國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering & Computer Science

National Taiwan University

Master Thesis

距離及相關性資料探勘演算法的隱私權保護

Privacy Preservation for Distance and Correlation-based Mining Algorithms

蘇俊維

Chun-Wei Su

指導教授：陳銘憲 博士

Advisor: Ming-Syan Chen, Ph.D.

中華民國 98 年 7 月

July, 2009

國立臺灣大學碩士學位論文
口試委員會審定書

距離及相關性資料探勘演算法的隱私權保護
Privacy Preservation for Distance and Correlation-based Mining Algorithms

本論文係蘇俊維君（R96942126）在國立臺灣大學電信工程學研究所完成之碩士學位論文，於民國 98 年 6 月 19 日承下列考試委員審查通過及口試及格，特此證明

口試委員：陳銘憲（簽名）

（指導教授）

曾新穆

沈錳坤

鄧維光　葉彌妍

王暉

系主任、所長　王暉（簽名）

Acknowledgements

At the very first, I have to express my deepest gratitude to my advisor, Prof. Ming-Syan Chen, who has offered me valuable ideas and criticisms with his profound knowledge and rich research experience. I'm obliged to his efforts of helping me to complete the thesis. This thesis could not be finished without his unconditional help and priceless suggestions.

Moreover, I would like to thank my oral committees, who gave me invaluable comments and recommendations. I would also like to thank my family and friends for their support and encouragement all the way from the beginning of this work. Thank you.

中文摘要

這篇論文設計出一個轉換方式使得當資料送到第三方被研究時還能保護到資料的隱私性。大部分傳統的轉換方式都有兩種限制，演算法侷限性與資訊量流失。在這篇論文中，我們提出了一個新穎的隱私權保護方式而沒有這兩種限制。這種轉換演算法我們稱之為 FISIP: 一階和、二階和與內積維護。特別的是，我們將證明，藉由 FISIP 保護隱私資料的這三種性質（一階和、二階和與內積），當它被轉換成公開資料時，資料還能用在只依據這三種性質的所有演算法。由於距離與相關性能從這三種性質推導出來，因此，只依據距離與相關性的所有演算法依舊能被應用到。FISIP 的評估有兩部分，第一部分是資料的有用性，第二部分是資料的強大性，這兩個目標本質上很難同時被達到。然而，從我們的實驗結果顯示，FISIP 能同時滿足這兩個目標。總而言之，FISIP 能提供一種轉換使得轉換前的原始資料與轉換後的公開資料的距離及相關性皆一致。當資料的隱私性被保護到時，資料的探勘品質在轉換後的（公開）資料能與轉換前的（隱私）資料達到一致。

關鍵字：資料探勘, 隱私權保護, 距離性, 相關性

ABSTRACT

This paper devises a transformation scheme to protect data privacy in the case that data has to be sent to the third party for analysis purpose. Most conventional transformation schemes suffer from two limits, i.e. algorithm dependency and information loss. In this paper, we propose a novel privacy preserving scheme without these two limitations. This transformation algorithm is referred to as FISIP: FIrst and Second order sum and Inner product Preservation. Explicitly, as will be proved, by preserving three basic properties, (i.e. first order sum, second sum, and inner products) of private data, algorithms whose measures can be derived from the three properties can still be applied to public data transformed by FISIP. Specifically, distance and correlation can be derived from the three properties. Hence, distance-based algorithms and correlation-based algorithms can be applied. Evaluation of FISIP is done in two parts. The first part is data usefulness. The second part is data robustness. The two goals are intrinsically difficult to achieve at the same time. However, FISIP attains these two goals shown by our experimental results later. In all, FISIP is able to provide a transformation that preserves the distance and the correlation for the original private data after their transformation to the public data. As a result, while the privacy is protected, the mining quality from the transformed (public) data can be obtained to be the same as that from the original (private) data.

Index Terms — data mining, privacy preserving, distance-based, correlation-based

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Privacy infringement is an important issue in data mining. People or organizations usually tend not to provide their data or locations because of the privacy concern [14] [20]. Hence, to conduct effective data mining, privacy preservation has become an research issue to address.

Note that there are two major limitations in most privacy preserving approaches. The first limitation is due to the algorithm dependency in that the protection schemes are intrinsically incorporated into certain mining algorithms. Such protection schemes are hard to be generalized for other algorithms. The second limitation is information loss. Most algorithms add some controlled noise in their private data or truncate part of private data to make them unrecoverable to the original one. Though privacy is thus protected, it usually has side effects: the mining results are somewhat altered due to these changes. In this paper, we propose a novel privacy preserving scheme without these two limitations. This transformation algorithm is referred to as FISIP: FIrst and Second order sum and Inner product Preservation.

Explicitly, as will be proved, by preserving three basic properties, (i.e. first order sum, second sum, and inner products) of private data, algorithms whose measures can be derived from the three properties can still be applied to public data transformed by FISIP. Specifically, distance and correlation can be derived from the three properties. Hence, distance-based algorithms and correlation-based algorithms can be applied. To be more concrete, we list part of practical and applicable algorithms below.

- The most well-known clustering algorithm, k-means [16] is applicable since it only deals with

distance. K-medoids and fuzzy c-means are also the same case. Another clustering algorithm, for example, DBSCAN [7] is also applicable since the preserved distances can keep the relative density of transformed data. Correlation clustering [1] [5] is also one of the applications. The most well-known classification algorithm, kNN can still be applied because it is neighbor-based, or to be more concisely, distance-based. Another well known classification, support vector machine is also applicable, because inner products between vectors are also the same and then suitable kernel functions can be used.

- Feature selection is an important task in pattern recognition and machine learning. It can reduce redundant information and enhance program performance. There are many types of feature selection and here we focus on correlation-based feature selection (CFS) [8]. Note that the properties we preserved here are horizontally-partitioned based where correlations between records (instead of between attributes) are kept to be the same. Note that when we want to preserve vertically-partitioned properties, and we can simply transpose private data before transformation.

- Sometimes, input data are not suitable for algorithm's needs and require to be normalized or standardize before feeding into algorithms. However, the existence of outliers will make process of normalization or standardization biased and lead to inferior results. Outliers can be detected by checking the distances between records [4] [11], which can be carried out faithfully if FISIP is employed since the corresponding distance is preserved. When distances are preserved, outliers remain.

Moreover, if some ingenious algorithms are invented in the future and their measurement only depends on the properties we already preserved, that algorithm can still be applied and does not need any special privacy-preserving procedures.

Evaluation of FISIP is done in two parts. The first part is data usefulness. Does the transformed public data preserve the relations of original private data? The second part is data robustness. Is the transformed data robust enough against reconstruction attack? The two goals are intrinsically difficult

to achieve at the same time. However, FISIP attains these two goals shown by our experimental results later.

In all, by preserving some mathematical properties, FISIP is able to provide a transformation that preserves the distance and the correlation for the original private data after their transformation to the public data. As a result, while the privacy is protected, the mining quality from the transformed (public) data can be obtained to be the same as that from the original (private) data.

The rest of this paper is organized as follows. Preliminaries are given in Chapter 2 where related work is reviewed and the problem description is given. Theoretical properties of FISIP are derived in Chapter 3, and systematic procedures to construct FISIP transformations are presented in Chapter 4 and Chapter 5. Experimental studies are conducted in Chapter 7 and this paper concludes with Chapter 8.

Chapter 2

Preliminaries

2.1 Related Work

Most privacy preservation schemes can be roughly categorized into two fields. The first field targets on hiding the data entities of published database. This field of research becomes popular in recent years because of the threat of quasi-identifiers.

It was reported that 87% of US citizen can be uniquely identified by only their zip code, date of birth and gender [19]. Even though their names or social security numbers are truncated from published database, identities of citizen may still be found via easily obtainable fields by linking attacks.

Typical solutions to this concern are based on the k-anonymity model. The published data set is generalized such that there are at least k records in each group of quasi-identifiers. Other well-known models for protecting data entities include l-diversity [10] and t-closeness [12], etc.

The second field of privacy preservation focuses on hiding data values, instead of entities. Most well-known schemes in this field are based on data perturbation [3] [20]. Consider we draw n values $x_1, x_2, \ldots, x_n$ from original, private data distribution X, and perturb them by adding n independent values $y_1, y_2, \ldots, y_n$ drawn by a random variable Y. As long as the probability distribution function of Y is known, the distribution of X can be reconstructed by $x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n$.

However, even the distribution can be reconstructed; it cannot guarantee to have the same mining result from the private database and the reconstructed public database. It also cannot hold analytical properties, such as having the first order sum and the second order sum be the same under privacy

preservation transformation. Anonymity-based preservation also suffers from this drawback, since the process of anonymization generalized the original data values into ambiguous data intervals. Some information is lost, thus making the mining result from the transformed data be different from that from the original private database.

In most cases, we want to preserve basic properties that are only critical to algorithms. Reconstruction based preservation is not able to preserve such properties. For example, the most well-known data mining algorithms, such as k-means and kNN, use distance as basic metric function for clustering and classification. The above reconstruction approach will result in distance error depending on the variance of random variable Y.

Some studies have been reported on preserving such basic properties. For instance, condensation approach [2] condenses the data into multiple groups, which have at least k records, say, $\{X_1, ..., X_k\}$ and a record X_i contains d dimensions as $(x_i^1, ..., x_i^d)$. Within each condensed group, the vertically partitioned first order sum $\sum_{t=1}^{k} x_t^j$ and second order sum $\sum_{t=1}^{k} \left(x_t^j\right)^2$ are preserved.

There is another work on [17] that preserves distances by Fourier transform. It horizontally transforms each record into frequency domain and vertically truncates small coefficients to strike on a trade-off between mining quality and privacy preservation. Note, however, that from the point of view of mining quality, it cannot produce exact mining result between private and public database. From the point of view of mining versatility, it preserves only distances, which is more limited and less satisfactory than our works.

The problem we want to solve is to find a transformation that is both distance preserving and correlation preserving. The works in [2] [17] cannot preserve inner products, and thus cannot preserve correlation. The work in [15] can preserve inner products and correlation, but the private data publisher needs to horizontally normalize his private data to zero mean. The first order sums of private records are lost! Relations between vertically-partitioned attributes are lost, too. Our work needs no such normalization and we do not lost the information of first order sum and relations between attributes. This is due to the attracting property of our spreading matrix, as will be introduced later, the sum of each column is equal to one. As will be proved in Chapter 3, FISIP can preserve inner products, and is

thus able to preserve correlation. To the best of our knowledge, our work in this paper is the first work that can preserve correlation without normalization.

Studies in privacy preservation can be classified into two fields as stated above. This paper focuses on the first type of problem and targets at numerical data.

2.2 Problem Description

In this thesis, we want to devise a transformation that preserves the distance and the correlation for the original private data after their transformation to the public data. Moreover, protection against possible attacks via matrix perturbation is also considered. To facilitate our presentation, we here introduce the following definitions.

Definition 1: Distance Preserving Transformation: In k-dimensional vector space, given n points $\{r_1, r_2, ..., r_n\}$, a transformation T_d that can produce n points $\{u_1, u_2, ..., u_n\}$, such that $dist(u_i, u_j) = dist(r_i, r_j)$, $u_i = T_d(r_i)$, $1 \leq i, j \leq n$ is defined as distance preserving transformation.

The distance metric used in this paper is Euclidean distance. However, the results are of general usefulness since it has been shown that most distance metrics can be reduced to Euclidean form [13].

Definition 2: Correlation Preserving Transformation: In k-dimension vector space, given n points $\{r_1, r_2, ..., r_n\}$, a transformation T_c that can produce n points $\{u_1, u_2, ..., u_n\}$, such that $corr(u_i, u_j) = corr(r_i, r_j)$, $u_i = T_c(r_i)$, $1 \leq i, j \leq n$ is defined as correlation preserving transformation, where $corr(x, y)$ denotes the correlation of x and y.

For vectors $\mathbf{x} = [x_i]$, $\mathbf{y} = [y_i]$, $1 \leq i \leq k$, the definition of the correlation is formulated as below, which is commonly adopted in the literature [18],

$$\frac{k\sum_{i=1}^{k} x_i y_i - \sum_{i=1}^{k} x_i \sum_{i=1}^{k} y_i}{\sqrt{k\sum_{i=1}^{k} x_i - (\sum_{i=1}^{k} x_i)^2}\sqrt{k\sum_{i=1}^{k} y_i - (\sum_{i=1}^{k} y_i)^2}}.$$

Example 1: Suppose we have four 4-dimensional private column vectors, r_1 to r_4, and we want to transform them into public vectors via distance and correlation preserving transformation T, then T can produce four corresponding public column vectors, u_1 *to* u_4. The construction of A will be described

in Chapter 4.

$$\begin{bmatrix} r_1 & r_2 & r_3 & r_4 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 1 & 7 \\ 3 & 5 & 1 & 6 \\ 2 & 6 & 6 & 7 \\ 4 & 0 & 2 & 8 \end{bmatrix}$$

$$\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \end{bmatrix} = \begin{bmatrix} 4.5 & 4.0 & 4.0 & 7.0 \\ 1.5 & 2.0 & 4.0 & 8.0 \\ 2.5 & 1.0 & -1.0 & 7.0 \\ 0.5 & 7.0 & 3.0 & 6.0 \end{bmatrix}$$

$$A = \begin{bmatrix} -0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & -0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & -0.5 \end{bmatrix}$$

The distance of each pair is preserved. $dist(r_1, r_2) = dist(u_1, u_2) = 6.7082$. For example, in the case of kNN, the nearest three neighbors of r_2 are, in order, r_3, r_1, and r_4. In transformed domain, the nearest three neighbors of u_2 are, in order, u_3, u_1, and u_4. For correlation preserving, $corr(r_1, r_2) = corr(u_1, u_2) = -0.3320$. ■

Chapter 3

Theoretical Properties of FISIP

We first consider the distance preserving. For private vectors r_i and public vectors u_i, the distance between public vectors can be written as $(u_i - u_j)^\intercal (u_i - u_j) = (r_i - r_j)^\intercal A^\intercal A(r_i - r_j)$, $r_i, u_i \in R^{k\times 1}$. If A is orthogonal, then the distance is preserved. However, it can not guarantee the preservation of correlation between vectors. Before we explore the preservation of correlation, we define the FISIP transformation as follows.

Definition 3: FISIP Transformation: A linear transformation with its matrix representation A is called a FISIP transformation if A has the following three properties: $u_i = Ar_i,\ A \in R^{k\times k}, A = \left[\ A_1 \quad A_2 \quad ... \quad A_k\ \right]$, $u_i, r_i, A_i \in R^{k\times 1}$, $A_i = [A_{im}]$, $1 \le m \le k$, $A_{im} \in R$,

1. $\sum_{m=1}^{k} A_{im} = 1$
2. $\sum_{m=1}^{k} A_{im}^2 = 1$
3. $\sum_{m=1}^{k} A_{im}A_{jm} = 0, for\ i \neq j.$

We then have the following three lemmas for a FISIP transformation.

Lemma 1 (First Order Sum Preservation): For a FISIP transformation $u_i = Ar_i$, it follows that $\sum_{m=1}^{k} u_{im} = \sum_{m=1}^{k} r_{im}$.

Proof. We expand the expression of $\sum_{m=1}^{k} u_{im}$ and checks if it can be reduced to $\sum_{m=1}^{k} r_{im}$.

$$\begin{aligned}
\sum_{m=1}^{k} u_{im} &= \sum_{m=1}^{k} (u_i)_m \\
&= \sum_{m=1}^{k} \left(\begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} \begin{bmatrix} r_{i1} \\ r_{i2} \\ \vdots \\ r_{ik} \end{bmatrix} \right)_m \\
&= \sum_{m=1}^{k} [r_{i1}A_{1m} + r_{i2}A_{2m} + \ldots + r_{ik}A_{km}] \\
&= \left[r_{i1} \sum_{m=1}^{k} A_{1m} + \ldots + r_{ik} \sum_{m=1}^{k} A_{km} \right] \\
&= r_{i1} + r_{i2} + \cdots + r_{ik} = \sum_{m=1}^{k} r_{im}.
\end{aligned}$$

■

Lemma 2 (Second Order Sum Preservation): For a FISIP transformation $u_i = Ar_i$, it follows that $\sum_{m=1}^{k} u_{im}^2 = \sum_{m=1}^{k} r_{im}^2$.

Proof. We expand the expression of $\sum_{m=1}^{k} u_{im}^2$ and checks if it can be reduced to $\sum_{m=1}^{k} r_{im}^2$.

$$\begin{aligned}
\sum_{m=1}^{k} u_{im}^2 &= \sum_{m=1}^{k} (u_i)_m^2 \\
&= \sum_{m=1}^{k} \left(\begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} \begin{bmatrix} r_{i1} \\ r_{i2} \\ \vdots \\ r_{ik} \end{bmatrix} \right)_m^2 \\
&= \sum_{m=1}^{k} [r_{i1}A_{1m} + r_{i2}A_{2m} + \ldots + r_{ik}A_{km}]^2 \\
&= \left[r_{i1}^2 \sum_{m=1}^{k} A_{1m}^2 + \ldots + r_{ik}^2 \sum_{m=1}^{k} A_{km}^2 \right] \\
&= r_{i1}^2 + r_{i2}^2 + \cdots + r_{ik}^2 = \sum_{m=1}^{k} r_{im}^2.
\end{aligned}$$

■

Lemma 3 (Inner Product Preservation): For a FISIP transformation $u_i = Ar_i$, it follows that $\sum_{m=1}^{k} u_{im}u_{jm} = \sum_{m=1}^{k} r_{im}r_{jm}$.

Proof. We expand the expression of $\sum_{m=1}^{k} u_{im}u_{jm}$ and checks if it can be reduced to $\sum_{m=1}^{k} r_{im}r_{jm}$.

$$\sum_{m=1}^{k} u_{im}u_{jm} = u_i \cdot u_j = u_j^\intercal u_i = r_j^\intercal A^\intercal A r_i$$

$$= r_j^\intercal \begin{bmatrix} A_1^\intercal \\ A_2^\intercal \\ \vdots \\ A_k^\intercal \end{bmatrix} \begin{bmatrix} A_1 & A_2 & \cdots & A_k \end{bmatrix} r_i$$

$$= r_j^\intercal r_i = r_i \cdot r_j = \sum_{m=1}^{k} r_{im}r_{jm}.$$

■

These three lemmas lead to Theorem 1, which states the property of distance and correlation preserving for a FISIP transformation.

Theorem 1 (Property of FISIP transformation): FISIP transformation is both distance and correlation preserving transformation.

Proof. For distance preserving, we can show that $(u_i - u_j)^\intercal (u_i - u_j) = (r_i - r_j)^\intercal A^\intercal A (r_i - r_j) = (r_i - r_j)^\intercal (r_i - r_j)$, therefore, distance is preserved.

For correlation preserving, it follows from Lemma 1, Lemma 2, and Lemma 3 that $\sum_{m=1}^{k} u_{im} = \sum_{m=1}^{k} r_{im}$, $\sum_{m=1}^{k} u_{im}^2 = \sum_{m=1}^{k} r_{im}^2$, $\sum_{m=1}^{k} u_{im}u_{jm} = \sum_{m=1}^{k} r_{im}r_{jm}$, for a FISIP transformation. The correlation preservation thus follows. ■

Chapter 4

Perfect FISIP Transformation

Note that an orthogonal matrix $A = \begin{bmatrix} A_1 & A_2 & \dots & A_k \end{bmatrix}$, $A_i \in R^{k\times 1}$ has the properties of $\sum_{m=1}^{k} A_{im}^2 = 1$, $\sum_{m=1}^{k} A_{im}A_{jm} = 0$, $i \neq j$, However, an orthogonal matrix A may not have the property of $\sum_{m=1}^{k} A_{im} = 1$. Finding a FISIP matrix which corresponds to a FISIP transformation was not solved before. In Section 4.1, we devise a general procedure to construct FISIP matrices and fast computation in Section 4.2. Transformation proposed in this chapter can perfectly preserve the distance and correlation of data, and therefore, we name the transformation introduced in this chapter as Perfect FISIP Transformation, or simply Perfect FISIP. We will introduce another type of transformation, Strong FISIP in Chapter 5.

4.1 General Form Realization

We firstly define a form of base matrix, which is referred to as spreading matrix, and prove that it is a form of FISIP matrices.

Definition 4: Spreading matrix: A k-dimensional spreading matrix, denoted by $A^{[k]}$, is defined as a k by k matrix as constructed by the following formula. Any row permutations of spreading matrices are also spreading matrices.

1. Basic type $A^{[k]} = \begin{bmatrix} \frac{2-k}{k} & \frac{2}{k} & \frac{2}{k} & \dots & \frac{2}{k} \\ \frac{2}{k} & \frac{2-k}{k} & \frac{2}{k} & \dots & \frac{2}{k} \\ \frac{2}{k} & \frac{2}{k} & \frac{2-k}{k} & \dots & \frac{2}{k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{2}{k} & \frac{2}{k} & \frac{2}{k} & \dots & \frac{2-k}{k} \end{bmatrix}$.

2. Composition type: $A_c^{[k]} = \begin{bmatrix} A^{[k_1]} & 0 & 0 & \cdots & 0 \\ 0 & A^{[k_2]} & 0 & \cdots & 0 \\ 0 & 0 & A^{[k_3]} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A^{[k_n]} \end{bmatrix}$

3. Derived type: any permutation of the basic type or composition type.

For $A^{[k]} = [a_{ij}]$, $1 \leq i, j \leq k$, $a_{ii} = \frac{2-k}{k}$, $a_{ij} = \frac{2}{k}$ for $i \neq j$. For example, FISIP matrices constructed this way are

$$A^{[2]} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, A^{[3]} = \begin{bmatrix} \frac{-1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{-1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{-1}{3} \end{bmatrix}.$$

$$A^{[4]} = \begin{bmatrix} \frac{-2}{4} & \frac{2}{4} & \frac{2}{4} & \frac{2}{4} \\ \frac{2}{4} & \frac{-2}{4} & \frac{2}{4} & \frac{2}{4} \\ \frac{2}{4} & \frac{2}{4} & \frac{-2}{4} & \frac{2}{4} \\ \frac{2}{4} & \frac{2}{4} & \frac{2}{4} & \frac{-2}{4} \end{bmatrix}$$

Theorem 2 states that spreading matrices are FISIP matrices.

Theorem 2 (Property of $A^{[k]}$): A linear transformation with a spreading matrix representation $A^{[k]}$ is a FISIP transformation.

Proof. For a k by k spreading matrix $A^{[k]}$, the first order sum of each column is $\frac{2-k}{k} + \frac{2}{k} + \cdots + \frac{2}{k} = \frac{2-k}{k} + \frac{2}{k} \times (k-1) = 1$. The second order sum of each column is $\left(\frac{2-k}{k}\right)^2 + \left(\frac{2}{k}\right)^2 + \cdots + \left(\frac{2}{k}\right)^2 = \left(\frac{2-k}{k}\right)^2 + \left(\frac{2}{k}\right)^2 \times (k-1) = \frac{4-4k+k^2+4k-4}{k^2} = 1$. Each pair of inner product is $\frac{2-k}{k} \times \frac{2}{k} + \frac{2}{k} \times \frac{2-k}{k} + \frac{2}{k} \times \frac{2}{k} + \cdots + \frac{2}{k} \times \frac{2}{k} = \frac{8-4k}{k^2} + \frac{4}{k^2} \times (k-2) = 0$. Since spreading matrices satisfies the three properties of FISIP transformation, a linear transformation with its matrix representation $A^{[k]}$ is FISIP transformation. ■

4.2 FISIP Matrices for Fast Computation

Note that there are other methods conceivable to do the transformation. One can construct an high dimensional FISIP matrix directly or use low dimensional spreading matrix as a building block to construct a high dimensional FISIP matrix. The advantage of using low dimensional spreading matrix as base matrix is that the constructed matrix has more zeros, which can reduce the calculation time of

public vectors. Hence we next propose a procedure to construct FISIP matrix for fast computation and introduce the method by Example 2.

Example 2: Suppose we have two private 8-dimensional vectors, say, R_1 =[12 32 48 4 -4 6 58 74]$^\intercal$and R_2 =[26 18 16 2 11 72 -31 20]$^\intercal$ and we want to transform them to public vectors U_1, U_2. The method is as follows. We use the 4-dimensional spreading matrix, $A^{[4]}$, to build the 8-dimensional FISIP matrix, A_{new}.

$$
\begin{aligned}
U_1 &= A_{new}R_1 = P_r \begin{bmatrix} A^{[k]} & 0 & 0 & \cdots & 0 \\ 0 & A^{[k]} & 0 & \cdots & 0 \\ 0 & 0 & A^{[k]} & \vdots & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 0 & A^{[k]} \end{bmatrix} P_c R_1 \\
&= P_r \begin{bmatrix} \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} \end{bmatrix} P_c R_1 \\
&= \begin{bmatrix} \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{-1}{2} \end{bmatrix} P_c R_1 \\
&= \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{-1}{2} & 0 & \frac{1}{2} \\ \frac{-1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{-1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{-1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{-1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{-1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{-1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{-1}{2} & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix} R_1 \\
&= [\ 52 \quad 45 \quad -16 \quad -1 \quad 61 \quad 26 \quad 54 \quad 9\ ]^\intercal
\end{aligned}
$$

$$P_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_c = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} U_2 &= A_{new} R_2 \\ &= [\ -16 \quad -15 \quad 36 \quad 42 \quad 0 \quad 38 \quad 54 \quad -5\ ]^\intercal \end{aligned}$$

P_r and P_c are the row and column permutation matrices choosed randomly by private data publisher. For example, it can permute row 2 to 6, row 4 to 7, column1 to 6 and column 3 to 8 to break the block structure. After calculation, we get U_1=[52 45 -16 -1 61 26 54 9]$^\intercal$ and U_2 =[-16 -15 36 42 0 38 54 -5]$^\intercal$. For 8-dimensional case, it will preserve properties listed below

TABLE 4.1
Preserved properties of private, public pairs

x_1x_2	$\left(\sum_{i=1}^{8} x_{1i}, \sum_{i=1}^{8} x_{2i}\right)$	$\left(\sum_{i=1}^{8} x_{1i}^2, \sum_{i=1}^{8} x_{2i}^2\right)$	$\sum_{i=1}^{8} x_1x_2$	$\overline{x_1x_2}$	$\rho_{x_1x_2}$
R_1R_2	$(230, 104)$	$(12380, 7926)$	1734	129.7613	-0.3701
U_1U_2	$(230, 104)$	$(12380, 7926)$	1734	129.7613	-0.3701

The computation of transforming one 8-dimensional private vector by using $A^{[8]}$ directly needs 8×8 multiplications and 8×7 additions, but it can be reduced to 8×4 multiplications and 8×3 additions

by using A_{new}, as stated above. For a database with ten thousand records, we save $10,000 \times 8 \times 4$ multiplications and $10,000 \times 8 \times 4$ additions. ■

In general, for $u = A^{[k]}r$ and $u, r \in R^{k\times 1}$, each element of u needs k multiplications and $k-1$ additions, i.e. $O(k \times multi + (k-1) \times add)$. Here we denote the computation of multiplication as $multi$ and computation of addition as add. To obtain the vector of u, the computation needs $O(k^2 \times multi + k \times (k-1) \times add)$. However, each element of u does not necessarily need k multiplications and $k-1$ additions. The computation for each element of u can be reduced to c multiplications and $c-1$ additions, where c is a constant. Thus, the computation of obtaining u can be reduced to $O(k \times c \times multi + k \times (c-1) \times add) = O(k \times multi + k \times add)$. If the database contains n private records, the computation is $O(n \times k \times multi + n \times k \times add)$, not $O(n \times k^2 \times multi + n \times k^2 \times add)$.

4.3 Variation of Transformations

We have already shown how to generate public vectors from private vectors with distance and correlation preservation, and now we are going to evaluate how well the privacy is protected. How to quantify the protection of privacy? Here we define it as the number of possible variations of private vectors when public vectors are given.

$$\begin{aligned} [u_i] &= P_r A_{normal} P_c [r_i] \\ [r_i] &= P_c^{-1} A_{normal}^{-1} P_r^{-1} [u_i] \\ &= P_c A_{normal}^{\intercal} P_r [u_i] \end{aligned}$$

We can calculate it by observing the possible variations of transformation matrix $P_r A_{normal} P_c$ and find the number of transformation increases factorially as k increases with a lower bound of $k!$, which means finding real private vectors from possible private vectors is very infeasible.

Chapter 5

Strong FISIP Transformation

5.1 Privacy Enhancement via Matrix Perturbation

Special protections can be implemented to prevent attackers from doing reverse transform. Given public data, what can attackers do to get the correct private counterpart? At first, independent component analysis (ICA) [9] seems to be an effective attack in this situation, however, it is not practical. The requirement of ICA is that the input data needs to be not correlated, for example, audio signal, but records in databases typically have some correlations between attributes and thus make ICA an inefficient attack [6].

Let us consider a more effective attack: the attacker has a few samples of private data and its corresponding public data. This kind of attack is referred to as known private data attack, or simply known data attack. Known data attack is more threatening, and we will propose a counter measure to prevent the public data being inverse-transformed.

Consider an n-dimensional vector space and the attacker has already acquired m, $m \geq n$, linearly independent private vectors and their transformed versions. Typically, the attacker can calculate remaining private vectors by linear combination. Consider the following example.

Example 4: Suppose the attacker knows the following transforming pair

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} 1 & 9 & 8 & 2 \\ 8 & 7 & 9 & 1 \\ 1 & 0 & 1 & 5 \\ 9 & 1 & 2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 9 & 1 & 2 & 8 \\ 4.5 & 5.5 & 3.5 & 11.5 \\ 2.5 & 3.5 & 2.5 & 1.5 \\ -3 & 5 & 4 & 6 \end{bmatrix}$$

and now he gets the public vector u=[10 -3 21 4]. He doesn't know what the private vector is, but he can do the following and and find that

$$\begin{aligned}
& \left(\left(\begin{bmatrix} 9 & 1 & 2 & 8 \\ 4.5 & 5.5 & 3.5 & 11.5 \\ 2.5 & 3.5 & 2.5 & 1.5 \\ -3 & 5 & 4 & 6 \end{bmatrix}^{\top} \right)^{-1} \right) \begin{bmatrix} 10 \\ -3 \\ 21 \\ 4 \end{bmatrix} \\
&= \begin{bmatrix} 10.48 & -12.24 & 0.65 & 10.31 \end{bmatrix}^{\intercal}. \\
u &= 10.48u_1 - 12.24u_2 + 0.65u_3 + 10.31u_4 \\
&= 10.48Ar_1 - 12.24Ar_2 + 0.65Ar_3 + 10.31Ar_4 \\
&= A(10.48r_1 - 12.24r_2 + 0.65r_3 + 10.31r_4). \\
r &= (10.48r_1 - 12.24r_2 + 0.65r_3 + 10.31r_4) \\
&= \begin{bmatrix} 6 & 19 & -5 & 12 \end{bmatrix}^{\top}
\end{aligned}$$

■

Defensive manipulations to avoid such kind of attack can be done if we are willing to give up some mining accuracy. If each record is originally multiplied by different A, then corresponding records will not be calculated easily. Methods for making different transformation matrix A is quite easy and efficient. The method that we produce different A is to use data perturbation. Special perturbation can be done on A to preserve first order sum by making the sum of each column's perturbation equal to zero. For example, we denote the perturbed A as A'. for each column i of A, we randomly select row j and set element of A' as $a'_{ji} = a_{ji} - 2^{pert}$, $a'_{ki} = a_{ki} + \frac{2^{pert}}{d-1}$, $pert \leq 1$ for $k \neq j$ and d is the number

of rows. The sum of each column of A' is

$$\begin{aligned} a'_{ji} + \sum_{k \neq j} a'_{ki} &= \left(a_{ji} - 2^{pert}\right) + \sum_{k \neq j} \left(a_{ki} + \frac{2^{pert}}{d-1}\right) \\ &= a_{ji} + \sum_{k \neq j} a_{ki} - 2^{pert} + (d-1) \times \frac{2^{pert}}{d-1} = 1. \end{aligned}$$

Therefore, the sums of transformed vectors are still equal to private vectors. We can formulate the estimation error analytically as follows. For k-dimensional vectors, the attack has k linearly independent vectors, r_1 *to* r_k, at hand and he want to find the corresponding unknown private vector r_x by the known public vector u_x. We assume that $u_x = A_x r_x$, $u_1 = A_1 r_1, \cdots, u_k = A_k r_k$. Therefore, the error of the estimated private vector r_{est}, is

$$\begin{aligned} |r_{est} - r_x| &= \left| \sum_{i=1}^{k} \alpha_i r_i - A_x^{-1} u_x \right| \\ &= \left| \sum_{i=1}^{k} \alpha_i r_i - A_x^{-1} (\sum_{i=1}^{k} \alpha_i u_i) \right| \\ &= \left| \sum_{i=1}^{k} \alpha_i r_i - A_x^{-1} (\sum_{i=1}^{k} \alpha_i A_i r_i) \right| \\ &= \left| \sum_{i=1}^{k} \alpha_i (I_k - A_x^{-1} A_i) r_i \right| \end{aligned}$$

If each record is transformed by the same A, i.e., $A_x = A_i$, then the error is zero. Otherwise, the error will increase as the dimension k increases. We can see from Fig. 7.8 that the perturbation method can effectively produce different A and make the estimation error large.

Chapter 6

Dimension Adaptation

It is desired that data owner can adjust the dimension of its published data since the dimension itself may sometimes be sensitive, too. Moreover, suppose attackers know the permutation matrices used in transformation, they cannot do reverse transformation because the dimension of private data must be known for perfect reconstruction. Here we propose a new modification of our solution to this concern.

6.1 Up Dimension: from k to $k + c$

For private vectors with dimension k, it can be easily transform to $k + c$ dimensions ($c \in N^+$) by concatenating c-dimensional zero vector. The permutation matrix, P_r and Pc, are both k+c dimensions.

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{k+c} \end{bmatrix} = P_r A^{[k+c]} P_c \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

For the case of dimension expansion, distances are still perfectly preserved, but the change of dimension varies a coefficient in correlation's formula, and thus correlation is no longer preserved.

6.2 Down Dimension: from k to $k-c$

We can lower the dimension of public data by the following.

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{k-c} \end{bmatrix} = P_r A_{new} P_c \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{bmatrix},$$

where A_{new} is $\left[A^{[k-c]} | \ A_1 \quad A_2 \quad \cdots \quad A_c \ \right]$, $A_i \in R^{k-c}$, $P_r \in R^{(k-c)x(k-c)}$, $Pc \in R^{kxk}$. If $A_{new}^T A_{new} = I_k$, then distances are still preserved from our previous proof. However, we can not find such matrix A_{new} since in $k-c$ dimensional vector space, we can only find $k-c$ orthogonal bases. In other words, for the criterion $\sum_{m=1}^{k-c} A_{im}A_{jm} = 0, i \neq j$, we can only find $k-c$, not k, different column vectors if the number of rows of A_{im} are $k-c$.

Approximations can be made here. We select c different columns from the first $k-c$ columns of $A^{[k-c]}$. Thus, two constraints of FISIP criterion $\sum_{m=1}^{k-c} A_{im} = 1$ and $\sum_{m=1}^{k-c} A_{im}^2 = 1$ can still be satisfied and $\sum_{m=1}^{k-c} A_{im}A_{jm} = 0, i \neq j$ can be partially satisfied.

Example 3: Suppose we have 6-dimensional private data pairs $r_1 = [\ 1 \quad 3 \quad 7 \quad 5 \quad 6 \quad 2\ ]^\intercal$ and $r_2 = [\ 8 \quad 3 \quad 1 \quad 5 \quad 9 \quad 7\ ]^\intercal$, and we can transform it to 8-dimensional public vectors, u_1, u_2 and 4-dimensional public vectors, l_1, l_2 as follows.

$$u_1 = A^{[8]} \times [\ 1 \quad 3 \quad 7 \quad 5 \quad 6 \quad 2 \quad 0 \quad 0\ ]^\intercal = [\ 5 \quad 3 \quad -1 \quad 1 \quad 0 \quad 4 \quad 6 \quad 6\ ]^\intercal$$

$$u_2 = A^{[8]} \times [\ 8 \quad 3 \quad 1 \quad 5 \quad 9 \quad 7 \quad 0 \quad 0\ ]^\intercal = [\ 0.25 \quad 5.25 \quad 7.25 \quad 3.25 \quad -0.75 \quad 1.25 \quad 8.25 \quad 8.25\ ]^\intercal$$

$$l_1 = \left[\ A^{[4]} \quad A_1^{[4]} \quad A_2^{[4]}\ \right] \times [\ 1 \quad 3 \quad 7 \quad 5 \quad 6 \quad 2\ ]^\intercal = [\ 5 \quad 7 \quad 5 \quad 7\ ]^\intercal$$

$$l_2 = \left[\ A^{[4]} \quad A_1^{[4]} \quad A_2^{[4]}\ \right] \times [\ 8 \quad 3 \quad 1 \quad 5 \quad 9 \quad 7\ ]^\intercal = [\ -0.5 \quad 6.5 \quad 15.5 \quad 11.5\ ]^\intercal$$

TABLE 6.1
Preserved properties of private, public pairs

x_1x_2	$\left(\sum x_{1i}, \sum x_{2i}\right)$	$\left(\sum x_{1i}^2, \sum x_{2i}^2\right)$	$\sum x_1x_2$	$\overline{x_1x_2}$	$\rho_{x_1x_2}$
r_1r_2	$(24, 23)$	$(124, 229)$	117	10.9087	-0.4113
u_1u_2	$(24, 23)$	$(124, 229)$	117	10.9087	0.2590
l_1l_2	$(24, 23)$	$(148, 415)$	201	12.6886	0.1255

■

The advantage of dimension adjustability is that it can hide the dimensionality of private data to public, but the price it takes is that distance and correlation is no longer exactly preserved. From formula, variation of dimension makes the preservation of correlation not easily achievable. However, from the experiment shown in later sections, this method still has good mining quality.

Chapter 7

Experimental Results

We firstly evaluate data usefulness after transformation. Section 7.1 measures the preservation of the basic properties of FISIP under perturbation and Section 7.2 shows the preservation of neighbors, which is served as distance preservation measurement. Correlation preservation is showed in Section 7.3. Data robustness against known data attack is evaluated in Section 7.4. We conclude this section with a real case example in Section 7.5. We use three real datasets which can be obtained from UCI Machine Learning Repository.

TABLE 7.1
Test Databases

Databases	iris	pendigits	satlog
Number of attributes	4	16	36
Number of records	150	7,494	4,435

7.1 FISIP Preservation

Let us see how the data changes by using the first two records in iris as an example. The first two private records are

$$\begin{bmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \end{bmatrix}$$

and their transformed records with no perturbation are

$$\begin{bmatrix} 0.00 & 1.60 & 3.70 & 4.90 \\ -0.15 & 1.75 & 3.35 & 4.55 \end{bmatrix}.$$

With perturbation $pert = -4$, they change to

$$\begin{bmatrix} 0.20 & 1.81 & 3.80 & 4.40 \\ -0.38 & 1.83 & 3.55 & 4.50 \end{bmatrix}.$$

The distances of above three pairs are 0.5385, 0.5385, and 0.6398. The correlation of above three pairs are 0.9960, 0.9960, and 0.9946. Note that the transformed records with perturbation and without perturbation looks similar, but it is not a big deal as long as they are not close to private counterparts. Let us first examine the three properties in FISIP. The first is first order sum, which is theoretically preserved and needs no experiment. The second and third properties are second order sum and inner products. For each property, we calculate the difference between private records and public records. Normalization is done by dividing the value of private records, i.e., difference of second order sums =

$$\frac{\left|\sum_{i=1}^{d} r_i^2 - \sum_{i=1}^{d} u_i^2\right|}{\sum_{i=1}^{d} r_i^2},$$

and difference of innper products =

$$\frac{|\mathbf{r}_p \cdot \mathbf{r}_q - \mathbf{u}_p \cdot \mathbf{u}_q|}{|\mathbf{r}_p \cdot \mathbf{r}_q|},$$

where $\mathbf{r} = [r_i]$ and $\mathbf{u} = [u_i]$, $1 \leq p, q \leq$ number of total records.

As Fig. 7.1-7.3 show, we can confirm that perturbation does not deteriorate our preservation of FISIP very much.

7.2 Neighborhood Preservation

Neighbors of vectors are important in many data mining algorithms, e.g. clustering, classification, outlier detection. Once the far-near relationships are maintained, we get the same mining result even

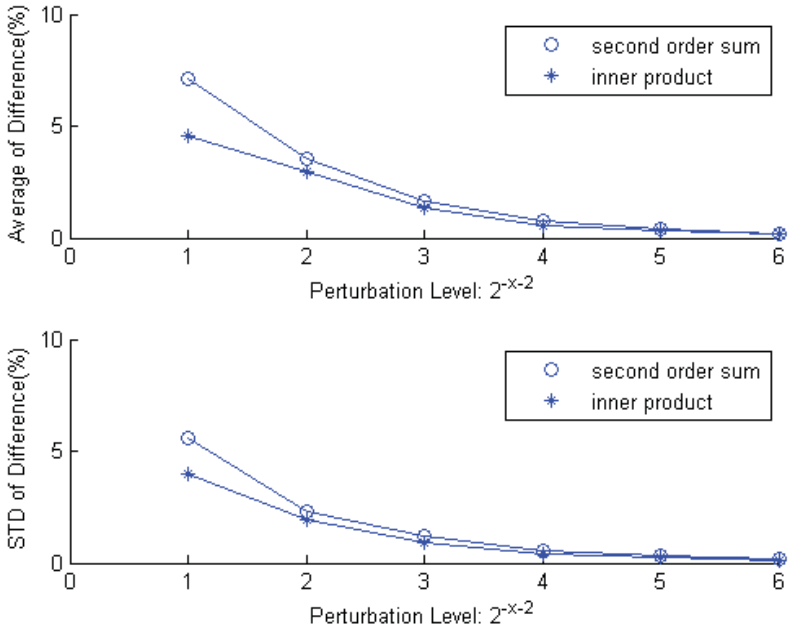

Fig. 7.1: Preservation of FISIP: iris

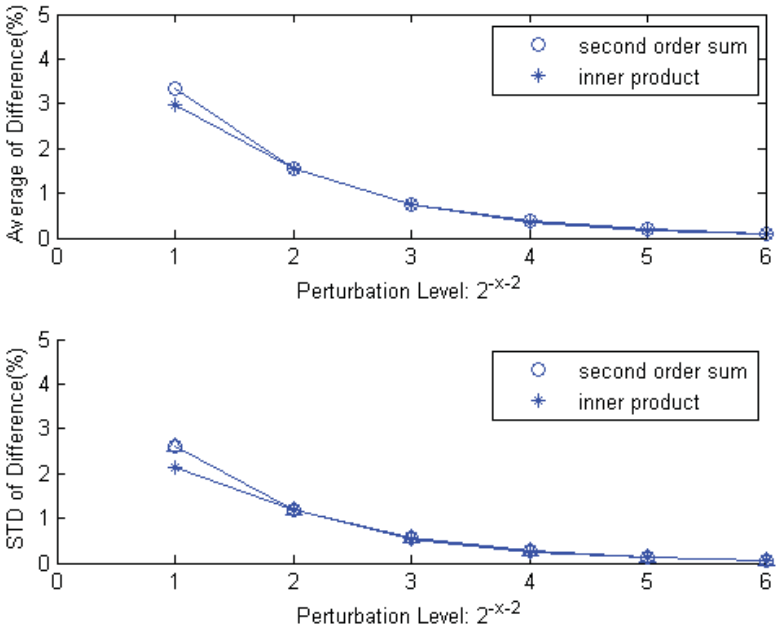

Fig. 7.2: Preservation of FISIP: pendigits

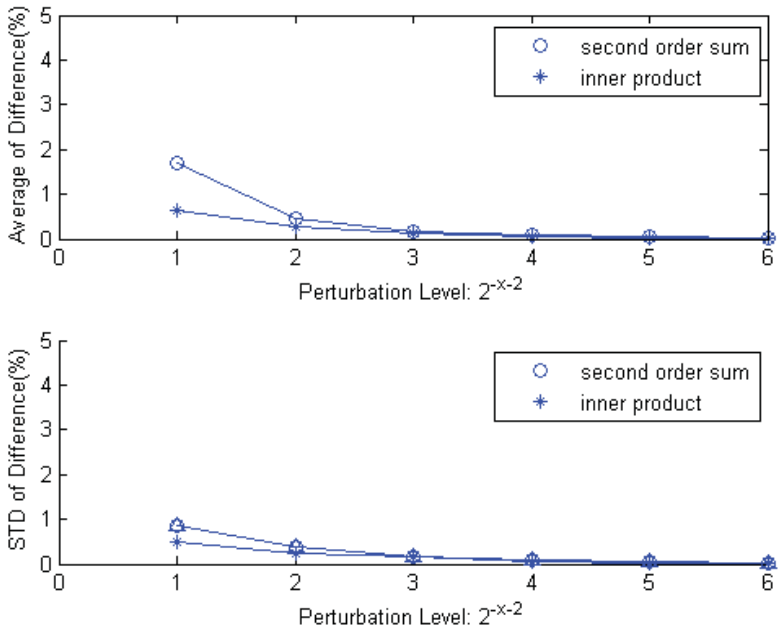

Fig. 7.3: Preservation of FISIP: satlog

though the data is changed and differently distributed. The first experiment we want to do is to check how well the neighbors of vectors can be kept the same. The first variant is the number of neighbors and the second variant is the level of perturbation $pert$ against known public data attack as stated above. We define the performance metric as numbers of intersections normalized by k:

$$\frac{1}{n}\sum_{j=1}^{n}\frac{|C_{r,j}\cap C_{u,j}|}{k},$$

where $C_{r,j}$represents the set of neighbors of private vector j and $C_{u,j}$ represents the set of neighbors of public record j. Smaller value of 2^{pert}, or smaller value of $pert$ means smaller perturbation and should have higher accuracies as results show. Accuracies here are evaluated by neighbor's change.

Consider a record A that has four nearest neighbors, record B, C, D, E. If the transformation makes A's four nearest neighbor to B, C, D, F, then the accuracy is 3/4. From Fig. 7.4-7.6, they confirm that when the number of nearest neighbors increases, the error margin in data space also increases and thus make the accuracies higher.

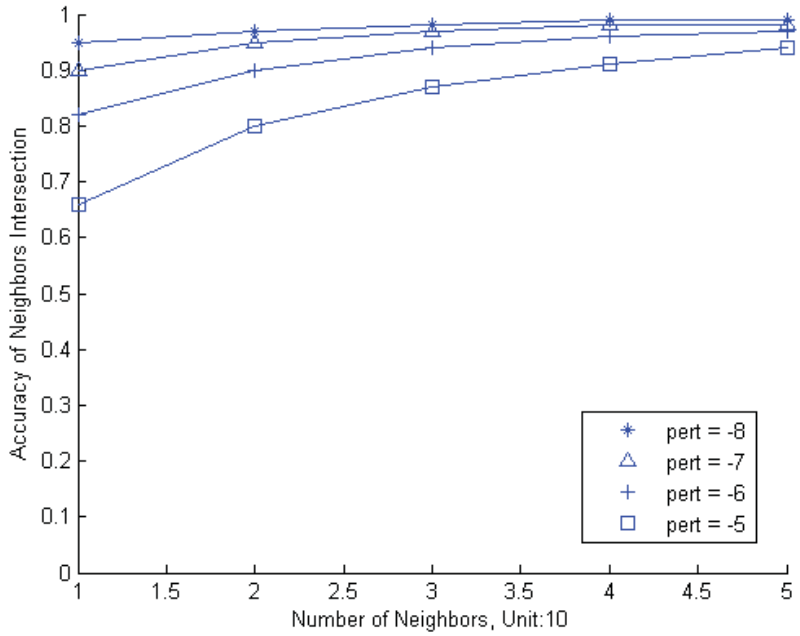

Fig. 7.4: Preservation of Neighbors: iris

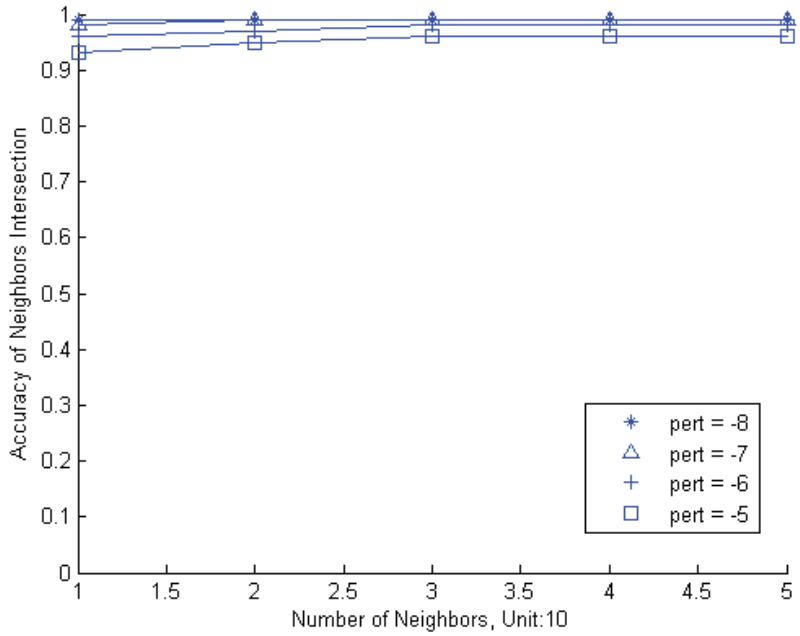

Fig. 7.5: Preservation of Neighbors: pendigits

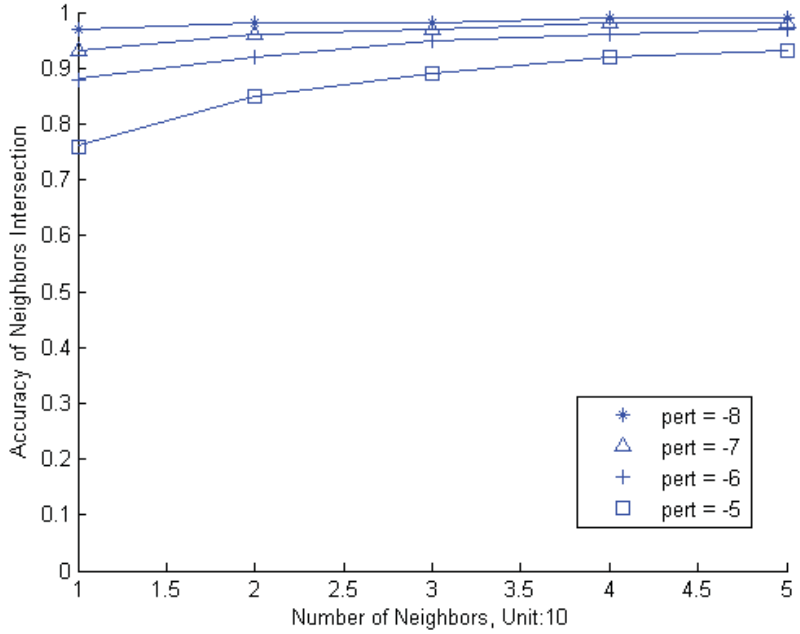

Fig. 7.6: Preservation of Neighbors: satlog

7.3 Correlation Preservation

We randomly select 10,000 pairs of vectors in database and measure the differences between private pairs and public pairs, i.e.,

$$|Correlation(\mathbf{r}_p, \mathbf{r}_q) - Correlation(\mathbf{u}_p, \mathbf{u}_q)|\,.$$

For pairs p, q in private record **r** and public record **u.** Average and standard deviation of this differences are plotted on Fig. 7. When the perturbation level decreases, the differences, i.e., errors, are also decreased as expected. Maximum difference is 2 and minimum is zero. From Fig. 7.7, we know that FISIP can maintain the relations of correlations very well under perturbation.

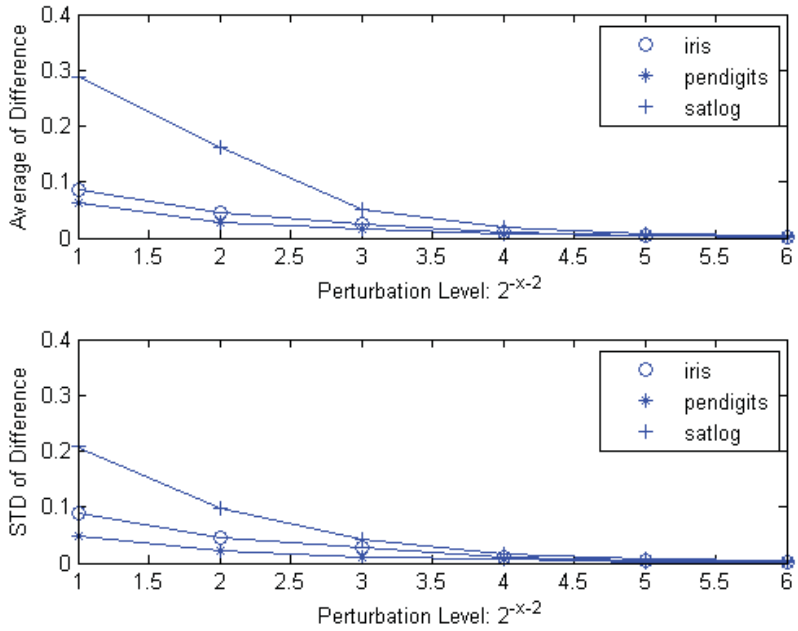

Fig. 7.7: Preservation of Correlation

7.4 Protection against Known Data Attack

The next experiment illustrates how well the protection scheme behaves if different perturbed transformation matrix are adopted. For k-dimensional database, we assume the attacker has k private-public pairs at hand and he uses the techniques we described in previous section to estimate the remaining unknown private records. As stated in previous section, the estimation error is $\left|\sum_{i=1}^{k} \alpha_i (I_k - A_x^{-1} A_i) r_i\right|$ and we can do normalization by asumming $\alpha_i = 1$ and $r_i = \left(\frac{1}{k}, \cdots, \frac{1}{k}\right)$. We take the norm of estimation error and plot the result as Fig. 8. The fact that orthogonal matrix can preserve distance is not a news. Commonly believed drawback is that all records are transformed in the same way, and thus, reverse transformation seems possible. However, in our work, we can transform each record in difference way (by using differently perturbed FISIP matrices). Therefore, our work has no such issue, as illustrated in Fig. 7.8.

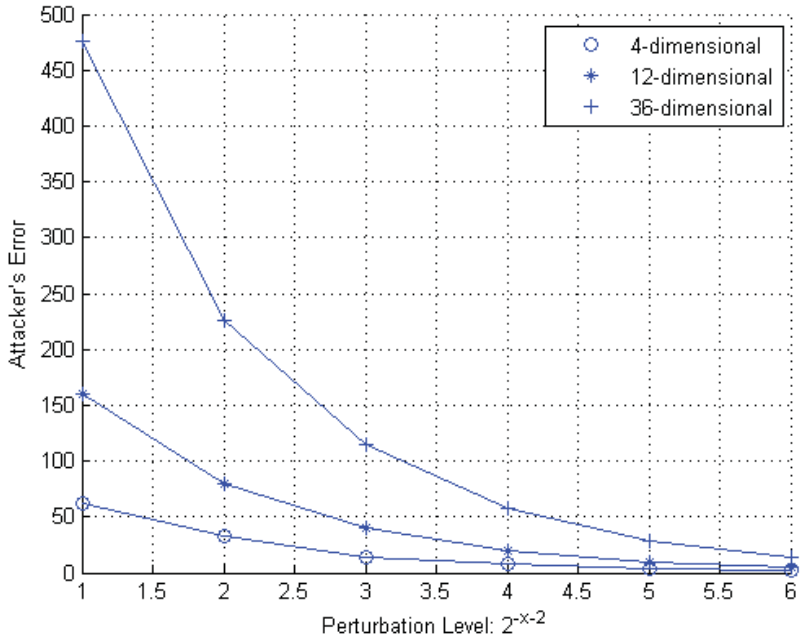

Fig. 7.8: Error of Attacker's Estimation

7.5 A Real Case Example: k-means

We use precision rate to measure the result of clustering. Let $C_1, C_2, \ldots, C_n$ be correct clusters generated by private vectors and $C'_1, C'_2, \ldots, C'_n$ be new clusters generated by public vectors. The clustering performance metrics are defined as follows:

$$precision = \frac{|C_i \cap C'_i|}{|C'_i|}$$

We use pendigits as our testing data and cluster it into 10 clusters by k means. Pendigits originally has 16 dimensions, and we transform it into 12 dimensional public vectors. Transforming without truncating dimensions yields exactly the same result but once we transform it into fewer dimensions, the result will be partially different from original clustering since some information is lost. Table 2 is the precision rate of the public database, horizontal axis denotes the ten clusters and vertical axis shows

the corresponding precision rate. Though the results are not equal to 1, they are very close to 1.

TABLE 7.2
Precision rate after dimension adaptation

cluster	1	2	3	4	5	6	7	8	9	10
precision	0.97	0.98	0.95	0.97	0.97	0.97	0.95	0.95	0.96	0.95

Chapter 8

Conclusion

Most privacy preservation schemes focus on specific algorithms but we target at more general applications. Given k-dimensional database, we propose a scheme that can transform it to published database without breaking the relation of distances and correlations between vectors. The mining quality from the transformed (public) data can be obtained to be the same as that from the original (private) data. Security of private data can be further enhanced at the expense of slight mining quality. We perturb the transformation matrix and make each record be transformed by different matrix. Though the protection slightly affects the mining quality, it greatly enhance the security of private data and make the inverse transformation of public data infeasible. The number of different transformation grows in a factorial way as the dimension of data increases, and thus makes attackers hard to recover them back to private counterparts. Possible attacks are studied, solved and shown via experimental results.

Bibliography

[1] E. Achtert, C. Bohm, H.-P. Kriegel, P. Kroger, and A. Zimek. Robust, complete, and efficient correlation clustering. In *SDM*. SIAM, 2007.

[2] C. Aggarwal, , C. C. Aggarwal, and P. S. Yu. A condensation approach to privacy preserving data mining. In *In EDBT*, pages 183–199, 2004.

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, 2000.

[4] F. Angiulli, S. Basta, and C. Pizzuti. Distance-based detection and prediction of outliers. *IEEE Trans. on Knowl. and Data Eng.*, 18(2):145–160, 2006.

[5] C. Bohm, K. Kailing, P. Kroger, and A. Zimek. Computing clusters of correlation connected objects. In *SIGMOD Conference*, pages 455–466. ACM, 2004.

[6] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 589–592, Washington, DC, USA, 2005. IEEE Computer Society.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996.

[8] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th International Conf. on Machine Learning*, pages 359–366. Morgan Kaufmann, San Francisco, CA, 2000.

[9] A. Hyvarinen. *Independent Component Analysis*. New York: Wiley, 2001.

[10] D. Kifer and J. Gehrke. l-diversity: Privacy beyond k-anonymity. In *In ICDE*, page 24, 2006.

[11] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *VLDB Journal: Very Large Data Bases*, 8(3–4):237–253, 2000.

[12] N. Li and T. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *In Proc. of IEEE 23rd Int'l Conf. on Data Engineering, ICDE*, 2007.

[13] J. T. li Wang, X. Wang, K. ip Lin, D. Shasha, B. A. Shapiro, and K. Zhang. Evaluating a class of distance-mapping algorithms for data mining and clustering. In *In Knowledge Discovery and Data Mining*, pages 307–311. ACM Press, 1999.

[14] D. Lin, E. Bertino, R. Cheng, and S. Prabhakar. Position transformation: a location privacy protection method for moving objects. In *SPRINGL '08: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 62–71, New York, NY, USA, 2008. ACM.

[15] K. Liu and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Trans. on Knowl. and Data Eng.*, 18(1):92–106, 2006. Senior Member-Kargupta, Hillol.

[16] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[17] S. Mukherjee, Z. Chen, and A. Gangopadhyay. A privacy-preserving technique for euclidean distance-based mining algorithms using fourier-related transforms. *The VLDB Journal*, 15(4):293–315, 2006.

[18] S. Ross. *First Course in Probability*. Prentice Hall, 2005.

[19] L. Sweeney. k-anonymity: A model for protecting privacy. In *In International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[20] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 1(33), 2004.