Data & Knowledge Engineering 68 (2009) 552–575

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Privacy-preserving data publishing for cluster analysis

Benjamin C.M. Fung [a,*], Ke Wang [b], Lingyu Wang [a], Patrick C.K. Hung [c]

[a] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada H3G 1M8
[b] School of Computing Science, Simon Fraser University, BC, Canada V5A 1S6
[c] Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada L1H 7K4

ARTICLE INFO

Article history:
Received 21 October 2007
Received in revised form 27 November 2008
Accepted 5 December 2008
Available online 27 December 2008

Keywords:
Privacy
Knowledge discovery
Anonymity
Cluster analysis

ABSTRACT

Releasing person-specific data could potentially reveal sensitive information about individuals. *k*-anonymization is a promising privacy protection mechanism in data publishing. Although substantial research has been conducted on *k*-anonymization and its extensions in recent years, only a few prior works have considered releasing data for some specific purpose of data analysis. This paper presents a practical data publishing framework for generating a masked version of data that preserves both individual privacy and information usefulness for cluster analysis. Experiments on real-life data suggest that by focusing on preserving cluster structure in the masking process, the cluster quality is significantly better than the cluster quality of the masked data without such focus. The major challenge of masking data for cluster analysis is the lack of class labels that could be used to guide the masking process. Our approach converts the problem into the counterpart problem for classification analysis, wherein class labels encode the cluster structure in the data, and presents a framework to evaluate the cluster quality on the masked data.

1. Introduction

Information sharing is a vital building block for today's business world. In June 2004, the Information Technology Advisory Committee released a report entitled Revolutionizing Health Care Through Information Technology [31]. A key point is the establishment of a nationwide system of electronic medical records that encourages sharing medical knowledge through computer-assisted clinical decision support. The report states that "all information about a patient from any source could be securely available to any health care provider when needed, while assuring patient control over privacy." However, in many real-life data publishing scenarios, individual participants (e.g., the patients) do not even have the right to opt out from the sharing. For example, licensed hospitals in California are required to submit specific demographic data on every patient discharged from their facility [7]. Thus, the burden of data privacy protection falls on the shoulder of the data holder (e.g., the hospital). This paper presents a technical response to the demand for simultaneous privacy protection and information sharing, specifically for the task of cluster analysis.

In this paper, we define the data publishing scenario as follows. Consider a person-specific data table T with patients' information on *Zip code*, *Birthplace*, *Gender*, and *Disease*. The data holder wants to publish T to some recipient for cluster analysis. However, if a set of attributes, called a *Quasi-Identifier* or a *QID*, on {*Zip code*, *Birthplace*, *Gender*} is so specific that few people match it, publishing the table will lead to linking a unique or small number of individuals with the sensitive information on *Disease*. Even if the currently published table T does not contain sensitive information, individuals in T can

* Corresponding author. Tel.: +1 514 8482424x5919; fax: +1 514 8483171.
E-mail addresses: fung@ciise.concordia.ca (B.C.M. Fung), wangk@cs.sfu.ca (K. Wang), wang@ciise.concordia.ca (L. Wang), patrick.hung@uoit.ca (P.C.K. Hung).
URLs: http://www.ciise.concordia.ca/~fung (B.C.M. Fung), http://www.cs.sfu.ca/~wangk (K. Wang), http://www.ciise.concordia.ca/~wang (L. Wang), http://www.hrl.uoit.ca/~ckphung (P.C.K. Hung).

0169-023X/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2008.12.001

be linked to the sensitive information in some (readily available) external source by a join on the common attributes [33,40]. The problem studied in this paper is to generate a masked version of T that satisfies two requirements: the *anonymity requirement* and the *clustering requirement*.

1.1. Anonymity requirement

To protect privacy, instead of publishing the raw table T(*QID*, *Sensitive_attribute*), the data holder publishes a masked table T^*, where *QID* is a set of quasi-identifying attributes masked to some general concept. For example, *QID* = {*Zip code*, *Birthplace*, *Gender*}. The data holder could generalize the values in *Birthplace* from city level to region level so that more records will match the generalized description and, therefore, individuals who match the description will become less identifiable. The anonymity requirement is specified by k-anonymity [33,40]: A masked table T^* satisfies k-anonymity if each record in T^* shares the same value on *QID* with at least $k-1$ other records, where k is an anonymity threshold specified by the data holder.[1] All records in the same *QID* group are made indistinguishable and, therefore, it is difficult to determine whether a matched individual actually has the disease from T^*.

1.2. Clustering requirement

The data holder wants to publish a masked table T^* to a recipient for the purpose of cluster analysis, the goal of which is to group similar objects into the same cluster and group dissimilar objects into different clusters. We assume that the *Sensitive_attribute* is important for the task of cluster analysis; otherwise, it should be removed. The recipient may or may not be known at the time of data publication.

We study the *anonymity problem for cluster analysis*: for a given anonymity requirement and a raw data table T, a data holder wants to generate an anonymous version of T, denoted by T^*, that preserves as much of the information as possible for cluster analysis, and then publish T^* to a data recipient. The data holder, for example, could be a hospital that wants to share its patients' information with a drug company for pharmaceutical research.

There are many possible masked versions of T^* that satisfy the anonymity requirement. The challenge is how to identify the appropriate one for cluster analysis. An inappropriately masked version could put originally dissimilar objects into the same cluster, or put originally similar objects into different clusters because other masked objects become more similar to each other. Therefore, a quality-guided masking process is crucial. Unlike the anonymity problem for classification analysis [12], the anonymity problem for cluster analysis does not have class labels to guide the masking. Another challenge is that it is not even clear what "information for cluster analysis" means, nor how to evaluate the cluster quality of generalized data. In this paper, we define the anonymity problem for cluster analysis and present a solution framework to address the challenges in the problem. Our contributions to the literature can be summarized by answering the following key questions:

(1) *Can a masked table simultaneously satisfy both anonymity and clustering requirements?* Our insight is that the two requirements are indeed dealing with two types of information: The anonymity requirement aims at masking identifying information that specifically describes individuals; the clustering requirement aims at extracting general structures that capture patterns. If masking is carefully performed, identifying information can be masked while still preserving the patterns for cluster analysis. Our experimental results on real-life data support this insight.

(2) *What information should be preserved for cluster analysis in the masked data?* We present a framework to convert the anonymity problem for cluster analysis to the counterpart problem for classification analysis. The idea is to extract the cluster structure from the raw data, encode it in the form of class labels, and preserve such class labels while masking the data. The framework also permits the data holder to evaluate the cluster quality of the anonymized data by comparing the cluster structures before and after the masking. This evaluation process is important for data publishing in practice, but very limited study has been conducted in the context of privacy preservation and cluster analysis.

(3) *Can cluster-quality guided anonymization improve the cluster quality in anonymous data?* A naive solution to the studied privacy problem is to ignore the clustering requirement and employ some general purpose anonymization algorithms, e.g. [20], to mask data for cluster analysis. Extensive experiments suggest that by focusing on preserving cluster structure in the masking process, the cluster quality outperforms the cluster quality on masked data without such focus. Our experiments also demonstrate there is a trade-off between privacy protection and cluster quality. In general, the cluster quality on the masked data degrades as the anonymity threshold k increases.

(4) *Can the specification of multiple quasi-identifiers improve the cluster quality in anonymous data?* The classic notion of k-anonymity assumes that a single united quasi-identifier *QID* contains all quasi-identifying attributes, but research shows that it often leads to substantial loss of data quality as the *QID* size increases [1]. Our insight is that, in practice, an attacker is unlikely to know all identifying attributes of a target victim (the person being identified), so the data is over-protected by a single *QID*. Our proposed method allows the specification of multiple *QID*s, each of which has a smaller size, and therefore avoids over-masking and improves the cluster quality.

[1] To avoid confusion with the number of clusters k in k-means clustering algorithm discussed later, we use h to denote anonymity threshold in the rest of this paper.

Given that the clustering task is known in advance, why not publish the analysis result instead of the data records? Unlike classification trees and association rules, publishing the cluster statistics (e.g., cluster centers, together with their size and radius) usually cannot fulfil the information needs for cluster analysis. Often, data recipients want to browse into the clustered records to gain more knowledge. For example, a medical researcher may browse into some clusters of patients and examine their common characteristics. Publishing data records not only fulfills the vital requirement for cluster analysis, but also increases the availability of information for the recipients.

The paper is organized as follows. We review related works in Section 2, define the problem in Section 3, present the framework of our approach in Section 4, and evaluate it in Section 5. Then, we show the extensions of the framework to achieve other privacy notions in Section 6 and conclude the paper in Section 7.

2. Related works

Recently, the research topic of privacy-preserving data publishing (PPDP) has received a great deal of attention in the database and data mining research communities. The literature in PPDP can be broadly categorized by linkage prevention models. A privacy violation occurs when a person is linked to a record or to a value on *Sensitive_attribute*; these violations are called *record linkage* and *attribute linkage*. In both types of violations, the attacker knows the *QID* of the victim and that the victim has a record in the released table.

In the attack of *record linkage*, some value *qid* on *QID* identifies a small number of records in the released table *T*. If the victim's *QID* matches the value *qid*, the victim is vulnerable to being linked to the small number of records in the *qid* group. In this case, the attacker faces only a small number of possibilities for the victim's record, and with the help of additional knowledge, there is a chance that the attacker could uniquely identify the victim's record from the group. The notion of *k*-anonymity [33,40] and its variations [21,27] are proposed to prevent record linkage through *QID*.

In the attack of *attribute linkage*, the attacker may not precisely identify the record of the victim, but could infer his or her sensitive values from the published data *T*, based on the set of sensitive values associated with the group that the victim belongs to. If some sensitive values predominate in a *qid* group, a successful inference becomes relatively easy even if *k*-anonymity is satisfied. Alternative privacy notions, such as ℓ-diversity [25] and confidence bounding [45,46], are proposed to prevent attribute linkage. The general idea is to de-associate the correlation between *QID* and *Sensitive_attribute* so that even if the attacker can identify the *QID* group of the victim, the attacker cannot infer the victim's sensitive information. The ℓ-diversity requires every *qid* group to contain at least ℓ "well-represented" sensitive values. Thus, a ℓ-diverse table also satisfies *k*-anonymity with $k = \ell$. Yet, not all privacy notions that thwart attribute linkages can thwart record linkages. For example, confidence bounding [46] cannot prevent record linkages. Wong et al. [47] proposed a unified notion of *k*-anonymity and confidence bounding to prevent both linkages.

Many generalization methods [8,19,20,23,25,39,48,49] have been proposed to achieve the above mentioned privacy notions, but they use simple quality measures to guide the masking process and do not consider data mining tasks such as classification analysis and cluster analysis. As a result, the data mining results extracted from their anonymous data are often unsatisfactory [12,16,22]. Preserving anonymity for classification analysis was studied in [4,12,13,16,22,29,43–46]. The idea was to use the available class labels to guide the masking process so that the class labels could still be identified in the masked *QID*. A *genetic algorithm* solution was proposed in [16] to preserve the usefulness to classification in a data table while satisfying an anonymity requirement. However, it suffered from poor efficiency and handled only a single quasi-identifier. [4] proposed an algorithm, called *K-Optimize*, to identify the optimal *k*-anonymized version on training data, but such optimality on the training data does *not* guarantee the lowest possible error rate in future (or testing) data. Fung et al. [13] presented a greedy approach to the same problem based on information gain. LeFevre et al. [22] proposed a multi-dimensional generalization method, called *InfoGain Mondrian*, to identify a *k*-anonymous solution. Mohammed [29], Wang and Fang [43], Wang et al. [44] addressed the extended data publishing scenarios, such as multiple releases and multiple data holders. Fung et al. [10] presented a suppression method for anonymizing high-dimensional sequential data.

There are two major differences that distinguish our work from the above mentioned anonymization algorithms. First, in the anonymization problem for cluster analysis studied in this paper, no class label is available for guiding the masking process. This creates a new problem in privacy-preserving data publishing. Second, most of the authors have not proposed algorithms for anonymizing both continuous and categorical attributes with and without data holder-specified taxonomy trees. LeFevre et al. [20] briefly discussed different possible generalization schemes, but did not show their experimental results. The anonymization method presented in [22] can anonymize both categorical attributes with taxonomy trees and continuous attributes without taxonomy trees, but [22] did not discuss how to anonymize categorical attributes without taxonomy trees. Our proposed method presents a unified approach to mask all these types of attributes; this feature is important for real-life data anonymization.

There is a family of anonymization methods [2,3] that achieves privacy *by* clustering similar data records together. Their objective is very different from our studied problem, which is publishing data *for* cluster analysis. Aggarwal and Yu [2] proposed an anonymization approach, called *condensation*, to first condense the records into multiple non-overlapping groups in which each group has a size of at least *h* records. Then, for each group, the method extracts some statistical information, such as sum and covariance, that suffices to preserve the mean and correlation across different attributes. Finally, based on the statistical information, the method generates synthetic data records for each group. In a similar spirit, *r-gather clustering*

[3] partitions records into several clusters such that each cluster contains at least r data points. Then the cluster centers, together with their size, radius, and a set of associated sensitive values, are released. Compared to the masking approach, one limitation of the clustering approach is that the published records are "synthetic" in that they may not correspond to the real world entities represented by the raw data. As a result, the analysis result is difficult to justify if, for example, a police officer wants to determine the common characteristics of some criminals from the data records.

Many secure protocols have been proposed for distributed computation among multiple parties. For example,[41,15] presented secure protocols to generate a clustering solution from vertically and horizontally partitioned data owned by multiple parties. In their model, accessing data held by other parties is prohibited, and only the final cluster solution is shared among participating parties. We consider a completely different problem, of which the goal is to share data that is immunized against privacy attacks.

We highlight some recent development in cluster analysis. Beringer and Hüllermeier [5] presented a method for clustering parallel data streams. Birant and Kut [6] studied the problem of clustering spatial-temporal data. McClean et al. [28] presented a number of efficient clustering strategies for distributed database. Gelbard et al. [14] conducted an extensive empirical study on different clustering methods.

3. Problem statements

A *labelled* table has the form $T(D_1,\ldots,D_m,Class)$ and contains a set of records of the form $\langle v_1,\ldots,v_m,cls\rangle$, where v_j, for $1\leqslant j\leqslant m$, is a domain value of attribute D_j, and *cls* is a class label of the *Class* attribute. Each D_j is either a categorical or a continuous attribute. An *unlabelled* table has the same form as a labelled table but without the *Class* attribute.

Suppose that a data holder wants to publish a person-specific table T^*, but also wants to protect against linking an individual to sensitive information either inside or outside T^* through some sets of identifying attributes, called *quasi-identifiers QID*. A *sensitive record linking* occurs if some value on a quasi-identifier is shared by only a small number of records in T^*. This requirement is formally defined below.

Definition 3.1 (*Anonymity requirement*). Consider p quasi-identifiers $QID_1,\ldots,QID_p$ on T^*, where $QID_i\subseteq\{D_1,\ldots,D_m\}$ for $1\leqslant i\leqslant p$. $a(qid_i)$ denotes the number of data records in T^* that share the value qid_i on QID_i. The *anonymity* of QID_i, denoted by $A(QID_i)$, is the minimum $a(qid_i)$ for any value qid_i on QID_i. A table T^* *satisfies* the *anonymity requirement* $\{\langle QID_1,h_1\rangle,\ldots,\langle QID_p,h_p\rangle\}$ if $A(QID_i)\geqslant h_i$ for $1\leqslant i\leqslant p$, where QID_i and the *anonymity thresholds* h_i are specified by the data holder.

If some QID_j could be "covered" by another QID_i, then QID_j can be removed from the anonymity requirement. This observation is stated as follows:

Observation 3.1 (*Cover*). Suppose $QID_j\subseteq QID_i$ and $h_j\leqslant h_i$ where $j\neq i$. If $A(QID_i)\geqslant h_i$, then $A(QID_j)\geqslant h_j$. We say that QID_j is *covered by* QID_i; therefore, QID_j is redundant and can be removed.

Example 3.1. Consider the data in Table 1 and taxonomy trees in Fig. 1. Ignore the dashed line in Fig. 1 for now. The table has 34 records, with each row representing one or more raw records that agree on (*Education*, *Gender*, *Age*). The *Class* column stores a count for each class label. The anonymity requirement $\langle QID_1=\{Education,Gender\},4\rangle$ states that every existing qid_1 in the table must be shared by at least 4 records. Therefore, $\langle 9th,M\rangle$, $\langle Masters,F\rangle$, $\langle Doctorate,F\rangle$ violate this requirement. To make the "female doctor" less unique, we can generalize *Masters* and *Doctorate* to *Grad School*. As a result, "she" becomes less identifiable by being one of the four females who have a graduate degree in the masked table T^*.

Definition 3.1 generalizes the classic notion of k-anonymity [34] by allowing multiple *QID*s with different anonymity thresholds. The specification of multiple *QID*s is based on an assumption that the data holder knows exactly what external information source is available for sensitive record linkage. The assumption is realistic in some data publishing scenarios. Suppose that the data holder wants to release a table $T^*(A,B,C,D,S)$, where A,B,C,D are identifying attributes and S is a sensitive attribute, and knows that the recipient has access to previously released tables $T1^*(A,B,X)$ and $T2^*(C,D,Y)$, where X and

Table 1
The labelled table.

Rec ID	Education	Gender	Age	···	Class	Count
1–3	9th	M	30		$0C_1$ $3C_2$	3
4–7	10th	M	32		$0C_1$ $4C_2$	4
8–12	11th	M	35		$2C_1$ $3C_2$	5
13–16	12th	F	37		$3C_1$ $1C_2$	4
17–22	Bachelors	F	42		$4C_1$ $2C_2$	6
23–26	Bachelors	F	44		$4C_1$ $0C_2$	4
27–30	Masters	M	44		$4C_1$ $0C_2$	4
31–33	Masters	F	44		$3C_1$ $0C_2$	3
34	Doctorate	F	44		$1C_1$ $0C_2$	1
Total					$21C_1$ $13C_2$	34

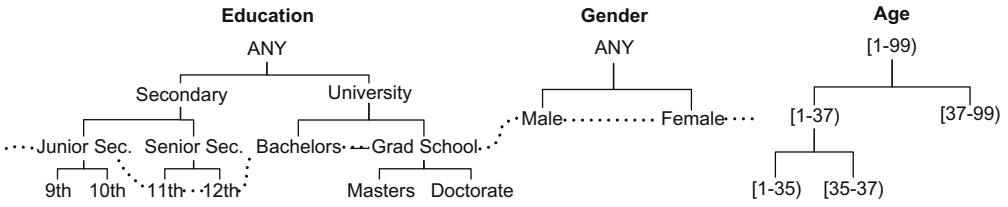


Fig. 1. Taxonomy trees.

Y are attributes not in T. To prevent linking the records in T to X or Y, the data holder only has to specify the anonymity requirement on $QID_1 = \{A, B\}$ and $QID_2 = \{C, D\}$. In this case, enforcing anonymity on $QID = \{A, B, C, D\}$ will distort the data more than is necessary. Most previous works suffer from this over-masking problem because they simply include all potential identifying attributes into a single QID. The experimental results in Section 5 confirm that the specification of multiple QIDs can reduce masking and, therefore, improve the data quality.

3.1. Masking operations

To transform a table T to satisfy an anonymity requirement, we apply one of the following three types of masking operations on every attribute D_j in $\cup QID_i$: if D_j is a categorical attribute with pre-specified taxonomy tree, then we *generalize* D_j. Specifying taxonomy trees, however, requires expert knowledge of the data. In case the data holder lacks such knowledge or, for any reason, does not specify a taxonomy tree for the categorical attribute D_j, then we *suppress* D_j. If D_j is a continuous attribute without a pre-discretized taxonomy tree, then we *discretize* D_j.[2] These three types of masking operations are formally described as follows:

(1) *Generalize* D_j if it is a categorical attribute with a taxonomy tree specified by the data holder. Fig. 1 shows the taxonomy trees for categorical attributes *Education* and *Gender*. A leaf node represents a domain value and a parent node represents a less specific value. A generalized D_j can be viewed as a "cut" through its taxonomy tree. A *cut* of a tree is a subset of values in the tree, denoted Cut_j, that contains exactly one value on each root-to-leaf path. Fig. 1 shows a cut on *Education* and *Gender*, indicated by the dash line. If a value v is generalized to its parent, all siblings of v must also be generalized to its parent. This property ensures that a value and its ancestor values will not coexist in the generalized table T^*. This generalization scheme was previously employed in [4,11,12,16,43,44].

(2) *Suppress* D_j if it is a categorical attribute without a taxonomy tree. Suppressing a value on D_j means replacing *all* occurrences of the value with the special value $\perp_j$. All suppressed values on D_j are represented by the same value $\perp_j$. We use Sup_j to denote the set of values suppressed by $\perp_j$. This type of suppression is performed at the value level, in that Sup_j in general contains a subset of the values in the attribute D_j. A clustering algorithm treats $\perp_j$ as a new value. Suppression can be viewed as a special case of generalization by considering $\perp_j$ to be the root of a taxonomy tree and $child(\perp_j)$ to contain all domain values of D_j. In this special case of generalization (which we call it suppression), we could selectively generalize (suppress) some values in $child(\perp_j)$ to $\perp_j$ while some other values in $child(\perp_j)$ remain intact.

(3) *Discretize* D_j if it is a continuous attribute. Discretizing a value v on D_j means replacing *all* occurrences of v with an interval containing the value. Our algorithm dynamically grows a taxonomy tree for intervals at runtime. Each node represents an interval. Each non-leaf node has two child nodes representing some optimal binary split of the parent interval. Fig. 1 shows such a dynamically grown taxonomy tree for *Age*, where [1–99) is split into [1–37) and [37–99). More details will be discussed in Section 4.2.1. A discretized D_j can be represented by the set of intervals, denoted Int_j, corresponding to the leaf nodes in the dynamically grown taxonomy tree of D_j.

A masked table T can be represented by $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$, where Cut_j, Sup_j, Int_j are defined above. If the masked table T^* satisfies the anonymity requirement, then $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ is called a *solution set*. Generalization, suppression, and discretization have their own merits and flexibility; therefore, our unified framework employs all of them.

What kind of information should be preserved for cluster analysis? Unlike classification analysis, wherein the information utility of attributes can be measured by their power of identifying class labels [4,12,16,22], no class labels are available for cluster analysis. One natural approach is to preserve the cluster structure in the raw data. Any loss of structure due to the anonymization is measured relative to such "raw cluster structure." We define the anonymity problem for cluster analysis as follows to reflect this natural choice of approach.

Definition 3.2 (*Anonymity problem for cluster analysis*). Given an unlabelled table T, an anonymity requirement $\{\langle QID_1, h_1 \rangle, \ldots, \langle QID_p, h_p \rangle\}$, and an optional taxonomy tree for each categorical attribute in $\cup QID_i$, the *anonymity problem*

[2] A continuous attribute with a pre-discretized taxonomy tree is equivalent to a categorical attribute with a pre-specified taxonomy tree.

for cluster analysis is to mask T on the attributes $\cup QID_i$ such that the masked table T^* satisfies the anonymity requirement and has a cluster structure as similar as possible to the cluster structure in the raw table T.

Intuitively, two cluster structures, before and after masking, are similar if the following two conditions are *generally* satisfied:

(1) two objects that belong to the same cluster before masking remain in the same cluster after masking, and
(2) two objects that belong to different clusters before masking remain in different clusters after masking.

A formal measure for the similarity of two structures will be discussed in Section 4.3.

4. Our approach

In this section, we present an algorithmic framework to generate a masked table T^*, represented by a solution set $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ that satisfies a given anonymity requirement and preserves as much as possible the raw cluster structure.

4.1. Overview of solution framework

Fig. 2 provides an overview of our proposed framework. First, we generate the cluster structure in the raw table T and label each record in T by a class label. This labelled table, denoted by T_l, has a *Class* attribute that contains a class label for each record. Essentially, preserving the raw cluster structure is to preserve the power of identifying such class labels during masking. Masking that diminishes the difference among records belonging to different clusters (classes) is penalized. As the requirement is the same as the anonymity problem for classification analysis, theoretically we can apply existing anonymization algorithms for classification analysis [4,12,16,22] to achieve the anonymity, although none of them in practice can perform all of the three types of masking operations discussed in Section 3. We explain each step in Fig. 2 as follows.

(1) *Convert T to a labelled table T_l.* Apply a clustering algorithm to T to identify the raw cluster structure, and label each record in T by its class label. The resulting labelled table T_l has a *Class* attribute containing the labels.
(2) *Mask the labelled table T_l.* Employ an anonymization algorithm for classification analysis to mask T_l. The masked T_l^* satisfies the given anonymity requirement.

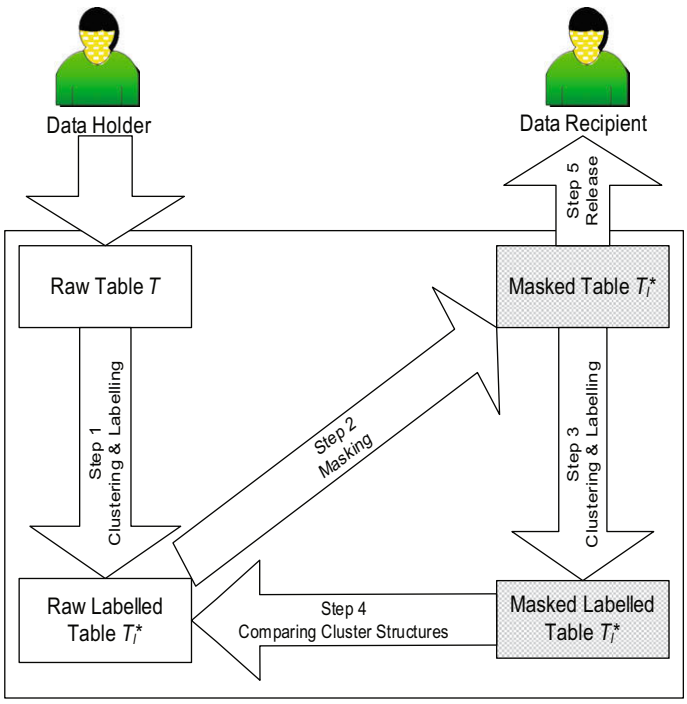


Fig. 2. The framework.

(3) *Clustering on the masked* T_l^*. Remove the labels from the masked T_l^* and then apply a clustering algorithm to the masked T_l^*, where the number of clusters is the same as in Step 1. By default, the clustering algorithm in this step is the same as the clustering algorithm in Step 1, but can be replaced with the recipient's choice if this information is available. See more discussion below.
(4) *Evaluate the masked* T_l^*. Compute the similarity between the cluster structure found in Step 3 and the raw cluster structure found in Step 1. The similarity measures the loss of cluster quality due to masking. If the evaluation is unsatisfactory, the data holder may repeat Steps 1–4 with different specification of taxonomy trees, choice of clustering algorithms, masking operations, number of clusters, and anonymity thresholds if possible. We study how these choices could influence the cluster quality in Section 5.
(5) *Release the masked* T_l^*. If the evaluation in Step 4 is satisfactory, the data holder can release the masked T_l^* together with some optional supplementary information: all the taxonomy trees (including those generated at runtime for continuous attributes), the solution set, the similarity score computed in Step 4, and the class labels generated in Step 1.

In some data publishing scenarios, the data holder does not even know who the prospective recipients are and, therefore, does not know how the recipients will cluster the published data. For example, when the Census Bureau releases data on the World Wide Web, how should the bureau set the parameters, such as the number of clusters, for the clustering algorithm in Step 1? In this case, we suggest releasing one version for each reasonable cluster number so that the recipient can make the choice based on her desired number of clusters, but this will cause a potential privacy breach because an attacker can further narrow down a victim's record by comparing different releases. A remedy is to employ the privacy notion of *BCF-anonymity* [11], which guarantees *k*-anonymity even in the presence of multiple releases. The general idea is to first compute the number of "cracked" records in each *QID* group by comparing multiple releases, and then compute the "true" anonymity of a *qid* group by subtracting the number of cracked records from the *qid* group size. Since *BCF*-anonymity is a generalized notion of *k*-anonymity, our privacy-preserving framework for cluster analysis can easily adopt *BCF*-anonymity to guarantee anonymization over multiple releases.

4.2. Anonymization for classification

The anonymity problem for classification has been studied in [4,12,16,22]. However, none of these anonymization algorithms could perform *all* masking operations, namely generalization, suppression, and discretization, specified in Section 3. To effectively mask both categorical and continuous attributes in real-life data, we proposed and implemented an anonymization algorithm called *top-down refinement* (*TDR*) that can perform all three types of masking operations in a unified fashion. TDR shares a similar top-down specialization (TDS) approach in [12], but TDS cannot perform suppression and, therefore, cannot handle categorical attributes without taxonomy trees.

TDR takes a labelled table and an anonymity requirement as inputs. The main idea of TDR is to perform maskings that preserve the information for identifying the class labels. The next example illustrates this point.

Example 4.1. Suppose that the raw cluster structure produced by Step 1 has the class (cluster) labels given in the *Class* attribute in Table 1. In Example 3.1 we generalize *Masters* and *Doctorate* into *Grad School* to make linking through (*Education*,*Gender*) more difficult. No information is lost in this generalization because the class label C_1 does not depend on the distinction of *Masters* and *Doctorate*. However, further generalizing *Bachelors* and *Grad School* to *University* makes it harder to separate the two class labels involved.

Instead of masking a labelled table T_l^* starting from the most specific domain values, TDR masked T_l^* by a sequence of refinements starting from the most masked state in which each attribute is generalized to the topmost value, suppressed to the special value $\perp$, or represented by a single interval. TDR iteratively refines a masked value selected from the current set of cuts, suppressed values, and intervals, and stops if any further refinement would violate the anonymity requirement. A refinement is *valid* (with respect to T_l^*) if T_l^* satisfies the anonymity requirement after the refinement.

We formally describe different types of refinements in Section 4.2.1, define a selection criterion for a single refinement in Section 4.2.2, and provide the anonymization algorithm TDR in Section 4.2.3.

4.2.1. Refinement

4.2.1.1. Refinement for generalization. Consider a categorical attribute D_j with a pre-specified taxonomy tree. Let $T_l^*[v]$ denote the set of generalized records that currently contains a generalized value v in the table T_l^*. Let $child(v)$ be the set of child values of v in a pre-specified taxonomy tree of D_j. A refinement, denoted by $v \to child(v)$, replaces the parent value v in all records in $T_l^*[v]$ with the child value $c \in child(v)$, where c is either a domain value d in the raw record or c is a generalized value of d. For example, a raw data record r contains a value *Masters* and the value has been generalized to *University* in a masked table T_l^*. A refinement *University* $\to$ {*Bachelors*, *Grad School*} replaces *University* in r by *Grad School* because *Grad School* is a generalized value of *Masters*.

4.2.1.2. Refinement for suppression. For a categorical attribute D_j without a taxonomy tree, a refinement $\perp_j \to \{v, \perp_j\}$ refers to disclosing one value v from the set of suppressed values Sup_j. Let $T_l^*[\perp_j]$ denote the set of suppressed records that currently contain $\perp_j$ in the table T_l^*. Disclosing v means replacing $\perp_j$ with v in all records in $T_l^*[\perp_j]$ that originally contain v.

4.2.1.3. Refinement for discretization. For a continuous attribute, refinement is similar to that for generalization except that no prior taxonomy tree is given and the taxonomy tree has to be grown dynamically in the process of refinement. Initially, the interval that covers the full range of the attribute forms the root. The refinement on an interval v, written $v \to child(v)$, refers to the optimal split of v into two child intervals $child(v)$, which maximizes the information gain. Suppose there are i distinct values in an interval. Then, there are $i-1$ number of possible splits. The optimal split can be efficiently identified by computing the information gain of each possible split in one scan of data records containing such an interval of values. See Section 4.2.2 for the definition of information gain. Due to this extra step of identifying the optimal split of the parent interval, we treat continuous attributes separately from categorical attributes with taxonomy trees.

4.2.2. Selection criterion

Each refinement increases information utility and decreases anonymity of the table because records are more distinguishable by refined values. The key is selecting the best refinement at each step with both impacts considered. At each iteration, TDR greedily selects the refinement on value v that has the highest score, in terms of the information gain ($InfoGain(v)$) per unit of anonymity loss ($AnonyLoss(v)$):

$$Score(v) = \frac{InfoGain(v)}{AnonyLoss(v)+1}, \tag{1}$$

1 is added to $AnonyLoss(v)$ to avoid division by zero. Each choice of $InfoGain(v)$ and $AnonyLoss(v)$ gives a trade-off between classification and anonymization. We borrow Shannon's information theory to measure information gain [37]. Consider a categorical attribute D_j with pre-specified taxonomy tree. Let $T^*[v]$ denote the set of records generalized to the value v and let $T^*[c]$ denote the set of records generalized to a child value c in $child(v)$ after specializing v. Let $|x|$ be the number of elements in a set x. $|T^*[v]| = \sum_c |T^*[c]|$, where $c \in child(v)$.

$$InfoGain(v) = I(T^*[v]) - \sum_c \frac{|T^*[c]|}{|T^*[v]|} I(T^*[c]), \tag{2}$$

where $I(T^*[x])$ is the *entropy* of $T^*[x]$ [37]:

$$I(T^*[x]) = -\sum_{cls} \frac{freq(T^*[x], cls)}{|T^*[x]|} \times log_2 \frac{freq(T^*[x], cls)}{|T^*[x]|}, \tag{3}$$

$freq(T[x], cls)$ is the number of data records in $T^*[x]$ having the class cls. Intuitively, $I(T^*[x])$ measures the entropy (or "impurity") of classes in $T[x]$. The more dominating the majority class in $T^*[x]$, the smaller $I(T^*[x])$ is (i.e., less entropy in $T^*[x]$). Therefore, $I(T^*[x])$ measures the error because non-majority classes are considered as errors. $InfoGain(v)$ then measures the reduction of entropy after refining v. $InfoGain(v)$ is non-negative. For more details on information gain and classification, see [32]

$$AnonyLoss(v) = avg\{A(QID_i) - A_v(QID_i)\}, \tag{4}$$

where $A(QID_i)$ and $A_v(QID_i)$ represent the anonymity before and after refining v. $avg\{A(QID_i) - A_v(QID_i)\}$ is the average loss of anonymity for all QID_i that contain the attribute of v.

If D_j is a categorical attribute without taxonomy tree, the refinement $\perp_j \to \{v, \perp_j\}$ means refining $T^*[\perp_j]$ into $T^*[v]$ and $T^{*\prime}[\perp_j]$, where $T^*[\perp_j]$ denotes the set of records containing $\perp_j$ before the refinement, $T^*[v]$ and $T^{*\prime}[\perp_j]$ denote the set of records containing v and $\perp_j$ after the refinement, respectively. We employ the same $Score(v)$ function to measure the goodness of the refinement $\perp_j \to \{v, \perp_j\}$, except that $InfoGain(v)$ is now defined as:

$$InfoGain(v) = I(T^*[\perp_j]) - \frac{|T^*[v]|}{|T^*[\perp_j]|} I(T^*[v]) - \frac{|T^{*\prime}[\perp_j]|}{|T^*[\perp_j]|} I(T^{*\prime}[\perp_j]). \tag{5}$$

Algorithm 1. Top-Down Refinement (TDR)

1. Initialize every value of D_j to the topmost value or suppress every value of D_j to $\perp_j$ or include every continuous value of D_j into the full range interval, where $D_j \in \cup QID_i$.
2. Initialize Cut_j of D_j to include the topmost value, Sup_j of D_j to include all domain values of D_j, and Int_j of D_j to include the full range interval, where $D_j \in \cup QID_i$.
3. **while** some candidate x in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$ is valid **do**
4. Find the *Best* refinement from $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.
5. Perform *Best* on T_l^* and update $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.
6. Update $Score(x)$ and validity for $x \in \langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.
7. **end while**
8. **return** Masked T_l^* and $\langle \cup Cut_j, \cup Sup_j, \cup Int_j \rangle$.

Table 2
The masked table for release, satisfying $\{\langle QID_1 = \{Education, Gender\}, 4\rangle, \langle QID_2 = \{Gender, Age\}, 11\rangle\}$.

Rec ID	Education	Gender	Age	···	Count
1–7	Junior Sec.	ANY	[1–37)	···	7
8–12	11th	ANY	[1–37)	···	5
13–16	12th	ANY	[37–99)	···	4
17–26	Bachelors	ANY	[37–99)	···	10
27–34	Grad School	ANY	[37–99)	···	8
Total				···	34

4.2.3. The anonymization algorithm (TDR)

Algorithm 1 summarizes the conceptual algorithm. All attributes not in $\cup QID_i$ are removed from T_1^*, and duplicates are collapsed into a single row with the *Class* column storing the count for each class label. Initially, Cut_j contains only the top-most value for a categorical attribute D_j with a taxonomy tree, Sup_j contains all domain values of a categorical attribute D_j without a taxonomy tree, and Int_j contains the full range interval for a continuous attribute D_j. The valid refinements in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$ form the set of *candidates*. At each iteration, we find the candidate of the highest *Score*, denoted *Best* (Line 4), apply *Best* to T^* and update $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$ (Line 5), and update *Score* and the validity of the candidates in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$ (Line 6). The algorithm terminates when there is no more candidate in $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$, in which case it returns the masked table together with the solution set $\langle \cup Cut_j, \cup Sup_j, \cup Int_j\rangle$.

The following example illustrates how to achieve a given anonymity requirement by performing a sequence of refinements, starting from the most masked table.

Example 4.2. Consider the labelled table in Table 1, where *Education* and *Gender* have pre-specified taxonomy trees and the anonymity requirement:

$$\{\langle QID_1 = \{Education, Gender\}, 4\rangle, \langle QID_2 = \{Gender, Age\}, 11\rangle\}.$$

Initially, all data records are masked to

$$\langle ANY\_Edu, ANY\_Gender, [1{-}99)\rangle$$

and

$$\cup Cut_i = \{ANY\_Edu, ANY\_Gender, [1{-}99)\}.$$

To find the next refinement, we compute the *Score* for each of *ANY_Edu*, *ANY_Gender*, and [1–99). Table 2 shows the masked data after performing the following refinements in order:

$$[1{-}99) \rightarrow \{[1{-}37), [37{-}99)\}$$
$$ANY\_Edu \rightarrow \{Secondary, University\}$$
$$Secondary \rightarrow \{JuniorSec., SeniorSec.\}$$
$$SeniorSec. \rightarrow \{11th, 12th\}$$
$$University \rightarrow \{Bachelors, GradSchool\}.$$

The solution set $\cup Cut_i$ is:

$$\{JuniorSec., 11th, 12th, Bachelors, GradSchool, ANY\_Gender, [1{-}37), [37{-}99)\}.$$

4.3. Evaluation

This step compares the raw cluster structure found in Step 1 in Section 4.1, denoted by $\mathscr{C}$, with the cluster structure found in the masked data in Step 3, denoted by $\mathscr{C}_g$. Both $\mathscr{C}$ and $\mathscr{C}_g$ are extracted from the same set of records, so we can evaluate their similarity by comparing their record groupings. We propose two evaluation methods: *F-measure* [42] and *match point*.

4.3.1. F-measure

F-measure [42] is a well-known evaluation method for cluster analysis with known cluster labels. The idea is to treat each cluster in $\mathscr{C}$ as the relevant set of records for a query, and treat each cluster in $\mathscr{C}_g$ as the result of a query. The clusters in $\mathscr{C}$ are called "natural clusters," and those in $\mathscr{C}_g$ are called "query clusters."

For a natural cluster C_i in $\mathscr{C}$ and a query cluster K_j in $\mathscr{C}_g$, let $|C_i|$ and $|K_j|$ denote the number of records in C_i and K_j respectively, let n_{ij} denote the number of records contained in both C_i and K_j, let $|T|$ denote the total number of records in T^*. The *recall*, *precision*, and *F-measure* for C_i and K_j are calculated as follows:

$$Recall(C_i, K_j) = \frac{n_{ij}}{|C_i|} \tag{6}$$

read as the fraction of relevant records retrieved by the query.

$$Precision(C_i, K_j) = \frac{n_{ij}}{|K_j|} \tag{7}$$

read as the fraction of relevant records among the records retrieved by the query.

$$F(C_i, K_j) = \frac{2 * Recall(C_i, K_j) * Precision(C_i, K_j)}{Recall(C_i, K_j) + Precision(C_i, K_j)}, \tag{8}$$

$F(C_i, K_j)$ measures the quality of query cluster K_j in describing the natural cluster C_i, by the harmonic mean of *Recall* and *Precision*.

The success of preserving a natural cluster C_i is measured by the "best" query cluster K_j for C_i, i.e., K_j maximizes $F(C_i, K_j)$. We measure the quality of $\mathscr{C}_g$ using the weighted sum of such maximum *F*-measures for all natural clusters. This measure is called the *overall F-measure* of $\mathscr{C}_g$, denoted $F(\mathscr{C}_g)$:

$$F(\mathscr{C}_g) = \sum_{C_i \in \mathscr{C}} \frac{|C_i|}{|T|} max_{K_j \in \mathscr{C}_g}\{F(C_i, K_j)\}. \tag{9}$$

Note that $F(\mathscr{C}_g)$ is in the range $[0, 1]$. A larger value indicates a higher similarity between the two cluster structures generated from the raw data and the masked data, i.e., better preserved cluster quality.

Example 4.3. Table 3 shows a cluster structure with $k = 2$ produced from the masked Table 2. The first 12 records are grouped into K_1, and the rest are grouped into K_2. By comparing with the raw cluster structure in Table 1, we can see that, among the 21 records in C_1, 19 remain in the same cluster K_2 and only 2 are sent to a different cluster. C_2 has a similar pattern. Table 4 shows the comparison between the clusters of the two structures, and Table 5 shows the *F*-measure. The overall *F*-measure is:

$$F(\mathscr{C}_g) = \frac{|C_1|}{|T|} \times F(C_1, K_2) + \frac{|C_2|}{|T|} \times F(C_2, K_1) = \frac{21}{34} \times 0.88 + \frac{13}{34} \times 0.8 = 0.85.$$

F-measure is an efficient evaluation method, but it considers *only* the best query cluster K_j for each natural cluster C_i; therefore, it does not capture the quality of other query clusters and may not provide a full picture of the similarity between two cluster structures. Thus, we propose an alternative evaluation method, called match point, to directly measure the preserved cluster structure.

Table 3
The masked labelled table for evaluation.

Rec ID	Education	Gender	Age	···	Class	Count
1–7	Junior Sec.	ANY	[1–37)	···	K_1	7
8–12	11th	ANY	[1–37)	···	K_1	5
13–16	12th	ANY	[37–99)	···	K_2	4
17–26	Bachelors	ANY	[37–99)	···	K_2	10
27–34	Grad School	ANY	[37–99)	···	K_2	8
Total						34

Table 4
The similarity of two cluster structures.

Clusters in	Clusters in Table 3	
Table 1	K_2	K_1
C_1	19	2
C_2	3	10

Table 5
The *F*-measure computed from Table 4.

$F(C_i, K_j)$	K_2	K_1
C_1	0.88	0.12
C_2	0.17	0.8

4.3.2. Match point

Intuitively, two cluster structures $\mathscr{C}$ and $\mathscr{C}_g$ are similar if two objects that belong to the same cluster in $\mathscr{C}$ remain in the same cluster in $\mathscr{C}_g$, and if two objects that belong to different clusters in $\mathscr{C}$ remain in different clusters in $\mathscr{C}_g$. To reflect the intuition, we build two square matrices $Matrix(\mathscr{C})$ and $Matrix(\mathscr{C}_g)$ to represent the grouping of records in cluster structures $\mathscr{C}$ and $\mathscr{C}_g$, respectively. The square matrices are $|T|$-by-$|T|$, where $|T|$ is the total number of records in table T. The (i,j)th element in $Matrix(\mathscr{C})$ (or $Matrix(\mathscr{C}_g)$) has value 1 if the ith record and the jth record in the raw table T (or the masked table T^*) are in the same cluster; 0 otherwise. Then, we define *match point*[3] to be the percentage of matched values between $Matrix(\mathscr{C})$ and $Matrix(\mathscr{C}_g)$:

$$MatchPoint(Matrix(\mathscr{C}), Matrix(\mathscr{C}_g)) = \frac{\sum_{1 \leqslant i,j \leqslant |T|} M_{ij}}{|T|^2}, \tag{10}$$

where M_{ij} is 1 if the (i,j)th element in $Matrix(\mathscr{C})$ and $Matrix(\mathscr{C}_g)$ have the same value; 0 otherwise. Note that match point is in the range of [0,1]. A larger value indicates a higher similarity between the two cluster structures generated from the raw data and the masked data, i.e., better preserved cluster quality.

Example 4.4. Continue from Example 4.3. Among the five records with Rec IDs 8–12, two records are not in its original clusters in $\mathscr{C}_g$. Among the 24 records with Rec IDs 13–34, three are not in its original clusters in $\mathscr{C}_g$. The match point is: $\frac{924}{34^2} = 0.80$.

4.4. Analytical discussion

We discuss some open issues and possible improvements in our proposed privacy framework for cluster analysis. Then, we present an analysis on the efficiency of the TDR algorithm.

4.4.1. Open issues and improvements

Refer to Fig. 2. One open issue is the choice of clustering algorithms employed by the data holder in Step 1. Each clustering algorithm has its own search bias or preference. Experimental results in Section 5 suggest that if the same clustering algorithm is employed in Steps 1 and 3, then the cluster structure from the masked data is very similar to the raw cluster structure; otherwise, the cluster structure in the masked data could not even be extracted. We suggest two methods for choosing clustering algorithms.

4.4.1.1. Recipient oriented. This approach minimizes the difference generated if the recipient had applied her clustering algorithm to both the raw data and the masked data. It requires the clustering algorithm in Step 1 to be the same, or to use the same bias, as the recipient's algorithm. We can implement this approach in a similar way as for determining the cluster number: either the recipient provides her clustering algorithm information, or the data holder releases one version of masked data for each popular clustering algorithm, leaving the choice to the recipient. Refer to the earlier discussion in Section 4.1 for handling potential privacy breaches caused by multiple releases.

4.4.1.2. Structure oriented. This approach focuses on preserving the "true" cluster structure in the data instead of matching the recipient's choice of algorithms. Indeed, if the recipient chooses a bad clustering algorithm, matching her choice may minimize the difference but is not helpful for cluster analysis. This approach aims at preserving the "truthful" cluster structure by employing a *robust* clustering algorithm in Steps 1 and 3. Dave and Krishnapuram [9] specified a list of requirements in order for a clustering algorithm to be robust. The principle is that "the performance of a robust clustering algorithm should not be affected significantly by small deviations from the assumed model and it should not deteriorate drastically due to noise and outliers." If the recipient employs a less robust clustering algorithm, it may not find the "true" cluster structure. This approach is suitable for the case in which the recipient's preference is unknown at the time of data release, and the data holder wants to publish only one or a small number of versions. Optionally, the data holder may release the class labels in Step 1 as a sample clustering solution. In the rest of this section, we discuss the anonymization in Step 2 and the evaluation in Step 4.

Our study in TDR focuses mainly on single-dimensional global recoding, defined in Section 3. LeFevre et al. [20,21] presented alternative masking operations, such as local recoding and multidimensional recoding, for achieving k-anonymity and its extended privacy notions. For example, in Table 1 the *Bachelors* with Rec ID# 17–22 can be generalized to *University*, while the *Bachelors* with Rec ID# 23–26 can remain ungeneralized. Compared with global recoding, local recoding and multidimensional recoding are more flexible and result in less distortion; therefore, they may further improve the preserved cluster quality in the anonymous data. Nonetheless, it is important to note that local recoding and multidimensional recoding may cause a data exploration problem: most standard data mining methods treat *Bachelors* and *University* as two independent values; but, in fact, they are not. Building a decision tree from such a generalized table may result in two branches, *Bachelors* → *class1* and *University* → *class2*. It is unclear which branch should be used to classify a new *Bachelor*. Though very

[3] We acknowledge the anonymous reviewer of DKE for suggesting this intuitive evaluation method.

important, this aspect of data utility has been ignored by all works that employed the local recoding and multidimensional recoding schemes. Data produced by global generalization and global suppression does not suffer from this data exploration problem.

4.4.2. Efficiency of TDR

Let $T_l^*[v]$ denote the set of records containing value v in a masked table T_l^*. Each iteration in TDR involves two types of work. The first type accesses data records in $T_l^*[Best]$ or $T_l^*[\perp]$ for updating the anonymity counts $a[qid_i]$ and entropy. If *Best* is an interval, an extra step is required for determining the optimal split for each child interval c in $child(Best)$. This requires making a scan on records in $T_l^*[c]$, which is a subset of $T_l^*[Best]$. To determine a split, $T_l^*[c]$ has to be sorted, which can be an expensive operation. Fortunately, resorting $T_l^*[c]$ is unnecessary for each iteration because its superset $T_l^*[Best]$ is already sorted. Thus, this type of work involves one scan of the records being refined in each iteration. The second type of work computes $Score(x)$ for the candidates $x \in \langle \cup Cut_j, \cup Int_j, \cup Int_j \rangle$ without accessing data records. For a table with m attributes and each taxonomy tree with at most p nodes, the number of such x is at most $m \times p$. This computation makes use of the maintained counts and does not access data records. Let h be the maximum number of times that a value in a record will be refined. For an attribute with a taxonomy tree, h is bounded by the height of the taxonomy tree, and for an attribute without a taxonomy tree, h is bounded by 1 (that is, a suppressed value is refined at most once). In the whole computation, each record will be refined at most $m \times h$ times and, therefore, accessed at most $m \times h$ times because only refined records are accessed. Since $m \times h$ is a small constant, independent of the table size, the TDR algorithm is linear in the table size.

Our current implementation assumes that the *qid* groups fit in memory. Often, this assumption is valid because the *qid* groups are much smaller than the original table. If the *qid* groups do not fit in the memory, we can store some *qid* groups on disk in the process of TDR, if necessary. Favorably, the memory is used to keep only *qid* groups that are smaller than the page size to avoid fragmentation of disk pages. A nice property of TDR is that the *qid* groups that cannot be further refined (that is, on which there is no candidate refinement) can be discarded, and only some statistics for them need to be kept. This likely applies to small *qid* groups in memory; therefore, the memory demand is unlikely to build up.

5. Experimental study

In this section, our objectives are to:

(1) study the loss of cluster quality for achieving various anonymity requirements;
(2) verify that the cluster-quality in the masked data produced by our cluster-oriented anonymization method is better than the cluster quality in the masked data produced by some general purpose anonymization method without a specific usage of data analysis;
(3) verify that the employment of multiple *QID*s relaxes the anonymity requirement and, therefore, improves the cluster quality;
(4) study the effects on cluster quality when the data recipient and the data holder use different clustering algorithms; and
(5) evaluate the efficiency and scalability on large data sets of the proposed anonymization method.

We employ the CLUTO-2.0 Clustering Toolkit [17], in particular, bisecting k-means [18] and basic k-means [26], to generate cluster structures in Steps 1 and 3 (refer to Section 4 and Fig. 2). These two clustering algorithms are chosen due to their popularity and wide applicability to different clustering problems [35,36,38]. We first give a brief description of both clustering algorithms. Basic k-means is a partitioning clustering algorithm. The general idea is to position k points in the space represented by the records. These k points represent the initial cluster centroid. Then, assign each record to the cluster that has the closest centroid, recompute the centroid, and repeat the computation of centroid and assignment of records until the centroid no longer moves. Bisecting k-means [18] is a divisive hierarchical clustering algorithm. It starts with a single cluster of all records and first selects a cluster (e.g., the largest cluster) to split. Then, it utilizes basic k-means to form two sub-clusters and repeats until the desired number of clusters is reached.

A naive approach to the studied privacy problem is to ignore the cluster structure and simply employ general purpose anonymization methods [20,33] to anonymize the data. So, one objective of the experiment is to compare the cluster quality, in terms of overall F-measure and match point, of our cluster quality-guided anonymization approach with the general purpose anonymization approach. To ensure a fair comparison, both approaches employ the same modified TDR anonymization method but with different *Score* functions. The overall F-measure and match point produced by different *Score* functions are labelled as follows:

- *clusterFM* and *clusterMP* denote the overall F-measure and match point, respectively, of the cluster structures before and after masking by our cluster structure-guided anonymization approach, whereas the *Score* function is specified in Eq. 1.
- *distortFM* and *distortMP* denote the overall F-measure and match point of the cluster structures before and after masking by the general purpose anonymization approach that aims at minimizing *distortion* [34]. The intuition is to charge one unit of distortion for each occurrence of value generalized to its parent value or suppressed to $\perp$. Following such intuition, at

each iteration, the *Score* function biases to refine on a value v that results in the maximum number of refined records in table T_i^*. Specifically, $Score(v) = |T_i^*[v]|$, where $|T_i^*[v]|$ is number of records containing v in T_i^*. Note, this *Score* function ignores the cluster structure.

Given the above objective measures, we can compare the two anonymization approaches in terms of cluster quality. $\frac{clusterFM - distortFM}{distortFM}$ calculates the *benefit* in F-measure of our cluster quality-guided anonymization over the general purpose anonymization. Similarly, $\frac{clusterMP - distortMP}{distortMP}$ calculates the *benefit* in match point of our cluster quality-guided benefit over the general purpose benefit. In this section, the term "benefit" refers to such ratios.

We are also interested in computing the loss in cluster quality due to masking. Both the overall *F*-measure and the match point equal to 1 if the two cluster structures, before and after the masking, are identical; therefore, $\frac{1 - clusterFM}{1} = 1 - clusterFM$ calculates the *cost* for achieving anonymity measured in *F*-measure. Similarly, $1 - clusterMP$ calculates the *cost* for achieving anonymity measured in match point. In this section, the term "cost" refers to such differences.

All experiments were conducted on an Intel Pentium IV 2.6 GHz PC with 1 Gbyte RAM. We adopted two publicly available real-life data sets, *Adult* and *Japanese Credit Screen (a.k.a. CRX)*, from the University of California, Irvine (UCI) machine learning repository [30]. Extensive experimental results for *Adult* and *CRX* are presented in Sections 5.1 and 5.2, respectively. We summarize the results in Section 5.3.

5.1. The Adult data set

The *Adult* data set contains real-life census data. It is a *de facto* benchmark for testing anonymization algorithms, previously used in [4,11,12,16,20,22–25,43–46]. After removing records with missing values, we have 45,222 records. Every record represents an individual in the United States. The data set is intended for the purpose of classification analysis, so we dropped the class attribute, but kept the six continuous attributes and eight categorical attributes; see Table 6 for their description. All 14 attributes are used in cluster analysis. We used discretization and generalization to mask the continuous and categorical attributes. The taxonomy trees for categorical attributes are adopted from [12]. The continuous attributes are normalized as a standard preprocessing step in many clustering algorithms. The taxonomy trees for continuous attributes are dynamically generated by our TDR algorithm.

In Section 5.1.1, we present the results for the scenario called *homogeneous clustering*, where the same clustering algorithm is applied in both Steps 1 and 3. In Section 5.1.5, we present the results for the scenario called *heterogeneous clustering*, where different clustering algorithms are applied in Steps 1 and 3. In Section 5.1.6, we evaluate the efficiency of the proposed method.

5.1.1. Homogenous clustering

Homogeneous clustering refers to the scenario in which the same clustering algorithm is applied in both Steps 1 and 3. In the scenario it models, the recipient applies the same clustering algorithm as the one used by the data holder. We first evaluate the cost and benefit, in terms of cluster quality, of employing our proposed method for anonymity requirements with a single *QID*. Next, we study how the cluster quality is influenced by the anonymity threshold and *QID* size. Then, we study cluster quality for anonymity requirement with multiple *QID*s.

5.1.1.1. Single QID. Observation 3.1 implies that for the same anonymity threshold, a single *QID* is always more restrictive than breaking it into multiple *QID*s. We first consider the case of a single *QID*. To ensure that the *QID* contains attributes that have an impact on clustering, we use the C4.5 classifier [32] to rank the attributes with respect to the raw cluster labels. The

Table 6
The attributes for *Adult* data set.

Attribute	Type	Numerical range	
		# of Leaves	# of Levels
Age (Ag)	continuous	17–90	
Capital-gain (Cg)	continuous	0–99999	
Capital-loss (Cl)	continuous	0–4356	
Education-num (En)	continuous	1–16	
Final-weight (Fw)	continuous	13492–1490400	
Hours-per-week (Hw)	continuous	1–99	
Education (Ed)	categorical	16	5
Marital-status (Ms)	categorical	7	4
Native-country (Nc)	categorical	40	5
Occupation (Oc)	categorical	14	3
Race (Ra)	categorical	5	3
Relationship (Re)	categorical	6	3
Sex (Se)	categorical	2	2
Work-class (Wc)	categorical	8	5

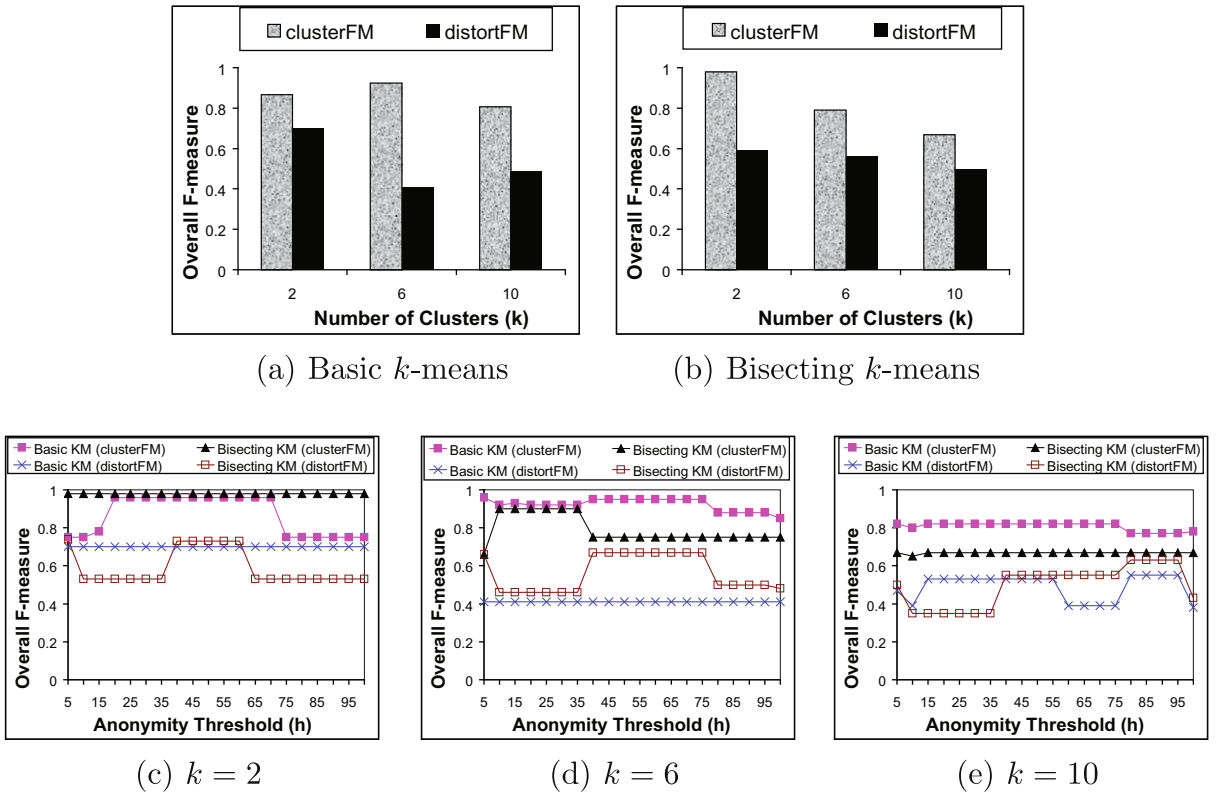


Fig. 3. Overall *F*-measure for homogeneous clustering on *Adult*.

top rank attribute is the attribute at the top of the C4.5 decision tree. Then, we remove the top attribute and repeat this process to determine the rank of other attributes. In our experiments, Top9 denotes the anonymity requirement in which the *QID* contains the top 9 attributes. Note that these attributes depend on the raw cluster structure, which depends on the cluster number and clustering algorithm for extraction. For example, for *k*-means algorithm with $k = 6$, Top9 is *En*, *Nc*, *Ms*, *Oc*, *Ed*, *Re*, *Ag*, *Ra*, *Se*, ranked in that order.

5.1.2. Benefit

Fig. 3a shows the averaged *clusterFM* and *distortFM* over anonymity thresholds $5 \leqslant h \leqslant 100$ for basic *k*-means. The benefit of our cluster quality-guided anonymization over the general purpose anonymization spans from 24% to 125% for the number of clusters $k = 2, 6, 10$. To show the benefit is statistically significant, we conducted a one-tail *t*-test on the 20 pairs of test cases on $5 \leqslant h \leqslant 100$. The *p*-values for $k = 2, 6, 10$ are 5.02E−7, 8.97E−25, 6.14E−14, respectively. Fig. 3b shows the results for the same experimental setting for bisecting *k*-means. The benefit spans from 34% to 66%. The *p*-values for $k = 2, 6, 10$ are 7.44E−14, 2.55E−6, 4.72E−7, respectively, showing that the benefit is statistically significant at $\alpha = 5\%$ and the benefit is unlikely to have occurred by chance.

Fig. 4a shows the averaged *clusterMP* and *distortMP* over anonymity thresholds $5 \leqslant h \leqslant 100$ for basic *k*-means. The benefit spans from 28% to 85% for the number of clusters $k = 2, 6, 10$. The *p*-values for $k = 2, 6, 10$ are 7.94E−8, 3.72E−22, 1.67E−10, respectively. Fig. 4b shows the results for the same experimental setting for bisecting *k*-means. The benefit spans from 5% to 32%. The *p*-values for $k = 2, 6, 10$ are 5.71E−34, 4.24E−5, 1.07E−8, respectively, showing that the benefit is statistically significant at $\alpha = 5\%$. Both measures in overall F-measure and match point suggest that our cluster quality-guided anonymization generally preserves better cluster quality than the general purpose anonymization on the *Adult* data set.

5.1.3. Cost

Consider Figs. 3a, b and 4a, b again. The averaged cost, measured in overall *F*-measure, for achieving a given anonymity requirement spans from 2% to 20% for basic *k*-means and bisecting *k*-means at cluster numbers $k = 2$ and $k = 6$. The averaged cost, measured in match point, spans from 9% to 23% for basic *k*-means and bisecting *k*-means at cluster numbers $k = 2, 6, 10$. In general the loss of cluster quality is mild and the raw cluster structure has been preserved. There is an exception. For example, the cost increases to 33% in Fig. 3b at $k = 10$, indicating that the number of clusters *k* plays an important role in the preserved cluster quality. It also strengthens the importance of the evaluation phase (Steps 3 and 4) in our framework

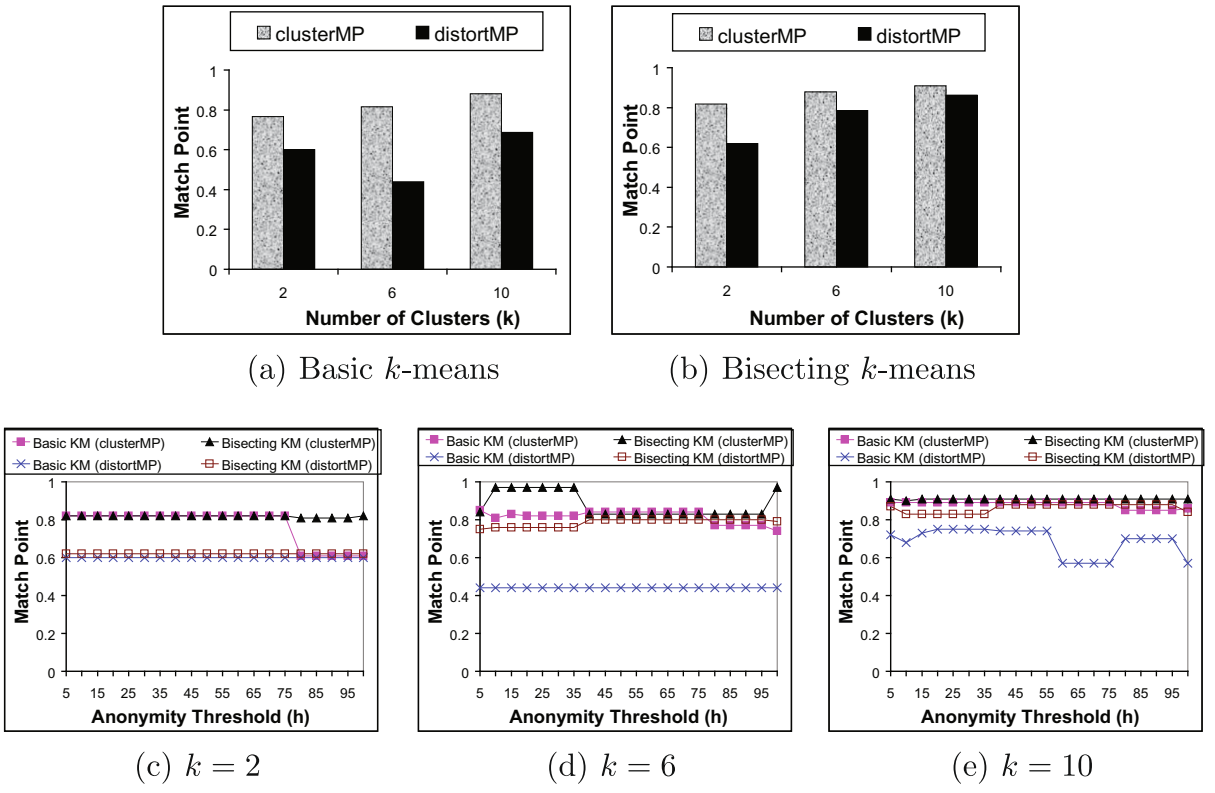


Fig. 4. Match point for homogeneous clustering on *Adult*.

because it provides the data holder an opportunity to evaluate the cluster quality before releasing the data. If the loss is large, e.g., at $k = 10$, the data holder may consider releasing an alternative version with a different number of clusters k, which usually is not a hard constraint. The problem of determining the cluster number is part of cluster analysis, not a new issue in our anonymization problem.

5.1.4. Sensitivity to anonymity threshold h

Figs. 3c–e and 4c–e plot the *clusterFM*, *distortFM*, *clusterMP*, and *distortMP* of anonymity thresholds $5 \leqslant h \leqslant 100$ at $k = 2$, 6, 10 for basic k-means and bisecting k-means. Each data point in the figures represent one test case. We made two observations from these figures.

(1) Both *clusterFM* and *clusterMP* span narrowly with a difference less than 0.2, suggesting that the cluster quality is not sensitive to the increase of anonymity threshold. The result also suggests that both basic and bisecting k-means are robust enough to recapture the clustering structures from the generalized data with different anonymity thresholds h and numbers of clusters k. We examined the masked data closely and found that, for example, the masked data at $h = 20$ is identical to the masked data at $h = 70$ in Fig. 3c, meaning that the same masked version has room to satisfy a broad range for anonymity thresholds h.

(2) Both overall F-measure and match point do not decrease monotonically with respect to the increase of h because both the TDR anonymization algorithm and the clustering algorithms do not aim at identifying the global optimal solution. As a result, in *some* test cases, the masked data with higher anonymity threshold h may result in higher preserved cluster quality in the evaluation. However, if the anonymity threshold is increased to some unreasonable range, say $h = 5000$, then the cluster quality will be completely destroyed and both overall F-measure and match point will drop significantly to below 0.1. Thus, in general, there is a trend that the clustering quality degrades as the anonymity threshold increases, but the trade-off is not obvious when h is relatively small (e.g., $h \leqslant 100$) compared to the number of records (e.g., 45,222 records). We will revisit the influence of anonymity threshold in the experiment on *CRX*, which is a smaller data set, in Section 5.2.

In addition to F-measure and match point, we also manually examined the cluster structures generated from the raw data and from the masked data in the case of $h = 120$ and $k = 6$ as shown in Table 7. The overall F-measure is 0.90 and the match point is 0.97. Note that the cluster labels are arbitrarily named. Among the 12,669 records in the natural cluster C_1, 12,655

Table 7
The similarity of two cluster structures ($h = 120$ and $k = 6$).

Clusters in	Clusters in Masked T_l^*					
Unmodified T_l	K_2	K_5	K_1	K_3	K_6	K_4
C_1	12655	0	0	0	13	1
C_2	0	6198	0	0	19	22
C_3	0	0	6513	0	0	59
C_4	0	0	0	4239	386	0
C_5	0	0	0	3846	3171	0
C_6	0	0	0	0	0	8100

remain together in the new cluster K_2. So, C_1 is almost perfectly preserved. The natural cluster C_5 is less preserved as its members are split between two new clusters.

A closer look at the masked data reveals that among the nine top ranked attributes, four are generalized to a different degree of granularity, and five, namely *Nc* (ranked 2nd), *Oc* (ranked 4th), *Ed* (ranked 5th), *Re* (ranked 6th), and *Ra* (ranked 8th), are generalized to the topmost value *ANY*. Even for this drastic masking, the overall F-measure and match point remain at 0.90 and 0.97, respectively. This suggests that there is much room for masking within the constraint of preserving the cluster structure. Such room comes from the fact that some values are unnecessarily specific for cluster analysis, and masking them to less specific values does not affect the cluster structure. Our approach seizes the opportunity provided by this flexibility for masking identifying information.

We also conducted some experiments to study how the structure of a taxonomy tree could influence the generalization on a categorical attribute and the overall cluster quality. In general, a taller taxonomy tree increases the flexibility of generalization because domain values have more opportunities to be generalized into different granularity. As a result, masking is reduced and the overall cluster quality is improved. However, data could become hard to interpret if a taxonomy tree is unreasonably tall. For the dynamically generated taxonomy tree in continuous attributes, we also examined the split points computed by information gain. Fig. 5 shows the dynamically generated taxonomy tree for *Education-num* (*En*). The split point at 13 (the years of education, not age) is very reasonable because it indicates whether a person has post-secondary education.

5.1.4.1. Sensitivity to QID size. Fig. 6 studies the influence of *QID* size on the preserved cluster quality. Fig. 6a has two portions. The left and right portions, separated by a vertical dashed line, show the averaged overall F-measure over anonymity thresholds $5 \leqslant h \leqslant 100$ for basic *k*-means and bisecting *k*-means, respectively, at $k = 6$ with *QID* size from three attributes (Top3) to nine attributes (Top9). Fig. 6b shows the averaged match point for the same experimental setting. Both overall

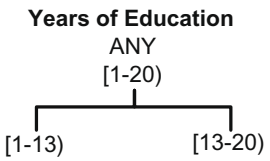
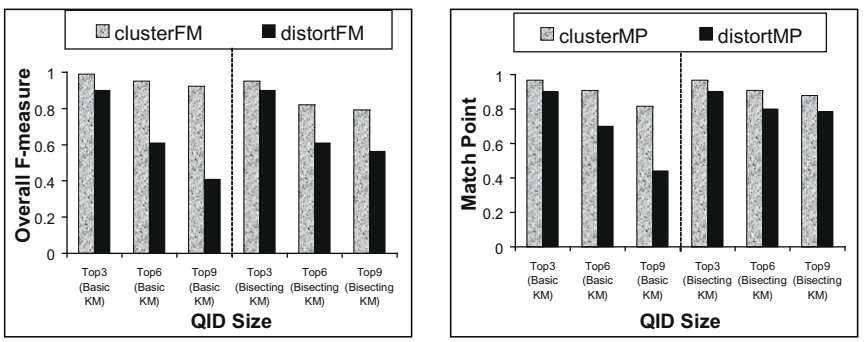


Fig. 5. The generated taxonomy tree for *Education-num*.



(a) F-measure (b) Match point

Fig. 6. Increasing *QID* size for $k = 6$ on *Adult*.

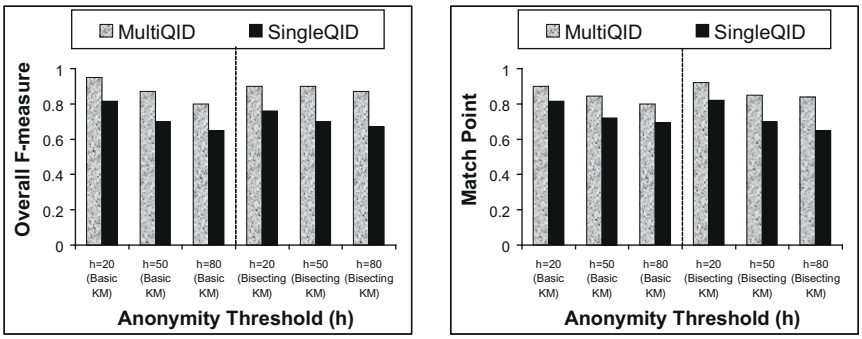


Fig. 7. MultiQID vs. SingleQID for $k = 6$ on *Adult*.

F-measure and match point exhibit a similar pattern and suggest that as the *QID* increase, the preserved cluster quality decreases because more attributes are included for masking. It is interesting to note that as *QID* size decreases, the benefit of the cluster quality-guided anonymization over the general purpose anonymization becomes smaller because the non-*QID* attributes dominate the clustering effect and, therefore, diminish the difference between the two anonymization methods.

5.1.4.2. Multiple QIDs. To verify the claim that multi-*QID* anonymity requirements can help reduce unnecessary masking, we compared the overall *F*-measure between a multi-*QID* requirement and the *corresponding* single *QID* requirement, where the *QID* is the union of the multiple *QID*s. For example, a requirement of three length-2 *QID*s is

$$\{\langle\{Ag, En\}, h\rangle, \langle\{Ag, Re\}, h\rangle, \langle\{Se, Hw\}, h\rangle\}$$

and the corresponding single *QID* requirement is

$$\{\langle\{Ag, En, Re, Se, Hw\}, h\rangle\}.$$

We randomly generated 30 multi-*QID* requirements as follows. For each requirement, we first determined the number of *QID*s using the uniform probability distribution $U[3,7]$ (i.e., randomly drew a number between 3 and 7 where the probability of selecting each number was the same) and the length of *QID*s using $U[2,9]$. For simplicity, all *QID*s in the same requirement had the same length and same threshold h. For each *QID*, we randomly selected attributes according to the *QID* length from the 14 attributes. A repeating *QID* was discarded.

Fig. 7 studies the effect on the preserved cluster quality between multi-*QID*, denoted by MultiQID, and its corresponding single *QID*, denoted by SingleQID. Fig. 7(a) has two portions. The left and right portions show the averaged overall *F*-measure over the 30 randomly generated test cases described above at $h = 20, 50, 80$ for basic k-means and bisecting k-means, respectively, where $k = 6$. To show the difference between MultiQID and SingleQID is statistically significant, we conducted a one-tail t-test on the 30 pairs of test cases. The p-values for basic k-means at $h = 20, 50, 80$ are 4.20E−6, 9.27E−6, 2.27E−5, respectively. The p-values for bisecting k-means at $h = 20, 50, 80$ are 2.05E−9, 2.49E−9, 6.94E−8, respectively. The difference between MultiQID and SingleQID is statistically significant at $\alpha = 5\%$. Fig. 7b shows the averaged match point for the same experimental setting and exhibits similar results, so we omit the explanation. Fig. 7 suggests that a multi-*QID* anonymity requirement generally results in higher cluster quality than its corresponding single *QID* anonymity requirement.

5.1.5. Heterogeneous clustering

Heterogeneous clustering refers to the case that different clustering algorithms are applied in Steps 1 and 3. It models the scenario that the data recipient applies a clustering algorithm to the masked data that is different from the one used by the data holder for masking the data. We applied bisecting and basic k-means in Steps 1 and 3 in two different orders, denoted by (*Basic KM* → *Bisecting KM*) and (*Bisecting KM* → *Basic KM*), respectively, in Fig. 8. In both cases, compared to the homogenous clustering in Figs. 3-4, there is a very severe drop on *clusterFM* and *clusterMP*. The drop on *clusterFM* and *clusterMP* spans from 33% to 78%, and from 16% to 50%, respectively. To explain the drops, we encoded two raw cluster structures separately using the two clustering methods (without any masking), and then measured the overall *F*-measure and match point between the two raw cluster structures, denoted by *Raw Data* in the figures. Because the drops on *clusterFM* and *clusterMP* of *Raw Data* are also severe, we can conclude that the drops on (*Basic KM* → *Bisecting KM*) and (*Bisecting KM* → *Basic KM*) are caused by the nature of heterogeneous clustering, not by the masking.

The above studies suggest that if the data recipient applies the same clustering algorithm as the one used by the data holder for masking the data, the cluster structure obtained will be more similar to the raw cluster structure because the second clustering could extract the embedded structure preserved in the masked data. In contrast, if different clustering

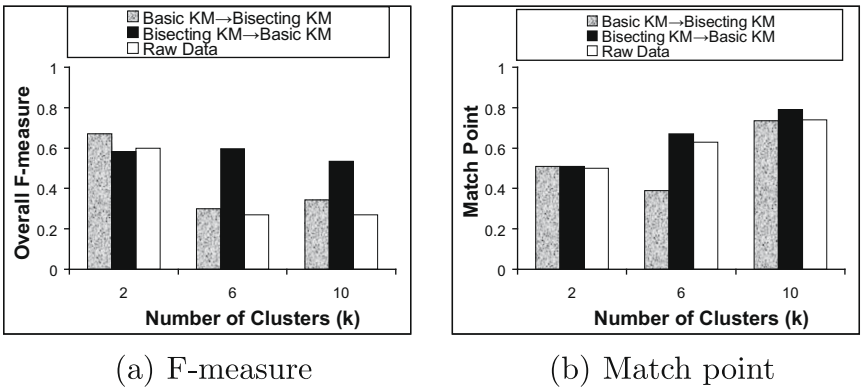


Fig. 8. Heterogeneous clustering on *Adult*.

algorithms are used, the structure preserved by masking may not be useful to the second clustering due to a different search bias. This explains the significant drops in overall F-measure and match point for heterogeneous clustering.

5.1.6. Efficiency and scalability

We evaluated the efficiency and scalability of the TDR anonymization algorithm, which is capable of masking continuous attributes and categorical attributes with and without pre-specified taxonomy trees. For all previous experiments on *Adult*, TDR takes at most 10 s to complete. Out of the 10 s, approximately 8 s are spent on reading data records from disk and writing the masked data to disk. The actual processing time for masking the data is relatively short.

The next experiment evaluates the scalability of TDR by adding some randomly generated records to the *Adult* data set, which originally had 45,222 records. For each raw record r in *Adult*, we created $\alpha - 1$ "variations" of r, where $\alpha > 1$ is the expansion scale. For each variation of r, we randomly selected q attributes from $\cup QID_i$, where q has the uniform probability distribution $U[1, |\cup QID_j|]$, and replaced the values on the selected attributes with values randomly drawn from the domain of the attributes. Together with all raw records, the expanded data set had $\alpha \times 45{,}222$ records. To provide a precise evaluation, the runtime reported excludes the time for loading data records from disk and the time for writing the masked data to disk.

Fig. 9 depicts the runtime of TDR, using generalization and discretization, for 200,000 to 1 million data records at the anonymity threshold $h = 50$ and the cluster number $k = 6$ with two types of anonymity requirements. AllAttQID refers to the single *QID* having all 14 attributes. This is one of the most time-consuming settings because of the large number of candidate refinements to consider at each iteration. For TDR, the small anonymity threshold of $h = 50$ requires more iterations to reach a solution, hence more runtime, than a larger threshold. TDR takes approximately 80 s to transform 1 million records.

In Fig. 9, MultiQID refers to the average runtime over the 30 random multi-*QID* requirements generated as described in Section 5.1.1 with $h = 50$ and $k = 6$. Compared to AllAttQID, TDR becomes less efficient for handling multi-*QID* because an anonymity requirement on multi-*QID* is a less restrictive constraint than the single *QID* anonymity requirement containing all attributes; therefore, TDR has to perform more refinements to reach a local optimal solution. The runtime of suppression on this expanded data set is roughly the same as shown in Fig. 9, so we omit the figure.

5.2. The Japanese credit screening (CRX) data set

The *Japanese Credit Screening* data set, also known as *CRX*, contains real-life credit card applications. It was previously used in [45,46]. With the binary class attribute removed, there are six continuous attributes and nine categorical attributes on 653

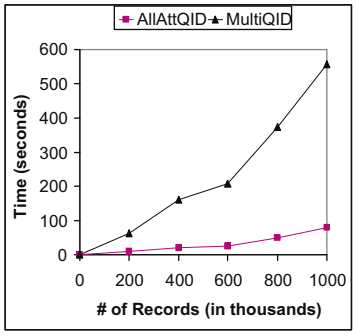


Fig. 9. Scalability ($h = 50$ and $k = 6$).

records. Every record represents a credit card application of an individual in Japan. In the UCI repository, all values and attribute names in *CRX* have been changed to meaningless symbols, e.g., $A_1 \ldots A_{15}$. The continuous attributes are normalized as a standard preprocessing step in many clustering algorithms. No taxonomy trees are given for the categorical attributes, so we use discretization and suppression to mask the continuous and categorical attributes. All 15 attributes are used in cluster analysis.

Following the same experimental settings for *Adult* in Section 5.1, we present the result of *CRX* below. In general, the result exhibits patterns similar to those shown in Section 5.1 The only difference is that the cluster quality in *CRX* is more sensitive to the increase of anonymity threshold h due to its smaller data set size. We first present the results for the scenario of homogenous clustering, followed by heterogeneous clustering.

5.2.1. Homogenous clustering

We first evaluate the cost and benefit, in terms of cluster quality, of employing our proposed method for anonymity requirements with single *QID*. Next, we study how the cluster quality is influenced by the anonymity threshold and *QID* size. Then, we study cluster quality for anonymity requirement with multiple *QID*s.

5.2.1.1. Single QID. For each test case, we used the same method described in Section 5.1 to identify the top nine most important attributes to form the anonymity requirement, denoted by Top9.

5.2.1.2. Benefit. Fig. 10a shows the averaged *clusterFM* and *distortFM* over anonymity thresholds $5 \leqslant h \leqslant 100$ for basic k-means. The benefit of our cluster quality-guided anonymization over the general purpose anonymization spans from 40% to 155% for the number of clusters $k = 2, 6, 10$. To show the benefit is statistically significant, we conducted a one-tail t-test on the 20 pairs of test cases from $5 \leqslant h \leqslant 100$. The p-values for $k = 2, 6, 10$ are 2.24E−26, 4.67E−17, 1.40E−10, respectively. Fig. 10b shows the results for the same experimental setting for bisecting k-means. It is interesting to note that the raw cluster structure is perfectly preserved at $k = 2$ for both *clusterFM* and *distortFM* in all 20 test cases of anonymity threshold $5 \leqslant h \leqslant 100$, so the benefit and the cost are 0. The benefit at $k = 6$ and $k = 10$ are 45% and 57%, respectively. The p-values for $k = 10$ and $k = 10$ are 4.25E−9 and 1.11E−10, respectively, showing that the benefit is statistically significant at $\alpha = 5\%$; the benefit is unlikely to have occurred by chance.

Fig. 11a shows the averaged *clusterMP* and *distortMP* over anonymity thresholds $5 \leqslant h \leqslant 100$ for basic k-means. The benefit spans from 9% to 60% for the number of clusters $k = 2, 6, 10$. The p-values for $k = 2, 6, 10$ are 1.67E−16, 8.50E−14,

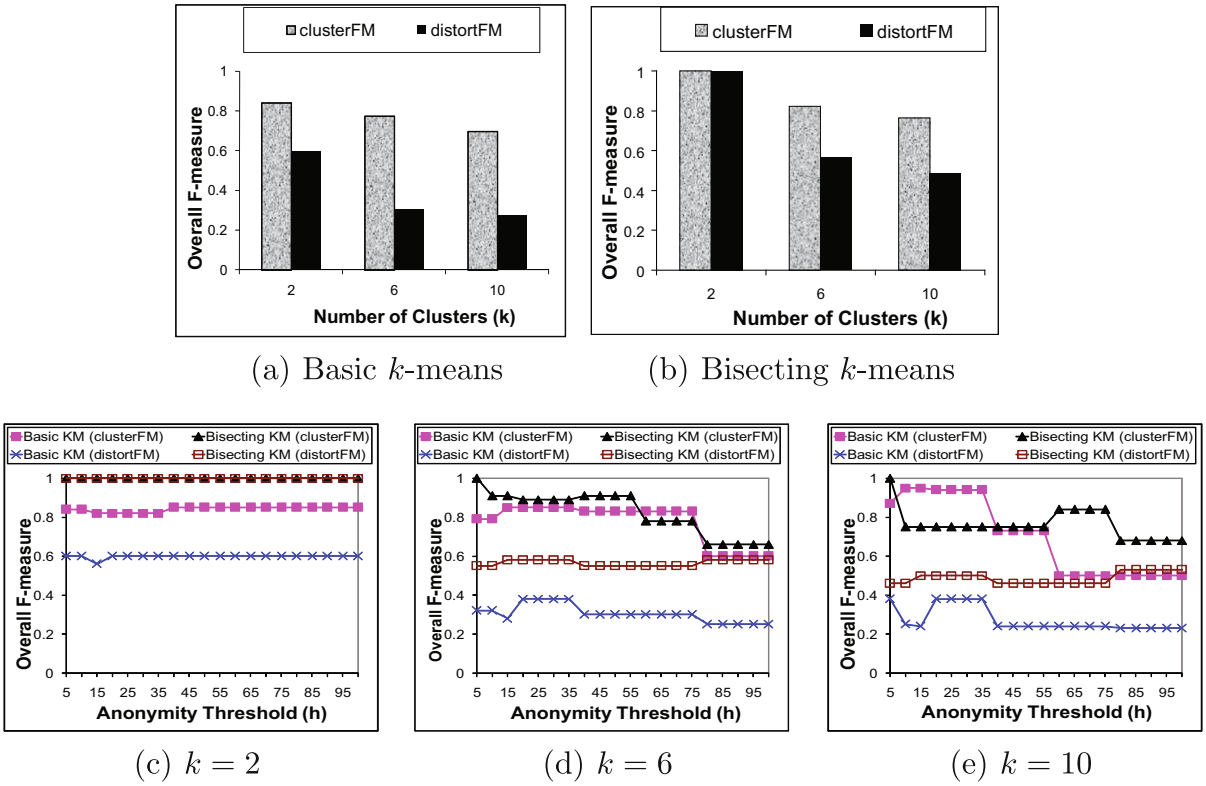


Fig. 10. Overall *F*-measure for homogeneous clustering on *CRX*.

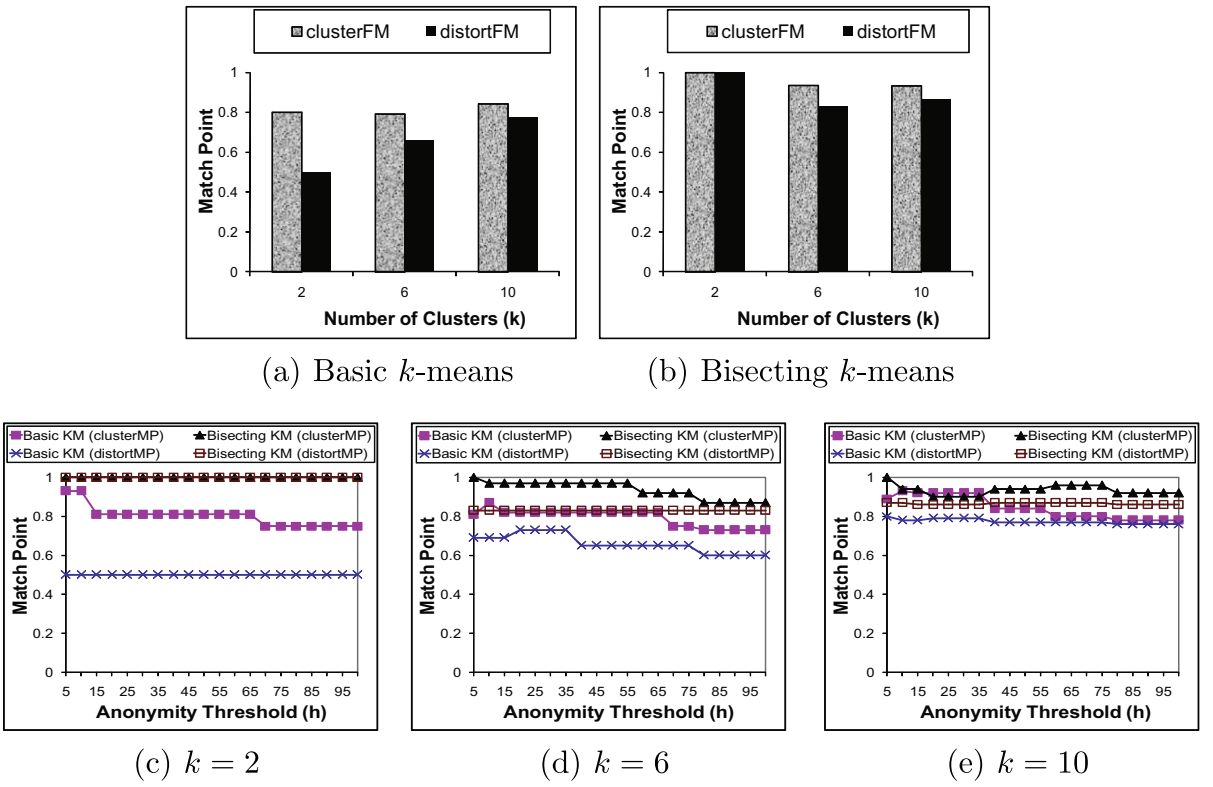


Fig. 11. Match point for homogeneous clustering on *CRX*.

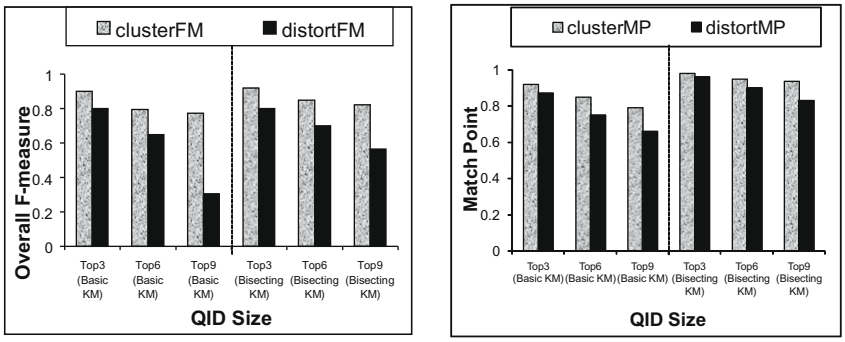
1.81E−6, respectively. Fig. 11b shows the results for the same experimental setting for bisecting k-means. The benefit at $k = 6$ and $k = 10$ are 13% and 8%, respectively. The p-values for $k = 6$ and $k = 10$ are 1.04E−9 and 1.06E−11, respectively, showing that the benefit is statistically significant at $\alpha = 5\%$. The results suggest that our cluster quality-guided anonymization generally preserves better cluster quality than the general purpose anonymization on *CRX*.

5.2.1.3. Cost. Consider Figs. 10a, b and 11a, b again. The averaged cost, measured in overall F-measure, for achieving a given anonymity requirement spans from 0% to 23% for basic k-means and bisecting k-means at cluster numbers $k = 2$ and $k = 6$. The averaged cost, measured in match point, spans from 0% to 20% for basic k-means and bisecting k-means at cluster numbers $k = 2, 6, 10$. The results suggest that the cost is mild and support the claim that the cluster structure can be preserved for restrictive anonymity requirements. However, the cost increases to 30% when $k = 10$ in Fig. 10a. Thus, it strengthens our claim that it is important to evaluate the cluster quality before releasing the data.

5.2.1.4. Sensitivity to anonymity threshold h. Figs. 10c–e and 11c–e plot the *clusterFM*, *distortFM*, *clusterMP*, and *distortMP* for basic k-means and bisecting k-means with anonymity thresholds $5 \leqslant h \leqslant 100$ at $k = 2, 6, 10$. From the figures, we noted that at $k = 6$ and $k = 10$, both *clusterFM* and *clusterMP* are more sensitive to the increase of anonymity threshold h, and the preserved cluster quality generally degrades as h increases. The trade-off between cluster quality and anonymity is more obvious in *CRX* than in *Adult* because the data set size of *CRX* (653 records) is much smaller than the data set size of *Adult* (45,222 records); therefore, increasing the anonymity threshold h, for example, from 20 to 80, requires more maskings in *CRX* but not in *Adult*. The data set size influences the required masking for achieving a given anonymity requirement and indirectly affects the cluster quality.

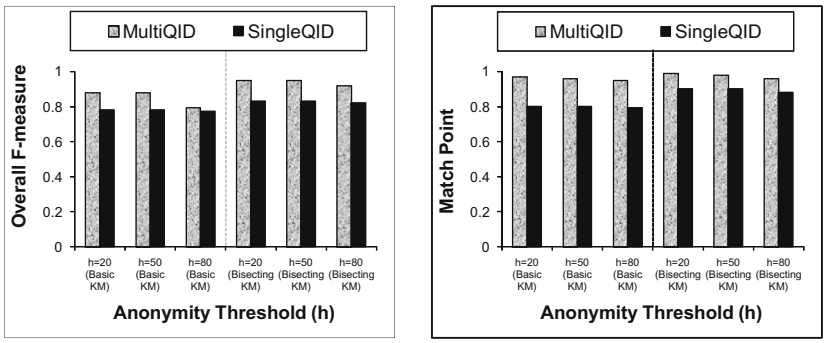
5.2.1.5. Sensitivity to QID size. Fig. 12 illustrates how the *QID* size affects the preserved cluster quality. As the result is similar to the result in Fig. 6 on *Adult*, we omit the explanation here.

5.2.1.6. Multiple QIDs. Fig. 13 shows the effect on the preserved cluster quality between MultiQID and SingleQID. Fig. 13a has two portions. The left and right portions show the averaged overall F-measure over the 30 randomly generated test cases described in Section 5.1 at $h = 20, 50, 80$ for basic k-means and bisecting k-means, respectively, where $k = 6$. The p-values for basic k-means at $h = 20, 50, 80$ are 2.12E−9, 6.25E−6, 3.12E−3, respectively. The p-values for bisecting k-means at $h = 20, 50, 80$ are 1.03E−12, 2.49E−12, 1.42E−12, respectively. Fig. 13b shows the averaged match point for the same



(a) F-measure (b) Match point

Fig. 12. Increasing *QID* size for $k = 6$ on *CRX*.



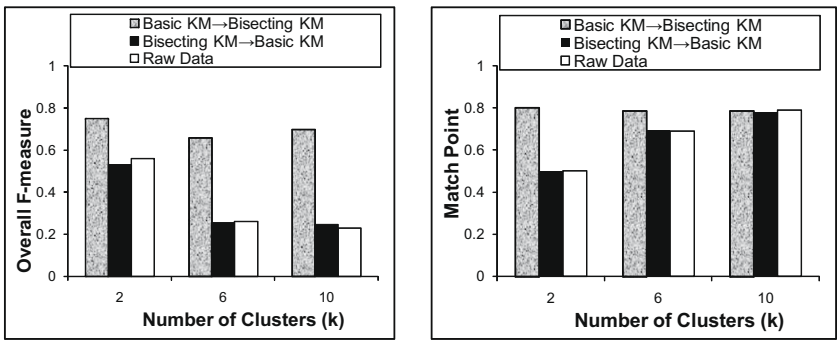
(a) F-measure (b) Match point

Fig. 13. MultiQID vs. SingleQID for $k = 6$ on *CRX*.

experimental setting. The difference between MultiQID and SingleQID is statistically significant at $\alpha = 5\%$. Although the result suggests that an anonymized data set satisfying a multi-*QID* anonymity requirement generally yields higher cluster quality than an anonymized data set satisfying its corresponding single *QID* anonymity requirement, the difference is not very large on *CRX*.

5.2.2. Heterogeneous clustering

Fig. 14 depicts the averaged overall *F*-measure and averaged match point over anonymity thresholds $0 \leqslant h \leqslant 100$ for heterogeneous clustering on *CRX* at cluster numbers $k = 2, 6, 10$. Compared to homogenous clustering, there is a very severe drop on both the overall *F*-measure and the match point for (*Bisecting KM* → *Basic KM*). We notice that the drop is caused by



(a) F-measure (b) Match point

Fig. 14. Heterogeneous clustering on *CRX*.

heterogeneous clustering, not by masking, because the overall *F*-measure and match point also show severe drops on *Raw Data*, which does not involve masking. Note, the drop for (*Basic KM* → *Bisecting KM*) is less severe than the drop for (*Bisecting KM* → *Basic KM*), implying that bisecting *k*-means can extract the cluster structure encoded by basic *k*-means in the masked data. Interestingly, the overall *F*-measure for (*Bisecting KM* → *Basic KM*) is even better than the overall F-measure on *Raw Data* because the masking removes some overly specific information (i.e., noise) from the data.

5.3. Summary

These experiments verified the claim that the proposed approach of converting the anonymity problem for cluster analysis to the counterpart problem for classification analysis is effective. This is demonstrated by the preservation of most of the cluster structure in the raw data after masking identifying information for a broad range of anonymity requirements. The experimental results also suggest that our cluster quality-guided anonymization can preserve better cluster structure than the general purpose anonymization.

The experiments demonstrated the cluster quality with respect to the variation of anonymity thresholds, *QID* size, and number of clusters. In general, the cluster quality degrades as the anonymity threshold increases. This trend is more obvious if the data set size is small or if h is relatively large, e.g., $h = 5000$. The cluster quality degrades as the *QID* size increases. The cluster quality exhibits no obvious trend with respect to the number of clusters, as the natural number of clusters is data dependent.

The experiments confirmed that the specification of the multi-*QID* anonymity requirement helps avoid unnecessary masking and, therefore, preserves more of the cluster structure. However, if the data recipient and the data holder employ different clustering algorithms, then there is no guarantee that the encoded raw cluster structure can be extracted. Thus, in practice, it is important for the data holder to validate the cluster quality, using the evaluation methods proposed in our framework, before releasing the data. Finally, experiments suggest that the proposed anonymization approach is highly efficient and scalable for single *QID*, but less efficient for multi-*QID*.

6. Extension: beyond anonymity

The above approach provides a flexible framework that makes use of existing solutions as "plug-in" components. These include the cluster analysis in Steps 1 and 3, the anonymization in Step 2, and the evaluation in Step 4. For example, instead of using the proposed TDR algorithm, the data holder has the option to perform the anonymization by employing one of the genetic algorithms [16], top-down specialization [12], or InfoGain Mondrian [22]. None of them can perform all three types of masking operations discussed in Section 3, so some modification is necessary.

This paper focuses on preventing the privacy threats caused by sensitive record linkage, but the framework can also prevent sensitive attribute linkage by adopting different anonymization algorithms and achieving other privacy requirements, such as ℓ-diversity [25] and confidence bounding [45,46], discussed in Section 2. The extension requires modification of the *Score* or cost functions in these algorithms to bias on refinements or maskings that can distinguish class labels. The framework can also adopt other evaluation methods, such as entropy [37], or any ad-hoc methods defined by the data holder. For future work, we are interested in building a visualization tool to allow the data holder to adjust the parameters, such as the number of clusters and anonymity thresholds, and visualize their influence on the clusters interactively.

7. Conclusions

We studied the problem of releasing person-specific data for cluster analysis while protecting privacy. The proposed solution is to mask unnecessarily specific information into a less specific but semantically consistent version, so that person-specific identifying information is masked but essential cluster structure remains. The major challenge is the lack of class labels that could be used to guide the masking process. Our main contribution is a general framework for converting this problem into the counterpart problem for classification analysis so that the masking process can be properly guided. The key idea is to encode the original cluster structure into the class label of data records and subsequently preserve the class labels for the corresponding classification problem. The experimental results verified the effectiveness of this approach.

We also studied several practical issues arising from applying this approach in a real-life data publishing scenario. These include how the choices of clustering algorithms, number of clusters, anonymity threshold, and size and type of quasi-identifiers can affect the effectiveness of this approach, and how to evaluate the effectiveness in terms of cluster quality. These studies lead to the recommendation of two strategies for choosing the clustering algorithm in the masking process, each having a different focus. The contribution in this paper provides a useful framework of secure data sharing for the purpose of cluster analysis.

Acknowledgements

The research is supported in part by the Discovery Grants (356065-2008) from the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Faculty Start-up Funds from Concordia University.

References

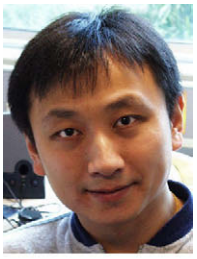
[1] C.C. Aggarwal, On *k*-anonymity and the curse of dimensionality, in: Proceedings of the 31st Very Large Data Bases (VLDB), Trondheim, Norway, 2005.
[2] C.C. Aggarwal, P.S. Yu, A condensation approach to privacy preserving data mining, in: Proceedings of the 9th International Conference on Extending Database Technology (EDBT), 2004.
[3] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, A. Zhu, Achieving anonymity via clustering, in: Proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Principles of Database Systems (PODS), 2006.
[4] R.J. Bayardo, R. Agrawal, Data privacy through optimal *k*-anonymization, in: Proceedings of the 21st IEEE International Conference on Data Engineering (ICDE), 2005.
[5] J. Beringer, E. Hüllermeier, Online clustering of parallel data streams, Data and Knowledge Engineering (DKE) 58 (2) (2006) 180–204.
[6] D. Birant, A. Kut, ST-DBSCAN: an algorithm for clustering spatial-temporal data, Data and Knowledge Engineering (DKE) 60 (1) (2007) 208–221.
[7] D.M. Carlisle, M.L. Rodrian, C.L. Diamond, California inpatient data reporting manual, medical information reporting for california, fifth ed., Tech. rep., Office of Statewide Health Planning and Development, July 2007.
[8] R. Chaytor, Utility-preserving *k*-anonymity, Master's thesis, Memorial University of Newfoundland, Canada, 2006.
[9] R.N. Dave, R. Krishnapuram, Robust clustering methods: a unified view, IEEE Transactions on Fuzzy Systems 5 (2) (1997) 270–293.
[10] B.C.M. Fung, M. Cao, B.C. Desai, H. Xu, Privacy protection for RFID data, in: Proceedings of the 24th ACM SIGAPP Symposium on Applied Computing (SAC) Special Track on Database Theory, Technology, and Applications (DTTA), ACM Press, Honolulu, HI, 2009.
[11] B.C.M. Fung, K. Wang, A.W.C. Fu, J. Pei, Anonymity for continuous data publishing, in: Proceedings of the 11th International Conference on Extending Database Technology (EDBT), ACM Press, Nantes, France, 2008.
[12] B.C.M. Fung, K. Wang, P.S. Yu, Top-down specialization for information and privacy preservation, in: Proceedings of the 21st IEEE International Conference on Data Engineering (ICDE), Tokyo, Japan, 2005.
[13] B.C.M. Fung, K. Wang, P.S. Yu, Anonymizing classification data for privacy preservation, IEEE Transactions on Knowledge and Data Engineering (TKDE) 19 (5) (2007) 711–725.
[14] R. Gelbard, O. Goldman, I. Spiegler, Investigating diversity of clustering methods: an empirical comparison, Data and Knowledge Engineering (DKE) 63 (1) (2007) 155–166.
[15] A. Inan, S.V. Kaya, Y. Saygin, E. Savas, A.A. Hintoglu, A. Levi, Privacy preserving clustering on horizontally partitioned data, Data and Knowledge Engineering (DKE) 63 (3) (2007) 646–666.
[16] V.S. Iyengar, Transforming data to satisfy privacy constraints, in: Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2002.
[17] G. Karypis, Cluto – family of data clustering software tools, October 2006. <http://glaros.dtc.umn.edu/gkhome/views/cluto/>.
[18] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons, 1990.
[19] D. Kifer, J. Gehrke, Injecting utility into anonymized datasets, in: Proceedings of ACM International Conference on Management of Data (SIGMOD), 2006.
[20] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Incognito: Efficient full-domain *k*-anonymity, in: Proceedings of ACM International Conference on Management of Data (SIGMOD), 2005.
[21] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Mondrian multidimensional *k*-anonymity, in: Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE), Atlanta, GA, 2006.
[22] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Workload-aware anonymization, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.
[23] G. Loukides, J. Shao, Capturing data usefulness and privacy protection in *k*-anonymisation, in: Proceedings of the 2007 SAC, 2007.
[24] G. Loukides, J. Shao, Data utility and privacy protection trade-off in *k*-anonymization, in: Proceedings of the International Workshop on Privacy and Anonymity in the Information Society (PAIS), 2008.
[25] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, ℓ-diversity: privacy beyond *k*-anonymity, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1).
[26] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Atlanta, GA, 1967.
[27] B. Malin, *k*-unlinkability: a privacy protection model for distributed data, Data and Knowledge Engineering (DKE) 64 (1) (2008) 294–311.
[28] S. McClean, B. Scotney, P. Morrow, K. Greer, Knowledge discovery by probabilistic clustering of distributed databases, Data and Knowledge Engineering (DKE) 54 (2) (2005) 189–210.
[29] N. Mohammed, B.C.M. Fung, K. Wang, P.C.K. Hung, Privacy-preserving data mashup, in: Proceedings of the 12th International Conference on Extending Database Technology (EDBT), ACM Press, Saint-Petersburg, Russia, 2009.
[30] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
[31] President Information Technology Advisory Committee, Revolutionizing health care through information technology, Tech. rep., Executive Office of the President of the United States, June 2004.
[32] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[33] P. Samarati, Protecting respondents' identities in microdata release, IEEE Transactions on Knowledge and Data Engineering (TKDE) 13 (6) (2001) 1010–1027.
[34] P. Samarati, L. Sweeney, Generalizing data to provide anonymity when disclosing information, in: Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Principles of Database Systems (PODS), Seattle, WA, 1998.
[35] S. Savaresi, D.L. Boley, S. Bittanti, G. Gazzaniga, Cluster selection in divisive clustering algorithms, in: Second SIAM International Conference on Data Mining (SDM), 2002.
[36] S.M. Savaresi, D.L. Boley, On the performance of bisecting *k*-means and pddp, in: First SIAM International Conference on Data Mining (SDM), 2001.
[37] C.E. Shannon, A mathematical theory of communication, Bell System Technical Journal 27 (1948) 379–423 and 623–656.
[38] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: Proceedings of KDD Workshop on Text Mining, 2000.
[39] L. Sweeney, Achieving *k*-anonymity privacy protection using generalization and suppression, International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems 10 (5) (2002) 571–588.
[40] L. Sweeney, *k*-anonymity: a model for protecting privacy, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10 (5) (2002) 557–570.
[41] J. Vaidya, C. Clifton, Privacy-preserving *k*-means clustering over vertically partitioned data, in: Proc. of ACM SIGMOD International Conference on Management of Data, 2003.
[42] C.J. van Rijsbergen, Information Retrieval, second ed., Butterworths, London, 1979.
[43] K. Wang, B.C.M. Fung, Anonymizing sequential releases, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.
[44] K. Wang, B.C.M. Fung, G. Dong, Integrating private databases for data analysis, in: Proceedings of IEEE International Conference on Intelligence and Security Informatics (ISI), 2005.
[45] K. Wang, B.C.M. Fung, P.S. Yu, Template-based privacy preservation in classification problems, in: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), Houston, TX, 2005.

[46] K. Wang, B.C.M. Fung, P.S. Yu, Handicapping attacker's confidence: an alternative to *k*-anonymization, Knowledge and Information Systems: An International Journal (KAIS) 11 (3) (2007) 345–368.
[47] R.C.W. Wong, J. Li., A.W.C. Fu, K. Wang, (α,*k*)-anonymity: An enhanced *k*-anonymity model for privacy preserving data publishing, in: Proceedings of the 12th ACM SIGKDD, Philadelphia, PA, 2006.
[48] L. Xiong, K. Rangachari, Towards application-oriented data anonymization, in: Proceedings of the 4th SIAM Workshop on Practical Privacy-Preserving Data Mining, 2008.
[49] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, A.W.C. Fu, Utility-based anonymization using local recoding, in: Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2006.

Benjamin Fung is an Assistant Professor of the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Canada. He received the Ph.D. degree in Computing Science from Simon Fraser University, Canada. His current research interests include data mining, database, privacy preservation, information security, and digital forensics, as well as their interdisciplinary applications on current and emerging technologies. He has published in ACM Computing Surveys, ACM SIGKDD, IEEE TKDE, IEEE ICDE, IEEE ICDM, and EDBT. He serves as an editorial board member for IJDATS, and a program committee member for ACM SIGKDD, ACM CIKM, IEEE ICDM, and SDM. His research has been supported in part by the Discovery Grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Ke Wang is currently a professor at School of Computing Science, Simon Fraser University. His research interests include database technology, data mining and knowledge discovery, machine learning, and emerging applications, with recent interests focusing on the end use of data mining. This includes explicitly modeling the business goal and user utility (such as profit mining, bio-mining and web mining) and exploiting user prior knowledge (such as extracting unexpected patterns and actionable knowledge). His recent work includes privacy –preserving data mining and data publishing.

Lingyu Wang is an Assistant Professor of the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Canada. He received his Ph.D. degree in Information Technology from George Mason University, USA. His current research interests include database security, data privacy, vulnerability analysis, intrusion detection, and security metrics. His research has been supported in part by the Discovery Grants from the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Fonds de recherche sur la nature et les technologies (FQRNT).



Patrick C.K. Hung is an Associate Professor at the Faculty of Business and Information Technology in UOIT and an Adjunct Faculty Member at the Department of Electrical and Computer Engineering in University of Waterloo. Patrick has been collaborating with Boeing Phantom Works (Seattle, USA), Bell Canada, and Southeast University (Nanjing, China) on several research projects. Recently he is working on a mobile healthcare project with the Hong Kong Red Cross with the Chinese University of Hong Kong. He is an executive committee member of the IEEE Computer Society's Technical Steering Committee for Services Computing, a steering member of EDOC "Enterprise Computing," and an associate editor in several international journals such as the IEEE Transactions on Services Computing and International Journal of Web Services Research (JWSR). His research interests include Web services security, privacy, and business process integration.