An Attacker's View of Distance Preserving Maps For Privacy Preserving Data Mining

Kun Liu, Chris Giannella, and Hillol Kargupta*

Department of Computer Science and Electrical Engineering,
University of Maryland Baltimore County,
1000 Hilltop Circle, Baltimore, MD 21250, USA,
{kunliu1,cgiannel,hillol}@cs.umbc.edu

Abstract. We examine the effectiveness of distance preserving transformations in privacy preserving data mining. These techniques are potentially very useful in that some important data mining algorithms can be *efficiently* applied to the transformed data and produce *exactly the same* results as if applied to the original data *e.g.* distance-based clustering, k-nearest neighbor classification. However, the issue of how well the original data is hidden has, to our knowledge, not been carefully studied. We take a step in this direction by assuming the role of an attacker armed with two types of prior information regarding the original data. We examine how well the attacker can recover the original data from the transformed data and prior information. Our results offer insight into the vulnerabilities of distance preserving transformations.

1 Introduction

Recent interest in the collection and monitoring of data using data mining technology for the purpose of security and business-related applications has raised serious concerns about privacy issues. For example, mining health-care data for detection of bio-terrorism may require analyzing clinical records and pharmacy transaction data of certain off-the-shelf drugs. However, combining such diverse data sets belonging to different parties may violate privacy laws. Privacy Preserving Data Mining (PPDM) strives to provide a solution to this dilemma. It aims to allow useful data patterns to be extracted without compromising privacy.

Data perturbation represents one common approach in PPDM. Here, the original dataset is perturbed and the result is released for data analysis. Perturbation approaches typically face a "privacy/accuracy" trade-off. On the one hand, perturbation must not allow the original data records to be adequately recovered. On the other, it must allow "patterns" in the original data to be recovered. In many cases, increased privacy comes at the cost of reduced accuracy and vice versa. For example, Agrawal and Srikant [1] proposed adding randomly generated i.i.d. noise to the dataset. They showed how the distribution from which the original data arose can be estimated using only the perturbed data.

* Also affiliated with AGNIK, LLC, USA.

However, Kargupta *et al.* [2] and Huang *et al.* [3] pointed out how, in many cases, the noise can be filtered off leaving a reasonably good estimation of the original data. These results point to the fact that unless the variance of the additive noise is sufficiently large, original data records can be recovered unacceptably well. However, this increase in variance reduces the accuracy with which the original data distribution can be estimated. This privacy/accuracy trade-off is not limited to additive noise, some other perturbation techniques suffer from a similar problem *e.g.* k-anonymity [4].

Recently, distance preserving data perturbation [5, 6] has gained attention since it mitigates the privacy/accuracy trade-off by guaranteeing perfect accuracy. Many important data mining algorithms can be *efficiently* applied to the transformed data and produce *exactly the same* results as if applied to the original data. *e.g.* distance-based clustering and k-nearest neighbor classification. However, the issue of how well the original data is hidden has, to our knowledge, not been carefully studied. In this paper, we address this issue by studying how well an attacker can recover the original data from the transformed data and prior information. We restrict our attention to the class of distance preserving transformations that fix the origin and consider recovery of the original data in the presence of two different classes of prior information (described later). Our analysis explicitly illuminates scenarios where privacy can be breached. As such, valuable information is gained into the effectiveness of distance preserving transformation for privacy preserving data mining.

The remainder of this paper is organized as follows. Section 2 discusses some basic mathematical properties of distance preserving transformations, the application of these transformations to privacy-preserving data mining, and two classes of attacker prior knowledge. Sections 3 and 4 examine in detail how knowledge in each of these classes can be used to estimate the original data from the transformed data. Section 5 discusses related work. Finally, section 6 concludes the paper with a brief discussion of a suggested remedy for the attacker's approach in one of the classes of prior knowledge.

2 Distance Preserving Transformations

Throughout this paper (unless otherwise stated), all matrices and vectors discussed are assumed to have real entries. All vectors are assumed to be column vectors and M' denotes the transpose of any matrix M. An $m \times n$ matrix M is said to be orthogonal if $M'M = I_n$, the $n \times n$ identity matrix.[1] Let $\mathbb{O}_n$ denote the set of all $n \times n$, orthogonal matrices. A function $T : \mathbb{R}^n \to \mathbb{R}^n$ is distance preserving if for all $x, y \in \mathbb{R}^n$, $||x - y|| = ||T(x) - T(y)||$, where $||.||$ denotes l^2-norm of a vector. Here T is also called a *rigid motion*. It has been shown that any distance preserving transformation is equivalent to an orthogonal transformation followed by a translation [7, pg. 128]. In other words, there exists $M_T \in \mathbb{O}_n$ and $v_T \in \mathbb{R}^n$ such that T equals $x \in \mathbb{R}^n \mapsto M_T x + v_T$. If T fixes the origin,

[1] If M is square, it is orthogonal if and only if $M' = M^{-1}$ [7, pg. 17].

$T(0) = 0$, then $v_T = 0$, hence, T is an orthogonal transformation. Henceforth we assume T is a distance preserving transformation which fixes the origin – an orthogonal transformation. Next we describe the privacy application scenarios where orthogonal transformation can be used to hide the data while allowing important patterns to be discovered *without error*.

2.1 Privacy Application Scenarios

We consider two privacy application scenarios as follows.
Census scenario: An organization has a private dataset X (each column is a data record) and wishes to make it publicly available for data analysis while keeping the original data records private. To accomplish this, $Y = M_T X$ is released to the public. The distance preserving nature of T allows a public entity to easily recovery many useful patterns from Y. For example, the cluster membership produced by a Euclidean distance-based K-means clustering on Y will be exactly the same as that produced on X. This model is widely studied in the field of security control for statistical databases. We refer the reader to [8] for a nice overview on this topic.
Storage outsourcing scenario: An organization continuously generates private data records, but does not wish to invest in the infrastructure (both personnel and hardware) needed to manage the storage. Outsourcing this job can be an attractive alternative *i.e.* the data records are handed over to an outside agency who manages their storage. However, the original data records are sensitive and the organization would rather avoid releasing them in the plain to the outsourcing agency. To accomplish this, the owner applies T to each data record and releases the results to the outsourcing agency. Whenever the owner wishes to retrieve records from the outsourced database, she transforms her query by the same T and sends it to the outsourcing agency who carries out similarity comparison on the data and, in turn, sends the results back to the owner. This scenario is closely related to work on secure database outsourcing, *e.g.* [17].

2.2 Prior Knowledge

Let the $n \times m$ matrix X denote a private dataset, with each column of X being a record and each row an attribute. We assume that the attacker knows that T is an orthogonal transformation and knows the perturbed data $Y = M_T X$. In most realistic scenarios, the attacker has some additional *prior knowledge* which can potentially be used effectively for breaching privacy. We consider two types of prior knowledge.
Known input-output: The attacker knows some collection of linearly independent private data records. In other words, the attacker has a set of linearly independent input-output pairs.
Known sample: The attacker knows that the original dataset arose as independent samples of some n-dimensional random vector V with unknown p.d.f. Also the attacker has another collection of independent samples from V. For

technical reasons, we make a mild additional assumption: the covariance matrix of V has distinct eigenvalues.

In the next two sections, we describe and analyze an attack technique for *each type of* prior knowledge listed above.

3 Known Input-Output Attack

Let X_k denote the first k columns of X and X_{m-k} the remainder (likewise for Y). We assume that columns of X_k are all linearly independent and X_k is known to the attacker (Y is, of course, also known). The goal of the attacker is to recover some columns in X_{m-k} with at most $\epsilon \geq 0$ error (described later). If $k = n$, then the attacker can recover X_{m-k} perfectly as it equals $(Y_k X_k^{-1})' Y_{m-k}$. Thus, we assume $k < n$. Based on known information, the attacker can narrow down the space of possibilities for M_T to $\mathbb{M}(X_k, Y_k) = \{M \in \mathbb{O}_n : MX_k = Y_k\}$. Since the attacker has no additional information, any of these matrices is equally likely to have been M_T. The attacker chooses $\hat{M}$ uniformly from $\mathbb{M}(X_k, Y_k)$ and chooses index $1 \leq \hat{i} \leq m-k$ using some criterion (described later), then produces $\hat{x} = \hat{M}' y_{\hat{i}} = \hat{M}' M_T x_{\hat{i}}$ as an estimate of $x_{\hat{i}}$, where $x_{\hat{i}}$ is the $\hat{i}^{th}$ column of X_{m-k}. We say that an *ϵ-privacy breach* occurs if $||\hat{x} - x_{\hat{i}}|| \leq ||x_{\hat{i}}||\epsilon$. We define $\rho(x_{\hat{i}}, \epsilon)$ as the probability that an *ϵ-privacy breach* occurs. This serves as the criterion for choosing $\hat{i}$.

Next, for any vector $x \in \mathbb{R}^n$, we develop a closed form expression for $\rho(x, \epsilon)$, the probability that $||\hat{M}' M_T x - x|| \leq ||x||\epsilon$. This is the *$\epsilon$-privacy breach* probability for x. Due to space limitations, all proofs are omitted.

3.1 Probability of Privacy Breach

Let $Col(X_k)$ denote the column space of X_k and $Col_\perp(X_k)$ denote its orthogonal complement, *i.e.* $\{z \in \mathbb{R}^n : z'w = 0, \forall w \in Col(X_k)\}$. Since the columns of X_k are linearly independent, then there exists orthogonal matrices U_k $(n \times k)$ and U_{n-k} $(n \times (n-k))$ such that $Col(X_k) = Col(U_k)$ and $Col_\perp(X_k) = Col(U_{n-k})$. It can be proved that

$$\mathbb{M}(X_k, Y_k) = \{M_T U_k U_k' + M_T U_{n-k} P U_{n-k}' : P \in \mathbb{O}_{n-k}\}.$$

Hence, linear map $L : M \in \mathbb{M}(X_k, Y_k) \mapsto (M_T U_{n-k})' M U_{n-k} \in \mathbb{O}_{n-k}$ is a bijection. It can be further shown that

$$||\hat{M}' M_T x - x|| = ||L(\hat{M})' U_{n-k}' x - U_{n-k}' x||.$$

Thus, $\rho(x, \epsilon)$ equals the probability that a matrix $\hat{P}$ drawn uniformly from $\mathbb{O}_{n-k}$ satisfies

$$||\hat{P}' U_{n-k}' x - U_{n-k}' x|| \leq ||x||\epsilon. \tag{1}$$

Now let $S_{n-k}(U_{n-k}' x)$ be the hypersphere in $\mathbb{R}^{n-k}$ centered at the origin with radius $||U_{n-k}' x||$. Vector $\hat{P}' U_{n-k}' x$ and $U_{n-k}' x$ from inequality (1) are points on

the surface of $S_{n-k}(U'_{n-k}x)$. Let $S_{n-k}(U'_{n-k}x, ||x||\epsilon)$ be the portion of S_{n-k} whose distance from $U'_{n-k}x$ is no larger than $||x||\epsilon$, *i.e.* $S_{n-k}(U'_{n-k}x, ||x\epsilon||) = \{z \in S_{n-k}(U'_{n-k}x) : ||z - U'_{n-k}x|| \leq ||x||\epsilon\}$. From inequality (1), it follows that $\rho(x, \epsilon)$ is the probability that a randomly chosen $\hat{P} \in \mathbb{O}_{n-k}$ satisfies $\hat{P}'U'_{n-k}x \in S_{n-k}(U'_{n-k}x, ||x||\epsilon)$. Therefore, this probability equals the ratio of the surface area of $S_{n-k}(U'_{n-k}x, ||x||\epsilon)$ to the surface area of $S_{n-k}(U'_{n-k}x)$. Then, it can be shown:

$$\rho(x, \epsilon) = (\frac{1}{\pi})2arcsin(\frac{||x||\epsilon}{2||U'_{n-k}x||}) \text{ if } ||x||\epsilon < 2||U'_{n-k}x||; \text{ 1 otherwise.}$$

An alternate characterization of $||U'_{n-k}x||$ yields a more intuitive form of the second right-hand side. Consider $U_kU'_kx$ the projection of x into $Col(X_k)$. The distance, $d(x, X_k)$, of x from $Col(X_k)$ is $||x - U_kU'_kx||$. It can be shown that $||U'_{n-k}x|| = d(x, X_k)$. Therefore,

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

This formula allows us to observe the behavior of the *ϵ-privacy breach* probability for x in terms of $||x||\epsilon$ and the distance of x from $Col(X_k)$. Indeed the probability is approximately inversely proportional to $d(x, X_k)$ for $d(x, X_k) >> ||x||\epsilon$. [2] On the other hand, as $||x||\epsilon \to 2d(x, X_k)$, the breach probability goes to one. In the extreme case where $x \in Col(X_k)$, a breach occurs with probability 1 for any ϵ.

3.2 Attack Technique

Using equation (2), $\rho(x_{\hat{i}}, \epsilon)$ can be computed from $||x_{\hat{i}}||$, ϵ, and $d(x_{\hat{i}}, X_k)$. Since the attacker knows Y, she can compute $||y_{\hat{i}}|| = ||M_Tx_{\hat{i}}|| = ||x_{\hat{i}}||$ and V_k an $n \times k$, orthogonal matrix such that $Col(V_k) = Col(Y_k)$. It can be shown that $d(x_{\hat{i}}, X_k) = d(y_{\hat{i}}, Y_k) = ||M_Tx_{\hat{i}} - VV'M_Tx_{\hat{i}}||$. Therefore, the attacker chooses $\hat{i}$ to maximize $\rho(x_{\hat{i}}, \epsilon)$. If the data owner knows that X_k is in the attacker's prior knowledge, then the owner can protect against this attack by simply not releasing M_Tx_i for any x_i where $d(x_i, X_k)$ is unacceptably small. On the other hand, if the owner does not know X_k is prior knowledge, then this attack technique can be quite damaging.

4 Known Sample Attack

In this scenario, we assume that each data record arose as an independent sample from a random vector V with unknown p.d.f. We also make the following

[2] For small z, $arcsin(z)$ is approximately linear.

mild technical assumption: the population covariance matrix Σ_V of V has all distinct eigenvalues. We make this assumption because it holds in most practical situations [9, pg. 27]. Furthermore, we assume that the attacker has a collection of p samples that arose independently from V – these are denoted as the columns of matrix S. In this section we design a Principal Component Analysis (PCA)-based attack technique by which the attacker produces $\hat{X}$, an estimate of X, from $Y = M_T X$ and S. Unlike Section 3, we do not attempt a rigorous analysis of the attacker's success probability. Instead, we analyze the recovery error through experiments.

4.1 PCA Preliminaries

Let Σ_V denote the population covariance matrix of V. Since Σ_V is an $n \times n$, symmetric matrix (and we assume it has all distinct eigenvalues), it has n real eigenvalues $\lambda_1 > \ldots > \lambda_n$ [10, pg. 295]. Associated with each eigenvalue λ_i is its eigenspace, $\{z \in \mathbb{R}^n : \Sigma_V z = z\lambda_i\}$. It can be shown that since Σ_V has distinct eigenvalues, the eigenspaces are pair-wise orthogonal and each has dimension one [10, pg. 295]. As is standard practice, we restrict our attention to only a small number of eigenvectors. Let $\mathcal{Z}(V)_i$ denote the set of all eigenvectors $z \in \mathbb{R}^n$ such that $\Sigma_V z = z\lambda_i$ and $||z|| = 1$. Now consider random vector $T(V) = M_T V$ and let $\Sigma_{M_T V}$ denote its covariance matrix. The eigenspaces of Σ_V are related in a natural way to those of $\Sigma_{M_T V}$, as shown by the following theorem (all proofs are omitted due to space constraints).

Theorem 1 *The eigenvalues of Σ_V and $\Sigma_{M_T V}$ are the same and $M_T \mathcal{Z}(V)_i = \mathcal{Z}(M_T V)_i$ where $M_T \mathcal{Z}(V)_i$ equals $\{M_T w : w \in \mathcal{Z}(V)_i\}$.*

Since all the eigenspaces of Σ_V have dimension one, it can be shown that $\mathcal{Z}(V)_i$ contains only two eigenvectors $z_i, -z_i$, *i.e.* $\mathcal{Z}(V)_i = \{z_i, -z_i\}$. Let z_i be the lexicographically larger vector among $z_i, -z_i$, and let Z be the $n \times n$ matrix whose i^{th} column is z_i. Since the eigenspaces of Σ_V are pairwise orthogonal and $||z_i|| = 1$, Z is orthogonal. Similarly, we have that $\mathcal{Z}(M_T V)_i = \{w_i, -w_i\}$ (w_i is the lexicographically larger among $w_i, -w_i$) and W is the matrix with i^{th} column w_i (W is orthogonal). The following result forms the basis of the attacker's attack algorithm.

Corollary 1 *Let $\mathbb{I}_n$ be the space of all $n \times n$, matrices with each diagonal entry ± 1 and each off-diagonal entry 0 (2^n matrices in total). There exists $D_0 \in \mathbb{I}_n$ such that $M_T = W D_0 Z'$.*

4.2 PCA Attack Algorithm

First assume the attacker knows the population covariance Σ_V and $\Sigma_{M_T V}$. Thus, the attacker can compute W and Z'. By Corollary 1, the attacker knows that M_T equals $W D_0 Z'$ for some $D_0 \in \mathbb{I}_n$, and therefore, the original data would be recovered by $M_T' Y = Z D_0 W' Y$. The problem is how to choose the right D_0

from all the possible 2^n elements in $\mathbb{I}_n$. To do so, the attacker must utilize S and Y, in particular, the fact that these arose as independent samples from V and $M_T V$, respectively. For each $D \in \mathbb{I}_n$, each column of $WDZ'S$ arose as an independent sample from $WDZ'V$. If $D = D_0$, then $WDZ' = M_T$, so, $WDZ'S$ and Y should come from the same p.d.f. The attacker will choose $D \in \mathbb{I}_n$ such that $WDZ'S$ is most likely to have arisen from the same p.d.f. as Y. To make this choice, a similarity function $G(WDZ'S, Y)$ is introduced, and the D that maximizes G is chosen. There might be many ways to define this function. In this paper, we use a multivariate two-sample hypothesis test for equal distributions [11]. The two-sample problem assumes that there are two sets of independent samples $x_1, x_2, \ldots, x_{m1}$ and $y_1, y_2, \ldots, y_{m2}$ of independent random vectors with distributions F_1 and F_2, respectively. The goal of two-sample problem is to test $H_0 : F_1 = F_2$, versus the composite alternative $H_1 : F_1 \neq F_2$. For each $D \in \mathbb{I}_n$, we compute the p-value of the test on $WDZ'S$ and Y, denoted by $\rho(D)$. Here the p-value is defined as the smallest level of significance at which H_0 would be rejected on a given data set. Small p-values suggest that the null hypothesis is unlikely to be true. The smaller it is, the more convincing is the rejection of the null hypothesis. Therefore the value of function G is nothing but the p-value, and the D matrix that is associated with the highest p-value is chosen.

In practice, the population covariance Σ_V and $\Sigma_{M_T V}$ are unknown, and will be replaced by the sample covariance Σ_S and Σ_Y from S and Y (independent samples arising from V and $M_T V$). Algorithm 4.2.1 shows the complete PCA-based attack procedure.

Algorithm 4.2.1 PCA-based Attack Technique

Inputs: S, an $n \times p$ matrix where each column arose as an independent sample from V (a random vector with unknown p.d.f). $Y = M_T X$ where M_T is an unknown, $n \times n$, orthogonal matrix; and X is an $n \times m$ unknown matrix where each column arose as an independent sample from V.

Outputs: $\hat{X}$, an estimation of X.

Assumptions: Σ_V has all distinct eigenvalues.

1: Compute sample covariance matrix $\hat{\Sigma}_S$ from S and sample covariance matrix $\hat{\Sigma}_Y$ from Y. $[O(n^2 m + n^2 p)]$

2: Compute the eigenvector matrix $\hat{Z}$ of $\hat{\Sigma}_S$ and $\hat{W}$ of $\hat{\Sigma}_Y$. Each eigenvector has unit length and is sorted in the matrix by the corresponding eigenvalue. $[O(n^3)]$

3: Choose $D_0 = argmax\{G(\hat{W}D\hat{Z}'S, Y) : D \in \mathbb{I}_n\}$. $[O(2^n B)]$

4: Compute $\hat{X} = \hat{Z} D_0 \hat{W}' Y$. $[O(n^3 + n^2 m)]$

The computation cost of Algorithm 4.2.1 is $O(n^2(m+p)+n^3+2^n B)$ assuming $G(.,.)$ requires $O(B)$ computation. For the two-sample test, $B = (m+p)^2$, so, the total computation of the algorithm is $O(2^n(m+p)^2)$.

4.3 Effectiveness

The effectiveness of the PCA Attack algorithm depends on two correlated aspects: 1) the p.d.f., f, of V; and 2) the quality of covariance estimation.

PDF of V: First, suppose for some $D_1 \neq D_0 \in \mathbb{I}_n$, f is *invariant over* D_1 in the sense that $f_{D_1} = f_{D_0}$ where f_{D_i} is the p.d.f. $x \in \mathbb{R}^n \mapsto f(WD_iZ'x)$. Then, $WD_0Z'S$, $WD_1Z'S$ and Y all arose from the same p.d.f., so $\rho(D_0)$ may not be larger than $\rho(D_1)$, and the attack algorithm will fail. An example of such an f is the n-variate Gaussian with mean vector zero and covariance matrix I_n. This distribution is invariant to orthogonal transformation. Second, suppose the eigenvalues of Σ_V are nearly identical. For example, suppose f has a diagonal covariance matrix whose diagonal entries (from top-left to bottom-right) are d, $d-\epsilon$, $d-2\epsilon$, ..., $d-n\epsilon$ where $d-n\epsilon > 0$ and $0 < \epsilon < 1$. Small errors in estimating Σ_V from S can produce a different ordering of the eigenvectors, hence, large errors in the attacker's recovery.

Quality of Covariance Estimation: A great deal of work has been conducted in the statistics community on estimating the covariance matrix of a random vector based on an independent sample [9, Chapter 10.4]. Any estimation technique can be used in our technique. In experiments we use the simple, standard sample covariance estimator.

4.4 Experiments

To validate the PCA-based attack algorithm, we conducted experiments on both synthetic and real world data. One such synthetic dataset contains 1000 data points, which are generated from a two-dimensional Gaussian distribution with mean $(-10, 10)$ and covariance $\begin{pmatrix} 1 & 1.5 \\ 1.5 & 3 \end{pmatrix}$. The attacker has 50 sample data points (5% of the size of original data) chosen from the same distribution. Figure 1 shows the results of perturbation and recovery. It can be seen that although the perturbed data is very different from the original one, the recovered data almost overlaps with the original data.[3] To further examine how sample size affects the quality of the attack, we fixed the orthogonal perturbation matrix, and variated the number of samples from 1% of the original data to 20%. For each sample ratio, 20 independent trials were conducted. We computed 95% confidence interval of the results. Figure 2 shows that as the sample size increases, the average relative distance between the columns of X and $\hat{X}$ decreases.[4]

For real world data, we chose the Adult Database from the UCI machine learning repository. This data set contains 32,561 records, and it is extracted from the census bureau database. For the purpose of visualization, we only selected three continuous attributes: age, education-num and hours-per-week, for the experiment. We first randomly separated the dataset into two disjoint sets.

[3] Note that the shape of the perturbed data does not appear very similar to the shape of the original data because the axes scales are not even.

[4] The average relative distance between the columns is defined as $\frac{\sum_i^{numCols} \frac{||X_i - \hat{X}_i||}{||X_i||}}{numCols}$.

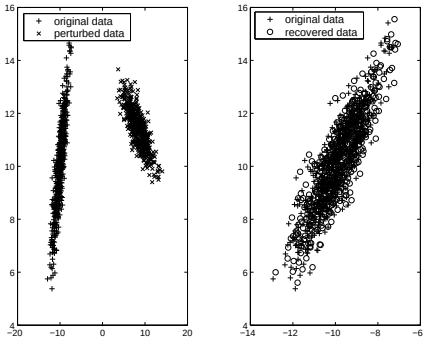

Fig. 1. Performance of PCA-based attack for two-dimensional Gaussian data.

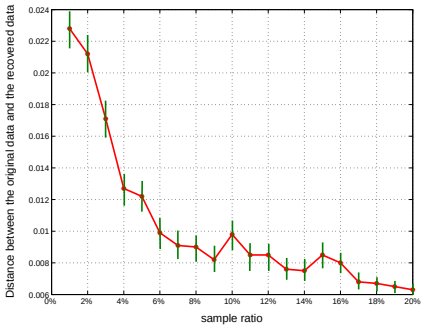

Fig. 2. Performance (average of 20 independent trials) w.r.t. sample size. Error bars show 95% confidence intervals.

One set is viewed as the original data, and the other one is the attacker's sample data, which accounts for 5% of the original data. The left column of Figure 3 shows the difference between the original data and the perturbed data; the right column of Figure 3 depicts the results of PCA-based attack. It can be seen that the recovered data approximates the original data very well. To examine the influence of sample size, we fixed the orthogonal perturbation matrix, and variated the number of samples from 2% of the original data to 20%. For each sample ratio, 20 independent trials were conducted. Figure 4 gives the result.

To evaluate the complexity of the PCA attack algorithm, we generated multivariate Gaussian data with dimensionality ranging from 2 to 12. Each data set contains 5250 records, 250 records of which are used as samples. The energy test proposed in [11] was used to quantify similarity ($G(.,.)$), The experiment was conducted in Matlab on a dual-processor workstation with 3.00GHz and 2.99GHz Xeon CPUs and 3.00GB RAM. We observed that for 2-dimensional data, it took 143.1090 seconds, and for 12-dimensional data, it took 1.2442×10^5 seconds. Although the running time goes up rapidly as the dimension increases, this algorithm is still computationally feasible for relatively high dimensional data.

5 Related Work

This section presents a brief overview of the literature on data perturbation for PPDM. There is another class of PPDM technique using secure multi-party computation (SMC) protocols for implementing common data mining algorithms across distributed datasets. We refer interested readers to [12] for more details.

Additive perturbation: Agrawal and Srikant [1] proposed the addition of i.i.d., white noise for privacy protection. They describe a technique by which the original data distribution can be estimated from the perturbed data. Kargupta

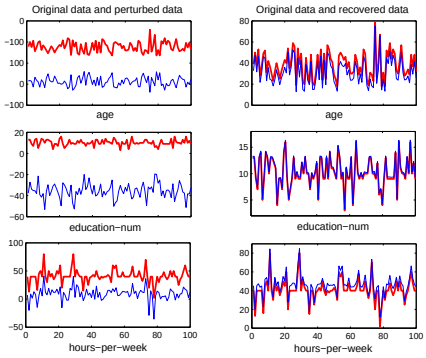

Fig. 3. (Left Column) The first 100 records from the original data and the perturbed data. (5% samples) (Right Column) The first 100 records from the original data and the recovered data. (5% samples)

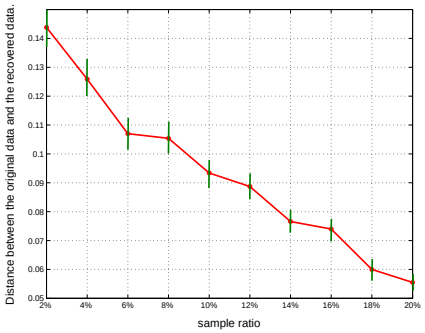

Fig. 4. Performance (average of 20 independent trials) of PCA-based attack w.r.t. sample size for Adult data. Error bars show 95% confidence intervals.

et al. [2] questioned the use of additive, white noise by showing how, in some cases, the noise can be effectively filtered off revealing a good approximation of the original data. This technique was further investigated by Huang *et al.* [3]. To our knowledge, these techniques are not applicable to this paper since it is concerned with non-additive perturbation.

Multiplicative perturbation: Two basic forms of multiplicative noise have been studied in the Statistics community [13]. One multiplies each data element by a random number that has a truncated Gaussian distribution with mean one and small variance. The other takes a logarithmic transformation of the data first, adds multivariate Gaussian noise, then takes the exponential function *exp(.)* of the noise-added data. Neither of these perturbations preserve distance and are fundamentally different than the type we study, orthogonal transformations. To facilitate large scale data mining applications, Liu *et al.* [14] proposed an approach where the data is multiplied by a randomly generated matrix – in effect, the data is projected into a lower dimensional space. This technique preserves distance on expectation. However, the privacy analysis there did not take into account prior knowledge as we do. Oliveira and Zaiane [6], Chen and Liu [5] discuss the use of random rotation for privacy-preserving clustering and classification. These authors observe that the distance preserving nature of random rotation makes it useful in this setting, but do not analyze its privacy limitations.

Categorical data perturbation: Evfimievski *et al.* [15], Rizvi and Haritza [16] consider the use of data categorical perturbation. They develop algorithms from which association rules present in the original data can be estimated from the perturbed data. Along a related line, Verykios [18] consider perturbation techniques which allow the discovery of *some* association rules while hiding others considered to be sensitive.

Data anonymization: Sweeney [4] developed the *k-anonymity* framework wherein the original data is transformed so that the information for any individual cannot be distinguished from k-1 others. Values from the original data are generalized (replaced by a less specific value) to produce the anonymized data. This technique makes no accuracy guarantees for subsequent analysis of the transformed data.

Data swapping: This technique transforms the database by switching a subset of attributes between selected pairs of records so that the individual record entries are unmatched, but the statistics are maintained across the individual fields. A variety of refinements and applications of data swapping have been addressed since its initial appearance. We refer readers to [19] for a thorough treatment.

6 Conclusions

We considered the use of distance-preserving maps as a data perturbation technique for privacy-preserving data mining. On the one hand, this technique is quite useful as it is computationally efficient, and it allows many interesting data mining algorithms to be applied directly to the perturbed data and produce an error-free result *e.g.* K-means clustering and k-nearest neighbor classification. On the other hand, the privacy offered by distance preserving transformations has, to our knowledge, not been well-studied. We take a step in this direction by considering two types of prior knowledge an attacker may have and use to design attack techniques to recover the original data. The first is based on basic properties of linear algebra and the second on principal component analysis.

We conclude the paper by pointing out a potential remedy to the privacy problems described earlier for the PCA attack. Recall that the attacker, with a good estimate of the original and transformed covariance matrices, could gain a lot of information about the orthogonal transformation T itself and, therefore, undo it quite well to recover the original data. We suggest, however, that the data owner instead use a randomized transformation which is orthogonal on expectation – namely, random projection. The owner generates $\hat{R}$, a $\ell \times n$ matrix with each entry sampled independently from a distribution with mean zero and variance one and releases $Y = RX$ where $R = \ell^{-1/2}\hat{R}$ (this type of data perturbation for $\ell \leq n$ was discussed in [14]). It can be shown that matrix R is orthogonal on expectation and the probability of orthogonality approaches one exponentially fast with ℓ. By increasing ℓ, the data owner can guarantee that distances are preserved with arbitrarily high probability. However, it can be shown that the randomness introduced by R kills the covariance in Y used by the PCA based attack. Specifically, given random vector V, it can be shown that, Σ_{RV} (the covariance matrix of RV) equals $I_n\gamma$ for some constant γ. *Any* vector in $\mathbb{R}^n$ is an eigenvector of Σ_{RV}, therefore, the PCA based attack will not work. The exploration of this kind of randomized orthogonal transformation is a good direction for future work.

7 Acknowledgment

This research is supported by the U.S. NSF Grant IIS-0329143. H. Kargupta also received partial support from NSF CAREER award IIS-0093353. The authors thank R. Wolff for several helpful discussions.

References

1. Agrawal, R., Srikant, R.: Privacy preserving data mining. In: Proc. ACM SIGMOD. (2000) 439–450
2. Kargupta, H., Datta, S., Wang, Q., Sivakumar, K.: Random data perturbation techniques and privacy preserving data mining. Knowledge and Information Systems **7**(5) (2005) 387–414
3. Huang, Z., Du, W., Chen, B.: Deriving private information from randomized data. In: Proc. ACM SIGMOD. (2005) 37–48
4. Sweeney, L.: K-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10**(5) (2002) 557–570
5. Chen, K., Liu, L.: Privacy preserving data classification with rotation perturbation. In: Proc. IEEE ICDM. (2005) 589–592
6. Oliveira, S.R.M., Zaïane, O.R.: Privacy preservation when sharing data for clustering. In: Proc. Workshop on Secure Data Management in a Connected World. (2004) 67–82
7. Artin, M.: Algebra. Prentice Hall (1991)
8. N. R. Adam, J.C.W.: Security-control methods for statistical databases: A comparative study. ACM Computing Surveys **21**(4) (1989) 515–556
9. Jolliffe, I.T.: Principal Component Analysis. Second edn. Springer Series in Statistics. Springer (2002)
10. G. Strang: Linear Algebra and Its Applications (3rd Ed.). Harcourt Brace Jovanovich College Publishers, New York (1986)
11. Székély, G.J., Rizzo, M.L.: Testing for equal distributions in high dimensions. InterStat **November**(5) (2004)
12. Vaidya, J., Clifton, C., Zhu, M.: Privacy Preserving Data Mining. Volume 19 of Series: Advances in Information Security. Springer (2006)
13. Kim, J.J., Winkler, W.E.: Multiplicative noise for masking continuous data. Technical Report Statistics #2003-01, Statistical Research Division, U.S. Bureau of the Census (2003)
14. Liu, K., Kargupta, H., Ryan, J.: Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining. IEEE Transactions on Knowledge and Data Engineering **18**(1) (2006) 92–106
15. Evfimevski, A., Gehrke, J., Srikant, R.: Limiting privacy breaches in privacy preserving data mining. In: Proc. ACM PODS. (2003)
16. Rizvi, S.J., Haritsa, J.R.: Maintaining data privacy in association rule mining. In: Proc. 28th VLDB. (2002) 682–693
17. Hore, B., Mehrotra S., Tsudik G.: A privacy-preserving index for range queries. In: Proc. 30th VLDB. (2004) 720–731
18. Verykios, V.S., Elmagarmid, A.K., Elisa, B., Saygin, Y., Elena, D.: Association rule hiding. IEEE Transactions on Knowledge and Data Engineering **16**(4) (2004) 434–447
19. Fienberg, S.E., McIntyre, J.: Data swapping: Variations on a theme by dalenius and reiss. Technical report, U.S. National Institute of Statistical Sciences (2003)