Knowl Inf Syst (2011) 29:657–695
DOI 10.1007/s10115-010-0362-4

REGULAR PAPER

Geometric data perturbation for privacy preserving outsourced data mining

Keke Chen · Ling Liu

Received: 10 March 2010 / Revised: 3 October 2010 / Accepted: 4 November 2010 /
Published online: 23 November 2010

Abstract Data perturbation is a popular technique in privacy-preserving data mining. A major challenge in data perturbation is to balance privacy protection and data utility, which are normally considered as a pair of conflicting factors. We argue that selectively preserving the task/model specific information in perturbation will help achieve better privacy guarantee and better data utility. One type of such information is the multidimensional geometric information, which is implicitly utilized by many data-mining models. To preserve this information in data perturbation, we propose the Geometric Data Perturbation (GDP) method. In this paper, we describe several aspects of the GDP method. First, we show that several types of well-known data-mining models will deliver a comparable level of model quality over the geometrically perturbed data set as over the original data set. Second, we discuss the intuition behind the GDP method and compare it with other multidimensional perturbation methods such as random projection perturbation. Third, we propose a multi-column privacy evaluation framework for evaluating the effectiveness of geometric data perturbation with respect to different level of attacks. Finally, we use this evaluation framework to study a few attacks to geometrically perturbed data sets. Our experimental study also shows that geometric data perturbation can not only provide satisfactory privacy guarantee but also preserve modeling accuracy well.

Keywords Privacy-preserving data mining · Data perturbation · Geometric data perturbation · Privacy evaluation · Data mining algorithms

K. Chen (✉)
Department of Computer Science and Engineering,
Wright State University, Dayton, OH 45435, USA
e-mail: keke.chen@wright.edu

L. Liu
College of Computing, Georgia Institute of Technology,
Atlanta, GA 30332, USA

1 Introduction

With the rise of cloud computing, service-based computing is becoming the major paradigm [4,17]. Either to use the cloud platform services [5], or to use existing services hosted on clouds, users will have to export their private data to the service provider. Since these service providers are not within the trust boundary, the privacy of the outsourced data has become one of the top-priority problems [5,7]. As data mining is one of the most popular data-intensive tasks, privacy preserving data mining for the outsourced data has become an important enabling technology for utilizing the public computing resources. Different from other settings of privacy preserving data mining such as collaboratively mining private data sets from multiple parties [29,32,39,40], this paper will focus on the following setting: the data owner exports data to and then receives a model (with the quality description such as the accuracy for a classifier) from the service provider. This setting also applies to the situation that the data owner uses the public cloud resources for large-scale scalable mining, where the service provider just provides computing infrastructure.

We present a new data perturbation technique for privacy preserving outsourced data mining [1,8] in this paper. A data perturbation procedure can be simply described as follows. Before the data owners publish their data, they change the data in certain ways to disguise the sensitive information while preserving the particular data property that is critical for building meaningful data-mining models. Perturbation techniques have to handle the intrinsic tradeoff between preserving data privacy and preserving data utility, as perturbing data usually reduces data utility. Several perturbation techniques have been proposed for mining purpose recently, but these two factors are not satisfactorily balanced. For example, random noise addition approach [3,13] is weak to data reconstruction attacks and only good for very few specific data-mining models. The condensation approach [1] cannot effectively protect data privacy from naive estimation. The rotation perturbation [8,35] and random projection perturbation [31] are all threatened by prior-knowledge enabled Independent Component Analysis [22]. Multidimensional k-anonymization [27] is only designed for general-purpose utility preservation and may result in low-quality data-mining models. In this paper, we propose a new *multidimensional* data perturbation technique: geometric data perturbation that can be applied for several categories of popular data-mining models with better utility preservation and privacy preservation.

1.1 Data privacy vs. data utility

Perturbation techniques are often evaluated with two basic metrics: the level of preserved privacy guarantee and the level of preserved data utility. Data utility is often task/model-specific and measured by the quality of learned models. An ultimate goal for all data perturbation algorithms is to maximize both data privacy and data utility, although these two are typically representing conflicting goals in most existing perturbation techniques.

Level of Privacy Guarantee Data privacy is commonly measured by the difficulty level in estimating the original data from the perturbed data. Given a data perturbation technique, the more difficult the original values can be estimated from the perturbed data, the higher level of data privacy this technique provides. In [3], the variance of the added random noise is used as the level of difficulty for estimating the original values. However, recent research [2,12] reveals that variance of added noise only is not an effective indicator of privacy guarantee. More research [21,26] has shown that privacy guarantee is subject to the attacks that can reconstruct the original data (or some records) from the perturbed data. Thus, attack analysis has to be integrated into privacy evaluation. Furthermore, since the amount of attacker's prior knowledge on the original data determines the type of attacks and its effectiveness, we should also study privacy guarantee according to the level of prior knowledge the attacker may have. With this study, the data owner can decide whether the perturbed data can be released under the assumption of certain level of prior knowledge. In this paper, we will study the proposed geometric data perturbation under a new privacy evaluation framework that incorporates attack analysis and calculates multi-level privacy guarantees according to the level of attacker's prior knowledge.

Level of Data Utility The level of data utility typically refers to the amount of critical information preserved after perturbation. More specifically, the critical information should be task or model oriented. For example, decision tree and k-Nearest-Neighbor (kNN) classifier for classification modeling typically utilize different sets of information about the data sets: decision tree construction primarily concerns the related column distributions; the kNN model relies on the distance relationship which involves all columns. Most of existing perturbation techniques do not explicitly address that the critical information is actually task/model-specific. We argue that by narrowing down to preserve only the task/model-specific information, we are able to provide better quality guarantee on both privacy and model accuracy. The proposed geometric data perturbation aims to approximately preserve the geometric properties that many data-mining models are based on.

It is interesting to note that privacy guarantee and data utility have exhibited contradictive relationship in most data perturbation techniques. Typically, data perturbation algorithms that aim at maximizing the level of privacy guarantee often have to bear with reduced data utility. The intrinsic correlation between the two factors makes it challenging to find a right balance for them in developing a data perturbation technique.

1.2 Contributions and scope

Bearing the above issues in mind, we have developed the geometric data perturbation approach to privacy preserving data mining. In contrast to other perturbation approaches [1,3,8,31], our method exploits the task and model specific multidimensional information about the data sets and produces a robust data perturbation method that not only preserves such critical information well but also provides a better balance between the level of privacy guarantee and the level of data utility. The contributions of this paper can be summarized into three aspects.

First, we articulate that the multidimensional geometric properties of data sets are the critical information for many data-mining models. We define a data-mining model to be "perturbation invariant", if the model built on the geometrically perturbed data set presents a quality to that over the original data set. With geometric data perturbation, the perturbed data can be exported to the public platform, where these perturbation-invariant data-mining models are applied to obtain equivalent models. We have proved that a batch of data-mining models, including kernel methods, SVM classifiers with the three popular kernels, linear classifiers, linear regression, regression trees, and all Euclidean distance– based clustering algorithms, are invariant to geometric data perturbation with the rotation and translation components only, and we have also studied the effect of the distance perturbation component to the model invariance property.

Second, we also study whether random projection perturbation [31] can be an alternative component in geometric data perturbation, based on the formal analysis of the effect of multiplicative perturbation to model quality. We use the Gaussian mixture model [33] to show in which situations the multiplicative component can affect the model quality. It helps us understand why the rotation component is a better choice than other multiplicative components in terms of preserving model accuracy.

Third, since a random geometric transformation–based perturbation is a multidimensional perturbation, the privacy guarantee of the multiple dimensions (attributes) should be evaluated collectively, not separately. We use a unified privacy evaluation metric for all dimensions and a generic framework to incorporate attack analysis in privacy evaluation. We also analyze a set of attacks according to different levels of knowledge an attacker may have. A randomized perturbation optimization algorithm is presented to incorporate the evaluation of attack resilience into the perturbation algorithm design.

The rest of paper is organized as follows. Section 2 briefly reviews the related work in data perturbation. Section 3 defines some notations and gives the background knowledge about geometric data perturbation. Then, in Sects. 4 and 5, we define the geometric data perturbation and prove that many major models in classification, regression and clustering modeling are invariant to rotation and translation perturbation. In Sect. 5, we also extend the discussion to the effect of noise component and other choices of multiplicative components such as random projection to model quality. In Sect. 6, we first introduce a generic privacy evaluation model and define a unified privacy metric for multidimensional data perturbation. Then, a few inference attacks are analyzed under the proposed privacy evaluation model, which results in a randomized perturbation optimization algorithm. Finally, we present experimental results in Sect. 7.

2 Related work

A considerable amount of work on privacy preserving data-mining methods have been reported in recent years [1–3,10,13,40]. The most relevant work about perturbation techniques for data mining includes the random noise addition methods [3,13], the condensation-based perturbation [1], rotation perturbation [8,35] and projection perturbation [31]. In addition, k-anonymization [38] can also be regarded as a perturbation technique, and there are a large body of literatures focusing on the k-anonymity model [15]. Since our work is less relevant to the k-anonymity model, we will focus on other perturbation techniques.

Noise Additive Perturbation The typical additive perturbation technique [3] is column-based additive randomization. This type of techniques relies on the facts that (1) Data owners may not want to equally protect all values in a record; thus, a column-based value distortion can be applied to perturb some sensitive columns. (2) Data classification models to be used do not necessarily require the individual records, but only the column value distributions [3] with the assumption of independent columns. The basic method is to disguise the original values by injecting certain amount of additive random noise, while the specific information, such as the column distribution, can still be effectively reconstructed from the perturbed data.

A typical random noise addition model [3] can be precisely described as follows. We treat the original values $(x_1, x_2, \ldots, x_n)$ from a column to be randomly drawn from a random variable $\mathbf{X}$, which has some kind of distribution. The randomization process changes the original data by adding random noises $\mathbf{R}$ to the original data values, and generates a perturbed data column $\mathbf{Y}$, $\mathbf{Y} = \mathbf{X} + \mathbf{R}$. The resulting record $(x_1 + r_1, x_2 + r_2, \ldots, x_n + r_n)$ and the distribution of $\mathbf{R}$ are published. The key of random noise addition is the distribution reconstruction algorithm [2,3] that recovers the column distribution of $\mathbf{X}$ based on the perturbed data and the distribution of $\mathbf{R}$.

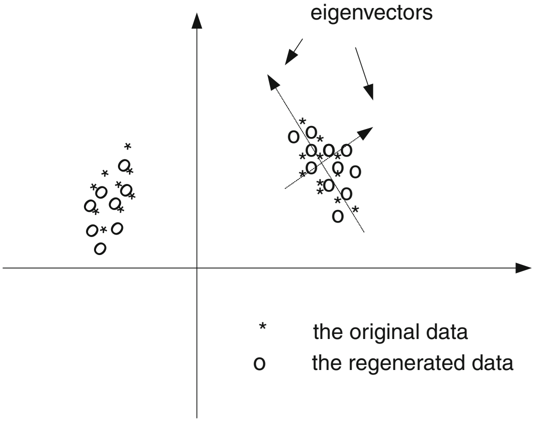


Fig. 1 Condensation approach

While the randomization approach is simple, several researchers have recently identified that reconstruction-based attacks are the major weakness of the randomization approach [21,26]. In particular, the spectral properties of the randomized data can be utilized to separate noise from the private data. Furthermore, only the mining algorithms that meet the assumption of independent columns and work on column distributions only, such as decision-tree algorithms [3], and association-rule mining algorithms [13], can be revised to utilize the reconstructed column distributions from perturbed data sets. As a result, it is inconvenient to apply this method for data mining in practice.

Condensation-based Perturbation The condensation approach [1] is a typical multi-dimensional perturbation technique, which aims at preserving the covariance matrix for multiple columns. Thus, some geometric properties such as the shape of decision boundary are well preserved. Different from the randomization approach, it perturbs multiple columns as a whole to generate the entire "perturbed data set". As the perturbed data set preserves the covariance matrix, many existing data mining algorithms can be applied directly to the perturbed data set without requiring any change or new development of algorithms.

The condensation approach can be briefly described as follows. It starts by partitioning the original data into k-record groups. Each group is formed by two steps—randomly selecting a record from the existing records as the center of group, and then finding the $(k-1)$ nearest neighbors of the center to be the other $(k-1)$ members. The selected k records are removed from the original data set before forming the next group. Since each group has small locality, it is possible to regenerate a set of k records to approximately preserve the distribution and covariance. The record regeneration algorithm tries to preserve the eigenvectors and eigenvalues of each group, as shown in Fig. 1. The authors demonstrated that the condensation approach can well preserve the accuracy of classification models if the models are trained with the perturbed data.

However, we have observed that the condensation approach is weak in protecting data privacy. As stated by the authors, the smaller the size of the locality is in each group, the better the quality of preserving the covariance with the regenerated k records is. However, the regenerated k records are confined in the small spatial locality as shown in Fig. 1. Our result (Sect. 7) shows that the differences between the regenerated records and the nearest neighbor in original data are very small on average, and thus, the original data records can be estimated from the perturbed data with high confidence.

Rotation Perturbation Rotation perturbation was initially proposed for privacy preserving data clustering [34]. As one of the major components in geometric perturbation, we first applied rotation perturbation to privacy-preserving data classification in our paper [8] and addressed the general problem of privacy evaluation for multiplicative data perturbations. Rotation perturbation is simply defined as $G(X) = RX$ where $R_{d\times d}$ is a randomly generated rotation matrix and $X_{d\times n}$ is the original data. The unique benefit and also the major weakness is distance preservation, which ensures many modeling methods are perturbation invariant while bringing distance-inference attacks. Distance-inference attacks have been addressed by recent study [9,18,30]. In [9], we discussed some possible ways to improve its attack resilience, which results in our proposed geometric data perturbation. To be self-contained, we will include some attack analysis in this paper under the privacy evaluation framework. In [35], the scaling transformation, in addition to the rotation perturbation, is also used in privacy preserving clustering. Scaling changes the distances; however, the geometric decision boundary is still preserved.

Random Projection Perturbation Random projection perturbation [31] refers to the technique of projecting a set of data points from the original multidimensional space to another randomly chosen space. Let $P_{k\times d}$ be a random projection matrix, where P's rows are orthonormal [41]. $G(X) = \sqrt{\frac{d}{k}}PX$ is applied to perturb the data set X. The rationale of projection perturbation is based on its approximate distance preservation, which is supported by the Johnson-Lindenstrauss Lemma [25]. This lemma shows that any data set in Euclidean space could be embedded into another space, such that the pair-wise distance of any two points are maintained with small error. As a result, model quality can be approximately preserved. We will compare random projection perturbation to the proposed geometric data perturbation.

3 Preliminaries

In this section, we first give the notations and then define the components in geometric perturbations. Since geometric perturbation works only for *numerical* data classification, by default, the data sets discussed in this paper are all numerical data.

3.1 Training data set

Training data set is the part of data that has to be exported/published in privacy-preserving data classification or clustering. A classifier learns the classification model from the training data and then is applied to classify the unclassified data. Suppose that X is a training data set consisting of N data rows (records) and d columns (attributes, or dimensions). For the convenience of mathematical manipulation, we use $X_{d\times N}$ to denote the data set, i.e., $X = [\mathbf{x}_1 \dots \mathbf{x}_N]$, where $\mathbf{x}_i$ is a data tuple, representing a vector in the real space $\mathbb{R}^d$. Each data tuple $\mathbf{x}_i$ belongs to a predefined class if the data is for classification modeling, which is indicated by the class label attribute y_i. The data for clustering do not have labels. The class label can be nominal (or continuous for regression), which is public, i.e., privacy-insensitive. All other attributes containing private information needs to be protected. Unclassified data set could also be exported/published with privacy-protection if necessary.

If we consider X is a sample data set from the d-dimension random vector $[\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_d]^T$, we use bold $\mathbf{X}_i$ to represent the random variable for the column i. In general, we will use bold lower case to represent vectors, bold upper case to represent random variables, and regular upper case to represent matrices.

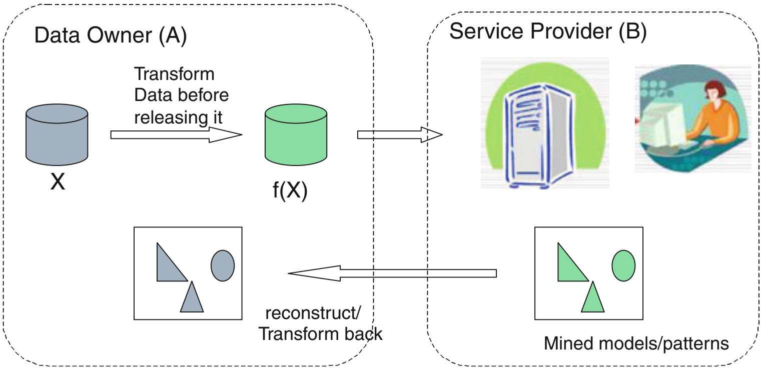

Data Owner (A)
Transform Data before releasing it
X
f(X)
reconstruct/ Transform back
Service Provider (B)
Mined models/patterns

Fig. 2 Applying geometric data perturbation to outsourced data

3.2 Framework and threat model for applying geometric data perturbation

We study geometric data perturbation under the following framework (Fig. 2). The data owner wants to use the data-mining service provider (or the public cloud service provider). The outsourced data needs to be perturbed first and then sent to the service provider. Then, the service provider develops a model based on the perturbed data and returns it to the data owner, who can use the model either by transforming it back to the original space or perturb new data to use the model. In the middle of developing models at the service provider, there is no additional interaction happening between the two parties. Therefore, the major costs for the data owner incur in optimizing perturbation parameters that can use a sample set of the data and perturbing the entire data set.

We take the popular and reasonable honest-but-curious service provider approach for our threat model. That is, we assume the service provider will honestly provide the data-mining services. However, we also assume that the provider might look at the data stored and processed on their platforms. Therefore, only well-protected data can be processed and stored on such an untrusted environment.

4 Definition of geometric data perturbation

Geometric data perturbation consists of a sequence of random geometric transformations, including multiplicative transformation (R) , translation transformation (Ψ), and distance perturbation Δ.

$$G(X) = RX + \Psi + \Delta \tag{1}$$

We briefly define these transformations and describe their properties.

4.1 Multiplicative transformation

The component R can be rotation matrix [8] or random projection matrix [31]. Rotation matrix exactly preserves distances, while random projection matrix only approximately preserve distances. We will compare the advantages and disadvantages of the two choices.

It is intuitive to understand a rotation transformation in two-dimensional or three-dimensional (2D or 3D, for short) space. We extend it to represent all kind of orthonormal transformation in multi-dimensional space. A rotation perturbation is defined as follows: $G(X) = RX$. The matrix $R_{d\times d}$ is an orthonormal matrix [36], which has some important properties. Let R^T represent the transpose of R, r_{ij} represent the (i, j) element of R, and $\mathbf{I}$ be the identity matrix. Both rows and columns of R are orthonormal: for any column j, $\sum_{i=1}^{d} r_{ij}^2 = 1$, and for any two columns j and k, $j \neq k$, $\sum_{i=1}^{d} r_{ij} r_{ik} = 0$; a similar property is held for rows. This definition infers that $R^T R = RR^T = I$. It also implies that by changing the order of the rows or columns of an orthogonal matrix, the resulting matrix is still orthonormal. A random orthonormal matrix can be efficiently generated following the Haar distribution [37], which preserves some important statistical properties [24].

A key feature of rotation transformation is preserving the Euclidean distance. Let $\mathbf{x}^T$ represent the transpose of vector $\mathbf{x}$, and $\|\mathbf{x}\| = \mathbf{x}^T\mathbf{x}$ represent the length of a vector $\mathbf{x}$. By the definition of rotation matrix, we have $\|R\mathbf{x}\| = \|\mathbf{x}\|$. Similarly, inner product is also invariant to rotation. Let $\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^T\mathbf{y}$ represent the inner product of $\mathbf{x}$ and $\mathbf{y}$. We have $\langle R\mathbf{x}, R\mathbf{y}\rangle = \mathbf{x}^T R^T R\mathbf{y} = \langle \mathbf{x}, \mathbf{y}\rangle$. In general, rotation transformation also completely preserves the geometric shapes such as hyperplane and manifold in the multidimensional space. Thus, many modeling methods are "rotation-invariant" as we will see. Rotation perturbation is a key component of geometric perturbation, which provides the primary protection to the perturbed data from naive estimation attacks. Other components of geometric perturbation are used to protect rotation perturbation from more complicated attacks.

A random projection matrix [41] $R_{k\times d}$ is defined as $R = \sqrt{\frac{d}{k}}R_0$. R_0 is randomly generated and its row vectors are orthonormal (note there is no such requirement on column vectors). The Johnson-Lindenstrauss Lemma [25] proves that random projection can approximately preserve Euclidean distances if certain conditions are satisfied. Concretely, let $\mathbf{x}$ and $\mathbf{y}$ be any original data vectors. Given $0 < \epsilon < 1$ and $k = O(ln(N)/\epsilon^2)$, there is a random projection $f : \mathcal{R}^d \to \mathcal{R}^k$, so that $(1-\epsilon)\|\mathbf{x}-\mathbf{y}\| \leq \|f(\mathbf{x}) - f(\mathbf{y})\| \leq (1+\epsilon)\|\mathbf{x}-\mathbf{y}\|$. ϵ defines the accuracy of distance preservation. Therefore, in order to precisely preserve distances, k has to be large. For large data set (N is large), it would be difficult to well preserve distances with computationally acceptable k. We will discuss the effect of random projection and rotation transformation to the result of perturbation.

4.2 Translation transformation

It is easy to understand a translation in low-dimensional ($< 4D$) space. We extend the definition to any d-dimensional spaces as follows. Ψ is a translation matrix if $\Psi = [\mathbf{t}, \mathbf{t}, \ldots, \mathbf{t}]_{d\times n}$, i.e., $\Psi_{d\times n} = \mathbf{t}_{d\times 1}\mathbf{1}_{N\times 1}^T$, where $\mathbf{1}$ is a vector of one in all elements. A translation transformation is simply: $G(X) = X + \Psi$. For any two points $\mathbf{x}$ and $\mathbf{y}$ in the original space, with translation, we have the distance $\|(\mathbf{x}-\mathbf{t}) - (\mathbf{y}-\mathbf{t})\| \equiv \|\mathbf{x}-\mathbf{x}\|$. Therefore, translation always preserves distances. However, it does not preserve inner product according to the definition of inner product.

Translation perturbation only does not provide protection to the data. The Ψ component can be simply canceled if the attacker knows only translation perturbation is applied. However, when combined with rotation perturbation, translation perturbation can increase the overall resilience to attacks.

4.3 Distance perturbation

The above two components preserve the distance relationship. By preserving distances, a bunch of important classification models will be "perturbation-invariant", which is the core of geometric perturbation. However, distance preserving perturbation may be under distance-inference attacks in some situations (Sect. 6.2). The goal of distance perturbation is to preserve distances approximately, while effectively increasing the resilience to distance-inference attacks. We define the third component as a random matrix $\Delta_{d\times n}$, where each entry is an independent sample drawn from the same distribution with zero mean and small variance. By adding this component, the distance between a pair of points is disturbed slightly.

Again, solely applying distance perturbation without the other two components will not preserve privacy, since the noise intensity is low. However, a low-intensity noise component will provide sufficient resilience to attacks to rotation and translation perturbation. The major issue brought by distance perturbation is the tradeoff between the reduction of model accuracy and the increase of privacy guarantee. In most cases, if we can assume the original data items are secure and the attacker knows no information about the original data, the distance-inference attacks cannot happen and thus the distance perturbation component can be removed. The data owner can decide to remove or keep this component according to their security assessment.

4.4 Cost analysis

The major cost of perturbation is determined by the Eq. 1 and a randomized perturbation optimization process that applies to a sample set of data set. The perturbation can be applied to data records in a streaming manner. Based on the Eq. 1, it will cost $O(d^2)$ to perturb each d-dimensional data record. Note that this is a one-time cost, and no further cost incurs with the service provider.

5 Perturbation-invariant data-mining models

In this section, first, we give the definition of perturbation invariant data-mining models that would be appropriate for our setting of mining on outsourced data. Then, we prove that several categories of data-mining models are invariant to rotation and translation perturbation. We also formally analyze the effect of the noise components and arbitrary multiplicative perturbations (including random projection) to the quality of data-mining models, using the Gaussian mixture model.

5.1 A general definition of perturbation invariance

We say a data-mining model is invariant to a transformation, if the model mined with the transformed data has a *similar* model quality as that mined with the original data. We formally define this concept as follows.

Let M represent a type of data-mining model (or modeling method) and M_X be a specific model mined from the data set X, and $Q(M_X, Y)$ be the model quality evaluated on a data set Y, e.g., the accuracy of classification model. Let $T()$ be any perturbation function, which transforms the data set X to another data set $T(X)$. Given a small real number ε, $0 < \varepsilon < 1$,

Definition 5.1 The model M_X is invariant to the perturbation $T()$ if and only if $|Q(M_X, Y) - Q(M_{T(X)}, T(Y))| < \varepsilon$ for any training data set X and testing data set Y.

If $Q(M_X, Y) \equiv Q(M_{T(X)}, T(Y))$, we call the model is *strictly invariant* to the perturbation $T()$. In the following subsections, we will prove some of the data-mining models are strictly invariant to the rotation and translation components of geometric data perturbation and discuss how the invariance property is affected by the distance perturbation component.

5.2 Perturbation-invariant classification models

In this section, we show some of the classification models that are invariant to geometric data perturbation (with only rotation and translation components). The model quality $Q(M_X, Y)$ is the classification accuracy of the trained model tested on the test data set.

kNN Classifiers and Kernel Methods A k-Nearest-Neighbor (kNN) classifier determines the class label of a point by looking at the labels of its k nearest neighbors in the training data set and classifies the point to the class that most of its neighbors belong to. Since the distance between any pair of points is not changed with rotation and translation, the k nearest neighbors are not changed and thus the classification result is not changed either.

Theorem 1 *kNN classifiers are strictly invariant to rotation and translation perturbations.*

kNN classifier is a special case of kernel methods. We assert that any kernel methods will be invariant to rotation, too. Same as the kNN classifier, a typical kernel method [1] is a local classification method, which classifies the new data record only based on the information of its neighbors in the training data.

Theorem 2 *Kernel methods are strictly invariant to rotation and translation.*

Proof Let us define kernel methods first. Like kNN classifiers, a kernel method also estimates the class label of a point $\mathbf{x}$ with the class labels of its neighbors. Let $K_\lambda(\mathbf{x}, \mathbf{x}_i)$ be the kernel function used for weighting any point $\mathbf{x}_i$ in $\mathbf{x}$'s neighborhood, and let λ define the geometric width of the neighborhood. We assume $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be the points in the $\mathbf{x}$'s neighborhood determined by λ. A kernel classifier for continuous class labels[2] is defined as

$$\hat{f}_X(\mathbf{x}) = \frac{\sum_{i=1}^{n} K_\lambda(\mathbf{x}, \mathbf{x}_i) y_i}{\sum_{i=1}^{n} K_\lambda(\mathbf{x}, \mathbf{x}_i)} \tag{2}$$

Specifically, the kernel $K_\lambda(\mathbf{x}, \mathbf{x}_i)$ is defined as

$$K_\lambda(\mathbf{x}, \mathbf{x}_i) = D\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{\lambda}\right) \tag{3}$$

$D(t)$ is a function, e.g., the Gaussian kernel $D(t) = \frac{1}{\sqrt{2\pi}} \exp\{-t^2/2\}$. Since $\|R\mathbf{x} - R\mathbf{x}_i\| = \|\mathbf{x} - \mathbf{x}_i\|$ for rotation perturbation and λ is constant, $D(t)$ is not changed after rotation, and thus $K_\lambda(R\mathbf{x}, R\mathbf{x}_i) = K_\lambda(\mathbf{x}, \mathbf{x}_i)$. Since the geometric area around the point is also not changed, the point set in the neighborhood of $R\mathbf{x}$ are still the rotation of those in the neighborhood of $\mathbf{x}$, i.e., $\{R\mathbf{x}_1, R\mathbf{x}_2, \ldots, R\mathbf{x}_n\}$ and these n points are used in training M_{RX}, which makes $Q(M_{RX}, (R\mathbf{x}) = \hat{f}_X(\mathbf{x})$. It is similar to prove that kernel methods are invariant to translation perturbation. □

Support Vector Machines Support Vector Machine (SVM) classifiers also utilize kernel functions in training and classification. However, it has an explicit training procedure to generate a global model, while kernel methods are local methods that use training samples in classifying new instances. Let y_i be the class label to a tuple $\mathbf{x}_i$ in the training set, α_i and β_0 be the

[1] SVM is also a kind of kernel method, but its training process is different from the kernel methods we discuss here.

[2] It has different form for discrete class labels, but the proof will be similar.

parameters determined by training. A SVM classifier calculates the classification result of **x** using the following function.

$$\hat{f}_X(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + \beta_0 \tag{4}$$

First, we prove that SVM classifiers are invariant to rotation with two key steps: (1) training with the rotated data set generates the same set of parameters α_i and β_0; (2) the kernel function $K()$ is invariant to rotation. Second, we prove that some SVM classifiers are also invariant to translation (empirically, SVM classifiers with the discussed kernels are all invariant to translation).

Theorem 3 *SVM classifiers using polynomial, radial basis, and neural network kernels are strictly invariant to rotation, and SVM classifiers using radial basis are also strictly invariant to translation.*

Proof The SVM training problem is an optimization problem, which finds the parameters α_i and β_0 to maximize the Lagrangian (Wolfe) dual objective function [20]

$$L_D = \sum_{i=1}^{N} \alpha_i - 1/2 \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j),$$

subject to:

$$0 < \alpha_i < \gamma, \qquad \sum_{i=1}^{N} \alpha_i y_i = 0,$$

where γ is a parameter chosen by the user to control the allowed errors around the decision boundary. The training result of α_i is only determined by the form of kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. With the determined α_i, β_0 can be determined by solving $y_i \hat{f}_X(\mathbf{x}_i) = 1$ for any $\mathbf{x}_i$ [20], which is again determined by the kernel function. It is clear that if $K(T(\mathbf{x}), T(\mathbf{x}_i)) = K(\mathbf{x}, \mathbf{x}_i)$ is held, the training procedure generates the same set of parameters.

Three popular choices for kernels have been discussed in the SVM literature [11,20].

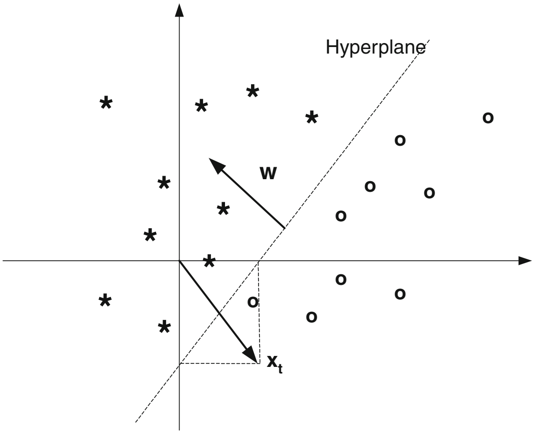
$$\begin{aligned} \text{dth degree polynomial:} \quad & K(\mathbf{x}, \mathbf{x}') = (1 + <\mathbf{x}, \mathbf{x}'>)^d, \\ \text{radial basis:} \quad & K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|/c), \\ \text{neural network:} \quad & K(\mathbf{x}, \mathbf{x}') = \tanh(\kappa_1 <\mathbf{x}, \mathbf{x}'> + \kappa_2) \end{aligned}$$

Note that the three kernels only involve distance and inner product calculation. As we discussed in Sect. 4, the two operations keep invariant to the rotation transformation. Thus, $K(R\mathbf{x}, R\mathbf{x}') = K(\mathbf{x}, \mathbf{x}')$ is held for the three kernels, and, thus, training with the rotated data will not change the parameters for the SVM classifiers using the three popular kernels. However, with this method, we can only prove that the radial basis kernel is invariant to translation, while the other two are not.

It is easy to verify that the classification function (Eq. 4) is invariant to rotation, which involves only the invariant parameters and the invariant kernel functions. Similarly, we can prove that the classification function with radial basis kernel is also invariant to translation. □

Although we cannot prove that polynomial and neural network kernels are also invariant to translation with this method, we use experiments to show that they are also invariant to translation.

Fig. 3 Hyperplane and its parameters



Linear Classifiers A linear classifier uses a hyperplane to separate the training data. Let the weight vector be $\mathbf{w}^T = [w_1, \ldots, w_d]$ and the bias be β_0. The weight and bias parameters are determined by the training procedure [20]. A trained classifier is represented as follows.

$$\hat{f}_X(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \beta_0$$

Theorem 4 *Linear classifiers are strictly invariant to rotation and translation.*

Proof First, it is important to understand the relationship between the parameters and the hyperplane. As Fig. 3 shows, the hyperplane can be represented as $\mathbf{w}^T(\mathbf{x} - \mathbf{x_t}) = 0$, where $\mathbf{w}$ is the perpendicular axis to the hyperplane, and $\mathbf{x}_t$ represents the deviation of the plane from the origin (i.e., $\beta_0 = -\mathbf{w}^T \mathbf{x}_t$).

Intuitively, rotation will rotate the classification hyperplane and feature vectors. The perpendicular axis $\mathbf{w}$ is changed to $R\mathbf{w}$ and the deviation $\mathbf{x}_t$ becomes $R\mathbf{x}_t$ after rotation. Let $\mathbf{x}^r$ represent the data in the rotated space. Then, the rotated hyperplane is represented as $(R\mathbf{w})^T(\mathbf{x}^r - R\mathbf{x}_t) = 0$, and the classifier is transformed to $\hat{f}_{RX}(\mathbf{x}^r) = \mathbf{w}^T R^T(\mathbf{x}^r - R\mathbf{x}_t)$. Since $\mathbf{x^r} = R\mathbf{x}$ and $R^T R = I$, $\hat{f}_{RX}(\mathbf{x}^r) = \mathbf{w}^T R^T R(\mathbf{x} - \mathbf{x_t}) = \mathbf{w}^T(\mathbf{x} - \mathbf{x_t}) = \hat{f}_X(\mathbf{x})$. The two classifiers are equivalent.

It is also easy to prove that linear classifiers are invariant to translation. We will ignore the proof. □

5.3 Perturbation invariant regression methods

Regression modeling [20] is very similar to classification modeling. The only difference is that the class label is changed from discrete to continuous, which requires the change of the criterion for model evaluation. A regression model is often evaluated by the loss function $L(f(X), \mathbf{y})$, where $f(X)$ is the response vector of applying the regression function $f()$ to the training instances X, and $\mathbf{y}$ is the original target vector (i.e., the class labels in classification modeling). A typical loss function is mean square error (MSE).

$$L(f(X), \mathbf{y}) = \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$$

As the definition of model quality is instantiated by the loss function L, we give the following definition of perturbation invariant regression model.

Definition 5.2 A regression method is invariant to a transformation T if and only if $|L(f_X(Y), \mathbf{y}_Y) - L(f_{T(X)}(T(Y)), \mathbf{y}_Y)| < \varepsilon$ for any training data set X, and any testing data set Y. $0 < \varepsilon < 1$ and $\mathbf{y}_Y$ is the target vector of the testing data Y.

Similarly, the strictly invariant condition becomes $L(f_X(Y), \mathbf{y}_Y) \equiv L(f_{T(X)}(T(Y)), \mathbf{y}_Y)$. We prove that

Theorem 5 *The linear regression model using MSE as the loss function is strictly invariant to rotation and translation.*

Proof The linear regression model based on the MSE loss function can be represented as $\mathbf{y} = X^T\beta + \epsilon$, where ϵ is a vector of random Gaussian noise with mean zero and variance σ^2. The estimate of β is $\hat{\beta} = (XX^T)^{-1}X\mathbf{y}_X$. Thus, for any testing data Y, the estimated model is $\hat{\mathbf{y}}_\mathbf{Y} = Y^T\hat{\beta}$. Since the loss function for the testing data Y is $L(f_X(Y), \mathbf{y}) = \|Y^T(XX^T)^{-1}X\mathbf{y}_X - \mathbf{y}_Y\|$. After rotation, it becomes

$$\begin{aligned} L(f_{X^TR^T}(Y^TR^T), \mathbf{y}) &= \|Y^TR^T(RX(RX)^T)^{-1}RX\mathbf{y}_X - \mathbf{y}_Y\| \\ &= \|Y^TR^T(RXX^TR^T)^{-1}RX\mathbf{y}_X - \mathbf{y}_Y\| \\ &= \|Y^TR^T(R^T)^{-1}(XX^T)^{-1}R^{-1}RX\mathbf{y}_X - \mathbf{y}_Y\| \\ &= \|Y^T(XX^T)^{-1}X\mathbf{y}_X - \mathbf{y}_Y\| \equiv L(f_X(Y), \mathbf{y}) \end{aligned} \tag{5}$$

The linear regression model can also be represented as $y = \hat{\beta}_0 + \sum_{i=1}^{d}\hat{\beta}_i x_i$, where x_i is the value of dimension i for the vector $\mathbf{x}$. It is clear that if $\mathbf{x}$ is translated to $\mathbf{x}' = \mathbf{x} + \mathbf{t}$, we can reuse the model parameters except $\hat{beta}_0$ is replaced with $\hat{beta}_0 - d\mathbf{t}$. Thus, the new model does not change MSE as well. □

Other regression models, such as regression tree–based methods [14], which partitions the global space based on Euclidean distance, are also strictly invariant to rotation and translation. We skip the details here.

5.4 Perturbation invariant clustering algorithms

There are several metrics used to evaluate the quality of clustering result, all of which are based on cluster membership, i.e., record i belongs to cluster C_j. Suppose the number of cluster is fixed as K. The same clustering algorithm applied to the original data and the perturbed data will generate two clustering results. Since the record ID does not change before and after perturbation, we can compare the difference between two sets of clustering results to evaluate the invariance property. We use the *confusion matrix* method [23] to evaluate this difference, where each element c_{ij} $1 \le i, j \le K$ represents the number of points from the cluster j in the original data set assigned to cluster i by the clustering result on the perturbed data. Since cluster labels may represent different clusters in two clustering results. Let {(1), (2),..., (K)} be any permutation of the sequence of cluster labels {1, 2, ..., K}. There is a permutation that best matches the clustering results of before and after data perturbation and maximizes the number of consistent points m_C for clustering algorithm C.

$$m_C = \max\left\{\sum_{i=1}^{K} c_{i(i)}, \text{for any } \{(1), (2), \ldots, (K)\}\right\}$$

We define the error rate as $DQ_C(X, T(X)) = 1 - \frac{m_C}{N}$, where N is the total number of points. DQ_C is the quality difference between the two clustering results. Then, the criterion for perturbation invariant clustering algorithm can be defined as

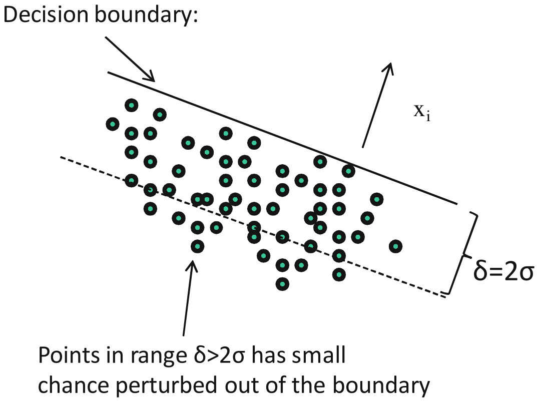

Decision boundary:
x_i
δ=2σ
Points in range δ>2σ has small chance perturbed out of the boundary

Fig. 4 Analyzing the points being perturbed out of boundary

Definition 5.3 A clustering algorithm is invariant to a transformation T if and only if $DQ_C(X, T(X)) < \varepsilon$ for any data set X, and a small value $0 < \varepsilon < 1$.

For strict invariance, $DQ_C(X, T(X)) = 0$.

Theorem 6 *Any clustering algorithms or cluster visualization algorithms that are based on Euclidean distance are strictly invariant to rotation and translation.*

Since geometric data perturbation aims at preserving the Euclidean distance relationship, the cluster membership does not change before and after perturbation. Thus, it is easy to prove that the above theorem is true and we skip the proof.

5.5 Effect of noise perturbation to the invariance property

Intuitively, the noise component will affect the quality of data- mining model. In this section, we give a formal analysis on how the noisy intensity affects the model quality for classification (or clustering) modeling.

Assume the boundary for the data perturbed without the noise component is shown in Fig. 4 and the noises are drawn from the normal distribution $N(0, \sigma^2)$. Let's look at the small band with δ distance (one side) around the classification or clustering boundary. The increased error rate is determined by the number of points that are original properly classified or clustered but now are perturbed to the other side of the boundary. Out of the band, the points are less likely perturbed to the other side of the boundary. For a d-dimension point $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, its perturbed version (with only the noise component) is represented as $\mathbf{x}' = (x_1 + \epsilon_1, x_2 + \epsilon_2, \ldots, x_d + \epsilon_d)$, where ϵ_i is drawn from the same distribution $N(0, \sigma^2)$. To further simplify the analysis, assume a decision boundary is perpendicular to one of the dimensions (we can always rotate the data set to meet this setting), say x_i, and there are n points uniformly distributed in the δ band.

According to normal distribution, for $\delta > 2\sigma$, the points located out of the δ band, will have small probability (< 0.025) to be perturbed to the other side of the boundary. Therefore, we consider only the points within the $\delta = 2\sigma$ band. Let $p(y)$ be the probability of a point that has distance y to the boundary perturbed out of the boundary, then the average number of points perturbed out of the boundary is

$$\int_0^{2\sigma} p(y) \frac{n}{2\sigma} \mathrm{d}y = \int_0^{2\sigma} \int_y^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{x^2}{2\sigma^2}} \mathrm{d}x \frac{n}{2\sigma} \mathrm{d}y.$$

Expanding $\exp^{-\frac{x^2}{2\sigma^2}}$ with Taylor series [16] for the first three terms we obtain $\exp^{-\frac{x^2}{2\sigma^2}} \approx 1 - \frac{x^2}{2\sigma^2} + \frac{x^4}{8\sigma^4}$. With the fact $\int_y^\infty \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{x^2}{2\sigma^2}} \mathrm{d}x = 1/2 - \int_0^y \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{x^2}{2\sigma^2}} \mathrm{d}x$, we solve the equation and get the number of out-of-the-boundary points $\approx \left(\frac{1}{2} - \frac{4}{5\sqrt{2\pi}}\right) n \approx 0.18n$. The other side of the boundary has the similar amount of points perturbed out of the boundary. Depending on the data distribution and σ, the amount of affected data points can vary. Borrowing the concept of "margin" from SVM literature, we understand that if the margin is greater than 2δ, the model accuracy is not affected at all; if the margin is less than 2δ, the model quality is affected by the amount of points in the 2δ region.

5.6 Effect of general multiplicative perturbation to model quality

In geometric data perturbation, the rotation and translation components strictly preserve distance, which is then slightly perturbed by distance perturbation. If we relax the condition of strictly preserving distance, what will happen to the discussed mining models? This relaxation may use any linear transformation matrix to replace the rotation component, e.g., projection perturbation [31]. In this section, we will discuss the effect of a general multiplicative perturbation with $G(\mathbf{x}) = A\mathbf{x}$ to classification model quality, where A is a $k \times d$ matrix and k may not equal to d. We analyze why arbitrary projection perturbations do not generally preserve geometric decision boundaries and what are the alternative ways to rotation perturbation to generate decision-boundary (or approximately) preserving multiplicative perturbations.

This analysis is based on a simplified model of data distribution—multidimensional Gaussian mixture model. Assume the data set can be modeled with multiple data clouds, each of which has approximately normal (Gaussian) distribution $N(\mu_i, \Sigma_i)$, where μ_i is the mean vector and Σ_i is the covariance matrix. Since such a general multiplicative perturbation does not necessarily preserve all of the geometric properties for the data set, it is not guaranteed that the discussed data-mining models will be invariant to these transformations. Let us first consider a more general case that does not put a constraint on k. The rationale of projection perturbation is based on approximate distance preservation supported by the Johnson-Lindenstrauss Lemma [25].

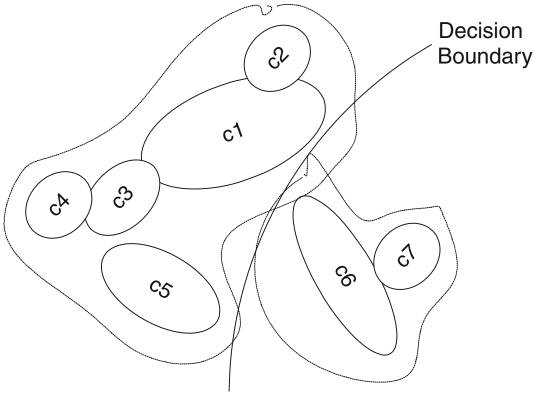
Theorem 7 *For any $0 < \epsilon < 1$ and any integer n, let k be a positive integer such that $k \geq \frac{4 \log n}{\epsilon^2/2 - \epsilon^3/3}$. Then, for any set $\mathbf{S}$ of n data points in d dimensional space $\mathbb{R}^d$, there is a mapping function $f \colon \mathbb{R}^d \to \mathbb{R}^k$ such that, for all $\mathbf{x} \in \mathbf{S}$,*

$$(1 - \epsilon)\|\mathbf{x} - \mathbf{x}\|^2 \leq \|f(\mathbf{x}) - f(\mathbf{x})\|^2 \leq (1 + \epsilon)\|\mathbf{x} - \mathbf{x}\|^2$$

where $\|\cdot\|$ denotes the vector 2-norm.

This lemma shows that any set of n points in d-dimensional Euclidean space could be embedded into a $O\left(\frac{\log n}{\epsilon^2}\right)$—dimensional space with some linear transformation f, such that the pair-wise distance of any two points are maintained with a controlled error. However, there is a cost to achieve high precision in distance preserving. For example, a setting of $n = 1000$, a quite small data set, and $\epsilon = 0.01$, will require $k \approx 0.5$ million dimensions, which makes the transformation impossible to perform. Increasing ϵ to 0.1, we still need about $k \approx 6,000$. In order to further reduce k, we have to increase ϵ more, which brings larger errors, however. In the case of increased distance error, the decision boundary may not be well preserved.

Fig. 5 Use mixture to describe the classes in the original dataset



We also analyze the effect of transformation from a more intuitive perspective. In order to see the connections between the general linear transformation and data-mining models, we use classifiers that are based on geometric decision boundaries for example.

Below we name a dense area (a set of points are similar to each other) as *cluster*, while the points with the same class label are in the same *class*. We can approximately model the whole data set with Gaussian mixtures based on its density property. Without loss of generality, we suppose that a geometrically separable class consists of one or more Gaussian clusters as shown in Fig. 5. Let μ be the density center, and Σ be the covariance matrix of one Gaussian cluster. A cluster C_i can be represented with the following distribution.

$$\mathcal{N}_d^i(\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\{-(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)/2\}$$

μ describes the position of the cluster and Σ describes the *hyper-elliptic shape* of the dense area. After the transformation with invertible A, the center of the cluster is moved to $A\mu$ and the covariance matrix (corresponding to the shape of dense area) is changed to $A\Sigma A^T$.

Let $\mathbf{x}$ and $\mathbf{y}$ be any two points. After the transformation, the distance between the two becomes $D' = ||A(\mathbf{x}-\mathbf{y})|| = (\mathbf{x}-\mathbf{y})^T A^T A(\mathbf{x}-\mathbf{y})$. If we compare this distance to the original distance D, we get their difference as

$$D' - D = (\mathbf{x}-\mathbf{y})^T(A^T A - I)(\mathbf{x}-\mathbf{y}) \tag{6}$$

We study the property of $A^T A - I$ to find how the distance changes. First, for a random *invertible* and *diagonalizable* [6] matrix A that preserves dimensions, i.e., $k = d$, we will have $A^T A$ positive definite for the following reason. Since A is diagonalizable, A can be eigen-decomposed to $U^T \Lambda U$, where U is an orthogonal matrix, Λ is the diagonal matrix of eigenvalues, and all eigenvalues are non-zero for the invertibility of A. Then, we have $A^T A = U^T \Lambda U U^T \Lambda U = U^T \Lambda^2 U$, where all eigenvalues of Λ^2 are positive. Therefore, $A^T A$ is positive definite. If all eigenvalues of $A^T A$ are greater than 1, then $A^T A - I$ will be positive definite and $D' - D > 0$ for all distances. Similarly, if all eigenvalues of $A^T A$ are less than 1, then $A^T A - I$ will be negative definite and $D' - D < 0$ for all distances. For any case else, we are unable to determine how distances change—it can be lengthened or shortened. Because of the possibly arbitrary change of distances for an arbitrary A, the points belonging to one cluster may possibly become members of another cluster. Since we define classes based on clusters, the change of clustering structure may also perturb the decision boundary.

Then, what kind of perturbations will preserve clustering structures? Besides the distance preserving perturbations, we may also use a family of *distance-ordering preserving* perturbations. Assume $\mathbf{x}$, $\mathbf{y}$, $\mathbf{u}$, $\mathbf{v}$ are any four points in the original space, and $||\mathbf{x} - \mathbf{y}|| \leq ||\mathbf{u} - \mathbf{v}||$, i.e., $\sum_{i=1}^{d}(x_i - y_i)^2 \leq \sum_{i=1}^{d}(u_i - v_i)^2$, which defines the order of distances, where x_i, y_i, u_i, and v_i are dimensional values. It is easy to verify that if distance ordering is preserved after transformation, i.e., $||G(\mathbf{x}) - G(\mathbf{y})|| \leq ||G(\mathbf{u}) - G(\mathbf{v})||$, the clustering structure is preserved as well and thus the decision boundary is preserved. Therefore, distance ordering preserving perturbation is an alternative choice to rotation perturbation.

In the following, we discuss how to find a distance ordering preserving perturbation. Let $\lambda_i^2, i = 1, \ldots, d$, be the eigenvalues of $A^T A$. Then, the distance ordering preserving property requires $\sum_{i=1}^{d} \lambda_i^2 (x_i - y_i)^2 \leq \sum_{i=1}^{d} \lambda_i^2 (u_i - v_i)^2$. Apparently, for arbitrary A, this condition cannot be satisfied. One simple setting will guarantee to preserve distance ordering that is $\lambda_i = \lambda$, where λ is some constant. This results in distance ordering preserving matrices $A = \lambda R$ where R is a rotation matrix and λ is an arbitrary constant − we name it *scaling* of the rotation matrix. Based on this analysis, we can also derive approximate distance ordering preserving by perturbing λ_i to $\lambda + \delta_i$, where δ_i is a small value drawn from a distribution. In fact, scaling is also discussed in transformation-based data perturbation for privacy preserving clustering [35].

6 Attack analysis and Privacy guarantee of geometric data perturbation

The goal of random geometric perturbation is twofold: preserving the data utility and preserving the data privacy. The discussion about the transformation-invariant classifiers has proven that geometric transformations theoretically guarantee preserving the model accuracy for many models. As a result, numerous such geometric perturbations can present the same model accuracy, and we only need to find one that *maximizes the privacy guarantee* in terms of various potential attacks.

We dedicate this section to discuss how good the geometric perturbation approach is in terms of preserving privacy. The first critical step is to define a *multi-column privacy measure* for evaluating the privacy guarantee of a geometric perturbation to a given data set. It should be distinct from that used for additive perturbation [3], which assumes each column is independently perturbed, since geometric perturbation changes the data on all columns (dimensions) together. We will use this multi-column privacy metric to evaluate several attacks and optimize the perturbation in terms of attack resilience.

6.1 A conceptual privacy model for multidimensional perturbation

Unlike the existing random noise addition methods, where multiple columns are perturbed independently, random geometric perturbation needs to perturb *all* columns together. Therefore, the privacy quality of all columns is correlated under one single transformation and should be evaluated under a unified metric. We first present a conceptual model for privacy evaluation in this section, and then we will discuss the design of the unified privacy metric and a framework for incorporating attack evaluation.

In practice, since different columns (attributes) may have different privacy concern, we consider that the general-purpose privacy metric Φ for entire data set should be based on *column privacy metric*. A conceptual privacy evaluation model is defined as follows. Let $\mathbf{p}$ be the column privacy metric vector $\mathbf{p} = (p_1, p_2, \ldots, p_d)$, and there are *privacy weights* associated with the d columns, respectively, denoted as $\mathbf{w} = (w_1, w_2, \ldots, w_d)$. Without loss of generality, we assume that the weights are normalized, i.e., $\sum_{i=1}^{d} w_i = 1$. Then, $\Phi = \Phi(\mathbf{p}, \mathbf{w})$ defines the privacy guarantee. In summary, the design of the specific privacy model should consider the three factors **p**, **w**, and the function Φ.

We will leave the concrete discussion about the design of **p** in the next section, and define the other two factors first. We notice that different columns may have different importance in terms of the level of privacy sensitivity. The first design idea is to take the column importance into consideration. Intuitively, the more important the column is, the higher level of privacy guarantee will be required for the perturbed data, corresponding to that column. If we use w_i to denote the importance of column i in terms of preserving privacy, p_i/w_i can be used to represent the *weighted column privacy* for column i.

The second intuition is the concept of *minimum privacy guarantee* among all columns. Normally, when we measure the privacy quality of a multi-column perturbation, we need to pay special attention to the column that has the lowest weighted column privacy, because such a column could become the breaking point of privacy. Hence, we design the first composition function $\Phi_1 = \min_{i=1}^{d}\{p_i/w_i\}$ and call it the minimum privacy guarantee. Similarly, the *average privacy guarantee* of the multi-column perturbation, defined by $\Phi_2 = \frac{1}{d}\sum_{i=1}^{d} p_i/w_i$, could be another interesting measure.

With the definition of privacy guarantee, we can evaluate the privacy quality of a perturbation to a specific data set, and most importantly, we can use it to find a multi-dimensional perturbation that locally maximizes the privacy guarantees. With random geometric perturbation, we demonstrate that it is convenient to adjust the perturbation method to obtain high privacy guarantees, without the concern of preserving the model accuracy for the discussed classifiers.

6.1.1 A unified column privacy metric

Intuitively, for a data perturbation approach, the quality of preserved privacy can be understood as the difficulty level of estimating the original data from the perturbed data. We name such estimation methods as "inference attacks". A unified metric should be a generic metric that can be used to evaluate as many types of inference attacks as possible. In the following, we first derive a unified privacy metric from the mean-square-error method, and then discuss how to apply the metric to evaluate the attacks to geometric perturbation.

We compare the original value and the estimated value to determine the uncertainty brought by the perturbation. This uncertainty is the privacy guarantee that protects the original value. Let the difference between the original column data **Y** and the perturbed/reconstructed data $\hat{\mathbf{Y}}$ be a random variable **D**. We use the root of mean square error (RMSE) to estimate this difference. Assume the original data samples are $y_1, y_2, \ldots, y_N$. Correspondingly, the estimated values are $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N$. The root of mean square error of estimation, r, is defines as

$$r = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

As we have discussed, to evaluate the privacy quality of multi-dimensional perturbation, we need to evaluate the privacy of all perturbed columns together. Unfortunately, the single-column metric is subject to the specific column distribution, i.e., the same amount is not equally effective for different value scales. For example, the same amount of RMSE for the "age" column has much stronger protection than for "salary" due to the dramatically

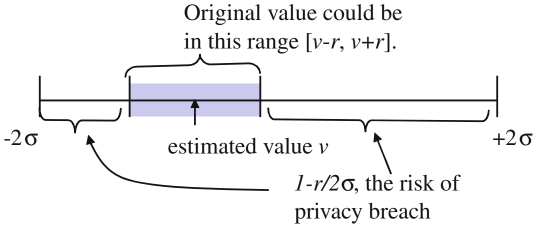

Original value could be in this range [v-r, v+r].
-2σ
estimated value v
+2σ
1-r/2σ, the risk of privacy breach

Fig. 6 The intuition behind the privacy metric

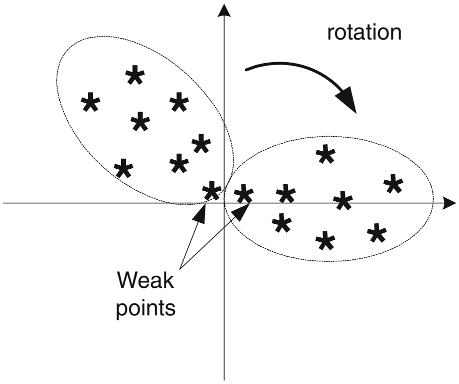


Fig. 7 Weak points around the default rotation center

different value ranges. One effective way to unify the different value ranges is via *normalization*, e.g., max–min normalization or standardization. We employ the standardization procedure, which is simply described as a transformation to the original value $y' = \frac{y-\mu}{\sigma}$, where μ is the mean of the column and σ is the standard deviation. By using this procedure, all columns are approximately unified into the same data range. The rationale behind the standardization procedure is that for large sampleset (e.g. hundreds of samples) normal distribution would be a good approximation for most distributions [28]. The standardization procedure normalizes all distributions to standard normal distribution (with mean zero and variance one). According to normal distribution, the range $[\mu - 2\sigma, \mu + 2\sigma]$ covers more than 95% points in the population. Let's use this range, i.e., 4σ to approximately represent the value range. We use the normalized values, the definition of RMSE, and the normalized value range to represent the unified privacy metric.

$$Priv(\hat{\mathbf{y}}, \mathbf{y}') = \frac{2r}{4\sigma} = \frac{1}{2\sigma}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - \mu}{\sigma} - \hat{y}_i\right)^2}$$

This definition[3] can be explained with Fig. 6. The normalized RMSE $r = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\frac{y-\mu}{\sigma} - \hat{y}_i)^2}$ represents the average error on value estimation. The real value can be in the range of $[\hat{y}_i - r, \hat{y}_i + r]$. The rate of this range $2r$ to the value range 4σ represents the normalized uncertainty of estimation. This rate can be possibly higher than 1, which means extremely large uncertainty. If the original data is standardized ($\sigma = 1$), this metric is reduced to $p = r/2$.

[3] Note that this definition is improved from the one we gave in the SDM paper [9].

6.1.2 Incorporating attack analysis into privacy evaluation

The proposed metric should compare the difference between two datasets: the original data set and the *observed or estimated data set*. With different level of knowledge, the attacker observes the perturbed data set differently. The attacks we know so far can be summarized into the following three categories: (1) the basic statistical methods that estimate the original data directly from the perturbed data [2,3], without any other knowledge about the data (as known as "naive inference"); (2) data reconstruction methods that reconstruct data from the perturbed data with any released information about the data and the perturbation, and then use the reconstructed data to estimate the original data [21,26] (as known as "reconstruction-based inference"); and (3) if some particular original records and their image in the perturbed data can be identified, e.g., outliers of the data sets, based on the preserved distance information, the mapping between these points can be used to discover the perturbation (as known as distance-based inference).

Let X be the normalized original data set, P be the perturbed data set, and O be the observed data set. We calculate $\Phi(X, O)$, instead of $\Phi(X, P)$, in terms of different attacks. Using rotation perturbation $G(X) = RX$ for example, we can summarize the evaluation of privacy in terms of attacks.

1. Naive inference: $O = P$, there is no more accurate estimation than the released perturbed data;
2. Reconstruction-based inference: methods like Independent Component Analysis (ICA) are used to estimate R. Let $\hat{R}$ be the estimate of R, and $O = \hat{R}^{-1}P$;
3. Distance-based inference: the attacker knows a small set of special points in X that can be mapped to certain set of points in P, so that the mapping helps to estimate R, and then $O = \hat{R}^{-1}P$.
4. Subset-based inference: the attacker knows a significant number of original points that can be used to estimate R and then $O = \hat{R}^{-1}P$.

The higher the inference level is, the more knowledge about the original data set the attacker needs to break the perturbation. In the following sections, we analyze some inference attacks and see how geometric perturbation provides resilience to these attacks.

Note that the proposed privacy evaluation method is a generic method that can be used to evaluate the effectiveness of a general perturbation, where nothing but the perturbed data is released to the attacker. It is important to remember that this metric should be evaluated on the original data and the *estimated* data. We cannot simply assume the perturbed data is the estimated data as the original additive perturbation does [3], which makes the assumption that the attacker has no knowledge about the original data.

6.2 Attack analysis and perturbation optimization

In this section, we will use the unified multi-column privacy metric to analyze a few attacks. The similar methodology can be used to analyze any new attacks. Based on the analysis, we develop a randomized perturbation optimization algorithm.

6.2.1 Privacy analysis on naive estimation attack

We start with the analysis on multiplicative perturbation, which is the key component in geometric perturbation. With the proposed metric over the normalized data, we can formally analyze the privacy quality of random rotation perturbation. Let X be the normalized data set, $X' = RX$ be the rotation of X, and I_d be the d-dimensional identity matrix. Thus, the difference matrix $X' - X$ can be used to calculate the privacy metric, and the columnwise metric is based on the element (i, i) in $K = (X' - X)(X' - X)^T$ (note that X and X' are column vector matrices as we defined), i.e., $\sqrt{K_{(i,i)}}/2$, where $K_{(i,i)}$ is represented as

$$K_{(i,i)} = ((R - I_d)XX^T(R - I_d)^T)_{(i,i)} \tag{7}$$

Since X is normalized, $\mathbf{XX}^T$ is also the covariance matrix, where the diagonal elements are the column variances. Let r_{ij} represent the element (i, j) in the matrix R, and c_{ij} be the element (i, j) in the matrix of $\mathbf{XX}$. $K_{(i,i)}$ is transformed to

$$K_{(i,i)} = \sum_{j=1}^{d}\sum_{k=1}^{d} r_{ij}r_{ik}c_{kj} - 2\sum_{j=1}^{d} r_{ij}c_{ij} + c_{ii} \tag{8}$$

When the random rotation matrix is generated following the Haar distribution, a considerable number of matrix entries are approximately independent normal distribution $N(0, 1/d)$ [24]. The full discussion about the numerical characteristics of the random rotation matrix is out of the scope of this paper. For simplicity and easy understanding, we assume that all entries in random rotation matrix approximately follow independent normal distribution $N(0, 1/d)$. Therefore, sample randomly rotations should make $K_{(i,i)}$ changing around the mean value c_{ii} as shown in the following result.

$$E[K_{(i,i)}] \sim \sum_{j=1}^{d}\sum_{k=1}^{d} E[r_{ij}]E[r_{ik}]c_{kj} - 2\sum_{j=1}^{d} E[r_{ij}]c_{ij} + c_{ii} = c_{ii}$$

It means that the original column variance could substantially influence the result of random rotation. However, $E[K_{(i,i)}]$ is not the only factor determining the final privacy guarantee. We should also look at the variance of $K_{(i,i)}$. If the variance is considerably large, we still have great chance to get a rotation with large $K_{(i,i)}$ in a set of sample random rotations, and the larger the variance is, the more likely the randomly generated rotation matrices can provide a high privacy level. With the simplicity assumption, we can also roughly estimate the factors that contribute to the variance.

$$\begin{aligned} Var(K_{(i,i)}) &\sim \sum_{i=1}^{d}\sum_{j=1}^{d} Var(r_{ij})Var(r_{ik})c_{ij}^2 + 4\sum_{j=1}^{d} Var(r_{ij})c_{ij}^2 \\ &\sim O\left(1/d^2\sum_{i=1}^{d}\sum_{j=1}^{d} c_{ij}^2 + 4/d\sum_{j=1}^{d} c_{ij}^2\right). \end{aligned} \tag{9}$$

The above result shows that the variance is approximately related to the average of the squared covariance entries, with more influence from the row i of covariance matrix.

A simple method is to select the best rotation matrix among a bunch of randomly generated rotation matrices. But we can do better or be more efficient in a limited number of iterations. In Eq. 8, we also notice that the ith row vector of rotation matrix, i.e., the values r_{i*}, plays a dominating role in calculating the metric. Hence, it is possible to simply swap the rows of R to locally improve the overall privacy guarantee, which drives us to propose a row-swapping based fast local optimization method for finding a better rotation from a given rotation matrix. This method can significantly reduce the search space and thus provides better efficiency. Our experiments show that, with the local optimization, the minimum privacy level can be increased by about 10% or more. We formalize the swapping-maximization method as follows. Let $\{(1), (2), \ldots, (d)\}$ be a permutation of the sequence $\{1, 2, \ldots, d\}$. Let the importance level of privacy preserving for the columns be $\mathbf{w} = (w_1, w_2, \ldots, w_d)$. The goal is to find the permutation of rows of a given rotation matrix that results in a new rotation matrix that maximizes the minimum or average privacy guarantee.

$$argmax_{\{(1),(2),\ldots,(d)\}} \left\{ \min_{1 \le i \le d} \left\{ \left(\sum_{j=1}^{d} \sum_{k=1}^{d} r_{(i)j} r_{(i)k} c_{kj} - 2 \sum_{j=1}^{d} r_{(i)j} c_{ij} + c_{ii} \right) / w_i \right\} \right\} \tag{10}$$

Since the matrix R' generated by swapping the rows of R is still a rotation matrix, the above local optimization step will not change the rotation-invariance property of the discussed classifiers.

Attacks to Rotation Center The basic rotation perturbation uses the origin as the rotation center. Therefore, the points closely around the origin are still around the origin after the perturbation (as Fig. 7 shows), which leads to weaker privacy protection about these points. We address this problem with random translation so that the weakly perturbed points around the rotation center are not detectable due to the randomness of the rotation center. Attacks to translation perturbation will depend on the success of the attack to rotation perturbation, which will be described in later sections.

6.2.2 *Privacy analysis on ICA-based attack*

The unified privacy metric evaluates the privacy guarantee and the resilience against the first type of privacy attack – the naive inference. Considering the reconstruction-based inference, we identify that Independent Component Analysis (ICA) [22] could be the most powerful one to estimate the original dataset X, if more column statistics are known by the attacker. We dedicate this section to analyze the ICA-based attacks with the unified privacy metric.

Requirements of Effective ICA ICA is a fundamental problem in signal processing which is highly effective in several applications such as blind source separation [22] of mixed electro-encephalographic(EEG) signals, audio signals and the analysis of functional magnetic resonance imaging (fMRI) data. Let matrix X composed by source signals, where row vectors represent source signals. Suppose we can observe the mixed signals X', which is generated by linear transformation $X' = AX$. The ICA model can be applied to estimate the independent components (the row vectors) of the original signals X, from the mixed signals X', if the following conditions are satisfied:

1. The source signals are independent, i.e., the row vectors of X are independent;
2. All source signals must be non-Gaussian with possible exception of one signal;
3. The number of observed signals, i.e. the number of row vectors of X', must be at least as large as the independent source signals.
4. The transformation matrix A must be of full column rank.

For rotation matrices and full rank random projection, the 3rd and 4th conditions are always satisfied. However, the first two conditions, especially the independency condition, although practical for signal processing, seem not very common in data mining. Concretely, there are a few basic difficulties in applying the above ICA-based attack to the rotation-based perturbation. First of all, if there is significant dependency between any attributes, ICA fails to precisely reconstruct the original data, which thus cannot be used to effectively detect the private information. Second, even if ICA can be done successfully, the order of the original independent components cannot be preserved or determined through ICA [22].

Formally, any permutation matrix P and its inverse P^{-1} can be substituted in the model to give $X' = AP^{-1}PX$. ICA could possibly give the estimate for some permutated source PX. Thus, we cannot identify the particular column if the original column distributions are unknown. Finally, even if the ordering of columns can be identified, ICA reconstruction does not guarantee to preserve the variance of the original signal − the estimated signal may scale up the original one, but we do not know how much it scales, without knowing the statistical property of the original column.

In summary, without the necessary knowledge about the original data set, the attacker cannot simply use the ICA reconstruction. In case that attackers know enough distributional information that includes the maximum/minimum values and the probability density functions (PDFs) of the original columns, the effectiveness of ICA reconstruction will totally depend on the independency condition of the original columns. We observed in experiments that, since pure independency does not exist in the real data sets, we can still tune the rotation perturbation so that we can find one resilient enough to ICA-based attacks, even though the attacker knows the column statistics. In the following, we analyze how the sophisticated ICA-based attacks can be done and develop a simulation-based method to evaluate the resilience of a particular perturbation.

ICA Attacks with Known Column Statistics When the basic statistics, such as the max/ min values and the PDF of each column are known, ICA data reconstruction can possibly be done more effectively. We assume that ICA is quite effective to the data set (i.e., the four conditions are approximately satisfied) and the column PDFs are *distinctive*. Then, the reconstructed columns can be approximately matched to the original columns by comparing the PDFs of the reconstructed columns and the original columns. When the maximum/minimum values of columns are known, the reconstructed data can be scaled to the proper value ranges. We define an enhanced attack with the following procedure.

1. Running ICA algorithm to get a reconstructed data;
2. Estimate column distributions for the reconstructed columns, and for each reconstructed column find the closest match to the original column by comparing their column distributions;
3. Scale the columns with the corresponding maximum/minimum values of the original columns;

Note if the four conditions for effective ICA are exactly satisfied and the basic statistics and PDFs are all known, the basic rotation perturbation approach will not work. However, in practice, since the independency conditions are not all satisfied for most data sets in classification, we observed that different rotation perturbations may result in different quality of privacy, and it is possible to find one rotation that is considerably resilient to the enhanced ICA-based attacks. For this purpose, we can simulate the enhanced ICA attack to evaluate the privacy guarantee of a rotation perturbation. Concretely, it can be done in the following steps.

First step is called “PDF Alignment”. We need to calculate the similarity between the PDF of the original column and that of the reconstructed data column to find the best matches between the two sets of columns. A direct method is to calculate the difference between the two PDF functions. Let $f(x)$ and $g(x)$ be the original PDF and the PDF of the reconstructed column, respectively. A typical method to define the difference of PDFs employs the following function.

$$\Delta PDF = \int |f(x) - g(x)| \mathrm{d}x \tag{11}$$

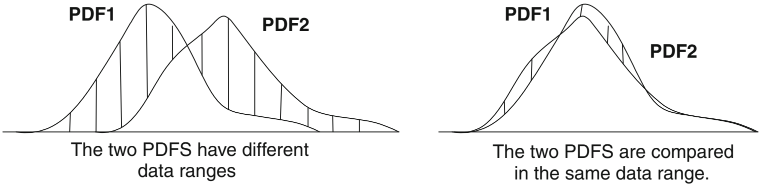
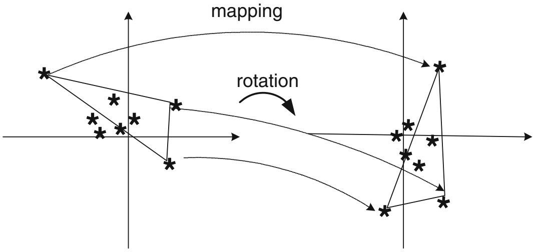

PDF1
PDF2
The two PDFS have different data ranges
PDF1
PDF2
The two PDFS are compared in the same data range.

Fig. 8 Comparing PDFs in different ranges results in large error (the *lined areas* are calculated as the difference between the PDFs)

Fig. 9 Using known points and distance relationship to infer the rotation matrix



In practice, for easy calculation we discretize the PDF into bins. It is then equivalent to use the discretized version: $\sum_{i=1}^{n} |f(b_i) - g(b_i)|$, where b_i is the discretized bin i. The discretized version is easy to implement by comparing the two histograms with a same number of bins. However, the evaluation is not accurate if the values in the two columns are not in the same range as shown in Fig. 8. Hence, the reconstructed PDF needs to be translated to match the range, which requires to know the maximum/minimum values of the original column. Since the original column is already scaled to [0, 1] in calculating unified privacy metric, we can just scale the reconstructed data to [0, 1], making it consistent with the normalized original data (Sect. 6.1.1). Meanwhile, this also scales the reconstructed data down so that the variance range is consistent with the original column. As a result, after the step of PDF Alignment, we can directly calculate the privacy metrics between the matched columns to measure the privacy quality.

Without loss of generality, we suppose that the level of confidence for an attack is primarily based on the PDF similarity between the two matched columns. Let O be the reconstruction of the original data set X. $\Delta PDF(\mathbf{O}_i, \mathbf{X}_j)$ represents the PDF difference of the column i in X and the column j in O. Let $\{(1), (2), \ldots, (d)\}$ be a permutation of the sequence $\{1, 2, \ldots, d\}$, which means a match from the original column i to (i). Let an optimal match minimize the sum of PDF differences of all pairs of matched columns. We define the minimum privacy guarantee based on the optimal match as follows.

$$p^{\min} = \min \left\{ \frac{1}{w_k} priv(\mathbf{X}_k, \mathbf{O}_{(k)}), 1 \le k \le d \right\} \tag{12}$$

where $\{(1), (2), \ldots, (d)\} = argmin_{\{(1),(2),\ldots,(d)\}} \sum_{i=1}^{d} \Delta PDF(\mathbf{X}_i, \mathbf{X}_{(i)})$. Similarly, we can define the average privacy guarantee based on an optimal match.

With the above multi-column metric, we are able to estimate how resilient a rotation perturbation is to the ICA-based attack equipped with the known column statistics. We observed in experiments that, although the ICA method may effectively reduce the privacy guarantee for certain rotation perturbations, we can always find some rotation matrices so that they can provide satisfactory privacy guarantee to ICA-based attacks.

6.2.3 Attacks to translation perturbation

Previously, we use random translation to address the weak protection on the points around the rotation center. We will see how translation perturbation is attacked if the ICA-based attack is applied.

Let each dimensional value of the random translation vector $\mathbf{t}$ is uniformly drawn from the range [0, 1], so that the center hides in the normalized data space. The perturbation can be represented as

$$f(X) = RX + \Psi = R(X + R^{-1}\Psi)$$

It is easy to verify that $T = R^{-1}\Psi$ is also a translation matrix. An effective attack to estimate the translation component should be based on the ICA inference to R and then remove the component $R^{-1}\Psi$ based on the unknown distribution of X. Concretely, the process can be described as follows.

By applying ICA attack, the estimate to $X + T$ is $\widehat{X + T} = \hat{R}^{-1}P$. Suppose that the original column i has maximum and minimum values $\max_i$ and $\min_i$, respectively, and $\hat{R}^{-1}P$ has $\max'_i$ and $\min'_i$, respectively. Since translation does not change the shape of column PDFs, we can align the column PDFs first. As scaling is one of the major effect of ICA estimation, we rescale the reconstructed column with some factor s, which can be estimated by $s \approx \frac{\max'_i - \min'_i}{\max_i - \min_i}$. Then, the column i of $\hat{R}^{-1}P$ is scaled down to the same span of X by the factor s. Then, we can extract the translation t_i for column i with

$$\hat{t}_i \approx \min'_i \times s - \min_i$$

Since the quality of the estimation is totally dependent on that of ICA reconstruction to rotation perturbation, a good rotation perturbation will protect translation perturbation as well. We will show some experimental results to see how well we can protect the translation component.

6.2.4 Privacy analysis on distance-inference attack

In the previous section, we have discussed naive-inference attacks and ICA-based attacks. In the following discussion, we assume that, besides the information necessary to perform these two kinds of attacks, the attacker manages to get more knowledge about the original data set: s/he also knows at least $d + 1$ original data points, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{d+1}\}$, d points of which are also linearly independent. Since the basic geometric perturbation preserves the distances between the points, the attacker can possibly find the mapping between these points and their images in the perturbed data set, $\{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{d+1}\}$, if the point distribution is peculiar, e.g. the points are outliers (as Fig. 9 shows). With the known mapping the rotation component R and translation component $\mathbf{t}$ can be calculated consequently. There is also discussion about the scenario that the attacker knows less than d points [30].

The mapping might be identified precisely for low-dimensional small data sets (<4 dimensions). With considerable cost, it is not impossible for higher dimensional larger data sets by simple exhaustive search if the known points have special distribution. There may have multiple matches, but the threat can be substantial.

So far we have assumed the attacker has obtained the right mapping between the known points and their images. In order to protect from the distance-inference attack, we use the noise component Δ to protect geometric perturbation $- G(X) = RX + \Psi + \Delta$. After we append the distance perturbation component, we have the original points and their maps be

$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{d+1}\} \rightarrow \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{d+1}\}, o_i = Rx_i + \mathbf{t} + \varepsilon_i$, where ε_i is the noise. The attacker can perform a linear regression–based estimation method.

1. R is estimated with the known mapping. The translation vector $\mathbf{t}$ can be canceled from the perturbation and we get d equations: $o_i - o_{d+1} = R(x_i - x_{d+1}) + \varepsilon_i - \varepsilon_{d+1}$, $1 \le i \le d$. Let $\bar{O} = [o_1 - o_{d+1}, o_2 - o_{d+1}, \ldots, o_d - o_{d+1}]$, $\bar{X} = [x_1 - x_{d+1}, x_2 - x_{d+1}, \ldots, x_d - x_{d+1}]$, and $\bar{\varepsilon} = [\varepsilon_1 - \varepsilon_{d+1}, \varepsilon_2 - \varepsilon_{d+1}, \ldots, \varepsilon_d - \varepsilon_{d+1}]$. The equations are unified to $\bar{O} = R\bar{X} + \bar{\varepsilon}$, and estimating R becomes a linear regression problem. The minimum variance unbiased estimator for R is $\hat{R} = \bar{O}'\bar{X}(\bar{X}'X)^{-1}$ [20].
2. With $\hat{R}$, the translation vector t can also be estimated. Since $o_i - Rx_i - \varepsilon_i = \mathbf{t}$ and ε_i has mean value 0, with $\hat{R}$ the attacker has the estimate of $\mathbf{t}$ as $\hat{\mathbf{t}} = \frac{1}{d+1}\{\sum_{i=1}^{d+1}(o_i - \hat{R}x_i) - \sum_{i=1}^{d+1}\varepsilon_i\} \approx \frac{1}{d+1}\sum_{i=1}^{d+1}(o_i - \hat{R}x_i)$. $\hat{\mathbf{t}}$ will have certain variance brought by the components $\hat{R}$ and ε_i.
3. Finally, the original data X can be estimated as follows. As $O = RX + \Psi + \Delta$, using the estimators $\hat{R}$ and $\hat{\Psi} = [\hat{\mathbf{t}}, \ldots, \hat{\mathbf{t}}]$, we get $\hat{X} = \hat{R}^{-}1(O - \hat{\Psi})$. Due to the variance introduced by $\hat{R}$, $\hat{\Psi}$, and Δ, the attacker may need to run several times to get the average of estimated $\hat{X}$, in practice.

By simulating the above process, we are able to estimate the effectiveness of the added noise. As we have discussed, as long as the geometric boundary is preserved, the geometric perturbation with noise addition can preserve the model accuracy. We have formally analyzed the effect of the noise component to model quality in Sect. 5.5. We will further study the relationship between the noise level, the privacy guarantee, and the model accuracy in experiments.

6.2.5 Privacy analysis on other attacks

We have studied a few attacks, according to the different levels of knowledge that an attacker may have. There are also studies about the extreme case that the attacker can know a considerable number of points ($\gg d$)in the original data set. In this case, classical methods, such as Principle Component Analysis (PCA) [30] and ICA [19], can be used to reconstruct the original data set with the higher order statistical information derived from both the known points and the perturbed data. In order to make these methods effective, the known points should be representative for the original data distribution, so that higher order statistics can be preserved, such as the covariance matrix of the original data set that both PCA- and ICA-based methods depend on. As a result, what portion of samples are known by the attacker and how different the known sample distribution is from the original one become the important factor for the success of attacks. Most importantly, these attacks become less meaningful in practice: when a large number of points have been cracked, it is too late to protect data privacy and security. In addition, outliers in the data set may be easily under attacks, if additional knowledge about the original outliers is available. Further study should be performed on the outlier-based attacks. We will leave these issues for future study.

6.2.6 A randomized algorithm for finding a better perturbation

We have discussed the unified privacy metric for evaluating the quality of a random geometric perturbation. Three kinds of inference attacks are analyzed under the framework of multi-column privacy evaluation, based on which we design an algorithm to choose good geometric perturbations that are resilient to the discussed attacks. In addition, the algorithm itself, even published, should not be a weak point in privacy protection. Since a deterministic algorithm in optimizing the perturbation may also provide extra clues to privacy attackers, we try to bring some randomization into the optimization process.

Algorithm 1 runs in a given number of iterations, aiming at maximizing the *minimum privacy guarantee*. At the beginning, a random translation is selected. In each iteration, the algorithm randomly generates a rotation matrix. Local maximization of variance through swapping rows is then applied to find a better rotation matrix. And then, the candidate rotation matrix is tested by the ICA-based attacks 6.2.2 assuming the attacker knows column statistics. The rotation matrix is accepted as the currently best perturbation, if it provides higher minimum privacy guarantee in terms of both naive estimation and ICA-based attacks than the previous perturbations. Finally, the noise component is appended to the perturbation, so that the distance-inference attack cannot reduce the privacy guarantee to a safety level ϕ, e.g., $\phi = 0.2$. Algorithm 1 outputs the rotation matrix R_t, the random translation matrix Ψ, the noise level σ^2, and the corresponding minimum privacy guarantee. If the privacy guarantee is lower than the anticipated threshold, the data owner can choose not to release the data. Note that this optimization process is applied to a sample set of the data. Therefore, the cost will be manageable even for very large original data set.

Algorithm 1 Finding a Good Perturbation ($X_{d\times N}$, $\mathbf{w}$, m)

```
Input: $X_{d\times N}$:the original data set, $\mathbf{w}$: weights of attributes in privacy evaluation, $m$: the number of iterations.
Output: $R_t$: the selected rotation matrix, $\Psi$: the random translation, $\sigma^2$: the noise level, $p$: privacy guarantee
calculate the covariance matrix $C$ of $X$;
$p = 0$, and randomly generate the translation $\Psi$;
for Each iteration do
  randomly generate a rotation matrix $R$;
  swapping the rows of $R$ to get $R'$, which maximizes $\min_{1\le i\le d}\{\frac{1}{w_i}(Cov(R'X - X)_{(i,i)}\}$;
  $p_0 =$ the privacy guarantee of $R'$, $p_1 = 0$;
  if $p_0 > p$ then
    generate $\hat{X}$ with ICA;
    $\{(1), (2), \ldots, (d)\} = argmin_{\{(1),(2),\ldots,(d)\}} \sum_{i=1}^{d} \Delta PDF(X_i, O_{(i)})$
    $p_1 = \min_{1\le k\le d} \frac{1}{w_k} Priv(X_k, O_{(k)})$
  end if
  if $p < \min(p_0, p_1)$ then
    $p = \min(p_0, p_1)$, $R_t = R'$;
  end if
end for
$p_2 =$ the privacy guarantee to the distance-inference attack with the perturbation $G(X) = R_t X + \Psi + \Delta$.
Tune the noise level $\sigma^2$, so that $p_2 \ge p$ if $p < \phi$ or $p_2 > \phi$ if $p < \phi$.
```

7 Experiments

We design four sets of experiments to evaluate the geometric perturbation approach. The first set is designed to show that the discussed classifiers are invariant to rotations and translations. In this set of experiments, general linear transformations, including dimensionality-preserving transformation and projection transformation are also investigated to see the advantage of distance preserving transformations. The second set shows the optimization of the privacy guarantee in geometric perturbation without the noise component, in terms of the naive-inference attack and the ICA-based attack. In the third set of experiments, we explore the relationship between the intensity of the noise component, the privacy guarantee, and the model accuracy, in terms of distance-inference attack. Finally, we compare the overall privacy guarantee provided by our geometric perturbation and another multidimensional perturbation—condensation approach. All datasets used in the experiments can be found in UCI machine learning database.[4]

7.1 Classifiers invariant to rotation perturbation

In this experiment, we verify the invariance property of several classifiers discussed in Sect. 5.1 to rotation perturbation. Three classifiers: kNN classifier, SVM classifier with RBF kernel, and perceptron are used as the representatives. To show the advantage of distance preserving transformations, we will test the invariance property of dimensionality-preserving general linear transformation and projection perturbation.

Each data set is randomly rotated 10 times in the experiment. Each of the 10 resultant data sets is used to train and cross-validate the classifiers. The reported numbers are the average of the 10 rounds of tests. We calculate the difference of model accuracy, between the classifier trained with the original data and those trained with the rotated data.

In the Table 1, 'orig' is the classifier accuracy to the original datasets, 'R' denotes the result of the classifiers trained with rotated data, and the numbers in 'R' columns are the performance difference between the classifiers trained with original and rotated data, for example, "-1.0 ± 0.2" means that the classifiers trained with the rotated data have the accuracy rate 1.0% lower than the original classifier on average, and the standard deviation is 0.2%. We use single-perceptron classifiers in the experiment. Therefore, the data sets having more than two classes, such as "E.Coli", "Iris" and "Wine" data sets, are not evaluated for perceptron classifier. 'A' means arbitrarily generated nonsingular linear perturbations that preserves the dimensionality of the original data set. From this result, we can see that rotation perturbation almost fully preserves the model accuracy for all of the three classifiers, except that perceptron might be sensitive to rotation perturbation for some data sets (e.g., "Breast-w"). Arbitrarily generated linear perturbations may downgrade the model accuracy a lot for some data sets, such as "Inonosphere" for kNN (−30.0%), and "Iris" for SVM (RBF) (−10.2%).

7.2 The effect of random projection to model accuracy

To see whether random projection can safely replace the rotation perturbation component in geometric data perturbation, we perform a set of experiments to check how model accuracy is affected by random projection. We implement the standard random projection method [41]. Random projection is defined as

$$G(\mathbf{x}) = \sqrt{\frac{k}{d}} R^T \mathbf{x},$$

where R is a $d \times k$ matrix with orthonormal columns. R can be generated in multiple methods. One simple method is to generate a random matrix with each element drawn from the standard normal distribution $N(0, 1)$ first, and then apply Gram-Schmidt process [6] to orthogonalize the columns. From the Johnson-Lindenstrauss Lemma [25], we understand that the number of projected dimensions is the major factor affecting the accuracy of models. We will look at this relationship in the experiment. For clear presentation, we will pick three data sets for each classifier that show great impact to accuracy. Similarly, each result is based on the average of ten rounds of different random projections.

For clear presentation, for each classifier modeling, we select only three data sets that show the most representative patterns. In Fig. 10, the x-axis is the difference between the projected

[4] http://www.ics.uci.edu/~mlearn/Machine-Learning.html.

Table 1 Experimental result on transformation-invariant classifiers

Dataset	*N*	*d*	*k*	kNN			SVM(RBF)			Perceptron		
				Orig	R	A	Orig	R	A	Orig	R	A
Breast-w	699	10	2	97.6	−0.5 ± 0.3	−0.5 ± 0.3	97.2	0 ± 0	−0.2 ± 0.2	80.4	−8.7 ± 0.3	−8.0 ± 1.5
Credit-a	690	14	2	82.9	0 ± 0.8	−0.7 ± 0.8	85.5	0 ± 0	+0.9 ± 0.3	73.6	−7.3 ± 1.0	−8.8 ± 0.7
Credit-g	1, 000	24	2	72.9	−1.2 ± 0.9	−1.8 ± 0.8	76.3	0 ± 0	+0.9 ± 0.9	75.1	0 ± 0	−0.2 ± 0.2
Diabetes	768	8	2	73.3	+0.4 ± 0.5	−0.4 ± 1.4	77.3	0 ± 0	−3.6 ± 1.0	68.9	0.0 ± 0.7	−2.5 ± 2.8
E.Coli	336	7	8	85.7	−0.4 ± 0.8	−2.0 ± 2.2	78.6	0 ± 0	−4.3 ± 1.5	–	–	–
Heart	270	13	2	80.4	+0.6 ± 0.5	−1.7 ± 1.1	84.8	0 ± 0	−2.3 ± 1.0	75.6	−5.2 ± 0.3	−3.9 ± 1.1
Hepatitis	155	19	2	81.1	+0.8 ± 1.5	0 ± 1.4	79.4	0 ± 0	+3.3 ± 1.9	77.4	−1.2 ± 0.4	−1.8 ± 2.4
Ionosphere	351	34	2	87.4	+0.5 ± 0.6	−30.0 ± 1.0	89.7	0 ± 0	+0.4 ± 0.4	75.5	−3.5 ± 1.0	−5.6 ± 1.0
Iris	150	4	3	96.6	+1.2 ± 0.4	−2.0 ± 2.1	96.7	0 ± 0	−10.2 ± 3.0	–	–	–
Tic-tac-toe	958	9	2	99.9	−0.3 ± 0.4	−8.3 ± 0.4	98.3	0 ± 0	+1.6 ± 6.9	76.4	−5.3 ± 0.0	−5.2 ± 0.1
Votes	435	16	2	92.9	0 ± 0.4	−12.6 ± 3.1	95.6	0 ± 0	−0.5 ± 1.9	90.7	−4.3 ± 1.0	−8.3 ± 4.9
Wine	178	13	3	97.7	0 ± 0.5	−2.0 ± 0.2	98.9	0 ± 0	−5.7 ± 1.3	–	–	–

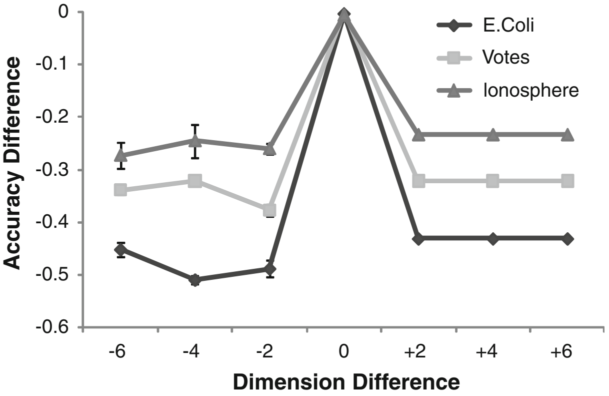


Fig. 10 The effect of projection perturbation to kNN

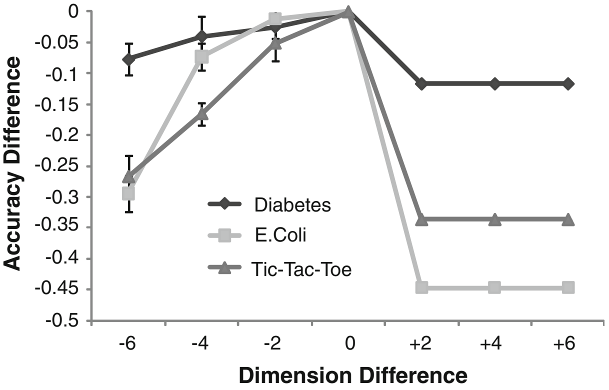


Fig. 11 The effect of projection perturbation to SVM

dimensions and the original dimensions and the y-axis is the difference between original model accuracy and the perturbed model accuracy (perturbed accuracy—original accuracy). Note that random projections that preserve dimensionality (dimension difference=0) is as same as rotation perturbation. It shows that the kNN model accuracy for the three data sets can decrease dramatically regardless of increased or decreased dimensionality. The numbers are the average of ten runs for each dimensionality setting. In Fig. 11, SVM models also show the model accuracy is significantly reduced with a dimensionality different to the original one. The "Diabetes" data is less affected by changed dimensionality. Interestingly, the perceptron models (Fig. 12) are less sensitive to changed dimensionality for some data sets such as "Diabetes" and "Heart", while very sensitive to others such as "BreastW". In general, the error caused by random projection perturbation that changes dimensionality is so large that the resultant models are not useful.

7.3 Effectiveness of translation perturbation

The effectiveness of translation perturbation is two-fold. First, we show that translation perturbation cannot be effectively estimated based on the discussed techniques. Then, we give complementary experimental results to show that the classifiers: SVMs with polynomial kernel and sigmoid kernel indeed invariant to translation perturbation.

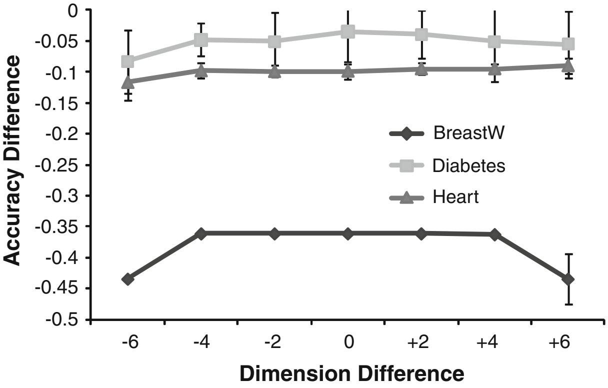


Fig. 12 The effect of projection perturbation to Perceptron

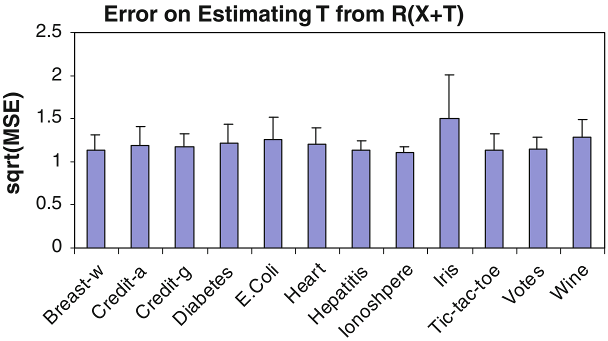


Fig. 13 Resilience to the attack to random translation

As we have mentioned, if the translation vector could be precisely estimated, the rotation center would be exposed. We applied the ICA-based attack to rotation center that is described in Sect. 6.2.3. The data in Fig. 13 shows SD ($\hat{\mathbf{t}} - \mathbf{t}$). Compared to the range of the elements in $\mathbf{t}$, i.e., [0, 1], the standard deviations are quite large, so we can conclude that random translation is also hard to estimate if we have optimized rotation perturbation in terms of ICA-based attacks.

SVMs with polynomial kernel, and sigmoid kernel, are also invariant to translation transformation. Table 2 lists the experimental result on random translation for the 12 data sets. We randomly translate each data set for ten times. The result is the average of the ten runs. For most data sets, the result shows zero or tiny deviation from the standard model accuracy.

7.4 Perturbation optimization against naive estimation and ICA-based attack

We run the randomized optimization algorithm and show how effective it can generate resilient perturbations[5]. Each column in the experimental data set is considered equally important in privacy evaluation. Thus, the weights are not included in evaluation.

[5] Since we slightly changed the definition of privacy guarantee from our SDM paper [9], we need to re-run the experiments that use this metric for comparison. Therefore, the numbers in this section can be slightly different from those in the paper [9].

Table 2 Experimental result on random translation

Dataset	SVM (polynomial)		SVM (sigmoid)	
	Orig	Tr	Orig	Tr
Breast-w	96.6	0 ± 0	65.5	0 ± 0
Credit-a	88.7	0 ± 0	55.5	0 ± 0
Credit-g	87.3	-0.4 ± 0.4	70	0 ± 0
Diabetes	78.5	0 ± 0.3	65.1	0 ± 0
E. coli	89.9	-0.1 ± 0.5	42.6	0 ± 0
Heart	91.1	-0.2 ± 0.2	55.6	0 ± 0
Hepatitis	96.7	-0.4 ± 0.3	79.4	0 ± 0
Ionosphere	98	$+0.3 \pm 0$	63.5	$+0.6 \pm 0$
Iris	97.3	0 ± 0	29.3	-1.8 ± 0.4
Tic-tac-toe	100	0 ± 0	65.3	0 ± 0
Votes	99.2	$+0.2 \pm 0.1$	65.5	-4.7 ± 0.6
Wine	100	0 ± 0	39.9	0 ± 0

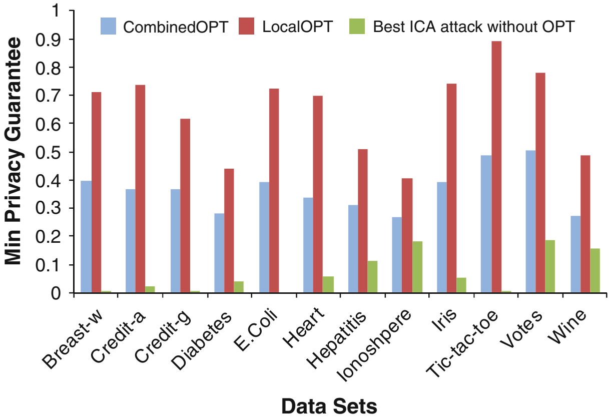


Fig. 14 Minimum privacy guarantee generated by local optimization, combined optimization, and the performance of ICA-based attack

Figures 14 and 15 summarize the evaluation of privacy quality on experimental datasets. The results are obtained in 50 iterations with the optimization algorithm described in Sect. 6.2.6. “LocalOPT” represents the locally optimized minimum privacy guarantee addressing naive estimation at a number of iterations. “Best ICA attack” is the worst perturbation that gives the best ICA attack performance, i.e., getting the lowest privacy guarantee among the perturbations tried in the rounds. “CombinedOPT”is the combined optimization result given by Algorithm 1 after a number of iterations. The above values are calculated with the proposed privacy metric based on the estimated data set and the original data set. The LocalOPT values can often reach a relatively high level after 50 iterations, which means that the swapping method is very efficient in locally optimizing the privacy quality in terms of naive estimation. In contrast, the best ICA attacks often result in very low privacy guarantee, which means some rotation perturbations are very weak to ICA-based attacks. Combined-

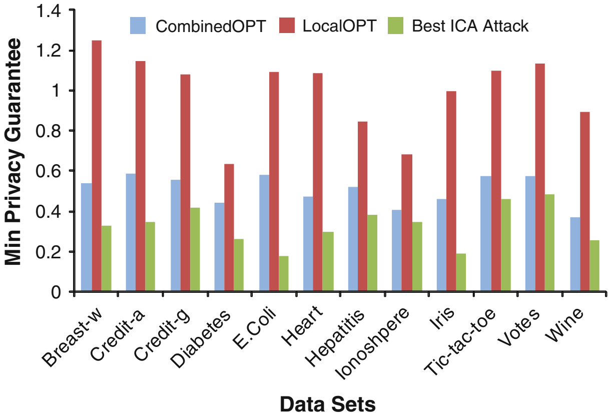


Fig. 15 Average privacy guarantee generated by local optimization, combined optimization, and the performance of ICA-based attack

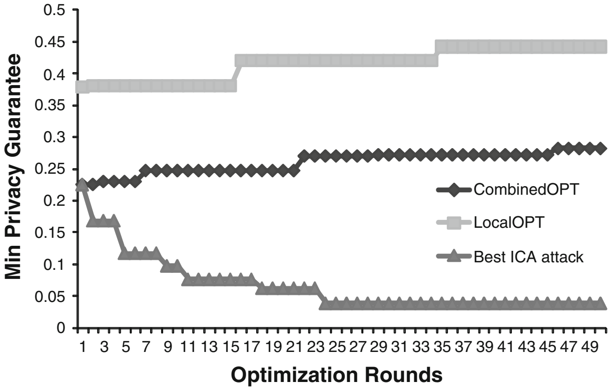


Fig. 16 Optimization of perturbation for diabetes data

OPT values are much higher than the corresponding ICA-based attacks, which supports our conjecture that we can always find one perturbation that is sufficiently resilient to ICA-based attacks in practice.

We also show the detail in the course of optimization for two data sets “Diabetes” and “Votes” in Figs. 16 and 17, respectively. For both data sets, the combined optimal result is between the curves of best ICA-attacks and the best local optimization result. Different data sets or different randomization processes may cause different change patterns of privacy guarantee in the course of optimization. However, we see after a few rounds the results are quickly stabilized round satisfactory privacy guarantee, which means the proposed optimization method is very efficient.

7.5 Distance perturbation: the tradeoff between privacy and model accuracy

Now we extend the geometric perturbation with random noise component : $G(X) = RX + \Psi + \Delta$, to address the potential distance-inference attacks. From the formal analysis, we know that the noise component Δ can conveniently protect the perturbation from distance-

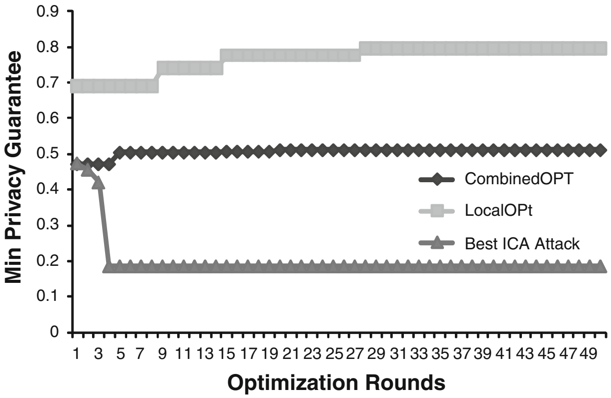


Fig. 17 Optimization of perturbation for votes data

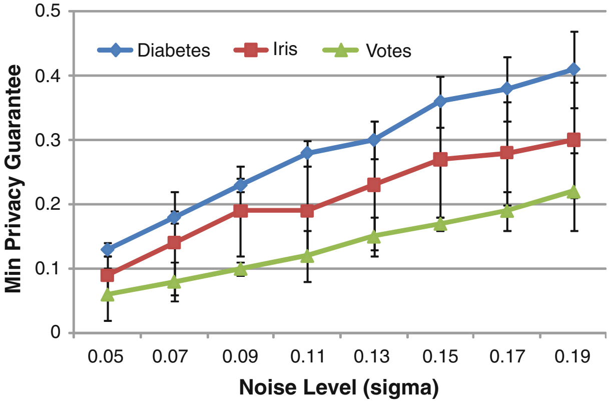


Fig. 18 The change of minimum privacy guarantee vs. the increase of noise level for the three datasets

inference attack. Intuitively, the higher the noise level is, the better the privacy guarantee. However, with the increasing noise level, the model accuracy could also be affected. In this set of experiments, we first study the relationship between the noise level, represented by its variance σ^2, and the privacy guarantee, and then the relationship between the noise level and the model accuracy.

Each known I/O attack is simulated by randomly picking a number of records (e.g., 5% of the total records) as the known records and then applying the estimation procedure discussed in Sect. 6.2.4. After running 500 runs of simulated attacks for each noise level, we get the average of minimum privacy guarantee. In addition, since the paper [21] showed that the PCA-based noise filtering technique may help reduce the noise for some noise perturbed data sets, we also simulate the PCA filtering method based on the described algorithm [21] and checked its effectiveness. The results show that in most cases (except for some noise levels for "Iris" data) when the number of principal components equals the number of the original dimensions (i.e., no noise reduction is applied), the attack is most effective. Since the PCA method cannot clearly distinguish the noise from other perturbation components, removing the smallest principal components will inevitably change the non-noise part as well. Figure 18 shows the best attacking results for different noise levels. Overall, the pri-

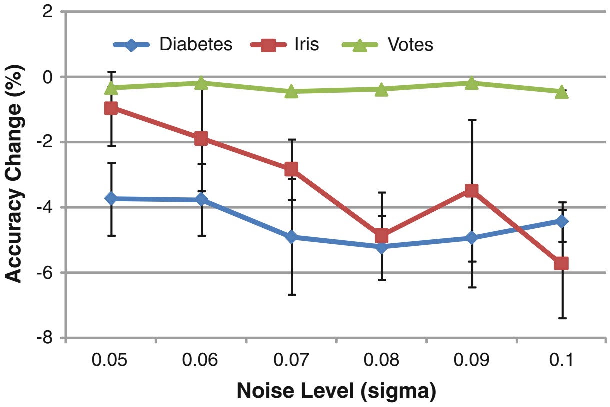


Fig. 19 The change of accuracy of KNN classifier vs. the increase of noise level

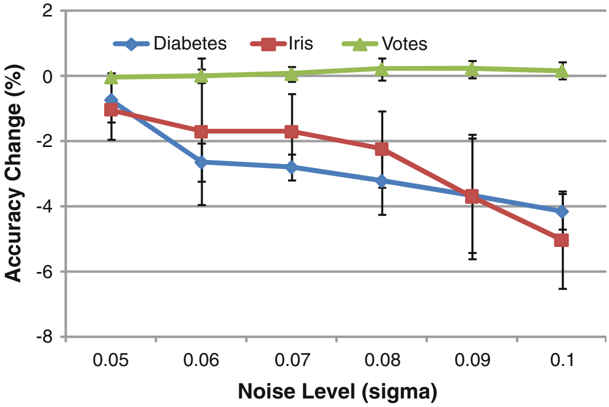


Fig. 20 The change of accuracy of SVM(RBF) classifier vs. the increase of noise level

vacy guarantee increases with the increase of noise level for all three data sets. At the noise level $\sigma = 0.1$, the privacy guarantee is between the range 0.1–0.2. Figures 19 and 20 show a trend of decreasing accuracy for KNN classifier and SVM (RBF kernel) classifier, respectively. However, with the noise level lower than 0.1, the accuracy of both classifiers is only reduced less than 6%, which is quite acceptable. Meanwhile, perceptron (Fig. 21) is less sensitive to different levels of noise intensity. We perform experiments on all data sets at the noise level $\sigma = 0.1$ to see how the model accuracy is affected by the noise component.

We summarize the privacy guarantees at the noise level 0.1 for all experimental data sets[6] in Fig. 22, and also the change of model accuracy for KNN, SVM(RBF), and Perceptron in Fig. 23. Among the three types of classifiers, KNN is the most stable one while perceptron classifiers are most sensitive to distance perturbation. Overall, the distance perturbation component may affect model accuracy, but it has much less impact on model accuracy. Last but not least, it is worth noting that the noise component can be removed if the data owner makes sure to securely store the original data. This important feature provides extra and valuable flexibility in geometric perturbation for data mining.

[6] "Ionosphere" is not included because the existence of nearly constant value in one column.

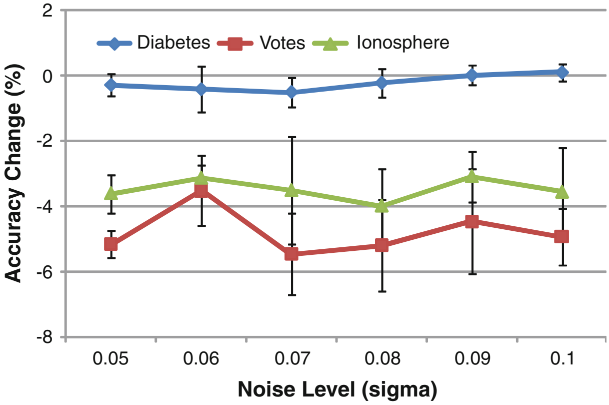


Fig. 21 The change of accuracy of perceptron classifier vs. the increase of noise level

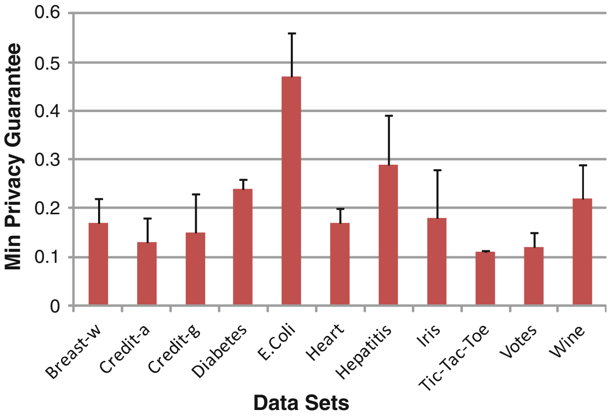


Fig. 22 Minimum privacy guarantee at the noise level $\sigma = 0.1$

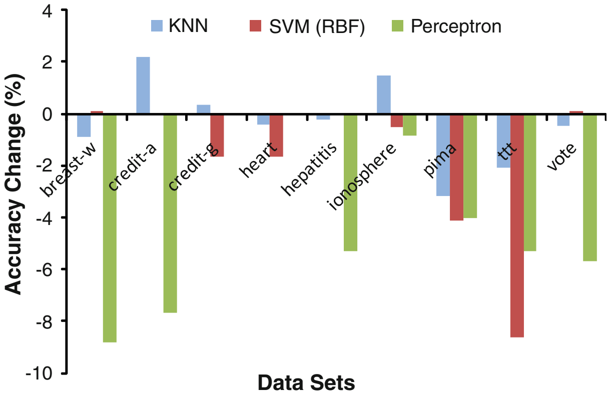


Fig. 23 The change of model accuracy at the noise level $\sigma = 0.1$

8 Conclusion

We present a random geometric perturbation approach to privacy preserving data classification. Random geometric perturbation, $G(X) = RX + \Psi + \Delta$, includes the linear combination of the three components: rotation perturbation, translation perturbation, and distance perturbation. Geometric perturbation can preserve the important geometric properties; thus, most data-mining models that search for geometric class boundaries are well preserved with the perturbed data. We proved that many data-mining models, including classifier, regression models, and clustering methods, are invariant to geometric perturbation.

Geometric perturbation perturbs multiple columns in one transformation, which introduces new challenges in evaluating the privacy guarantee for multidimensional perturbation. We propose a multi-column privacy evaluation model and design a unified privacy metric to address these problems. We also thoroughly analyze the resilience of the rotation perturbation approach against three types of inference attacks: naive-inference attacks, ICA-based attacks, and distance-inference attacks. With the privacy model and the analysis of attacks, we are able to construct a randomized optimization algorithm to efficiently find a good geometric perturbation that is resilient to the attacks. Our experiments show that the geometric perturbation approach not only preserves the accuracy of models, but also provides much higher privacy guarantee, compared to existing multidimensional perturbation techniques.

Acknowledgments The second author acknowledges the partial support from grants of NSF NetSE program, NSF CyberTrust program, an IBM SUR grant and a grant from Intel Research Council.

References

1. Aggarwal CC, Yu PS (2004) A condensation approach to privacy preserving data mining. In: Proceedings of international conference on extending database technology (EDBT), vol 2992. Springer, Heraklion, pp 183–199
2. Agrawal D, Aggarwal CC (2002) On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of ACM conference on principles of database systems (PODS). ACM, Madison
3. Agrawal R, Srikant R, (2000) Privacy-preserving data mining. In: Proceedings of ACM SIGMOD conference. ACM, Dallas
4. Amazon (n.d.) Applications hosted on amazon clouds. http://aws.amazon.com/solutions/case-studies/
5. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: a berkeley view of cloud computing. Technical report, University of Berkerley
6. Bhatia R (1997) Matrix analysis. Springer, Berlin
7. Bruening PJ, Treacy BC (2009) Privacy, security issues raised by cloud computing. BNA Privacy Security Law Report 8(10)
8. Chen K, Liu L (2005) A random rotation perturbation approach to privacy preserving data classification In: Proceedings of international conference on data mining (ICDM). IEEE, Houston
9. Chen K, Liu L, Sun G (2007) Towards attack-resilient geometric data perturbation. In: SIAM data mining conference
10. Clifton C (2003) Tutorial: Privacy-preserving data mining, In Proceedings of ACM SIGKDD Conference
11. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
12. Evfimievski A, Gehrke J, Srikant R (2003) Limiting privacy breaches in privacy preserving data mining. In: Proceedings of ACM conference on principles of database systems (PODS)
13. Evfimievski A, Srikant R, Agrawal R, Gehrke J (2002) Privacy preserving mining of association rules. In: Proceedings of ACM SIGKDD conference

14. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232
15. Fung BC, Wang K, Chen R, Yu PS (2010) Privacy-preserving data publishing: a survey on recent developments. ACM Comput Surv 42(4):1–53
16. Gallier J (2000) Methods and applications for computer science and engineering. Springer, New York
17. Google (n.d.) Google appengine gallery. http://appgallery.appspot.com/
18. Guo S, Wu X (2007) Deriving private information from arbitrarily projected data. In: Proceedings of the 11th European conference on principles and practice of knowledge Discovery in databases (PKDD07). Warsaw, Poland
19. Guo S, Wu X, Li Y (2008) Determining error bounds for spectral filtering based reconstruction methods in privacy preserving data mining. Knowl Inform Syst 17(2):217–240
20. Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer, Berlin
21. Huang Z, Du W, Chen B, (2005) Deriving private information from randomized data. In: Proceedings of ACM SIGMOD conference
22. Hyvarinen A, Karhunen J, Oja E (2001) Independent component analysis. Wiley, London
23. Jain A, Murty M, Flynn P (1999) Data clustering: a review. ACM Comput Surv 31:264–323
24. Jiang T (2006) How many entries in a typical orthogonal matrix can be approximated by independent normals. Ann Prob 34(4):1497–1529
25. Johnson WB, Lindenstrauss J (1984) Extensions of lipshitz mapping into hilbert space. Contemp Math 26:189–206
26. Kargupta H, Datta S, Wang Q, Sivakumar K (2003) On the privacy preserving properties of random data perturbation techniques. In: Proceedings of international conference on data mining (ICDM)
27. LeFevre K, DeWitt DJ, Ramakrishnan R (2006) Mondrain multidimensional k-anonymity. In: Proceedings of IEEE international conference on data engineering (ICDE)
28. Lehmann EL, Casella G (1998) Theory of point estimation. Springer, Berlin
29. Lindell Y, Pinkas B (2000) Privacy preserving data mining. J Cryptol 15(3):177–206
30. Liu K, Giannella C, Kargupta H (2006) An attacker's view of distance preserving maps for privacy preserving data mining In: European conference on principles and practice of knowledge discovery in databases (PKDD). Berlin, Germany
31. Liu K, Kargupta H, Ryan J (2006) Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. IEEE Trans Knowl Data Eng 18(1):92–106
32. Luo H, Fan J, Lin X, Zhou A, Bertino E (2009) A distributed approach to enabling privacy-preserving model-based classifier training. Knowl Inform Syst 20(2):157–185
33. McLachlan G, Peel D (2000) Finite mixture models. Wiley, London
34. Oliveira SRM, Zaïane OR (2004) Privacy preservation when sharing data for clustering. In: Proceedings of the international workshop on secure data management in a connected world. Toronto, Canada, pp 67–82
35. Oliveira SR, Zaiane OR (2010) Privacy preserving clustering by data transformation. J Inform Data Manag (JIDM) 1(1):67–82
36. Sadun L (2001) Applied linear algebra: the decoupling principle. Prentice Hall, Englewood Cliffs
37. Stewart G (1980) The efficient generation of random orthogonal matrices with an application to condition estimation. SIAM J Num Anal 17:403–409
38. Sweeney L (2002) k-anonymity: a model for protecting privacy. Int J Uncert Fuzz Knowl Based Syst 10(5):557–570
39. Teng Z, Du W (2009) A hybrid multi-group approach for privacy-preserving data mining. Knowl Inform Syst 19(2):133–157
40. Vaidya J, Clifton C (2003) Privacy preserving k-means clustering over vertically partitioned data. In: Proceedings of ACM SIGKDD conference
41. Vempala SS (2005) The random projection method. American Mathematical Society, Providence

Author Biographies

Keke Chen is currently an assistant professor in Wright State University, Dayton OH, USA, where he directs the Data Intensive Analytics and Computing lab. He received his PhD degree in Computer Science from the College of Computing at Georgia Tech, Atlanta GA, USA, in 2006. Keke's research focuses on data privacy protection, visual analytics, and distributed data intensive scalable computing, including web search, data mining and visual analytics. From 2002 to 2006, Keke worked with Dr. Ling Liu in the Distributed Data Intensive Systems Lab at Georgia Tech, where he developed a few well-known research prototypes, such as the VISTA visual cluster rendering and validation system, the iVIBRATE framework for large-scale visual data clustering, the "Best K" cluster validation method for categorical data clustering, and the geometric data perturbation approach for outsourced data mining. From 2006 to 2008, he was a senior research scientist in Yahoo! Labs, Santa Clara, CA, working on international web search relevance and data mining algorithms for large distributed data sets on cloud computing. In Yahoo! Labs, he developed the tree adaptation method for ranking function adaptation.

Ling Liu is a full Professor in the School of Computer Science at Georgia Institute of Technology. There she directs the research programs in Distributed Data Intensive Systems Lab (DiSL), examining various aspects of data intensive systems with the focus on performance, availability, security, privacy, and energy efficiency. Prof. Liu and her students have released a number of open source software tools, including WebCQ, XWRAPElite, PeerCrawl, GTMobiSim. Currently she is the lead PI on two NSF sponsored research projects: Location Privacy in Mobile and Wireless Internet Computing, and Privacy Preserving Information Networks for Healthcare Applications. Prof. Liu has published over 250 International journal and conference articles in the areas of databases, distributed systems, and Internet Computing. She is a recipient of the best paper award of ICDCS 2003, WWW 2004, the 2005 Pat Goldberg Memorial Best Paper Award, and 2008 Int. conf. on Software Engineering and Data Engineering. Prof. Liu has served as general chair and PC chairs of numerous IEEE and ACM conferences in data engineering, distributed computing, service computing and cloud computing fields and is a co-editor-in-chief of the 5 volume Encyclopedia of Database Systems (Springer). She is currently on the editorial board of several international journals, such as Distributed and Parallel Databases (DAPD, Springer), Journal of Parallel and Distributed Computing (JPDC), IEEE Transactions on Service Computing (TSC), and Wireless Network (WINET, Springer). Dr. Liu's current research is primarily sponsored by NSF, IBM, and Intel.