PCTA: Privacy-constrained Clustering-based Transaction Data Anonymization

Aris Gkoulalas-Divanis
IBM Research–Zurich
Rüschlikon, Switzerland
agd@zurich.ibm.com

Grigorios Loukides
Vanderbilt University
Nashville, TN, USA
grigorios.loukides@vanderbilt.edu

ABSTRACT

Transaction data about individuals are increasingly collected to support a plethora of applications, spanning from marketing to biomedical studies. Publishing these data is required by many organizations, but may result in privacy breaches, if an attacker exploits potentially identifying information to link individuals to their records in the published data. Algorithms that prevent this threat by transforming transaction data prior to their release have been proposed recently, but incur significant information loss due to their inability to accommodate a range of different privacy requirements that data owners often have. To address this issue, we propose a novel clustering-based framework to anonymizing transaction data. Our framework provides the basis for designing algorithms that explore a larger solution space than existing methods, which allows publishing data with less information loss, and can satisfy a wide range of privacy requirements. Based on this framework, we develop PCTA, a generalization-based algorithm to construct anonymizations that incur a small amount of information loss under many different privacy requirements. Experiments with benchmark datasets verify that PCTA significantly outperforms the current state-of-the-art algorithms in terms of data utility, while being comparable in terms of efficiency.

Categories and Subject Descriptors

H.2.7 [**Database Administration**]: Security, integrity, and protection; H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Privacy, Experimentation, Theory

Keywords

Anonymity, Privacy-preserving data mining, Transaction data, Clustering, Database utility

PAIS 2011 March 25, 2011, Uppsala, Sweden

1. INTRODUCTION

Transaction datasets containing information about individuals' behaviors or activities are commonly used in a wide spectrum of applications, including recommendation systems [6], e-commerce [38], and biomedical studies [22]. These datasets are comprised of records, called *transactions*, which consist of a set of items, such as the products purchased by a customer of a supermarket, or the diagnosis codes contained in the electronic medical record of a patient.

Unfortunately, publishing transaction data may lead to privacy breaches, even when explicit identifiers, such as individuals' names or social security numbers, have been removed prior data release. This occurs because potentially identifying information (e.g., the diagnosis codes given to an individual during a hospital visit [5]), can still be used to link an individual to her transaction in the published dataset. Consider, for example, releasing the dataset of Fig. 1(a), which records the items purchased by customers of a supermarket, after removing customers' names. This allows an attacker, who knows that *Anne* has purchased the items a, b, and c during her visit to a supermarket, to associate *Anne* with her transaction, since no other transaction in this dataset contains these 3 items together. This is a major privacy threat to individual's privacy, which needs to be addressed to comply with data sharing regulations (e.g., those that guide the sharing of health-related information [1, 2]) and legislation (e.g., the EU Directive on privacy and electronic communications [1]). Having associated *Anne* with her transaction, for example, allows an attacker to infer any other item purchased by her.

Several methods for anonymizing transaction data have been proposed recently [5, 11, 21, 23, 32, 33, 37], but they all produce solutions that incur a large amount of information loss. This is because these methods consider a small number of possible transformations to anonymize data and are unable to accommodate specific privacy requirements data owners often have. For instance, the method introduced in [14] assumes that an item in the original data, represented as a leaf-level node in a generalization hierarchy such as the one shown in Fig. 1(c), can only be replaced by a node lying in the path between itself and the root of the hierarchy, and that an attacker has knowledge of all items associated with an individual transaction. This may lead to producing data with unnecessarily low data utility, particularly when attackers have knowledge of some of the items that are associated with an individual, as is the case for transac-

[1] `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:EN:NOT`

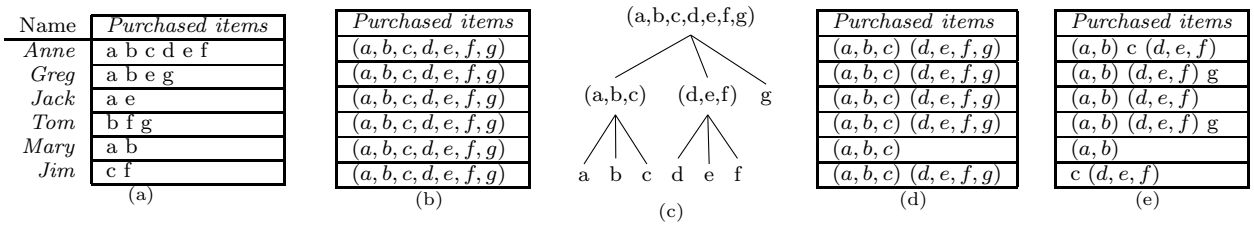

Figure 1: An example of: (a) original dataset, (b) output of Apriori Anonymization (AA), (c) item generalization hierarchy, (d) output of COAT, and (e) output of PCTA

tion data that are high-dimensional and sparse [21, 33, 37]. Note that these characteristics of transaction data make it difficult to use a gamut of methods that have been developed for anonymizing relational data, such as those proposed in [3, 4, 8, 10, 18, 24, 26, 30].

To anonymize transaction data with less information loss, we propose a novel clustering-based framework inspired from agglomerative clustering [36]. The framework we propose is independent of the way items are transformed and allows flexible algorithms that can anonymize transactions with low information loss under various privacy requirements to be developed. We also design PCTA, an effective and efficient algorithm that uses our clustering-based framework. PCTA explores a large number of possible data transformations, which helps produce data with less information loss, and exploits a lazy updating strategy, which is crucial to achieving efficiency. Through an extensive experimental evaluation, we verify that PCTA significantly outperforms the state-of-the-art algorithms in terms of retaining data utility, while also maintaining good scalability.

The rest of this paper is organized as follows. Section 2 provides the necessary background and introduces our clustering-based framework. In Section 3, we present our anonymization algorithm and, in Section 4, we evaluate our approach against the state-of-the-art algorithms for transaction data anonymization. Next, Section 5 discusses how PCTA can be extended to support utility and privacy requirements that are common in real-world applications. Finally, Section 6 concludes the paper.

2. BACKGROUND

In this section, we provide the background that is necessary to introduce our algorithm, given in Section 3. First, we review transformation strategies for anonymizing transaction data, discuss measures that capture the utility loss incurred by data transformation, and provide an overview of transaction data anonymization principles and algorithms. After that, we present a clustering-based formulation of the transaction data anonymization problem, based on which our algorithm is built.

2.1 Notation

Let $\mathcal{I} = \{i_1, ..., i_M\}$ be a finite set of literals, called *items*. Any subset $I \subseteq \mathcal{I}$ is called an *itemset* over $\mathcal{I}$, and is represented as the concatenation of the items it contains. An itemset that has m items or equivalently a *size* of m, is called an m-itemset, and its size is denoted with $|I|$. A dataset $\mathcal{D} = \{T_1, ..., T_N\}$ is a set of N transactions. Each *transaction* T_n, $n = 1, ..., N$, corresponds to a unique individual and is a pair $T_n = \langle tid, I \rangle$, where *tid* is a unique identifier and I is the itemset. A transaction $T_n = \langle tid, J \rangle$ *supports* an itemset I over $\mathcal{I}$, if $I \subseteq J$. Given an itemset I over $\mathcal{I}$ in $\mathcal{D}$, we use $sup(I, \mathcal{D})$ to represent the number of transactions $T_n \in \mathcal{D}$ that support I. This set is called the set of *supporting transactions* of I in $\mathcal{D}$.

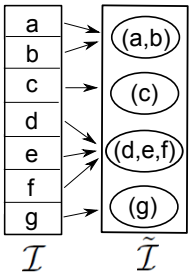

Figure 2: Mapping original to generalized items using global generalization.

2.2 Data transformation strategies

Constructing an anonymous transaction dataset is possible through techniques that transform items. One such technique is perturbation [7, 9], which operates by adding or deleting items from transactions with certain probability [29]. While data produced by perturbation can be used to build accurate data mining models, they cannot be analyzed at a record level, which is crucial in several applications, such as biomedical analysis [9, 22]. On the other hand, the techniques of suppression and generalization produce data that are not falsified. Both of these techniques can be applied either *globally*, in which case all items of the dataset undergo the same type of transformation, or *locally*, when items of certain transactions of the dataset are transformed. Suppression is an operation which removes items from the dataset before they are anonymized [37]. Global suppression is generally preferred, because it produces data in which all items have the same support as in the original dataset. This is important in building accurate data mining models using the anonymized data [37].

Generalization transforms an original dataset $\mathcal{D}$ to an anonymized dataset $\tilde{\mathcal{D}}$ by mapping original items in $\mathcal{D}$ to generalized items [21,33]. This technique often retains more information than suppression, as suppression is a special case of generalization where an original item is mapped to a generalized item that is not released [21]. As an example of applying generalization, consider the items a and b in Fig. 1(a), which are mapped to a generalized item (a, b) in the anonymized dataset of Fig. 1(e). The generalized item (a, b)

is interpreted as a, or b, or a and b and appears in the same transactions as those that have these items in the data of Fig. 1(a). This generalization is performed by a global generalization model, since a and b have been replaced by (a, b) in all transactions of Fig. 1(e).

As it can be easily observed, global generalization is essentially a mapping function from $\mathcal{I}$ to the space of generalized items $\tilde{\mathcal{I}}$, which is constructed by assigning each item $i \in \mathcal{I}$ to a unique generalized item $\tilde{i} \in \tilde{\mathcal{I}}$ that contains i. As an example, consider Fig. 2 which illustrates the mapping of original items, contained in the dataset of Fig. 1(a), to the anonymized items of the dataset shown in Fig. 1(e). Based on this mapping, the item a is mapped to the generalized item (a, b), and c to the generalized item (c)[2]. Observe also that the generalized item (a, b) appears in all the transactions that contained a and/or b before the anonymization.

2.3 Information loss measures

There are numerous ways to anonymize a transaction dataset, but the one that harms data utility the least, is typically preferred. To capture data utility, many criteria measure the information loss that is incurred by generalization based on item generalization hierarchies [32, 35]. The Normalized Certainty Penalty (*NCP*) measure, originally introduced in [35], has been employed in [32, 33]. *NCP* is expressed as the weighted average of the information loss of all generalized items, which are penalized based on the number of ascendants they have in the hierarchy. Other measures are the *multiple level mining loss* (ML^2), and *differential multiple level mining loss* (dML^2), which express utility based on how well anonymized data supports frequent itemset mining [32]. However, all the above measures require the items to be generalized according to hierarchies. A measure that can be used in the absence of hierarchies is *Utility Loss* (*UL*) [21], which is defined below.

Definition 2.1 (Utility loss). *The Utility Loss (*UL*) for a generalized item $\tilde{i}$ is defined as*

$$UL(\tilde{i}) = \frac{2^{|\tilde{i}|} - 1}{2^{|\mathcal{I}|} - 1} \times w(\tilde{i}) \times \frac{sup(\tilde{i}, \tilde{\mathcal{D}})}{N}$$

where $|\tilde{i}|$ denotes the number of items in $\mathcal{I}$ that are mapped to $\tilde{i}$, and $w : \tilde{\mathcal{I}} \rightarrow [0, 1]$ is a function assigning a weight according to the perceived usefulness of $\tilde{i}$ in analysis. Based on this definition, the Utility Loss *(UL) for a generalized dataset $\tilde{\mathcal{D}}$ is defined as $UL(\tilde{\mathcal{D}}) = \sum_{\forall \tilde{i} \in \tilde{\mathcal{I}}} UL(\tilde{i})$.*

UL quantifies information loss based on the size, weight and support of generalized items, imposing a "large" penalty on generalized items that are comprised of a large number of "important" items that appear in many transactions. The size is taken into account because $\tilde{i}$ can represent any of the $(2^{|\tilde{i}|} - 1)$ non-empty subsets of the items mapped to it. That is, the larger $\tilde{i}$ is, the less certain we are about the set of original items represented by $\tilde{i}$. The support of $\tilde{i}$ also contributes to the loss of utility, as highly supported items will affect more transactions, resulting more distortion. The denominators $(2^{|\mathcal{I}|} - 1)$ and N in Definition 2.1 are used for normalization purposes, so that the scores for *UL* are in $[0, 1]$. Moreover, a weight w is used to penalize generalizations exercised on more "important" items. This weight is specified by the data owner based on the perceived importance of the items to the subsequent analysis tasks. We note, however, that w can also be computed based on the semantic similarity of the items that are mapped to a generalized item [16, 21]. For example, to compute the *UL* score for $\tilde{i} = (a, b)$ in Fig. 1(e) assuming $w(\tilde{i}) = 1$, we have $UL(\tilde{i}) = \frac{2^2-1}{2^7-1} \times 1 \times \frac{5}{6} \approx 0.02$.

[2]We note that we may skip notation () from a generalized item when a single item is mapped to it.

2.4 Principles and algorithms for transaction data anonymization

In this section, we review the privacy principles that have been proposed in the transaction anonymization literature and explain why and how these principles offer privacy protection from the main threats in data publishing, namely identity [33] and sensitive itemset disclosure [23, 37]. We also survey algorithms that transform the original transaction data to satisfy these principles.

Identity disclosure. A well-established and widely used anonymization principle is k-anonymity, which was originally proposed for relational data [30,31], but has also been employed to protect many other types of data, including sequential [28], mobility [12,13,17,34], and graph data [20]. He et al. [14] applied a k-anonymity-based principle, called *complete k-anonymity*, to transaction datasets, requiring each transaction to be indistinguishable from at least $k - 1$ other transactions, as explained below.

Definition 2.2 (Complete k-anonymity). *Given a parameter k, a dataset $\mathcal{D}$ satisfies complete k-anonymity when $sup(I_j, \mathcal{D}) \geq k$, for each itemset I_j of a transaction $T_j = \langle tid_j, I_j \rangle$ in $\mathcal{D}$, with $j \in [1, N]$.*

Satisfying complete k-anonymity guarantees protection against identity disclosure, because it ensures that an attacker cannot link an individual to less than k transactions of the released dataset, even when this attacker knows all items of a transaction. To enforce this principle, He et al. [14] proposed a top-down algorithm, called *Partition*, that uses a local generalization model. *Partition* starts by generalizing all items to the most generalized item lying in the root of the hierarchy and then replaces this item with its immediate descendants in the hierarchy if complete k-anonymity is satisfied. In subsequent iterations, generalized items are replaced with less general items (one at a time, starting with the one that incurs the least amount of data distortion), as long as complete k-anonymity is satisfied, or the generalized items are replaced by leaf-level items in the hierarchy. As mentioned in the Introduction, *Partition* has two shortcomings that lead to producing data with excessive information loss: (i) it cannot be readily extended to accommodate various privacy requirements that data owners may have, since its effectiveness and efficiency depend on the use of complete k-anonymity, and (ii) it explores a small number of possible generalizations due to the hierarchy-based model it uses to generalize data.

Terrovitis et al. [33] argued that it may be difficult for an attacker to acquire knowledge about all items of a transaction, in which case protecting all items would unnecessarily incur excessive information loss. In response, the authors proposed the k^m-anonymity principle, defined as follows.

DEFINITION 2.3 (k^m-ANONYMITY). *Given parameters k and m, a dataset $\mathcal{D}$ satisfies k^m-anonymity when $sup(I, \mathcal{D}) \geq k$, for each m-itemset I in $\mathcal{D}$.*

A k^m-anonymous dataset offers protection from attackers who know up to m items of an individual, because it ensures that these items cannot be used to link this individual to less than k transactions of the released dataset. Terrovitis et al. [33] designed the Apriori algorithm to efficiently construct k^m-anonymous datasets. Apriori operates in a bottom-up fashion, beginning with 1-itemsets (items) and subsequently considering incrementally larger itemsets. In each iteration, the proposed algorithm enforces k^m-anonymity using the full-subtree, global generalization model [15]. The same authors have recently proposed two other algorithms to enforce k^m-anonymity [32], namely Vertical Partitioning Anonymization (VPA) and Local Recoding Anonymization (LRA). These algorithms operate in the following way. VPA first partitions the domain of items into sets and then generalizes items in each set to achieve k^m-anonymity. Next, the algorithm merges the generalized items to ensure that the entire dataset satisfies k^m-anonymity. LRA, on the other hand, partitions a dataset horizontally into sets in a way that would result in low information loss when the data is anonymized, and then generalizes items in each set separately, using local generalization. These algorithms are more flexible than *Partition* in the sense that they can be configured to offer protection against attackers who do not know all items of a transaction, but, contrary to our approach, still perform hierarchy-based generalization.

Loukides et al. [21] proposed a privacy principle that imposes a lower bound of k to the support of combinations of items that need to be protected from identity disclosure. Different from previous works, the approach of [21] limits the amount of allowable generalization for each item to ensure that the generalized dataset remains useful for specific data analysis requirements. To satisfy this principle, the authors of [21] proposed COAT, an algorithm that operates in a greedy fashion and employs both generalization and suppression. The choice of the items generalized by COAT is governed by utility constraints that model data analysis requirements and correspond to the most generalized item that can replace a set of items. Thus, COAT allows constructing any generalized item that is not more general than an owner-specified utility constraint. When such an item is not found, COAT selectively suppresses a minimum number of items from the corresponding utility constraint to ensure privacy. Our method is similar to COAT in that it addresses the aforementioned limitations of the approaches of [14,32,33], but it significantly outperforms COAT in terms of retaining data utility due to the use of clustering-based heuristics, as our experiments verify.

Sensitive itemset disclosure. Beyond identity disclosure is the threat of sensitive itemset disclosure, in which an individual is associated with an itemset that reveals some sensitive information, e.g., purchased items an individual would not be willing to be associated with. The afore-mentioned principles do not guarantee preventing sensitive itemset disclosure, since a large number of transactions that have the same generalized item can all contain the same sensitive itemset. To guard against this type of inferences, several approaches have been recently proposed. Ghinita et al. [11] developed an approach that releases transactions in groups, each of which contains public items in their original form and a summary of the frequencies of the sensitive items, while Cao et al. [5] introduced ρ-uncertainty, a privacy principle that limits the probability of inferring any sensitive itemset and a greedy algorithm to enforce it. The proposed algorithm for ρ-uncertainty iteratively suppresses sensitive items and then generalizes non-sensitive ones using the generalization model of [33].

Identity and sensitive itemset disclosure. Different from [11] and [5], which provide no protection guarantees against identity disclosure, the works of [37] and [23] are able to prevent both identity and sensitive itemset disclosure. In particular, Xu et al. [37] proposed (h, k, p)-coherence, a privacy principle which treats public items similarly to k^m-anonymity (the function of parameter p is the same as m in k^m-anonymity) and additionally limits the probability of inferring any sensitive item using a parameter h. More recently, Loukides et al. [23] examined how to anonymize data to ensure that owner-specified itemsets are sufficiently protected. The authors proposed the notion of PS-rules to effectively capture privacy protection requirements and designed a generalization-based anonymization algorithm. This algorithm operates in a top-down fashion, starting with the most generalized transaction dataset, and then gradually replaces generalized items with less general ones, as long as the data remain protected.

Our approach focuses on guarding against identity disclosure but can be easily extended to additionally prevent sensitive itemset disclosure, as we will discuss in Section 5.

2.5 Achieving anonymity through clustering

In this section, we model the task of anonymizing transaction data as a clustering problem. The latter problem requires assigning a label to each record of a dataset so that the records that are similar, according to an objective function, are assigned the same label. A series of papers, such as [4,19,24], have shown that anonymized relational datasets can be constructed based on clustering. In these approaches, records that incur low information loss when anonymized end up in the same cluster, and each cluster needs to contain at least k records to satisfy k-anonymity.

To anonymize a transaction dataset $\mathcal{D}$, in this work, we attempt to solve the following problem.

PROBLEM 2.1. *Construct a set of clusters $\mathcal{C}$ of generalized items such that: (i) each cluster $c \in \mathcal{C}$ corresponds to a unique generalized item, (ii) $\mathcal{C}$ satisfies the owner-specified privacy constraints, and (iii) the anonymized version $\tilde{\mathcal{D}}$ of $\mathcal{D}$, constructed based on $\mathcal{C}$, incurs minimal Utility Loss.*

We note that Problem 2.1 is fundamentally different from the one considered in [4, 19, 24]. First, clusters are built around generalized items, and not transactions. As a result, a cluster that represents a generalized item $\tilde{i}$ may be associated with more than one transactions, since it is associated with the supporting transactions of $\tilde{i}$ in $\tilde{\mathcal{D}}$. Second, instead of requiring all clusters to have at least k elements for achieving k-anonymity, we require the entire anonymized dataset $\tilde{\mathcal{D}}$ to adhere to a set of specified privacy constraints that can span clusters. A privacy constraint is modeled as a set of potentially linkable items from $\mathcal{I}$ and needs to be satisfied to thwart identity disclosure, as explained below.

DEFINITION 2.4. *A* privacy constraint $p = \{i_1, ..., i_r\}$ *is a set of potentially linkable items in* $\mathcal{I}$. *Given a parameter* k *of anonymity,* p *is* satisfied *in* $\tilde{\mathcal{D}}$ *when* $sup(p, \tilde{\mathcal{D}}) \geq k$.

Privacy constraints can be satisfied by using several models, such as complete k-anonymity [14] and k^m-anonymity [33], as explained in [21]. For example, consider the privacy constraint $\{a, d\}$ (which translates to "at least k transactions of the anonymized dataset should be associated with a, or d, or a and d") and that $k = 4$. This privacy requirement is satisfied in the anonymized data of Fig. 1(d), because the generalized item $(a, b, c) \cup (d, e, f, g)$ to which a and d are mapped, is supported by at least 4 transactions. It is also worth noting that an attacker does not gain any advantage by using subsets of p in linkage attacks, since, for every $p' \subseteq p$ and anonymized dataset $\tilde{\mathcal{D}}$, it holds that $sup(p', \tilde{\mathcal{D}}) \geq sup(p, \tilde{\mathcal{D}})$.

The clustering-based model we propose aims to satisfy privacy constraints by progressively merging clusters as in hierarchical agglomerative clustering algorithms do [36]. As one can observe, the support of a privacy constraint in $\tilde{\mathcal{D}}$ will either increase or remain the same as more items from $\mathcal{D}$ are mapped to the same generalized item $\tilde{i} \in \tilde{\mathcal{D}}$. This implies that a clustering that satisfies the specified privacy constraints will eventually be found by following a bottom-up approach that iteratively merges clusters formed by the items in $\mathcal{D}$, for any $k \in [2, N]$. This approach initially considers each original item as a singleton cluster and then iteratively merges singleton clusters (leading to the corresponding item generalizations) until the privacy constraints are met. Although there are alternative approaches, such as divisive methods that split large clusters in a top-down fashion, these approaches have been shown to incur more information loss than the bottom-up methods [35]. Since disparate item generalization decisions may incur a substantially different amount of information loss, the entire clustering process is driven by the *UL* measure, so that the two clusters that lead to minimizing information loss are merged at each step. Figures 3 and 4 illustrate this process through an example.

Assume that the original dataset of Fig. 4(a) needs to be anonymized to satisfy the privacy constraints $p_1 = \{i_1\}$ and $p_2 = \{i_5, i_6\}$ for $k = 3$. First, a set of singleton clusters are constructed, each built around one of the (generalized) items (i_1) to (i_7), so that the data of Fig. 4(a) are transformed as shown in Fig. 4(b). Since the specified privacy constraints are not satisfied in the dataset of Fig. 4(b), the current (singleton) clusters are subsequently merged. Among the different merging options, assume that merging the clusters for (i_1) and (i_2) incurs the minimum amount of utility loss, as measured by *UL*. This merging operation leads to a new cluster for the generalized item (i_1, i_2), which is associated with transactions T_1 and T_2. Note that the latter cluster will always have a higher *UL* score than each of the clusters from which it was constructed. Still, the dataset produced by this clustering, shown in Fig. 4(c), does not satisfy the privacy constraints, because (i_1, i_2) is associated with less than k transactions. As a next step, the clusters for (i_3) and (i_4) are merged to create the cluster (i_3, i_4) that has the lowest *UL* score. This produces the dataset of Fig. 4(d). After additional cluster merging operations, the clusters (i_1, i_2, i_3, i_4), (i_5, i_6), and (i_7), are obtained and not extended any further, as they correspond to the dataset of Fig. 4(e) which satisfies the specified privacy constraints.

An important benefit of adopting our clustering-based framework when designing anonymization algorithms is that it is independent of generalization models and anonymization requirements. This allows algorithms that exploit several generalization and privacy models to be developed. In terms of generalization models, the soft (overlapping) clustering solution that is produced in the transactions-space by our clustering-based model, leads to the generation of a cover instead of a partition of the original transactions, thus allowing each produced cluster to be anonymized differently. This is important because it can lead to anonymizations with significantly less information loss [32]. Furthermore, the proposed model can be easily employed to anonymize data that satisfies stringent privacy and utility constraints, as we discuss in Section 5. In any case, we note that finding the clustering that incurs the minimum information loss is an NP-hard problem (the proof follows from [21]), and thus one needs to resort to heuristics to tackle it.

3. PCTA ALGORITHM

Dealing with Problem 2.1 is possible by mapping original items to generalized ones to construct a clustering and then examining whether this clustering satisfies the specified privacy constraints. This is conceptually similar to how Apriori [33] and Partition [14] algorithms work. However, this strategy is likely to incur excessive information loss, because generalization is not "focused" on the items that are potentially linkable and need to be protected. For this reason, we opt for a different strategy that exploits the knowledge of which items need to be protected by targeting items contained in privacy constraints. Specifically, our strategy considers the imposed privacy constraints one at a time, selecting the privacy constraint p that is most likely to require a small amount of generalization in order to be satisfied. Then, it examines all possible cluster merging decisions that correspond to items in p and applies the one that leads to the minimum utility loss. The same process continues until the privacy constraint is satisfied, at which point the next non-satisfied privacy constraint is selected. By coupling this strategy with a novel lazy cluster-updating heuristic, we developed the Privacy-constrained Clustering-based Transaction Anonymization (PCTA) algorithm to anonymize transaction data with low utility loss. The pseudocode of PCTA is provided in Algorithm 1.

The algorithm works as follows. In steps 1 and 2, we initialize $\tilde{\mathcal{D}}$ to $\mathcal{D}$ and a priority queue PQ to the set containing all the specified privacy constraints $\mathcal{P}$. PQ orders the constraints with respect to their support in decreasing order and implements the usual operations *top*(), which retrieves the privacy constraint that corresponds to an itemset with the maximum support in $\tilde{\mathcal{D}}$ without deleting it from PQ, and *pop*(), which deletes the privacy constraint with the maximum support from PQ. In steps $3 - 27$, PCTA iteratively merges clusters to increase the support of each privacy constraint in PQ to at least k, so that the constraint is satisfied in $\tilde{\mathcal{D}}$. More specifically, we assign the privacy constraint that lies in the top of PQ to p (step 4) and update its items to reflect the generalizations that have occurred in previous iterations of PCTA (steps $5-11$). This lazy updating strategy significantly improves the runtime cost of PCTA, as experimentally shown in Section 4.3, since the generalized items that are needed to update p are retrieved without scanning

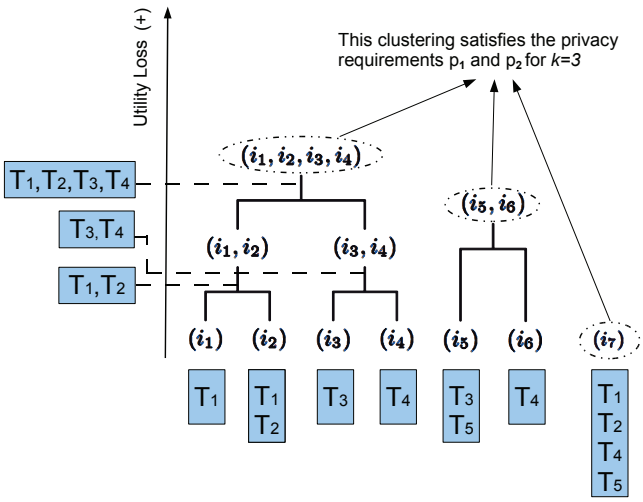

Figure 3: Data anonymization as a clustering problem.

tid	*Items*
T_1	$i_1\ i_2\ i_7$
T_2	$i_2\ i_7$
T_3	$i_3\ i_5$
T_4	$i_4\ i_6\ i_7$
T_5	$i_5\ i_7$

(a)

tid	*Items*
T_1	$(i_1)\ (i_2)\ (i_7)$
T_2	$(i_2)\ (i_7)$
T_3	$(i_3)\ (i_5)$
T_4	$(i_4)\ (i_6)\ (i_7)$
T_5	$(i_5)\ (i_7)$

(b)

tid	*Items*
T_1	$(i_1, i_2)\ (i_7)$
T_2	$(i_1, i_2)\ (i_7)$
T_3	$(i_3)\ (i_5)$
T_4	$(i_4)\ (i_6)\ (i_7)$
T_5	$(i_5)\ (i_7)$

(c)

tid	*Items*
T_1	$(i_1, i_2)\ (i_7)$
T_2	$(i_1, i_2)\ (i_7)$
T_3	$(i_3, i_4)\ (i_5)$
T_4	$(i_3, i_4)\ (i_6)\ (i_7)$
T_5	$(i_5)\ (i_7)$

(d)

tid	*Items*
T_1	$(i_1, i_2, i_3, i_4)\ (i_7)$
T_2	$(i_1, i_2, i_3, i_4)\ (i_7)$
T_3	$(i_1, i_2, i_3, i_4)\ (i_5, i_6)$
T_4	$(i_1, i_2, i_3, i_4)\ (i_5, i_6)\ (i_7)$
T_5	$(i_5, i_6)\ (i_7)$

(e)

Figure 4: Example of original data and its different anonymizations for Fig. 3.

Algorithm 1 PCTA($\mathcal{D}, \mathcal{P}, k$)

input: Dataset $\mathcal{D}$, set of privacy constraints $\mathcal{P}$, parameter k
output: Anonymous dataset $\tilde{\mathcal{D}}$

1. $\tilde{\mathcal{D}} \leftarrow \mathcal{D}$
2. $PQ \leftarrow$ privacy constraints of $\mathcal{P}$
3. **while** $(PQ \neq \emptyset)$
4. $\quad p \leftarrow PQ.top()$
5. $\quad$ **foreach** $(i_m \in p)$ *//lazy updating strategy*
6. $\quad\quad$ **if** $(H(i_m) \neq i_m)$
7. $\quad\quad\quad \tilde{i_m} \leftarrow H(i_m)$
8. $\quad\quad\quad$ **if** $(\tilde{i_m} \in p)$
9. $\quad\quad\quad\quad p \leftarrow p \backslash i_m$
10. $\quad\quad\quad$ **else**
11. $\quad\quad\quad\quad p \leftarrow (p \backslash i_m) \cup \tilde{i_m}$
12. $\quad$ **if** $(sup(p, \tilde{\mathcal{D}}) \geq k)$ *//p is protected*
13. $\quad\quad PQ.pop()$
14. $\quad$ **else** *// apply generalization to protect p*
15. $\quad\quad$ **while** $(sup(p, \tilde{\mathcal{D}}) < k)$
16. $\quad\quad\quad \mu \leftarrow 1$ *// maximum UL score*
17. $\quad\quad\quad$ **foreach** $(i_m \in p)$
18. $\quad\quad\quad\quad i_s \leftarrow \arg \min_{\forall i_r \in H, i_r \neq i_m} UL(\ (i_m, i_r)\ )$
19. $\quad\quad\quad\quad$ **if** $(UL(\ (i_m, i_s)\ ) < \mu)$
20. $\quad\quad\quad\quad\quad \mu \leftarrow UL(\ (i_m, i_s)\ )$
21. $\quad\quad\quad\quad\quad \sigma \leftarrow \{i_m, i_s\}$
22. $\quad\quad\quad \tilde{i} \leftarrow (i_m, i_s)$ *// generalize σ (cluster merging)*
23. $\quad\quad\quad$ update transactions of $\tilde{\mathcal{D}}$ based on σ
24. $\quad\quad\quad p \leftarrow (p \cup \{\tilde{i}\}) \backslash \sigma$
25. $\quad\quad\quad$ **foreach** $(i_r \in \sigma)$
26. $\quad\quad\quad\quad H(i_r) \leftarrow \tilde{i}$
27. $\quad\quad PQ.pop()$
28. **return** $\tilde{\mathcal{D}}$

the anonymized dataset. This leads to considerably better efficiency, particularly when many clusters need to be merged, as is the case for large k values. For this purpose, we use a hashtable H which has each item of $\tilde{\mathcal{D}}$ as key and the generalized item that corresponds to this item as value. Then, we remove the privacy constraint p from PQ if its support is at least k (step 13), in which case p is satisfied by the current clustering solution, or we merge clusters to protect it (steps $11-27$), if p is still unprotected in $\tilde{\mathcal{D}}$. In steps $16-21$, we select the best cluster merging decision among the clusters that affect the support of privacy constraint p. This is achieved by identifying the item i_m that can be generalized with another item i_s such that the resultant item σ incurs the least amount of information loss as measured by UL. When the best pair of clusters is found, PCTA performs the merging of the clusters by generalizing the items' pair σ to construct a new generalized item $\tilde{i}$ (step 22). Following that, the affected transactions in $\tilde{\mathcal{D}}$, the items in privacy constraint p, and the hashtable H, are all updated to reflect the new generalization (steps $23-26$). Steps $15-26$ are repeated until the support of p becomes at least k, in which case the current clustering satisfies the privacy constraint p. Then, p is removed from PQ in step 27. Finally, the dataset $\tilde{\mathcal{D}}$ is returned in step 28.

EXAMPLE 1. *To illustrate the operation of the PCTA algorithm, we apply it to the dataset of Fig. 1(a), assuming a single privacy constraint $p = \{a, b, e, f\}$ and $k = 3$. In steps 1 and 2, we initialize the anonymized dataset $\tilde{\mathcal{D}}$ to the original dataset $\mathcal{D}$ and add p to the priority queue PQ. Then, in step 4, we retrieve p from PQ and subsequently (steps 5-11) iterate over its items a, b, e and f, replacing each of them with its value in the hashtable H. Since these items have not been generalized before, their values in H contain the items themselves and thus p is left intact. Next, in step 12, we compute the support of p in $\tilde{\mathcal{D}}$, and, since it is less than k, we execute the loop beginning in step 15. In steps 16 to 21, PCTA considers all possible cluster merging operations that affect the privacy constraint p. Put in terms of item generalization decisions, the algorithm considers generalizing each of the items $\{a, b, e, f\}$ together with any other item in the domain $\mathcal{I}$ and constructs the generalized item (d, f), which incurs the minimum utility loss among all the examined generalized items. Next, the algorithm assigns (d, f) to $\tilde{i}$, in step 22, and updates $\tilde{\mathcal{D}}$, p and H (steps 23-26). Specifically, the generalized item (d, f) replaces d, and the values of d and f in H are updated. Since the support of p remains less than k after generalizing d to (d, f), the loop of step 15 is executed again. Now, PCTA considers a, b, e, and (d, f) for generalization and constructs (a, b) that has the minimum utility loss. While p is updated to $(a, b)e(d, f)$, it still has*

a support of less than k, and thus PCTA performs another iteration of the loop of step 15*. In the latter, p is updated to* $\{(a,b)(d,e,f)\}$*, which has a support of* 3*. Thus, in step* 27*, p is removed from PQ and, in step* 28*, the anonymized dataset of Fig. 1(e) is returned.* □

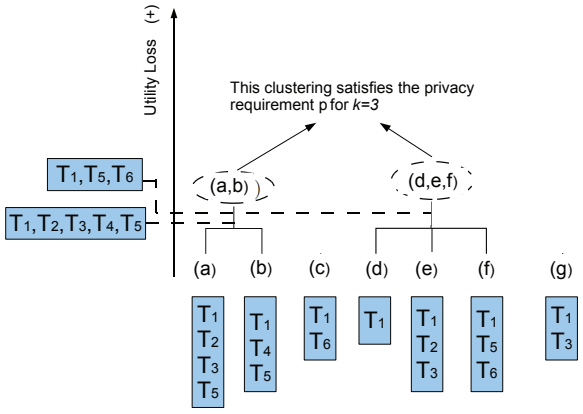

Figure 5: Anonymizing the data of Fig. 1(a)

Cost analysis. Assuming that we have $|\mathcal{P}|$ privacy constraints and each of them has $|p|$ items, PCTA takes $O(|\mathcal{P}| \times |p| \times (N + |\mathcal{I}|^2))$ time. This is because we need $O(|\mathcal{P}| \times log(|\mathcal{P}|))$ time to build PQ and $O(|\mathcal{P}| \times |p| \times (log(|\mathcal{I}|) + N + |\mathcal{I}|^2))$ time for steps $3-27$. More specifically, the lazy updating strategy for the items of p in steps $5-11$ takes $O(|p| \times log(|\mathcal{I}|))$ time, the support computation of p in step 12 takes $O(|p| \times N)$ time, and the while loop in step 15 takes $O(|p| \times (|\mathcal{I}|-1) + (|p|-1) \times (|\mathcal{I}|-2) + \ldots + 1 \times 1) \approx O(|p| \times |\mathcal{I}|^2)$ time.

4. EXPERIMENTAL EVALUATION

In this section, we present extensive experiments to evaluate the ability of PCTA to produce anonymized data with low information loss efficiently. Specifically, in Section 4.1 we discuss the experimental setup and provide information about the datasets that we used. Then, Section 4.2, evaluates our algorithm against Apriori [33] and COAT [21], in terms of data utility, under several different privacy requirements. The results of this set of experiments confirm that PCTA is able to retain much more data utility when compared to other methods under all tested scenarios, as: (1) it allows aggregate queries to be answered many times more accurately (e.g., the average error for our method was up to 26 and 6 times lower than that of Apriori and COAT respectively), and (2) it incurs an amount of information loss that is smaller by several orders of magnitude. In the second set of experiments, presented in Section 4.3, we examine the runtime of our method. Our results indicate that PCTA can produce anonymized data efficiently, as it scales well with respect to both dataset size and k.

4.1 Experimental setup and data

To allow a direct comparison between the tested algorithms, we configured all of them as in [21] and transformed the anonymized datasets produced by them by replacing each generalized item with the set of items it contains. We note that, in this setup, COAT does not take the specified utility constraints into account. We used a C++ implementation of Apriori provided by the authors of [33] and implemented PCTA and COAT in C++. All methods ran on an Intel 2.8GHz machine with 4GB of RAM and tested using a common framework to measure data utility that is built in Java.

We use the datasets *BMS-WebView-1* and *BMS-WebView-2* (referred to as *BMS1* and *BMS2* respectively), which contain click-stream data from two e-commerce sites and have been used extensively in evaluating prior work [11,21,33]. The first of these datasets is comprised of 59602 transactions, whose maximum and average size are 267 and 2.5, respectively, and it has a domain size of 497, while the second one is comprised of 77512 transactions, whose maximum and average size are 161 and 5, respectively, and has a domain size of 3340.

4.2 Data utility evaluation

We compare the amount of data utility preserved by all methods by considering three utility measures: Average Relative Error (*AvgRE*) measure [11,18], Utility Loss (*UL*) [21], and Normalized Certainty Penalty (*NCP*) [35]. *AvgRE* captures the accuracy of query answering on anonymized data. It is computed as the mean error of answering a workload of queries and reflects the average number of transactions that are retrieved incorrectly as part of query answers. To measure *AvgRE*, we constructed workloads comprised of 1000 COUNT() queries that retrieve the set of supporting transactions of 5-itemsets, following the methodology of [11,21]. The items participating in these queries were selected randomly from the generalized items. Due to space limitations, we only report a small subset of our experiments.

4.2.1 Anonymization using k^m-anonymity

We first assumed that combinations of up to 2 items need to be protected. Thus, we set $m = 2$ and configured COAT and PCTA by using all 2-itemsets as privacy constraints. We evaluated data utility for various k values in $[2, 50]$. Fig. 6 illustrates the result with respect to *AvgRE* for *BMS1* and Fig. 7 for *BMS2*. It can be seen that PCTA allows at least 7 and 2 times (and up to 26 and 6 times) more accurate query answering than Apriori and COAT respectively. Furthermore, PCTA incurred significantly less information loss to anonymize data, as shown in Fig. 8, which illustrates the *UL* scores for *BMS1*. These results verify that the clustering-based search strategy that is employed by PCTA is much more powerful than the space partitioning strategies of Apriori and COAT.

Another observation is that the performance of both Apriori and COAT in terms of preserving data utility deteriorates much faster when k increases. This is mainly because these algorithms create much larger groups of items than PCTA. Specifically, Apriori considers fixed groups of items, whose size depends on the fan-out of the hierarchy, and generalizes together all items in each group, while COAT partitions items based on the utility loss incurred by generalizing a single item in a group.

Next, we assumed that combinations of 1 to 3 items need to be protected using $k = 5$. Figure 9 illustrates the result with respect to *AvgRE* for *BMS1*. Observe that the amount of information loss incurred by all methods decreases as a function of the number of items attackers are expected to know, as more generalization is required to pre-

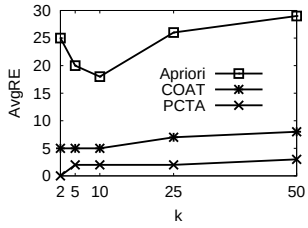

Figure 6: *AvgRE* vs. k (*BMS1*)

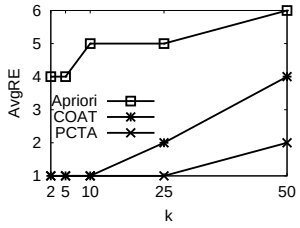

Figure 7: *AvgRE* vs. k (*BMS2*)

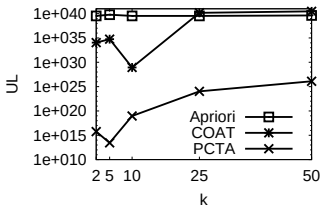

Figure 8: UL vs. k (*BMS1*)

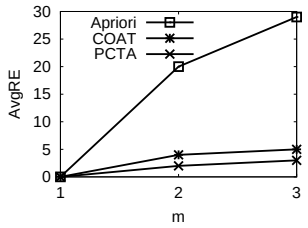

Figure 9: *AvgRE* vs. m (*BMS1*)

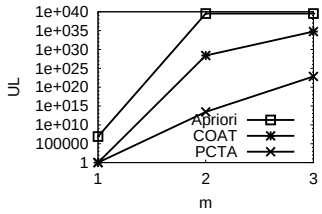

Figure 10: UL vs. m (*BMS1*)

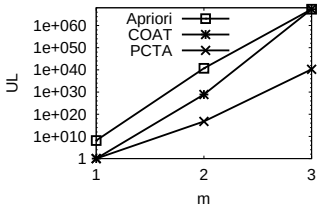

Figure 11: UL vs. m (*BMS2*)

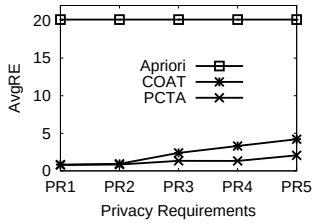

Figure 12: *AvgRE* vs. PR (*BMS1*)

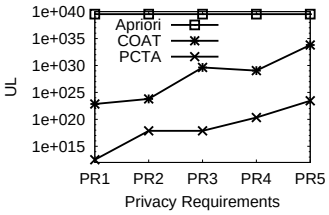

Figure 13: *UL* vs. PR (*BMS1*)

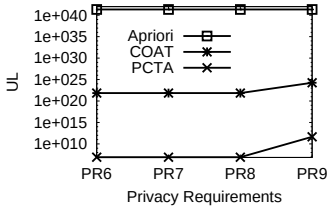

Figure 14: *UL* vs. PR (*BMS1*)

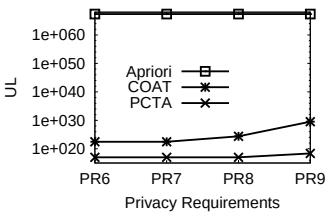

Figure 15: *UL* vs. PR (*BMS2*)

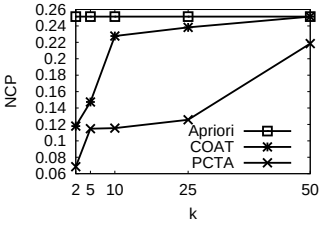

Figure 16: *NCP* vs. k (*BMS1*)

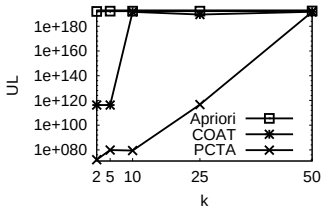

Figure 17: *UL* vs. k (*BMS2*)

serve privacy. However, PCTA outperformed Apriori and COAT in all cases, permitting queries to be answered with an error that is *at least* 8 *times lower than that of Apriori and* 1.6 *times lower than that of COAT*. Similar results were obtained when *UL* was used to capture data utility, as can be seen in Figs. 10 and 11 for *BMS1* and *BMS2*, respectively. This demonstrates that protecting incrementally larger itemsets as Apriori does, leads to significantly more generalization compared to applying generalization to protect that items in each privacy constraint as in the PCTA algorithm.

4.2.2 Anonymization using privacy constraints with various characteristics

For this set of experiments, we constructed 5 sets of privacy constraints: *PR1*,...,*PR5*, comprised of 2-itemsets, and we assumed that they need protection with $k = 5$. Each set contains a certain percentage of randomly selected items, which is 2% for *PR1*, 5% for *PR2*, 10% for *PR3*, 25% for *PR4*, and 50% for *PR5*. The *AvgRE* scores for all methods, when applied on *BMS1*, are shown in Fig. 12.

Since Apriori does not take into account the specified privacy constraints, its performance remains constant in this experiment and is the worst among the tested algorithms. PCTA outperformed both Apriori and COAT, achieving *up to* 26 *times lower AvgRE scores than those of Apriori and* 2.5 *times lower than those of COAT*. Furthermore, the difference in *AvgRE* scores between PCTA and COAT increases as policies become more stringent, which confirms the benefit of our clustering-based strategy. The ability of PCTA to preserve data utility better than Apriori and COAT was also confirmed when *UL* was used, as shown in Fig. 13.

Next, we constructed 4 sets of privacy constraints *PR6*,...,*PR9* that are comprised of 1000 itemsets and need to be protected with $k = 5$. A summary of these constraints appears in Table 1. Figure 14 illustrates the *NCP* scores for BMS1 and Fig. 15 the *UL* scores for BMS2. As can be seen, PCTA consistently outperformed Apriori and COAT, being able to incur less information loss. This again demonstrates the ability of the clustering-based strategy employed in PCTA to preserve data utility.

Privacy Constraints	% of items	% of 2-itemsets	% of 3-itemsets	% of 4-itemsets
PR6	33%	33%	33%	1%
PR7	30%	30%	30%	10%
PR8	25%	25%	25%	25%
PR9	16.7%	16.7%	16.7%	50%

Table 1: Summary of privacy constraints *PR6*, *PR7*, *PR8* and *PR9*

4.2.3 Anonymization using complete k-anonymity

Last, we evaluated the effectiveness of the methods when privacy is enforced through complete k-anonymity, which requires protecting all items of a transaction. To achieve this, we configured Apriori by setting m to the size of the largest transaction of each dataset, and COAT and PCTA by generating itemsets using the *Pgen* algorithm introduced in [21]. As shown in Figs. 16 and 17, which present results for *NCP* and *UL* respectively, PCTA performs better than Apriori and COAT, while Apriori and COAT incur much information loss particularly when k is 10 or larger. In fact, in these cases, these algorithms created generalized items whose size was much larger than those constructed by PCTA. This again shows that PCTA is more effective in retaining information loss.

4.3 Runtime efficiency

We used BMS1 to evaluate the runtime efficiency of PCTA, assuming that all 2-itemsets require protection. We first tested scalability in terms of dataset size, using increasingly larger subsets of *BMS1*. Since the size and items of a transaction can affect the runtime of the algorithms, we require the transactions of a subset to be contained in all larger subsets. From Fig. 18 we can see that PCTA is more efficient than Apriori, because it discards protected itemsets, whereas Apriori considers all m-itemsets, as well as their possible generalizations. This incurs more overhead, particularly for large datasets. However, PCTA is less efficient than COAT, as it explores a larger number of potential generalizations.

Finally, we examined how our method scales with respect to k and report the result in Fig. 19. Observe that Apriori becomes slightly more efficient as k increases, while the runtime of both PCTA and COAT follows an opposite trend. This is because Apriori generalizes entire subtrees of items, while both our method and COAT generalize one item (or generalized item) at a time. Nevertheless, PCTA was found to be *up to* 44% *more efficient than COAT*. This is attributed to the lazy updating strategy that it adopts. Since data needs to be scanned after each item generalization, the savings from using this strategy increase as k gets larger, as discussed in Section 3.

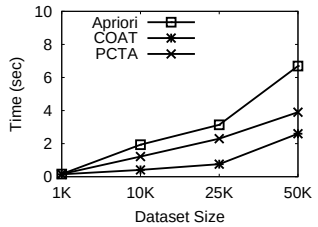

Figure 18: Runtime vs. $|D|$

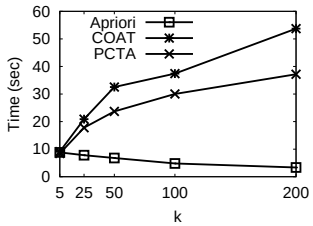

Figure 19: Runtime vs. k

5. EXTENSIONS

This section discusses how our approach can be extended to deal with constraints that anonymized transactions need to satisfy to be protected and useful in many real-world applications. The first class of constraints we consider are related to privacy and impose the need to prevent the inference of sensitive items. Items that are considered to be sensitive are those that an individual would not want to be associated with, such as diagnosis codes for HIV or alcohol/drug abuse [27], or purchased goods that may reveal an individual's political or religious beliefs [25]. As mentioned in Section 2.4, an anonymized dataset may still be susceptible to sensitive itemset disclosure when it allows an attacker to associate an individual with one or more sensitive items in this dataset with a sufficiently large probability. To illustrate the threat of sensitive itemset disclosure, we provide the following example.

EXAMPLE 2. *Consider the anonymized data shown in Fig. 1(e) and assume that inferring whether an individual has purchased the sensitive item g with a probability of at least* 1/2 *needs to be prevented. Observe that the dataset of Fig. 1(e) is not protected against sensitive itemset disclosure. This is because, knowing that* Greg *has purchased items a and e, an attacker can infer that* Greg *purchased g with a probability of* 1/2*, since there are 4 transactions that are associated with a and e, and 2 of these transactions contain the sensitive item g.*

To tackle sensitive itemset disclosure, we assume a categorization of items in $\mathcal{I}$ into public and sensitive items, which are contained in the sets $\mathcal{I}_p$ and $\mathcal{I}_s$ respectively. Following [23, 37], we also assume that all potentially linkable items are contained in $\mathcal{I}_p$, $\mathcal{I}_n \cup \mathcal{I}_s = \mathcal{I}$, and $\mathcal{I}_n \cap \mathcal{I}_s = \varnothing$. Our approach can produce an anonymized dataset $\tilde{\mathcal{D}}$ that prevents sensitive itemset disclosure in this setting. This requires ensuring that an attacker with the knowledge of a privacy constraint p cannot associate an individual to a transaction of $\tilde{\mathcal{D}}$ that contains any item $i \in p$ together with any item $i_s \in \mathcal{I}_s$ with a probability that exceeds h, where $h \in [0, 1]$ is a threshold specified by data owners. PCTA can be extended to forestall sensitive itemset disclosure by constructing an anonymized dataset $\tilde{\mathcal{D}}$ such that $sup(p, \tilde{\mathcal{D}}) \geq k$ and $\frac{sup(p \cup i_s, \tilde{\mathcal{D}})}{sup(p, \tilde{\mathcal{D}})} < h$, for each privacy constraint p and item $i_s \in \mathcal{I}_s$. This is possible through generalizing public items, since this process can increase the number of transactions that support p and sensitive items.

We also consider anonymizing data to support applications in which accurately computing the number of transactions that are associated with specific items or itemsets is required [22, 27]. In these applications, we cannot assume that a generalized item can replace any set of items in $\mathcal{I}$, as most approaches do [14, 33, 37]. Consider, for example, a

marketing study that needs to determine exactly how many individuals have purchased a or b. It easy to see that it is difficult to conduct this study using the anonymized data of Fig. 1(d), since the fact that a, b, and c are all mapped to (a, b, c) makes it difficult to accurately compute the number of transactions that support a or b.

To deal with these applications, we assume a clustering $\mathcal{C}_U$ that is specified by data owners according to the applications anonymized data are intended for, so that each cluster in $\mathcal{C}_U$ corresponds to the most generalized item that can be contained in $\tilde{\mathcal{D}}$. We also state that a cluster $c \in \mathcal{C}$ is *subsumed* by a cluster $c' \in \mathcal{C}_U$ when, for each set of items that is mapped to a generalized item $\tilde{i}$ (represented as $c \in \mathcal{C}$), this set of items is mapped to exactly one generalized item $\tilde{i'}$, represented as $c' \in \mathcal{C}_U$. To satisfy the constraints expressed through $\mathcal{C}_U$, PCTA can be extended to produce an anonymized dataset $\tilde{\mathcal{D}}$, such that all clusters of $\mathcal{C}$ are subsumed by clusters of $\mathcal{C}_U$.

Due to space limitations, we do not provide details on how PCTA can be modified to accommodate the aforementioned extensions.

6. CONCLUSIONS

Existing algorithms are unable to anonymize transaction data under a range of different privacy requirements without incurring excessive information loss, because they are built upon the intrinsic properties of a single privacy model. To address this issue, we introduced a novel formulation of the problem of transaction data anonymization based on clustering. The generality of this formulation allows designing algorithms that are independent of generalization strategies and privacy models and able to achieve high data utility and privacy. We also proposed PCTA, a clustering-based algorithm that can produce a significantly better result than the state-of-the-art methods in terms of data utility and be extended to accommodate privacy and utility constraints that are common in real-world applications.

7. ACKNOWLEDGEMENTS

We would like to thank Manolis Terrovitis, Nikos Mamoulis and Panos Kalnis for providing the implementation of the Apriori anonymization algorithm [33], and Bradley Malin for helpful discussions.

8. REFERENCES

[1] National Institutes of Health. Policy for sharing of data obtained in NIH supported or conducted genome-wide association studies. NOT-OD-07-088. 2007.

[2] Health insurance portability and accountability act of 1996 united states public law.

[3] R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE*, pages 217–228, 2005.

[4] J. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymity using clustering technique. In *DASFAA*, pages 188–200, 2007.

[5] J. Cao, P. Karras, C. Raïssi, and K. Tan. *rho*-uncertainty: Inference-proof transaction anonymization. *PVLDB*, 3(1):1033–1044, 2010.

[6] C.-C. Chang, B. Thompson, H. Wang, and D. Yao. Towards publishing recommendation data with predictive anonymization. In *5th ACM Symposium on Information, Computer and Communications Security*, pages 24–35, 2010.

[7] B. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala. Privacy-preserving data publishing. *Found. Trends databases*, 2(1–2):1–167, 2009.

[8] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *DMKD*, 11(2):195–212, 2005.

[9] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey on recent developments. *ACM Comput. Surv.*, 42, 2010.

[10] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *ICDE*, pages 205–216, 2005.

[11] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data. In *ICDE*, pages 715–724, 2008.

[12] A. Gkoulalas-Divanis and V. Verykios. A free terrain model for trajectory k-anonymity. In *DEXA*, pages 49–56, 2008.

[13] A. Gkoulalas-Divanis and V. S. Verykios. *Privacy in Trajectory Data*, chapter 11, pages 199–212. Social Implications of Data Mining and Information Privacy: Interdisciplinary Frameworks and Solutions. Information Science Reference, 2008.

[14] Y. He and J. F. Naughton. Anonymization of set-valued data via top-down, local generalization. *PVLDB*, 2(1):934–945, 2009.

[15] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD*, pages 279–288, 2002.

[16] S. Jha, L. Kruger, and P. McDaniel. Privacy preserving clustering. In *ESORICS*, pages 397–417, 2005.

[17] S. Kisilevich, L. Rokach, Y. Elovici, and B. Shapira. Efficient multidimensional suppression for k-anonymity. *TKDE*, 22:334–347, 2010.

[18] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *ICDE*, page 25, 2006.

[19] J. Li, R. Wong, A. Fu, and J. Pei. Achieving -anonymity by clustering in attribute hierarchical structures. In *DaWaK*, pages 405–416, 2006.

[20] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *2008 SIGMOD*, pages 93–106, 2008.

[21] G. Loukides, A. Gkoulalas-Divanis, and B. Malin. COAT: COnstraint-based Anonymization of Transactions. *KAIS*. To Appear.

[22] G. Loukides, A. Gkoulalas-Divanis, and B. Malin. Anonymization of electronic medical records for validating genome-wide association studies. *PNAS*, 17:7898–7903, 2010.

[23] G. Loukides, A. Gkoulalas-Divanis, and J. Shao. Anonymizing transaction data to eliminate sensitive inferences. In *DEXA*, pages 400–415, 2010.

[24] G. Loukides and J. Shao. Capturing data usefulness and privacy protection in k-anonymisation. In *SAC*, pages 370–374, 2007.

[25] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE S&P*, pages 111–125, 2008.

[26] M. E. Nergiz and C. Clifton. Thoughts on k-anonymization. *DKE*, 63(3):622–645, 2007.

[27] T. D. of State Health Services. User manual of texas hospital inpatient discharge public use data file. `http://www.dshs.state.tx.us/THCIC/`, 2008.

[28] R. G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *Workshop on Privacy in Location-Based Applications*, 2008.

[29] S. J. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *VLDB*, pages 682–693, 2002.

[30] P. Samarati. Protecting respondents identities in microdata release. *TKDE*, 13(9):1010–1027, 2001.

[31] L. Sweeney. k-anonymity: a model for protecting privacy. *IJUFKS*, 10:557–570, 2002.

[32] M. Terrovitis, N. Mamoulis, and P. Kalnis. Local and global recoding methods for anonymizing set-valued data. *VLDB J.* To appear.

[33] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. *PVLDB*, 1(1):115–125, 2008.

[34] V. S. Verykios, M. L. Damiani, and A. Gkoulalas-Divanis. *Privacy and Security in Spatiotemporal Data and Trajectories*, chapter 8, pages 213–240. Mobility, Data Mining and Privacy: Geographic Knowledge Discovery. Springer, 2008.

[35] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. Utility-based anonymization using local recoding. In *KDD*, pages 785–790, 2006.

[36] R. Xu and D. C. Wunsch. *Clustering*. Wiley-IEEE Press, 2008.

[37] Y. Xu, K. Wang, A. W.-C. Fu, and P. S. Yu. Anonymizing transaction databases for publication. In *KDD*, pages 767–775, 2008.

[38] Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *KDD*, pages 401–406, 2001.