Publishing Sensitive Transactions for Itemset Utility

Yabo Xu[1], Benjamin C. M. Fung[2], Ke Wang[1†], Ada W. C. Fu[3], Jian Pei[1]

Simon Fraser University[1]
Burnaby, BC, Canada
{yxu,wangk,jpei}@cs.sfu.ca

Concordia University[2]
Montreal, QC, Canada
fung@ciise.concordia.ca

Chinese University of Hong Kong[3], Hong Kong
adafu@cse.cuhk.edu.hk

Abstract

We consider the problem of publishing sensitive transaction data with privacy preservation. High dimensionality of transaction data poses unique challenges on data privacy and data utility. On one hand, re-identification attacks tend to use a subset of items that infrequently occur in transactions, called moles. On the other hand, data mining applications typically depend on subsets of items that frequently occur in transactions, called nuggets. Thus the problem is how to eliminate all moles while retaining nuggets as much as possible. A challenge is that moles and nuggets are multi-dimensional with exponential growth and are tangled together by shared items. We present a novel and scalable solution to this problem. The novelty lies in a compact border data structure that eliminates the need of generating all moles and nuggets.

1. Introduction

Transaction data such as web search queries, online/offline purchase records, click streams, and emails are rich sources for data mining applications. However, the use of such data may pose serious privacy threats. An example is the release of AOL query logs (New York Times, August 9 2006), where the searcher No. 4417749 was traced back to Ms Thelma Arnold by examining query content. Recently, Netflix released a movie rating data set to improve movie recommendation algorithm (New York Times, October 2 2006). The study in [13] showed that by as little prior knowledge as no more than 8 movie ratings and approximate dates, 96% of subscribers can be uniquely re-identified!

This type of re-identification attack has been studied for *relational data* [3]. In relational data, a set of public attributes, called *quasi-identifier* (QI), is used to link an individual to a private record. The classic example given in [3] is that prior knowledge on QI={gender, date of birth, zip code} can uniquely link a record in a public voter list (with explicit names) to a private record in a medical database. Surprisingly, the re-identification problem has received very little attention on *transaction data*, despite the widespread collection and publication of transaction data. Compared to relational data, transaction data has several prominent characteristics, namely, high dimensionality, lack of QI, and itemset-based data utility.

High Dimensionality Transaction data does not have a natural notion of "attributes" because each transaction is a set of "items" taken from a large universe. For example, the Netflix data set has 18,000 distinct movies and 5 possible rating scales. If each "movie=rating" pair is treated as an item and each item is treated as a dimension, there would be 18,000×5 items in the universe, thus, 18,000×5 dimensions in QI. Existing methods of anonymizing QI will lose too much information on such high dimensional QI [1].

Data Privacy With the high dimensionality of transaction data, it is unrealistic to assume that the attacker has prior knowledge on all items in the universe. Instead, the attacker tends to obtain prior knowledge on a *subset* of items in each attack (such as no more than 8 movie ratings and approximate dates as discussed early). We can measure the attacker's "power" by the maximum number of items for such prior knowledge in a single attack. Borrowed from [5], the term "mole" refers to any such subset of items that can be used to link an individual to a transaction.

Data Utility Typically transaction data are published for data mining applications where sets of items that co-occur frequently, also called *frequent itemsets* in [7], represent associations between items and are fundamental to various data analysis tasks, including association rules mining, classification, correlation, causality and emerging pattern. We use the term "nugget" to refer to a frequent itemset.

In this paper, we study the following problem: given a collection D of transactions, we want to publish a modified version of D such that all moles are

† This work is supported in part by a grant from Natural Sciences and Engineering Research Council of Canada.

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.98

eliminated and nuggets are retained as much as possible. This problem faces major challenges. Apparently, to preserve nuggets, which are frequent itemsets, it does not work to preserve each item individually; it is necessary to preserve a subset of items as an information unit. Due to exponentially many such subsets of items, any method that requires generating and storing all moles and nuggets is impractical. Moreover, moles and nuggets are tangled together by shared items, making it non-trivial to eliminate moles but retain nuggets.

To address these challenges, we present an item suppression solution that greedily eliminates moles for each nugget lost. To address the exponential blow-up of moles and nuggets, we present a novel border-based method that examines only maximal and minimal moles/nuggets and yet produces exactly same solution as if all moles and nuggets were examined. We study the effectiveness of this approach on several public data sets. The results are however not included in this paper due to the space.

2. Related work

Though there has been a great deal of works on privacy-preserving data publishing (PPDP) for relational data, led by *k*-anonymity[2] and *l*-diversity[3], much less is known for transaction data. The goal of publishing data is stronger than publishing frequent itemsets as in [9]. With the data being available, the users can visualize transactions, try different methods and parameters, and evaluate models against data. All such tasks cannot be done without data. To prevent re-identification attack, we consider prior knowledge having a small support as a threat because fewer matching transactions mean a higher probability of re-identification. This is different from the scenario in [10] that considers having a small support as a protection, as evidenced by their method of hiding a sensitive pattern by decreasing support.

Only until recently several works start to address re-identification attacks for transaction data [4][5][6]. The authors in [4] adapt the bucketization approach for relational data to transaction data, which is vulnerable to background knowledge attacks As pointed out by [12]. In addition, [4] does not model the attack's power, consequently, it may purge some useful rules even if such rules are beyond a realistic attacker's power. Finally, bucketization produces transactions with "probabilistic" private items. Mining such transactions requires modifying standard data mining algorithms.

Modeling attacker's power by a maximum number of items for prior knowledge was considered in [5][6]. The authors in [6] consider *k*-anonymity as the privacy goal, which is vulnerable to homogeneity attacks [2]. Our privacy goal is similar to [5] with two major differences. First, we model frequent itemsets as utility, whereas [5] models individual items as utility. Second, we employ a border-based method to address the exponential blow-up of moles and nuggets, therefore, offer better scalability.

3. Problem Statements

We assume a universe of items denoted by *I*. An item is either *public* (or *identifying*), on which an attacker may acquire prior knowledge, or *private* (or *sensitive*), which are to be protected. For example, HIPAA provides such guidelines in health domains for classifying public and private items. Let *D* be a set of transactions where each transaction corresponds to a distinct individual and contains some public items and private items. An *itemset* refers to a set of items, with a *public itemset* containing only public items. For any itemset α, $|\alpha|$ denotes the length of α, α-*cohort* denotes the set of all transactions that contain all the items in α, $\mathrm{Sup}(\alpha)$ denotes the *support* of α, i.e., the number of transactions in α-cohort. $\Pr(s|\alpha) = \mathrm{Sup}(\alpha \cup \{s\})/\mathrm{Sup}(\alpha)$ is the probability of associating α with an item *s*.

3.1. Moles and Nuggets

We assume that the attacker knows that a target individual *P* has a transaction in *D* and tries to identify this transaction from the published version of *D*. As prior knowledge, the attacker also knows that *P* possesses all the items in some public itemset α. This makes all transactions in the α-cohort the candidates for *P*'s transaction. Clearly, it is unrealistic to assume that the attacker has unlimited prior knowledge. To this end, we use a parameter *p* to specify the maximum length of prior knowledge α that the attacker may obtain. The publisher can use *p* to balance his perception on attacker's "*power*" and his need for data utility.

We consider two privacy goals. To prevent *identity attack* [3], which occurs when *P* is linked to a particular transaction, we require $\mathrm{Sup}(\alpha) \geq k$, where *k* is a privacy parameter. This requirement bounds the probability of linking *P* to a particular transaction by $1/k$. To prevent *attribute attack* [2], which occurs when *P* is linked to a private item, we introduce the notion *breach probability*.

Breach probability. For a public itemset α, the *breach probability* of α, denoted by $\mathrm{BPr}(\alpha)$, is the maximum $\Pr(s|\beta)$ for any private item *s* and any $\beta \subseteq \alpha$. $\mathrm{BPr}(\alpha)$ captures the maximum probability of inferring

any private item through α or its subsets. Note that for $\alpha \subseteq \beta$, $BPr(\alpha) \leq BPr(\beta)$. To prevent attribute attacks, our second privacy goal is to bound $BPr(\alpha)$ by h, where h is a privacy parameter. We consider only s that is a single private item because $Pr(s|\alpha) \geq Pr(S|\alpha)$ for any set S containing s.

Definition 1 (Moles) Given integers $k>1$, $p>0$ and a real $0<h\leq 1$, a public itemset α with $|\alpha| \leq p$ is a *mole* wrt (h,k,p) if either $Sup(\alpha)<k$ or $BPr(\alpha)>h$; otherwise α is called a *non-mole* wrt (h,k,p). $M(D)$ denotes the set of moles in D wrt (h,k,p). D is *(h,k,p)-coherent* if D contains no mole wrt (h,k,p). ■

Intuitively, (h,k,p)-coherence guarantees that, for the attacker with the power p, no identity attacks (limited by k) or attribute attacks (limited by h) are possible on D. A larger power p means more privacy protection and more data distortion. With the parameter p, useful rules can be preserved while eliminating moles. A rule $\alpha \rightarrow s$ with high $Pr(s|\alpha)$ tends to have a long antecedent α. Our problem seeks to eliminate only rules with $|\alpha| \leq p$, thus, accurate rules, which are useful to research, may still be preserved after eliminating all moles.

We consider preserving "frequent itemsets" as information nuggets, defined below.

Definition 2 (Nuggets) Given integers $k'>1$ and $p'>0$, an itemset α (containing either pubic or private items) is a *nugget* wrt (k',p') if $|\alpha| \leq p'$ and $Sup(\alpha) \geq k'$. $N(D)$ denotes the set of nuggets in D wrt (k',p'). ■

	Public	Private
T_1	*a, b, e, f*	s_1
T_2	*c, e, f, g*	s_2
T_3	*a, b, g*	s_3
T_4	*a, b, f, g*	s_2
T_5	*a, b, d, g*	s_2
T_6	*e, f, g*	s_1
T_7	*b, e, f, g*	s_3

Figure 1 h=50%, k=3, p=3, k'=4 and p'=∞

Example 1 (Running example) Consider D in Figure 1. *a-g* are public items and s_1-s_3 are private items. D violates $(h=50\%, k=3, p=3)$-coherence. For example, *ae* is a mole because $Sup(ae)=1<k$. If the attacker knows that Jane engages in the activities *a* and *e*, Jane will be uniquely linked to T_1. *ag* is a mole because $BPr(ag)=2/3>h=50\%$: *ag* occurs in T_3-T_5, two of which contain s_2. Examples of nuggets are *a*, *ab* and *bg*. ■

3.2. Item Suppression

If D is not (h,k,p)-coherent, we shall suppress some items to achieve coherence. Suppressing an item means *deleting* the item from *all* transactions that contain the item. Such item suppression has the following properties.

Observation 1 (1) Suppressing an item eliminates all itemsets that contain the item. (2) Suppressing an item does not alter any itemset, and its support, that does not contain the item. (3) Suppressing an item does not introduce a new itemset.

Let D' be the modified data obtained from D by suppressing items. (1) and (3) imply that $N(D') \subseteq N(D)$ and $M(D') \subseteq M(D)$. (2) implies that any nugget in D' has *exactly the same* support as in D. This property is essential to data analysis that relies on support of itemsets for probability estimation. For example, the probability $Pr(s|\alpha)$ of a rule $\alpha \rightarrow s$ can be estimated by the ratio $Sup(\alpha \cup \{s\})/Sup(\alpha)$. A small change in support could result in instability of estimated probability, thus an arbitrary decision making. *Partial suppression* does not preserve supports of itemsets because it allows suppressing some but not all occurrences of an item. For this reason, we do not consider partial suppression.

The authors in [6] consider generalizing items assuming that a taxonomy on items is available. In practice, however, a taxonomy may not be available. Another reason that we do not consider generalization is that, in a global recoding for generalization, either all sibling items or none must be generalized. For a large universe of items, such generalization tends to lose too much information because the item taxonomy typically has a large fan-out. Though this problem may be addressed by a local recoding for generalization where only some selected sibling items are generalized, local recoding does not preserve the support of itemsets. Item suppression has less information loss, because the decision on suppressing an item is made for each item independently, and also preserves the support for the remaining itemsets.

3.3. The Problem

Definition 3 (Optimal (h, k, p)-Cohesion) D' is *a (h,k,p)-cohesion* of D if D is transformed to (h,k,p)-coherent D' by suppressing public items. D' is called *an optimal (h,k,p)-cohesion* if D' is a (h,k,p)-cohesion and for any other (h,k,p)-cohesion D'', $|N(D'')| \leq |N(D')|$ wrt given k' and p'. ■

In this paper, we consider nuggets of any length, i.e., $p'=\infty$. First, we can show the optimal cohesion is NP-hard by a reduction from the *vertex cover problem*. The detail is omitted here.

Theorem 1 For $k'=p'=1$, $k=2$, $p=2$, and any h, the optimal cohesion problem is NP-hard. ■

We can show that D has a (h,k,p)-cohesion if and only if the empty itemset $\varnothing$ is not a mole. Therefore, in the rest of paper we assume that D has a (h,k,p)-cohesion and we present an algorithm for finding a "good" cohesion *efficiently*.

4. Our approach

Our goal is to find a (h,k,p)-cohesion of D while retaining as many nuggets as possible wrt (k',p'). Consider a pubic item e. If $\text{Sup}(e)<k'$, e does not occur in any nugget; if $\text{Sup}(e)<k$, e is a mole by itself, thus, must be suppressed. This leads to the following observation.

Observation 2 If a public item e has $\text{Sup}(e)<\max(k,k')$, deleting the item either has no effect on nugget or loses only nuggets that must be lost in every (h,k,p)-cohesion.

Based on this observation, we assume that all public items e with $\text{Sup}(e)<\max(k,k')$ have been deleted from D. Our algorithm is described in Figure 2. This algorithm assumes a function Score(v) that determines the "worth" of suppressing an item v. In each iteration, it suppresses a remaining public item v with the maximum Score(v), by adding v to SuppItems, and updates Score(v') for the remaining public items v'. The algorithm terminates until no mole remains.

1. initialize SuppItems to the empty set;
2. **while** there is some mole **do**
3. select a remaining public item v with max Score(v);
4. add v to SuppItems;
5. update Score(v') for remaining items v';
6. **end while**
7. suppress all items in SuppItems from the database;

Figure 2 Item suppression algorithm

To define Score(v), let M(v) and N(v) denote the set of moles and nuggets that contain an item v, and $|M(v)|$ and $|N(v)|$ denote the number of such moles and nuggets.

Defining Score(v) This score evaluates the "worth" of suppressing an item v. From Observation 1(1), suppressing the item v will eliminate all moles in M(v) and all nuggets in N(v). We define Score(v) = $|M(v)|/|N(v)|$. Score(v) gives the number of moles eliminated for each nugget lost due to suppressing the item v. Let Score(v) =∞ if $|N(v)|=0$.

Updating Score(v') After suppressing v, for a remaining public item v', all moles and nuggets containing vv' are eliminated, so $|M(v')|$ and $|N(v')|$ should be decreased by the number of such moles and nuggets. The next proposition, which follows from Definition 1 and Definition 2, shows that moles and nuggets have exponential growth; therefore, it is not efficient to update $|M(v')|$ and $|N(v')|$ by examining all moles and nuggets.

Proposition 1 (1) If α is a mole, every itemset β with $\alpha\subseteq\beta$ and $|\beta|\le p$ is a mole. (2) If α is a nugget, every itemset β with $\beta\subseteq\alpha$ is a nugget. ■

5. Border Representation

Materializing all moles and nuggets is infeasible due to their exponential growth (Proposition 1). A novelty in our approach is materializing *only* "maximal" and "minimal" moles and nuggets, which form the "borders" that enclose all moles and nuggets. We introduce such a border representation and operations on a border. In the next section, we devise an efficient update of $|M(v')|$ and $|N(v')|$ based on the border representation.

5.1 Border definition

The notion of borders has been studied in [8].

Definition 4 (Borders) An ordered pair [U, L] is called a *border* if (i) each of U and L is an anti-chain[2] collection of itemsets, and (ii) each element of U is a subset of some element in L, and each element of L is a superset of some element in U. U is the *upper bound* and L is the *lower bound*. A border [U, L] represents the set of itemsets $\{\gamma \mid \exists\alpha\in U, \beta\in L \text{ such that } \alpha\subseteq\gamma\subseteq\beta\}$. ■

A collection S of itemsets is *interval-closed* if, for all itemsets α, $\beta\in S$ and for all itemsets γ, whenever $\alpha\subseteq\gamma\subseteq\beta$, $\gamma\in S$. It follows from Proposition 1 that moles M(D) and nuggets N(D) are interval-closed, therefore, can be represented by borders.

Proposition 2 M(D) and N(D) are interval-closed.

It is shown in [8] that every interval-closed collection S has a unique border [U,L]: U is the collection of *minimal* itemsets in S (i.e., those that have no subset in S) and L is the collection of *maximal* itemsets in S (i.e., those that have no superset in S). Specifically, we can represent all nuggets and all moles by their borders.

The nugget border B_N The upper bound U contains all minimal nuggets, which are all singleton items v with $\text{Sup}(v)\ge\max(k,k')$ because all items v with

[2] A collection S of sets is an *anti-chain* if X and Y are incomparable sets (i.e. X is not a subset of Y, neither Y is a subset of X) for all X, $Y\in S$.

Sup(v) <max(k,k') have been removed (Observation 2). The lower bound L consists of all maximal itemsets α with Sup(α)≥k'.

The mole border B_M The upper bound U consists of all minimal moles and the lower bound L consists of all maximal itemsets with support≥1. Note that not all itemsets represented by [U,L] have length > p. Our counting procedure in Section 5.2 will not count such itemsets as moles, though they are represented by the mole border.

The problem of finding a border [U,L] for an interval-closed collection has been studied [8]. For the rest of our approach, we assume that the borders B_M and B_N have been found.

For convenience, we represent a border [U, L] by the set of *edges* {<α, β>| $\alpha\in$U, $\beta\in$L, $\alpha\subseteq\beta$}. We say that an itemset γ is *covered* by an edge <α, β> if $\alpha \subseteq \gamma \subseteq \beta$. An itemset γ is *covered* by a set of edges if it is covered by some edge in the set.

Example 2 Refer to the D and settings in Figure 1. First, we delete c and d according to Observation 2. The mole border B_M is shown in Figure 3. Each link represents an edge in the border. *aef* is a mole and is covered by edges <*ae, abef*> and <*af, abef*>. a*bef*, though covered by the border, is not a mole because it exceeds the maximum length p=3. ■

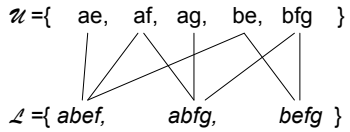

Figure 3 The mole border B_M**.**

5.2 The counting function

Suppose that we suppress an item v at Line 4 in Figure 2. For a remaining item v', |M(v')| and |N(v')| should be decreased by the number of moles/nuggets that contain vv' because such moles/nuggets are also eliminated. The question is how to compute the number of such moles and nugget from only the borders B_M and B_N. To compute these numbers, we define the following counting function:

$$W_{X,,len}([U,L]) = |\{\gamma | \gamma\in[U,L], X\subseteq\gamma, |\gamma|\le len\}|.$$

This function returns the number of itemsets that contain the itemset X as a subset, have length ≤len, and are covered by the border [U,L]. We are interested in computing $W_{X,,len}$([U,L]) using the border but not generating all covered itemsets.

Example 3 Refer to Figure 3. $W_{ag,,3}$ (B_M)=3 gives the number of moles that contain *ag* (note p=3), i.e. *ag*, *afg*, *abg* (covered by <*ag*, *abfg*>). Notice that *afg* is also covered by <*af*, *abfg*> ■

To compute $W_{X,,len}$([U,L]) using the borders, first we define some operations. Consider a single edge <α,β> in the border. Suppose that $W_{X,,len}$(<α,β>) returns the number of itemsets that are covered by <α,β>, have length ≤len, and contain all the items in X:

$$W_{X,,len}(<\alpha,\beta>) = |\{\gamma | \alpha\subseteq\gamma\subseteq\beta, X\subseteq\gamma, |\gamma|\le len\}|.$$

Observe that γ always contains $\alpha\cup X$ and have length ≤len. The other items in γ are chosen from β-($\alpha\cup X$) and are no more than m=Min(|β-($\alpha\cup X$)|, len-|$\alpha\cup X$|). The number of such γ is

$$W_{X,len}(<\alpha,\beta>) = \sum_{i=0}^{m} C^{i}_{|\beta-(\alpha\cup X)|},$$

where $C_n^i = n!/[i!(n-i)!]$. In the special case of X=∅ and len=∞, W(<α,β>)=$2^{|\beta-\alpha|}$ returns the number of itemsets covered by <α,β>. The efficiency of this operation lies at not enumerating the itemsets being counted.

Now we consider computing $W_{X,,len}$([U,L]). It does not work to sum up $W_{X,,len}$(<α,β>) over all edges <α,β> in [U,L] since an itemset may be covered by several edges (e.g., *aef* in Example 2). To remove duplicate counting, we introduce two more operations on edges:

Edge intersection, denoted <α_1,β_1>∩<α_2,β_2>, applies the intersection operator ∩ to the sets of itemsets covered by <α_1,β_1> and <α_2,β_2>.

Edge difference, denoted <α_1,β_1>-<α_2,β_2>, applies the difference operator - to the sets of itemsets covered by <α_1,β_1> and <α_2,β_2>.

Theorem 2 (Edge intersection) (1) <α_1,β_1>∩<α_2,β_2> is equal to the set of itemsets covered by <$\alpha_1\cup\alpha_2$, $\beta_1\cap\beta_2$>. (2) <α_1, β_1>∩<α_2, β_2>≠∅ if and only if $\alpha_1\cup\alpha_2\subseteq\beta_1\cap\beta_2$. ■

Theorem 3 (Edge difference) Assume that <α_1,β_1>∩<α_2,β_2>≠∅. Let x=$\alpha_1\cup\alpha_2$ and y=$\beta_1\cap\beta_2$. <α_1,β_1>-<α_2,β_2> is equal to the set of itemsets covered by {<α_1,β_1-{v}>| $v\in x$-α_1}∪{<$\alpha_1\cup${v}, β_1> | $v\in\beta_1$-y}.

Proof: omitted.■

Example 4 Consider <*ae*,*adefg*>-<*ae*,*ade*>. Refer to Theorem 3. x=$\alpha_1\cup\alpha_2$=*ae* and y=$\beta_1\cap\beta_2$=*ade*. x-α_1=∅ and β_1-y=*fg*. The first term {<α_1,β_1-{v}>| $v\in x$-α_1} is empty. Thus, <*ae*,*adefg*>-<*ae*,*ade*>= {<$\alpha_1\cup${v},β_1> | $v\in$ *fg*}={<*aef*, *adefg*>, <*aeg, adefg*>}■

6. The border-based update of Score(v')

Now let us consider the update of Score(v') in Line 5 of Figure 2, i.e., the update of |M(v')| and |N(v')|. We consider |M(v')|; the situation is similar for |N(v')|.

Consider the algorithm in Figure 2. Let v be the item suppressed in the current iteration, and let $E(v)$ be the set of all the edges $<\alpha,\beta>$ in B_M with $v\in\beta$. By suppressing v, all moles that contain v, called *losers*, are eliminated. Note that any loser must be covered by some edge in $E(v)$. Let $\sigma=\cup\beta-\{v\}$, where $\cup$ is over all $<\alpha,\beta>$ in $E(v)$. For a remaining item v', $M(v')|$ is affected only if $v'\in\sigma$. Therefore, for every $v'\in\sigma$, let $\delta(v')$ denote the number of losers that contain vv', and $M(v')$ should be decreased by $\delta(v')$. Below, we discuss how to compute $\delta(v')$ for all $v'\in\sigma$.

Computeδ(E^*, $\hat{E}$, v, *len*, δ). v is the item suppressed in the current iteration, and *len* is the maximum length for the itemsets being counted. E^* and $\hat{E}$ are the two disjoint partitions of $E(v)$, i.e. $E(v)=E^*\cup\hat{E}$, and $E^*\cap\hat{E}=\varnothing$. E^* is the set of *unexamined* edges (initially $E(v)$) and $\hat{E}$ is the set of *examined* edges (initially empty). For all items $v'\in\cup\beta-\{v\}$, where $\cup$ is over all $<\alpha,\beta>$ in E^*, this operation returns $\delta(v')$ as the number of losers that contain vv' and are covered by E^* but not by $\hat{E}$.

This operation makes one pass of the edges in E^*. At each step, we consider the next edge $<\alpha,\beta>$ in E^*: for every item $v'\in\sigma$, count the *new* losers containing vv' that are covered by $<\alpha,\beta>$ but not covered by any (examined) edge in $\hat{E}$, and increment $\delta(v')$ by the count. Then move $<\alpha,\beta>$ from E^* to $\hat{E}$. This process is repeated until E^* becomes empty. The final $\delta(v')$ gives the number of losers containing vv'.

To count the new losers covered by $<\alpha,\beta>$ but not by any edge in $\hat{E}$, we count the losers covered by $<\alpha,\beta>$ and by some edges in $\hat{E}$. To this end, we identify the set of edges in $\hat{E}$ that "overlap with" $<\alpha,\beta>$:

ovset = $\{\hat{e} \mid \hat{e}\in \hat{E}$ such that $<\alpha,\beta>\cap\hat{e}\neq\varnothing\}$,

where $<\alpha,\beta>\cap\hat{e}\neq\varnothing$ can be tested by Theorem 2(2). We exclude all losers covered by *ovset* in three cases:

Case 1: $|ovset|=0$. The losers covered by $<\alpha,\beta>$ are not covered by $\hat{E}$, so $W_{X,len}(<\alpha,\beta>)$ gives the number of new losers containing vv', where $X=vv'$ and $len=p$. We update $\delta(v')$ to $\delta(v') + W_{X,len}(<\alpha,\beta>)$.

Case 2: $|ovset| = 1$. In this case, only one edge in $\hat{E}$, say $\hat{e}$, has overlap with $<\alpha,\beta>$. Following Theorem 2, the number of losers covered by both $<\alpha,\beta>$ and $\hat{e}$ is given by $W_{X,len}(<\alpha,\beta>\cap\hat{e})$, where $X=vv'$, $len=p$. We increment $\delta(v')$ by $W_{X,len}(<\alpha,\beta>)-W_{X,len}(<\alpha,\beta>\cap\hat{e})$.

Case 3: $|ovset| > 1$. In this case, more than one edge in $\hat{E}$ has overlap with $<\alpha,\beta>$. Simply excluding the intersection $<\alpha,\beta>\cap\hat{e}$ for every $\hat{e}$ in *ovset* does not work because intersections themselves might have intersection. Our approach is as follows. We pick any $\hat{e}$ in *ovset* and compute $<\alpha,\beta>-\hat{e}$. Following Theorem 3, this edge difference can be replaced with a set of new edges *newset*. Then we count the losers covered by the *unexamined* E^* = *newset* but not by the *examined* $\hat{E}$=*ovset*-$\{\hat{e}\}$ by a recursive call of Computeδ(*newset*, *ovset*-$\{\hat{e}\}$, v, *len*, δ) . The recursion terminates in either Case 1 or Case 2.

Intuitively, the efficiency of Computeδ lies at *computing* the number of certain itemsets instead of *enumerating* such itemsets. This eliminates the need of storing all moles and nuggets in memory, which is the real bottleneck due to the exponential blowup of moles and nuggets. The detailed algorithm will be reported in the full version of this paper.

7. References

[1] C. Aggarwal. On *k*-Anonymity and the curse of dimensionality. VLDB 2005.

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

[3] L. Sweeney. *k*-anonymity: a model for protecting privacy. *Int.J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5), 2002.

[4] G. Ghinita, Y. Tao, P. Kalnis. On the anonymization of sparse high-dimensional data. ICDE 2008.

[5] Y. Xu, K. Wang, Ada. W.C. Fu, and Philip. S. Yu. Anonymizing transaction databases for publication. SIGKDD 2008.

[6] M. Terrovitis, N. Mamoulis, and P. Kalnis. Anonymity in unstructured data. VLDB, 2008.

[7] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. SIGMOD 1993.

[8] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. KDD 1999.

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

[10] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. TKDE, 16(4):434-447, 2004.

[11] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. TKDE, 16(4):434-447, 2004.

[12] T. Li and N. Li Injector: Mining Background knowledge for data anonymization. ICDE 2008.

[13] A. Narayanan and V. Shmatikov. How to break anonymity of the Netflix prize data set. ArXiv Computer Science e-prints, October 2006.