Chapter 1

A SURVEY OF MULTIPLICATIVE PERTURBATION FOR PRIVACY PRESERVING DATA MINING

Keke Chen
College of Computing
Georgia Institute of Technology
kekechen@cc.gatech.edu

Ling Liu
College of Computing
Georgia Institute of Technology
lingliu@cc.gatech.edu

Abstract The major challenge of data perturbation is to achieve the desired balance between the level of privacy guarantee and the level of data utility. Data privacy and data utility are commonly considered as a pair of conflicting requirements in privacy-preserving data mining systems and applications. Multiplicative perturbation algorithms aim at improving data privacy while maintaining the desired level of data utility by selectively preserving the mining task and model specific information during the data perturbation process. By preserving the task and model specific information, a set of "transformation-invariant data mining models" can be applied to the perturbed data directly, achieving the required model accuracy. Often a multiplicative perturbation algorithm may find multiple data transformations that preserve the required data utility. Thus the next major challenge is to find a good transformation that provides a satisfactory level of privacy guarantee. In this chapter, we review three representative multiplicative perturbation methods: rotation perturbation, projection perturbation, and geometric perturbation, and discuss the technical issues and research challenges. We first describe the mining task and model specific information for a class of data mining models, and the transformations that can (approximately) preserve the information. Then we discuss the design of appropriate privacy evaluation models for multiplicative perturbations, and give an overview of how we use the privacy evaluation model to measure the level of privacy guarantee in the context of different types of attacks.

1. Introduction

Data perturbation refers to a data transformation process typically performed by the data owners before publishing their data. The goal of performing such data transformation is two-fold. On one hand, the data owners want to change the data in a certain way in order to disguise the sensitive information contained in the published datasets, and on the other hand, the data owners want the transformation to best preserve those domain-specific data properties that are critical for building meaningful data mining models, thus maintaining mining task specific data utility of the published datasets.

Data perturbation techniques are one of the most popular models for privacy preserving data mining. It is especially useful for applications where data owners want to participate in cooperative mining but at the same time want to prevent the leakage of privacy-sensitive information in their published datasets. Typical examples include publishing micro data for research purpose or outsourcing the data to the third party data mining service providers. Several perturbation techniques have been proposed to date [4–1, 8, 3, 13, 14, 26, 35], among which the most popular one is the randomization approach that focuses on single-dimensional perturbation and assumes independency between data columns [4, 13]. Only recently, the data management community has shown some development on multi-dimensional data perturbation techniques, such as the condensation approach using k-nearest neighbor (kNN) method [1], the multi-dimensional K-anonymization using kd-tree [24], and the multiplicative data perturbation techniques [31, 8, 28, 9]. Compared to single-column-based data perturbation techniques that assume data columns to be independent and focus on developing single-dimensional perturbation techniques, multi-dimensional data perturbation aims at perturbing the data while preserving the *multi-dimensional information* with respect to inter-column dependency and distribution.

In this chapter, we will discuss multiplicative data perturbations. This category includes three types of particular perturbation techniques: Rotation Perturbation, Projection Perturbation, and Geometric Perturbation. Comparing to other multi-dimensional data perturbation methods, these perturbations exhibit unique properties for privacy preserving data classification and data clustering. They all preserve (or approximately preserve) distance or inner product, which are important to many classification and clustering models. As a result, the classification and clustering mining models based on the perturbed data through multiplicative data perturbation show similar accuracy to those based on the original data. The main challenge for multiplicative data perturbations thus is how to maximize the desired data privacy. In contrast, many other data perturbation techniques focus on seeking for the better trade-off between the

level of data utility and accuracy preserved and the level of data privacy guaranteed.

1.1 Data Privacy vs. Data Utility

Perturbation techniques are often evaluated with two basic metrics: level of privacy guarantee and level of model-specific data utility preserved, which is often measured by the loss of accuracy for data classification and data clustering. An ultimate goal for all data perturbation algorithms is to optimize the data transformation process by maximizing both data privacy and data utility achieved. However, the two metrics are typically representing two conflicting goals in many existing perturbation techniques [4, 3, 12–1].

Data privacy is commonly measured by the difficulty level in estimating the original data from the perturbed data. Given a data perturbation technique, the higher level of difficulty in which the original values can be estimated from the perturbed data, the higher level of data privacy this technique supports. In [4], the variance of the added random noise is used as the level of difficulty for estimating the original values as traditionally used in statistical data distortion [23]. However, recent research [12, 3] reveals that variance of the noise is not an effective indicator for random noise addition. In addition, [22] shows that the level of data privacy guaranteed is also bounded to the types of special attacks that can reconstruct the original data from the perturbed data and noise distribution. k-Anonymization is another popular way of measuring the level of privacy, originally proposed for relational databases [34], by enabling the effective estimation of the original data record to a k-record group, assuming that each record in the k-record group is equally protected. However, recent study [29] shows that the privacy evaluation of k-Anonymized records is far more complicated than this simple k-anonymization assumption.

Data utility typically refers to the amount of mining-task/model specific critical information preserved about the dataset after perturbation. Different data mining tasks, such as classification mining task vs. association rule mining, or different models for the same task, such as decision tree model vs. k-Nearest-Neighbor (kNN) classifier for classification, typically utilize different sets of data properties about the dataset. For example, the task of building decision trees primarily concerns the column distribution. Hence, the quality of preserving column distribution should be the key data utility to be maintained in perturbation techniques for decision tree model, as shown in the randomization approach [4]. In comparison, the kNN model relies heavily on the distance relationship, which is quite different from the column distribution. Furthermore, such task/model-specific information is often multidimensional. Many classification models typically concern the multidimensional information rather than single column distribution. Multi-dimensional perturbation techniques with

the focus on preserving the model-specific multidimensional information will be more effective for these models.

It is also interesting to note that the data privacy metric and the data utility metric are often contradictory rather than complimentary in many existing data perturbation techniques [4, 3, 12–1]. Typically data perturbation algorithms that aim at maximizing the level of data privacy often have to bear with higher information loss. The intrinsic correlation between the data privacy and the data utility raises a number of important issues regarding how to find a right balance between the two measures.

In summary, we identify three important design principles for multiplicative data perturbations. First, preserving the mining task and model-specific data properties is critical for providing better quality guarantee on both privacy and model accuracy. Second, it is beneficial if data perturbation can effectively preserve the task/model-specific data utility information, and avoid the need for developing special mining algorithms that can use the perturbed data as random noise addition requires. Third and most importantly, if one can develop a data perturbation technique that does not induce any lost of mining-task/model specific data utility, this will enable us to focus on optimizing perturbation algorithms by maximizing the level of data privacy against attacks, which ultimately leads to better overall quality of both data privacy and data utility.

1.2 Outline

In the remaining of the chapter we will first give the definition of multiplicative perturbation in Section 2. Specifically, we categorize multiplicative perturbations into three categories: rotation perturbation, projection perturbation, and geometric perturbation. Rotation perturbation is often criticized not resilient to attacks, while geometric perturbation is a direct enhancement to rotation perturbation by adding more components, such as translation perturbation and noise addition, to the original rotation perturbation. Both rotation perturbation and geometric Perturbation keep the dimensionality of dataset unchanged, while projection perturbation reduces the dimensionality, and thus incurs more errors in distance or inner product calculation.

One of the unique features that distinguish multiplicative perturbations from other perturbations is that it provides high guarantee on data utility in terms of data classification and clustering. Since many data mining models utilize distance or inner product, as long as such information is preserved, models trained on perturbed data will have similar accuracy to those trained on the original data. In Section 3, we define transformation-invariant classifiers and clustering models, the representative models to which multiplicative perturbations are applied.

Evaluation of privacy guarantee for perturbations is an important component in the analysis of multiplicative perturbation. In Section 4, we review a set of privacy metrics specifically designed for multiplicative perturbations. We argue that in multidimensional perturbation, the values of multiple columns should be perturbed together and the evaluation metrics should be unified for all columns. We also describe a general framework for privacy evaluation of multiplicative data perturbation by incorporating attack analysis.

We argue that attack analysis is a necessary step in order to accurately evaluate the privacy guarantee of any particular perturbation. In Section 5, we review a selection of known attacks to multiplicative perturbations based on different levels of attack's knowledge about the original dataset. By incorporating attack analysis under the general framework of privacy evaluation, a randomized perturbation optimization is developed and described in Section 5.5.

2. Definition of Multiplicative Perturbation

We will first describe the notations used in this chapter, and then describe three categories of multiplicative perturbations and their basic characteristics.

2.1 Notations

In privacy-preserving data mining, either a portion of or the entire data set will be perturbed and then exported. For example, in classification, the training data is exported and the testing data might be exported, too, while in clustering, the entire data for clustering is exported. Suppose that X is the exported dataset consisting of N data rows (records) and d columns (attributes, or dimensions). For presentation convenience, we use $X_{d\times N}$, $X = [\mathbf{x}_1 \dots \mathbf{x}_N]$, to denote the dataset, where a column $\mathbf{x}_i$ ($1 \leq i \leq N$) is a data tuple, representing a vector in the real space $\mathbb{R}^d$. In classification, each of such data tuples $\mathbf{x}_i$ also belongs to a predefined class, which is indicated by the class label attribute y_i. The class label can be nominal (or continuous for regression), and is public, i.e., privacy-insensitive.

For clear presentation, we can also consider X is a sample dataset from the d-dimension random vector $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_d]^T$. As a convention, we use bold lower case to represent vectors, bold upper case to represent random variables, and upper case to represent matrices or datasets.

2.2 Rotation Perturbation

This category does not cover traditional "rotations" only, but literally, it includes all orthonormal perturbations. A rotation perturbation is defined as following $G(X)$:

$$G(X) = RX$$

The matrix $R_{d\times d}$ is an orthonormal matrix [32], which has following properties. Let R^T represent the transpose of R, r_{ij} represent the (i,j) element of R, and I be the identity matrix. The rows and columns of R are orthonormal, i.e., 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 orthogonal. A random orthonormal matrix can be efficiently generated following the Haar distribution [33].

A key feature of rotation transformation is that it preserve the Euclidean distance of multi-dimensional points during the transformation. 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 also preserves the geometric shapes such as hyperplane and hyper curved surface in the multidimensional space [7]. We observed that since many classifiers look for geometric decision boundary, such as hyperplane and hyper surface, rotation transformation will preserve the most critical information for many classification models.

There are two ways to apply rotation perturbation. We can either apply it to the whole dataset X [8], or group columns to pairs and apply different rotation perturbations to different pairs of columns [31].

2.3 Projection Perturbation

Projection perturbation refers to the technique of projecting a set of data points from a high-dimensional space to a randomly chosen lower-dimensional subspace. Let $P_{k\times d}$ be a projection matrix.

$$G(X) = PX$$

Why can it also be used for perturbation? The rationale is based on the Johnson-Lindenstrauss Lemma [21].

Theorem 1. *For any $0 < \epsilon < 1$ and any integer n, let k be a positive integer such that $k \geq \frac{4\ln n}{\epsilon^2/2 - \epsilon^3/3}$. Then, for any set $\mathbf{S}$ of n data points in d dimensional*

space $\mathbb{R}^d$, there is a map f: $\mathbb{R}^d \to \mathbb{R}^k$ 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 $O(\frac{\log n}{\epsilon^2})$ -dimensional space, such that the pair-wise distance of any two points are maintained with small error. With large n (large dataset) and small ϵ (high accuracy in distance preservation), the ideal dimensionality might be large and may not be practical for the perturbation purpose. Furthermore, although this lemma implies that we can always find one good projection that approximately preserves distances for a particular dataset, the geometric decision boundary might still be distorted and thus the model accuracy is reduced. Due to the different distributions of dataset and particular properties of data mining models, it is challenging to develop an algorithm that can find random projections that preserves model accuracy well for any given dataset.

In paper [28] a method is used to generate random projection matrix. The process can be briefly described as follows. Let P be the projection matrix. Each entry $r_{i,j}$ of P is independent and identically chosen from some distribution with mean zero and variance σ^2. A row-wise projection is defined as

$$G(X) = \frac{1}{\sqrt{k\sigma}}PX$$

Let $\mathbf{x}$ and $\mathbf{y}$ be two points in the original space, and $\mathbf{u}$ and $\mathbf{v}$ be their projections. The statistical properties of inner product under projection perturbation can be shown as follows.

$$E[\mathbf{u}^t\mathbf{v} - \mathbf{x}^t\mathbf{y}] = 0$$

and

$$Var[\mathbf{u}^t\mathbf{v} - \mathbf{x}^t\mathbf{y}] = \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 + (\sum_i x_i y_i)^2)$$

Since $\mathbf{x}$ and $\mathbf{y}$ are not normalized by rows, but by columns in practice, with large dimensionality d and relatively small k, the variance is substantial. Similarly, the conclusion can be extended to the distance relationship. Therefore, projection perturbation does not strictly guarantee the preservation of distance/inner product as rotation or geometric perturbation does, which may significantly downgrade the model accuracy.

2.4 Sketch-based Approach

Sketch-based approach is primarily proposed to perturb high-dimensional sparse data [2], such as the datasets in text mining and market basket mining. A sketch

of the original record $\mathbf{x} = (x_1, \ldots, x_d)$ is defined by a r dimensional vector $\mathbf{s} = (s_1, \ldots, s_r)$, $r \ll d$, where

$$s_j = \sum_{i=1}^{d} x_i r_{ij}$$

The random variable r_{ij} is drawn from {-1,+1} with a mean of 0, and is generated from a pseudo-random number generator [5], which produces 4-wise independent values for the variable r_{ij}.

Note that the sketch based approach defers from projection perturbation with the following two features. First, the number of components for each sketch, i.e., r, can vary across different records, and is carefully controlled so as to provide a uniform measure of privacy guarantee across different records. Second, for each record, r_{ij} is different $-$ there is no fixed projection matrix across records.

The sketch based approach has a few statistical properties that enable approximate calculation of dot product of the original data records with their sketches. Let $\mathbf{s}$ and $\mathbf{t}$ with the same number of components r, be the sketches of the original records $\mathbf{x}$ and $\mathbf{y}$, respectively. The expected dot product $\mathbf{x}$ and $\mathbf{y}$ is given by the following.

$$E[\langle \mathbf{x}, \mathbf{y} \rangle] = \langle \mathbf{s}, \mathbf{t} \rangle / r$$

and the variance of the above estimation is determined by the few non-zeros entries in the sparse original vectors

$$Var(\langle \mathbf{s}, \mathbf{t} \rangle / r) = (\sum_{i=1}^{d} \sum_{l=1}^{d} x_i^2 y_l^2 - (\sum_{i=1}^{d} x_i y_i)^2)/r \tag{1.1}$$

On the other side, the original value x_k in the vector $\mathbf{x}$ can also be estimated by privacy attackers, the precision of which is determined by its variance $(\sum_{i=1}^{d} x_i^2 - x_k^2)/r, k = 1 \ldots d$. The larger the variance is, the better the original value is protected. Therefore, by decreasing r the level of privacy guarantee is possibly increased. However, the precision of dot-product estimation (Eq. 1.1) is decreased. This typical tradeoff has to be carefully controlled in practice [2].

2.5 Geometric Perturbation

Geometric perturbation is an enhancement to rotation perturbation by incorporating additional components such as random translation perturbation and noise addition to the basic form of multiplicative perturbation $Y = R \times X$. We show that by adding random translation perturbation and noise addition,

Geometric perturbation exhibits more robustness in countering attacks than simple rotation based perturbation [9]. Let $\mathbf{t}_{d\times 1}$ represent a random vector. We define a *translation matrix* as follows.

Definition 1. *Ψ is a translation matrix if $\Psi = [\mathbf{t}, \mathbf{t}, \ldots, \mathbf{t}]_{d\times n}$, i.e., $\Psi_{d\times n} = \mathbf{t}_{d\times 1}\mathbf{1}^T_{N\times 1}$.*

where $\mathbf{1}_{N\times 1}$ is the vector of N '1's. Let $\Delta_{d\times N}$ be a random noise matrix, where each element is Independently and Identically Distributed (iid) variable ε_{ij}, e.g., a Gaussian noise $N(0, \sigma^2)$.

The definition of geometric perturbation is given by a function $G(X)$,

$$G(X) = RX + \Psi + \Delta$$

Clearly, translation perturbation does not change distance, as for any pair of points $\mathbf{x}$ and $\mathbf{y}$, $\|(\mathbf{x} + \mathbf{t}) - (\mathbf{y} + \mathbf{t})\| = \|\mathbf{x} - \mathbf{y}\|$. Comparing with rotation perturbation, it protects the rotation center from attacks and adds additional difficulty to ICA-based attacks. However, translation perturbation does not preserve inner product.

In [9], it shows that by adding an appropriate level of noise Δ, one can effectively prevent knowledgeable attackers from distance-based data reconstruction, since noise addition perturbs distances, which protects perturbation from distance-inference attacks. For example, the experiments in [9] shows that a Gaussian noise $N(0, \sigma^2)$ is effective to counter the distance-inference attacks. Although noise addition prevents from fully preserving distance information, a low intensity noise will not change class boundary or cluster membership much.

In addition, the noise component is optional $-$ if the data owner makes sure that the original data records are secure and no people except the data owner knows any record in the original dataset, the noise component can be removed from geometric perturbation.

3. Transformation Invariant Data Mining Models

By using multiplicative perturbation algorithms, we can mine the the perturbed data directly with a set of existing "transformation-invariant data mining models", instead of developing new data mining algorithms to mine the perturbed data [4]. In this section, we will define the concept of transformation-invariant mining models with the example of "transformation-invariant classifiers", and then we extend our discussion to the transformation-invariant models in data classification and data clustering.

3.1 Definition of Transformation Invariant Models

Generally speaking, a transformation invariant model, if trained or mined on the transformed data, performs as good as the model based on the original data. We take the classification problem as an example. A classification problem is also a function approximation problem − classifiers are the functions learned from the training data [16]. In the following discussion, we use functions to represent classifiers. Let $\hat{f}_X$ represent a classifier $\hat{f}$ trained with dataset X and $\hat{f}_X(Y)$ be the classification result on the dataset Y. Let $T(X)$ be any transformation function, which transforms the dataset X to another dataset X_T. We use $Err(\hat{f}_X(Y))$ to denote the error rate of classifier $\hat{f}_X$ on testing data Y and let ε be some small real number, $|\varepsilon| < 1$.

Definition 2. *A classifier $\hat{f}$ is invariant to a transformation T if and only if $Err(\hat{f}_X(Y)) = Err(\hat{f}_{T(X)}(T(Y))) + \varepsilon$ for any training dataset X and testing dataset Y.*

With the strict condition $\hat{f}_X(Y) \equiv \hat{f}_{T(X)}(T(Y))$, we get the Proposition 2.

Proposition 2. *In particular, if $\hat{f}_X(Y) \equiv \hat{f}_{T(X)}(T(Y))$ is satisfied for any training dataset X and testing dataset Y, the classifier is invariant to the transformation $T(X)$.*

For instance, if a classifier $\hat{f}$ is invariant to rotation transformation, we call it *rotation-invariant classifier*. Similar definition applies to *translation-invariant classifier*.

In subsequent sections, we will list some examples of transformation invariant models for classification and clustering. Some detailed proofs can be found in [7].

3.2 Transformation-Invariant Classification Models

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 dataset and classifies the point to the class that most of its neighbors belong to. Since the distances between any points are not changed with rotation and translation transformation, the k nearest neighbors are not changed and thus the classification result is not changed either.

Since kNN classifier is a special case of kernel methods, we can also extend our conclusion to kernel methods. Here, we refer kernel methods to the traditional local methods [16]. In general, since the kernels are dependent on

the local points, the locality of which is evaluated by distance, transformations that preserve distance will make kernel methods invariant.

Support Vector Machines

Support Vector Machine (SVM) classifier also utilizes kernel functions in training and classification. However, it has an explicit training procedure, which differentiates itself from the traditional kernel methods we just discussed. We can use a two-step procedure to prove that a SVM classifier is invariant to a transformation. 1) Training with the transformed dataset generates the same set of model parameters; 2) the classification function with the model parameters is also invariant to the transformation. The detailed proof will involve the quadratic optimization procedure for SVM. We have demonstrated that SVM classifiers with typical kernels are invariant to rotation transformation [7]. It turns out that if a transformation makes the kernel invariant, then the SVM classifier is also invariant to the transformation.

There are the three popular choices for the kernels discussed in the SVM literature [10, 16].

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

Apparently, all of the three are invariant to rotation transformation. Since translation does not preserve inner product, it is not straightforward to prove that SVMs with polynomial and neural network kernels are invariant to translation perturbation. However, experiments [9] showed that these classifiers are also invariant to translation perturbation.

Linear Classifiers

Linear classification models are popular methods due to their simplicity. In linear classification models, the classification boundary is modeled as a hyperplane, which is clearly a geometric concept. It is easy to understand that distance-preserving transformations, such as rotation and translation, will still make the classes separated if they are originally separated. There is also a detailed proof showing that a typical linear classifier, perceptron, is invariant to rotation transformation [7].

3.3 Transformation-Invariant Clustering Models

Most clustering models are based on Euclidean distance such as the popular k-means algorithm [16]. Many are focused on the density property, which is derived from Euclidean distance, such as DBSCAN [11], DENCLUE [17] and OPTICS [6]. All of these clustering models are invariant to Euclidean-distance-preserving transformations, such as rotation and translation.

There are other clustering models, which employ different distance metrics [19], such as linkage based clustering and cosine-distance based clustering. As long as we can find a transformation preserving the particular distance metric, the corresponding clustering model will be invariant to this transformation.

4. Privacy Evaluation for Multiplicative Perturbation

The goal of data perturbation is twofold: preserving the accuracy of specific data mining models (data utility), and preserving the privacy of original data (data privacy). The discussion about transformation-invariant data mining models has shown that multiplicative perturbations can theoretically guarantee zero-loss of accuracy for a number of data mining models. The challenge is to find one that maximizes the privacy guarantee in terms of potential attacks.

We dedicate this section to discuss how good a multiplicative perturbation is in terms of preserving privacy under a set of privacy attacks. We first define a multi-column (or multidimensional) privacy measure for evaluating the privacy quality of a multiplicative perturbation over a given dataset. Then, we introduce a framework of privacy evaluation, which can incorporate different attack analysis into the evaluation of privacy guarantee. We show that using this framework, we can employ certain optimization methods (Section 5.5) to find a good perturbation among a bunch of randomly generated perturbations, which is locally optimal for the given dataset.

4.1 A Conceptual Multidimensional Privacy Evaluation Model

In practice, different columns (or dimensions, or attributes) may have different privacy concern. Therefore, we advocate that the general-purpose privacy metric Φ defined for an entire dataset should be based on **column privacy metric**, rather than point-based privacy metrics, such distance-based metrics. A conceptual privacy model is defined as $\Phi = \Phi(\mathbf{p}, \mathbf{w})$, where $\mathbf{p}$ denotes the column privacy metric vector $\mathbf{p} = [p_1, p_2, \ldots, p_d]$ of a given dataset X, and $\mathbf{w} = (w_1, w_2, \ldots, w_d)$ denote **privacy weights** associated to the d columns respectively. The column privacy p_i itself is defined by a function, which we will discuss later. In summary, the model suggests that the column-wise privacy metric should be calculated first and then use Φ to generate a composite metric. We will first describe some basic designs to the components in function Φ. Then, we dedicate another subsection to the concrete design of the function for generating $\mathbf{p}$.

The first design idea is to take the column importance into unification of different column privacy. Intuitively, the more important the column is, the higher level of privacy guarantee will be required for the perturbed data column. Since

$\mathbf{w}$ is used to denote the importance of columns in terms of preserving privacy, we use p_i/w_i to represent the *weighted column privacy* of column i.

The second concept is the *minimum privacy guarantee* and the *average privacy guarantee* among all columns. Normally, when we measure the privacy guarantee of a multidimensional perturbation, we need to pay more attention to the column that has the lowest weighted column privacy, because such a column could become the weakest link of privacy protection. Hence, the first composition function is the minimum privacy guarantee.

$$\Phi_1 = \min_{i=1}^{d}\{p_i/w_i\}$$

Similarly, the *average privacy guarantee* of the multi-column perturbation is defined by $\Phi_2 = \frac{1}{d}\sum_{i=1}^{d} p_i/w_i$, which could be another interesting measure. Note that these two functions assume that p_i should be comparable crossing columns, which is one of the important requirement in the following discussion.

4.2 Variance of Difference as Column Privacy Metric

After defining the conceptual privacy model, we move to the design of column-wise 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. Therefore, how statistically different the *estimated data* is from the original data could be an intuitive measure. We use a variance-of-difference (VoD) based approach, which has a similar form to the naive variance-based evaluation [4], but with very different semantics.

Let the difference between the original column data and the estimated data be a random variable $\mathbf{D}_i$. Without any knowledge about the original data, the mean and variance of the difference present the quality of the estimation. The perfect estimation will have zero mean and variance. Since the mean of difference, i.e., the bias of estimation, can be easily removed if the attacker knows the original distribution of column, we use only the variance of the difference (VoD) as the primary metric to determine the level of difficulty in estimating the original data.

VoD is formally defined as follows. Let $\mathbf{X}_i$ be a random variable representing the column i, $\mathbf{X}_i'$ be the *estimated result*[1] of $\mathbf{X}_i$, and $\mathbf{D}_i$ be $\mathbf{D}_i = \mathbf{X}_i' - \mathbf{X}_i$. Let $E[\mathbf{D}_i]$ and $Var(\mathbf{D}_i)$ denote the mean and the variance of $\mathbf{D}$ respectively.

[1] It would not be appropriate to use only the perturbed data for privacy estimation, if we consider the potential attacks.

Then VoD for column i is $Var(\mathbf{D}_i)$. Let an estimate of certain value, say x_i, be x_i', $\sigma = \sqrt{Var(\mathbf{D}_i)}$, and c denote confidence parameter depending on both the distribution of $\mathbf{D}_i$ and the corresponding confidence level. The corresponding original value x_i in $\mathbf{X}_i$ is located in the range defined below:

$$[x_i' - E[\mathbf{D}_i] - c\sigma, x_i' - E[\mathbf{D}_i] + c\sigma]$$

By removing the effect of $E[\mathbf{D}_i]$, the width of the estimation range, $2c\sigma$, presents the quality of estimating the original value, which proportionally reflects the level of privacy guarantee. The smaller range means better estimation, i.e., a lower level of privacy guarantee. For simplicity, we often use σ to represent the privacy level.

VoD only defines the privacy guarantee for a single column. However, we usually need to evaluate the privacy level of all perturbed columns together if a multiplicative perturbation is applied. The single-column VoD does not work across different columns since different column value ranges may result in very different VoDs. For example, the VoD of age may be much smaller than VoD of salary. Therefore, a same amount of VoD is not equally effective for columns with different value ranges. One straightforward method to unify the different value ranges is via *normalization* over the original dataset and the perturbed dataset. Normalization can be done with various ways, such as max/min normalization or standardized normalization [30]. After normalization, the level of privacy guarantee for each column should be approximately comparable. Note that normalization after VoD calculation, such as relative variance $VoD_i/Var(\mathbf{X}_i)$ is not appropriate, since small $Var(\mathbf{X}_i)$ will inappropriately increase the value.

4.3 Incorporating Attack Evaluation

Privacy evaluation has to consider the resilience to attacks as well. The VoD evaluation has a unique advantage in incorporating attack analysis in privacy evaluation. In general, let X be the normalized original dataset, P be the perturbed dataset, and O be the estimated/observed dataset through "attack simulation". We can calculate $VoD(\mathbf{X}_i, \mathbf{O}_i)$ for the column i in terms of different attacks. For example, the attacks to rotation perturbation can be evaluated by following steps. Details will be discussed shortly.

1 Naive Estimation: $O \equiv P$;

2 ICA-based Reconstruction: Independent Component Analysis (ICA) is used to estimate R. Let $\hat{R}$ be the estimate of R, and the estimated data $\hat{R}^{-1}P$ aligned with the known column statistics to get the dataset O;

3 Distance-based Inference: knowing a set of special points in X that can be mapped to certain set of points in P, so that the mapping helps to get the estimated rotation $\hat{R}$, and then $O = \hat{R}^{-1}P$.

4.4 Other Metrics

Other metrics include distance-based risk of privacy breach, which was used to evaluate the level of privacy breach when a few pairs of original data points and their maps in perturbed data are known [27]. Assume $\hat{\mathbf{x}}$ is the estimate of an original point $\mathbf{x}$. An *ϵ-privacy breach* occurs if

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \leq \epsilon \|\mathbf{x}\|$$

This roughly represents that, if the estimate is within an arbitrarily small local area around the original point, then the risk of privacy breach is high. However, even though the estimated point is distant from the original point, the estimation can still be effective − large distance may only be determined by the difference between a few columns, while other columns may be very similar. That is the reason why we should consider column-wise privacy metrics.

5. Attack Resilient Multiplicative Perturbations

Attack analysis is the essential component in privacy evaluation of multiplicative perturbation. The previous section has set up an evaluation model that can conveniently incorporate attack analysis through "attack simulation". Namely, privacy attacks to multiplicative perturbations are the methods for estimating original points (or values of particular columns) from the perturbed data, with certain level of additional knowledge about the original data. As the perturbed data goes public, the level of effectiveness is solely determined by the additional knowledge the attacker may have. In the following sections, we describe some potential inference attacks to multiplicative perturbations, primarily focused on rotation perturbation.

These attacks are organized according to the different levels of knowledge that an attacker may have. We hope that, from this section the interested readers will have more ideas about the attacks to general multiplicative perturbations and are able to apply appropriate tools to counter attacks. Most content of this section can be found in the paper [9], and we will just present the basic ideas here.

5.1 Naive Estimation to Rotation Perturbation

When the attacker knows no additional information, we call attacks under such circumstance as naive estimation, which simply estimates the original data from perturbed data. In this case, an appropriate rotation perturbation is enough to achieve high level of privacy guarantee. With the VoD metric over

the normalized data, we can formally analyze the privacy guarantee provided by the rotation perturbed data. Let X be the normalized dataset, X' be the rotation of X, and I_d be the d-dimensional identity matrix. VoD of column i can be evaluated by

$$\begin{aligned} Cov(\mathbf{X}' - \mathbf{X})_{(i,i)} &= Cov(R\mathbf{X} - \mathbf{X})_{(i,i)} \\ &= ((R - I_d)Cov(\mathbf{X})(R - I_d)^T)_{(i,i)} \end{aligned} \tag{1.2}$$

Let r_{ij} represent the element (i, j) in the matrix R, and c_{ij} be the element (i, j) in the covariance matrix of $\mathbf{X}$. The VoD for ith column is computed as follows.

$$Cov(\mathbf{X}' - \mathbf{X})_{(i,i)} = \sum_{j=1}^{d}\sum_{k=1}^{d} r_{ij}r_{ik}c_{kj} - 2\sum_{j=1}^{d} r_{ij}c_{ij} + c_{ii} \tag{1.3}$$

When the random rotation matrix is generated following the Haar distribution, a considerable number of matrix entries are approximately independent normal distribution $N(0, 1/d)$ [20]. For simplicity and easy understanding, we assume that all entries in random rotation matrix approximately follow independent normal distribution $N(0, 1/d)$. Therefore, random rotations will make VoD_i changing around the mean value c_{ii} as shown in the following equation.

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

It means that the original column variance could substantially influence the result of random rotation. However, the expectation of VoDs is not the only factor determining the final privacy guarantee. We should also look at the variance of VoDs. If the variance of VoDs is considerably large, we still get great chance to find a rotation with high VoDs in a set of sample random rotations, and the larger the $Var(VoD_i)$ is, the more likely the randomly generated rotation matrices can provide a high privacy level. With the approximately independency assumption, we have

$$\begin{aligned} Var(VoD_i) &\sim \sum_{i=1}^{d}\sum_{j=1}^{d} Var(r_{ij})Var(r_{ik})c_{ij}^2 \\ &\quad +4\sum_{j=1}^{d} Var(r_{ij})c_{ij}^2 \\ &\sim O(1/d^2\sum_{i=1}^{d}\sum_{j=1}^{d} c_{ij}^2 + 4/d\sum_{j=1}^{d} c_{ij}^2). \end{aligned}$$

The above result shows that $Var(VoD_i)$ seems approximately related to the average of the squared covariance entries, with more influence from the row i of covariance matrix. Therefore, by looking at the covariance matrix of the original dataset and estimate the $Var(VoD_i)$, we can estimate the chance of finding a random rotation that can give high privacy guarantee.

Rotation Center. The basic rotation perturbation uses the origin as the rotation center. Therefore, the points around the origin will be still close to the origin after the perturbation, which leads to weaker privacy protection over these points. The attack to rotation center can be regarded as another kind of naive estimation. This problem is addressed by random translation perturbation, which hides the rotation center. More sophisticated attacks to the combination of rotation and translation would have to utilize the ICA technique with sufficient additional knowledge, which will be described shortly.

5.2 ICA-Based Attacks

In this section, we introduce a high-level attack based on data reconstruction. The basic method for reconstructing X from the perturbed data RX would be Independent Component Analysis (ICA) technique, derived from the research of signal processing [18].

The ICA technique can be applied to estimate the independent components (the row vectors in our definition) of the original dataset X from the perturbed data, if the following conditions are satisfied:

1 The source row vectors are independent;

2 All source row vectors should be non-Gaussian with possible exception of one row;

3 The number of observed row vectors must be at least as large as the independent source row vectors.

4 The transformation matrix R must be of full column rank.

For rotation matrices, the 3rd and 4th conditions are always satisfied. However, the first two conditions although practical for signal processing, are often not satisfied in data classification or clustering. Furthermore, there are a few more difficulties in applying direct ICA-based attack. First of all, even ICA can be done successfully, the order of the original independent components cannot be preserved or determined through only ICA [18]. Formally, any permutation matrix P and its inverse P^{-1} can be substituted in the model to give $X' = RP^{-1}PX$. ICA could possibly give the estimate for some permutated source PX. Thus, we cannot identify the particular column without more knowledge about the original data. Second, 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 is often scaled up but we do not know how much the scaling is unless we know the original value range of the column. Therefore, without knowing the basic statistics of original columns, ICA-attack is not effective.

However, such basic column statistics are not impossible to get in some cases. Now, we assume that attackers know the basic statistics, including the column max/min values and the probability density function (PDF), or empirical PDF of each column. An enhanced ICA-based attack can be described as follows.

1 Run ICA algorithm to get a reconstructed dataset;

2 For each pair of ($\mathbf{O}_i$, $\mathbf{X}_j$), where $\mathbf{O}_i$ is a reconstructed column and $\mathbf{X}_i$ is an original column, scale $\mathbf{O}_i$ with the max/min values of $\mathbf{X}_j$;

3 Compare the PDFs of the scaled $\mathbf{O}_i$ and $\mathbf{X}_j$ to find the closest match among all possible combinations.

Note the the PDFs should be aligned before comparison. [9] gives one method to align it.

The above procedure describes how to use ICA and additional knowledge about the original dataset to precisely reconstruct the original dataset. Note if the four conditions for effective ICA are exactly satisfied and the basic statistics and PDFs are all known distinct from each other, the basic rotation perturbation will be totally broken by the enhanced ICA-based attack. In practice, we can test if the first two conditions for effective ICA are satisfied to decide whether we can safely use rotation perturbation, when the column distributional information is released. If ICA-based attacks can be effectively done, it is also trivial to reveal an additional translation perturbation, which is used to protect the rotation center.

If the first and second conditions are not satisfied, as for most datasets in data classification and clustering, precise ICA reconstruction cannot be achieved. Under this circumstance, different rotation perturbations may result in different levels of privacy guarantee and the goal is to find one perturbation that is resilient to the enhanced ICA-based attacks.

For projection perturbation [28], the third condition of effective ICA is not satisfied either. Although overcomplete ICA is available for this particular case [25], it is generally ineffective to break projection perturbation with ICA-based attacks. The major concern of projection perturbation is to find one that preserves the utility of perturbed data.

5.3 Distance-Inference Attacks

In the previous sections, we have discussed naive estimation and ICA-based attacks. In the following discussion, we assume that, besides the informa-

tion necessary to perform the discussed attacks, the attacker manages to get more knowledge about the original dataset. We assume two scenarios: 1) s/he also knows at least $d + 1$ linearly independent original data records, $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{d+1}\}$; or 2) s/he can only get less then d linearly independent points. S/he then tries to find the mapping between these points and their images in the perturbed dataset, denoted by $O = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{d+1}\}$, to break rotation perturbation and possible also translation perturbation.

For both scenarios, it is possible to find the images of the known points in the perturbed data. Particularly, if a few original points are highly distinguishable, such as "outliers", their images in the perturbed data can be correctly identified with high probability for low-dimensional small datasets (< 4 dimensions). With considerable cost, it is not impossible for higher dimensional and larger datasets by simple exhaustive search, although the probability to get the exact images is relatively low. For scenario 1), with the known mapping, the rotation R and translation $\mathbf{t}$ can be precisely calculated if the incomplete geometric perturbation $G(X) = RX + \Psi$ is applied. Therefore, the threat will be substantial to any other data point in the original dataset.

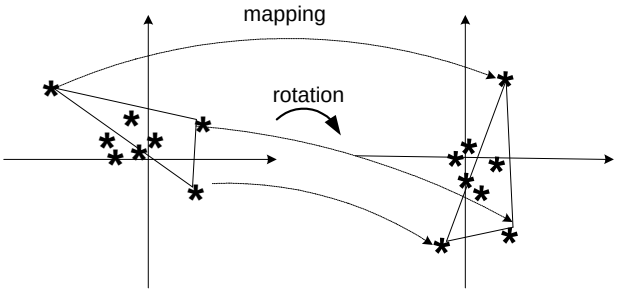

Figure 1.1. Using known points and distance relationship to infer the rotation matrix.

For scenario 2), if we assume the exact images of the known original points are identified, there is a comprehensive discussion about the potential privacy breach to rotation perturbation [27]. For rotation perturbation, i.e., $O = RX$ between the known points X and their images O, if X consists of less than d points, there are numerous estimates of R, denoted by $\hat{R}$, satisfying the relationship between X and O. The weakest points, except the known points X, are those around X. Paper [27] gives some estimation to the risk of privacy breach for certain point $\mathbf{x}$ if a set of points X and their image O are known. The definition is based on ϵ-privacy breach (Section 4.1). The probability of ϵ -privacy breach, $\rho(\mathbf{x}, \epsilon)$, for any $\mathbf{x}$ in the original dataset can be estimated as

follows. Let $d(\mathbf{x}, X)$ be the distance between $\mathbf{x}$ and X.

$$\rho(\mathbf{x}, \epsilon) = \frac{2}{\pi} \arcsin(\frac{\epsilon \|\mathbf{x}\|}{2d(\mathbf{x}, X)}), \text{if} \quad \epsilon\|\mathbf{x}\| < 2d(\mathbf{x}, X); \quad 1 \quad \text{otherwise.}$$

Note that ϵ-privacy breach is not sufficient to column-wise privacy evaluation. Thus, the above definition may not be sufficient as well.

In order to protect from distance-inference attack for both scenarios, an additional noise component Δ is introduced to form the complete version of geometric perturbation $G(X) = RX + \Psi + \Delta$, where $\Delta = [\delta_1, \delta_2, \ldots, \delta_N]$, and δ_i is a d-dimensional Gaussian random vector. The Δ component reduces the probability of getting exact images and the precision of estimation to R and Ψ, which significantly increases the resilience to distance-inference attacks.

Assume the attacker still knows enough pairs of independent (point, image). Now, with the additional noise component, the most effective way to estimate the rotation/translation component is linear regression. The steps include 1) filtering out the translation component first; 2) applying linear regression to estimate R; 3) plugging the estimate $\hat{R}$ back to estimate the translation component; 4) estimating the original data with $\hat{R}$ and $\hat{\Psi}$. There is a detailed procedure in [9]. We can simulate the procedure to estimate the resilience of a perturbation.

Note that the additional noise component also implies that we have to sacrifice some model accuracy for gaining the stronger privacy protection. An empirical study has been performed on a bunch of datasets to evaluate the relationship between noise intensity, resilience to attacks and model accuracy [9]. In general, a low-intense noise component will be enough to reduce the risk of being attacked, while still preserving model accuracy. However, the noise component is required only when the data owner is sure that a small part of the original data is released.

5.4 Attacks with More Prior Knowledge

There are also extreme cases that may not happen in practice, which assume the attacker knows a considerable amount of original data points and these points form a sample set that the higher-order statistical properties of the original dataset, like the covariance matrix, are approximately estimated from the sample set. By using the sample statistics and the sample points, the attacker can have more effective attacks.

Note that, in general, if the attacker has known so much information about the original data, its privacy may already be breached. It should not be advised to publish more original data. Further discussion about perturbations will make less sense. However, the techniques developed in these attacks, such as PCA-based attack [27] and AK-ICA attack [15] might be eventually utilized in other aspects to enhance multiplicative perturbations in the future. We will not give

detailed description about these attacks due to the space limitation. Instead, they will be covered by another dedicated chapter.

5.5 Finding Attack-Resilient Perturbations

We have discussed the unified privacy metric for evaluating the quality of a random geometric perturbation. Some known inference attacks have been analyzed under the framework of multi-column privacy evaluation, which allows us to design an algorithm to choose a good geometric perturbation in terms of these attacks − if the attacker knows considerable amount of original data, it is advised not to release the perturbed dataset, however. A deterministic algorithm in optimizing the perturbation may also provide extra clue to privacy attackers. Therefore, it is also expected to have certain level of randomization in the perturbation optimization.

A randomized perturbation-optimization algorithm for geometric perturbation was proposed in [9]. We briefly describe it as follows. Algorithm 1 is a hill-climbing method, which runs in a given number of iterations to find a geometric perturbation that maximizes the minimum privacy guarantee as possible. Initially, a random translation is selected, which needs not optimization at all. In each iteration, the algorithm randomly generates a rotation matrix. Local maximization of VoD [9] is applied to find a better rotation matrix in terms of naive estimation, which is then tested by the ICA reconstruction with the algorithm described in section 5.2. The rotation matrix is accepted as the currently best perturbation if it provides higher minimum privacy guarantee than the previous perturbations. After the iterations, if necessary, a 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 privacy guarantee (we use minimum privacy guarantee in the following algorithm) in terms of the known attacks. If the final privacy guarantee is lower than the expected threshold, the data owner can select not to release the data. This algorithm provides a framework, in which any discovered attacks can be simulated and evaluated.

6. Conclusion

We have reviewed the multiplicative perturbation method as an alternative method to privacy preserving data mining. The design of this category of perturbation algorithms is based on an important principle: by developing perturbation algorithms that can always preserve the mining task and model specific data utility, one can focus on finding a perturbation that can provide higher level of privacy guarantee. We described three representative multiplicative perturbation methods − rotation perturbation, projection perturbation, and ge-

Algorithm 1 Finding_a_resilient_perturbation ($X_{d\times N}$, **w**, m)

Input: $X_{d\times N}$:the original dataset, **w**: weights for attributes in privacy evaluation, m: the number of iterations.
Output: R_t: the selected rotation matrix, Ψ: the random translation, σ^2: the noise level, p: privacy quality
calculate the covariance matrix C of X;
$p = 0$, and randomly generate the translation Ψ;
for Each iteration **do**
  randomly generate a rotation matrix R;
  swapping the rows of R to get R', which maximizes $min_{1\leq i\leq d}\{\frac{1}{w_i}(Cov(R'X - X)_{(i,i)}\}$;
  $p_0 =$ the privacy guarantee of R', $p_1 = 0$;
  if $p_0 > p$ **then**
    generate $\hat{X}$ with ICA;
    $\{(1), (2), \ldots, (d)\} = argmin_{\{(1),(2),\ldots,(d)\}} \sum_{i=1}^{d} \Delta PDF(X_i, O_{(i)})$
    $p_1 = min_{1\leq k\leq d} \frac{1}{w_k} VoD(X_k, O_{(k)})$
  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 σ^2, so that $p_2 \geq p$ if $p < \phi$ or $p_2 > \phi$ if $p < \phi$.

ometric perturbation. All aim at preserving the distance relationship in the original data, thus achieving good data utility for a set of classification and clustering models. Another important advantage of using these multiplicative perturbation methods is the fact that we are not required to re-design the existing data mining algorithms in order to perform data mining over the perturbed data.

Privacy evaluation and attack analysis are the major challenging issues for multiplicative perturbations. We reviewed the multi-column variance of difference (VoD) based evaluation method and the distance-based method. Since column distribution information has high probability to be released publicly, in principle it is necessary to evaluate privacy guarantee based on columns. Although this chapter does not intend to enumerate all possible attacks, as we know, attack analysis to multiplicative perturbation is still a very active area, we describe several types of attacks and organize the discussion according to the level of knowledge that the attacker may have about the original data. We also outlined some techniques developed to date for addressing these attacks. Based on attack analysis and the VoD-based evaluation method, we show how to find the perturbations that locally optimize the level of privacy guarantee in terms of various attacks.

Acknowledgment

This work is partially supported by grants from NSF CISE CyberTrust program, IBM faculty award 2006, and an AFOSR grant.

References

[1] AGGARWAL, C. C., AND YU, P. S. A condensation approach to privacy preserving data mining. *Proc. of Intl. Conf. on Extending Database Technology (EDBT) 2992* (2004), 183–199.

[2] AGGARWAL, C. C., AND YU, P. S. On privacy-preservation of text and sparse binary data with sketches. *SIAM Data Mining Conference* (2007).

[3] AGRAWAL, D., AND AGGARWAL, C. C. On the design and quantification of privacy preserving data mining algorithms. *Proc. of ACM PODS Conference* (2002).

[4] AGRAWAL, R., AND SRIKANT, R. Privacy-preserving data mining. *Proc. of ACM SIGMOD Conference* (2000).

[5] ALON, N., MATIAS, Y., AND SZEGEDY, M. The space complexity of approximating the frequency moments. *Proc. of ACM PODS Conference* (1996).

[6] ANKERST, M., BREUNIG, M. M., KRIEGEL, H.-P., AND SANDER, J. OPTICS: Ordering points to identify the clustering structure. *Proc. of ACM SIGMOD Conference* (1999), 49–60.

[7] CHEN, K., AND LIU, L. A random geometric perturbation approach to privacy-preserving data classification. *Technical Report, College of Computing, Georgia Tech* (2005).

[8] CHEN, K., AND LIU, L. A random rotation perturbation approach to privacy preserving data classification. *Proc. of Intl. Conf. on Data Mining (ICDM)* (2005).

[9] CHEN, K., AND LIU, L. Towards attack-resilient geometric data perturbation. *SIAM Data Mining Conference* (2007).

[10] CRISTIANINI, N., AND SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[11] ESTER, M., KRIEGEL, H.-P., SANDER, J., AND XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Second International Conference on Knowledge Discovery and Data Mining* (1996), 226–231.

[12] EVFIMIEVSKI, A., GEHRKE, J., AND SRIKANT, R. Limiting privacy breaches in privacy preserving data mining. *Proc. of ACM PODS Conference* (2003).

[13] EVFIMIEVSKI, A., SRIKANT, R., AGRAWAL, R., AND GEHRKE, J. Privacy preserving mining of association rules. *Proc. of ACM SIGKDD Conference* (2002).

[14] FEIGENBAUM, J., ISHAI, Y., MALKIN, T., NISSIM, K., STRAUSS, M., AND WRIGHT, R. N. Secure multiparty computation of approximations. In *ICALP '01: Proceedings of the 28th International Colloquium on Automata, Languages and Programming,* (2001), Springer-Verlag, pp. 927–938.

[15] GUO, S., AND WU, X. 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, Sept 2007).

[16] HASTIE, T., TIBSHIRANI, R., AND FRIEDMANN, J. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[17] HINNEBURG, A., AND KEIM, D. A. An efficient approach to clustering in large multimedia databases with noise. *Proc. of ACM SIGKDD Conference* (1998), 58–65.

[18] HYVARINEN, A., KARHUNEN, J., AND OJA, E. *Independent Component Analysis*. Wiley-Interscience, 2001.

[19] JAIN, A. K., AND DUBES, R. C. Data clustering: A review. *ACM Computing Surveys 31* (1999), 264–323.

[20] JIANG, T. How many entries in a typical orthogonal matrix can be approximated by independent normals. *To appear in The Annals of Probability* (2005).

[21] JOHNSON, W. B., AND LINDENSTRAUSS, J. Extensions of lipshitz mapping into hilbert space. *Contemporary Mathematics 26* (1984).

[22] KARGUPTA, H., DATTA, S., WANG, Q., AND SIVAKUMAR, K. On the privacy preserving properties of random data perturbation techniques. *Proc. of Intl. Conf. on Data Mining (ICDM)* (2003).

[23] KIM, J. J., AND WINKLER, W. E. Multiplicative noise for masking continuous data. Tech. Rep. Statistics #2003-01, Statistical Research Division, U.S. Bureau of the Census, Washington D.C., April 2003.

[24] LEFEVRE, K., DEWITT, D. J., AND RAMAKRISHNAN, R. Mondrain multidimensional k-anonymity. *Proc. of IEEE Intl. Conf. on Data Eng. (ICDE)* (2006).

[25] LEWICKI, M. S., AND SEJNOWSKI, T. J. Learning overcomplet representations. *Neural Computation 12*, 2 (2000).

[26] LINDELL, Y., AND PINKAS, B. Privacy preserving data mining. *Journal of Cryptology 15*, 3 (2000), 177–206.

[27] LIU, K., GIANNELLA, C., AND KARGUPTA, H. An attacker's view of distance preserving maps for privacy preserving data mining. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'06)* (Berlin, Germany, September 2006).

[28] LIU, K., KARGUPTA, H., AND RYAN, J. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering (TKDE) 18*, 1 (January 2006), 92–106.

[29] MACHANAVAJJHALA, A., GEHRKE, J., KIFER, D., AND VENKITASUBRAMANIAM, M. l-diversity: Privacy beyond k-anonymity. *Proc. of IEEE Intl. Conf. on Data Eng. (ICDE)* (2006).

[30] NETER, J., KUTNER, M. H., NACHTSHEIM, C. J., AND WASSERMAN, W. *Applied Linear Statistical Methods*. WCB/McGraw-Hill, 1996.

[31] OLIVEIRA, S. R. M., AND ZAÏANE, O. R. Privacy preservation when sharing data for clustering. In *Proceedings of the International Workshop on Secure Data Management in a Connected World* (Toronto, Canada, August 2004), pp. 67–82.

[32] SADUN, L. *Applied Linear Algebra: the Decoupling Principle*. Prentice Hall, 2001.

[33] STEWART, G. The efficient generation of random orthogonal matrices with an application to condition estimation. *SIAM Journal on Numerical Analysis 17* (1980).

[34] SWEENEY, L. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10*, 5 (2002).

[35] VAIDYA, J., AND CLIFTON, C. Privacy preserving k-means clustering over vertically partitioned data. *Proc. of ACM SIGKDD Conference* (2003).