On the Anonymization of Sparse High-Dimensional Data

Gabriel Ghinita [#1], Yufei Tao [*2], Panos Kalnis [#1]

[#]*Department of Computer Science, National University of Singapore*
Computing 1, Singapore 117590
[1]{ghinitag,kalnis}@comp.nus.edu.sg

[*]*Department of Computer Science and Engineering, Chinese University of Hong Kong*
Sha Tin, New Territories, Hong Kong SAR, China
[2]taoyf@cse.cuhk.edu.hk

***Abstract*— Existing research on privacy-preserving data publishing focuses on relational data: in this context, the objective is to enforce privacy-preserving paradigms, such as k-anonymity and ℓ-diversity, while minimizing the information loss incurred in the anonymizing process (i.e. maximize data utility). However, existing techniques adopt an indexing- or clustering-based approach, and work well for fixed-schema data, with low dimensionality. Nevertheless, certain applications require privacy-preserving publishing of transaction data (or basket data), which involves hundreds or even thousands of dimensions, rendering existing methods unusable.**

We propose a novel anonymization method for sparse high-dimensional data. We employ a particular representation that captures the correlation in the underlying data, and facilitates the formation of anonymized groups with low information loss. We propose an efficient anonymization algorithm based on this representation. We show experimentally, using real-life datasets, that our method clearly outperforms existing state-of-the-art in terms of both data utility and computational overhead.

I. Introduction

The problem of privacy-preserving data publishing has received a lot of attention in recent years. Most of existing work is formulated in the following context: Several organizations, such as hospitals, publish detailed data (also called *microdata*) about individuals (e.g. medical records) for research or statistical purposes. However, sensitive personal information may be disclosed in this process, due to the existence in the data of quasi-identifying attributes, or simply quasi-identifiers (QID), such as age, zipcode, etc. An attacker can join the QID with external information, such as voting registration lists, to re-identify individual records.

Existing privacy-preserving techniques focus on anonymizing personal data, which have a fixed schema with a small number of dimensions. Through *generalization* or *suppression*, existing methods prevent attackers from re-identifying individual records.

However, anonymization of personal data is not sufficient in some applications. Consider, for instance, the example of a large retail company which sells thousands of different products, and has numerous daily purchase transactions. The large amount of transactional data may contain customer spending patterns and trends that are essential for marketing and planning purposes. The company may wish to make the data available to a third party which can process the data and extract interesting patterns (e.g. perform data mining tasks). Since the most likely purpose of the data is to infer certain purchasing trends, characterized by *correlations* among purchased products, the personal details of the customers are not relevant, and are altogether suppressed. Instead, only the contents of the shopping cart is published for each transaction. Still, there may be particular purchasing habits that disclose customer identity and expose sensitive customer information.

A. Motivation

Consider the example in Fig. 1a, which shows the contents of five purchase transactions (the customer name is not disclosed, we include it just for ease of presentation). The sensitive products (items), which are considered to be a privacy breach if associated to a certain individual, are shown shaded. The rest of the items, which are non-sensitive, can be used by an attacker to re-identify individual transactions, similarly to a quasi-identifier, with the distinctive characteristic that the number of potentially identifying items is very large in practice (hence, the QID has very high dimensionality). Consider the transaction of Claire, who has bought a pregnancy test. An attacker (Eve) may easily learn about some of the items purchased by Claire on a certain day, possibly from a conversation with her, or from knowing some of her personal preferences. For instance, Claire may treat her guests, including Eve, with fresh cream and strawberries, and Eve can therefore infer that Claire must have purchased these items recently. Joining this information with the purchase transaction log, Eve can re-identify Claire's transaction, and find out that Claire may be pregnant.

The privacy breach occurs because Eve was able to identify with certainty the purchase transaction of Claire, and hence associate her with the sensitive item *pregnancy test*. As we will show later in Section V, we found that for a real-life dataset, an attacker can re-identify the transaction of a particular individual with 20% probability based on knowledge on two purchased items. The probability increases to 40% with three known items, and to over 90% with four items. To protect

978-1-4244-1837-4/08/$25.00 © 2008 IEEE

	Wine	Strawberries	Meat	Cream	Pregnancy Test	Viagra
Bob	X		X			X
David	X		X			
Claire		X		X	X	
Andrea		X	X			
Ellen	X		X	X		

(a) Original Data

	Wine	Meat	Cream	Strawberries	Pregnancy Test	Viagra
Bob	X	X				X
David	X	X				
Ellen	X	X	X			
Andrea		X		X		
Claire			X	X	X	

(b) Re-organized Data

	Wine	Meat	Cream	Strawberries	Sensitive Items
Bob	X	X			Viagra: 1
David	X	X			
Ellen	X	X	X		
Andrea		X		X	Pregnancy Test: 1
Claire			X	X	

(c) Published Groups

Fig. 1. Purchase Transaction Log Example

Claire's privacy, we must prevent the association of her QID items to a particular sensitive item, with probability larger than a certain threshold.

To address this privacy threat, one solution would be to employ ℓ-diversity [1]: a well-established paradigm in relational data privacy, which prevents sensitive attribute (i.e. item) disclosure. ℓ-diversity partitions the data into groups of records (i.e. transactions in our case) such that ℓ sensitive item values are well-represented in each group. Currently, there exist two broad categories of ℓ-diversity techniques: *generalization*- and *permutation*-based. Both categories assume fixed-schema data, with a relatively low number of QID items. In our case, the transactional data is represented as a table with one row for each transaction t, and one column for each possible item. For each transaction t, a certain column has value **1** if the corresponding item belongs to t, and **0** otherwise.

An existing generalization method [2], [3], [4] would partition the data into disjoint groups of transactions, such that each group contains sufficient records with ℓ distinct, well-represented sensitive items. Then, all quasi-identifier values in a group would be generalized to the entire group extent in the QID space. If at least two transactions in a group have distinct values in a certain column (i.e. one contains an item and the other does not), then all information about that item in the current group is lost. The QID used in this process includes all possible items in the log. Due to the high-dimensionality of the quasi-identifier, with the number of possible items in the order of thousands, it is likely that any generalization method would incur extremely high information loss, rendering the data useless [5].

In contrast, a permutation method such as Anatomy [6] would randomly pick groups of transactions with distinct sensitive items, and permute these items among transactions, to reduce the association probability between an individual transaction and a particular sensitive item. However, the group formation phase does not consider similarity among QID values of transactions, hence correlations between QID and sensitive items may be lost. This is undesirable, as it may prevent the extraction of useful information from the data, reducing its utility.

B. Contributions

We propose an anonymization technique which combines the advantages of both generalization and permutation, and also addresses the difficult challenge of high dimensionality. First, we devise a novel representation of data which takes advantage of its sparseness and preserves correlation. We organize the data as a *band matrix* (Fig. 1b) by performing permutations of rows and columns in the original table, such that most non-zero entries are near the main diagonal. The advantage of this representation is that neighboring rows have high correlation, i.e. share a large number of common items. Next, we propose an efficient heuristic to create good-quality groups, which only needs to group together nearby transactions, therefore reducing the search space of the solution. Our group formation phase accounts for QID similarity, and builds anonymized groups that preserve correlation.

In each anonymized group, sensitive items are separated from the QID, and published in a separate summary table, as shown in Fig. 1c. This publishing format is similar to permutation methods, such as Anatomy. However, as opposed to relational data, where all records have the same number of sensitive attributes, a transaction can contain any number of sensitive items. We elaborate this further in Section II. In our example, Fig. 1c shows how two groups with high intra-group correlation are formed: {*Bob, David, Ellen*}, all with probability $1/3$ of buying *viagra*, and {*Andrea, Claire*} with probability of buying *pregnancy test* $1/2$.

Intuitively, our approach addresses the concern of high data dimensionality by anonymizing each group of transactions according to a *relevant* quasi-identifier, consisting of items that exist in the group. The underlying assumption is that each transaction can be re-identified based on its items, which are a small subset of the entire item space. This has the potential of circumventing the dimensionality curse, by not using a unique, high-dimensional QID item combination for all groups. Although each transaction has a low number of items, they are distributed in the entire item space, so the challenge is how to effectively group together transactions with similar QID. Our band matrix organization tackles this challenge by placing transactions with similar QID in close proximity.

Our specific contributions are:

- we introduce a novel representation of transaction data which takes advantage of data sparseness, preserves correlations among items and arranges transactions with similar QID in close proximity to each other
- we devise an efficient heuristic to create anonymized groups with low information loss

- we evaluate experimentally our method with real dataset workloads, and show that it clearly outperforms existing state-of-the-art in both data utility and computational overhead

The rest of the paper is organized as follows: in Section II, we introduce fundamental concepts and definitions. In Section III, we outline our proposed data organization technique, and present effective methods to achieve it. In Section IV, we present our heuristic to create anonymized groups of transactions. In Section V, we experimentally evaluate our technique. We survey related work in Section VI. We conclude with directions for future work in Section VII.

II. Background

Our objective is to anonymize data consisting of a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$, $n = |T|$. Each transaction $t \in T$ contains items from an item set $I = \{i_1, i_2, \ldots, i_d\}$, $d = |I|$. We represent the data as a binary matrix A with n rows and d columns, where

$$A[i][j] = \begin{cases} 1, & i_j \in t_i \\ 0, & i_j \notin t_i \end{cases}$$

For instance, the matrix associated to the data in Fig. 1a is

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Among the set of items I, some are privacy-sensitive, such as *pregnancy test* or *viagra* in our running example.

Definition 1 (Sensitive Items): The set $S \in I$ of items that represent a privacy threat if associated to a certain transaction, constitutes the *sensitive items* set, $S = \{s_1 \ldots, s_m\}$, $m = |S|$.

The rest of the items in I, such as *wine, cream* etc, are not sensitive, in the sense that their association with a certain individual is not detrimental. On the other hand, these innocuous items can be used by an attacker to re-identify individual transactions, as shown in our introductory section. We denote these items by *quasi-identifier (QID)* items.

Definition 2 (Quasi-identifier Items): The set of items in I that an attacker can gain knowledge on in order to re-identify individual transactions constitute the set of *quasi-identifiers*. Potentially, any non-sensitive item is a quasi-identifier, hence $Q = I \setminus S = \{q_1, \ldots, q_{d-m}\}$.

We denote a transaction which contains items from S as *sensitive transaction*, and one which contains only items from Q as *non-sensitive*.

In previous work on privacy preservation of relational data, the underlying assumption is that a single, fixed schema exists, and all records abide the schema, therefore a single QID is used for all records. However, such an approach is not suitable for the problem at hand due to the high dimensionality of the data. On the other hand, the data is sparse, and each transaction can be re-identified based on a small number of items. For a given transaction $t \in T$, we define the *relevant quasi-identifier* Q^t as the intersection of Q with the items in t. For instance, in the example in Fig. 1, $Q^{Bob} = Q^{David} = \{Wine, Meat\}$ and $Q^{Claire} = \{Cream, Strawberries\}$. In Section III, we will show how to re-organize the data effectively, such that transactions with similar relevant QID are in close proximity to each other.

A. Privacy Requirements

Two main privacy-preserving paradigms have been established for relational data: k-anonymity [7], which prevents identification of individual records in the data, and ℓ-diversity [1], which prevents the association of an individual record with a sensitive attribute value. ℓ-diversity is a more suitable paradigm, since it is the association of individuals with sensitive information that ultimately threats privacy. Similarly, in the case of transactional data, the privacy threat is defined as the association of an individual transaction to a sensitive item.

Nevertheless, the privacy preservation of transactional data is slightly different from its relational database counterpart, where all records have the same number (usually one) of sensitive attributes. In our case, some transactions may not contain any sensitive item, hence are completely innocuous, while others may have multiple sensitive items.

Definition 3 (Privacy): A privacy-preserving transformation of transaction set T has *privacy degree* p if the probability of associating any transaction $t \in T$ with a particular sensitive item $s \in S$ does not exceed $1/p$.

This is similar to saying that the transaction of an individual can be associated to a certain sensitive item with probability at most $1/p$ among $p-1$ other transactions.

Note that, the association of an individual with an item in Q does *not* represent a privacy breach: there is no detrimental information in the fact that Bob, for instance, has purchased *meat*. For this particular reason, the items in Q can be released directly, and we can employ a permutation-based approach, similar to [6], for privacy preservation. This has a considerable impact in reducing information loss, compared to generalization-based approaches.

We enforce the privacy requirement by partitioning the set T into disjoint sets of transactions, which we refer to as *anonymized groups*. For each group G, we publish the exact QID items, together with a summary of the frequencies of sensitive items contained in G. In our running example (Fig. 1c) the second group contains two transactions, corresponding to Andrea and Claire, and one occurrence for sensitive item *pregnancy test*. The probability of associating any transaction in G to that item is $1/2$. In general, let $f_1^G \ldots f_m^G$ be the number of occurrences for sensitive items $s_1 \ldots s_m$ in group G. Then group G offers privacy degree

$$p^G = \min_{i=1\ldots m} |G|/f_i$$

The privacy degree of an entire partitioning $\mathcal{P}$ of T is

$$p^{\mathcal{P}} = \min_{G \in \mathcal{P}} p^G$$

We further discuss aspects related to the utility of the data transformed in accordance to given privacy degree p.

B. Utility Requirements

It is well-understood [8] that publishing privacy-sensitive data is caught between the conflicting requirements of privacy and utility. In order to preserve privacy of transactional data, a certain amount of information loss is inherent. Nevertheless, the data should maintain a reasonable degree of utility.

Transactional data is mainly utilized to derive certain patterns, such as consumer purchasing habits. Returning to the running example, we can observe there are two sorts of patterns: those that involve items from Q alone, and those that involve at least one item in S. For the former category, thanks to the permutation-based publishing method we adopt, the information derived from the anonymized data is identical to that from the original data. For instance, we can derive that half the customers that bought *strawberries* have also bought *cream*.

On the other hand, when sensitive items are involved, the information derived from the anonymized data is only an estimation of the real one. In our running example, for instance, by inspecting the second anonymized group (Fig. 1c) we can infer with 50% probability that whoever buys *cream* and *strawberries* also buys a *pregnancy test*, whereas from the original data we can infer this with 100% probability. Patterns can be expressed as queries of the form

$$\begin{aligned} &SELECT\ COUNT(*)\ FROM\ T \\ &WHERE\ (SensitiveItem\ s\ is\ present) \\ &AND\ (q_1 = val_1) \wedge \ldots \wedge (q_r = val_r) \end{aligned} \tag{1}$$

The process of estimating the result of the query for each anonymized group G, is referred to as *data reconstruction*. Denote the number of occurrences of item s in G by a, and the number of transactions that match the QID selection predicate (last line of (1)) by b. Then the estimated result of the query, assuming each permutation of sensitive items to each QID combination in G is equally likely, is

$$a \cdot b/|G| \tag{2}$$

Note that, if all transactions in G have identical QID, then either $b = |G|$ or $b = 0$, and the reconstruction error is 0. Ideally, to minimize reconstruction error, we need to minimize $|G| - b$, hence include in each group transactions with similar, and when possible identical, QID.

A meaningful way of modeling such queries that involve sensitive items is to use a *probability distribution function (pdf)* of an item $s \in S$ over the space defined by a number of r items in Q. In the running example, assume that a data analyst wishes to find the correlation between *pregnancy test* and quasi-identifier items *cream* and *meat*. Fig. 2 represents this scenario: every cell corresponds to a combination of the QID items, for instance $(1, 0)$ corresponds to transactions which contain *cream* but not *meat*, whereas $(1, 1)$ to transactions with both *cream* and *meat*. From the original data, we can infer that all customers who bought *cream* but not *meat* have also bought a *pregnancy test*. However, from the anonymized data, we can only say that half of such customers have bought a *pregnancy test*.

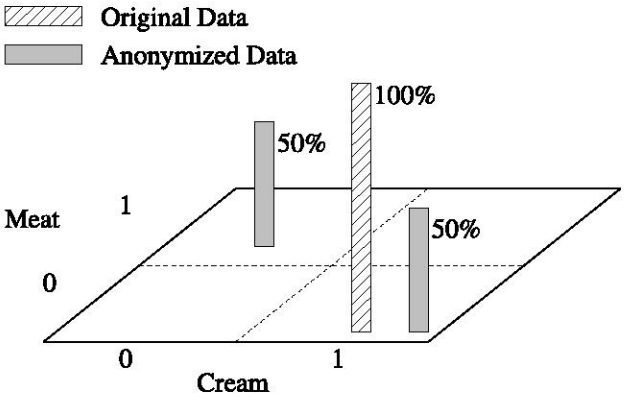


Fig. 2. Data Reconstruction

If the query to be evaluated includes r QID items, the total number of cells is 2^r, corresponding to all combinations when an item is or is not present in a transaction (this is the same as having a "group-by" query on items $q_1 \ldots q_r$). The actual pdf value of sensitive item s for a cell C is

$$Act^s_C = \frac{Occurrences\ of\ s\ in\ C}{Total\ Occurrences\ of\ s\ in\ T}$$

The estimated pdf Est^s_C is computed similarly, except that the numerator consists of eq.(2) summed over all groups that intersect cell C. We determine the utility of the anonymized data as the distance between the real and estimated pdf over all cells, measured by *KL-divergence*, already established [8] as a meaningful metric to evaluate the amount of information loss incurred by data anonymization:

$$KL\_Divergence(Act, Est) = \sum_{\forall cell\ C} Act^s_C \log \frac{Act^s_C}{Est^s_C}$$

If Act is identical to Est, $KL\_Divergence = 0$.

With both privacy and utility issues clarified, we give our **Problem Statement.** Given a set of transactions T containing items from I, and a subset $S \subset I$ of sensitive items, determine a partitioning $\mathcal{P}$ of T into anonymized groups with privacy degree at least p, such that the reconstruction error, measured by KL-divergence, is minimized.

III. Reduction to Band Matrix Representation

As discussed in Section II-B, in order to minimize the reconstruction error, it is necessary to group together transactions with similar relevant QID. We organize the data (i.e. matrix A) as a *band matrix*, so that consecutive rows are likely to share a large number of common items. Band matrix organization has been acknowledged as a beneficial mode to represent sparse data in various scientific applications [9], [10]. A band matrix has the general form shown in Fig. 3, where all elements of the matrix are **0**, except for the main diagonal d_0, a number of U upper diagonals $(d_1 \ldots d_U)$, and L lower diagonals $(d_{-1} \ldots d_{-L})$. U represents the *upper bandwidth*

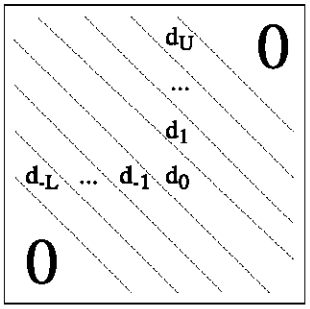


Fig. 3. Band Matrix Representation

Reverse Cuthill-McKee (RCM) Algorithm
Input: graph $\mathcal{G}(V, E)$ with adjacency matrix A
1. pick peripheral vertex $v \in V$ (compute pseudo-diameter)
2. $R = \{v\}$
3. $PrevLevel = R$
4. **while** $|R| < |V|$ **do**
5. $\quad CrtLevel = \emptyset$
6. $\quad$ **for** $i = 1$ *to* $|PrevLevel|$ **do**
7. $\quad\quad Tmp = \{v \in V | v \notin R \wedge dist(PrevLevel[i], v) = 1\}$
8. $\quad\quad$ sort Tmp in increasing order of vertex degree
9. $\quad\quad$ append Tmp to $CrtLevel$
10. $\quad$ **endfor**
11. $\quad$ append $CrtLevel$ to R
12. $\quad PrevLevel = CrtLevel$
13. **endwhile**
14. output R in reverse order

Fig. 4. Reverse Cuthill-McKee Algorithm

of the matrix and L the *lower bandwidth*. Our objective is to minimize the total bandwidth $B = U + L + 1$. A simple Gaussian elimination algorithm can be employed to obtain an upper or lower *triangular* matrix, where $L = 0$ or $U = 0$, respectively. However, finding an optimal band matrix, i.e. with minimum B, is NP-complete [11].

The Reverse Cuthill-McKee Algorithm. A general matrix can be transformed into a band matrix by performing permutations of rows and columns. Multiple heuristics have been proposed to obtain band matrices with low bandwidth. The most prominent is the *Reverse Cuthill-McKee (RCM)* algorithm, a variation of the Cuthill-McKee algorithm [12]. RCM works for square, symmetric matrices. Given sparse matrix A, it builds graph $\mathcal{G} = (V, E)$, where V contains one vertex for each matrix row, and there is an edge from vertex v_i to vertex v_j for every non-zero element $A[i][j]$. If A is symmetric, then $\mathcal{G}$ is undirected. RCM is based on the observation that a permutation of rows of A corresponds to a re-labeling of vertices for $\mathcal{G}$. Given a re-labeling (permutation) δ of V (i.e. a bijective application $\delta : \{1 \ldots |V|\} \rightarrow \{1 \ldots |V|\}$) , then the bandwidth of $\mathcal{G}$ (hence of matrix A with rows permuted according to δ) is

$$B(\mathcal{G}) = max\{|\delta(v_1) - \delta(v_2)| : (v_1, v_2) \in E\}$$

Unsymmetric Matrix Bandwidth Reduction
Input: unsymmetric matrix A
1. compute symmetric matrix $B = A \times A^T$
2. δ = Reverse Cuthill-McKee(B)
3. A' = permutation δ applied to A
4. output A'

Fig. 5. Unsymmetric Matrix Bandwidth Reduction

To determine a permutation that reduces B, RCM performs a breadth-first (BFS) traversal starting from an initially chosen *root* node. All nodes (i.e. rows) at the same distance from the root in the traversal constitute a *level set*. At each step, the vertices in the same level sharing the same parent are sorted increasingly according to vertex degree. By reversing the obtained order, we find the permutation that needs to be applied to the rows of matrix A. The selection of the root node is essential for the effectiveness of the transformation; usually, the root is determined by finding a pseudo-diameter of the graph (through an iterative process linear in the number of vertices) and choosing one of the ends. Fig. 4 shows the RCM algorithm pseudocode.

RCM is currently implemented in matrix routine libraries, such as MATLAB. The computational complexity of the RCM algorithm is $O(|V|D \log D)$, where D is the maximum degree of any vertex in the adjacency list. More recently, a linear time implementation to the size of adjacency list E has been proposed in [13].

Bandwidth Reduction for Unsymmetric Matrices. RCM only addresses the case of symmetric matrices. A recent work [10] investigates several approaches to reduce the bandwidth of unsymmetric matrices, based on the RCM algorithm. Two methods can be employed to achieve this, and, given unsymmetric matrix A, they consist of applying RCM to one of the following symmetric matrices: *(i)* $A + A^T$ and *(ii)* $A \times A^T$. The obtained permutation δ is then applied to A.

Method *(i)* is suitable mainly in cases where A is almost symmetric. This method is rather inexpensive, but the obtained result may not have good quality, especially if the original matrix is far from symmetric. Method *(ii)* can be applied to any arbitrary matrix. Computing $A \times A^T$ incurs an additional computational overhead, but the quality of the solution is much better (i.e. the resulting bandwidth is considerably smaller). Furthermore, note that we are not exactly performing a full matrix multiplication, since we are only interested in non-zero entries, therefore the overhead may not be substantial. For these reasons, we only consider this method further. Fig. 5 shows the pseudocode of the band reduction algorithm for unsymmetric matrices.

Evaluation of RCM. We exemplify the effectiveness of the unsymmetric RCM algorithm using a synthetic workload, which allows us to control the variation in data correlation, and highlight its impact on the RCM technique. We use the *IBM Quest Market-Basket Synthetic Data Generator*[1] to obtain

[1] http://www.almaden.ibm.com/cs/projects/

datasets, and the MatView[2] tool to visualize sparse matrices. To facilitate visualization, we consider square matrices of 1000 transactions and 1000 items, with an average of 20 items per transaction. We consider three degrees of data correlation (i.e. correlation among patterns of items): 0.1, 0.5 and 0.9 (low, medium and high correlation respectively). Fig. 6 shows the original matrices on the left-hand side, and the transformed matrices, using unsymmetric RCM, on the right-hand side. For the original matrices, we can observe the varying degree of data correlation: when correlation is high (Fig. 6c left), data tends to cluster into vertical lines, corresponding to items that appear in many transactions. Still, the items are distributed in the entire item space among consecutive transactions, so the representation does not properly capture the correlation.

On the right-hand side, we can observe how RCM captures data correlation, arranging transactions with similar items in close proximity. The larger the correlation, the more effective the bandwidth reduction is. Therefore, grouping together neighboring transactions into anonymized groups is likely to preserve the correlation between items, and hence increase data utility.

For the problem we study, the number of transactions n is likely to be larger than the number of items d. In this case, we can easily obtain an $n \times n$ square matrix by padding it with **0** columns (i.e. "fake" items). The algorithm remains unchanged, and the padding does not affect its complexity.

IV. Anonymized Group Formation

Once the data is transformed according to RCM, the next step is to create anonymized groups of transactions. To fulfill the privacy requirement, each sensitive transaction needs to be grouped with non-sensitive transactions, or sensitive ones with different sensitive items. We propose *CAHD (Correlation-aware Anonymization of High-dimensional Data)*, a greedy heuristic that capitalizes on the data correlation, and groups together transactions that are close-by in the band matrix representation.

Consider the example in Fig. 7, where non-sensitive transactions are shown light-shaded, and sensitive ones dark-shaded. CAHD scans transaction set T in row order, finds the first sensitive transaction in the sequence, and attempts to form an anonymizing group for it.

We say that two transactions are *conflicting* if they have at least one common sensitive item. In our example, t_0 and t_1 are conflicting, and are represented with similar hatching. t_2 is not in conflict with any of t_0 or t_1. Assume that we want to anonymize t_0 with privacy degree p. In that case, we need to group it with at least $p-1$ different transactions. We choose to adopt a *"one-occurrence-per-group"* heuristic that allows only one occurrence of each sensitive item in a group. We will show shortly that, if a solution to the anonymization problem with privacy degree p exists, then such an heuristic will always find a solution.

[2]http://www.csm.ornl.gov/ kohl/MatView/

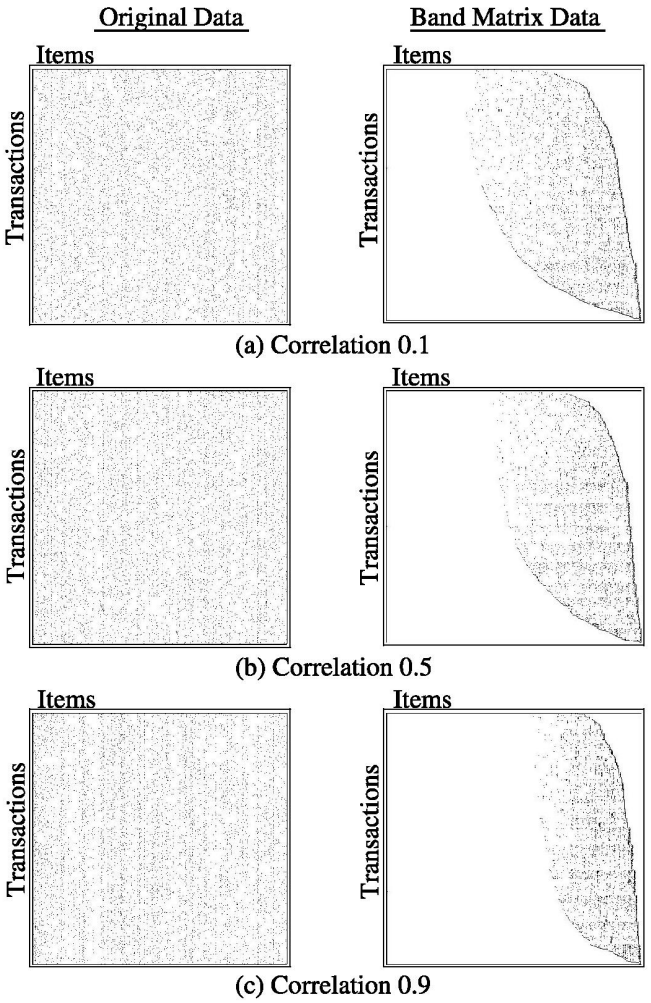


Fig. 6. Effectiveness of RCM Algorithm

CAHD works as follows: given sensitive transaction t_0, a *candidate list (CL)* is formed, with the αp transactions that precede, respectively follow t_0, and are not in conflict with t_0 or with each other (conflicting transactions are "skipped" when building CL). $\alpha \in \mathbb{N}$ is a system parameter which restricts the range of the search. Intuitively, the larger α, the better the chance to include in CL transactions with similar items, but at increased execution time. Nevertheless, as we show in Section V, even a low α can yield good results, thanks to the effective band matrix organization. Note that t_1 is excluded from $CL(t_0)$, and its predecessor (which is non-sensitive, hence not in conflict with t_0) is included. Then, out of the $2\alpha p$ transactions in $CL(t_0)$, the $p-1$ of them that have the largest number of QID items in common with t_0 are chosen to form an anonymized group. The intuition is that, the more transactions share the same QID, the smaller the reconstruction error is (see eq. (2)). All selected transactions are then removed from T, and the process continues with the next sensitive transaction in the order. Fig. 8 gives the pseudocode of CAHD.

Since we use a greedy heuristic, we must ensure that our method finds a solution, i.e. at any time, forming a group will not yield a remaining set of transactions that can not be

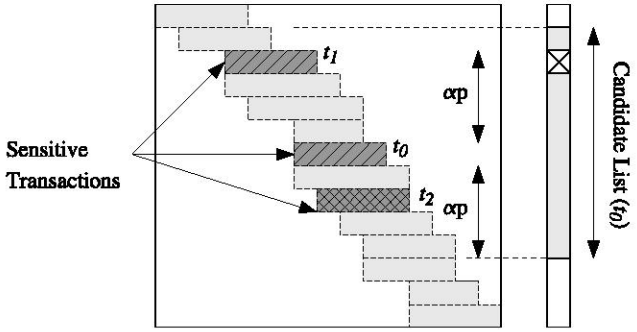


Fig. 7. Group Formation Heuristic

CAHD Group Formation Heuristic

Input: transaction set T, privacy degree p

1. initialize histogram H for each sensitive item $s \in S$
2. $remaining = |T|$
3. **while** $(\exists t \in T | t\ is\ sensitive)$ **do**
4. t = next sensitive transaction in T
5. $CL(t)$ = non-conflicting αp pred. and αp succ. of t
6. $G = \{t\} \cup p-1$ trans. in $CL(t)$ with closest QID to t
7. update H for each sensitive item in G
8. **if** $(\nexists s | H[s] \cdot p > remaining)$
9. $remaining = remaining - |G|$
10. **else**
11. roll back G and continue
12. **end while**
13. output remaining transactions as a single group

Fig. 8. CAHD Pseudocode

anonymized (for instance, if all remaining transactions share one common sensitive item). For this reason, we maintain a histogram with the number of remaining occurrences for each sensitive item. The histogram is initialized when the data is read (line 1), and is updated every time a new group is formed (line 7). Upon validating a group, we check (line 8) that the remaining set of transactions satisfies the privacy requirement. If not, the current group is not validated, and a new group formation is attempted starting from the next sensitive transaction in the sequence.

The algorithm stops when there are no more ungrouped sensitive transactions remaining, or when no new groups can be formed. If there remain un-grouped transactions, these are published as a single group. It is guaranteed, due to our group validation check, that this group satisfies the degree of privacy p. Furthermore, if all remaining transactions are non-sensitive, there is no information loss incurred if they are published as a single group, regardless of their number, since we publish the QID directly.

To implement efficiently the assembly of CL (i.e. find αp non-conflicting transactions) we can employ a linked-list data representation, where each transaction (list entry) points to its predecessor and successor with a particular sensitive item. The space requirement is $O(m)$ (where $m = |S|$, i.e. constant) per transaction, and the computational complexity of the algorithm is $O(pn)$.

TABLE I
DATASET CHARACTERISTICS

	Transactions	Items	Max. length	Avg. length
BMS1	59,602	497	267	2.5
BMS2	77,512	3,340	161	5.0

TABLE II
RE-IDENTIFICATION PROBABILITY

	Number of Known QID Items			
Dataset	1	2	3	4
BMS1	0.3%	9.5%	24.3%	50.0%
BMS2	0.8%	18.8%	41.6%	91.1%

V. EXPERIMENTAL EVALUATION

To evaluate our anonymization technique, we use a workload consisting of two representative real-world datasets, introduced in [14]. *BMS-WebView-1 (BMS1)* and *BMS-WebView-2 (BMS2)* represent several months of transaction logs corresponding to two on-line retailers[3]. Their characteristics are presented in Table I.

First, we reinforce our motivation in Section I, and we measure the probability of re-identifying individual transactions based on a number of known QID items. Table II shows the identification probability based on a randomly chosen set of QID items, with cardinality varying between 1 and 4. For the sparser BMS2 dataset, the re-identification probability with four known items is over 90%.

In the rest of the section, we evaluate the effectiveness (i.e. utility) and efficiency (execution time) of CAHD. As a competitor to our method, we consider a hybrid approach which combines the strengths of the current state-of-the-art: Mondrian [4] and Anatomy [6]. Mondrian is a generalization method: it recursively divides the dataset, based on QID values, until the privacy requirement does not allow any more splits. Mondrian preserves locality in the QID space, but because it generalizes the QID values within each group, it may incur considerable information loss. On the other hand, *Anatomy* is a permutation approach, which publishes directly QID values. This is beneficial for utility; however, in the process of group formation, Anatomy does not account for QID proximity, therefore it may not preserve well correlations. We compare against a combined method we denote by *PermMondrian (PM)*: similar to Mondrian, PM partitions the dataset according to QID proximity. Nevertheless, it publishes exact QID values, instead of generalizing them. Furthermore, we use an enhanced split heuristic, compared to the original one in [4], which considered only group cardinality. To allow PM to obtain fine-grained groups that enhance utility, we take into account the distribution of sensitive items in the resulting groups upon each split: we favor splits with a balanced distribution of sensitive items, since this increases the success probability of subsequent split attempts. Otherwise, many transactions with common sensitive items could be grouped

[3]These datasets have also been used as benchmarks in KDD-Cup 2000

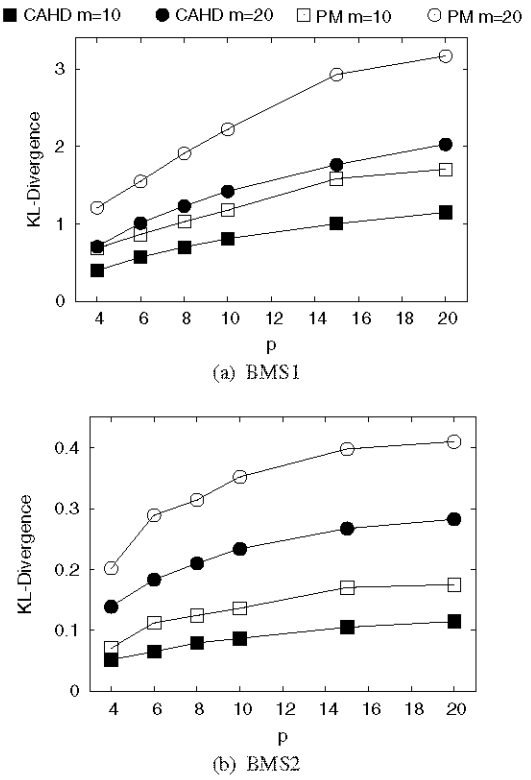


Fig. 9. Reconstruction Error vs p ($r = 4$)

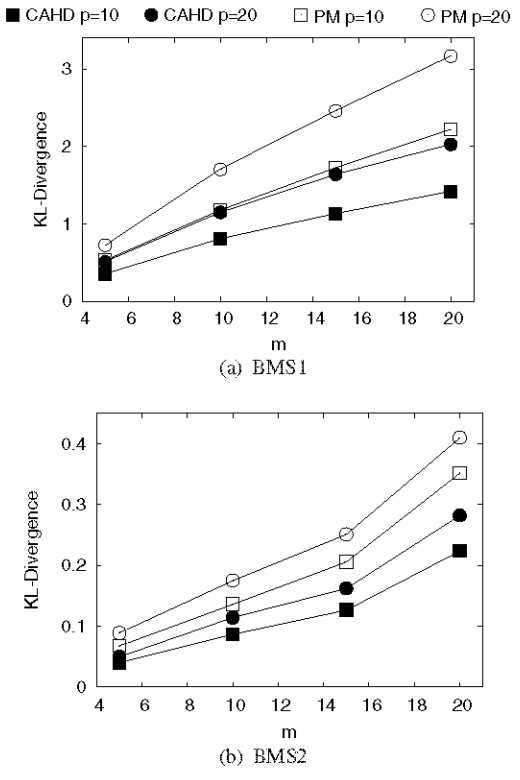


Fig. 10. Reconstruction Error vs m ($r = 4$)

together, increasing the frequency of a single item in a group, hence disallowing further splits.

We vary the degree of privacy p in the range $4 - 20$. We randomly choose a number of sensitive items m (i.e. cardinality of S) between 5 and 20, out of the entire set of items I. We consider group-by queries as discussed in Section II-B, and we vary the number r of QID items in the group-by clause between 2 and 8. For each (p, m, r) triplet setting, we generate 100 group-by queries by randomly selecting s and $q_1 \ldots q_r$, and we determine the average reconstruction error (i.e. KL-divergence, with ideal value 0). Unless otherwise specified, we set parameter $\alpha = 3$ (see Section IV).

First, we fix $r = 4$ and vary the privacy degree p. Fig. 9 shows that CAHD outperforms PM for both datasets, by up to a factor of 2. As expected, a higher privacy degree p increases the reconstruction error, i.e. reduces data utility.

In the next experiment, we determine the utility variation when the number of sensitive items m is varied. Fig. 10 shows that, again, CAHD is superior to PM, for both $p = 10$ and $p = 20$ settings. In fact, the utility given by CAHD for the considerably more restrictive $p = 20$ setting, is superior to that of PM for $p = 10$.

Next, we vary parameter r, for a fixed $m = 10$. Again, as shown in Fig. 11, we outperform PM in all cases. Note that, the larger the value of r, the larger the difference becomes between the two methods. In practice, this translates into the fact that PM is unable to preserve correlations among patterns of larger length. Also, for $r \geq 3$, CAHD with the more restrictive $p = 20$ setting outperforms even PM with $p = 10$.

Fig. 12 shows the execution time of CAHD and PM for $m = 20$ and varying p, which is the most influential factor on runtime performance. CAHD is time-efficient, with completion time of at most 5 and $15sec$ (not including RCM execution time) for the BMS1 and BMS2 datasets, respectively. A more substantial overhead is incurred by RCM execution, which requires 158 and respectively $457sec$ for the two datasets. However, this overhead is only incurred for transforming the input once, regardless of p values. PM only manages execution times in the range of 300 and $1500sec$ for the two datasets, respectively. Since PM is a top-down partitioning method, it shows a slight decrease in computational overhead as p increases.

Finally, we investigate the effect of varying parameter α, which determines the search range of CAHD in the process of assembling candidate lists. Fig. 13 shows how data utility and execution time evolve with α for the BMS2 dataset, $m = 10$ (similar trends were observed for BMS1, we omit those results for brevity): there is a slight decrease in information loss as α increases (i.e. when the search range for candidate lists expands). Nevertheless, the gain is not considerable, which proves once again that the band matrix representation is able to re-arrange highly-correlated transactions in close proximity. On the other hand, execution time increases linearly with α. We conclude that α values of 2 or 3 are a good compromise in practice.

VI. Related Work

Privacy-preserving data publishing has received considerable attention in recent years, especially in the context of relational data. Ref. [15] employs random perturbation to

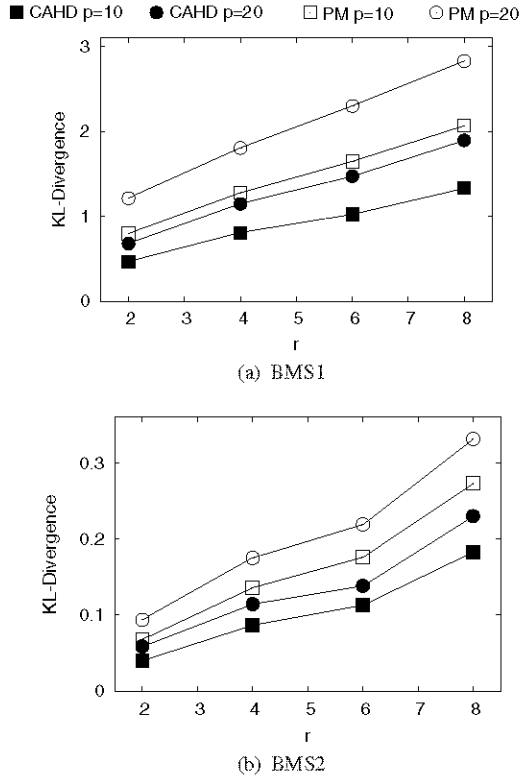


Fig. 11. Reconstruction Error vs r ($m = 10$)

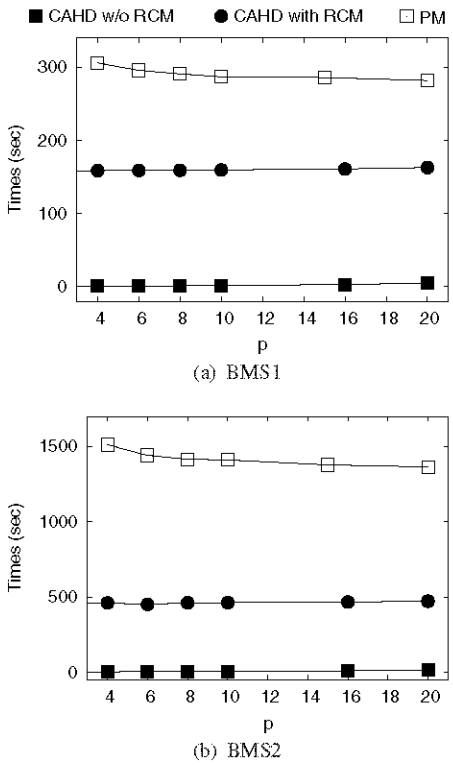


Fig. 12. Execution Time

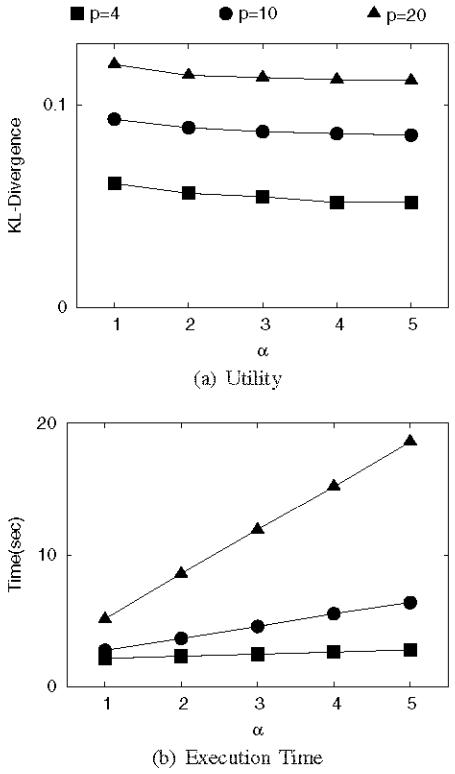
prevent re-identification of records, by adding noise to the data. Ref. [16] showed that an attacker could filter the random noise, and hence breach data privacy, unless the noise is correlated with the data. However, randomly perturbed data is not "truthful" [17], in the sense that it contains records which do not exist in the original data. Furthermore, random perturbation may expose privacy of outliers when an attacker has access to external knowledge.

Published data about individuals (*microdata*) may contain *quasi-identifier* attributes (QID), such as age, or zipcode, which may be joined with public databases (e.g. voting registration lists) to re-identify individual records. To address this threat, Sweeney [7] introduced k-anonymity, a privacy-preserving paradigm which requires each record to be indistinguishable among at least $k-1$ other records with respect to the set of QID attributes. Records with identical QID values form an *equivalence class*, or *anonymized group*. k-anonymity can be achieved through *generalization*, which maps detailed attribute values to value ranges, and *suppression*, which removes certain attribute values or records from the microdata. The process of data anonymization is called *recoding*, and it inadvertently results in information loss. Several privacy-preserving techniques have been proposed, which attempt to minimize information loss, i.e. maximize utility of the data. Ref. [2], [3] proposed optimal k-anonymity solutions for *single-dimensional* recoding, which performs value mapping independently for each attribute. Ref. [4] introduced *Mondrian*, an heuristic solution for *multi-dimensional* recoding, which performs mapping for the Cartesian product of multiple attributes. Mondrian outperforms optimal single-dimensional solutions, due to its increased flexibility in forming anonymized groups. Methods discussed so far perform *global* recoding, where a particular detailed value is always mapped to the same generalized value. In contrast, *local* recoding allows distinct mappings across different anonymized groups. Clustering-based local recoding methods are proposed in [18], [19].

k-anonymity prevents re-identification of individual records, but it is vulnerable to *homogeneity* attacks, where many (or all) of the records in an anonymized group share the same sensitive attribute (SA) value. ℓ-diversity [1] addresses this vulnerability, and creates anonymized groups in which at least ℓ SA values are "well-represented". Any k-anonymity technique can be adapted to account for SA value diversity, by changing the group validation condition. Nevertheless, k-anonymity techniques use generalization or suppression, and may result in high information loss, especially for high-dimensional QID. Ref. [20] proposes a framework for k-anonymous and ℓ-diverse transformations based on dimensionality mapping, which outperforms other generalization techniques[4]. However, dimensionality mapping is only effective for low-dimensional QID, hence the method is not suitable for transactional data.

Anatomy [6] introduced a novel approach to achieve ℓ-diversity: instead of generalizing QID values, it decouples the SA from its associated QID, and *permutes* the SA values among records. Since QID are published directly, the information loss is reduced. A similar approach is taken in [22]. However, neither of these methods account for correlation

[4]A similar dimensionality-mapping technique has been used for spatial anonymity in [21]



(a) Utility

(b) Execution Time

Fig. 13. Variable α

among QID and SA when forming anonymized groups. We also adopt a permutation approach for transactional data, but we create anonymized groups in a QID-centric fashion, therefore preserving correlation and increasing data utility. Furthermore, our novel data representation helps us tackle the difficult challenge of high-dimensional QID.

Privacy-preservation of transactional data has been acknowledged as an important problem in the data mining literature. However, existing works on the topic [23], [24] focus on publishing *patterns*, and not data. The patterns (or rules), are mined directly from the original data, and the resulting set of rules is sanitized to prevent privacy breaches. Such an approach has two limitations: *(i)* the usability of the data is constrained by the rules that the owner decides to disclose and *(ii)* it is assumed that the data owner has the resources and expertise to perform advanced data mining tasks, which may not be the case in practice. Publishing the *data*, instead of patterns, gives the recipient flexibility in choosing what rules to mine, and also allows for other types of data analysis, such as clustering. Furthermore, the processing cost does not have to be bared by the data owner.

VII. Conclusions

In this paper, we propose CAHD, an effective anonymization technique for sparse, high-dimensional data. CAHD relies on a novel data representation, in the form of a band matrix, which captures the correlation within the data. CAHD achieves superior data utility compared to existing state-of-the-art, and it also yields reduced computational overhead.

The necessity of anonymizing transactional data has been recently emphasized with the release of the "Netflix Prize" data[5], containing movie ratings of 500,000 subscribers. A recent study [25] shows that an attacker can re-identify 80% of the subscribers based on knowledge about 6 reviewed movies. In future work, we plan to address the problem of anonymization of high-dimensional data for non-binary databases. Another direction is to employ dimensionality-reduction techniques for more effective anonymization.

References

[1] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-Diversity: Privacy Beyond k-Anonymity." in *Proc. of ICDE*, 2006.

[2] R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization," in *Proc. of ICDE*, 2005, pp. 217–228.

[3] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-domain k-Anonymity," in *Proc. of ACM SIGMOD*, 2005, pp. 49–60.

[4] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," in *Proc. of ICDE*, 2006.

[5] C. C. Aggarwal, "On k-Anonymity and the Curse of Dimensionality." in *Proc. of VLDB*, 2005, pp. 901–909.

[6] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," in *Proc. of VLDB*, 2006.

[7] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[8] D. Kifer and J. Gehrke, "Injecting Utility into Anonymized Datasets," in *Proc. of ACM SIGMOD*, 2006, pp. 217–228.

[9] N. Gibbs, W. Poole, and P. Stockmeyer, "An algorithm for reducing the bandwidth and profile of a sparse matrix," *SIAM J. Numerical Analysis*, vol. 13, pp. 236–250, 1976.

[10] J. K. Reid and J. A. Scott, "Reducing the total bandwidth of a sparse unsymmetric matrix," *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 3, pp. 805–821, 2006.

[11] C. Papadimitriou, "The NP-completeness of the bandwidth minimization problem," *Computing*, vol. 16, pp. 263–270, 1976.

[12] E. Cuthill and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," in *24th National ACM Conference*, 1969, pp. 157–172.

[13] W. M. Chan and A. George, "A linear time implementation of the Reverse Cuthill-McKee algorithm," *BIT Numerical Mathematics*, vol. 20, no. 1, pp. 8–14, 1980.

[14] Z. Zheng, R. Kohavi, and L. Mason, "Real world performance of association rule algorithms," in *Proc. of KDD*, 2001, pp. 401–406.

[15] R. Agrawal and R. Srikant, "Privacy preserving data mining," in *Proc. of ACM SIGMOD*, 2000.

[16] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proc. of ACM SIGMOD*, 2005.

[17] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Workload-aware Anonymization." in *Proc. of KDD*, 2006, pp. 277–286.

[18] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering," in *Proc. of ACM PODS*, 2006, pp. 153–162.

[19] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. Fu, "Utility-Based Anonymization Using Local Recoding," in *Proc. of SIGKDD*, 2006, pp. 20–23.

[20] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast Data Anonymization with Low Information Loss," in *Proc. of VLDB*, 2007, pp. 758–769.

[21] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location-Based Identity Inference in Anonymous Spatial Queries," *IEEE TKDE*, vol. 19, no. 12, pp. 1719–1733, 2007.

[22] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate Query Answering on Anonymized Tables," in *Proc. of ICDE*, 2007.

[23] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, "Anonymity preserving pattern discovery," *VLDB Journal*, 2008 (to appear).

[24] V. Verykios, A. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, "Association rule hiding," *IEEE TKDE*, vol. 16, no. 4, pp. 434–447, 2004.

[25] A. Narayanan and V. Shmatikov, "How To Break Anonymity of the Netflix Prize Dataset," http://arxiv.org/abs/cs/0610105.

[5] http://www.netflixprize.com/faq