemma 4.2. *Given $S = \{s_1, s_2, \cdots, s_n\}$ as the sensitive ratings of T. Let S_1 and S_2 be two partitions of S and $S_1 \cup S_2 = S$. Then,*

$$SD(S) \geq min\{SD(S_1), SD(S_2)\}$$

Proof: Without loss of generality, suppose $S_1 = \{s_1, s_2, \cdots, s_k\}$ and $S_2 = \{s_{k+1}, \cdots, s_n\}$ and $SD(S_1) \leq SD(S_2)$. $\bar{s} = \frac{\sum_{i=1}^{n} s_i}{n}$, $\bar{s_1} = \frac{\sum_{i=1}^{k} s_i}{n}$ and $\bar{s_2} = \frac{\sum_{i=k+1}^{n} s_i}{n}$.

Next, we show that $SD(S) > SD(S_1)$.

$$SD^2(S) - SD^2(S_1) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n} - \frac{\sum_{i=1}^{k}(x_i - \bar{x_1})^2}{k}$$

$$= \frac{1}{nk}(k\sum_{i=1}^{n}(x_i - \bar{x})^2 - n\sum_{i=1}^{k}(x_i - \bar{x_1})^2)$$

$$= \frac{1}{nk}(k\sum_{i=1}^{n}(x_i - \bar{x})^2 - k\sum_{i=1}^{k}(x_i - \bar{x_1})^2 - (n-k)\sum_{i=1}^{k}(x_i - \bar{x_1})^2)$$

Since $SD(S_1) \leq SD(S_2)$, $\frac{\sum_{i=1}^{k}(x_i - \bar{x_1})^2}{k} \leq \frac{\sum_{i=1}^{n-k}(x_i - \bar{x_2})^2}{n-k}$

$$\geq \frac{1}{nk}(k\sum_{i=1}^{n}(x_i - \bar{x})^2 - k\sum_{i=1}^{k}(x_i - \bar{x_1})^2 - k\sum_{i=k+1}^{n}(x_i - \bar{x_2})^2)$$

$$= \frac{1}{n}(\sum_{i=1}^{n}(x_i - \bar{x})^2 - \sum_{i=1}^{k}(x_i - \bar{x_1})^2 - \sum_{i=k+1}^{n}(x_i - \bar{x_2})^2)$$

$$= \frac{1}{n}(\sum_{i=1}^{k}((x_i - \bar{x})^2 - (x_i - \bar{x_1})^2) + \sum_{i=k+1}^{n}((x_i - \bar{x})^2 - (x_i - \bar{x_2})^2))$$

Since $k\bar{x_1} = \sum_{i=1}^{k} x_i$ and $(n-k)\bar{x_2} = \sum_{i=k+1}^{n} x_i$, then

$$= \frac{1}{n}(k(\bar{x_1} - \bar{x})^2 + (n-k)(\bar{x_2} - \bar{x})^2) \geq 0$$

Therefore, the lemma holds. ■

Note that if $S = \cup_{i=1}^{n} S_i$, the result of the lemma can be extended to $SD(S) \geq \min_{i=1}^{n}\{SD(S_i)\}$. In our example with $\epsilon = 5$, the ratings of the sensitive issue 4 $S = \{6, 1, 1, 1, 5\}$ are divided into two groups $S_1 = \{6, 1, 1\}$ and $S_2 = \{1, 5\}$. It is easy to verify that $SD(S) = 2.23$, $SD(S_1) = 2.35$ and $SD(S_2) = 2$. Therefore, $SD(S) > \min\{SD(S_1), SD(S_2)\}$, the inequality in Lemma holds.

COROLLARY 4.1. *Let S be the ratings of the sensitive issue of T, and be divided into n groups, $S_1, \cdots, S_n$. If $\forall i$, $SD(S_i) \geq l_0$. Then, $SD(S) \geq l_0$.*

The following theorem gives the upper bound of the parameter l in the (k, ϵ, l)-anonymity model.

THEOREM 4.2. *Let S be the set of ratings of the sensitive issue of T. Suppose S_min and S_max be the minimum and maximum ratings in S, then the maximum standard deviation of S is $\frac{(S\_max - S\_min)}{2}$.*

Proof: For the ease of description, we write S_min as a and S_max as b, we only need to prove the following inequality holds with $(a \leq c \leq b)$:

$$\sqrt{\frac{(a - \frac{a+b+c}{3})^2 + (b - \frac{a+b+c}{3})^2 + (c - \frac{a+b+c}{3})^2}{3}} \leq \frac{(b-a)}{2} \quad (5)$$

Let $f(c)$ be written as:

$$f(c) = \frac{(a - \frac{a+b+c}{3})^2 + (b - \frac{a+b+c}{3})^2 + (c - \frac{a+b+c}{3})^2}{3}$$

The graph of $f(c)$ is a parabola, and after simplifying the function, the axis of symmetry is $c = \frac{a+b}{2}$, and since $f'(x) = 6 > 0$ and $a \leq \frac{a+b}{2} \leq b$, the function has the minimum value $\frac{(b-a)^2}{6}$, then

$$\frac{(b-a)^2}{6} \leq f(c) \leq min\{f(a), f(b)\}$$

because $f(a) = f(b) = \frac{6(b-a)^2}{27}$, then

$$\frac{(b-a)^2}{6} \leq f(c) \leq \frac{6(b-a)^2}{27}$$

Due to the fact that $\frac{6(b-a)^2}{27} < \frac{(b-a)^2}{4}$, then Equation (5) holds. The proof of Theorem 4.2 completes. ■

5. SATISFYING PRIVACY REQUIREMENTS

In this section, we formulate the satisfaction problem and develop a slicing technique based on the properties discussed in Section 4.3 to determine the following *Satisfaction Problem.*

DEFINITION 5.1 (SATISFACTION PROBLEM). *Given a survey rating data set T and privacy requirements k, ϵ, l, the satisfaction problem of (k, ϵ, l)-anonymity is to decide whether T satisfies the k, ϵ, l privacy requirements.*

The satisfaction problem is to determine whether the user's given privacy requirement is satisfied by the given data set. It is a very important step before anonymizing the survey rating data. If the data set has already met the requirements, it is not necessary to make any modifications before publishing. As follows, we propose a novel slice technique to solve the satisfaction problem.

5.1. Satisfaction algorithms

Recall that we are given a survey rating data set consisting of a set of transactions $T = \{t_1, t_2, \cdots, t_n\}$, $|T| = n$. Each transaction $t_i \in T$ contains issues from an issue set $I = \{i_1, i_2, \cdots, i_m\}$, $|I| = m$. Consider that both n (the number of survey participants) and m (the number of issues) may be very large. For example, a million of users rate thousands of movies. The efficient identification of the violation to privacy requirement is nontrivial. Firstly, the dissimilarity matrix is very big if we try to compute all pairwise distances. The time complexity is $O(n^2m)$. Secondly, the data matrix may not fit in the memory. An algorithm needs to read data from disk frequently.

We plan to utilize the sparseness of the survey rating data set to speed up the algorithm. The data set is very spare if we consider *null* values as empty. Here, we define a binary flag matrix F to record if there is a rating or not for each issue (column).

$$F_{ij} = \begin{cases} 1 & \text{if } i_j \in t_i \\ 0 & \text{if } i_j \notin t_i \end{cases}$$

	non-sensitive			sensitive
ID	issue 1	issue 2	issue 3	issue 4
t_1	3	6	*null*	6
t_2	2	5	*null*	1
t_3	4	7	*null*	4
t_4	5	6	*null*	1
t_5	1	*null*	5	1
t_6	2	*null*	6	5

TABLE 2: Sample rating data

For instance, the flag matrix associated with Table 1(a) is:

$$\mathbf{F} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \tag{6}$$

in which, each row corresponds to survey participants and each column corresponds to non-sensitive issues. If we want to find the transactions that are ϵ-proximate with t_1, intuitively, we need not to compute the dissimilarity between t_1 and t_4, and between t_1 and t_5 since both t_4 and t_5 do not rate issue 2. Based on the sparseness property, it could significant reduce the amount of the pairwise dissimilarity computation.

DEFINITION 5.2 (HAMMING DISTANCE). *[32] Hamming distance between two vectors in the flag matrix of equal length is the number of positions for which the corresponding symbols are different. We denote the Hamming distance between two vectors v_1 and v_2 as $H(v_1, v_2)$.*

In other words, Hamming distance measures the minimum number of substitutions required to change one into the other, or the number of errors that transformed one vector into the other. For example, if $v_1 = (1, 1, 0)$ and $v_2 = (1, 0, 1)$, then $H(v_1, v_2) = 2$. If the Hamming distance between two vectors are zero, then these two vectors are identical.

DEFINITION 5.3 (HAMMING GROUP). *Hamming group is the set of vectors, in which the Hamming distance between any two vectors of the flag matrix is zero. The maximal Hamming group is a Hamming group that is not a subset of any other Hamming group.*

For example, there are two maximal Hamming groups in the flag matrix (6), which are made of vectors $\{(1, 1, 0), (1, 1, 0), (1, 1, 0)\}$ and $\{(1, 0, 1), (1, 0, 1)\}$ and they are actually groups of $\{t_1, t_2, t_3\}$ and $\{t_4, t_5\}$ in T.

Now we focus on the how to group T in order to fulfill the privacy requirement. As we has explained in the previous example that the first three transactions form a maximal Hamming group and the last two transactions form the other one, which inspires us for the idea of the first step of the algorithm. It works as follows: firstly, we find out all the maximal Hamming groups, namely $H_1, \cdots, H_k$. For each Hamming group H_i, $1 \le i \le k$, we test for the privacy requirement. In our running example, if given $\epsilon = 5$, the two maximal Hamming groups made of $\{t_1, t_2, t_3\}$ and $\{t_4, t_5\}$ are already satisfying with the privacy requirement. However, if having a look at Table 2, the flag matrix of which is

$$\mathbf{F}' = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \tag{7}$$

The maximal Hamming groups are $H_1 = \{t_1, t_2, t_3, t_4\}$ and $H_2 = \{t_5, t_6\}$. If given $\epsilon = 1$, H_2 has already met the requirement, but H_1 does not. In this case, smarter technique is required to further process the group H_1. Here, we adopt a greedy slicing technique to solve challenge.

5.2. Search by slicing

Our slicing algorithm is based on the projection search paradigm first used by Friedman [33]. Friedman's simple technique works as follows. In the preprocessing step, d dimensional training points are ordered in d different ways by individually sorting each of their coordinates. Each of the d sorted coordinates arrays can be thought of as a 1-D axis with the entire d dimensional space projected onto it. Given a point q, the nearest neighbor is found as follows. A small ϵ is subtracted from and added to each of q's coordinates to obtain two values. Two binary search searches are performed on each of the sorted arrays to locate the positions of both values. An axis with the minimum number of points in between the position is chosen. Finally, points in between the positions on the chosen axis are exhaustively searched to obtain the closest point. The complexity of is $O(nd\epsilon)$ and is clearly inefficient in high d.

5.2.1. *To determine k and l when given ϵ*

Our slicing technique is proposed to efficiently search for the neighbor within distance ϵ in high dimension. As we shall see, the complexity of the proposed algorithm grows very slowly with dimension for small ϵ. We illustrate the proposed slicing technique using a simple example in 3-D space, as shown in Figure 1. Given $t = (t_1, t_2, t_3) \in T$, our goal is to slice out a set of transactions T ($t \in T$) that are ϵ-proximate. Our approach is first to find the ϵ-proximate of t, which is the set of transactions that lie inside a cube C_t of side 2ϵ centered at t. Since ϵ is typically small, the number of points inside the cube is also small. The ϵ-proximate of C'_t can then be found by an exhaustive comparison within the ϵ-proximate of t. If there are no transactions

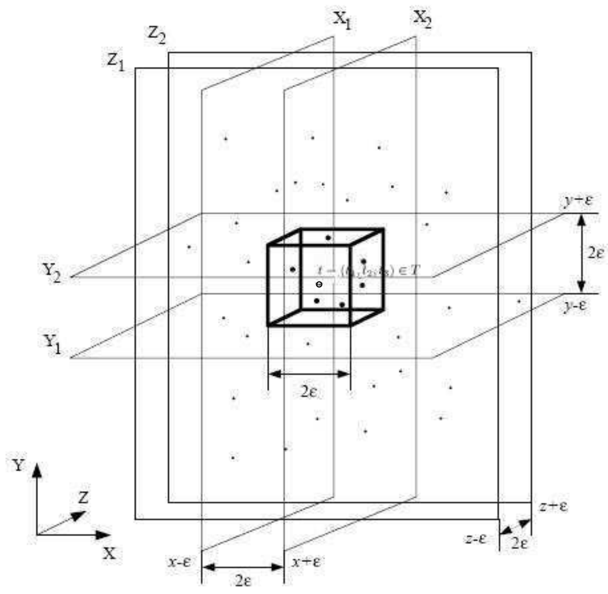


FIGURE 1: The slicing technique finds a set of transactions C_t inside a cube of size 2ϵ within the ϵ-proximate of t. The ϵ-proximate of the set C_t can then be found by an exhaustive search in the cube.

inside the cube C_t, we know that the ϵ-proximate of t is empty, so as the ϵ-proximate of the set C'_t.

The transactions within the cube can be found as follows. First we find the transactions that are sandwiched between a pair of parallel planes X_1, X_2 (See Figure 1) and add them to a *candidate set*. The planes are perpendicular to the first axis of coordinate frame and are located on either side of the transaction t at a distance of ϵ. Next, we trim the candidate set by disregarding transactions that are not also sandwiched between the parallel pair of Y_1 and Y_2, that are perpendicular to X_1 and X_2, again located on either side of t at a distance of ϵ. This procedure is repeated for Z_1 and Z_2 at the end of which, the candidate set contains only transactions within the cube of size 2ϵ centered at t. $Slicing(\epsilon, T, t_0)$ (**Algorithm 1**) describes how to find the ϵ-proximate of the set C_{t_0} with $t_0 \in C_{t_0}$.

Since the number of transactions in the final ϵ-proximate is typically small, the cost of the exhaustive comparison is negligible. The major computational cost in the slicing process occurs therefore in constructing and trimming the candidate set.

Suppose the set C'_t ($t \in C'_t$) is finally ϵ-proximate. We repeat the process for another transaction on the set $T \setminus C'_t$. Finally, there comes to two situations. One is that all transactions are grouped into anonymous groups with each group having at least two transactions. The other situation is that for some $t' \in T$ there is no ϵ-proximate for it, in this case, we let t' form an (k, ϵ)-anonymous group by itself.

Algorithm 1: $Slicing(\epsilon, T, t_0)(\ )$

$Candidate \leftarrow \{t_0\}; S \leftarrow \emptyset$
/ * To slice out the cube, ϵ-proximate of t_0 * /
for $j \leftarrow 1$ **to** n
do if $|t_j - t_0| < \epsilon$
  then $Candidate \leftarrow Candidate \cup \{t_j\}$
    $S \leftarrow S \cup \{j\}$
/ * To trim the ϵ-proximate of t_0 * /
$PCan \leftarrow Candidate$
for $i \leftarrow 1$ **to** $|S|$
do for $j \leftarrow 1$ **to** $|S|$
  do if $|t_{S(i)} - t_{S(j)}| > \epsilon$
    then $PCan \leftarrow PCan \setminus \{t_{S(i)}\}$
return $PCan$

We use the sample rating data in Table 2 to illustrate how the slicing algorithm works. If we want to find a (k, ϵ)-anonymity solution with $\epsilon = 1$. The first step is to slice out the transactions that are ϵ-proximate with the first transaction t_1, and we use C_t to denote the set of transactions, where $C_t = \{t_1, t_2, t_3\}$. The next step is to trim C_t to make it ϵ-proximate, and the method is to verify if the distance between any two elements in C_t is bounded by ϵ. In this example, dissimilarity between t_2 and t_3 is greater than ϵ, then we take one out of C_t (we choose t_3 here), and after that, we could obtain the new set $C'_t = C_t \setminus \{t_3\} = \{t_1, t_2\}$, which is already ϵ-proximate. Repeat this process on $T' = T \setminus C'_t$, and finally we can find one $(2, 1)$-anonymity solution consisting of three anonymous groups $\{\{t_1, t_2\}, \{t_3, t_4\}, \{t_5, t_6\}\}$. Further, if we consider sensitive issues, actually, there is enough diversity in each (k, ϵ)-anonymous group with $l = 1.5$. So for this example, it satisfies $(2, 1, 1.5)$-anonymity requirement.

Further, if we partition T into $\{G_1, G_2\}$, where $G_1 = \{t_1, t_2, t_3, t_4\}$ and $G_2 = \{t_5, t_6\}$, we get $maxsize(T) = 3$ and $maxsize(G_1) = 3$ with $\epsilon = 1$. So according to Theorem 4.1, $k \leq \lceil \frac{maxsize(T) \cdot |G_1|}{|T|} \rceil$, which is $\frac{3 \times 4}{6} = 2$. This example also verifies Theorem 4.1.

5.2.2. *To determine ϵ and l when given k*

In this section, we discuss the situation when k is known, and how to find out a solution that satisfies (k, ϵ, l)-anonymity principle with ϵ as smaller as possible. To solve this problem, we combine the slicing technique and binary search in our algorithm.

Binary search is a technique for locating a particular value in a sorted list of values. It makes progressively better guesses, and closes in on the sought value by selecting the middle element in the span (which, because the list is in sorted order, is the median value), comparing its value to the target value, and determining if the selected value is greater than, less than, or equal to the target value. A guess that turns out to be too high becomes the new upper bound of the span, and a guess that is too low becomes the new lower bound. Pursuing this strategy iteratively, it narrows the search by a factor of two each time, and finds the target value or else determines that it is not in the list at all.

Our algorithm starts from the upper bound $\epsilon = r$ (r is the maximum rating in T) and begins with transaction $t_1 \in T$, at the initial stage, all transactions fall into one (k, ϵ)-anonymous group. We further our search by setting ϵ to $\frac{r}{2}$, which is a middle element between 0 and r. For this new ϵ, we need to find out all transactions that are $\frac{r}{2}$-proximate by running slicing technique discussed before. Our objective is to determine whether or not the set of transactions that is $\frac{r}{2}$-proximate neighborhood has the capacity greater than the given k. If yes, we set new upper bound to $\frac{r}{2}$ and search among the interval $[0, \frac{r}{2}]$. Continue this process for interval $[0, \frac{r}{2}]$ with middle element $\frac{r}{4}$. Else, we set the new lower bound to $\frac{r}{2}$ and continue searching in $[\frac{r}{2}, r]$ with middle element $\frac{3r}{4}$. Repeat this until reaching the *termination condition*. We terminate searching if for the interval [upper bound, lower bound], $|$upper bound $-$ lower bound$| < 1$. Finally, ϵ returns to the unique integer in the interval [upper bound, lower bound].

Consider our running example with $k = 2$. We begin with $\epsilon = 6$ and return to an anonymous solution with all transactions in one group. Next we try $\epsilon = 3$ and the interval [0,6] is partitioned into [0,3] and [3,6]. By using the slicing algorithm, it returns that there is a set of transactions which is 3-proximate, and its capacity is less than 2. Then, we move to the interval [3,6] and try $\epsilon = 4.5$, the ϵ is still not large enough. We finish the search until we get that ϵ is in the interval [4.5, 5.25], and since $|5.25 - 4.5| < 1$, the search terminates and ϵ returns to 5. Finally we can find one $(2, 5, 2)$-anonymous solution consisting of two anonymous groups $\{\{t_1, t_2, t_3\}, \{t_4, t_5\}\}$.

5.2.3. *To determine k and ϵ when given l*

In this section, we discuss the situation when l is given, and how to find a solution satisfying (k, ϵ, l)-anonymity principle with ϵ as small as possible. Let S be the ratings of the sensitive issue of T, and $SD(S) = l_0$ be the standard deviation computed by Equation (3).

Case 1: When $l > l_0$. In this case, suppose there exists one solution that satisfies both principles. Let T be divided into n groups, and in each group, the similarity of any two transactions are bounded by ϵ, and the number of transactions in each group is at least k, and the standard deviation of the sensitive ratings in each group is at least l. According to Corollary 4.1, the standard deviation of the sensitive ratings of T $SD(S)$ is at least l as well, which makes $SD(S) > l_0$, and this is a contradiction with $SD(S) = l_0$. Hence, if $l > l_0$, there is no required solution.

Case 2: When $l \leq l_0$. The algorithm starts from $\epsilon = r$, and at this initial stage, all transactions fall into one (k, ϵ, l)-anonymous group. Next, we continue our search by setting ϵ to $\frac{r}{2}$, which is a middle element between 0 and r. For this new ϵ, we need to verify if the standard deviation of the sensitive ratings in each group formed by this new ϵ is at least l. If yes, we set new upper bound to $\frac{r}{2}$ and search among the interval $[0, \frac{r}{2}]$ and continue to test for the middle element $\frac{r}{4}$. Else, we set the new lower bound to $\frac{r}{2}$ and continue searching in $[\frac{r}{2}, r]$ by testing the middle element $\frac{3r}{4}$. Repeat this until reaching the *termination condition*. We terminate searching if there exists an ϵ in the interval [upper bound, lower bound] with $|$upper bound $-$ lower bound$| < 1$ and the sensitive ratings in each group formed by this ϵ is at least l. Finally, ϵ returns to the unique integer in the interval [upper bound, lower bound].

Consider the example in Table 2 with $l = 2$. The standard deviation of the sensitive ratings of T is 2.1. Since $l < 2.1$, then there exists a solution that meets the privacy principle. We begin with $\epsilon = 6$, which returns to a solution containing all transactions in one group. Obviously, it meets both principles. Next we try $\epsilon = 3$ and the interval [0,6] is partitioned into [0,3] and [3,6]. The (k, ϵ)-anonymous groups formed when $\epsilon = 3$ are $\{t_1, t_2, t_3, t_4\}$ and $\{t_5, t_6\}$. We further verify the standard deviation of sensitive ratings in both group, and both are greater than 2. It means when $\epsilon = 3$, there exists a solution that satisfies (2,3,2)-anonymity. In order to find the solution with smallest ϵ, we continue our search in the interval [0,3] and try the middle value $\epsilon = 1.5$. It returns to three groups $\{t_1, t_2\}$, $\{t_3, t_4\}$ and $\{t_5, t_6\}$, however, the standard deviation of the sensitive ratings of the second group is $1.5 < l$. Next, we continue for search in [1.5, 3] and still could not meet the (k, ϵ, l)-anonymity requirement. We finish the search until we get that ϵ is in the interval [2.375, 3], and since $|3 - 2.375| < 1$, the search terminates and ϵ returns to 3. Finally we can find one solution that meets (2,3,2)-anonymity principle, and it consists of two anonymous groups $\{t_1, t_2, t_3, t_4\}$ and $\{t_5, t_6\}$.

5.3. Pruning and adjusting

In this section, we discuss the refine technique used in order to obtain the accurate (k, ϵ)-anonymous groups. Without the refine process, some solutions are possibly missing due to the greedy choice of ϵ-proximate. Let us take Table 2 as an example. If we set $\epsilon = 2$ and try to find the (k, ϵ)-anonymous groups. The resulting (k, ϵ)-anonymous groups are made of $\{t_1, t_3, t_4\}, \{t_2\}, \{t_5, t_6\}$, which is not the desired solution, since t_2 is unique in the second group. However, with $\epsilon = 2$, we could easily find that the desired (k, ϵ)-anonymous groups consist of $\{t_1, t_2\}, \{t_3, t_4\}, \{t_5, t_6\}$ in Table 2. From this fact, we see that some solutions might be missed from our slicing process, and it is necessary to develop the appropriate method to retrieve the “missing” ones. The reason for the missing solutions is because of the greedy choice of ϵ-proximate. In every iteration of the algorithm, for the transaction t_i, we slice out all the transactions that are

ϵ-proximate with t_i and delete them from the original data set and continue the slicing process for the next transaction t_j. During this process, it might happen that there is no other transactions that are ϵ-proximate with t_j, but there might be some t_k which is ϵ-proximate with both t_i and t_j. Since the set that is ϵ-proximate was deleted in order to continue the next search, some inaccurate groupings occur.

In order to fix this problem, our idea is to re-check each group that is found by the algorithms to see if the singleton groups can borrow some transactions from large groups (refer to the group having more than three transactions). If there is some transaction t_i in the large group is ϵ-proximate with t_j in the singleton group, then we move the transaction t_i to the singleton group containing t_j. Repeat this until the following conditions are satisfied.

Case 1: No singleton group exists in the pruned (k, ϵ)-anonymous groups. In this case, we retrieve the missing solutions. For example, if we set $\epsilon = 2$ in Table 2 and try to find out the (k, ϵ)-anonymous groups. By using the slicing algorithm, three anonymous groups $\{t_1, t_3, t_4\}, \{t_2\}, \{t_5, t_6\}$ are found. Since there is a singleton, the pruning process is triggered, which happens between the large group $\{t_1, t_3, t_4\}$ and the singleton group $\{t_2\}$. Because $Dis|t_1 - t_2| < \epsilon = 2$, then transaction t_1 is moved from the large group $\{t_1, t_3, t_4\}$ to the singleton group $\{t_2\}$, and two adjusted groups $\{t_3, t_4\}$ and $\{t_1, t_2\}$ are formed after the moving.

Case 2: There still exist some singleton groups. In this case, we say there is no solution for this given ϵ. In order to find the solution, it is necessary to enlarge the value of ϵ.

6. ALGORITHM COMPLEXITY

In this section, we attempt to analyze the computational complexity of our proposed slicing algorithm. Recall that our data set consisting of a set of survey records $T = \{t_1, t_2, \cdots, t_n\}$, $|T| = n$. Each transaction $t_i \in T$ contains issues from $I = \{i_1, i_2, \cdots, i_m\}$, $|I| = m$. The major computational cost is in the process of candidate construction and trimming. The number of transactions initially added to the candidate list not only depends on ϵ, but also on the location and distribution of the transaction. Hence, to facilitate analysis, we assume uniformly distributed transaction set. In the following, we denote random variables by uppercase letter, for instance, X. Vector x is in the form of $\vec{x}$. Suffixes are used to denote individual elements of vectors, for instance, x_k is the k^{th} element of vector $\vec{x}$.

If we need to find the transactions that are ϵ-proximate with $\vec{t} \in T$, Figure 2 shows the transaction t and other $n-1$ transactions in 2-D drawn from a known distribution. Recall that the candidate set is initialized with transactions sandwiched between a hyperplane pair in the first dimension, or more generally, in the

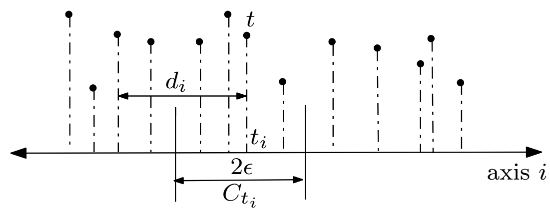


FIGURE 2: The projection of transactions to one dimension of the search space and the number of transactions inside C is given by binomial distribution.

i^{th} dimension. This corresponds to the transactions fall into area C_{t_i} in Figure 2, where the entire transaction set and $\vec{t}$ are projected to i^{th} coordinate axis. The boundaries of C_{t_i} are where the hyperplanes intersect the axis i, at $t_i - \epsilon$ and $t_i + \epsilon$. Let M_i be the number of transactions in C_{t_i}. In order to determine the average number of transactions added to the candidate set, we must compute $E[M_i]$. Let Z_i be the dissimilarity between t_i and any other transaction in the candidate set and denote P_i to be the possibility that any projected transaction is ϵ-proximate with t_i; that is,

$$P_i = P\{-\epsilon \leq Z_i \leq \epsilon | t_i\} \tag{8}$$

and if M_i is binomial distributed, the density of M_i in term of P_i is:

$$P\{M_i = k | t_i\} = P_i^k (1 - P_i)^{n-k} \binom{n}{k} \tag{9}$$

From (9), the average number of transactions in C_{t_i}, $E[M_i|t_i]$ is determined to be:

$$E[M_i|t_i] = \sum_{k=0}^{n} kP\{M_i = k|t_i\} = nP_i \tag{10}$$

Note that $E[M_i|t_i]$ is a random variable that depends on i and the location of $\vec{t}$. If the distribution of $\vec{t}$ is known, the expected number of transactions can be computed as $E[M_i] = E[E[M_i|t_i]]$. Next, we derive an expression for the total number of transactions remaining on the candidate set as we trim through the dimensions in the sequence $1, 2, \cdots, m$. If N_k is the total number of transactions before iteration k, then

$$N_k = P_i N_{k-1} = n \prod_{j=1}^{k} P_j, N_0 = n \tag{11}$$

Let N to be the total cost of the process of constructing and trimming the candidates. For each trimming, we need to perform constant times searches and comparisons. If we assign one unit cost to each operation, then with (11)

$$N = N_1 + c \sum_{k=1}^{m-1} N_k = n(P_i + c \sum_{k=1}^{m-1} \prod_{i=1}^{k} P_i) \tag{12}$$

whose expected values is:

$$E[N|\vec{t}] = nE[P_i + c\sum_{k=1}^{m-1}\prod_{i=1}^{k} P_i] \quad (13)$$

From the equation (13), if the distribution of $\vec{t}$ and $\vec{Z}$ are known, we can compute $E[N] = E[E[N|\vec{t}]]$ in term of ϵ. Next, we shall examine one particular case: uniformly distributed transaction records.

Uniformly distributed survey rating data: We denote $\vec{X}$ a random variable for the Transaction set T. Now, we look at a special case when $\vec{X}$ is uniformly distributed. For any dimension i, we assume an independent and uniform distribution with extent h on each of its coordinates as:

$$f_{X_i}(x) = \begin{cases} 1/h & \text{if } -h/2 \leq x \leq h/2 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

By using equation (14) and the fact that $Z_i = X_i - t_i$, an expression for density of Z_i can be written as:

$$f_{Z_i|t_i}(z) = \begin{cases} 1/h & \text{if } -h/2 - t_i \leq x \leq h/2 - t_i \\ 0 & \text{otherwise} \end{cases}, \forall i$$

Then, P_i in the equation (8) can be written as:

$$P_i = P\{-\epsilon \leq Z_i \leq \epsilon | t_i\} = \int_{-\epsilon}^{\epsilon} f_{Z_i|t_i}(z)dz \leq \int_{-\epsilon}^{\epsilon} \frac{1}{h}dz \leq \frac{2\epsilon}{h} \quad (15)$$

Putting (15) into (13), we obtain the upper bound:

$$\begin{aligned} E[N] &= n(\frac{2\epsilon}{h} + c(\frac{2\epsilon}{h} + (\frac{2\epsilon}{h})^2 + \cdots + (\frac{2\epsilon}{h})^{m-1}) \\ &= n(\frac{2\epsilon}{h} + c(\frac{1-(\frac{2\epsilon}{h})^m}{1-\frac{2\epsilon}{h}} - 1)) \quad (16) \\ &= O(n\epsilon + n\frac{1-\epsilon^m}{1-\epsilon}) \end{aligned}$$

We observe that for small ϵ, $\epsilon^m \approx 0$, and (16) becomes

$$E[N] \approx O(n\epsilon + n\frac{1}{1-\epsilon}) \quad (17)$$

which is independent of dimension m and note that we have left out the cost of exhaustive comparison for ϵ-proximate neighborhood within the final hypercube. The reason is that the cost of an exhaustive comparison is dependent on the distance metric used. It is very small and can be neglected in most cases when $n \gg m$. If it needs to be considered, it can be added to the equation (17). Overall, the total cost for transaction set T is $O(n^2\epsilon + n^2\frac{1}{1-\epsilon})$, which is more efficient than the heuristic pairwise approach running in $O(n^2 m)$.

7. EXPERIMENTAL STUDY

In this section, we experimentally evaluate the efficiency of the proposed slicing algorithm. Our objectives are two-fold. First, we verify that our slice algorithm is fast and scalable for the satisfaction problem. Second, we show that the slicing technique is not only time efficient, but also space efficient compared with the heuristic pairwise algorithm.

7.1. Data sets

Our experimentation deploys two real-world databases. MovieLens[5] and Netflix data sets[6]. MovieLens data set was made available by the GroupLens Research Project at the University of Minnesota. The data set contains 100,000 ratings (5-star scale), 943 users and 1682 movies. Each user has rated at lease 20 movies. Netflix data set was released by Netflix for competition. The movie rating files contain over 100,480,507 ratings from 480,189 randomly-chosen, anonymous Netflix customers over 17 thousand movie titles. The data were collected between October, 1998 and December, 2005 and reflect the distribution of all ratings received during this period. The ratings are on a scale from 1 to 5 (integral) stars. In both data sets, a user is considered as an object while a movie is regarded as an attribute and many entries are empty since a user only rated a small number of movies. Except for rating movies, users' ratings some simple demographic information (e.g., age range) are also included. In our experiments, we treat the users' ratings on movies as non-sensitive issues and ratings on others as sensitive ones.

7.2. Efficiency

Data used for Figure 3(a) is generated by re-sampling the Movielens and Netflix data sets while varying the percentage of data from 10% to 100%. For both data sets, we evaluate the running time for the (k, ϵ, l)-anonymity model with default setting $k = 20, \epsilon = 1, l = 2$. For both testing data sets, the execution time for (k, ϵ, l)-anonymity is increasing with the increased data percentage. This is because as the percentage of data increases, the computation cost increases too. The result is expected since the overhead is increased with the more dimensions.

Next, we evaluate how the parameters affect the cost of computing. Data set used for this sets of experiments are the whole sets of MovieLens and Netflix data and we evaluate by varying the value of ϵ, k and l. With $k = 20, l = 2$, Figure 3(b) shows the computational cost as a function of ϵ, in determining (k, ϵ, l)-anonymity requirement of both data sets. Interestingly, in both data sets, as ϵ increases, the cost initially becomes lower but then increases monotonically. This phenomenon is due to a pair of contradicting factors that push up and down the running time, respectively. At the initial stage, when ϵ is small, more computation efforts are put into finding ϵ-proximate of the transaction, but less used in exhaustive search for proper ϵ-proximate neighborhood, and this explains the initial decent of overall cost. On the other hand, as ϵ grows, there are fewer possible ϵ-proximate neighborhoods, thus reducing the searching time for this part, but the number of transactions

[5] http://www.grouplens.org/taxonomy/term/14.
[6] http://www.netflixprize.com/.

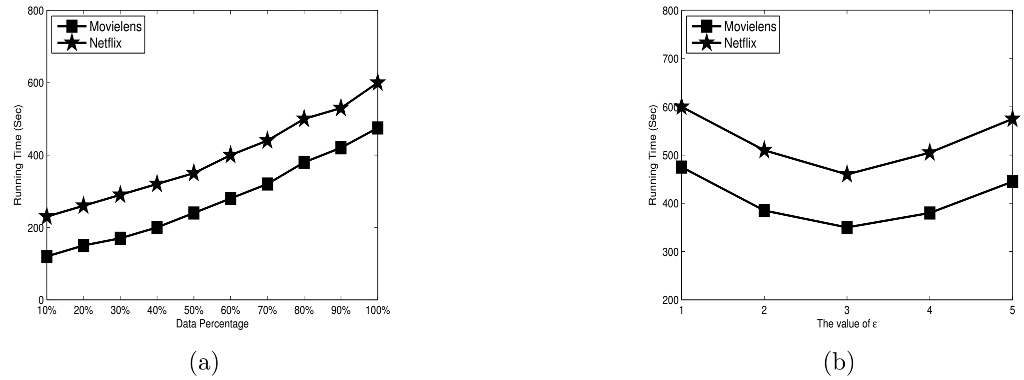


FIGURE 3: Running time comparison on Movielens and Netflix data sets vs. (a) Data percentage varies (b) ϵ varies

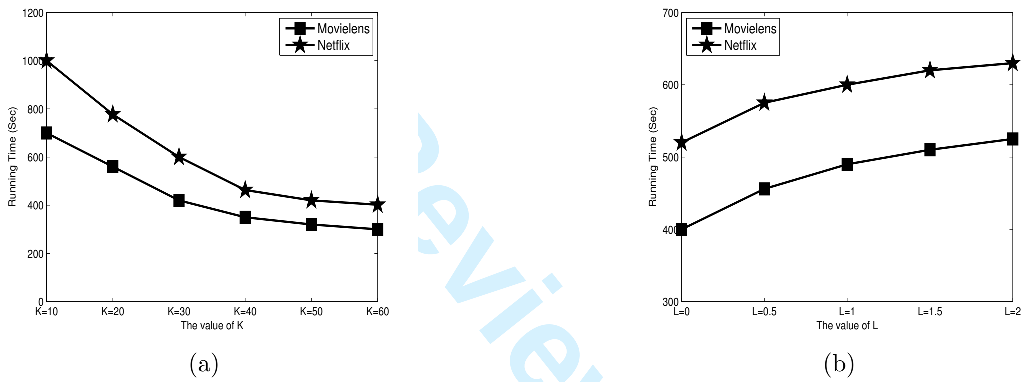
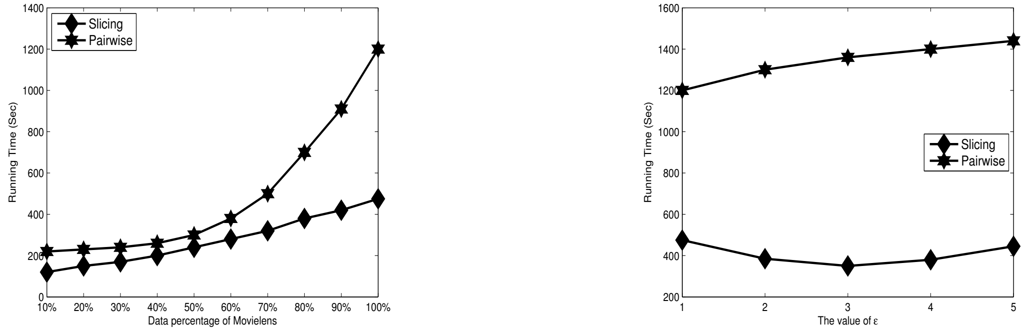


FIGURE 4: Running time comparison on Movielens and Netflix data sets vs. (c) k varies (d) L varies

in the ϵ-proximate neighborhood is increased, which results in huge exhaustive search for proper ϵ-proximate neighborhood and this causes the eventual cost increase. Setting $\epsilon = 2$, Figure 4(a) displays the results of running time by varying k from 10 to 60 for both data sets. The cost drops as k grows. This is expected, because fewer search efforts for proper ϵ-proximate neighborhoods needed for a greater k, allowing our algorithm to terminate earlier. We also run the experiment by varying the parameter l and the results are shown in Figure 4(b). Since the rating of both data sets are between 1 and 5, then according to Theorem 4.2, 2 is already the largest possible l. When $l = 0$, there is no diversity requirement among the sensitive issues, and the (k, ϵ, l)-anonymity model is reduced to (k, ϵ)-anonymity model. As we can see, the running time increases with l, because more computation is needed in order to enforce stronger privacy control.

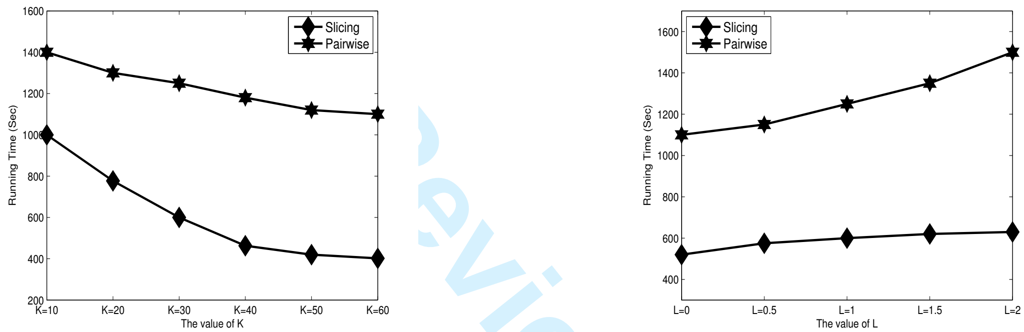
In addition to show the scalability and efficiency of the slicing algorithm itself, we also experimented the comparison between the slicing algorithm (Slicing) and the heuristic pairwise algorithm (Pairwise), which works by computing all the pairwise distance to construct the dissimilarity matrix and identify the violation of the privacy requirements. We implemented both algorithms and studied the impact of the execution time on the data percentage, the value of ϵ, the value of K and the value of L.

Figure 5 plots the running time of both slicing and pairwise algorithms on the Movielens data set. Figure 5(a) describe the trend of the algorithms by varying the percentage of the data set. From the graph we can see, the slicing algorithm is far more efficient than the heuristic pairwise algorithm especially when the volume of the data becomes larger. This is because, when the dimension of the data increases, the disadvantage of the heuristic pairwise algorithm, which is to compute all the dissimilarity distance, dominates the most of the execution time. On the other hand, the smarter grouping technique used in the slicing process makes less computation cost for the slicing algorithm. The similar trend is shown in Figure 5(b) by varying the value of ϵ, in which the slicing algorithm is almost 3 times faster than the the heuristic pairwise algorithm. The running time comparisons of both algorithms in Netflix data set by varying the value of K and L are shown in Figure 6(a) and (b). Even on a larger data set, the slicing algorithm outperformed the pairwise



(a) (b)

FIGURE 5: Running time comparison of Slicing and Pairwise methods on Movielens data set vs. (a) Data percentage varies (b) ϵ varies



(c) (d)

FIGURE 6: Running time comparison of Slicing and Pairwise methods on Netflix data set vs. (c) k varies (d) L varies

algorithm, and the running time of Slicing is quick enough to be used in practical.

7.3. Space complexity

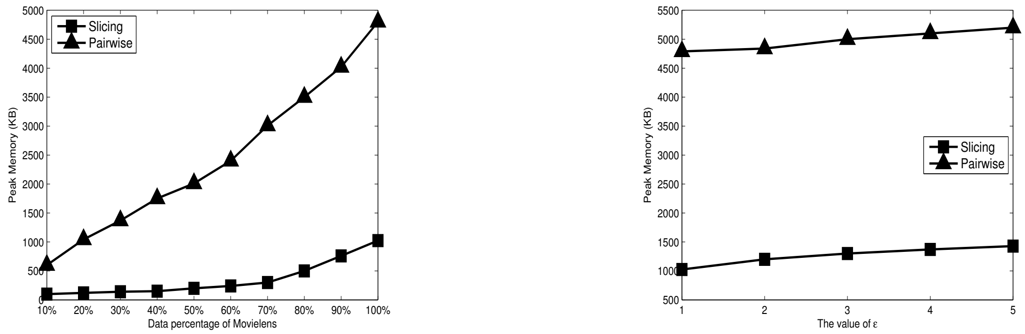
In addition to evaluate the efficiency of the proposed slicing technique, we also investigate the storage overheads of the algorithms. We adopt the peak memory to measure the storage overheads, which indicates the maximum memory used during the implementation.

Figure 7 shows the space complexity comparison of the slicing method and the pairwise approach on the Movielens data set by varying the percentage of the data and the value of ϵ. In both cases, the slicing algorithm takes less peak memory than the pairwise method, this is expected, since the pairwise approach computes all the possible distances and use them for identifying the validation of the privacy requirement, which takes much more space to store the dissimilarity matrix. We conduct the experiments by varying the value of K and L on a larger Netflix data set, and plot the storage overheads in Figure 8. From the figure, the space overhead is less for the slicing algorithm than for the pairwise method, which again outlines the disadvantage of the pairwise method, enumerating all the possible distances. The graph shows that the slicing algorithm need almost two times less memory than the heuristic pairwise approach.

8. CONCLUSION AND FUTURE WORK

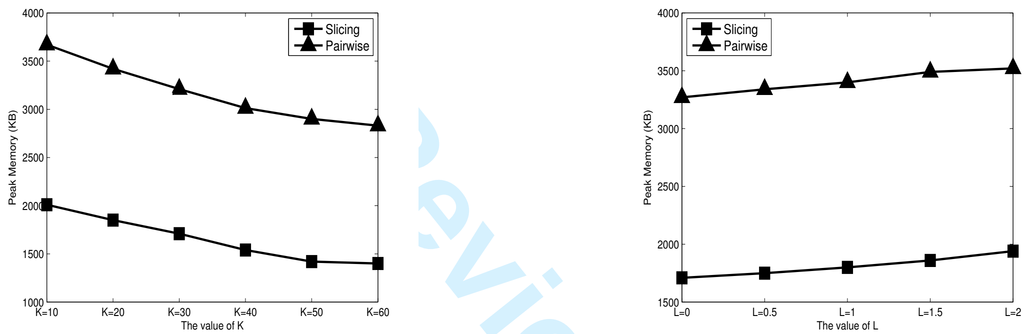
We have studied the problems of protecting sensitive ratings of individuals in a large public survey rating data. Such privacy risk has emerged in a recent study on the de-identification of published movie rating data. We proposed a novel (k, ϵ, l)-anonymity privacy principle for protecting privacy in such survey rating data. We theoretically investigated the properties of this model, and studied the satisfaction problem, which is to decide whether a survey rating data set satisfies the privacy requirements given by the user. A fast slicing technique was proposed to solve the satisfaction problem by searching closest neighbors in large, sparse and high dimensional survey rating data. The experimental results show that the slicing technique is fast and scalable in practical.

This work also initiates the future investigations of



(a) (b)

FIGURE 7: Space Complexity comparison of Slicing and Pairwise methods on Movielens data set vs. (a) Data percentage varies (b) ϵ varies



(a) (b)

FIGURE 8: Space Complexity comparison of Slicing and Pairwise methods on Netflix data set vs. (a) k varies (b) L varies

approaches on anonymizing the survey rating data. Traditional approaches on anonymizing no matter relational data sets or transactional data set are by generalization or suppression, and the published data set has the same number of data but with some fields being modified to meet the privacy requirements. As shown in the literatures, this kind of anonymization problem is normally NP-hard, and several algorithms are devised along this framework to minimize the certain pre-defined cost metrics. Inspired by the research in this paper, the satisfaction problem can be further used to develop a different method to anonymizing the data set. The idea is straightforward with the result of the satisfaction problem. If the rating data set has already satisfies the privacy requirement, it is not necessary to do any anonymization to publish it. Otherwise, we anonymize the data set by deleting some of the records to make it meet the privacy requirement. The criteria during the deletion can be various (for example, to minimize the number of deleted records) to make it as much as useful in the data mining or other research purposes. We believe that this new anonymization method is flexible in the choice of privacy parameters and efficient in the execution with the practical usage.

REFERENCES

[1] Sweeney, L (2002). k-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty Fuzziness Knowledge-based Systems*, 10(5), pp 557-570, 2002

[2] Machanavajjhala, A, Gehrke, J, Kifer, D, and Venkitasubramaniam, M (2006). l-Diversity: Privacy beyond k-anonymity. ICDE 2006.

[3] Li, N, Li, T and Venkatasubramanian, S (2007). t-Closeness: Privacy Beyond k-anonymity and l-diversity. *ICDE 2007*: 106-115

[4] Sun, X, Wang, H and Li, J (2009). Injecting purposes and trust into data anonymization. in CIKM 2009.

[5] Hansell, S. AOL removes search data on vast group of web users. New York Times, Aug 8 2006.

[6] Samarati, P and Sweeney, L (1998). Generalizing data to provide anonymity when disclosing Information. PODS 1998.

[7] Samarati, P and Sweeney, L (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.

Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998.

[8] Samarati, P (2001). Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6): pp: 1010-1027. 2001.

[9] Aggarwal, C (2005). On k-Anonymity and the curse of dimensionality. VLDB 2005.

[10] Bayardo, R and Agrawal, R (2005). Data privacy through optimal k-anonymisation. ICDE 2005.

[11] Fung, B, Wang, K, and Yu, P (2005). Top-down specialization for information and privacy preservation. ICDE 2005.

[12] Iyengar, V (2002). Transforming data to satisfy privacy constraints. SIGKDD 2002.

[13] LeFevre, K, DeWitt, D, and Ramakrishnan, R (2005). Incognito: efficient full-domain k-anonymity. SIGMOD 2005.

[14] LeFevre, K, DeWitt, D, and Ramakrishnan, R (2006). Mondrian multidimensional k-anonymity. ICDE 2006.

[15] J. Li, Y. Tao and X. Xiao. Preservation of Proximity Privacy in Publishing Numerical Sensitive Data. ACM Conference on Management of Data (SIGMOD), 2008

[16] Ghinita, G, Tao, Y and Kalnis, P (2008). On the Anonymisation of Sparse High-Dimensional Data, In Proceedings of International Conference on Data Engineering (ICDE) April 2008.

[17] Xu, Y, Wang, K, Fu, A and Yu, P (2008). Anonymizing Transaction Databases for Publication. KDD 2008.

[18] Hafner, K (2006). And if you liked the movie, a Netflix contest may reward you handsomely. New York Times, Oct 2 2006.

[19] Narayanan, A and Shmatikov, V (2008). Robust De-anonymisation of Large Sparse Datasets. to appear in IEEE Security & Privacy 2008.

[20] Xiao, X and Tao, Y (2006). Anatomy: simple and effective privacy preservation. VLDB 2006.

[21] Frankowski, D, Cosley, D, Sen, S, Terveen, L and Riedl, J (2006). You are what you say: privacy risks of public mentions. SIGIR 2006: 565-572

[22] Sun, X, Wang, H, Li, J and Pei, J (2010). Publish anonymous survey rating data. Accetped by *Data Mining and Knowledge Discovery*. 2010

[23] Agrawal, R and Srikant, R (2000). Privacy-Preserving Data Mining. SIGMOD 2000.

[24] Agrawal, D and Aggarwal, C (2001). On The Design and Qualification of Privacy Preserving Data Mining Algorithm. Proc. Symosium on Principles of Database Systems (PODS), pp247-255, 2001.

[25] Evfimievski, R, Srikant, R, Agrawal, R, and Gehrke, J (2002). Privacy preserving mining of association rules. SIGKDD 2002.

[26] Atzori, M, Bonchi, F, Giannotti, F, and Pedreschi, D (2005). Blocking anonymity threats raised by frequent itemset mining. ICDM 2005.

[27] Atzori, M, Bonchi, F, Giannotti, F, and Pedreschi, D (2005). k-anonymous patterns. PKDD 2005.

[28] Atzori, M, Bonchi, F, Giannotti, F, and Pedreschi, D (2008). Anonymity preserving pattern discovery. VLDB J. 17(4): 703-727 (2008)

[29] Verykios, V, Elmagarmid, A, Bertino, E, Dasseni, E and Saygin, Y (2004). Association Rule Hiding. IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 4, pp. 434-447, April 2004.

[30] Backstrom, L, Dwork, C and Kleinberg, J (2007). Wherefore Art Thou R3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography. WWW 2007.

[31] Huang, X, Xiang, Y, Chonka, A, Zhou, J, Deng, R (2010). A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems. IEEE Transactions on Parallel and Distributed Systems, 09 Nov. 2010. ¡http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.206¿

[32] Hamming, R (1980). Coding and Information Theory, Englewood Cliffs, NJ, Prentice Hall (1980)

[33] Friedman, J, Bentley, J, Finkel, R (1977). An algorithm for finding best matches in logarithmic expected time, ACM Trans. on Math. Software, 3(1977), pp. 209–226.