The VLDB Journal (2006) 15:293–315
DOI 10.1007/s00778-006-0010-5

SPECIAL ISSUE PAPER

A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier-related transforms

Shibnath Mukherjee · Zhiyuan Chen · Aryya Gangopadhyay

Received: 30 September 2005 / Accepted: 25 May 2006 / Published online: 19 August 2006

Abstract Privacy preserving data mining has become increasingly popular because it allows sharing of privacy-sensitive data for analysis purposes. However, existing techniques such as random perturbation do not fare well for simple yet widely used and efficient Euclidean distance-based mining algorithms. Although original data distributions can be pretty accurately reconstructed from the perturbed data, distances between individual data points are not preserved, leading to poor accuracy for the distance-based mining methods. Besides, they do not generally focus on data reduction. Other studies on secure multi-party computation often concentrate on techniques useful to very specific mining algorithms and scenarios such that they require modification of the mining algorithms and are often difficult to generalize to other mining algorithms or scenarios. This paper proposes a novel generalized approach using the well-known energy compaction power of Fourier-related transforms to hide sensitive data values and to approximately preserve Euclidean distances in centralized and distributed scenarios to a great degree of accuracy. Three algorithms to select the most important transform coefficients are presented, one for a centralized database case, the second one for a horizontally partitioned, and the third one for a vertically partitioned database case. Experimental results demonstrate the effectiveness of the proposed approach.

Keywords Privacy · Data mining · Fourier transform

S. Mukherjee · Z. Chen (✉) · A. Gangopadhyay
Information Systems Department,
University of Maryland Baltimore County,
1000 Hilltop Circle, Baltimore, MD 21250, USA
e-mail: zhchen@umbc.edu

S. Mukherjee
e-mail: Shibm1@umbc.edu

A. Gangopadhyay
e-mail: gangopad@umbc.edu

1 Introduction

With the explosive growth of data and its ever increasing distributed sources across organizations, accurate, efficient, and fast analysis of the data for extraction of knowledge has become a major challenge. In many instances these factors force storage and analysis to be separated and a third party is involved with the responsibility of analyzing the data. In such instances two typical problems arise: first is that of sending the data to the third party since transmitting huge volumes of data imposes huge resource overheads and consumes time; second is the issue of privacy of the transmitted data which has tremendous importance to the business strategies of an organization and its practices regarding confidentiality of customers' private data.

There has been a rich body of work on the second issue of privacy- preserving mining. Depending on the type of data privacy problems being addressed, these studies, though somewhat related, can be divided into two generic categories: ones that try to hide the data values themselves when the data are sent to a third party for analysis [3,5,6,18,24,28,34,36,37,41] and ones that try to hide the identity of entities when publishing data [2,7,19,26,35]. This paper focuses on the first type of privacy problem.

Till date, existing literature on the first category, widely report studies that use random perturbation approaches

that add or multiply random noise to the data such that individual data values are distorted while the underlying distribution can be reconstructed with fair degree of accuracy [5,6,18,34,41]. A random projection-based approach is also suggested elsewhere [30]. The performance of the method is compared with the approach suggested in this paper. There is also work on using secure multi-party computation techniques for a wide range of data mining algorithms to address the privacy issue in distributed environment where data are heterogeneously or homogeneously partitioned across multiple parties [28,36,37]. They share intermediate mining results to calculate mining functions securely over multiple sources. In the process the information exchange overhead also gets reduced. However, each of these methods is generally suited to just one algorithm and/or scenario as will be illustrated in Sect. 2. There is thus a lack of attempt to have one single integrated method for at least even a collection of algorithms and scenarios. For the random perturbation-based algorithms, the original data distributions can be reconstructed [5] with some fair degree of accuracy, but mutual Euclidean distances between individual data points are not preserved. This, however, is the basis of many simple but efficient and widely used mining algorithms. The two most popular ones are K-means clustering [14] and K-nearest neighbor classification technique [11,14]. In fact privacy-preserving techniques for these algorithms were still being sought for as reported in [13].

To demonstrate the weaknesses of random perturbation methods, the example in Fig. 1a shows two randomly generated clusters of data with two attributes following two 2-D normal distributions. Figure 1b shows the same data but added with a random noise for each attribute following normal distribution with mean equals zero and standard deviation equals 0.25. Clearly, the Euclidean distance is not preserved in the perturbed data, and the two clusters in Fig. 1a no longer exist in Fig. 1b. Thus, the random perturbation algorithms are inappropriate for mining algorithms using Euclidean distance.

One more drawback of the perturbation algorithms is their inability to reduce dimensionality of data. Unlike perturbation, random projection does reduce dimensionality, but often distorts mutual Euclidean distances between data points as shown later in the experimental section of this paper. This paper develops a novel technique of using Fourier-related discrete orthogonal (unitary) transforms [31] to address both the problem of data reduction and privacy preservation for the entire set of Euclidean distance-based algorithms under different scenarios. The novelty of the approach lies in the fact that it does not depend on modifying the mining algorithms themselves but just prepares the data that can then directly be fed to these available algorithms. This spares users from using different major modifications of the basic classification and clustering algorithms for various scenarios and leads the way to an integrated approach.

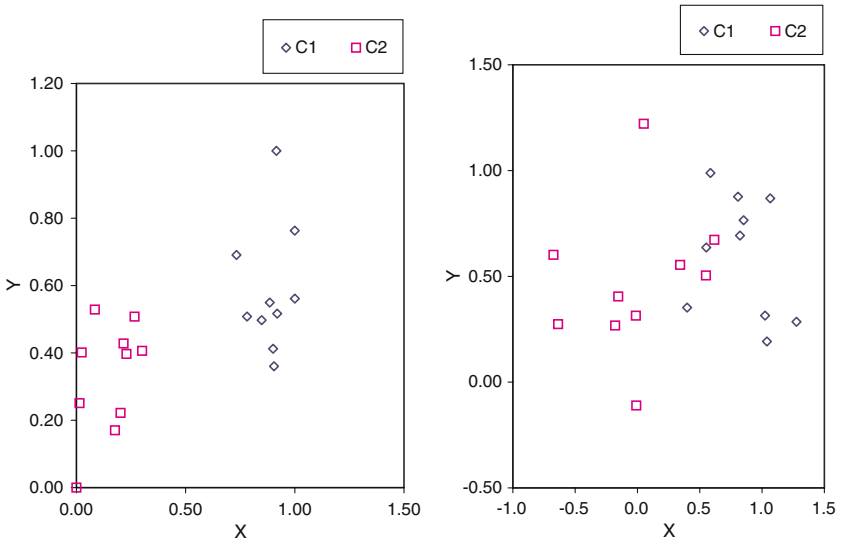
Discrete Fourier-related transforms such as Discrete Fourier Transform and Discrete Cosine Transform convert data in original domain to a transformed domain using a set of Fourier basis. Throughout this paper, each of the coefficients is associated with a corresponding index position. As an example, consider a data sequence as 5,000, 10,000, 50,000 and its corresponding DCT coefficients 37,528, –31,820, and 14,289. In this case the coefficient 37,528 is called the coefficient with index 1, i.e., the first coefficient and so on.

The algorithms in the paper are based on the following three properties of Fourier-related transforms.

1. First, they preserve Euclidean distance between data values in the transformed domain. Thus, one can apply Fourier-related transforms for each row such that the Euclidean distances between any pair of rows is preserved. This serves the purpose of using the transformed data for Euclidean-distance-based-data mining.
2. Second, they can be used as a lossy compression technique by suppressing small coefficients and keeping the large ones, as in the domain of image processing [38] and many others including band limited transmission of communication signals [32].
3. Third, due to the suppression of coefficients, the exact values of original data cannot be reconstructed from transformed data. This can be perfectly blended with a permutation protocol as will be described in details later to render enhanced privacy of data values.

One might argue that this paper focuses only on Euclidean distance. However, there exist algorithms as presented in [39] that can accurately map arbitrary distance functions to Euclidean distance domain as a baseline to simplify complicated distance function calculations. Given a set of objects and a distance function that could be non-Euclidean, these algorithms map these objects into a Euclidean space such that the distances between these objects are approximately preserved. Experimental results in [39] show that for several data sets, these mapping algorithms preserve distances to a high degree such that the mapping introduces small error (less than 10%) for clustering. Thus, the algorithms presented in this paper can potentially be generalized to

Fig. 1 A clustering example



(a) Original data **(b)** Perturbed data

any distance functions. The contributions of this paper are summarized as follows:

- This paper proposes a solution to a centralized database case where coefficient information about a data set is transmitted to a third party mining engine. The proposed greedy heuristics select a set of few coefficients retaining high energy across a vast majority of rows. Consequently-the Euclidean distances between the data points in the transformed data are preserved to a great degree of accuracy. In the process, data volume that needs to be sent is also reduced due to the fact that just a few coefficients are sent. Finally privacy of raw data is preserved since it is difficult for the third party to reconstruct the data from a few coefficients and that too with the number of attributes and the index correspondence information unknown due to a secure permutation order sharing protocol which will be discussed in Sect. 3.3.
- The paper also proposes a solution to a horizontally partitioned database case where each data source locally transforms their data and transmits the specific coefficient information to a third party. A similar algorithm is proposed for the vertically partitioned scenario using the linearity property [31] of Fourier-related transformations.
- Finally the paper conducts extensive experimental evaluation to compare the proposed methods with existing methods for two most popular Euclidean-distance-based mining algorithms: K-means clustering and K-nearest neighbor classification. The results demonstrate the superiority of the proposed methods.

This paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed solution to centralized database case, the horizontally partitioned database case, and the vertically partitioned database case, respectively. Section 4 presents experimental evaluation and Sect. 5 concludes the paper.

2 Related work

As discussed in the previous section, there has been quite some work on privacy-preserving mining of data [3,5,6,9,18,20,23,24,28,34,36,37,41]. A more complete set of references in the field is available elsewhere [10]. Two most relevant branches in the context are random perturbation methods and the secure multiparty computation algorithms. However, as mentioned in Sect. 1, perturbation methods are not appropriate for Euclidean distance- based mining which encompasses quite a few simple, efficient, and popular algorithms sharing a common theme of using Euclidean distance as the similarity measure between data points. Further they generally do not reduce the data size. The secure multiparty computation algorithms do reduce the amount of information being sent, by sharing intermediate results of mining algorithms [28,36,37] or using cryptography [15,27]. However, intermediate results are specific to the mining algorithms being used, hooking up these methods, in general to specific algorithms and/or scenarios. For example, in [28] the shared information is models

of clusters, in [36] the shared information is distances of each point to the cluster centroids, in [23] the shared information is Bayesian learning models in context of Naive Bayesian learning, and in [20] it is the binary vectors used in their decision tree learning algorithm. The problem with these algorithms is that, they are not flexible or generalizable for even a set of mining algorithms sharing a common theme (e.g., an algorithm for K-means clustering may not be directly used for K-nearest neighbor classification, although both use Euclidean distances as the similarity measure). The utility and need of such an integrated approach are stated in [13] from the perspective of industry usefulness of privacy-preserving data-mining algorithms.

A condensation approach proposed in [1], which also does not require modification of mining algorithms, and thus, is general. This approach first generates size-k clusters and then regenerates data based on properties of these clusters. The approach primarily focuses on preserving data correlations rather than Euclidean distances considered in this paper. Data reduction or multiparty computations are also not considered. Further, unlike the work presented here, the condensation approach is explicitly more concerned with hiding the identities of entities. Disclosure protection of the original data values is not good enough because the regenerated data values are very close to the original ones.

A privacy-preserving technique for clustering is proposed in [29]. This technique transforms data by several types of geometric transforms such as rotation, translation, and scaling that preserve Euclidean distance. However, since the transform is the same for all records, if third party can discover the original values of one data record, all original data records can be fully reconstructed. Two more methods are proposed in [30]. One method transmits the similarity between any pair of records instead of the data. However, this method is not scalable because the number of similarity pairs is quadratic to the total number of records. The other method uses random-projection to project original data of m dimensions to smaller number of k dimensions. It is observed that Euclidean distance is often distorted to a great extent using this method when k is small. The performance of the method will be compared against the one suggested in this paper in the experimental section. Blum et al. [8] proposed a technique to add random noise to the output of database queries to preserve privacy and support K-means algorithm. However, this paper considers the case of shipping data to a third party, thus noise has to be added to each record in the database instead of the results of a database query.

Fourier-related transforms have long been used in many areas such as physics [22] and image processing [38]. An offshoot of it, the Fourier series has also been used in [25] to give a compressed representation of decision trees for distributed mining. These transforms have also been used for similarity search in sequence databases [4]. In most of the applications, DFT/DCT is generally identified with time series analysis for the fact that they give the frequency spectrum of a time series signals and help identify its characteristics. However, in this paper it is used as a low complexity, discrete orthogonal (unitary) transform that transforms discrete data from the original domain to a different domain yet preserving Euclidean distance between data points and concentrating energy in few coefficients quite efficiently. Egecioglu et al. [16] show the general effectiveness of these category of transforms in compacting energy of a sequence and maintaining accurate approximations of Euclidean distances with few high energy coefficients. Fourier transforms have also been used in the context of privacy-preserving data mining by Wu [40] to perform estimation for randomized algorithms. However, the focus in context is on preserving Euclidean distances and its trade-off with privacy of data. To the best of knowledge gathered from literature, the features of Fourier-related transforms, described above, have not been fully explored till date in the context of privacy-preserving data mining.

3 Proposed approach

This section presents the solution. Section 3.1 briefly reviews Fourier-related transforms. Section 3.2 gives an overview of proposed approach. Section 3.3 frames the problem of selecting high energy coefficients. The solutions to centralized database case, horizontally partitioned case, and vertically partitioned case are presented in Sects. 3.4, 3.5, and 3.6, respectively.

3.1 Fourier-related transforms

Fourier-related transforms are a class of unitary transforms that convert data from the original domain to a transformed domain, keeping the energy same in both domains. This paper considers two such transforms: Discrete Fourier Transform and Discrete Cosine Transform. Consider a series of complex numbers $x_0, x_1, \ldots, x_{n-1}$, DFT generates a set of complex coefficients $f_0, \ldots, f_{n-1}$ such that

$$f_i = \frac{1}{n} \sum_{k=0}^{n-1} x_k e^{-jk2\pi i/n}$$

where j is square root of –1.

DCT works on real numbers and generates the following real coefficients:

$$f_i = \left(\frac{2}{n}\right)^{1/2} \sum_{k=0}^{n-1} \Lambda_k x_k \cos[(2k+1)i\pi/2n]$$

Where $\Lambda_k = \frac{1}{\sqrt{2}}$ for $k = 0$ and 1 otherwise. The energy of a series $x_0, \ldots, x_{n-1}$ is defined as $\frac{1}{n}\sum_{k=0}^{n-1} x_i^2$.

Parseval's theorem [31] states that for these transforms energy is preserved, i.e., the energy of $x_0, \ldots, x_{n-1}$ equals the energy of $f_0, \ldots, f_{n-1}$. These transforms are unitary transforms and Euclidean distance between two sequences is preserved in the transformed domain. A simple proof can be found in [4]. Two important lemmas serving as the backbone of the algorithms are presented as follows.

Lemma 1 *Given a set S of coefficients chosen out of the universal set $U = S \cup \bar{S}$ of all coefficients, the expected error of squared Euclidean distance between any two transformed records due to pruning elements in $\bar{S}$ is bounded on the upper and lower ends by the square of maximum and minimum Euclidean distances existing between any two records in the dataset calculated over the set of coefficients in $\bar{S}$, respectively.*

Proof Let $SE_{X,Y}$ be the error of squared Euclidean distance when distance is calculated over coefficient set S for any transformed pair of records X and Y. Thus

$$E(SE_{X,Y}) = E\left[\sum_{i\in U}(X_i - Y_i)^2 - \sum_{i\in S}(X_i - Y_i)^2\right]$$
$$= E\left[\sum_{i\in \bar{S}}(X_i - Y_i)^2\right]$$

This immediately implies

$$arg \min_{X',Y'}\left\{\sum_{i\in\bar{S}}(X_i' - Y_i')^2\right\} \le E(SE_{X,Y})$$
$$\le arg \max_{X',Y'}\left\{\sum_{i\in\bar{S}}(X_i' - Y_i')^2\right\}$$

Where X' and Y' are any two arbitrary pair of data vectors from the dataset. This proves the lemma. □

As an immediate consequence of the above result, it is apparent that if most coefficients in $\bar{S}$ have small values, the lower bound tends to zero while the upper bound is also small. It is also obvious that as $|\bar{S}|$ becomes smaller, the expected error tends to zero.

Lemma 2 *Given a distance margin ϵ to choose a nearest neighbor pool for a data vector y, all members of the pool will be members of a corresponding pool of nearest neighbors of Y (the vector transformed from y) built with the same margin ϵ after retaining only few coefficients.*

Proof The proof follows the same principle as that of Lemma 1 in [4]. Let y be a vector whose nearest neighbor pool is P . Now if S is a set of selected coefficients after pruning some low energy coefficients and $|S| \le n$, then using Parseval's theorem [31]

$$\epsilon \ge \sum_{i=1}^{n}(x_i - y_i)^2 = \sum_{i=1}^{n}(X_i - Y_i)^2$$
$$\ge \sum_{i\in S}(X_i - Y_i)^2 \quad \forall x \in P$$

This proves the lemma. □

If k nearest neighbors are selected instead of fixing ϵ for generating the pool P ($|P| = k$), then ϵ is replaced by the maximal distance to these k-nearest neighbors, i.e., $arg\ \max_{x_i\in P} ||x_i - y||$ in the derived equations. This basically gives the upper limit of distance between vectors in the transformed domain after pruning the coefficients. This lemma guarantees no false negatives in nearest neighbor pooling for the methods presented in the paper.

3.2 Overview of solution

This paper considers three scenarios: a centralized database scenario when the information about a whole database is sent to a third party for analysis, a horizontally partitioned and a vertically partitioned, database scenario.

The approach proposed in the paper consists of two steps. First, each record in the database is seen as a sequence and is converted to the transformed domain using one of the Fourier-related transforms. Since in practice different attributes may have very different ranges, normalizing each attribute to a value in the range of [0,1] before transformation is required. Second, a few coefficients that appear as high energy coefficients in a vast majority of the rows are selected and sent to the third party. In the centralized case, this selection is conducted by the data source. In the horizontally and vertically partitioned case, the selection is conducted by both data sources and the third party. This helps preserve Euclidean distances to a great degree of accuracy. A permutation protocol, discussed later, is also blended in the exchange, obfuscating the correspondence of coefficients and their indexes and enhancing the privacy of the scheme. In the process, the size of converted data is also reduced due to the suppression of coefficients. Suppose there are m attributes in the original data and

μ coefficients are selected. The size of the converted data is $\frac{\mu}{m}$ fraction of the size of the original data if we assume each attribute has the same size. The concept comes from the second property of Fourier transforms mentioned in Sect. 1, that is, the fact that generally, in the transformed domain, most of the energy of a sequence is concentrated in a few coefficients rather than being spread out across the sequence as in original domain. This suits the need for data reduction as well as accurate approximation of Euclidean distances and inner products. Egecioglu et al. [16] report its successful use for approximating Euclidean distances and inner products. The principle has also been used in Agrawal et al. [4] for similarity search in sequence databases and has been the motivation for the approach presented in this work.

In real life datasets, generally, some of the high energy coefficients in one row will have low energy in some other rows. However, it is observed that in almost all real life cases, the tendency of Fourier-related transforms is to concentrate energy in a small set of common coefficients across a large majority of rows of the data. Section 4.6 gives a metric for measuring how effective the transform is in achieving the above goal for a particular dataset. As will be seen, the experiments prove the merit of DCT in most of the real life datasets. The paper explores that finding the minimal combination of common high energy coefficients to preserve a given fraction or more of energy over all rows is an NP-hard problem and proposes three efficient rank-based greedy heuristics to solve the problem approximately in three different scenarios.

3.3 A permutation protocol enhancing privacy

Following the third property of Fourier transforms stated in Sect. 1, since some coefficients are pruned by the heuristics for data reduction, the third party automatically cannot reconstruct the original data values exactly in all cases. In the centralized database case, privacy can be further enhanced to a great degree by random permutation of coefficient indexes and not divulging the number of attributes to the third party. Without knowing the number of attributes and the order of coefficients, it is extremely difficult for the third party to reconstruct the original data. For Example, suppose a malicious third party assumes a maximal number of attributes of N, and the number of selected coefficients is μ, then it needs to check $\sum_{i=\mu}^{N} \frac{i!}{(i-\mu)!}$ possible permutations. Let $N = 30, \mu = 10$, he needs to check about 3×10^{14} permutations!

The third party may want to get rid of some combinations when the bounds of some attributes are known to him. Some of the combinations will give out of bound values and can be eliminated immediately. However, this approach is problematic. Taking an example, suppose the third party knows the DCT coefficients for sequence 5,000, 10,000, 50,000 but does not know the correspondence between coefficients and their indexes. Assume the third party reconstructs data using coefficients in the order of coefficient 1, 3, and 2. The reconstructed data is 18780, 47647, and –1427. Suppose the numbers are salary fields of three employees. The third party knows salary cannot be negative, thus, he can reject this permutation. However, suppose the third coefficient is pruned because it is the smallest, the reconstructed data using the correct permutation of the first two coefficient will be –833, 21667, and 44167, which also contains a negative number.

In essence, the success of this permutation filtering approach will entirely depend on the accuracy of information that the third party has on the bounds and the magnitude of distortion in data values that the coefficient pruning has rendered. It also depends on the number of attributes whose bounds are known. If r such combinations that can be eliminated then the third party has to evaluate $\sum_{i=\mu}^{N} \frac{i!}{(i-\mu)!} - r$ combinations. However, this process is still extremely expensive as the third party cannot eliminate a combination till it actually inverts it and matches the attributes to its known bounds. Further, as long as there are multiple permutations that satisfy the known bounds, it will be very difficult for the third party to actually figure out the correct permutation.

In the horizontally and vertically partitioned cases, if the third party is the sole coordinator, then all sources must send coefficients with the same indexes as requested by the server. Thus, the third party has to know the correspondence between indexes and coefficients and can potentially use this information to approximately reconstruct the original data via inverse of the transform being used. This situation can be made more secure and very much like the one in centralized case by introducing a minor cryptographic tweak as following. The data-sharing parties agree on a known, random permutation order which is not known to the server and send coefficients permuted in that order to the server. The number of permutations necessary for the server to evaluate the right one is again the same as the centralized case. In addition, it should be mentioned that in the vertically partitioned case, the number of attributes will be completely known information to the third party as opposed to the centralized and horizontally partitioned case with the permutation protocol. This is because the zero padding instructions are to be coordinated by the server as will be explained later. This slightly increases the chance of breaking the permutation protocol

described above when compared to the other two cases. However, even a moderate number of attributes will give enough permutations to make the discovery process prohibitively expensive as can be justifiably observed from the nature of the problem.

The protocol can be securely implemented using a public-key infrastructure, where every data source registers a public key and has its own private key. One of the data sources is selected as the coordinator and he selects a random permutation of coefficient indexes. He then sends the permutation to the other data sources encrypted in those sources' public key. The other sources will decrypt the permutation using their own private key. Note that the encrypted information is not encrypted using the third party's public key, and is never sent to the third party. Thus the third party cannot decrypt the permutation.

A certain degree of privacy is still preserved in the worst case, if somehow the permutation is discovered. This is due to the suppression of coefficients. In practice users can decide the tradeoff of keeping more coefficients for better mining quality and keeping fewer coefficients for better privacy. Section 4 will present experimental results that demonstrate that the approach preserves privacy to a high degree in the centralized case and the distributed cases with the permutation protocol. Privacy is also maintained to an appreciable degree in the worst case of horizontally and vertically partitioned scenario when the protocol breaks.

3.4 Selection of high energy coefficients

Let m be the number of attributes, n be the number of data records, and ζ be a value between (0,1) given by the user. Let X_i be a binary variable taking values 1 if a coefficient i is selected and 0 otherwise. Let W_{ij} be the fraction of energy of record j that coefficient i stores. The problem of selecting high energy coefficients for a single database case can be formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{m} X_i$$

$$\text{Subject to} \sum_{i=1}^{m} W_{ij} X_i \geq \zeta \qquad \forall j, 1 \leq j \leq n$$

This formulation attempts to minimize the number of coefficients selected making sure that in all rows the selected coefficients together preserve at least a certain fraction of energy. However, this formulation has the drawback that if there are a large number of rows and comparatively small number of attributes, the solution might give all $X_i = 1$ for a high ζ. Further being an integer linear program, it is NP-hard. Thus, this paper focuses on more feasible alternatives described below.

Assuming the data source has selected to transmit μ coefficients. In the case of centralized database, the problem is to select μ coefficients that maximize the number of records that preserve at least a certain fraction of energy. The formal definition is as follows.

Problem 1 Let μ be an integer given by the user such that $0 < \mu < m$. Let ζ be a value between (0,1) given by the user. Let X_i be a binary variable taking values 1 if a coefficient i is selected and 0 otherwise. Let W_{ij} be the fraction of energy of record j that coefficient i stores. Let R be a binary function over each record j such that

$$R(j) = 1 \; if \sum_{i=1}^{m} W_{ij} X_i \geq \zeta$$
$$= 0 \text{ otherwise}$$

The problem is to

$$\text{Maximize} \quad \sum_{j=1}^{n} R(j)$$

$$\text{Subject to} \quad \sum_{i=1}^{m} X_i = \mu.$$

In the definition, function R is used to identify whether the energy of a record has been sufficiently preserved, and ζ is the energy threshold.

A straightforward generalization of this definition to horizontally or vertically partitioned case is to maximize the total number of records across all sources whose energy has been sufficiently preserved. However, this is problematic because this may leave the energy of records in some partitions not well preserved and have a negative impact on mining. For example, suppose there are two classes C_1 and C_2, and two horizontal partitions D_1 and D_2. D_1 only contains records in C_1, D_2 only contains records in C_2, and D_1 contains far fewer records than D_2. If we maximize the total number of records whose energy is sufficiently preserved, the energy of records in D_1 may not be well preserved because there are far more records in D_2 and coefficients may be selected to keep energy of records in D_2 only. This may have a negative impact on mining because all records of C_1 are in D_1 and the characteristics of C_1 may not be well preserved.

Thus, it is important to keep a certain fraction of records for each partition for better mining results. Therefore, the goal in horizontally partitioned cases is to select μ coefficients that maximize the number of partitions whose energy has been sufficiently preserved. The energy of a partition is sufficiently preserved if for at

least ζ' fraction of records in that partition, energy is preserved by a fraction greater than or equal to ζ. Here ζ' is a parameter given by users. A formal definition is as follows.

Problem 2 Let $D, \mu, X_i, W_{ij}, \zeta$, and R be defined in the same way as in Problem 1. Further let r_j denote record j and ζ' be a value between (0,1) given by user. Let $D_1, D_2, \ldots, D_k$ be horizontal partitions of D. Let R' be a binary function defined over $D_1, \ldots, D_k$ such that

$$R'(l) = \begin{cases} 1 & \text{if } \sum_{r_j \in D_l} \frac{R(j)}{|D_l|} \geq \zeta' \\ 0 & \text{otherwise} \end{cases}$$

The problem is to

$$\text{Maximize} \sum_{l=1}^{k} R'(l)$$

$$\text{Subject to} \sum_{i=1}^{m} X_i = \mu$$

Here function R' is used to specify whether a partition's energy has been sufficiently preserved. These two problems can also be proved to be NP complete by reducing them from the set cover problem. The details of the proof are omitted. A naive solution to both Problem 1 and 2 is to examine all set of μ coefficients and count the number of records and/or partitions that exceed the energy threshold. The complexity of the naive algorithm is thus $O(\binom{m}{\mu}\ mn)$. For a large μ the naive algorithm is quite expensive. The next two sections will present approximate but efficient rank-based greedy heuristics for solving these two problems.

3.5 Solution to centralized database case

This section presents an algorithm for the centralized database case. The algorithm is based on the following observations:

1. For most real life datasets, the energy of each transformed record is represented by very few coefficients [16].
2. Although the high energy coefficients in one transformed record may have low energy in some others, for most real life data, on an average, energy tends to concentrate in a small set of transform coefficients common across a vast majority of rows.

The objective of the algorithm is to search a set of coefficients appearing as high energy coefficients across a large number of transformed records. The pseudocode is shown in Fig. 2.

The algorithm starts with taking the DFT/DCT for each record in the database D considering them as sequences of numbers in line 1–3. In line 4–6, for each record, the algorithm selects Δ coefficients with highest energy and stores their indexes in a $n \times \Delta$ matrix DL. DL will be called the high energy coefficient index matrix. Next the algorithm counts for each coefficient with index j, the number of rows in DL that contains it, in line 7–9. Once these frequencies are counted, the μ coefficients with highest frequencies will be sent to the third party for analysis. The order of these coefficients is also permuted randomly to prevent the third party from reconstructing the data even approximately from the small set of coefficients.

Note that for classification the class variable column is left as it is while the process is applied on the other attributes and the class variable column is sent to the third party along with the selected coefficients. The following example illustrates this algorithm. Figure 3a shows the original data D. Figure 3b shows the DCT coefficients. Figure 3c shows the index of high energy coefficients assuming $\Delta = 4$ (matrix DL in Algorithm 1). For example, in the first record, the four coefficients with the highest energy are coefficients 1, 6, 2, and 5. Figure 3d shows the frequencies of high energy coefficients. Assuming $\mu = 3$, the first, second, and fifth transform coefficients will be chosen. Next the indexes of these coefficients will be permuted randomly. Suppose the resulted order is second, fifth, and first, then these coefficients for each record will be sent to the third party in this order.

Let m be the number of attributes and n be the number of records. The complexity of DCT and DFT transform is $O(mn \log m)$ [31]. Quick-sort can be used to sort the coefficients of each record in descending order of energy and to select the Δ coefficients with highest energy. Thus, the complexity of generating high energy coefficient matrix DL is $O(mn \log m)$. The computation of the frequencies in DL takes $O(n\Delta)$. Thus, the overall complexity of the algorithm is $O(mn \log m)$. Note that the naive algorithm takes $O(\binom{m}{\mu} mn)$ and $\log m << \binom{m}{\mu}$, so the heuristic algorithm is far more cost efficient.

3.6 Solution to horizontally partitioned database case

The algorithm for horizontally partitioned case is shown in Fig. 4.

The algorithm for a classification problem is described in Fig. 5.

The objective of the proposed algorithm is the same as Algorithm 1 in previous section. At line 8, each data source sends these requested coefficients after permuting them in a predetermined order known to each data source but not the server. This is in accordance with the

```
Algorithm 1: Solution-Centralized(D, μ)
(1) for each record i of D
(2)  DI(i) = Transform( D(record i) )
(3) end
(4) for each row i of DI
(5)  select Δ coefficients with highest energy and store their indexes in row i of a n × Δ matrix DL
(6) end
(7) for each coefficient with index j
(8)  frq(j) = number of rows in DL that contains j
(9) end
(10) select coefficients corresponding to μ highest values in frq list and permute them in a random order
(11) send the permuted selected coefficients for each row to the third party
```

Fig. 2 Algorithm for centralized database case

Fig. 3 Example of running Algorithm 1

30	1.2	20.5	5	8.9	3
45	3.2	12	4	7	2
28	0.9	18.6	5	10	2
35	2.3	23.4	3	9.5	1
12	5	3.4	1	2.5	5

(a) Data

28.0	14.2	3.75	7.83	11.06	15.82
29.88	23.62	15.50	15.84	12.3	12.44
26.33	12.82	3.2	8.78	9.18	15.18
30.29	19.07	4.8	8.5	11.2	19.4
11.8	5.28	6.3	0.86	1.85	1.36

(b) DCT coefficients

```
1 6 2 5
1 2 4 3
1 6 2 5
1 6 2 5
1 3 2 5
```

(c) High energy coefficients

```
frq(1) = 5
frq(2) = 5
frq(3) = 2
frq(4) = 1
frq(5) = 4
frq(6) = 3
```

(d) Frequencies of high energy coefficients

```
Algorithm 2: Solution-Horizontally-Partitioned(D, μ, Ψ, Δ)
/* D = {D_1, ..., D_k} is a set of horizontal partitions of data,
 μ is the number of coefficients to select */
(1) for each data source D_i
(2)  run step 1-6 in Algorithm 1 over its local partition to generate high energy
 coefficient index matrix DL_i, given a common Δ suggested by third party.
(3)  count the frequency of each coefficient index in DL_i, sort them,
 and send indexes of Ψ top most ones to third party.
(4) end
(5) third party assembles the index information sent from each D_i
 with each record as a sorted index list of top Ψ frequent elements from each D_i. Call it DF.
(6) third party computes the frequency of each index (number of rows in DF
 having the coefficient as candidates) from the assembled index information.
(7) third party selects coefficients whose indexes have μ top most frequencies in step 6 and request
 them from each D_i.
(8) each D_i sends the requested coefficients after following the permutation order
 agreed upon between all D_is.
```

Fig. 4 Algorithm for horizontally partitioned case.

permutation protocol described in Sect. 3.3. At lines 2 and 3, each source selects Δ coefficients with the highest energy for each record in its local partition and generates its own high energy coefficient index matrix DL_i. Each source then counts the frequency of each coefficient index in DL_i, sorts the list, and sends indexes of coefficients having top Ψ frequencies to the third party. Ψ is selected by the third party and communicated to the data sources prior to local calculations. The third party forms a $k \times \Psi$ matrix DF where each row stores the sorted index list for a partition in line 5.

The selection of the μ coefficients is conducted at the third party. The heuristic tries to select the candidate coefficients that appear in the largest number of sources from the matrix DF. After this, these μ coefficients are requested from each data source. Finally, every data source will send the permuted coefficients to the server. The server will assemble the received coefficients.

Note that the frequency of a coefficient j in DL_i equals the number of records at source i where coefficient j is among the Δ highest energy coefficients. Thus, selecting a coefficient with high frequency means it is more likely that the energy of many records in that source will be preserved. Note that Problem 2 requires selecting coefficients that maximize the number of sources whose energy is sufficiently preserved, and each candidate coefficient has a high chance to preserve sufficient energy at each source.

Algorithm 3: Solution-Vertically-Partitioned(D, μ, Ψ, Δ)
(1) Each site sends the number of attributes of each source
as well as which source has the class variable information to server.
(2) Server indexes the data sources as $D_1, D_2, \ldots, D_k$. Without loss of generality,
we assume D_k has the class variable in question. Let $A(D_i)$ represent the number of attributes in source D_i.
(3) Server instructs the following to the sources based on the information and the virtual indexing:
For every source D_p where $p \neq 1$ or k,
Add $\sum_{i=p+1}^{k-1} A(D_i) + A(D_k) - 1$ zeros as padding to the right of each of the rows
Add $\sum_{i=1}^{p-1} A(D_i)$ zeros as padding to the left of each of the rows.
For $p = 1$ Add $\sum_{i=2}^{k-1} A(D_i) + A(D_k) - 1$ zeros as padding to the right of each row
For $p = k$ Add $\sum_{i=1}^{k-1} A(D_i)$ zeros as padding to the left of each row.
Clearly, let r_{ij} be the i-th row in the j-th source, and r'_{ij} be the same row after zero padding,
then $\sum_{j=1}^{k} r'_{ij}$ equals the vertical concatenation of all $r_{ij}, 1 \leq j \leq k$
(4) Each site takes row-wise DCT of their zero padded data.
The site having the class variable takes DCT of its rows excluding the class variable column.
(5) Run Algorithm 2 to select μ coefficients from each partition.
(6) Once the asked for μ coefficients are sent to the server after permutation in the predetermined order,
the server adds them up and places the corresponding class values
sent by the class value holder source as input to mining algorithms.

Fig. 5 Algorithm for vertically partitioned case

Let k be the number of sources. As shown in Algorithm 1, all sources take $O(nm \log m)$ to compute the high energy coefficient matrix DL_i. The total information sent from sources to the third party at line 3 is $O(\Psi k)$. The complexity of selecting the μ coefficients at third party is $O(km \log m)$ if quick sort is used to select candidate coefficients. Thus, the complexity of Algorithm 2 is still $O(nm \log m)$. The transmission cost is $O(\mu n + \Psi k)$ because μ coefficients and n rows in total will be transmitted from all sources.

3.7 Solution to vertically partitioned database case

This section describes yet another extension of the algorithm in the context of clustering and classification of vertically partitioned databases. The scheme is based on the linearity property of Fourier- related transforms that is, given two sequences X and Y,

$$\text{DCT/DFT}(X + Y) = \text{DCT/DFT}(X) + \text{DCT/DFT}(Y)$$

For clustering there is no need to send any class variables as is obvious and all sources will consider all their attributes before proceeding in the above fashion.

Below is a small example of a dataset with 3 partitions.

D_1

2	1.2	3.4	23
1.1	8	2.3	5.6
2.6	4.5	1	6.7
1.2	6.7	2.5	8

D_2

5	4.5	6.3
9	23	12
2.3	1.9	2
5.5	6	7.2

D_3

6.7	2	4.5	1
1.2	3	4	2
5.6	1.2	8	1
1.3	7.8	3	1

The last column in D_3 gives the class labels which is 1 and 2 in this specific case. After the sites perform the 0 padding as instructed in line 3 above, the sites will have

Padded D_1

2	1.2	3.4	23	0	0	0	0	0	0
1.1	8	2.3	5.6	0	0	0	0	0	0
2.6	4.5	1	6.7	0	0	0	0	0	0
1.2	6.7	2.5	8	0	0	0	0	0	0

Padded D_2

0	0	0	0	5	4.5	6.3	0	0	0
0	0	0	0	9	23	12	0	0	0
0	0	0	0	2.3	1.9	2	0	0	0
0	0	0	0	5.5	6	7.2	0	0	0

Padded D_3

0	0	0	0	0	0	0	6.7	2	4.5	1
0	0	0	0	0	0	0	1.2	3	4	2
0	0	0	0	0	0	0	5.6	1.2	8	1
0	0	0	0	0	0	0	1.3	7.8	3	1

Next DCT is applied on each partition and Algorithm 2 is used to select high energy coefficients. Suppose in the above example the coefficients 1 and 4 are selected, the server requests this information from the sites along with a special request for the entire column of the class variable to D_3. The server then adds up the coefficients, and together with the class variable value information frames a dataset which can be fed directly to any Euclidean distance-based mining algorithm.

4 Experimental evaluation

This section presents experimental evaluation of the proposed methods against existing methods. The major findings are summarized as follows.

- Methods presented in the paper using Fourier-related transforms preserve Euclidean distances to a great extent. The quality of K-means clustering and K-nearest neighbor classification (KNN) using these methods is significantly better than existing random perturbation methods and random projection method for the data sets tested.
- The heuristics proposed in Sect. 3 to select coefficients are highly effective. The quality of K-means and KNN over data generated using the proposed heuristics is significantly better than the quality of results over data generated by random selection of coefficients, and is very close to the benchmark case: mining the original data.
- The proposed Fourier-related methods reduce the data size significantly, yet maintain the accuracy of the algorithms to a great extent.
- The proposed methods achieve high degree of privacy in all the cases when the third party does not know the number of attributes and the correspondence of indexes of coefficients transmitted to the third party. Privacy is also achieved to an appreciable degree in the worst case when the third party does know the number of attributes and the indexes of coefficients.

Section 4.1 describes the setup. The results for centralized case are presented in Sect. 4.2. Sections 4.3 and 4.4 present the results for horizontally partitioned case and for vertically partitioned case, respectively. Section 4.5 reports the worst case privacy. Section 4.6 provides a guideline to choose the number of coefficients. Section 4.7 reports the overhead of proposed methods.

4.1 Setup

The experiments were conducted on a machine with Pentium 4, 3.4 GHz CPU, 4.0 GB of RAM, and running Windows XP Professional. All algorithms were implemented using Matlab 7.0. Since many algorithms (e.g., K-means and random perturbation methods) use randomization, the reported results of each algorithm are the average of five executions. Further the synthetic datasets were generated five times to empirically test the performance of the methods.

Datasets The experiments were run over two real datasets and one synthetic dataset. The two real datasets were Iris and Pendigits, both obtained from UCI Machine Learning Repository [21]. These two data sets contain numerical attributes with various distributions. E.g., the 14th attribute of Pendigits data has very skewed distribution, while the 13th attribute has more uniform distribution. The synthetic data was generated using a program from [12]. It contained ten clusters, each generated using a multi-dimensional Normal distribution. Table 1 reports the properties of these data sets. All attributes were numerical. For classification, 20% of data was randomly selected as the testing data, and the rest was used as training data. Each attribute was normalized to a value in the range of [0, 1].

Data mining algorithms K-means clustering and k-nearest neighbor classification (KNN) were used in experiments. k was set to 5 in KNN. Both algorithms use Euclidean distance.

Privacy-preserving algorithms A DCT-H algorithm was implemented using Discrete Cosine Transform based on algorithms proposed in Sect. 3. Though DFT could also be used, DCT was selected because the DCT coefficients are real numbers, thus data mining algorithms can be used without modifications. Further using DCT eliminates the possibility of the third party concluding anything about the coefficient indexes from the symmetry property that DFT offers.

The parameter Δ in Algorithms 1 and 2 was set to $\mu+1$ (μ is the number of coefficients transmitted) for DCT-H because it was found in experiments that the quality of data mining was quite good once Δ was slightly larger than μ, and only improved slightly with increase of Δ. Similarly, the parameter Ψ (number of candidates) in Algorithm 2 is set to $\mu + 1$ for horizontally and vertically partitioned case.

The following four algorithms were also implemented and compared against DCT-H:

- DCT-R It is the same as DCT-H except that coefficients were selected randomly. This is the baseline to test the effectiveness of the heuristic algorithms to select coefficients.
- Rand-N This algorithm adds to original data a random noise following Gaussian distribution with mean = 0. The standard deviation was varied in the experiments to generate different degree of privacy.

Table 1 Properties of datasets

	Iris	Pendigits	Synthetic
Number of attributes	4	16	100
Number of records	150	7,494	10,000
Number of classes	3	10	10

- Rand-U This algorithm adds to original data a random noise following uniform distribution with mean equals 0. The interval of the uniform distribution was varied to generate different degree of privacy.
- Rand-P This algorithm was proposed in Oliveira and Zaiane [30]. It maps the original m-dimensional data points to k-dimensional data points using random projection. The mapping is achieved by generating a $m \times k$ normalized random matrix with values chosen randomly from a given distribution and then multiplying the original data set (as an $n \times m$ matrix where n is number of records) by this matrix. $k < m$ brings data dimension reduction.

Setup for horizontally partitioned case The number of sites (partitions) was varied from 2 to 5 in experiments. The data records were distributed among these sites in two different ways: (1) data records with different class labels were uniform randomly distributed among these sites, thus each site will contain data with different class labels, and (2) data records were distributed based on their class labels in a round-robin fashion, e.g., suppose there are 5 classes and 3 sites, then records of class 1 and 4 are on site 1, records of class 2 and 5 are on site 2, and records of class 3 are on site 3.

Setup for vertically partitioned case The number of partitions was varied from 2 to 4 in experiments. Data columns were randomly distributed to each partition, and each partition contains about the same number of columns.

Privacy measure Three approaches have been proposed in the literature to measure privacy: the first using confidence interval [5], the second using information theory [3], and the third based on the notion of privacy breach [18,17]. However, information theory approach is inappropriate for K-means clustering and KNN classification because Euclidean distance is based on individual data values, while information theory only considers the distribution of values [41]. For example, suppose the original values of an attribute for 3 records are 0, 0.5, 1, respectively, and the transformed values are 0.5, 1, 0, respectively, the distributions of original and transformed values are the same, thus the privacy measure will be zero using information theory [3]. However, the transformed individual values are very different from the original values. Privacy breach-based methods consider the worst cases, but here interest lies in the average case. Thus, this paper uses the confidence interval method proposed in [5] to measure privacy. If a transformed attribute x can be estimated with c% confidence in the interval $[x_1, x_2]$, then the privacy equals

$$\frac{x_2 - x_1}{\max\{x\} - \min\{x\}}$$

where $\max\{x\}$ is the maximal value of x and $\min\{x\}$ is the minimal value. Ninety-five Percent confidence interval was used in the experiments.

For DCT-H and DCT-R, this paper considers two cases: the average case when the third party does not figure out the correct number of attributes and the permutation of coefficients, and the worst case when the third party does figure out the number of attributes and the permutation of coefficients. In the average case, the privacy is computed as following. The third party randomly guesses the number of attributes and a permutation of coefficients, and reconstructs the data using this permutation. The privacy is computed by comparing the reconstructed data and the original data. This process is repeated for 20 times and the average of privacy is reported. In the worst case, the privacy is computed by comparing the original data and the reconstructed data with correct number of attributes and permutation of coefficients. However, it should be justified to emphasize that in general the chances for the worst case to happen is very low because there are enormous number of permutations for even moderate number of attributes, and the third party has to at least reconstruct the whole data set for each permutation. The results section reports the average case privacy in Sect. 4.2, 4.3, and 4.4, and reports the worst case privacy in Sects. 4.5.

For Rand-P, it is also difficult for the third party to figure out the projection matrix because it is generated randomly. Thus, privacy is computed by directly comparing the transformed data with the original data, assuming the missing columns contain zeros.

Data mining quality measure: In this paper, the quality of classification is measured by accuracy. The quality of clustering is measured using the F measure that is widely used in information retrieval [33]. Let $C_1, C_2, \ldots, C_n$ be the correct clusters according to the dataset. Let $C1', \ldots, C'_n$ be the clusters generated by the clustering algorithm being examined. The F-measure of a correct cluster (or a class) C_i and an actual cluster C'_j is defined as follows:

$$F_{ij} = 2\frac{P_{ij}R_{ij}}{P_{ij} + R_{ij}}$$

Precision P_{ij} equals: $\frac{|C_i \cap C'_j|}{|C'_j|}$

Recall R_{ij} equals $\frac{|C_i \cap C'_j|}{|C_i|}$

The F-measure of a class C_i is given by

$$F_i = \max_j F_{ij}$$

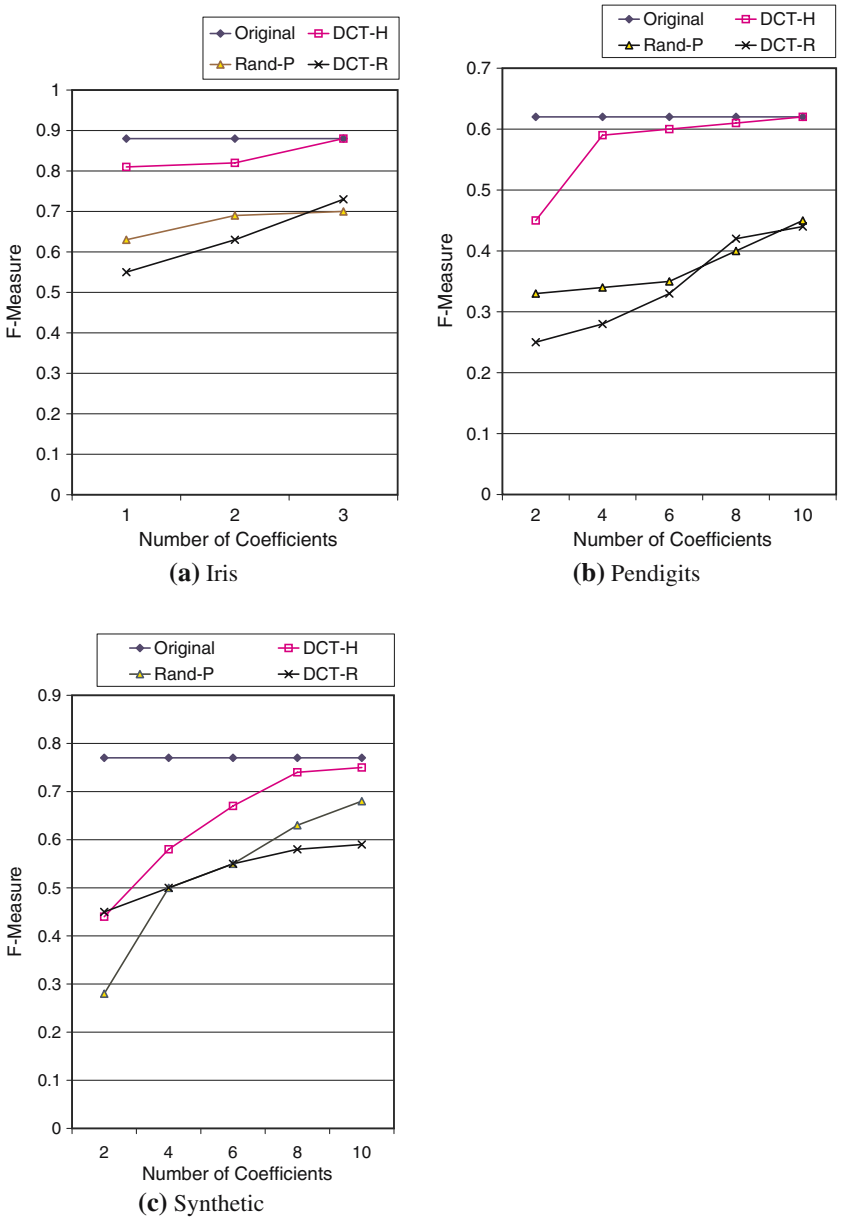


Fig. 6 Quality of clustering, varying μ

Finally, the overall F-measure is

$$F = \sum_{i=1}^{n} \frac{|C_i|}{N} F_i$$

where N is the total number of records.

4.2 Results for centralized database case

The privacy-preserving algorithms studied in experiments can be divided into two classes: those that both preserve privacy and reduce size of data (DCT-H, DCT-R, and Rand-P) and those that only preserve privacy (Rand-N and Rand-U). Thus, comparison is made between the first class of algorithms and then the first with the second class of algorithms.

4.2.1 Comparing DCT-H, DCT-R, and Rand-P

This section compares the following results of DCT-H, DCT-R, and Rand-P: (1) the quality of K-means and KNN over data generated by these algorithms, (2) the degree of privacy, and (3) the degree of data reduction.

Quality of mining The number of coefficients (μ) selected is an important parameter for algorithms using Fourier-related transform (DCT-H and DCT-R). For Rand-P, the number of dimensions being projected to is also important. For convenience μ is used to refer to both of these two parameters and is varied in experiments. Figure 6a–c reports the quality of K-means clustering for DCT-H, DCT-R, and Rand-P for different data sets. The quality of clustering of the original data is also plotted as a baseline for comparison. The results

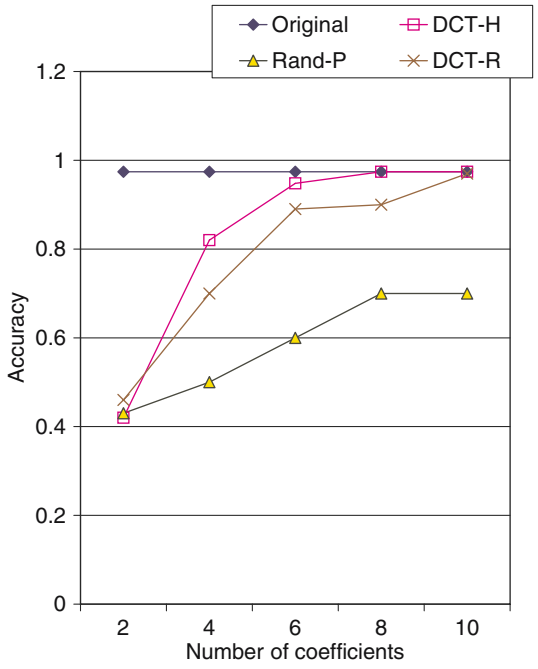


Fig. 7 Classification for Pendigits data, varying μ

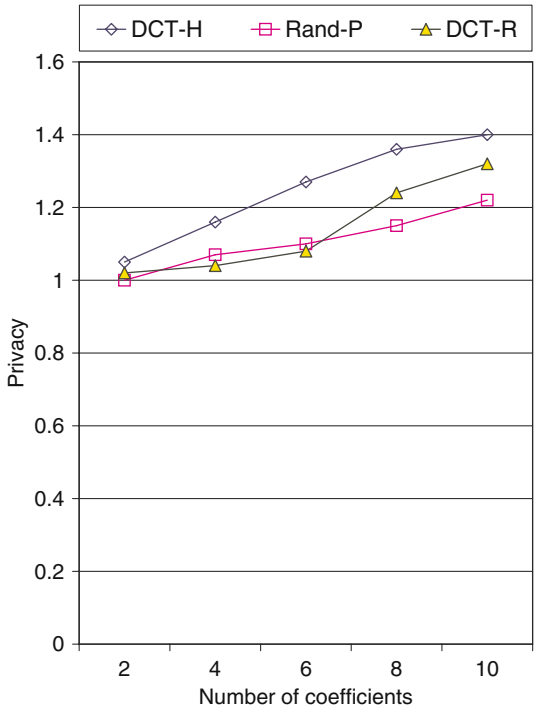


Fig. 8 Privacy for Pendigits, varying μ

for KNN classification using Pendigits data are reported in Fig. 7. The results for Iris and Synthetic data sets are similar and omitted.

The results show that for very small μ, the mining quality of all algorithms was poor because there were not enough coefficients to preserve the Euclidean distance with sufficient accuracy. However, as μ increased, the quality of all algorithms was improved. The mining quality of DCT-H grew the quickest among these algorithms, and became very close to the quality over original data. This indicates that DCT-H preserve the Euclidean distance quite accurately when using a few coefficients. On the other hand, the mining quality of DCT-R and Rand-P was significantly worse. DCT-H led to better mining quality than DCT-R because DCT-H used heuristics proposed in Sect. 3 to select coefficients that preserve most of the energy over a large number of rows. DCT-H led to better mining quality than Rand-P because random projection did not preserve Euclidean distance accurately when using a few dimensions.

Finally, as μ became sufficiently large (3 for Iris, and 10 for Pendigits), the differences between these algorithms became less significant because selection of coefficients became less important when most coefficients or dimensions were used. Overall, DCT-H led to better mining quality than DCT-R and Rand-P, and its mining quality is comparable to original data for a wide range of μ. Thus, in practice, users can start with a small μ and increase it if the Euclidean distance is not well preserved.

Degree of privacy Figure 8 reports the average case privacy of DCT-H, Rand-P, and DCT-R for Pendigits data with varying μ. The results for other datasets are similar and omitted. Section 4.2.2 will also present the results of privacy versus mining quality of all algorithms. The results show that all algorithms provided a high degree of privacy. However, unlike random projection methods, the degree of privacy may not decrease as the number of coefficients increases because the privacy is not only due to the loss of information caused by transform, but also due to the fact that the third party does not know the number of attributes and the correspondence between coefficients and their indexes. In fact the increase in number of selected coefficients decreases the probability of discovery of the right permutation order.

Degree of data reduction Figure 9 reports the size of transformed data divided by the size of original data for DCT-H, DCT-R, and Rand-P varying μ. Note that for the same μ, all three algorithms generate transformed data with the same size. The results show that all algorithms achieved significant data reduction. Note that DCT-H also achieved good mining quality for small μ, thus DCT-H achieved both good mining quality and data reduction. For example, according to Fig. 6c, DCT-H achieved the same quality of mining as original data when using data 12.5% the size of the original data (using 8 coefficients) for Synthetic data.

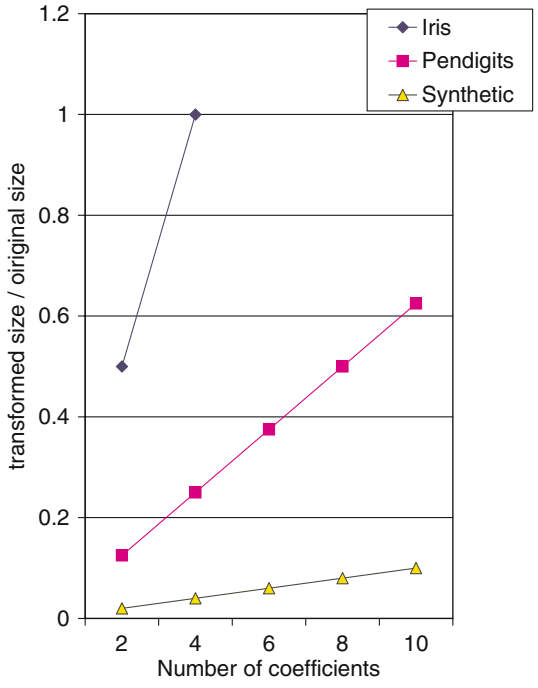


Fig. 9 Data reduction for DCT-H, DCT-R, and R and-P, varying μ

4.2.2 Comparing DCT-H and Rand-P with Rand-U, and Rand-N

This section compare DCT-H and Rand-P with Rand-U and Rand-N. The degree of privacy for Rand-U and Rand-P was varied from 20 to 200% to cover the range of privacy for DCT-H. Figure 10a–d reports the privacy and quality of K-means clustering and KNN classification for DCT-H, Rand-U, Rand-N, and Rand-P. DCT-H and Rand-P used 2 or more coefficients for Iris and 6 or more coefficients for Pendigits and Synthetic.

The results show that DCT-H always led to better mining quality than Rand-N and Rand-U when generating data with similar degree of privacy. The random noise added by Rand-N and Rand-U distorted the Euclidean distances, leading to poor mining quality. In all experiments, Rand-N and Rand-U only led to good mining quality when the degree of privacy is very low (around 20%). Further, unlike DCT-H, random perturbation methods also do not reduce the data size. The results also show that DCT-H led to better mining quality than Rand-P when generating data with similar degree of privacy, proving that DCT-H performs better in the accuracy/privacy trade-off than Rand-P.

4.3 Results for horizontally partitioned case

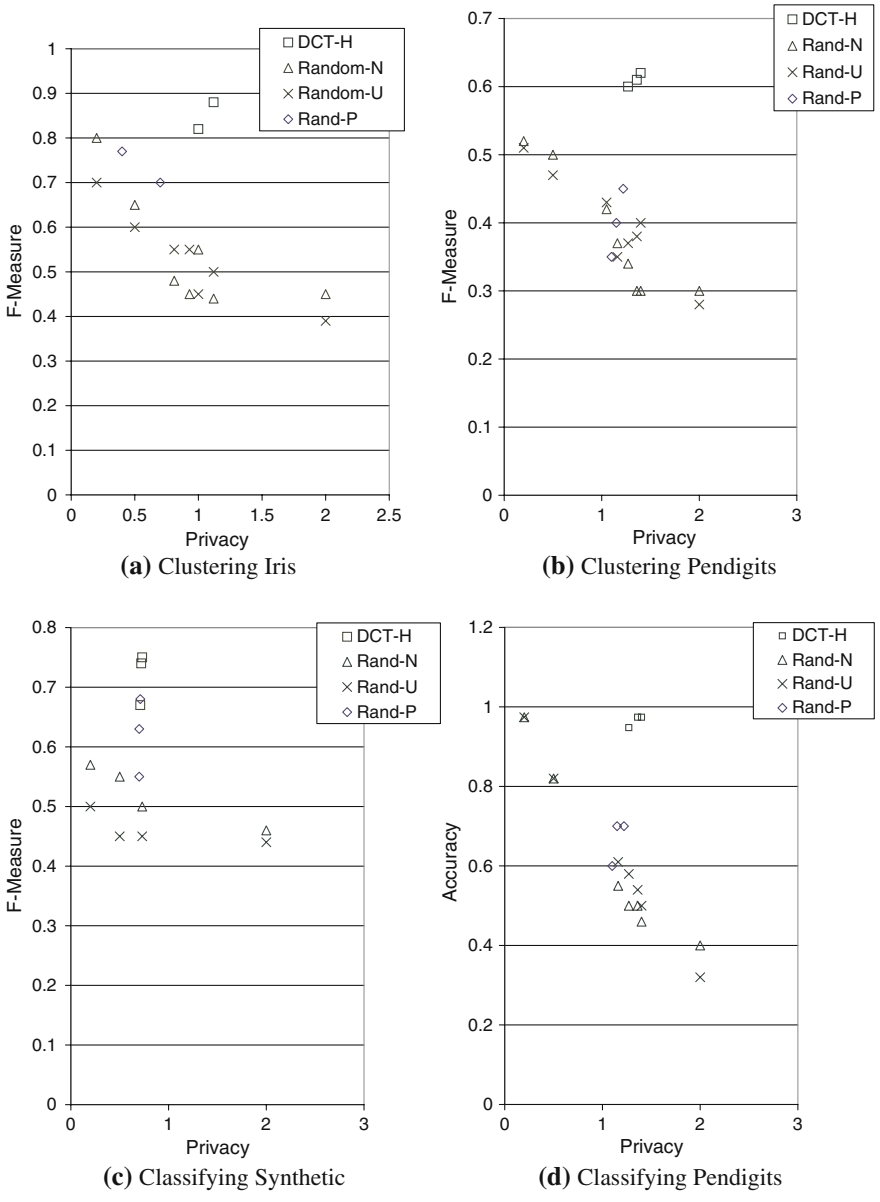
This section describes the results of KNN classification for the horizontally partitioned case. The results for clustering are similar and not reported. Since Rand-U and Rand-N add random noise to each data element independently, they can be applied to horizontally and vertically partitioned cases in the same way as the centralized database case. Rand-P can be applied to horizontally partitioned case as follows: the server first creates a random projection matrix and sends it to each partition, next each partition sends to the server a transformed partition which equals the product of the original partition and the random projection matrix, and finally the server concatenates these transformed partitions together.

Figure 11a reports the accuracy of KNN classification using distributed version of DCT-H for the Pendigits data when 6 coefficients were used and the number of sites was varied from 2 to 5. The data were distributed uniform randomly across these sites. Figure 11b reports the accuracy for Pendigits for the case of five sites when varying μ. The accuracy of DCT-H for centralized case and using original data, and the accuracy of Rand-P are also plotted. Figure 11c compares the privacy and accuracy of distributed DCT-H using 6–10 coefficients and five sites (number 5 arbitrarily chosen) with the results of Rand-U, Rand-N, and Rand-P. The degree of privacy was about the same for different number of sites. Thus, only the privacy for five sites was reported. It was also found that the results of Rand-U, Rand-N, and Rand-P were the same for different number of sources.

Figure 11a shows that the quality of KNN classification remained almost unchanged as the number of sites increases. Figure 11b shows that by using the heuristics proposed in Sect. 3.6, DCT-H achieved almost the same classification accuracy as the case of centralized data, and its accuracy is much higher than the accuracy achieved by Rand-P.

Figure 11c shows that in the average case, the degree of privacy using DCT-H was about the same as the degree of privacy for centralized case in Fig. 8. Further, Fig. 11c shows that DCT-H achieved significantly better accuracy than Rand-U and Rand-N for similar degree of privacy, when using six or more coefficients. Note that the accuracy of DCT-H was very close to the benchmark case (i.e., over original data) when using six or more coefficients. Further, DCT-H also reduced the data size (e.g., the transformed data is only 37.5% of the original data when using six coefficients), while Rand-U and Rand-N did not. The results also show that DCT-H achieves higher accuracy than Rand-P with similar degree of privacy.

Figure 11d reports the accuracy of KNN classification using DCT-H for the Pendigits data when classes were distributed to sites in a round-robin fashion. six coefficients were used and the number of sites was varied



(a) Clustering Iris **(b)** Clustering Pendigits

(c) Classifying Synthetic **(d)** Classifying Pendigits

Fig. 10 Privacy versus mining quality

from two to five. In this case, the classes were unevenly distributed in different partitions. The results show that DCT-H achieves about the same accuracy as the case when classes were uniform randomly distributed to all partitions. Thus, DCT-H works well for uneven distribution as well. This is expected because Algorithm 2 tries to preserve energy for every partition. The results, when varying μ and the results of privacy are similar for both uneven and evenly distributed cases too, and are not included due to space constraints.

4.4 Results for vertically partitioned case

This section describes the results for vertical partitioned case. Rand-U and Rand-N can be applied to vertically partitioned case in the same way as the centralized case. Rand-P can also be applied as follows: let each partition contains $m_1, m_2, \ldots, m_p$ columns, and $\sum_{i=1}^{p} m_i = m$. The server first generates a $m \times k$ random matrix R, then horizontally partitions R to p partitions where each contains m_i rows of R. Next, the server sends each partition R_i of R to the i-th site that has m_i columns, and each site sends back the product of the vertical partition stored at that site and R_i. Finally the server computes the sum of these products as the transformed data. It is easy to verify that the result of the above algorithm equals the product of the original data and R.

Figure 12a reports the quality of K-means clustering using distributed version of DCT-H for the Pendigits data when six coefficients were used and the number of sites was varied from one to four. Figure 12b reports the quality for Pendigits for the case of three sites when

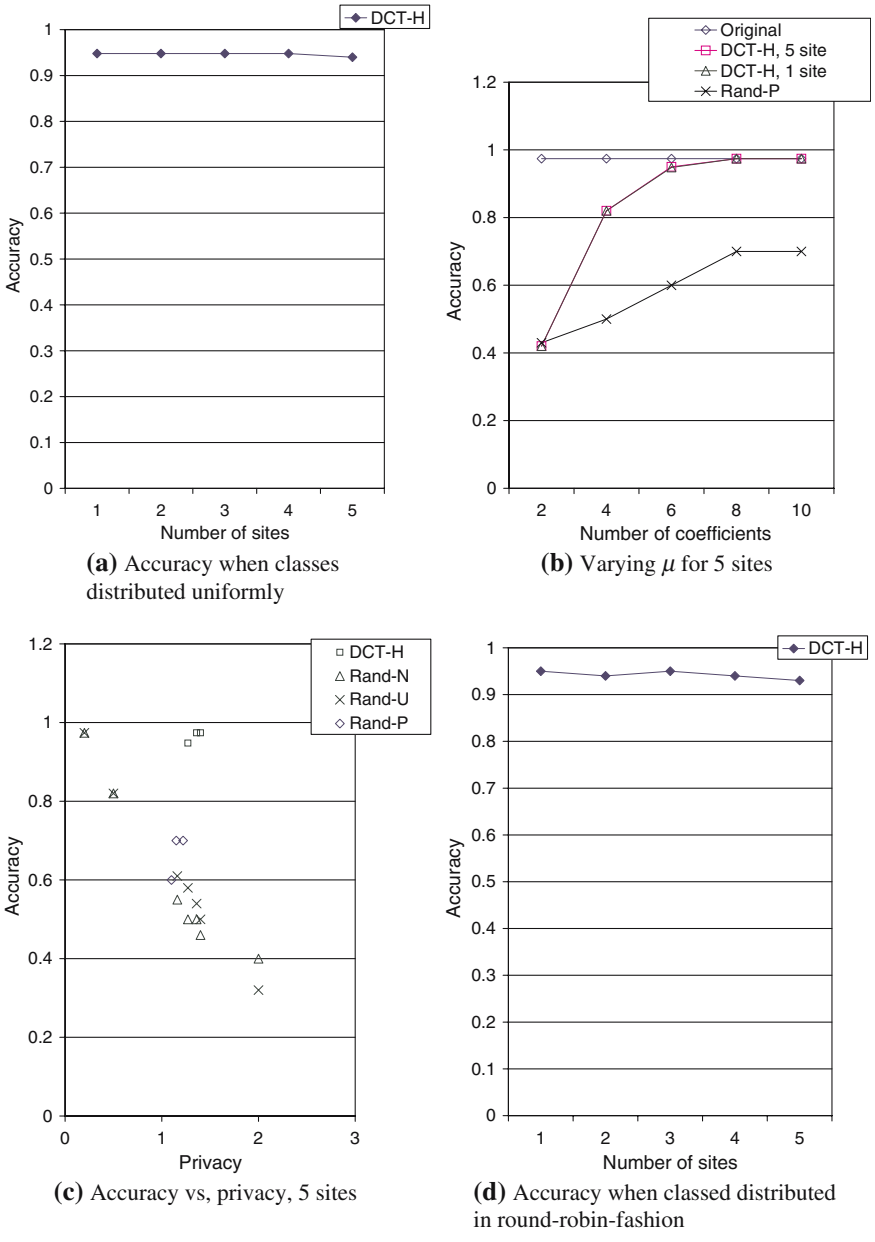


Fig. 11 Horizontally partitioned case, classifying Pendigits

(a) Accuracy when classes distributed uniformly

(b) Varying μ for 5 sites

(c) Accuracy vs, privacy, 5 sites

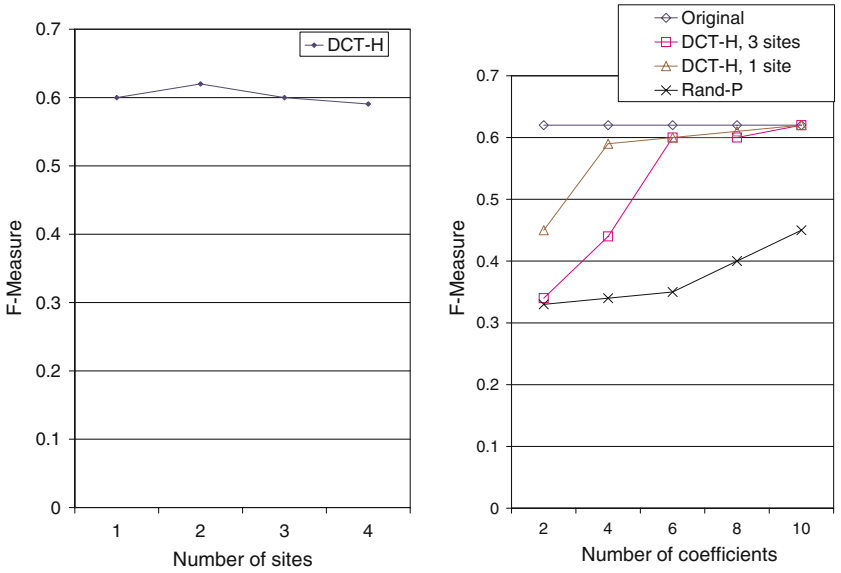
(d) Accuracy when classed distributed in round-robin-fashion

varying μ. Figure 12c compares the privacy and accuracy of vertically distributed DCT-H using 6–10 coefficients and three sites with the results of Rand-U, Rand-N, and Rand-P. The results for other data sets and classification are similar and omitted for space limitations. Figure 12a shows that the quality of clustering drops slightly as the number of sites increases from one to four. The mining quality was around 0.6 when 8 coefficients were used for four sites. Figure 12b shows that DCT-H achieved almost the same quality as the case of centralized case when using six or more coefficients, and its quality was much better than the quality achieved by Rand-P. Figure 12c also shows that DCT-H still achieved significantly better mining quality than Rand-U, Rand-N, and Rand-P when keeping similar degree of privacy.

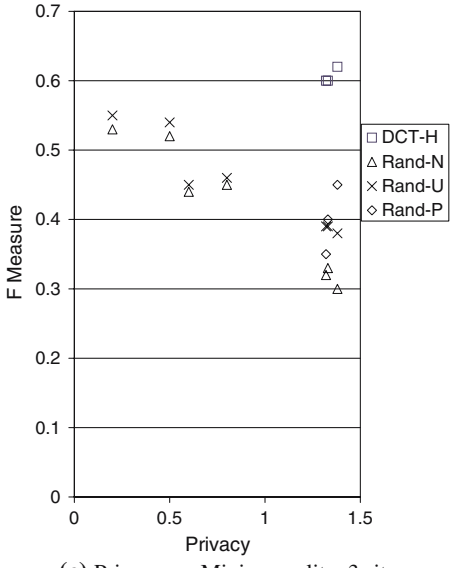
In the vertically partitioned case, privacy measure based on information theory also makes sense because if the privacy of any partition becomes zero, other partitions can learn the mean and standard deviation of the attributes in that partition. Figure 13 reports the conditional privacy defined in Agrawal and Aggarwal [3]. Suppose D_i is the i-th attribute values in original data and D_i' is the i-th attribute values in the reconstructed data. The conditional privacy for the i-th attribute is defined as:

$$2^{h(D_i|D_i')}$$

where $h(D_i|D_i')$ is the conditional entropy of original data D_i given the reconstructed data D_i'. The privacy is also computed as the average over all attributes and



(a) Varying number of sites $\mu = 6$ **(b)** Varying μ for 3 sites



(c) Privacy vs, Mining quality, 3 sites

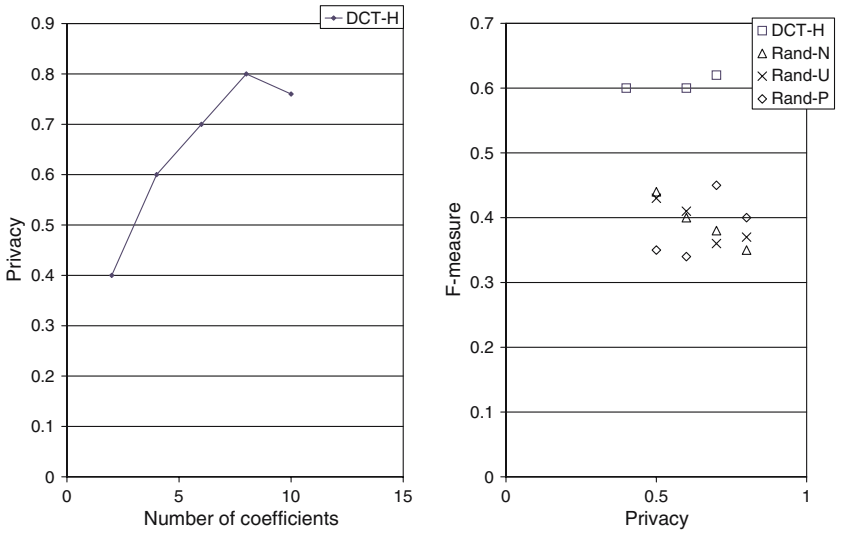
Fig. 12 Results for vertically partitioned case, clustering Pendigits

is then averaged over 20 reconstructed data sets, each using a randomly selected permutation of coefficients.

Figure 13a reports the degree of privacy for 3 sites when varying μ. According to Agrawal and Aggarwal [3], a conditional privacy of α means the privacy is the same as a random variable with a uniform distribution in $[0, \alpha]$. Since the data has been normalized to the range of [0, 1], the degree of privacy reported in Figure 13a is pretty high. Figure 13b compares the privacy and mining quality of vertically distributed DCT-H using 6–10 coefficients and three sites with the results of Rand-U, Rand-N, and Rand-P. DCT-H achieved better quality of mining than Rand-N, Rand-U, and Rand-P with similar degree of privacy using the entropy measure.

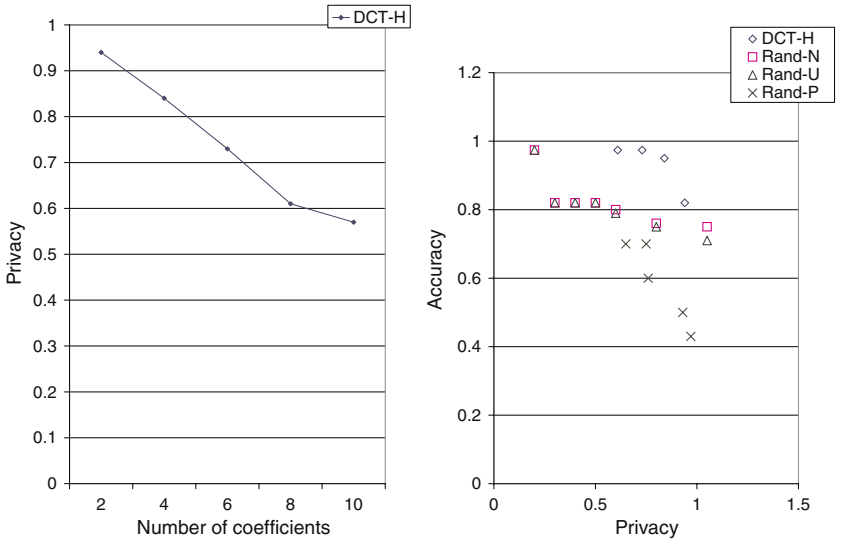
4.5 Worst case privacy

The proposed DCT-H method permutes the selected coefficients such that it is difficult, if not impossible for the third party to reconstruct the data. The previous sections have reported the degree of privacy when the third party does not discover the permutation. This section reports the worst case privacy when the third party does discover the correct permutation and reconstructs the data. Figure 14a reports the degree of privacy of running centralized DCT-H over Pendigits data when varying the number of coefficients. Figure 14b reports the accuracy of KNN classification against the degree of privacy for DCT-H, Rand-U, Rand-N, and Rand-P.



(a) Degree of privacy when varying μ **(b)** Privacy vs, Mining quality, 3 sites

Fig. 13 Information-based privacy for vertically partitioned case, clustering Pendigits



(a) Privacy when varying μ **(b)** Privacy vs, KNN accuracy

Fig. 14 Worst case privacy over Pendigits data

Figure 14a shows that the degree of privacy using DCT-H decreases as the number of coefficients increases, unlike when the permutation is not known because the third party can directly use inverse DCT to reconstruct the data. However, DCT-H still achieved considerable degree of privacy (75% when using 6 coefficients) because of the pruning of coefficients. Figure 14b shows that DCT-H still achieved significantly better accuracy than Rand-U and Rand-N for similar degree of privacy when using six or more coefficients (the case of 4 coefficients is the right most point of DCT-H in Fig. 14b). Note that the accuracy of DCT-H was very close to the benchmark case (i.e., over original data) when using six or more coefficients. Further, DCT-H also reduced the data size (e.g., the transformed data is only 37.5% of the original data when using six coefficients), while Rand-U and Rand-N did not. The results also show that DCT-H, in general, achieves higher accuracy than Rand-P, but with a lower degree of privacy.

4.6 Choosing the number of selected coefficients

Distance-based method The number of selected coefficients (μ) is an important parameter of the DCT-H algorithm. A large μ will not only preserve the distances better and lead to better mining quality, but also lead

to lower data reduction. However, in practice, it will be difficult to choose the value of μ based on mining quality because mining is done by the third party. Thus, a method to decide μ based on the loss of Euclidean distance is suggested. The intuition is that, if the distance is preserved to a high degree on an average between transformed records, the mining quality will also be high. The loss of distance is measured by the average loss of distance over all pairs of data points. Let d_{ij} be the distance between record i and j in the original data set, and d'_{ij} be the distance between record i and j in the transformed and pruned data. The average loss of distance is computed as the average of $\frac{d_{ij}-d'_{ij}}{d_{ij}}$ for all i and j. Note that the expression $d_{ij} - d'_{ij}$ will always be positive as a corollary of Lemma 2 derived earlier in Sect. 3.1. Figure 15 reports the average loss of distance for Pendigits data when varying μ. The results show that the loss of distance drops quickly to less than 10% when $\mu = 8$. Based on Figs. 6c and 7, the mining quality over transformed data is also very close to the mining quality over the original data when $\mu = 8$. In practice, users can plot the curve of loss of distance against μ and select an appropriate μ with little loss of distance, yet good data reduction and privacy.

Frequency-based method Computing the average loss of distance is expensive. Thus, an alternative simple heuristic is proposed. Consider the worst case when the energy of coefficients is randomly distributed for each row, that is, there is no energy concentrating factor and each coefficient is likely to have same energy. Thus, given m attributes and an integer Δ , where $0 < \Delta < m$, such that Δ coefficients with the highest energy are selected from each row, the probability for a coefficient i to be selected as a high energy coefficient is thus Δ/m. If there are n rows in a data set, then the theoretical expected value of frequency of appearance of that particular coefficient i in the high energy partition will be $n\Delta/m$. Now let $frq(i)$ denote the actual number of rows in the transformed data that have coefficient i in the high energy partition. The heuristic is to select all coefficients that have frequencies greater than $n\Delta/m$ in the high energy partition.

For example, Fig. 16 shows the plot of $frq(i)/n$ for Pendigits data when $\Delta = 8$ and $m = 16$, that is, $\Delta/m = 0.5$. The figure shows that there are 8 coefficients with $frq(i)/n$ greater than 0.5. Thus those 8 coefficients will be selected, which coincides with the selection of distance-based method.

There is still an issue of choosing Δ. Consider that when Δ increases, more coefficients will be considered high energy coefficients. The frequencies of coefficients increase as well. This may lead to more coefficients getting selected. However, the threshold Δ/m also increases, which may lead to fewer coefficients getting selected. A number of experiments were conducted with different Δ. The observation is that, almost same number of coefficients gets selected for a wide range of Δ. Thus, the above algorithm is not very sensitive to Δ. As a rule of thumb, users can select a Δ of $m/2$ because if energy is randomly distributed with the transform providing no energy compaction at all, then at least, on an

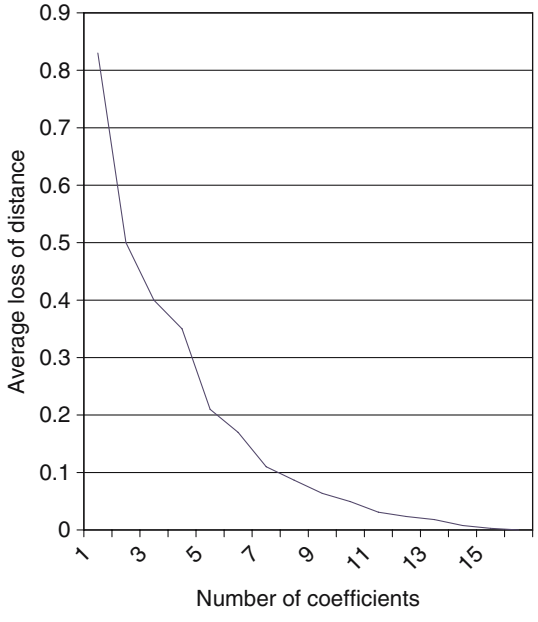


Fig. 15 Average loss of distance when varying μ for Pendigits

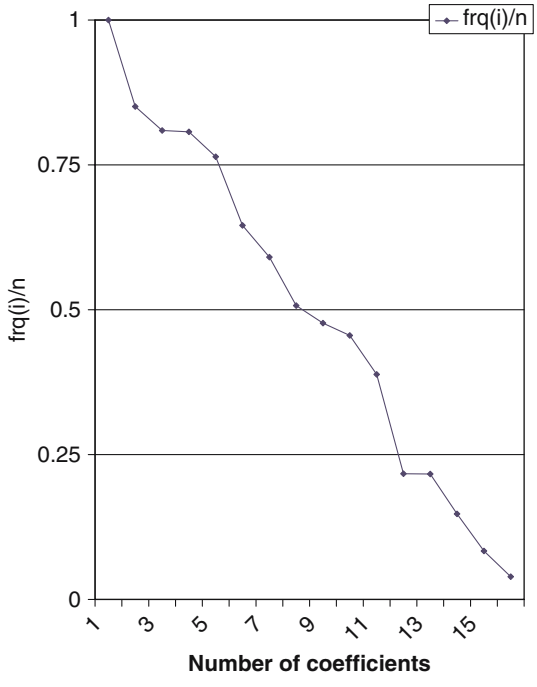


Fig. 16 Frequencies of high energy coefficients for Pendigits

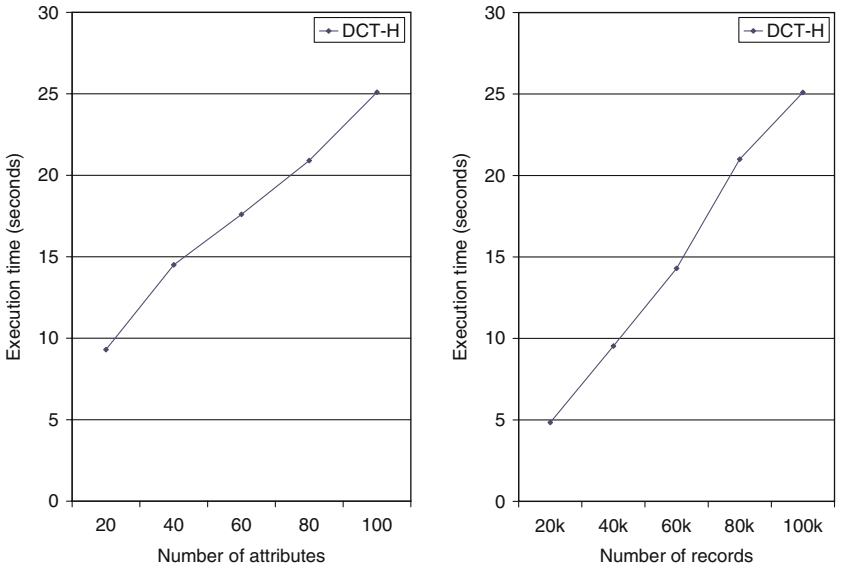


Fig. 17 Execution time of DCT-H

(a) Varying number of attributes **(b)** Varying number of records

average, it can be expected that half the coefficients will have energy above the average energy. Note that for a particular dataset, the comparison of the ratios $frq(i)/n$ and Δ/m gives the indication of how much effective the transform is, in compacting energy in a small set of common coefficients, across a large number of rows. Experiments showed that in almost all real life and synthetic data sets, DCT performs excellently in its ability to achieve the above. The results in the last section also point to the validity of the observation.

This heuristic runs much faster than the distance-based method because it adds no additional overhead to the algorithm (the frequencies are computed in Algorithm 1). In practice, the above heuristics turns out too conservative because it considers the worst case. Usually, energy is concentrated on fewer coefficients than the threshold found from this method. Thus, the threshold for frequencies can be raised and fewer coefficients can be selected. For example, 44 coefficients are selected for the Synthetic data set when 10 coefficients are selected by the distance-based method with 10% loss of distance. Thus, users can first use this simple heuristics to choose a more conservative μ, and if they want to further reduce the data size, they can use the distance-based method.

4.7 Overhead of proposed methods

This section studied the relationship between the overhead of DCT-H and the data size. There are two factors contributing to data size: the number of attributes and the number of records. Figure 17a reports the time for the centralized version of DCT-H to transform data (using 8 coefficients) when the data were generated the same way as the synthetic data, the number of records was fixed at 100,000, and the number of attributes increased from 20 to 100. Figure 17b reports the time when the number of attributes was fixed at 100, and the number of records increased from 20,000 to 100,000. The results for distributed versions of DCT-H are similar and omitted to conserve space. The results show that the execution time increased almost linearly with the number of attributes as well as the number of records. This is expected because the time complexity of DCT-H is $O(mn \log m)$ where n is number of records and m is the number of attributes, and $\log m$ increases very slowly.

DCT-H also reduces the size of the data, and thus reduces the time to mine the data. Figure 18 shows the breakup of time of running K-means clustering and heuristics to transform the data, over the Synthetic data set, when varying μ. The dotted line in the figure is the mining time on original data. The figure shows that the execution time of the heuristics is a small portion of the total execution time, and the mining time does get reduced due to the reduction of data size. The total execution time for DCT-H was lower than the total execution time over original data when less than 50 coefficients were selected.

5 Conclusions

This paper proposes a generalized integrated approach using Fourier-related transforms to support privacy preserving Euclidean distance-based mining algorithms and to reduce the data size to save the resources for

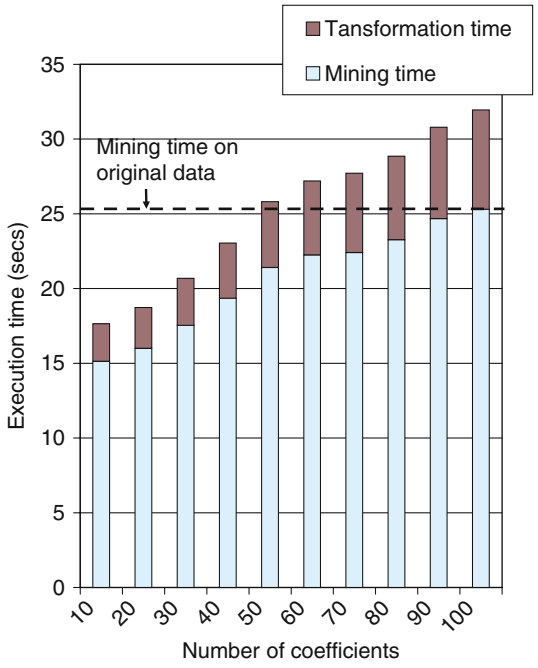


Fig. 18 Breakup of total execution time over Synthetic data

data transmission for distributed scenarios. Three algorithms to select the most important transform coefficients are presented, one for centralized database case, and the other two for horizontally and vertically partitioned database cases. A random permutation protocol is also suggested to be used to boost privacy. Experimental results demonstrate that the proposed approach leads to much better mining quality than the existing random perturbation and random projection approaches given the same degree of privacy in both centralized and distributed cases. The worst case privacy, arising when the random permutation protocol breaks, is also explored and the method achieves appreciable degree of privacy even there. The novelty of the approach is an attempt to bring a number of privacy-preserving mining techniques and scenarios sharing a common theme under the same umbrella. In the future the plan is to investigate how to provide probabilistic privacy and accuracy guarantee using the approach. As an off-shoot of this, it will be investigated if the problem is APX-hard. Further, the use of Fuzzy programming and entropy-based partitioning to refine the coefficient selection procedure will be explored. Research is also conducted in (1) extending the approach to a number of other mining tools that can be linked to distance-based mining with minor tweaks in the main algorithms and (2) extending to design of probabilistic coefficient filters for noise-resistant clustering and classification mining.

References

1. Aggarwal, C.C., Yu, P.S.: A condensation approach to privacy preserving data mining. EDBT, pp. 183–199 (2004)
2. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., Zhu, A.: Anonymizing tables. ICDT, pp. 246–258 (2005)
3. Agrawal, D., Aggarwal, C.C.: On the design and quantification of privacy preserving data mining algorithms. In: Twentieth ACM SIGMOD SIGACT-SIGART symposium on principles of database systems. pp. 247–255. Santa Barbara (2001)
4. Agrawal, R., Faloutsos, C., Swami, A.N.: Efficient similarity search in sequence databases. In: Fourth international conference of foundations of data organization and algorithms. pp. 69–84 (1993)
5. Agrawal, R., Srikant, R.: Privacy preserving data mining. In: 2000 ACM SIGMOD conference on management of data. pp. 439–450. Dallas (2000)
6. Agrawal, S., Haritsa, J.R.: A framework for high-accuracy privacy-preserving mining. ICDE, pp. 193–204 (2005)
7. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. ICDE. pp. 217–228 (2005)
8. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: The SuLQ framework. PODS, pp. 126–138 (2005)
9. Caragea, D., Silvescu, A., Honavar, V.: Decision tree induction from distributed, heterogeneous, autonomous data sources. In: Conference on intelligent systems design and applications, pp. 10–17 (2003)
10. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., Zhu, M.: Tools for privacy preserving distributed data mining. ACM SIGKDD Explor. **4**(2), 28–34 (2002)
11. Cover, T.M.: Rates of convergence for nearest neighbor procedures. In: Inter. Conference. on Systems Sciences. pp. 413–415 (1968)
12. D.Corney: Clustering with matlab. http://www.cs.ucl.ac.uk/staff/D.Corney
13. Du, W., Clifton, C., Atallah, M.J.: Distributed data mining to protect information privacy. In: NSF information and data management (IDM) workshop (2004)
14. Duda, R., Hart, P.E.: Pattern classification and scene analysis. John Wiley & Sons, Newyork (1973)
15. Dwork, C., Nissim, K.: Privacy-preserving data mining on vertically partitioned databases. CRYPTO, pp. 528–544 (2004)
16. Egecioglu, O., Ferhatosmanoglu, H., Ogras, U.: Dimensionality reduction and similarity computation by inner-product approximations. IEEE Trans. Know. Data Eng. **16**(6), 714–726 (2004)
17. Evfimevski, A., Gehrke, J., Srikant, R.: Limiting privacy breaches in privacy preserving data mining. In: 22nd ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems. pp. 211 – 222. San Diego (2003)
18. Evfimevski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: 8th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'02). pp. 217 – 228. Edmonton (2002)
19. Fung, B.C.M., Wang, K., Yu, P.S.: Top-down specialization for information and privacy preservation. ICDE, pp. 205–216 (2005)
20. Giannella, C., Liu, K., Olsen, T., Kargupta, H.: Communication efficient construction of decision trees over heterogeneously distributed data. In: Fourth IEEE international conference on data mining, pp. 67–74 (2004)
21. Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases (1998)

22. James, J.F.: A student's guide to Fourier transforms with applications in physics and engineering. Cambridge University Press, London (1995)
23. Kantarcioglu, M., Vaidya, J.: Privacy preserving naive Bayes classifier for horizontally partitioned data. In: IEEE ICDM workshop on privacy preserving data mining. pp. 3–9. Melbourne (2003)
24. Kargupta, H., Datta, S., Wang, Q., Sivakumar, K.: On the privacy preserving properties of random data perturbation techniques. ICDM, pp. 99–106 (2003)
25. Kargupta, H., Park, B.H.: A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments. IEEE Trans. Knowl Data Eng **16**(2), 216–229 (2004)
26. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: efficient full-domain k-anonymity. SIGMOD, pp. 49–60 (2005)
27. Lindell, Y., Pinkas, B.: Privacy preserving data mining. Advances in Cryptology (CRYPTO'00), Lect. Notes Comput. Sci. **180**, 36–53 (2000)
28. Merugu, S., Ghosh, J.: Privacy-preserving distributed clustering using generative models. In: 3rd IEEE international conference on data mining (ICDM'03) pp. 211–218. Melbourne (2003)
29. Oliveira, S., Zanane, O.R.: Privacy preserving clustering by data transformation. In: 18th Brazilian symposium on databases. pp. 304–318 (2003)
30. Oliveira, S., Zaiane, O.R.: Privacy-preserving clustering by object similarity-based representation and dimensionality reduction transformation. In: Workshop on privacy and security aspects of data mining (PSDM'04). pp. 21–30 (2004)
31. Oppenheim, A., Schafer, R.: Discrete-time signal processing. Prentice-Hall, Englewood Cliffs (1999)
32. Proakis, J.: Digital communications. McGraw-Hill, Newyork (2000)
33. Rijsbergen, C.J.V.: Information retrieval. Butterworths, London (1979)
34. Rizvi, S., Haritsa, J.R.: Maintaining data privacy in association rule mining. VLDB, pp. 682–693 (2002)
35. Samarati, P.: Protecting respondents' identities in microdata release. TKDE **13**(6), 1010–1027 (2001)
36. Vaidya, J.S., Clifton, C.: Privacy-preserving k-means clustering over vertically partitioned data. In: Nineth ACM SIGKDD international conference on knowledge discovery and data mining. pp. 206–215. Washington D.C (2003)
37. Vaidya, J.S., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. ACM SIGKDD 02, pp. 639–644. Edmonton (2002)
38. Wallace, G.: The JPEG still picture compression standard. Commun ACM, p. 35 (1991)
39. Wang, J.T., Wang, X., Lin, K.I., Shasha, D., Shapiro, B.A., Zhang, K.: Evaluating a class of distance-mapping algorithms for data mining and clustering. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp. 307–311 (1999)
40. Wu, C.W.: Privacy preserving data mining: a signal processing perspective and a simple data perturbation protocol. In: The 2nd workshop on privacy preserving data mining (PPDM'03), pp. 10–17 (2003)
41. Zhu, Y., Liu, L.: Optimal randomization for privacy preserving data mining. KDD, pp. 761–766 (2004)