Data & Knowledge Engineering 66 (2008) 264–288

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

A privacy preserving technique for distance-based classification with worst case privacy guarantees ☆

Shibnath Mukherjee, Madhushri Banerjee, Zhiyuan Chen *, Aryya Gangopadhyay

University of Maryland Baltimore County (UMBC), Information Systems Department (ITE 423), 1000 Hilltop Circle, UMBC, Baltimore, MD 21250, United States

ARTICLE INFO

Article history:
Received 27 February 2007
Received in revised form 27 February 2008
Accepted 26 March 2008
Available online 4 April 2008

Keywords:
Security and privacy
Data mining
Privacy preserving data mining
K-nearest neighbor classification

ABSTRACT

There has been relatively little work on privacy preserving techniques for distance based mining. The most widely used ones are additive perturbation methods and orthogonal transform based methods. These methods concentrate on privacy protection in the average case and provide no worst case privacy guarantee. However, the lack of privacy guarantee makes it difficult to use these techniques in practice, and causes possible privacy breach under certain attacking methods. This paper proposes a novel privacy protection method for distance based mining algorithms that gives worst case privacy guarantees and protects the data against correlation-based and transform-based attacks. This method has the following three novel aspects. First, this method uses a framework to provide theoretical bound of privacy breach in the worst case. This framework provides easy to check conditions that one can determine whether a method provides worst case guarantee. A quick examination shows that special types of noise such as Laplace noise provide worst case guarantee, while most existing methods such as adding normal or uniform noise, as well as random projection method do not provide worst case guarantee. Second, the proposed method combines the favorable features of additive perturbation and orthogonal transform methods. It uses principal component analysis to decorrelate the data and thus guards against attacks based on data correlations. It then adds Laplace noise to guard against attacks that can recover the PCA transform. Third, the proposed method improves accuracy of one of the popular distance-based classification algorithms: *K*-nearest neighbor classification, by taking into account the degree of distance distortion introduced by sanitization. Extensive experiments demonstrate the effectiveness of the proposed method.

1. Introduction

With the explosive growth of data and their shared and distributed sources, the need for cooperative and profitable analysis of the data has become increasingly high across organizations. Associated to this, however, are the concerns of privacy breaches of the shared data which might have important legal and strategic consequences for organizations. Information like salary, age, or credit ratings of individuals might turn out to be important data items to be mined and shared across companies to extract knowledge but cannot be directly shared for the concerns of privacy breaches. A typical scenario is as follows. Suppose a company *A* has a huge volume of home/car loan related data of a big customer pool. Based on this data, company *B* wants to classify some new individuals who applied for loans from company *B* as eligible or not. Clearly *A* cannot share the entire data directly with *B* without some scheme that will protect individual information yet allow the classification task of *B* to proceed with minimum error.

☆ The research was partially supported by NSF grant IIS 0713345.

* Corresponding author. Tel.: +1 410 455 8833; fax: +1 410 455 1073.
E-mail addresses: madhu2@umbc.edu (M. Banerjee), zhchen@umbc.edu (Z. Chen), gangopad@umbc.edu (A. Gangopadhyay).

0169-023X/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2008.03.004

There has been a rich volume of work on privacy preserving data mining which focuses on accurate mining while at the same time preserving privacy of the data [5,4,45,6,21,20,30,38,11,28,50,51,58]. However, there has been relatively little work on privacy preserving techniques for distance based mining methods. One of the most commonly used privacy preserving methods is additive perturbation approach which adds random noise to the data such that individual data values are distorted while the underlying distribution can be reconstructed with fair degree of accuracy [5,4,6,58]. However there are two major flaws for additive perturbation methods. First, Euclidean distances between individual data points are distorted. Thus the accuracy of distance based mining methods may drop. Second, additive perturbation methods are vulnerable to attacks based on data correlations. For example, [25] has shown that for highly correlated datasets, simply doing a Principal Component Analysis and reconstructing the data based on first few components can filter substantial amount of noise. [29] presented an even more sophisticated method of attacking additive perturbation through eigen analysis of random matrices and utilizing inherent correlations of datasets.

Of the very few works on privacy preserving techniques for distance based mining, most apply orthogonal transforms on the data so that distances between data points are preserved. Thus distance-based mining methods will achieve high accuracy over transformed data. Oliveira and Zaane [42] proposed several types of geometric transforms such as rotation, translation, and scaling that preserve Euclidean distance. Chen and Liu proposed a method that applies a random rotation [10] to the data. A random projection method has been proposed in [36]. This method projects original data of m dimensions to smaller number of k dimensions by multiplying the data with a random matrix of size $m \times k$. It is observed that Euclidean distance is often distorted to a great extent using this method when k is small [41]. Mukherjee et al. [41] have proposed a method using discrete cosine transform for distance-based mining. As shown in [41], the DCT method achieves a better privacy versus accuracy trade off than either the random projection method or the additive perturbation method for K-nearest neighbor classification and K-means clustering. To prevent attacks that do a reverse DCT, the selected coefficients are permuted column-wise and the permutation is concealed. Mukherjee et al. [40] further proposed a coefficient selection method using Fuzzy Linear Programming to achieve a tradeoff between privacy and accuracy of mining over the transformed data.

However the next section will show that none of the existing methods provide any worst-case privacy guarantee and this causes serious problems.

1.1. Problem of lack of worst-case privacy guarantee

Most existing studies consider some form or other of average privacy breach measures such as the confidence interval measure [5] or the entropy based measure discussed in [4]. However a privacy protection technique that works in the average case may not provide enough protection in the worst case. In the next few paragraphs, this paper will show that existing techniques are vulnerable to a number of attacking techniques. Further, without worst case guarantee, it is difficult to measure the extent to which an attacker can recover the data. In consequence, it will be difficult to apply such privacy preserving methods in real life because it is hard to convince users that these methods provide enough privacy protection.

As mentioned earlier, most existing privacy protection methods for distance-based mining are transform based. In the worst case, the transform may be recovered by attackers and the original data will be compromised. For example, for the DCT based methods proposed in [41], the transform will be compromised if the attacker somehow knows the permutation of DCT coefficients. For the random projection method, the random projection matrix will be compromised if the seed to generate this matrix is revealed.

Further, Liu et al. [35] proposed a method that can recover the transformation when attackers know a small sample of the original data or a sample of data that follows the same distribution of the original data. This attacking method uses the fact that eigen values of the covariance matrix of the sanitized data are the same as eigen values of the covariance matrix of the original data. This allows the attacker to infer the transform from the perturbed data and the known sample. Thus using transform based method alone is not safe in the worst case (e.g., when some sample points are known) because the transformation may be recovered in the worst case.

Since using transformation methods alone is not safe, one could certainly try to use an additive perturbation method or combine an additive perturbation method with a transform method to provide more privacy protection. However, as mentioned earlier, additive perturbation methods alone are also vulnerable to attacks using inherent correlations of data [25]. Additive perturbation methods are also vulnerable to other type of attacks that are based on conditional probability [20]. An example will illustrate this.

Example 1.1. consider a data set that contains salary of employees and only one employee has salary of 150,000 and other employees have a lower salary. Suppose a uniform noise ranging from $[-100{,}000, 100{,}000]$ has been added to the salary of employees. The ranges are known to an attacker.

Let X be a random variable representing original data value and Y be a random variable representing randomized data value. Suppose the attacker observes a randomized salary of 250,000 (i.e., $Y = 250,000$). The conditional probability of the original salary being 150,000 (i.e., $P[X = 150000|Y = 250000]$) is 1 because otherwise the perturbed salary will be lower than 250,000. Thus the original salary can be inferred as 150,000. This attacking method is called ρ_1 to ρ_2 privacy breach where ρ_1 here is the probability of the salary being 150,000 in the original data, and ρ_2 is the conditional probability of salary being 150,000 after observing a perturbed value of 250,000.

1.2. Contributions

This paper proposes an approach that will provide worst case privacy guarantees for distance-based mining algorithms. This approach combines additive perturbation and Principal Component Analysis [26] to provide better privacy protection. The basic idea is to apply PCA first to the data set, and then add a special type of noise such as Laplace noise (the reason will be explained in Section 4) to the principal components. PCA removes correlations in data. Thus unlike existing additive perturbation methods, the noise added by proposed approach cannot be filtered out using attacking methods such as the one proposed in [25] that use data correlations.

The noise addition step provides two additional privacy protections. First, it protects against attacking methods such as [35] that can recover the transform and thus can recover the original data. Such attacking methods can recover the PCA transform, but cannot remove the added noise. Second, the noise addition step protects against the ρ_1 to ρ_2 privacy breach discussed in [20].

The contributions of this work are the follows:

- This paper extends the worst case privacy guarantee model proposed in [20] to continuous data. It also proposes necessary and sufficient conditions for additive noise distributions to provide worst case privacy guarantees. Interestingly, this paper also shows that many existing methods such as additive perturbation with normal or uniform noise, as well as random projection method which are secure in the average case do not provide worst case guarantee. This paper shows that noise following Laplace distribution will give worst case privacy guarantees.
- This paper proposes a data sanitization method for distance-based mining. This method combines PCA and additive perturbation method, and guards against several existing attacking methods, including the method that recovers transformation [35], the method that filters noise using data correlation [25], and the method that uses ρ_1 to ρ_2 privacy breach [20]. To the best of our knowledge, our method is the first that guard against all these three attacking methods.
- This paper proposes modification to the well known *K*-nearest neighbor classification algorithm to make it perform more accurately over the perturbed data. To the best of our knowledge, no effort has been made so far to make additive perturbations suited for distance based algorithms. The major challenge is that data points are scattered after noise addition. This paper modifies the nearest neighbor-search space of the ordinary *K*-nearest neighbor algorithm by taking into account the distance distortion introduced by the added noise. This modification improves the accuracy of *K*-nearest neighbor classification over sanitized data.

Although this work primarily focuses on distance-based mining, the first contribution also applies to any additive perturbation methods, regardless of the mining methods. For example, one can add Laplace noise to data and apply the same distribution reconstruction techniques proposed in [5,4] for decision tree mining. The second contribution is also applicable to any distance-based mining method. As reported in [17], most existing work on privacy preserving mining only targets one type of mining methods, and there is a great need for methods that can work for multiple mining methods. The first and second contribution of this work is one step in the right direction and we plan to investigate privacy preserving methods that can provide worst case privacy guarantee for multiple mining methods as future work.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 provides necessary background to distance-based mining and the worst-case privacy protection model proposed in [20]. Section 4 extends the model to continuous data and proposes necessary and sufficient conditions for an additive noise distribution to provide worst case privacy guarantees. This section also analyzes whether several commonly used noise distributions give such guarantees. The results for random projection method is also investigated. Section 5 presents the details of the proposed method and the modification to *K*-nearest neighbor classification. Section 6 reports the results of extensive experiments over real-life and synthetic data. The results demonstrate the superiority of the proposed method over existing ones. Section 7 concludes the paper and discusses future work.

2. Related work

Existing research on privacy can be divided into two categories. The first category tries to hide the data values while the second category tries to hide the identity of entities when publishing data [47,34,2,8,22,46,37,55]. This paper focuses on the first type of privacy problem.

2.1. Privacy preserving work for distributed mining

The first type of research can be further divided into two subcategories. The first one considers a distributed environment where multiple parties want to do data mining jointly but at the same time keep their own data private. There has been a rich body of work in this field. Since many data mining algorithms generate learning models which are essentially functions, existing work in this field applies techniques called *secure multi-party computations (SMC)* [23]. SMC allows different parties to share information securely and jointly calculate some function over datasets of all parties. [16] discussed many of the protocols developed so far in the field. [44] discussed some applications of these methods in privacy preserving data mining.

Most proposed methods use various modifications of secure multi-party protocols in different kinds of mining techniques and scenarios. [27,51] proposed methods for secure computation of association rules over horizontally and vertically partitioned data. [19,9] addressed the issue of building decision trees for heterogeneous distributed data while [28] proposed a method to build Naive Bayes classifier over horizontally partitioned data. [32] reported the use of Fourier representation of decision trees on horizontally partitioned data. Methods for secure *K*-nearest neighbor classification and k-means clustering over distributed data were also discussed in [57,31,50], respectively. [52] gave a survey of research in this area. However, this paper considers the case that data is sanitized and then shipped to a third party for mining.

2.2. Privacy preserving methods for non-distributed mining

The second subcategory considers non-distributed environment. Unlike the work for distributed environment, work in this area typically does not use intermediate results of mining, thus the mining is completely done by the receiver of the data. The work for non-distributed environment typically sanitizes data through randomization of values. Additive perturbation is probably the first and simplest method in this category. [5] used the scheme to sanitize data and developed a method of building decision tree classifiers over the sanitized data. [58] discussed issues of optimal randomization. However, as mentioned in Section 1, additive perturbation is vulnerable to attacks based on data correlations [25,29]. Kim and Winkler [33] proposed a Multiplicative perturbation method that multiplies a random number with mean 1 to the original data. Note that multiplicative perturbation is the same as adding a random noise to the log of original data values. Thus this method is as vulnerable as additive perturbation under attacks based on data correlations.

As mentioned in Section 1, a few transform-based methods have been proposed for distance-based mining [36,42,41,40,10]. As described in Section 1, these methods are vulnerable to attacks that can recover the transform [35].

A data swapping approach was proposed in [13]. It randomly swaps the values of different data points. This method is not safe because there is no guarantee that the swapped values will be much different from the original values.

A condensation approach was proposed in [1]. The method first divides data into size k clusters, and then replaces data points in each cluster with a random sample generated based on the distribution of the original data in that cluster. This method may work for some distance-based mining algorithms because the distance between clusters are likely preserved. However, it has no guarantee in terms of privacy because the generated sample values may be very close to the original data values.

A randomized response approach was also proposed [18] for decision tree mining. It is unclear whether it will be applicable to distance-based mining.

Further, as mentioned in Section 1, all above methods only consider average case privacy and do not give any worst case privacy guarantee. This paper proposes a method that gives worst case privacy guarantees and guards against several existing attacking methods mentioned in Section 1.

Evfimevski et al. first proposed a worst-case privacy model and used it for association rule mining in [20]. [6] further developed an optimal matrix theoretic scheme for random perturbations of *categorical* data and provided guarantees through amplification. However both [20,6] suggest some specific randomization schemes suited for discrete data and only consider mining of association rules from the perturbed data. This paper extends the model to continuous data. This paper also proposes necessary and sufficient conditions for an additive perturbation method to give privacy guarantee under this model. It also proposes a method for preserving privacy for distance-based mining. To the best of our knowledge, this paper is the first to use amplification to provide worst case privacy guarantees for additive perturbation methods.

3. Background

This section gives the necessary background to distance based mining (Section 3.1) and to the worst case privacy model over discrete data (Section 3.2).

3.1. Distance based mining

Distance-based mining is a class of mining methods that use distance between data points. The basic assumption of these methods is that if two data points x and y are close to each other, they should have similar properties (e.g., having the same class label). Two of the most popular distance based mining methods are *K*-Nearest Neighbor (KNN) classification [12] and *K*-means clustering [39]. The basic KNN algorithm works as follows. Given a training data set and a test data point x, the k closest training data points to x are selected. The class label of x is set to the class label that appears most frequently in the *K*-nearest neighbors. There also has been a lot of research on efficient algorithms to classify a test data point using KNN, including using index structures [3,56] and optimizing the KNN search using cost models [49].

3.2. Worst case privacy model for discrete data

This section describes a worst case privacy model first proposed in [20]. This model is called the *amplification* model because it uses a concept called amplification to compute the degree of worst case privacy protection for a type of attack called ρ_1 to ρ_2 privacy breach. This model was developed for discrete data. Section 4 will extend this model to continuous data.

Below is the formal definition of ρ_1 to ρ_2 privacy breach. Let X be a random variable that represents an attribute in the original data. Let x be an instance of X. x will be randomized to y using a randomization operator $R(x)$ (i.e., $R(x) = y$). Let Y be the random variable represent the result of $R(X)$, i.e., the results after randomization. Let y be an instance of Y. Let V_X and V_Y be the domain of X and Y, respectively. Let $Q(X)$ be a property of X, for example, whether the value of X falls in a certain range.

Definition 1. There is a ρ_1 to ρ_2 privacy breach with respect to property $Q(X)$ if for some $y \in V_Y$

$$P[Q(X)] \leqslant \rho_1 \quad \text{and} \quad P[Q(X)|Y = y] \geqslant \rho_2,$$
$$\text{where } 0 < \rho_1 < \rho_2 < 1, \quad P[Y = y] > 0.$$

This definition is called upward ρ_1 to ρ_2 privacy breach. Here $Q(X)$ is the property of the original data (could be the values of the original data). The definition states that if $Q(X)$'s probability is very low (below ρ_1) in the original data, but its conditional probability given the observed randomized value of y is very high (over ρ_2), then a third party can infer that the original data satisfies $Q(X)$ if he sees the perturbed value of y. A ρ_2 to ρ_1 downward privacy breach is defined similarly in [20] which depicts a situation vice versa to the one stated above. Example 1.1 in Section 1 illustrates a case of upward privacy breach, where $Q(X)$ is the property that salary of an employee equals 150,000.

A randomization operator provides a certain degree of worst case privacy if it does not allow any ρ_1 to ρ_2 or ρ_2 to ρ_1 privacy breach. [20] introduces the definition of amplification to provide such guarantees.

Definition 2. A randomization operator $R(x)$ is at most γ-amplifying for $y \in V_Y$ if

$$\forall x_1, x_2 \in V_X : \quad \frac{P[x_1 \to y]}{P[x_2 \to y]} \leqslant \gamma.$$

Here $P[x \to y]$ represents the probability of mapping a value $x \in V_X$ to a value $y \in V_Y$. It is important to note that as explained in [20], this probability does not depend on the distribution of X (only the domain of X is needed). Intuitively, amplification defines the maximal ratio between probabilities of any two original data values being randomized to some same y value, and thus limits the possibility of inferring the original value based on y value. In [20] the authors proved a very useful bound relating to the amplification and ρ_1 to ρ_2 privacy breach.

Theorem 3. *Let R be a randomization operator, $y \in V_Y$ be a randomized value such that $\exists x : P[x \to y] > 0$ and $0 < \rho_1 < \rho_2 < 1$ be the two probabilities from Definition 1. Suppose R is at most γ-amplifying for y. Revealing $R(X) = Y$ will cause neither an upward ρ_1 to ρ_2 privacy breach nor a downward ρ_2 to ρ_1 privacy breach with respect to any property if*

$$\frac{\rho_2(1-\rho_1)}{\rho_1(1-\rho_2)} > \gamma. \tag{1}$$

Based on Theorem 3, a randomization operator will provide the worst case privacy guarantee if it has a bounded amplification value. The degree of the guarantee is determined by the amplification value as the way stated in this theorem. Considering an upward privacy breach, if ρ_1 is fixed, after some arrangement of Eq. (1), there will be no upward privacy breach if

$$\rho_2 \geqslant \frac{\gamma\rho_1}{1 + (\gamma - 1)\rho_1}. \tag{2}$$

For example, suppose $\gamma = 20$, $\rho_1 = 0.001$, then for any $\rho_2 \geqslant 0.02$, it is guaranteed that no ρ_1 to ρ_2 privacy breach will occur.

4. Worst case privacy model for continuous data

This section describes a worst case privacy model for distance-based mining. Since Euclidean distances are computed over continuous data but the amplification model described in Section 3.2 only applies to discrete data, Section 4.1 extends this model to continuous data. Section 4.2 identifies the necessary and sufficient conditions for additive noise distributions to give bounded amplification values. Section 4.3 examines whether several commonly used noise distributions as well as random projection method give bounded amplification values and hence provable privacy guarantees.

4.1. Extension to continuous data

This section extends the amplification model for continuous data and additive perturbation. Let X be a bounded continuous random variable defining data, following a distribution having finite support. In most real life scenarios, the assumption that the data is bounded is quite justified. Further, let Δ be a random variable that represents the additive noise and follows a continuous probability density distribution $f_\Delta(\delta)$, where δ is a specific value of Δ.

Note that for continuous data, the probability at each value is zero. Thus it is natural to consider the probability of mapping a small range of values of X to a certain y. The equation below represents the probability of mapping an x value in the range of $[x_1, x_1 + \mathrm{d}x]$ to y

$$P([x_1, x_1 + \mathrm{d}x]) \to y.$$

Similarly one can define a probability of $P([x_2, x_2 + \mathrm{d}x]) \to y$. Based on the definition of probability density function, we have

$$\lim_{\mathrm{d}x\to 0} P([x_1, x_1 + dx]) \to y/\mathrm{d}x = \lim_{\mathrm{d}x\to 0} P(x + \delta = y | x \in [x_1, x_1 + \mathrm{d}x])/\mathrm{d}x = \lim_{\mathrm{d}x\to 0} P(\delta = y - x | x \in [x_1, x_1 + \mathrm{d}x])/\mathrm{d}x$$
$$= \lim_{\mathrm{d}x\to 0} P(y - x_1 - \mathrm{d}x \leqslant \delta \leqslant y - x_1)/\mathrm{d}x = f_{\Delta}(y - x_1) \text{ since } f_{\Delta}(\delta) \text{ is continuous.}$$

Let A denote the limit of the ratio of $P([x_1, x_1 + \mathrm{d}x]) \to y / P([x_2, x_2 + \mathrm{d}x]) \to y$ as $\mathrm{d}x \to 0$. Then amplification can be defined for continuous data as follows:

$$A_{\max} = \max_{x_1, x_2 \in V_x, y \in V_y} \lim_{\mathrm{d}x\to 0} \frac{P([x_1, x_1 + \mathrm{d}x] \to y)}{P([x_2, x_2 + \mathrm{d}x] \to y)} = \max_{x_1, x_2 \in V_x, y \in V_y} \lim_{\mathrm{d}x\to 0} \frac{P([x_1, x_1 + \mathrm{d}x] \to y)/\mathrm{d}x}{P([x_2, x_2 + \mathrm{d}x] \to y)/\mathrm{d}x} = \max_{x_1, x_2, y} \frac{f_{\Delta}(y - x_1)}{f_{\Delta}(y - x_2)}. \quad (3)$$

Here $y - x_1$ and $y - x_2$ are essentially the amount of noise added to points x_1 and x_2. Similarly Theorem 3 can be extended to continuous data using the amplification defined for continuous data.

4.2. Necessary and sufficient conditions for bounded amplification

As mentioned earlier, a bounded amplification provides the worst case privacy guarantee. However, not all types of noise will give bounded amplifications. This section presents the necessary and sufficient conditions for the probability density function of added noise ($f_{\Delta}(\delta)$) to give bounded amplification ($A_{\max}$).

Theorem 4. *Amplification for an additive noise distribution is bounded if and only if the following three conditions hold:*

1. *The density distribution of noise, $f_{\Delta}(\delta)$, should have infinite support, that is $\Delta \in (-\infty; +\infty)$.*
2. *$f_{\Delta}(\delta)$ should not have a zero value for any value of Δ in $-\infty < \Delta < +\infty$.*
3. *$\frac{f_{\Delta}(y - x_1)}{f_{\Delta}(y - x_2)}$ should be bounded as $y \to \pm\infty \forall x_1, x_2 \in V_x$.*

Proof. *Necessity:*

Condition 1. This condition can be proved by contradiction. Fig. 1a gives the intuition of proof. Let the density distribution of Δ be supported in a finite interval $a \leqslant \Delta \leqslant b$. This paper assumes that x is bounded. Let $x_L \leqslant x \leqslant x_U (x_L < x_U)$, and y_{max} be the maximum possible value of y. Thus, $y_{\max} = x_U + b$. By the definition of amplification, amplification is bounded only if x_L can be also mapped to $y_{\max}$ with non zero probability. However, this is impossible because the maximal noise is b, and $x_L < x_U$, thus the maximal value x_L can be mapped to is $x_L + b$, which is less than $x_U + b$ (i.e., $y_{\max}$).

Condition 2. To prove the necessity of this condition, consider the following lemma.

Lemma 5. *If the density function f_{Δ} can have zero values, then there exists a pair of δ_1, δ_2 such that $f_{\Delta}(\delta_2) = 0, |\delta_2 - \delta_1| \leqslant x_U - x_L$, and $f_{\Delta}(\delta_1) > 0$.*

This lemma basically states that it is possible to find a pair of δ_1 and δ_2 that are close enough (within $x_U - x_L$) and have a positive and zero f_{Δ} values respectively.

If this lemma is true, then it is quite straightforward to prove the necessity of Condition 2. Fig. 1b shows the intuition of proof. Simply take the pair of δ_1 and δ_2 described above, and assume without loss of generality that $0 < \delta_2 - \delta_1 \leqslant x_U - x_L$. Let $x_2 = x_L, x_1 = x_L + \delta_2 - \delta_1$. Clearly, x_1 also falls in the range of x_L and x_U because $x_1 = x_L + \delta_2 - \delta_1$, and $0 < \delta_2 - \delta_1 \leqslant x_U - x_L$. Note that $x_1 + \delta_1 = x_L + \delta_2 - \delta_1 + \delta_1 = x_L + \delta_2$, and $x_2 + \delta_2 = x_L + \delta_2$. Thus $x_1 + \delta_1 = x_2 + \delta_2$ as shown in Fig. 1b.

Let $y = x_2 + \delta_2$, thus $y = x_1 + \delta_1$ too. Now $f_{\Delta}(y - x_1) = f_{\Delta}(\delta_1) > 0$, but $f_{\Delta}(y - x_2) = f_{\Delta}(\delta_2) = 0$. Thus the amplification which equals the ratio of $f_{\Delta}(y - x_1)$ to $f_{\Delta}(y - x_2)$ is infinite. Hence f_{Δ} has positive value at all points is necessary for finite amplification. The remaining task is to prove the lemma.

This lemma can be proved by contradiction. Let us assume $f_{\Delta}(\delta_2) = 0$. The intuition of the proof is given in Fig. 1 (c). If the lemma is not true, there is no δ_1 within distance $x_U - x_L$ to δ_2 and has positive f_{Δ} value. Thus $f_{\Delta}(\delta_1)$ must be 0 for any δ_1 such that $|\delta_2 - \delta_1| \leqslant x_U - x_L$, i.e., $f_{\Delta} = 0$ within the interval of $[\delta_2 - (x_U - x_L), \delta_2 + x_U - x_L]$. Note that the density function f_{Δ} is zero at the two boundaries $\delta_2 + x_U - x_L$ and $\delta_2 - x_U + x_L$ as well. Thus the above deduction can be applied again at these two boundaries (i.e., consider δ_2 be one of the two boundary points). Now the zero density interval can be expanded to $[\delta_2 - 2(x_U - x_L), \delta_2 + 2(x_U - x_L)]$ as shown in Fig. 1c. This process can be repeated forever. Thus all points in $(-\infty, +\infty)$ have zero density. This contradicts with the fact that f_{Δ} is a density function and should have some points with positive density. Hence the lemma must be true.

Condition 3. By Condition 1, $f_{\Delta}(\delta)$ is supported in $\delta \in (-\infty; +\infty)$. Thus if Condition 3 does not hold true, then $A \to \infty$ for either of $y \to \pm\infty$ and amplification becomes unbounded.

Sufficiency: The proof of sufficiency is straightforward. By Conditions 1 and 2, clearly for any $x_2, y, f_{\Delta}(y - x_2) > 0$. Thus for any finite x_1, x_2, y, the amplification is finite and bounded. Since x_1 and x_2 are always bounded, we only need to consider the case y tends to infinity. Since Condition 1 allows y to be in $(-\infty; +\infty)$, Condition 3 makes sure that as y tends to $+\infty$ or $-\infty$, the amplification is still finite and bounded. □

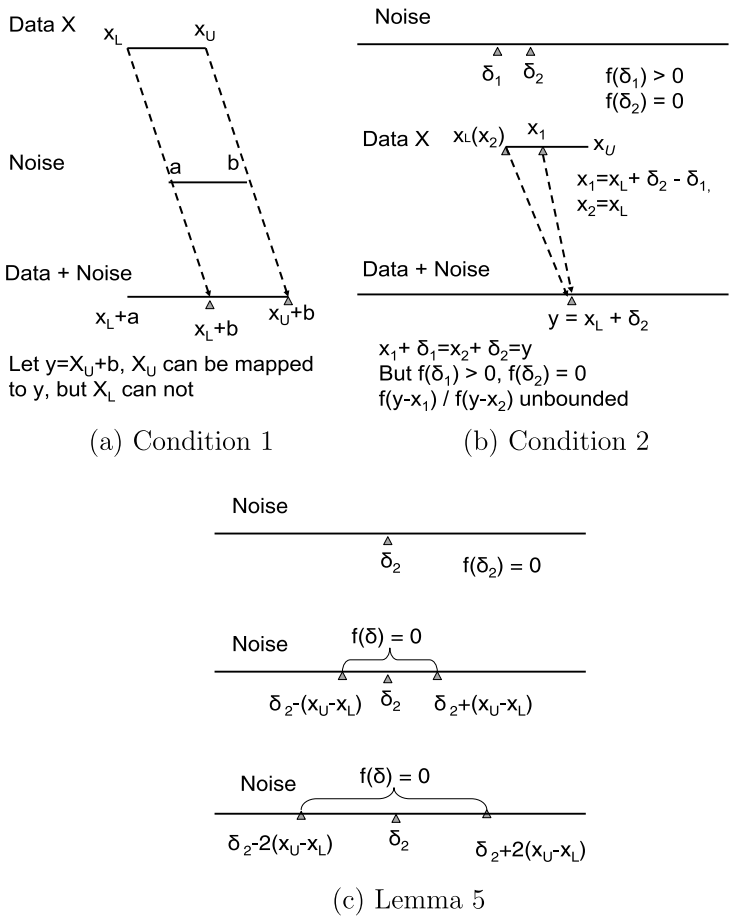


Fig. 1. Proof of necessity conditions. (a) Condition 1; (b) Condition 2; (c) Lemma 5.

4.3. Examination of commonly used noise distributions

Based on the necessary and sufficient conditions defined in previous section, this section investigates amplification for additive perturbation using some common density distributions considered in literature.

4.3.1. Uniform noise

In this case the support of the distribution is not infinite. Thus Condition 1 is violated leading to unbounded amplification.

Example 4.1. consider a data set with an attribute X, which varies from 0 to 1. Suppose uniform noise in the range of 0–1 is added. Let a property of X be $X \geqslant 0.99$. If an attacker observes a perturbed value of 1.99, he is 100% sure that the original value must satisfy this property (otherwise the perturbed value will be less than 1.99). However, the probability of this property in the original data may be very low. Thus an upward privacy breach occurs.

4.3.2. Normal noise

Let the density distribution of noise be $N(0,\sigma)$. In this case f_Δ is defined in $-\infty < \delta < +\infty$ and is not 0 for any value in the support domain. Thus Conditions 1 and 2 are both satisfied. However

$$A = \frac{f_\Delta(y-x_1)}{f_\Delta(y-x_2)} = e^{\frac{\{(y-x_2)^2-(y-x_1)^2\}}{2\sigma^2}} = e^{\frac{\{(2y-x_2-x_1)(x_1-x_2)\}}{2\sigma^2}}.$$

The above equation is unbounded for either of $y \to \pm\infty$ depending on the sign of $(x_1 - x_2)$. Thus Condition 3 is violated and there is no bounded amplification for additive perturbation with normal noise. Below is an example of privacy breach when normal noise is added.

Example 4.2. consider a data set with one attribute X with following distribution:

$$X = \begin{cases} 0 & \text{with probability } 0.99, \\ 1 & \text{with probability } 0.01. \end{cases}$$

Now suppose a normal noise with zero mean and unit variance is added to the data. Let Y be the perturbed attribute. Let property $Q(X)$ be $X = 1$. $P(Q(X)) = 0.01$. Suppose an attacker observes a large perturbed value of y (e.g., $y = 1000$), then

$$P(X = 1|Y = y) = \frac{P(X = 1, Y = y)}{P(Y = y)}.$$

Since the noise is added independent of the value of X, we have

$$P(X = 1|Y = y) = \frac{P(X = 1)P(1 \to y)}{P(Y = y)} = 0.01\frac{P(1 \to y)}{P(Y = y)}. \tag{4}$$

Similarly, we have

$$P(X = 0|Y = y) = \frac{P(X = 0, Y = y)}{P(Y = y)} = \frac{P(X = 0)P(0 \to y)}{P(Y = y)} = 0.99\frac{P(0 \to y)}{P(Y = y)}. \tag{5}$$

Divide Eq. (4) by Eq. (5), we have

$$\frac{P(X = 1|Y = y)}{P(X = 0|Y = y)} = \frac{P(1 \to y)}{99P(0 \to y)}.$$

Let $\gamma = \frac{P(1\to y)}{P(0\to y)}$. Since $P(X = 0|Y = y) = 1 - P(X = 1|Y = y)$, the above equation can be rewritten as

$$\frac{P(X = 1|Y = y)}{1 - P(X = 1|Y = y)} = \frac{\gamma}{99}.$$

$P(X = 1|Y = y)$ is the only variable in the equation. Solving this equation will get

$$P(X = 1|Y = y) = \frac{\gamma}{99 + \gamma}. \tag{6}$$

Next we compute γ. $P(1 \to y)$ is the probability of mapping 1 to y, i.e., the probability of noise being $y - 1$. Thus $P(1 \to y) = e^{\frac{-(y-1)^2}{2}}$. Similarly, $P(0 \to y) = e^{\frac{-y^2}{2}}$. Thus

$$\gamma = e^{\frac{y^2-(y-1)^2}{2}} = e^{\frac{2y-1}{2}},$$

γ becomes infinity as y goes to infinity. According to Eq. (6), $P(X = 1|Y = y)$ will increase and converge to 1. Note that $P(X = 1) = 0.01$, thus an upward privacy breach occurs and the attacker can infer that the value of X is 1 with high probability if he sees a large Y value in perturbed data.

4.3.3. Laplace noise

$\left(f(\delta) = \frac{1}{2b}e^{-|\delta-\mu|/b}\right)$. It will be shown that for this distribution, amplification is finite. Clearly Condition 1 and 2 are satisfied because the density function is supported at $-\infty < \delta < +\infty$ and the density function is always positive. Now consider a Laplace distribution with *location* parameter $\mu = 0$ and *scale* parameter b (amplification is still bounded if $\mu \neq 0$, however this paper consider the $\mu = 0$ case just for simplicity of mathematical calculation)

$$A = \frac{f_\Delta(y - x_1)}{f_\Delta(y - x_2)} = e^{\frac{\{|y-x_2|-|y-x_1|\}}{b}}.$$

There are four possible cases.

(a) $x_1 \leqslant y \leqslant x_2$

$$A = e^{\frac{\{|y-x_2|-|y-x_1|\}}{b}} = e^{\frac{\{x_2+x_1-2y\}}{b}},$$

$$\max_{x_1,x_2,y} A = \max_{x_1,x_2} e^{\frac{\{x_2+x_1-2x_1\}}{b}} = e^{\frac{\{\max(x_2)-\min(x_1)\}}{b}} = e^{\frac{\{\max(x)-\min(x)\}}{b}}.$$

(b) $x_2 \leqslant y \leqslant x_1$.
A similar calculation as *case (a)* leads to

$$\max_{x_1,x_2,y} A = e^{\frac{\{\max(x)-\min(x)\}}{b}}.$$

(c) $x_2, x_1 \leqslant y$

$$A = e^{\frac{\{|y-x_2|-|y-x_1|\}}{b}} = e^{\frac{\{x_1-x_2\}}{b}},$$

$$\max_{x_1,x_2,y} A = e^{\frac{\{\max(x_1)-\min(x_2)\}}{b}} = e^{\frac{\{\max(x)-\min(x)\}}{b}}.$$

(d) $x_2, x_1 \geqslant y$.
A similar calculation as case (c) leads to

$$\max_{x_1, x_2, y} A = e^{\frac{\{\max(x) - \min(x)\}}{b}}.$$

Thus in all cases amplification is bounded by the support bounds of data distribution and has the value of

$$\text{Amplification} = e^{\frac{\{\max(x) - \min(x)\}}{b}} \tag{7}$$

If Laplace noise is added to the data set in Example 4.2, a worst case privacy guarantee will be provided. The reasoning is the same as in Example 4.2 until we reach Eq. (6)

$$P(X = 1 | Y = y) = \frac{\gamma}{99 + \gamma}.$$

Here $\gamma = \frac{P(1 \to y)}{P(0 \to y)}$. Since Laplace noise is added, $P(1 \to y) = \frac{1}{2b} e^{-(y-1)/b}$, and $P(0 \to y) = \frac{1}{2b} e^{-y/b}$. Thus $\gamma = e^{(y-y+1)/b} = e^{1/b}$. Note that when normal noise is added, the value of γ tends to infinity if y tends to infinity. But for Laplace noise, γ is bounded as y tends to infinity and this will limit the probability ρ_2 (in this case, $P(X = 1 | Y = y)$). For example, if a Laplace noise with $b = 0.5$ is added, $\gamma = e^{1/0.5} = 7.4$, and $P(X = 1 | Y = y) = 0.069$ using Eq. (6). More generally, if $\rho_1 = 0.01$, it is guaranteed that no upward privacy breach with ρ_2 over 0.069 will occur in this case.

4.3.4. Random projection method

Although not directly related to additive perturbations, it is interesting to see why random projection method does not provide worst case privacy. Let D denote the dataset and $\vec{x_i} = (x_{i1}, x_{i2}, \ldots, x_{in})$ be a row in the data. Further, let $\vec{r_j} = (r_{1j}, r_{2j}, \ldots, r_{nj})$ be the jth column vector in the random matrix. Let $\vec{y}$ be the data vector mapped from $\vec{x_i}$ in the sanitized data. Then the jth attribute of $\vec{y}$ equals the inner product of $\vec{r_j}$ and $\vec{x_i}$.

Example 4.3. Consider a very sparse data set where most data points are all-zero vectors. Let a property $Q(\vec{x_i})$ be '$\vec{x_i}$ *is not an all-zero row vector*'. Probability of this property is very low in the original data. An important observation is that if $\vec{x_i}$ is an all zero vector, it will be projected to an all zero vector in the sanitized data because the inner product of an all zero vector and any $\vec{r_j}$ equals 0. Now if the attacker observes a sanitized data vector $\vec{y}$ which is not all zero, then he can infer that the original data vector must not be an all-zero row vector because otherwise it will be projected to an all zero vector. Thus the probability of $Q(\vec{x_i})$ given a non all zero vector in the sanitized data is one. An upward privacy breach has occurred in this case.

It can be verified that some other density distributions such as the Student's t-Distribution also gives bounded amplification. However, Laplace distribution will be used in this paper because it is easy to be dealt with mathematically.

5. Proposed method

This section describes the proposed method. The method consists of two steps. The first step is the data sanitization step and the second step is the data mining step. Section 5.1 describes the first step and Section 5.2 discusses privacy protection of this method. Section 5.3 describes the second step.

5.1. Data sanitization

The overview of the proposed method is given in Fig. 2. Table 1 summarizes the symbols used in this algorithm. The method takes the original data set D as input. Here D is represented by a $n \times m$ matrix where n is the number of records and m is the number of attributes. A noise parameter b and the number of selected principal components s are also given at input. s is selected such that only principal components with large values are retained. b determines the magnitude of noise and in turn determines the amplification and degree of worst case privacy guarantees. A larger b leads to a higher degree of privacy but at the same time may lead to less accurate mining results. Section 6 will show that in the experiments conducted, a wide range of value s and b = 0.2–0.3 lead to still accurate mining results and at the same time provide reasonable degree of worst case privacy guarantees.

The method starts with taking the principal component transform of the dataset D to produce a set of principal component values D_P. Noise from a *Laplace* distribution with location parameter 0 and scale parameter b_i is then added individually to each of the elements of the first s columns of D_P to produce D_{NP}. To scale the noise properly to suit the magnitude of values of the PC values, b_i is set to $b \cdot (\max\{D_P^i\} - \min\{D_P^i\})$. Section 5.2 will show this will lead to uniform amplification for all columns. D_P^i represents the ith principal component values (i.e., the ith column of D_P) (Table 2).

The first s columns of the noisy principal component values (called PCs) are then sent to the party asking for the data to be used as training set.

Complexity analysis: The noise addition process (steps 2–10) takes time $O(mns)$. PCA takes $O(m^2 n)$ if exact results are needed. However there exists efficient algorithms that compute approximate PCA [43]. The complexity of such algorithms is $O(mns)$. Thus the complexity of the proposed data sanitization method is $O(mns)$.

```
Sanitize(Dataset D,Noise parameter b, Number of selected coefficients s)
(1) D_P ← PCA(D)
(2) for each column i of D_P
(3)   b_i = b · (max{D_P^i} − min{D_P^i})
(4) end
(5) for i = 1 to s
(6)   for each record j
(7)     D_NP(j,i) = D_P(j,i) + n(b_i)
/* n(b_i) ~ Laplace(0, b_i) */
(8)   end
(9) end
(10) return D_NP
```

Fig. 2. Pseudo-code of the proposed method.

Table 1
Symbols

D	Dataset
n	Number of rows
m	Number of attributes
D_P	Data after PCA (principal component matrix)
D_P^i	The ith principal component values (ith column of D_P)
D_{NP}	Sanitized data after PCA and noise addition (the results of adding noises to D_P)
s	Number of principal components selected (appeared in D_{NP})
b	Scale parameter for Laplace Noise
b_i	Scale parameter for the ith principal component

Table 2
Properties of datasets

Data set	Number of attributes	Number of records	Number of classes
Wine	13	178	3
Pendigits	16	7494	10
Wisconsin diagnostic breast cancer	30	569	2
Magic04	10	19,020	2
Ionosphere	34	351	2
Iris	4	150	3
Waveform	21	5000	3
Synthetic	100	100,000	10

5.2. Privacy of proposed method

The proposed method combines PCA with additive perturbation. PCA completely decorrelates the data thereby guards additive perturbation against the typical noise filtering attacks based on data correlations [25]. The additive perturbation step guards against attacks that can recover the PCA transform (e.g., the attacking methods described in [35]) because such attacks cannot remove the added noise.

Next we compute the amplification for the proposed method. The result will show that the proposed method also provides worst case privacy guarantee for ρ_1 to ρ_2 privacy breach.

One could certainly compute amplification using the original data values. However this paper computes the amplification using the principal component values, not the original data values. The reason is that in the worst case, attackers can find out the PCA transform (e.g., if the attacker learns the covariance matrix of original data) and will be able to infer the original data values from the principal component values.

To compute amplification for the combined framework, let $\vec{a_i} = (a_{i1}, \ldots, a_{in})$ denote the ith eigen-vector of the covariance matrix and $\vec{x_1} = (x_{11}, x_{12}, \ldots, x_{1n})$ and $\vec{x_2} = (x_{21}, x_{22}, \ldots, x_{2n})$ be two arbitrary row vectors in the original data. Also let X_1 and X_2 denote the ith PC scores to which the two data vectors are mapped respectively (i.e., X_1 and X_2 equal the dot product of $\vec{a_i}$ with $\vec{x_1}$ and $\vec{x_2}$, respectively). Let D_P^i and D_{NP}^i denote the ith column of D_P and D_{NP}, respectively. Using the results from Eq. (3), amplification for the ith PC can be computed directly over these principal component values as follows:

$$A_{\max} = \max_{X_1, X_2, y} \lim_{dx \to 0} \frac{P([X_1, X_1 + dx] \to y)/dx}{P([X_2, X_2 + dx] \to y)/dx} \quad \forall X_1, X_2 \in D_P^i,\ y \in D_{NP}^i$$
$$= \max_{X_1, X_2, y} \frac{f_\Delta(y - X_1)}{f_\Delta(y - X_2)} \quad \text{by Eq. (3)}.$$

If the noise follows Laplace distribution with zero mean, using Eq. (7) which computes the amplification for this kind of noise, the amplification equals

$$A_{\max} = e^{\frac{\{\max(D_P^i) - \min(D_P^i)\}}{b_i}} = e^{\frac{\{\max(D_P^i) - \min(D_P^i)\}}{b \cdot \{\max(D_P^i) - \min(D_P^i)\}}} = e^{1/b}, \tag{8}$$

where b_i is the scale parameter of the *Laplace* noise added to the ith PC scores. The bounds of D_P^i are computed from the PC scores of the data. Note that b_i is proportional to the range of the ith principal components, thus the amplification for each column is the same and dependent only on b. Further the PCs being independent, amplification is calculated on each column individually since one cannot infer values of a column from another column.

5.3. Modifications to K-nearest neighbor

This section first describes the problem introduced by distance distortion, and then proposes modifications to KNN to improve accuracy.

Problem of distance distortion: the major challenge remaining is that distances between data points are distorted after the additive perturbation step. Thus a nearest neighbor in the original data may not remain a nearest neighbor in the sanitized data (called *false negative* in this paper). Similarly, a point that is not a nearest neighbor in the original data may become a nearest neighbor in the sanitized data (called *false positive* in this paper). Typically, a *K*-nearest neighbor algorithm uses a small value of k to define the nearest neighbor search space. Thus the probability of false negative and false positive could be pretty high for a small k value. This may lead to a substantial drop of the accuracy of *K*-nearest neighbor algorithm.

For example, Fig. 3a shows the points in the original data having two well defined clusters while Fig. 3b shows the perturbed data points. Let points in class 1 be represented with crosses and points in class 2 be represented with circles. Now suppose *A* and *B* denote two data points in the test data falling in the two clusters, respectively. In the original data shown in Fig. 3a, the two circles around point *A* and *B* represent the nearest neighbor search space for KNN when $k = 5$. For point *A*, four of the five nearest neighbors belong to class 1. Thus KNN will correctly predict point *A* belong to class 1. Similarly, KNN will correctly predict point *B* belong to class 2. In the sanitized data shown in Fig. 3a, the two inner circles represent the nearest neighbor search space when $k = 5$. Now for point *A*, only 2 of the nearest neighbors still belong to class 1, thus KNN will predict point *A* belong to class 2, which is wrong. Similarly, KNN's prediction of point *B*'s class is also wrong. The accuracy of KNN drops because the distances between data points are distorted.

While in general, a bigger value of k makes the ordinary KNN classification more resistant to noise, the approach can be problematic in this case. First, an ordinary k-nearest neighbor algorithm with a big k value will pick up too many false positives in perturbed data, resulting in poor classification. Second, the usual way to determine the right k is through cross validation, which is extremely expensive.

There has also been work on locally adaptive KNN methods where the value k is selected locally (i.e., different k values may be used for different test cases) [54]. The selection of k for each test case also depends on some form of cross validation. For example, one such method stores for each training data point a list of values of k that correctly classify the training point under leave-one-out cross validation. When a test data point is classified, M nearest neighbors of the test point are computed, and that k which classifies correctly most of these M neighbors is used to classify the test data. However, [54] shows

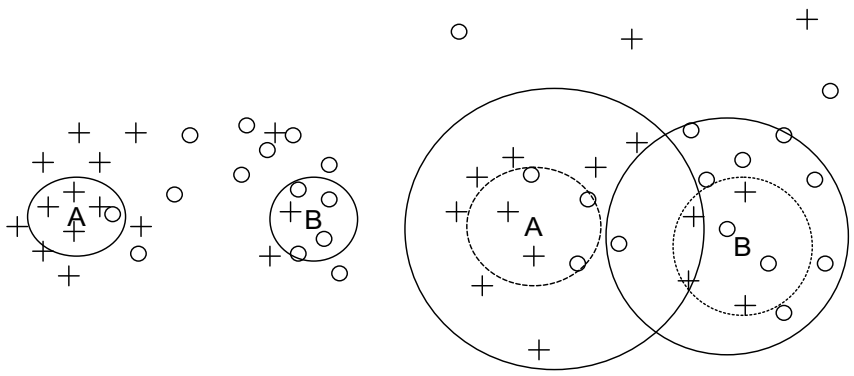


Fig. 3. *K*-nearest neighbor algorithm over original and sanitized data.

that there is no significant improvement using locally adaptive KNN methods for most commonly used data sets. Further, in our setting, the degree of noise (e.g., the scale parameter b of Laplace noise) is known, thus we can infer some properties of distance distortion introduced by noise. None of the existing methods considers such properties because in their settings, the statistics of noise is typically unknown. This paper investigates a few modifications to the basic KNN that take into account the properties of the added noises.

Modified KNN algorithm: the proposed algorithm uses two heuristics to improve the basic KNN algorithm over the sanitized data. The first heuristics tunes the search space for nearest neighbors according to the distance distortions. The second heuristics uses distance-based weighing when selecting the class label for the test data point. Fig. 4 shows the pseudo-code of the proposed algorithm. The algorithm assumes the data sender has computed the mean and variance of distance distortion. The detail of computation will be described in Section 5.3.1. In lines (1)–(3), distances from the test point t to training data are computed.

Adjusting nearest neighbor search space based on distance distortions: the new search space is a circle around the test data point Y where the radius is computed based on mean and variance of distance distortion. The circle has the property that most scattered data points will fall in this circle, thus false negative points will be included in the circle. The two outer circles in Fig. 3b are the modified search space based on distance distortion statistics. It is clear that now KNN will give accurate prediction.

Now we explain how to compute the radius of the search circle. Let d^2 be the initial squared distance between a test point Y and one of its neighbors X in the training set. Also let d'^2 denote the square of the distance between Y and X' where X' is distorted X. Let distance distortion $\widehat{D}$ be $d'^2 - d^2$. Let $\mu_{\widehat{D}}$ and $\sigma^2_{\widehat{D}}$ be the *mean* and *variance* for $\widehat{D}$, respectively, then Chebychev's inequality [53] for n standard deviations from *mean* gives

$$P(\widehat{D} \leqslant \mu_{\widehat{D}} + n\sigma_{\widehat{D}}) \geqslant P(\mu_{\widehat{D}} - n\sigma_{\widehat{D}} \leqslant \widehat{D} \leqslant \mu_{\widehat{D}} + n\sigma_{\widehat{D}}) \geqslant 1 - \frac{1}{n^2}. \tag{9}$$

If $\widehat{D}$ can be further assumed to have unimodal distribution, then Vysochanskij Petunin's inequality [53] gives

$$P(\widehat{D} \leqslant \mu_{\widehat{D}} + n\sigma_{\widehat{D}}) \geqslant P(\mu_{\widehat{D}} - n\sigma_{\widehat{D}} \leqslant \widehat{D} \leqslant \mu_{\widehat{D}} + n\sigma_{\widehat{D}}) \geqslant 1 - \frac{4}{9n^2} \quad \forall n \geqslant \sqrt{8/3}. \tag{10}$$

For a well-behaved data set, distances between members of the same class should be negligible compared to distances between points of different classes. Thus ideally d^2 will have a small value compared to inter-class distances. With a comparatively negligible d^2, it can be assumed that the original nearest neighbors are scattered by the noise with high probability to a zone with distance no more than $\mu_{\widehat{D}} + n\sigma_{\widehat{D}}$. This becomes the new search zone for neighbors in the noisy training set.

$n = 2$ is used in this paper because the probability of $\widehat{D} \leqslant \mu_{\widehat{D}} + 2\sigma_{\widehat{D}}$ is greater than 0.75 from Eq. (9) and is greater than 0.9 from Eq. (10). Thus using the radius with $n = 2$ will include most nearest neighbors in the original data, and thus greatly reduces false negatives.

Distance-based weighing: in lines (5)–(7), the algorithm uses a distance-based weighing scheme to reduce the impact of false positives. The search space based on distance distortion is typically larger than the search space of the basic KNN algorithm to include more false negatives. However, this may increases chances of including more false positives. For example, in Fig. 3b, some of the points in the two outer circles (the new search space) are not nearest neighbors in the original data.

This paper solves the problem by a distance weighted voting scheme [39]. Let d_i be the square of distance of the training point to the testing point. Let *NN* be the set of points falling in the new search space. For a training data point l falling in the

```
newKNN(test tuple t, sanitized training set S,E(D̂),Var(D̂))
(1) for each record s_i of S
(2)   d_i = (distance(t, s_i))^2
 /* distance means Euclidean-distance */
(3) end
(4) add those s_i to NN whose d_i satisfy
      d_i ≤ E(D̂) + n√Var(D̂)
(5) for each record s_l in NN
(6)   w_l=1/d_l ∑_{i∈NN} 1/d_i
(7) end
(8) for each class C_i
(9)   W_total(C_i) = ∑ w_j, ∀j ∈ NN and labeled C_i
(10) end
(11) class label of t = arg max_{C_i}(W_total)
```

Fig. 4. Modified KNN for perturbed data.

search zone, it is assigned the following weight which is inversely proportional to the square of distance to the testing data point.

$$w_i = \frac{1}{d_l} \sum_{i \in NN} \frac{1}{d_i}. \tag{11}$$

Here $\sum_{i \in NN} \frac{1}{d_i}$ is a normalizing factor that makes sure the total weight $\sum_{i \in NN} w_i$ equals 1. This weight will be added to the vote for the class that the training data point belongs to. The rationale of the weighing scheme is that, in an originally well-behaved dataset, inter-class distances are generally greater than intra-class distances. Thus, if such a dataset is perturbed with additive noise, a point in the outer periphery of the search space are more likely to be a false positive because a false positive point is not a nearest neighbor in the original data and is likely to be far from the testing data point.

In Lines (8)–(10) of the algorithm, the class with the highest vote will then be selected as the class label for the testing data. In Line (11), the test point will be assigned the label of the class with the highest vote.

Complexity analysis: Let n be the number of records and s be the number of principal components selected in data sanitization step. Let c be the number of classes. For a test case, lines (1)–(3) take $O(ns)$ time to compute the distances between a test case to training cases. Line (4) needs to sort the data points by their distances to the test case. The sort cost is $O(n \log n)$. Lines (5)–(7) take $O(n)$ time because there are at most n nearest neighbors. Lines (8)–(11) take $O(c)$ time. The complexity of the modified KNN is $O(ns + n \log n)$. The mean and variance of the distance distortion can be computed in $O(mn)$ time because the computation just requires some statistics about the principal components (please see the next section for details). This computation only needs to be done once for all testing data points.

Further, there exist many techniques such as [7,3,56] to speed up KNN using auxiliary data structures. These methods do not require computation of distances between the test data point to all training data points. Instead, an index structure will be used to return a small set of training data points that are likely to be nearest neighbors of the test data point. Thus it is only necessary to compute the distances between the test data point to these data points. The method proposed in this paper can use such techniques as well. The complexity of the algorithm thus can be reduced to $O(n's + n' \log n')$ where n' is the number of data points returned using the index structure and $n' \ll n$ in general.

5.3.1. Calculating mean and variance of distance distortion

This section computes the mean and variance of distance distortion. Let x be a row vector in a training set (the data to be sanitized and shipped to another party). Let y be a row in the test set owned by the party who is trying to classify its instances based on the sanitized training set shipped to it. Let X and Y be the corresponding principal component score vectors. X_i and Y_i denote the ith component of the score vectors X and Y. Also assume that S denotes the set of principal components retained and additively perturbed and N, those that are neglected. Since PCA is an orthogonal transform and hence distance preserving, the original squared distance between x and y can be written as

$$d^2 = \sum_i (x_i - y_i)^2 = \sum_{i \in S} (X_i - Y_i)^2 + \sum_{i \in N} (X_i - Y_i)^2.$$

After retaining the coefficients in S only, each PC coefficient X_i in the training set is perturbed with a noise δ_i following *Laplace* distribution with *location parameter* 0 and *scale parameter* b_i. The squared distance between the row vectors turns out as

$$d'^2 = \sum_{i \in S} (X_i + \delta_i - Y_i)^2.$$

From the above two expressions, the distortion in squared Euclidean-distance can be calculated as

$$\widehat{D} = d'^2 - d^2 = \sum_{i \in S} 2(X_i - Y_i)\delta_i + \sum_{i \in S} \delta_i^2 - \sum_{i \in N} (X_i - Y_i)^2.$$

Let σ_i^2 denote the *variance* of the ith principal component score of the data. The mean and variance of squared distance distortion are given below. Please refer to Appendix for details:

$$E(\widehat{D}) = 2 \sum_{i \in S} b_i^2 - 2 \sum_{i \in N} \sigma_i^2, \tag{12}$$

$$\mathrm{Var}(\widehat{D}) = 16 \sum_{i \in S} b_i^2 \sigma_i^2 + 20 \sum_{i \in S} b_i^4 + 8 \sum_{i \in N} \sigma_i^4. \tag{13}$$

6. Experimental evaluation

This section presents experimental evaluation of the proposed method. Section 6.1 describes the setup of the experiments. Section 6.2 compares the proposed method with existing methods. Section 6.3 reports the privacy of the proposed method under correlation-based attacks [25] and transform-based attacks [35]. Section 6.4 reports the impact of modifications to KNN on accuracy.

6.1. 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 noise was generated randomly, all experiments were run 20 times and the average results were reported.

6.1.1. Datasets

The experiments were run over seven real datasets from the UCI Machine Learning Repository [24], and a synthetic dataset generated using a program from [15]. The synthetic data contained 10 clusters, each generated using a multi-dimensional Normal distribution. The cluster centers were uniform randomly selected between the range of $[-5, 5]$ on each dimension. The standard deviations were varied at 2 and 8. Table 1 reports the properties of these data sets. Ten percent of data was randomly selected as the testing data, and the rest was used as training data. The data values are also normalized to the range of $[0, 1]$.

6.1.2. Algorithms

The following algorithms were compared in the experiments.

(1) A PCA-radius algorithm was implemented based on the method described in Section 5. The search space is within $E(\widehat{D}) + 2\sqrt{(\mathrm{Var}(\widehat{D}))}$, where $\widehat{D}$ is the squared distance distortion.
(2) Rand-P: this is a transform-based method to protect privacy for distance-based mining [36]. The data (as a $n \times m$ matrix) is multiplied by a $m \times k$ normalized random matrix with values chosen randomly from normal distribution.
(3) Rand-U: this method adds randomly generated uniform noise to the original data. There is no PCA step. This is one of the widely used additive perturbation methods [5].
(4) Rand-N: this method is the same as Rand-U except that normal noise is added to the original data.

For Rand-P, Rand-U, and Rand-N, the distance weighed KNN with best k (found through cross validation) is used to evaluate the method.

6.1.3. Privacy measures

Three approaches have been proposed in the literature to measure privacy: the first using confidence interval [5], the second using information theory [4], and the third using amplification to measure worst case privacy breach as mentioned. However, the information theory approach is inappropriate because Euclidean distance is based on individual data values, while information theory only considers the distribution of values [58].

This paper uses amplification to measure the *worst case* privacy breach and uses the confidence interval to measure the *average* privacy. The confidence interval measure is as follows. If a transformed attribute x can be estimated with $c\%$ confidence in the interval $[x_1, x_2]$, then the privacy equals $\frac{x_2 - x_1}{x_U - x_L}$ where x_U is the maximal value of x and x_L is the minimal value. 95% confidence interval was used in experiments.

Average privacy is reported after applying all the attacking methods proposed in [35,**?**]. More specifically, since transform-based methods (Rand-P and the PCA step of our method) are subject to attacks that can recover the transform matrix, this paper assumes the attacker knows the transform matrix. In our method (PCA-Radius), this means the attacker can apply a reverse PCA. For Rand-P, this means that the attacker can reconstruct the data by multiplying the modified data with the pseudo inverse matrix of the random matrix.

Similarly, since additive perturbation methods (Rand-U, Rand-N, and the noise addition step of our method) are subject to attacks based on data correlations, this paper also assumes such attacks have been applied. The attacking method first applies a PCA on perturbed data, selects the first few principal components, and then reconstructs the original data from these components using a reverse PCA. The number of selected components is selected such that the recovered data is closest to the original data.

6.2. Comparison of the proposed method with other methods

This section compares the performance of various privacy protection methods. These methods all have some parameters that determine the degree of noise added. These parameters are summarized below.

- *PCA-Radius*: the scale b in the Laplace distribution determines the degree of noise being added. We set $b = 0.1$, 0.2, and 0.3 in the experiments. As b increases, the degree of noise increases. Thus the degree of privacy increases but the accuracy of mining decreases. Another parameter is the number of principal components to select. Although there are many possible choices, a wide range of values give similar results. Thus this number is set to half of the total number of attributes (excluding the class label) in the data set.
- *Rand-P*: an important parameter is the number of columns (k) in the random matrix. Let m be the total number of attributes in the data set (excluding the class label), k is set to $m/4, m/2$, and $m - 1$ (note that k must be less than m). As k increases, the degree of privacy decreases but the accuracy of mining increases.

- *Rand-U and Rand-N*: the two parameters that determine the degree of noise is the mean and standard deviation of the noise added. The mean of noise is set to zero and the standard deviation is set to 0.1, 0.25, and 0.5. As the standard deviation increases, the degree of noise increases. Thus the degree of privacy increases but the accuracy of mining decreases.

Average privacy and accuracy tradeoff: there are two important metrics for a privacy protection method: the degree of privacy and the accuracy of mining methods. Thus Fig. 5a–i report both the average privacy and the accuracy of running KNN mining algorithm on sanitized data. For synthetic data set, σ represents the standard deviation used to generate cluster centers. The x-axis is average privacy and the y-axis is accuracy of mining. PCA-Radius uses the modified KNN algorithm described in Section 5.3. Other methods use the basic format of KNN, and use cross validation to find the best k value.

Each method has three different settings described above, thus there are three data points for each method. For each method, the right most data point (the one with highest average privacy) is the one with the parameter that generates the highest degree of noise. For example, in Fig. 5a, the right most point for PCA-Radius has $b = 0.3$. The middle point of PCA-Radius has $b = 0.2$ and the left most point of PCA-Radius has $b = 0.1$.

When we compare the performance of two methods A and B, we can draw a line along the data points of method A using linear regression. If the data points of method B lie to the lower and left to the line of method A (or line of method A is at the right and upper side of points of method B), then method A has higher accuracy and higher average privacy than method B. Thus method A has better tradeoff of accuracy and average privacy than method B.

Lines for proposed method (PCA-Radius) are drawn in these figures. The results show that almost all data points of other methods lie at the lower and left side of the line of PCA-Radius. This shows that PCA-Radius either has higher average degree of privacy or higher accuracy than the other methods. Thus PCA-Radius has similar and often better tradeoff of accuracy and average privacy than the other privacy protection methods.

There are two exceptions for Pendigits (Rand-P with $k = m/4$ in Fig. 5b) and Cancer (Rand-N with the standard deviation of noise equals 0.1 in Fig. 5c). However, the average privacy of these two points is much lower than the average privacy of PCA-Radius. Thus it is unlikely the user will use the two settings due to insufficient privacy protection.

For each method, the results also show that as more noise is added, average privacy increases but accuracy of mining decreases. For our method, when $b = 0.3$ (the right most point) and $b = 0.2$ (the second rightest point), the mining accuracy is still pretty high. As $b = 0.3$, the accuracy of our method is around 65–80% for real data sets and 100% for the two synthetic data sets. As $b = 0.2$, the accuracy of our method is around 70–85% for real data sets, and 100% for the two synthetic data sets. This shows that our method still allows accurate mining over sanitized data. In practice, user may decide the appropriate noise level by considering the accuracy-privacy tradeoff.

As $b = 0.3$, PCA-Radius also achieves high degree of average privacy. The average privacy exceeds 100% of the range of data (each column is normalized to 0–1) for 5 real data sets (Wine, Pendigits, Magic04, Ionosphere, Waveform) and the two synthetic data sets. The average privacy for the remaining two real data sets (Cancer and Iris) is also around 80% of the range of data. Note that the average privacy is computed after applying data transform attack [35] and correlation attack [25]. Thus the proposed method (PCA-Radius) gives adequate privacy protection under the average case.

Worst case privacy: as shown in Section 4.3, the proposed method (PCA-Radius) is the only method that provides worst case privacy for ρ_1 to ρ_2 privacy breach. Thus only the worst case privacy of proposed method is reported.

Fig. 6a reports the amplification for all data sets using PCA-Radius. Note that the amplification for all data sets are exactly the same because as shown in Eq. (8) in Section 5.2, they depend only on noise scaling parameter b. Thus one figure is used for all data sets. The results show that the amplification drops quickly (the closer the amplification value is to unity, the better the privacy) as noise increases, and is around 50 when $b = 0.25$ and 28 when $b = 0.3$.

Eq. (2) in Section 3.2 describes the relationship of amplification (γ) and the maximal value of ρ_2 in ρ_1 to ρ_2 privacy breach. Suppose ρ_1, the probability of a privacy sensitive property in the original data, is 0.1%. Fig. 6b reports the maximal possible value of ρ_2 (the probability of the privacy sensitive property given sanitized data) for various values of b. For example, for $b = 0.3$, the maximal possible value of ρ_2 is 2.8%. This means for $b = 0.3$, the probability of this privacy sensitive property will not exceed 2.8% given the sanitized data. Typically most privacy-sensitive data properties (can be the data values themselves) in real life appear in original data with low probabilities. The results show that the proposed method does not increase the conditional probabilities of such properties too much thus effectively restricting worst case privacy breaches.

6.3. Privacy under correlation-based and transform-based attacks

This section investigates the effect of correlation-based attacks [25] and transform-based attacks [35] on proposed method PCA-Radius. The focus is on the correlation-based attacks because it is assumed that attackers know the PCA transform. The basic assumption of the correlation-based attacking method is that after a PCA, data variances are concentrated on the first few principal components while noise is spread more or less evenly over all principal components. Thus simply reconstructing the data using the first few principal components can keep most of the data variances and at the same time filter most of the noise.

Fig. 7 shows the steps in PCA-Radius and the steps of applying correlation based attack along with the transform-based attack described in [35]. PCA-Radius sanitizes data by first applying a PCA, and then adds noise to the first s principal components (the remaining ones are discarded). The attacking process takes two steps. At the first step, the attacker discovers the transform matrix for PCA, and applies an inverse PCA over sanitized data to reconstruct the original data. Note that all

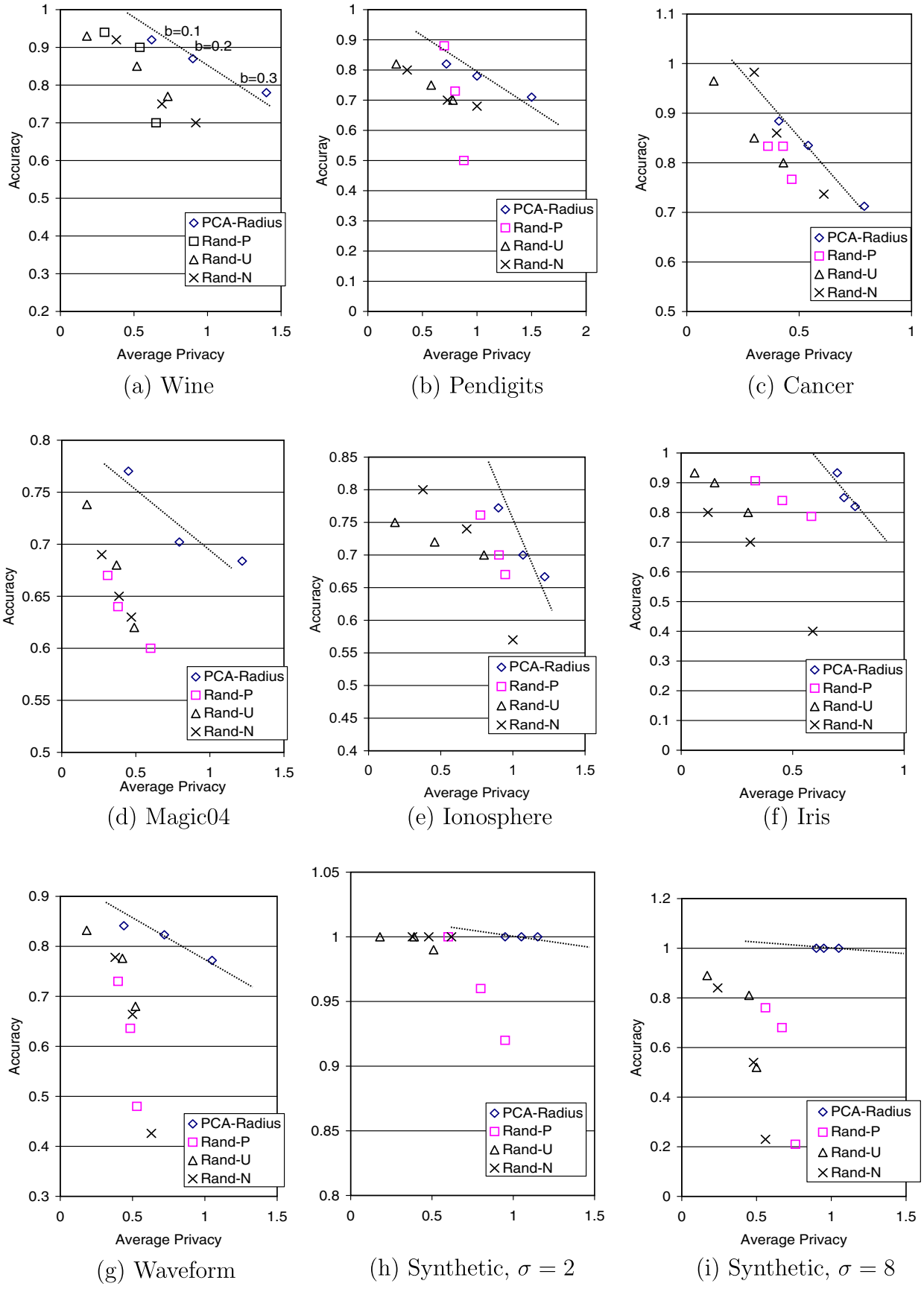
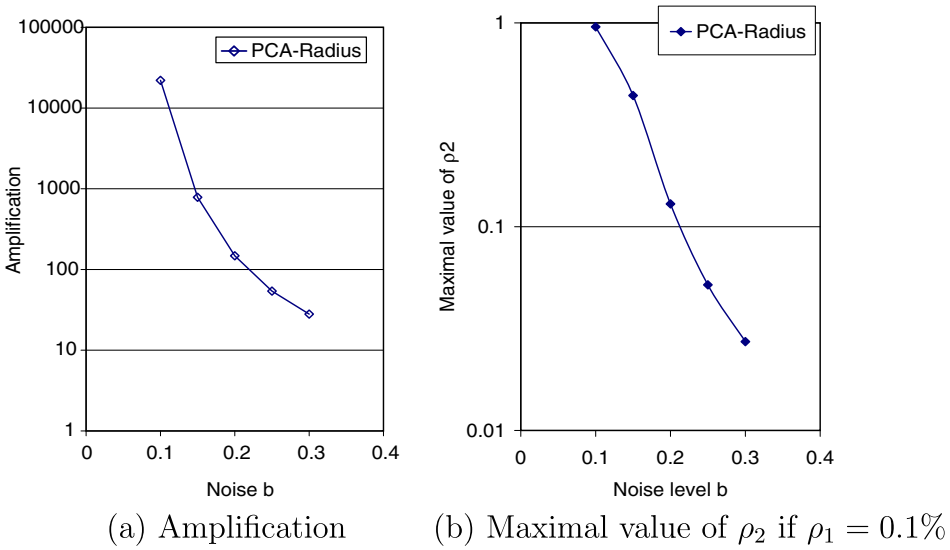


Fig. 5. Average privacy and accuracy of various methods.

columns in the sanitized data are used in reconstruction, thus this step does not filter out the noise added in the second stage of PCA-Radius. At the second step, the attacker applies the correlation based attack to filter out the added noise. This step



(a) Amplification (b) Maximal value of ρ_2 if $\rho_1 = 0.1\%$

Fig. 6. Worst case privacy for all data sets.

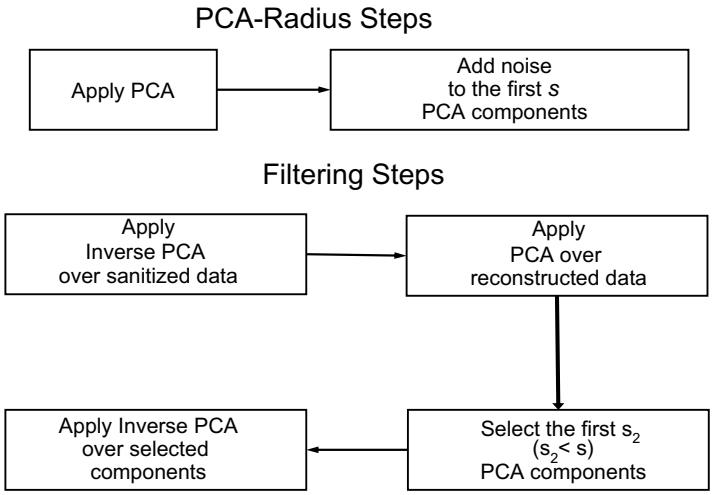


Fig. 7. Steps of PCA-Radius and filtering noise.

applies a PCA over the reconstructed data, and selects the first s_2 principal components, and uses them to reconstruct the original data using inverse PCA. s_2 must be less than s because we want to filter out noise added to the remaining $s - s_2$ principal components. s_2 is selected such that the reconstructed data is closest to the original data (i.e., with the lowest degree of privacy).

Three different scenarios are considered:

(1) *No-Filtering*: this is the average privacy computed over PCA-Radius without the correlation based attack. Transform-based attack [35] is still applied over sanitized data by a reverse PCA.
(2) *PCA + Filtering*: this is the average privacy computed over PCA-Radius with the correlation based attack and the transform-based attack described above. This is also the number reported in Section 6.2. Comparing the results of this scenario with the results of No-Filtering will indicate whether the noise filtering method can break privacy for PCA-Radius.
(3) *No-PCA + Filtering*: In this case, data is sanitized by adding Laplace noise directly to original data (not the principal components). The correlation based attack is then applied over the sanitized data. The results of this scenario shows that whether the correlation based attack will be effective if the data sanitization method does not use PCA to decorrelate the data.

Fig. 8a–i reports the average privacy for three different scenarios over all data sets. The results show that the average privacy of PCA + Filtering is very close to that of the No-Filtering. The biggest drop is about 10% and in most cases the drop is no more than 5%. This means that if the data is sanitized by PCA-Radius, the degree of privacy drop is negligible after

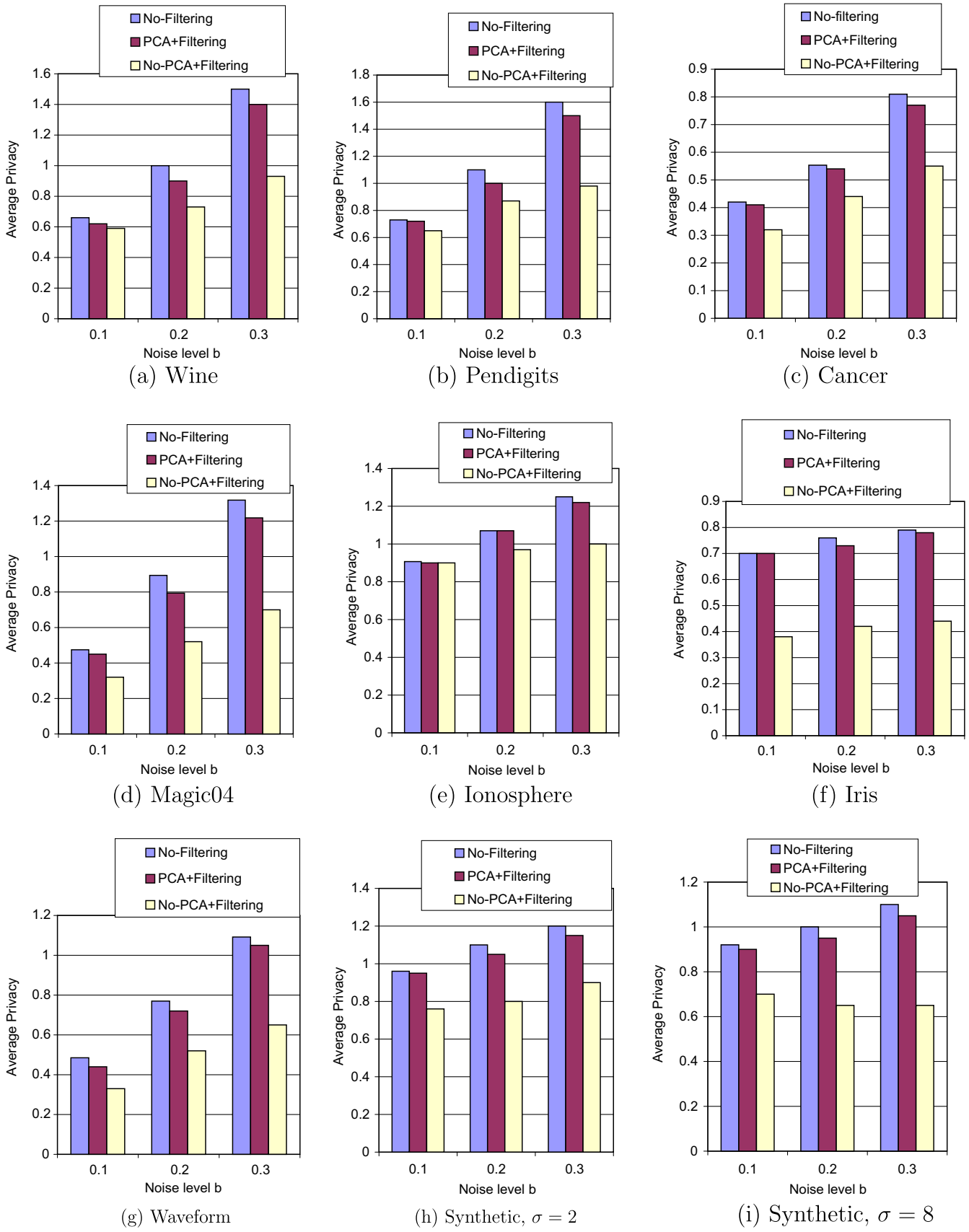


Fig. 8. Average privacy of PCA-Radius under correlation attack.

correlation-based attack. This is apparent because PCA + Radius applies PCA to decorrelate the data, thus the noise filtering method which uses data correlation is not effective. As described in Section 5.1, PCA-Radius also adds a larger amount of noise (with bigger b values) to principal components with large values. Thus the basic assumption of the noise filtering method that noise is distributed evenly to all principal components no longer holds.

The results also show that if the data sanitization step does not include a PCA (No-PCA + Filtering), the noise filtering method does significantly reduce the average degree of privacy. The drop is more than 20% for more than half of all cases, especially when noise level is high ($b = 0.2$ or 0.3). The drop is also always larger than the difference between PCA + Filtering and No-Filtering. This is expected because without a PCA step, the data is correlated and the noise filtering method can actually filter out a large amount of noise based on data correlations. Therefore, the results show that our proposed method (PCA-Radius) is not vulnerable to correlation-based attack and the protection is due to the PCA step.

6.4. Impact of modifications to KNN

This section investigates the impact of proposed modifications to KNN in Section 5.3 on the accuracy of KNN miming. The proposed method (PCA-Radius) is compared with two methods below:

(1) *PCA-Best-K*: this algorithm is the same as PCA-Radius in the PCA and noise addition step, but uses cross-validation to find an optimal k values in KNN. This k value is used for all test cases.
(2) *PCA-K* = 5: this algorithm is the same as PCA-Radius in the PCA and noise addition step, but uses a fixed k value of 5.

All three methods use distance-weighted KNN because it gives higher accuracies than using normal KNN in all the conducted experiments. Fig. 9a–i report the accuracy of classification for the three methods. The results on original data (i.e., $b = 0$ and no noise is added) are also shown as a baseline. As $b = 0$, no noise is added, thus PCA-Radius uses $k = 5$ in this case.

The results show that PCA-Radius has achieved accuracy slightly lower than using the optimal k. The drop in accuracy is less than 5% in all cases and is less than 2% for most cases. This demonstrates that using a distance radius inferred from the degree of noise achieves similar accuracy as using more expensive cross validation to find the optimal k value.

There are also a few cases when the accuracy of PCA-Radius exceeds the accuracy using the optimal k value for Waveform and Ionosphere data sets. The reason is that PCA-Radius uses a distance radius to identify nearest neighbors. This may include different number of nearest neighbors for different test cases. On the other hand, the k value selected using cross validation is used for all test cases. Thus in some cases PCA-Radius may outperform using the same k value selected by cross validation for all test cases.

The results also show that fixing $k = 5$ (PCA-K = 5) leads to much lower accuracy than PCA-Radius, especially when a larger amount of noise is added ($b = 0.2$ or 0.3). The drop in accuracy when compared to the PCA-Radius exceeds 5% for many cases, and exceeds 10% for some cases. For example, the drop is 18% and 21% for Waveform data when $b = 0.2$ and 0.3, and is 30% and 45% for Synthetic data with standard deviation equals 8. This is because distance gets distorted after noise addition and many of the nearest neighbors in the original data for $k = 5$ are no longer the nearest neighbors after noise addition. PCA-Radius avoids this problem by adjusting the radius of search space based on noise distortion.

6.5. Execution time

This section reports the execution time of proposed method (PCA-Radius). We vary the number of attributes and the number of records and use synthetic data with standard deviation of 2 in this experiment. The execution time can be also divided into the time to sanitize the data and the time of mining.

Fig. 10a and b report the time of sanitizing data (the PCA and noise addition step) of PCA-Radius. In Fig. 10a, we fix the number of records at 100,000, and vary the number of attributes from 20 to 100. In Fig. 10b, we fix the number of attributes at 100 and vary the number of records from 20,000 to 100,000. The time of sanitization is broken down into the time to load the data into main memory and save the sanitized data to a file, the time to apply principal component analysis, and the time to add Laplace noise. The results show that the time of data sanitization increases about linearly with the number of records and the number of attributes. The time to do PCA and add noise is also quite small compared to the time to load and save data.

We also examine the average execution time of KNN mining for each test case. The time for our method (PCA-Radius with modifications to KNN) is compared to the time for the basic KNN with $k = 5$. The noise level b is set to 0.3. Fig. 11a and b report the average mining time per test case. In Fig. 11a, we fix the number of records at 100,000, and vary the number of attributes from 20 to 100. In Fig. 11b, we fix the number of attributes at 100 and vary the number of records from 20,000 to 100,000.

The results show that the mining time of PCA-Radius scales about linearly with the number of records and the number of attributes. It is higher than the mining time of basic KNN with $k = 5$ because PCA-Radius needs to examine more nearest neighbors than basic KNN with $k = 5$ to avoid false negatives. However, the mining time of PCA-Radius is still quite short, taking about 2.6 s for the largest data set (with 100,000 records and 100 attributes). The results in Section 6.4 have shown

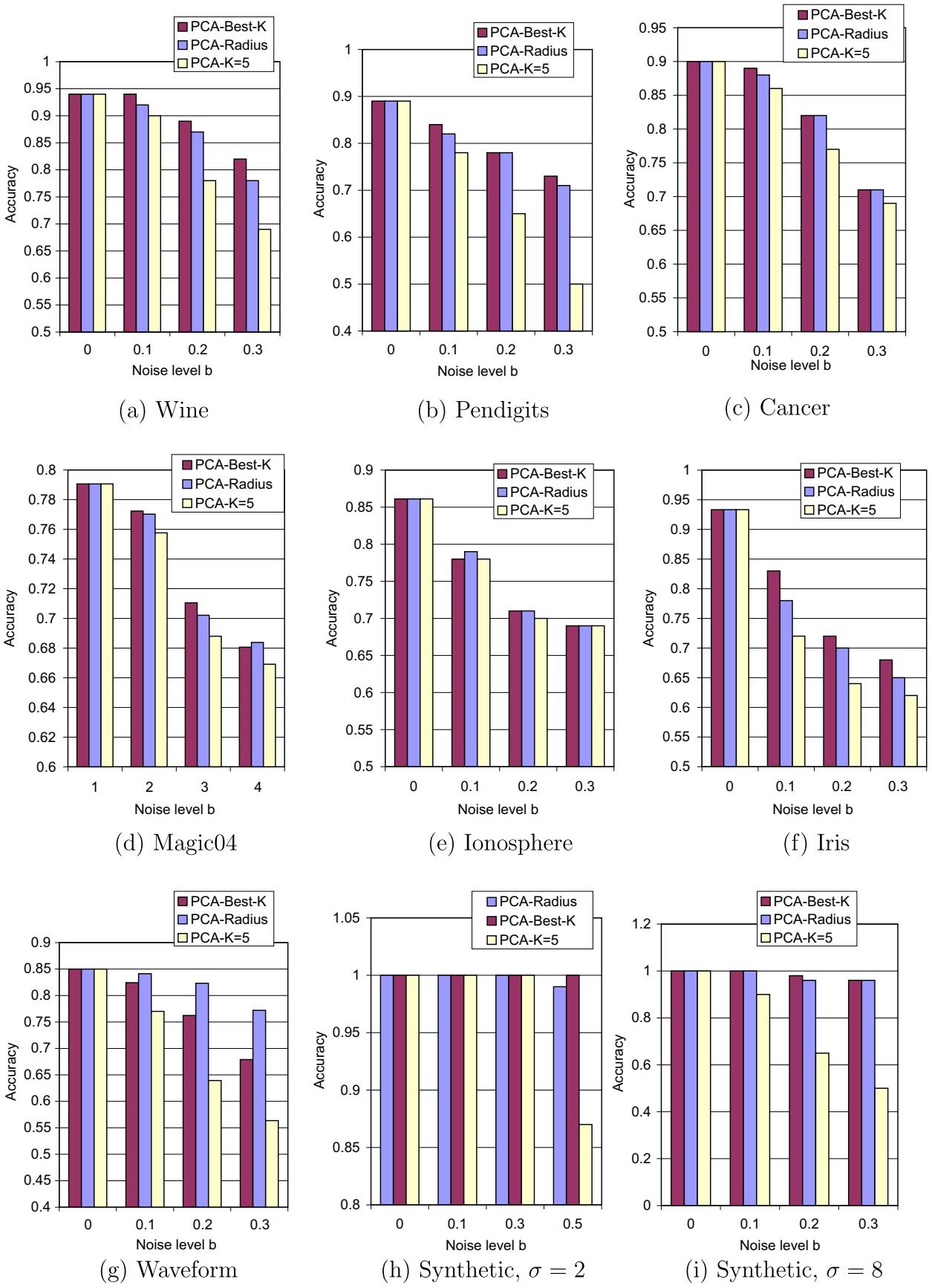
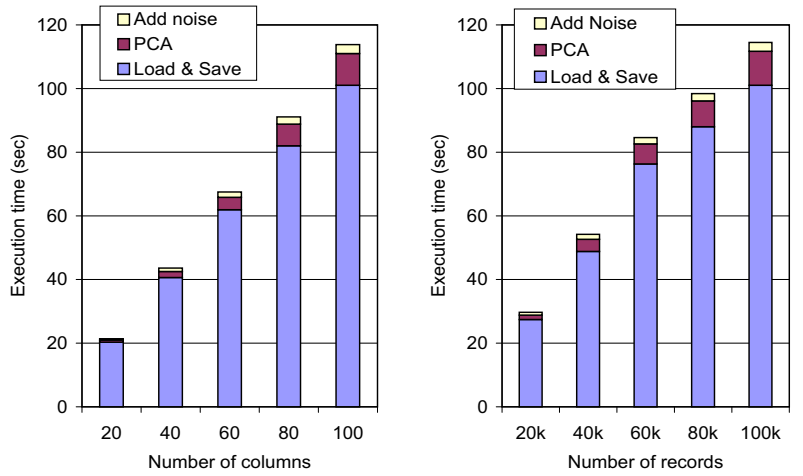
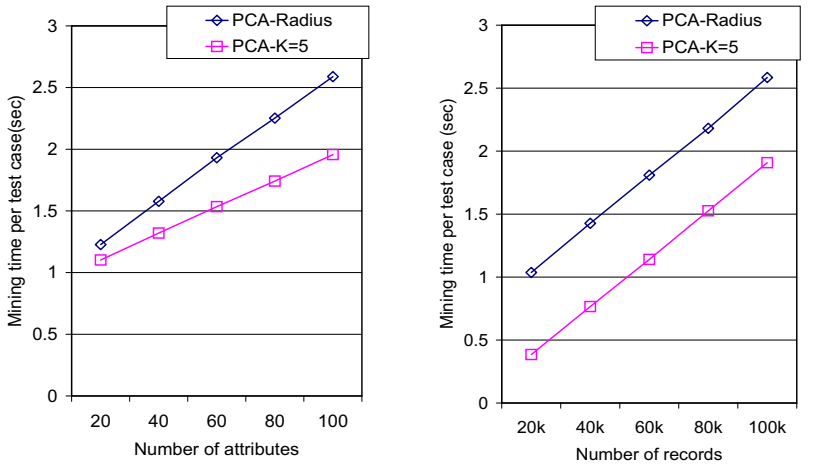


Fig. 9. Classification accuracy for real data sets using three KNN variants.



(a) Vary number of attributes (b) Vary number of records

Fig. 10. Data sanitization time.



(a) Vary number of attributes (b) Vary number of records

Fig. 11. Mining time per test case.

that the accuracy of PCA-Radius is much better than the accuracy of using $k = 5$, thus the extra time spent by PCA-Radius is justified. We may further reduce the mining time using techniques that use additional index structures [7].

7. Conclusion and future work

This paper proposes a method to provide worst case privacy guarantees for distance-based mining algorithm. It extends the worst case privacy measure – amplification – to continuous data and additive perturbation, and proposes a set of necessary and sufficient conditions for a noise distribution to give worst case privacy guarantees. It finds out that both uniform and normal distributions, as well as the random projection method, do not provide worst case privacy guarantees. On the other hand, noise following Laplace distribution does give worst case guarantee. This paper then proposes a method that combines noise addition and transform methods to protect privacy for distance-based mining. It also proposes a method to adjust the nearest neighbor search space based on the degree of noise added. Experimental results show that the proposed method provides worst case privacy guarantee, and guards against correlation-based and transform-based attacks. Interestingly, the results also show that the proposed method provides better or similar tradeoff between accuracy and average case privacy compared to existing methods. Experimental results also demonstrate that dynamically adjusting nearest neighbor search space achieves similar accuracy as using cross validation to find the optimal value of k.

Our future research plan is to study similar techniques to improve accuracy over other mining methods such as K-Means clustering.

Appendix A. Computation of mean and variance of distance distortion

A.1. Computation of mean distortion

As shown in Section 5.3.1, the distortion of square distances equals

$$\widehat{D} = d'^2 - d^2 = \sum_{i\in S} 2(X_i - Y_i)\delta_i + \sum_{i\in S}\delta_i^2 - \sum_{i\in N}(X_i - Y_i)^2.$$

Here X_i and Y_i denote the ith component for two data vector X and Y. S is the set of selected components and N is the set of pruned components. δ_i is a random variable representing the noise added to the ith component. δ_i follows a Laplace distribution with location parameter 0 and scale parameter b_i.

To compute the mean of distance distortion, the mean of δ_i, δ_i^2, and δ_i^4 need to be computed. This can be computed using the *moment generating function* of Laplace distribution. The MGF of a Laplace distribution with location parameter 0 and scale parameter b_i is given by $M(t) = \frac{1}{1-b_i^2t^2}$ [48]. Differentiating M with respect to t successively and plugging in $t = 0$ gives the following moments:

$$E(\delta_i) = 0,$$
$$E(\delta_i^2) = 2b_i^2,$$
$$E(\delta_i^4) = 24b_i^4.$$

Note that X_i and Y_i can be thought of being two *i.i.d* random variables following the same distribution as the ith principal component scores of the data. The mean of X_i and Y_i are 0 due to the PC transform. Let σ_i^2 denote the *variance* of the ith principal component score of the data ($\sigma_i^2 = \lambda_i$, the ith eigen value of the covariance matrix). Hence

$$E(X_i^2) = E(Y_i^2) = \sigma_i^2.$$

Further, all X_i and X_j pairs (and Y_i and Y_j pairs as well) are perfectly uncorrelated for $i \neq j$ because of the nature of PCA. Using the above stated facts and the expression for $\widehat{D}$ developed earlier, the next few lines calculate the *mean* of $\widehat{D}$.

$$E(\widehat{D}) = E\left\{\sum_{i\in S} 2(X_i - Y_i)\delta_i + \sum_{i\in S}\delta_i^2 - \sum_{i\in N}(X_i - Y_i)^2\right\} = E\left\{\sum_{i\in S} 2X_i\delta_i\right\} - E\left\{\sum_{i\in S} 2Y_i\delta_i\right\} + E\left\{\sum_{i\in S}\delta_i^2\right\} - E\left\{\sum_{i\in N}(X_i - Y_i)^2\right\}.$$

Removing the zero terms, the *expected* value of $\widehat{D}$ is as

$$E(\widehat{D}) = \sum_{i\in S} E\{\delta_i^2\} - \sum_{i\in N} E\{X_i^2\} - \sum_{i\in N} E\{Y_i^2\} = 2\sum_{i\in S} b_i^2 - 2\sum_{i\in N}\sigma_i^2.$$

A.2. Computation of variance of distortion

Let

$$V_i = \mathrm{Var}\{2(X_i - Y_i)\delta_i + \delta_i^2\} \quad \forall i \in S,$$
$$V_i' = \mathrm{Var}\{(X_i - Y_i)^2\} \quad \forall i \in N.$$

Thus

$$\mathrm{Var}(\widehat{D}) = \sum_{i\in S} V_i + \sum_{i\in N} V_i'.$$

The covariance terms vanish due to independence of X_i, Y_i and δ_i for different i values.

$$\begin{aligned} V_i &= \mathrm{Var}\{2(X_i - Y_i)\delta_i + \delta_i^2\} \\ &= 4\mathrm{Var}\{X_i\delta_i\} + 4\mathrm{Var}\{Y_i\delta_i\} + \mathrm{Var}\{\delta_i^2\} - 2\mathrm{Cov}\{2X_i\delta_i, 2Y_i\delta_i\} + 2\mathrm{Cov}\{2X_i\delta_i, \delta_i^2\} - 2\mathrm{Cov}\{2Y_i\delta_i, \delta_i^2\}. \end{aligned}$$

Note that

$$\mathrm{Var}(\delta_i^2) = E(\delta_i^4) - [E(\delta_i^2)]^2 = 24b_i^4 - 4b_i^4 = 20b_i^4$$

and

$$\mathrm{Var}(X_i\delta_i) = E(X_i^2\delta_i^2) - \{E(X_i\delta_i)\}^2 = E(X_i^2)E(\delta_i^2) - 0 = 2b_i^2\sigma_i^2.$$

Similarly $\mathrm{Var}(Y_i\delta_i) = 2b_i^2\sigma_i^2$. Also note that

$$\begin{aligned}\mathrm{Cov}\{2X_i\delta_i, 2Y_i\delta_i\} &= E\{(2X_i\delta_i - 2\overline{X_i\delta_i})(2Y_i\delta_i - 2\overline{Y_i\delta_i})\}\\ &= E\{(2X_i\delta_i - 0)(2Y_i\delta_i - 0)\} = 4E(X_i)E(Y_i)E(\delta_i^2) = 0,\\ \mathrm{Cov}\{2X_i\delta_i, \delta_i^2\} &= E\{(2X_i\delta_i - 2\overline{X_i\delta_i})(\delta_i^2 - \overline{\delta_i^2})\} = E\{(2X_i\delta_i - 0)(\delta_i^2 - 2b_i^2)\}\\ &= 2E\{(X_i)(\delta_i^3 - 2b_i^2\delta_i)\} = 2E(X_i)E(\delta_i^3 - 2b_i^2\delta_i) = 0.\end{aligned}$$

Similarly $\mathrm{Cov}\{2Y_i\delta_i, \delta_i^2\} = 0$. Thus

$$V_i = 16b_i^2\sigma_i^2 + 20b_i^4.$$

The coefficients belonging to set N are small with mean 0 and very small standard deviations due to the nature of PCA. Thus, they are assumed to follow normal distributions with means 0 and small standard deviations σ_i. Now let V_i' denote $\mathrm{Var}(X_i - Y_i)^2$ where $i \in N$. Expanding the expression

$$\begin{aligned}V_i' &= \mathrm{Var}(X_i^2) + \mathrm{Var}(Y_i^2) + \mathrm{Var}(2X_iY_i) + 2\mathrm{Cov}(X_i^2, Y_i^2) - 2\mathrm{Cov}(X_i^2, 2X_iY_i) - 2\mathrm{Cov}(Y_i^2, 2X_iY_i),\\ \mathrm{Var}(X_iY_i) &= E(X_i^2Y_i^2) - E(X_iY_i)^2 = \sigma_i^4,\\ \mathrm{Cov}(X_i^2, 2X_iY_i) &= E(X_i^2 - \overline{X_i^2})(2X_iY_i - 0) = 2E(X_i^3 - \sigma_i^2X_i)E(Y_i) = 0.\end{aligned}$$

Similarly $\mathrm{Cov}(Y_i^2, 2X_iY_i) = 0$. Thus

$$V_i' = \mathrm{Var}(X_i^2) + \mathrm{Var}(Y_i^2) + 4\sigma_i^4.$$

With the *Normality* assumption, $\frac{X_i^2}{\sigma_i^2}$ as well as $\frac{Y_i^2}{\sigma_i^2}$ becomes χ^2 distributed with *DOF* 1 having *mean* = 1 and *variance* = 2 [14]. Thus

$$V_i' = \sigma_i^4\mathrm{Var}\left(\frac{X_i^2}{\sigma_i^2}\right) + \sigma_i^4\mathrm{Var}\left(\frac{X_i^2}{\sigma_i^2}\right) + 4\sigma_i^4 = 8\sigma_i^4.$$

Finally

$$\mathrm{Var}(\widehat{D}) = \sum_{i\in S} V_i + \sum_{i\in N} V_i' = 16\sum_{i\in S} b_i^2\sigma_i^2 + 20\sum_{i\in S} b_i^4 + 8\sum_{i\in N}\sigma_i^4.$$

References

[1] C.C. Aggarwal, P.S. Yu, A condensation approach to privacy preserving data mining, in: Ninth International Conference on Extending Database Technology, Heraklion, Crete, Greece, 2004.
[2] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, A. Zhu, Anonymizing tables, in: ICDT, 2005.
[3] Z.A. Aghbari, Array-index: a plug and search k nearest neighbors method for high-dimensional data, Data and Knowledge Engineering 52 (3) (2005) 333–352.
[4] D. Agrawal, C.C. Aggarwal, On the design and quantification of privacy preserving data mining algorithms, in: Twentieth ACM PODS, Santa Barbara, CA, 2001.
[5] R. Agrawal, R. Srikant, Privacy preserving data mining, in: 2000 ACM SIGMOD, Dallas, TX, May 2000.
[6] S. Agrawal, J.R. Haritsa, A framework for high-accuracy privacy-preserving mining, in: ICDE, 2005.
[7] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A.Y. Wu, An optimal algorithm for approximate nearest neighbor searching fixed dimensions, Journal of the ACM 45 (6) (1998) 891–923.
[8] R.J. Bayardo, R. Agrawal, Data privacy through optimal k-anonymization, in: ICDE, 2005.
[9] D. Caragea, A. Silvescu, V. Honavar, Decision tree induction from distributed, heterogeneous, autonomous data sources, in: Conference on Intelligent Systems Design and Applications, 2003.
[10] K. Chen, L. Liu, A random rotation perturbation approach to privacy-preserving data classification, in: ICDM 2005, Houston, TX, November, 2005.
[11] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, M. Zhu, Tools for privacy preserving distributed data mining, ACM SIGKDD Explorations 4 (2) (2002) 28–34.
[12] T.M. Cover, Rates of convergence for nearest neighbor procedures, in: International Conference on Systems Sciences, 1968.
[13] T. Dalenius, S.P. Reiss, Data-swapping: a technique for disclosure control, Journal of Statistical Planning and Inference 6 (1982) 73–85.
[14] G.B. Dantzig, M.N. Thapa, Linear Programming, Springer Series in Operations Research and Financial Engineering, Springer, 2003.
[15] D.Corney, Clustering with matlab. <http://www.cs.ucl.ac.uk/staff/D.Corney/ClusteringMatlab.htm>.
[16] W. Du, M.J. Atallah, Secure multi-party computation problems and their applications: a review and open problems, in: 2001 Workshop on New Security Paradigms, Cloudcroft, NM, 2001.
[17] W. Du, C. Clifton, M.J. Atallah, Distributed data mining to protect information privacy, in: NSF Information and Data Management (IDM) Workshop, 2004.
[18] W. Du, Z. Zhan, Using randomized response techniques for privacy preserving data mining, in: Ninth ACM SIGKDD, Washington DC, August, 2003.
[19] W. Du, Z. Zhan, Building decision tree classifier on private data, in: IEEE International Conference on Privacy, Security and Data Mining, Maebashi City, Japan, December, 2002.
[20] A. Evfimevski, J. Gehrke, R. Srikant, Limiting privacy breaches in privacy preserving data mining, in: 22nd ACM PODS, San Diego, CA, June, 2003.
[21] A. Evfimevski, R. Srikant, R. Agrawal, J. Gehrke, Privacy preserving mining of association rules, in: Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), Edmonton, Alberta, Canada, July, 2002.
[22] B.C.M. Fung, K. Wang, P.S. Yu, Top-down specialization for information and privacy preservation, in: ICDE, 2005.
[23] O. Goldreich, Secure multi-party computation, unpublished manuscript, 2002. <http://www.wisdom.weizmann.ac.il/oded/pp.html>.
[24] S. Hettich, C. Blake, C. Merz, UCI repository of machine learning databases, 1998. <http://www.ics.uci.edu/mlearn/MLRepository.html>.
[25] Z. Huang, W. Du, B. Chen, Deriving private information from randomized data, in: SIGMOD 2005, Baltimore, MD, June, 2005.
[26] I. Jolliffe, Principal Component Analysis, second ed., Springer, 2002.
[27] M. Kantarcioglu, C. Clifton, Privacy-preserving distributed mining of association rules on horizontally partitioned data, IEEE TKDE 16 (9) (2004) 1026–1037.
[28] M. Kantarcioglu, J. Vaidya, Privacy preserving nave Bayes classifier for horizontally partitioned data, in: IEEE ICDM Workshop on Privacy Preserving Data Mining, Melbourne, FL, November, 2003a.
[29] H. Kargupta, S. Datta, Q. Wang, K. Sivakumar, On the privacy preserving properties of random data perturbation techniques, in: ICDM, 2003b.

[30] H. Kargupta, S. Datta, Q. Wang, K. Sivakumar, Random data perturbation techniques and privacy preserving data mining, Knowledge and Information Systems 7 (4) (2003) 387–414.
[31] H. Kargupta, W. Huang, S. Krishnamoorthy, E. Johnson, Distributed clustering using collective principal component analysis, KAIS 3 (4) (2000) 422–448.
[32] H. Kargupta, B.H. Park, A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments, IEEE TKDE 16 (2) (2004) 216–229.
[33] J.J. Kim, W.E. Winkler, Multiplicative noise for masking continuous data, Tech. Rep. 2003–2001, Statistical Research Division, US Bureau of the Census, April, 2003.
[34] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Incognito: efficient full-domain *k*-anonymity, in: SIGMOD, 2005.
[35] K. Liu, C. Giannella, H. Kargupta, An attacker's view of distance preserving maps for privacy preserving data mining., in: The 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'06), 2006.
[36] K. Liu, H. Kargupta, J. Ryan, Random projection-based multiplicative data perturbation for privacy preserving distributed data mining, IEEE TKDE 18 (1) (2006) 92–106.
[37] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkitasubramaniam, *l*-diversity: Privacy beyond k-anonymity, in: 22nd IEEE International Conference on Data Engineering (ICDE 2006), Atlanta, Georgia, April, 2006.
[38] S. Merugu, J. Ghosh, Privacy-preserving distributed clustering using generative models, in: 3rd IEEE International Conference on Data Mining (ICDM'03), Melbourne, FL, November, 2003.
[39] T.M. Mitchell, Machine Learning, International ed., MIT Press and McGraw-Hill, 1997.
[40] S. Mukherjee, Z. Chen, A. Gangopadhyay, A fuzzy programming approach for data reduction and privacy in distance based mining, International Journal of Information and Computer Security 2 (1) (2008) 27–47.
[41] S. Mukherjee, Z. Chen, A. Gangopadhyay, A privacy preserving technique for euclidean distance-based mining algorithms using Fourier-related transforms, VLDB Journal 15 (4) (2006) 292–315.
[42] S. Oliveira, O.R. Zaane, Privacy preserving clustering by data transformation, in: 18th Brazilian Symposium on Databases, 2003.
[43] M. Partridge, R. Calvo, Fast dimensionality reduction and simple PCA, Intelligent Data Analysis 2 (3) (1998) 203–214.
[44] B. Pinkas, Cryptographic techniques for privacy preserving data mining, SIGKDD Explorations 4 (2) (2002) 12–19.
[45] S. Rizvi, J.R. Haritsa, Maintaining data privacy in association rule mining, in: VLDB, 2002.
[46] P. Samarati, Protecting respondents' identities in microdata release, TKDE 13 (6) (2001) 1010–1027.
[47] L. Sweeney, *K*-Anonymity: a model for protecting privacy, International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10 (5) (2002) 557–570.
[48] H. Tanizaki, Computational Methods in Statistics and Econometrics, Statistics: A DEKKER Series of Textbooks and Monographs, Marcel Dekker, New York, 2004.
[49] Y. Tao, J. Zhang, D. Papadias, N. Mamoulis, An efficient cost model for optimization of nearest neighbor search in low and medium dimensional spaces, Transactions on Knowledge and Data Engineering 16 (10) (2004) 1169–1184.
[50] J.S. Vaidya, C. Clifton, Privacy-preserving *k*-means clustering over vertically partitioned data, in: Ninth ACM SIGKDD, Washington DC, August, 2003.
[51] J.S. Vaidya, C. Clifton, Privacy preserving association rule mining in vertically partitioned data, in: Eighth ACM SIGKDD, Edmonton, Canada, July, 2002.
[52] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. Saygin, Y. Theodoridis, State-of-the-art in privacy preserving data mining, ACM SIGMOD Record 33 (1) (2004) 50–57.
[53] D.F. Vysochanskij, Y.I. Petunin, Justification of the three sigma rule for unimodal distributions, Theory of Probability and Mathematical Statistics 21 (1980) 25–36.
[54] D. Wettschereck, T.G. Dietterich, Locally adaptive nearest neighbor algorithms, Advances in Neural Information Processing Systems, vol. 6, Morgan Kaufman Publishers, Inc., 1994.
[55] X. Xiao, Y. Tao, Anatomy: Simple and effective privacy preservation, in: VLDB, 2006.
[56] D. Yu, A. Zhang, Clustertree: integration of cluster representation and nearest-neighbor search for large data sets with high dimensions, Transactions on Knowledge and Data Engineering 15 (5) (2003) 1316–1337.
[57] J. Zhan, L. Chang, S. Matwin, Privacy preserving *k*-nearest neighbor classification, International Journal of Network Security 1 (1) (2005) 46–51.
[58] Y. Zhu, L. Liu, Optimal randomization for privacy preserving data mining, in: KDD, 2004.

Shibnath Mukherjee was a Ph.D. student in the Department of Information Systems, University of Maryland Baltimore County when this work was conducted. He currently holds a position at IBM India. He completed his bachelor's in electronics and telecommunication engineering and MBA in information systems and economics. His areas of interest and research include mathematical transforms for privacy preserving distributed data mining and data mining over mobile networks.

Ms. Madhushri Banerjee holds her BS degree in Computer Science from the University of Calcutta, India (2002). She did a Masters in Computer Science from University of Calcutta, India (2004) and another in Internet Software Systems from University of Birmingham, UK (2006). Presently she is pursuing her Doctoral degree at the department of Information Systems at the University of Maryland Baltimore County. Her research interests include Machine Learning, Data Warehousing and Mining.

Zhiyuan Chen is an assistant professor at information systems department, UMBC. He has a Ph.D. in Computer Science from Cornell University. His research interests include privacy preserving data mining, data navigation and visualization, XML, automatic database tuning, and database compression.

Aryya Gangopadhyay is an Associate Professor of Information Systems at the University of Maryland Baltimore County (UMBC). He has a Ph.D. in Computer Information Systems from Rutgers University. His research interests include privacy preserving data mining, OLAP data cube navigation, and core and applied research on data mining. He has co-authored and edited three books, many book chapters, and numerous papers in peer-reviewed journals.