Chapter 15

A Survey of Attack Techniques on Privacy-Preserving Data Perturbation Methods

Kun Liu[1], Chris Giannella[2], and Hillol Kargupta[3]

[1] *IBM Almaden Research Center*
650 Harry Road, San Jose, CA 95120
kun@us.ibm.com

[2] *Department of Computer Science*
Loyola College in Maryland
4501 N. Charles Street, Baltimore, MD. 21210
cgiannel@acm.org

[3] *Department of Computer Science and Electrical Engineering*
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250

Also affiliated with AGNIK, LLC
8840 Stanford Blvd. Suite 1300, Columbia, MD 21045
hillol@cs.umbc.edu

Abstract We focus primarily on the use of additive and matrix multiplicative data perturbation techniques in privacy preserving data mining (PPDM). We survey a recent body of research aimed at better understanding the vulnerabilities of these techniques. These researchers assumed the role of an attacker and developed methods for estimating the original data from the perturbed data and any available prior knowledge. Finally, we briefly discuss research aimed at attacking k-anonymization, another data perturbation technique in PPDM.

Keywords: Data perturbation, additive noise, matrix multiplicative noise, attack techniques, k-anonymity.

15.1 Introduction

Data perturbation represents one common approach in privacy preserving data mining (PPDM). It builds on a longer history in the areas of statistical disclosure control and statistical databases [1] where the original (private) dataset is perturbed and the result is released for data analysis. Typically, a "privacy/accuracy" trade-off is faced. 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 mined. Data perturbation includes a wide variety of techniques including (but not limited to): additive, multiplicative [24], matrix multiplicative, k-anonymization [38, 41], micro-aggregation [3, 26], categorical data perturbation [10, 45], data swapping [11], resampling [27], data shuffling [34] (see [1, 28] for a more complete survey).

In this chapter we mostly focus on two types of data perturbation that apply to continuous data: additive and matrix multiplicative. Additive data perturbation was originally introduced in statistical disclosure control more that twenty years ago and was further studied in the PPDM community in the last eight years. Matrix multiplicative data perturbation were introduced only five years ago in the PPDM community and is in its early stages of study. In order to better understand the privacy offered by these techniques, some PPDM researchers have assumed the role of an attacker and developed techniques for breaching privacy by estimating the original data from the perturbed data and any available additional prior knowledge. Their work offers insight into vulnerabilities of this type of data perturbation. We provide a detailed survey of their work in an effort to allow the reader to observe common themes and future directions. Moreover, due to its rapidly growing study, we also provide a brief overview of attacks on k-anonymization.

This chapter is organized as follows. Section 15.2 describes definitions and notation used throughout. Section 15.3 discusses additive data perturbation, its uses and several attack techniques in detail. Section 15.4 describes matrix multiplicative data perturbation, its uses and several attack techniques in detail. Section 15.5 discusses k-anonymization and recent literature addressing vulnerabilities of this data perturbation model. Finally, Section 15.6 concludes the paper with a summary.

15.2 Definitions and Notation

Throughout this chapter, the original dataset is represented as an $n \times m$, real-valued matrix X, with each column a data record. The data owner perturbs X to produce an $n' \times m$ data matrix Y, which is then released to the public or another party for analysis. The attacker uses Y and any other available information to produce an estimation of X, denoted by $\hat{X}$. Unless otherwise stated, we will assume that each record of the original dataset arose as an independent

sample from an n-dimensional random vector $\mathcal{X}$ with unknown probability density function (*p.d.f.*) (and this assumption is public knowledge). Let $\Sigma_{\mathcal{X}}$ denote the covariance matrix of $\mathcal{X}$. We will also assume that $\Sigma_{\mathcal{X}}$ has all distinct and non-zero eigenvalues (more details later) since, as argued in [20, pg. 27], this assumption holds in most practical situations.

Unless otherwise stated, all vectors are column-vectors. Given a matrix A, A^T denotes its transpose and A^{-1} denotes its inverse (provided one exists). I denotes the identity matrix with dimensions specified by context. Given vector x, $||x||$ denotes the Euclidean distance of x to the origin *i.e.* the Euclidean norm.

15.3 Attacking Additive Data Perturbation

The data owner replaces the original dataset X with

$$Y \quad = \quad X + R, \tag{15.1}$$

where R is a noise matrix with each column generated independently from a n-dimensional random vector $\mathcal{R}$ with mean vector zero. As is commonly done, we assume throughout that $\Sigma_{\mathcal{R}}$ equals $\sigma^2 I$, *i.e.,* the entries of R were generated independently from some distribution with mean zero and variance σ^2 (typical choices for this distribution include Gaussian and uniform). In this case, R is sometimes referred to as *additive white noise*.

While having a long history in the statistical disclosure control and statistical database fields (see [6] for a comprehensive survey), additive data perturbation was first revisited to address PPDM problems by Agrawal and Srikant [5]. They assumed the *p.d.f.* of $\mathcal{R}$ is public. They developed a technique for estimating the *p.d.f.* of $\mathcal{X}$ from Y and show how a decision tree classifier can then be constructed. Their distribution recovery technique is further developed in [4, 9].

We describe five different attack techniques against additive perturbation. The first three attacks filter off the random noise by analyzing the eigenstates of the data: spectral filtering [22], singular value decomposition (SVD) filtering [17], and principal component analysis (PCA) filtering [18]. They all use *eigen-analysis* for filtering out the protected data. The fourth attack is a Bayes approach based on maximum a posteriori probability (MAP) estimation [18]. The fifth attack shows that if the *p.d.f.* of $\mathcal{X}$ is reconstructed, in some cases, it can lead to disclosure. We refer to this attack as *distribution analysis*. Note that in all five we assume that the attacker knows the *p.d.f.* of $\mathcal{R}$, and attacker implicitly knows that the perturbed data records arose as independent samples from random vector $\mathcal{Y} = \mathcal{X} + \mathcal{R}$. Next, we describe each of these attacks in detail.

15.3.1 Eigen-Analysis and PCA Preliminaries

Before describing eigen-analysis based attacks, we first provide a brief background of eigen-analysis and PCA. Let $\mathcal{X}$ be an n-dimensional random vector. Generally speaking the eigenvalues of covariance $\Sigma_{\mathcal{X}}$ are the n roots (possible including repeats) of the degree n polynomial $|\Sigma_{\mathcal{X}} - I\lambda|$ where $|.|$ denotes the matrix determinant. Since $\Sigma_{\mathcal{X}}$ is positive semi-definite, all its eigenvalues are non-negative and real [13, pg. 295]. If we assume that they are also all distinct and non-zero, they can be denoted as $\lambda_{\mathcal{X}}^1 > \ldots > \lambda_{\mathcal{X}}^n > 0$. Associated with $\lambda_{\mathcal{X}}^j$ is its *normalized eigenspace*, $\mathbb{V}_{\mathcal{X}}^j = \{v \in \mathbb{R}^n : \Sigma_{\mathcal{X}} v = v\lambda_{\mathcal{X}}^j \text{ and } ||v|| = 1\}$. These normalized eigenspaces are pair-wise orthogonal and have dimension one [13, pg. 295]. Hence each can be written as $\{v_{\mathcal{X}}^j, -v_{\mathcal{X}}^j\}$ where $v_{\mathcal{X}}^j$ is lexicographically larger than $-v_{\mathcal{X}}^j$. Let $V_{\mathcal{X}}$ denote the normalized eigenvector matrix $[v_{\mathcal{X}}^1 \cdots v_{\mathcal{X}}^n]$ (which is orthogonal).

As is standard practice in PCA, we assume that $\mathcal{X}$ has mean vector zero (if not, it is replaced by $\mathcal{X} - E[\mathcal{X}]$). The j^{th} *principal component (PC)* of $\mathcal{X}$ is ${v_{\mathcal{X}}^j}^T \mathcal{X}$ (or $-{v_{\mathcal{X}}^j}^T \mathcal{X}$). It can be shown that the PCs are pair-wise uncorrelated and capture the maximum possible variance in the following sense. For each $1 \leq j \leq n$, there does not exist $v \in \mathbb{R}^n$ orthogonal to v_ℓ for all $1 \leq \ell < j$ such that $Var(v^T\mathcal{X}) > Var({v_{\mathcal{X}}^j}^T \mathcal{X})$. It can further be shown that $Var({v_{\mathcal{X}}^j}^T \mathcal{X}) = \lambda_{\mathcal{X}}^j$. Therefore, the dimensionality of $\mathcal{X}$ can be reduced by choosing $1 \leq k \leq n$ and transforming $\mathcal{X}$ to $\tilde{\mathcal{X}} = \tilde{V}_{\mathcal{X}}^T \mathcal{X}$ where $\tilde{V}_{\mathcal{X}}$ denotes the leftmost k columns of $V_{\mathcal{X}}$. The amount of "information" preserved is typically quantified by

$$100 \frac{\sum_{\ell=1}^{k} \lambda_{\mathcal{X}}^{\ell}}{\sum_{\ell=1}^{n} \lambda_{\mathcal{X}}^{\ell}}.$$

This is commonly referred to as the percentage of variance captured by $\tilde{\mathcal{X}}$. If this percentage is large, most of the information is preserved in the sense that $\tilde{V}_{\mathcal{X}}\tilde{\mathcal{X}}$ is a good approximation to $\mathcal{X}$. Indeed, if the percentage is 100, *i.e.*, $k = n$, then $\tilde{V}_{\mathcal{X}}\tilde{\mathcal{X}} = \tilde{V}_{\mathcal{X}}\tilde{V}_{\mathcal{X}}^T\mathcal{X} = \mathcal{X}$. The properties of left multiplication to $\mathcal{X}$ by $\tilde{V}_{\mathcal{X}}\tilde{V}_{\mathcal{X}}^T$ have special significance in the eigen-analysis based attacks. We call this transformation, a *projection through* the first k PCs.

In practice, one has a collection of data tuples on which dimensionality reduction via PCA is desired. If the tuples can all be regarded as independent samples from $\mathcal{X}$, PCA can be fruitfully carried out on their standard sample covariance matrix (after subtracting from each the row-mean vector of the dataset). The eigen-analysis based attacks will make critical use of the projection of the dataset through its first k PCs.

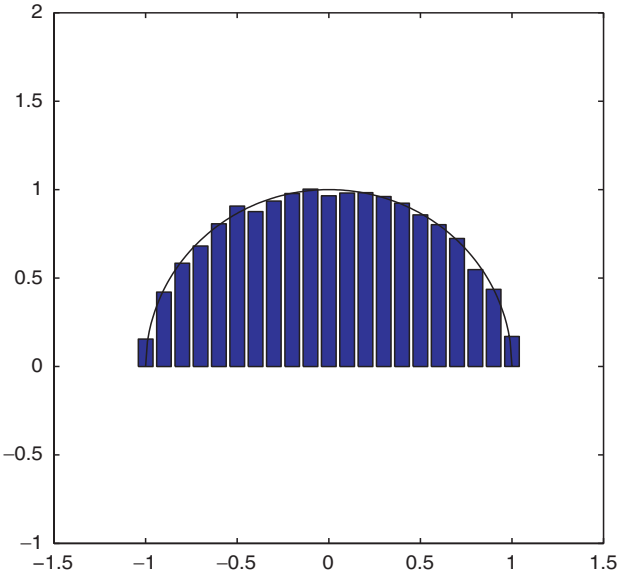

Figure 15.1. Wigner's semi-circle law: a histogram of the eigenvalues of $\frac{A+A'}{2\sqrt{2p}}$ for a large, randomly generated A

15.3.2 Spectral Filtering

This technique, developed by Kargupta *et al.* [22], utilizes the fact that the eigenvalues of a random matrix are distributed in a fairly predictable manner. For example, Wigner's semi-circle law [47] says that if A is a $p \times p$ matrix whose entries were generated independently from a distribution with zero mean and unit variance, then, for large p, the distribution of the eigenvalues of $\frac{A+A'}{2\sqrt{2p}}$ has *p.d.f.* depicted in Figure 15.1; it takes the shape of a semi-circle. As another example, consider $n \times m$ matrix R whose entries were generated independently from a distribution with mean zero and variance σ^2. For large m and n, the distribution of the eigenvalues of the sample covariance matrix of R is similar to the semi-circle law. And, key to the spectral filtering technique, this result allows bounds on these eigenvalues to be computed.

Kargupta *et al.* observe that if the j^{th} eigenvalue arising from Y is "large", it is a good approximation to the j^{th} eigenvalue arising from X. Therefore, the projection of Y through its PCs corresponding to these large eigenvalues (say the first k) is a good approximation to the projection of X through its first k PCs. As such $\hat{X}$ is set to the projection of Y through its first k PCs. Results from matrix perturbation theory and spectral analysis of large random matrices provide the basis for this observation.

LEMMA 15.1 *[40, Corollary 4.9] For any n-dimensional random vectors $\mathcal{X}$ and $\mathcal{R}$ ($\mathcal{R}$ has mean vector zero) and $\mathcal{Y} = \mathcal{X} + \mathcal{R}$, it is the case that: for $1 \leq j \leq n$, $\lambda_{\mathcal{Y}}^j \in [\lambda_{\mathcal{X}}^j + \lambda_{\mathcal{R}}^n, \lambda_{\mathcal{X}}^j + \lambda_{\mathcal{R}}^1]$.*

Therefore, if $\lambda_{\mathcal{Y}}^j \in [\lambda_{\mathcal{R}}^n, \lambda_{\mathcal{R}}^1]$, then this eigenvalue is largely affected by noise ($\mathcal{R}$). Hence, it is not regarded by Kargupta *et al.* as large and, therefore, not regarded as a good approximation of $\lambda_{\mathcal{X}}^j$. On the other hand, $\lambda_{\mathcal{Y}}^j > \lambda_{\mathcal{R}}^1$ is regarded as large and, therefore, is regarded as a good approximation of $\lambda_{\mathcal{X}}^j$. So how can the attacker use this threshold criterion given only Y?

Let $\hat{\Sigma}_Y$ and $\hat{\Sigma}_R$ be the standard sample covariance matrices computed from Y and R; let $\hat{\lambda}_Y^1 \geq \ldots \geq \hat{\lambda}_Y^n$ and $\hat{\lambda}_R^1 \geq \ldots \geq \hat{\lambda}_R^n$ be the associated eigenvalues, respectively. The above criterion can be modified to consider $\hat{\lambda}_Y^j > \hat{\lambda}_R^1$ as large. But how should the attacker estimate an upper-bound on $\hat{\lambda}_R^1$? This question is answered using a result from large random matrix theory alluded to in the opening paragraph of this subsection. Intuitively, as R grows large, the eigenvalues computed from R can be bounded by the attacker. And when m is large relative to n, these bounds are quite good. Formally stated [21, 39], as $m, n \to \infty$ and $\frac{m}{n} \to Q \geq 1$,

$$\hat{\lambda}_R^{max} = \sigma^2(1 + 1/\sqrt{Q})^2 \geq \hat{\lambda}_R^1 \geq \hat{\lambda}_R^n \geq \hat{\lambda}_R^{min} = \sigma^2(1 - 1/\sqrt{Q})^2.$$

As such, $\hat{\lambda}_R^{max}$ serves as the estimate of an upper-bound on $\hat{\lambda}_R^1$. Moreover, for Q large relative to σ^2, this bound will be quite good as all eigenvalues of $\hat{\Sigma}_R$ will be concentrated in a small band. Since the attacker is assumed to know σ^2, then she can compute $\hat{\lambda}_R^{max}$ and will deem any $\hat{\lambda}_Y^j > \hat{\lambda}_R^{max}$ as large.

The spectral filtering algorithm is given in Algorithm 3. The empirical results show that when the variance of the noise is low and the original data does not contain many inherent random components, the recovered data can be reasonably close to the original data. However, two important questions remain to be answered. 1) What are the theoretical bounds on the estimation accuracy? 2) What are the fundamental factors that determine the quality of the data estimation? The first is touched on in Section 15.3.3 and the second in Section 15.3.4.

15.3.3 SVD Filtering

Guo *et al.* [17] revisited spectral filtering to address the issue of an optimal choice of k and to develop bounds on the estimation accuracy. They showed that when $k = \min\{1 \leq j \leq n | \hat{\lambda}_Y^j < 2\sigma^2\} - 1$, the estimated data is approximately optimal, *i.e.*, the benefits due to the inclusion of the k^{th} eigenvector is greater than the information loss due to the noise projected along the k^{th} eigenvector. They further proposed a singular value decomposition-based data reconstruction approach, and proved the equivalence of this approach to spectral filtering. A lower bound and upper bound of the estimation error in terms

Protocol 3 Spectral Filtering

Require: Y, the perturbed data matrix and σ^2, the variance of the random noise.

Ensure: $\hat{X}$, an estimate of the original data matrix X.

1: Compute the sample mean of Y and subtract it from every column of Y.
2: Compute the standard sample covariance $\hat{\Sigma}_Y$ of Y, its eigenvalues $\hat{\lambda}_Y^1 \geq \ldots \geq \hat{\lambda}_Y^n$, and their associated normalized eigenvectors $\hat{v}_Y^1, \ldots, \hat{v}_Y^n$.
3: Compute $k = \max\{1 \leq j \leq n | \hat{\lambda}_Y^j > \hat{\lambda}_R^{max}\}$. Let $\tilde{\hat{V}}_Y$ denote the matrix $[\hat{v}_Y^1 \cdots \hat{v}_Y^k]$.
4: Set $\hat{X}$ to $\tilde{\hat{V}}_Y \tilde{\hat{V}}_Y^T Y$.

of Frobenius matrix norm were also derived. We refer readers to [14, 17] for more details.

15.3.4 PCA Filtering

Huang *et al.* [18] observe that a key factor in determining the accuracy of spectral filtering is the degree of correlation that exists among the attributes of $\mathcal{X}$ relative to σ^2. The higher the degree, the greater the accuracy in estimating the original data. Indeed, for small k, the higher the degree of correlation, the more variance will be captured by the first k PCs. The addition of $\mathcal{R}$ does not change this property. The attributes of $\mathcal{R}$ are uncorrelated and thus, the amount of variance captured by *any* direction is the same. Therefore, removing the last $n - k$ PCs of $\mathcal{X}$ does not cause much variance loss but will cause $100\frac{n-k}{n}$ percent of the variance in $\mathcal{R}$ to be lost.

Based on this observation, Huang *et al.* [18] proposed a filtering technique based on PCA. A major difference with spectral filtering, is that PCA filtering does not use matrix perturbation theory and spectral analysis to estimate the dominant PCs of X. Instead PCA filtering takes a more direct approach based on the fact that

$$\Sigma_{\mathcal{Y}} = \Sigma_{\mathcal{X}} + \Sigma_{\mathcal{R}} = \Sigma_{\mathcal{X}} + \sigma^2 I. \tag{15.2}$$

The first equality is due to the independence of $\mathcal{X}$ and $\mathcal{R}$ and the second by assumption. Therefore, the attacker can directly estimate $\Sigma_{\mathcal{X}}$ as $\hat{\Sigma}_Y - \sigma^2 I$, then compute the top k PCs of this. The PCA filtering procedure is given in Algorithm 4.

The original dataset estimate can be written as the sum of two parts: $\hat{X} = \tilde{\hat{V}}_X \tilde{\hat{V}}_X^T Y = \tilde{\hat{V}}_X \tilde{\hat{V}}_X^T X + \tilde{\hat{V}}_X \tilde{\hat{V}}_X^T R$. Therefore, the recovery error [1] is determined

[1] assuming the estimated sample covariance $\hat{\Sigma}_X$ is very close to $\Sigma_{\mathcal{X}}$

Protocol 4 PCA Filtering

Require: Y, the perturbed data matrix; σ^2, the variance of the random noise; and $1 \leq k \leq n$, the number of PCs to keep.

Ensure: $\hat{X}$, an estimate of the original data matrix X.

1: Compute the sample mean of Y and subtract it from every column of Y.
2: Compute the standard sample covariance $\hat{\Sigma}_Y$ of Y, and produce $\hat{\Sigma}_X = \hat{\Sigma}_Y - \sigma^2 I$ an estimate of $\Sigma_{\mathcal{X}}$.
3: Compute the eigenvalues of $\hat{\Sigma}_X$, $\hat{\lambda}_X^1 \geq \ldots \geq \hat{\lambda}_X^n$. Compute their their associated normalized eigenvectors, $\hat{v}_X^1, \ldots, \hat{v}_X^n$. Let $\tilde{\hat{V}}_X$ denote the matrix $[\hat{v}_X^1 \cdots \hat{v}_X^k]$.
4: Set $\hat{X}$ to $\hat{V}_X \tilde{\hat{V}}_X^T Y$.

by the the percentage of variance captured by the first k PCs of $\mathcal{X}$ and the noise. It can be shown that the mean squared recovery error caused by the noise part is $\sigma^2 \frac{k}{n}$. These results echo the empirical results observed in spectral filtering and suggests an approach for choosing k.

15.3.5 MAP Estimation Attack

Different from eigen-analysis, MAP estimation considers both prior and posterior knowledge via Bayes' theorem to estimate original dataset. For each $1 \leq i \leq m$, the attacker will produce $\hat{x}_i$ an estimate of x_i using[2] y_i. Let $f_{\mathcal{X}}$ and $f_{\mathcal{R}}$ denote the *p.d.f* of $\mathcal{X}$ and $\mathcal{R}$, respectively. Given $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^{n'}$, let $f_{\mathcal{X}|\mathcal{Y}=y}$ and $f_{\mathcal{Y}|\mathcal{X}=x}$ denote the *p.d.f* of $\mathcal{X}$ conditioned on $\mathcal{Y} = y$ and the *p.d.f* of $\mathcal{Y}$ conditioned on $\mathcal{X} = x$, respectively. The MAP estimate of x_i is[3]

$$\begin{aligned} \hat{x}_i &= argsup\{f_{\mathcal{X}|\mathcal{Y}=y_i}(x) : x \in \mathbb{R}^n\} \\ &= argsup\{f_{\mathcal{Y}|\mathcal{X}=x}(y_i) f_{\mathcal{X}}(x) : x \in \mathbb{R}^n\} \\ &= argsup\{f_{\mathcal{R}}(y_i - x) f_{\mathcal{X}}(x) : x \in \mathbb{R}^n\}. \end{aligned} \tag{15.3}$$

The second equality is due to Bayes' theorem and the third due to the fact that $\mathcal{Y} = \mathcal{X} + \mathcal{R}$ and $\mathcal{R}$ is independent of $\mathcal{X}$.

Huang *et al.* [18] considered the case where both $f_{\mathcal{X}}$ and $f_{\mathcal{R}}$ are multivariate normal (and the attacker knows this). The following closed form expression can then be derived with $\mu_{\mathcal{X}}$ denoting the mean vector of $\mathcal{X}$.

$$\hat{x}_i = (\Sigma_{\mathcal{X}}^{-1} + (1/\sigma^2)I)^{-1}(\Sigma_{\mathcal{X}}^{-1}\mu_{\mathcal{X}} + y_i/\sigma^2).$$

[2] Due to independence, the attacker will gain nothing more if using all of Y.

[3] Here $argsup\{\}$ is based on $supA$ which denotes the smallest upper bound on a set A (if A is upper-bounded, $supA$ always exists.

The assumption that $f_{\mathcal{X}}$ is multi-variate normal and known to the attacker is quite strong. Other cases are worth comment (in each, $f_{\mathcal{R}}$ is multi-variate normal and known to the attacker). When $f_{\mathcal{X}}$ is known but not multivariate normal, it may be difficult to derive a closed-form expression for $\hat{x}_i$. In this case, the attacker can use numerical methods such as Newton's gradient descent methods. When $f_{\mathcal{X}}$ is not known, the MAP estimate reduces to the maximum likelihood estimate (MLE) by assuming $f_{\mathcal{X}}$ is uniform over some interval. Therefore, $f_{\mathcal{X}}$ can be dropped from (15.3) and $\hat{x}_i = y_i$. However, this estimate may suffer from accuracy problems due to dropping $f_{\mathcal{X}}$.

It is worth noting that the MAP approach has been widely studied in statistical disclosure control. For example, Trottini *et al.* [44] used this approach to study the linkage privacy breaches in the scenario where microdata is masked by both additive and multiplicative noise. In their settings, the attacker tries to identify the identity (of a person) linked to a specific record, which is different from the primary focus of this chapter - data record recovery.

15.3.6 Distribution Analysis Attack

Recall that techniques exist for estimating $f_{\mathcal{X}}$ from Y. This is quite useful as $f_{\mathcal{X}}$ represents a useful data mining pattern. However, in some cases, this reconstructed distribution can be used by the attacker to gain extra knowledge about the private data. For example, assume the each entry of $\mathcal{R}$ is uniformly distributed over $[-1, 1]$ and the observed perturbed data $\mathcal{Y} = 1$. If there is no additional information, the attacker can determine $\mathcal{X} \in [0, 2]$. However, if a large amount of data is available, the reconstructed distribution will have a high degree of accuracy. Assume the attacker can perfectly recover $f_{\mathcal{X}}$ which is:

$$f_{\mathcal{X}}(x) = \begin{cases} 0.5, & 0 \leq x \leq 1; \\ 0.5, & 5 \leq x \leq 6; \\ 0, & \text{otherwise.} \end{cases}$$

Then, the estimate of $\mathcal{X}$ given $\mathcal{Y} = 1$ is localized to a smaller interval $[0, 1]$ instead of $[0, 2]$. When data has a multi-variate distribution, the attacker can determine intervals $I_1, I_2, \ldots, I_n$, which are narrow in one or more dimensions, and for which the number of data records that fall in the interval is very small. Such intervals make outliers/minorities more identifiable than they would seem when merely looking at the perturbed data set. This kind of disclosure leads to a bigger open problem - *when do data mining results cause privacy breach?* Further discussions can be found in [4, 9, 31, 16, 12].

15.3.7 Summary

This section surveyed recent research that investigated the vulnerability additive data perturbation. The research showed, in many cases, the private

information can be reasonably well derived from the perturbed data. The primary attack techniques presented are summarized in Table 15.1.

Table 15.1. Summarization of Attacks on Additive Perturbation

Categories	*Related Work*	*General Assumptions*
Eigen-Analysis	[14, 17, 18, 22]	the degree of correlation between the original data attributes is high relative to σ^2
MAP Estimation	[18]	data and noise arose from a multi-variate normal distribution
Distribution Analysis	[4, 9, 16]	reconstructed distribution describes the original data with sufficient accuracy

One possible improvement on additive perturbation is to use colored noise with similar correlation structure to the original data [23, 43], *i.e.*, $\mathcal{R} \sim (0, \Sigma_{\mathcal{R}})$, where $\Sigma_{\mathcal{R}} = \beta\Sigma_{\mathcal{X}}$ for $\beta > 0$. With this method, the covariance of the perturbed data is

$$\Sigma_{\mathcal{Y}} = \Sigma_{\mathcal{X}} + \beta\Sigma_{\mathcal{X}} = (1+\beta)\Sigma_{\mathcal{X}}.$$

The correlation coefficients of the perturbed attributes are the same as that of the original attributes:

$$\rho_{\mathcal{Y}_i,\mathcal{Y}_j} = \frac{1+\beta}{1+\beta}\frac{Cov(\mathcal{X}_i,\mathcal{X}_j)}{\sqrt{Var(\mathcal{X}_i)Var(\mathcal{X}_j)}} = \rho_{\mathcal{X}_i,\mathcal{X}_j}.$$

This kind of perturbation puts noise on the principal components of the original data, therefore, separating noise from the data using eigen analysis becomes difficult. However, this approach is not free from problem either. Domingo-Ferrer *et al.* [9] pointed out that the reconstructed distribution (using their p-dimensional reconstruction algorithm, a multivariate generalization of the approach describe in [5] for the univariate case) may still lead to disclosure in some cases. The higher the dimensionality, the more likely is the disclosure.

In summary, additive perturbation has its roots in statistical disclosure control. It offers a simply way to mask private data while allowing aggregate statistics to be queried; and making more sophisticated privacy preserving data mining possible. However, recent work from PPDM community has shown this technique vulnerable to attack in many cases (*e.g.,* high correlations between many attributes). Therefore, careful attention must be paid when applying this technique in practice.

Before closing this section, we note that several researchers have proposed privacy metrics *e.g.,* interval-based [5], entropy-based [4], mixture models [49]. However, the relationship between these and the recovery accuracy of the attack techniques is not clear.

15.4 Attacking Matrix Multiplicative Data Perturbation

The data owner replaces the original data X with

$$Y = MX, \tag{15.4}$$

where M is an $n' \times n$ matrix chosen to have certain useful properties. If M is orthogonal ($n' = n$ and $M^T M = I$) [7, 36, 37], then the perturbation exactly preserves Euclidean distances, *i.e.*, for any columns x_1, x_2 in X, their corresponding columns y_1, y_2 in Y satisfy $||x_1 - x_2|| = ||y_1 - y_2||$.[4] If each entry of M is generated independently from the same distribution with mean zero and variance σ^2 (n' not necessarily equal to n) [28, 30], then the perturbation approximately preserves Euclidean distances on expectation up to constant factor $\sigma^2 n'$. If M is the product of a discrete cosine transformation matrix and a truncated perturbation matrix [33], then the perturbation approximately preserves Euclidean distances.

Because matrix multiplicative perturbation preserves Euclidean distance with either small or no error, it allows many important data mining algorithms to be applied to the perturbed data and produce results very similar to, or exactly the same as those produced by the original algorithm applied to the original data, *e.g.,* hierarchical clustering, k-means clustering. However, the issue of how well X is hidden is not clear and deserves careful study. Without any prior knowledge, an attacker can do very little (if anything) to accurately recover X. However, no prior knowledge seems an unreasonable assumption in many situations. Motivated by this line of reasoning, several researchers have investigated the vulnerabilities of matrix multiplicative perturbation using various forms of prior knowledge [8, 15, 28–30]. In the bulk of this section (15.4.1 and 15.4.2), we discuss attack techniques based on two types of prior knowledge.

1 **Known input-output (I/O):** The attacker knows some small collection of original data records and the attacker knows the mapping between these known original data records and their perturbed counterparts in Y. In other words, the attacker has a set of input-output pairs.

2 **Known sample:** The attacker has a collection of independent samples (columns of S) from $\mathcal{X}$ (S may or may not overlap with X).

The first two attacks are based on the known I/O prior knowledge assumption. The first one [29] assumes an orthogonal perturbation matrix while the

[4] Conversely, any function $T : \mathbb{R}^n \to \mathbb{R}^n$ which preserves Euclidean distance (for all $x, y \in \mathbb{R}^n$, $||x - y|| = ||T(x) - T(y)||$) and fixes the origin is equivalent to left-multiplication by an $n \times n$ orthogonal matrix.

second [28] assumes a randomly generated perturbation matrix. The third attack is based on the known sample prior knowledge assumption and assumes an orthogonal perturbation matrix. It works by examining certain features of the original and perturbed data distributions (*i.e.*, the *p.d.f.* of $\mathcal{X}$ and $\mathcal{Y}$), namely the eigenvectors of $\Sigma_{\mathcal{X}}$ and $\Sigma_{\mathcal{Y}}$. These features have two important properties: (i) they are related to each other in a natural way allowing M to be estimated, and (ii) they can be accurately extracted from S and Y.

Before moving on, we emphasize the fact that the perturbation technique considered here, matrix multiplicative, is completely different than multiplicative data perturbation mentioned in the introduction. There each element of X is *separately multiplied* by a randomly generated number.

15.4.1 Known I/O Attacks

Without loss of generality, the attacker is assumed to know X_p ($1 \leq p < m$), the first p columns of X (of course, the attacker also knows Y_p, the first p columns of Y). In other words, the attacker knows a set of input/output pairs $(x_1, y_1), \ldots, (x_p, y_p)$ where $y_j = Mx_j$.

Orthogonal Perturbation Matrix. Liu *et al.* [29] assumed M is orthogonal. Unlike all other attacks in this chapter, they *do not assume* that the original data records arose as independent samples from $\mathcal{X}$. Their attacker uses Y_p and X_p to produce, $\hat{M}$, an estimation of M. Then, for any $p \leq i \leq m$, the attacker will produce $\hat{x}_i$, an estimation of x_i as

$$\hat{x}_i = \hat{M}^T y_i. \tag{15.5}$$

The rationale for (15.5) is: if $\hat{M} \approx M$, then $\hat{x}_i \approx M^T y_i = M^T(Mx_i) = x_i$. In choosing $\hat{M}$, the attacker knows that M must be in $\mathbb{M}(X_p, Y_p)$, the set of all $n \times n$, orthogonal matrices, O, such that $OX_p = Y_p$. However, with no additional information for further narrowing down this space of the possibilities, the attacker will assume each is equally likely to be M. Therefore, she will choose $\hat{M}$ uniformly from $\mathbb{M}(X_p, Y_p)$.

Given an error tolerance $\epsilon > 0$, the attacker's success probability, $\rho(x_i, \epsilon)$, is defined as the probability that the relative Euclidean distance between x_i and $\hat{x}_i$ is no larger than ϵ, *i.e.*, $Pr(||\hat{x}_i - x_i|| \leq ||x_i||\epsilon)$. Liu *et al.* developed closed form expression

$$\rho(x_i, \epsilon) = \begin{cases} \left(\frac{1}{\pi}\right) 2arcsin\left(\frac{||x_i||\epsilon}{2d(x_i, X_p)}\right) & \text{if } ||x_i||\epsilon < 2d(x_i, X_p); \\ 1 & \text{otherwise,} \end{cases} \tag{15.6}$$

where $d(x_i, X_p)$ denotes the Euclidean distance of x_i to the space of vectors spanned by the columns of X_p, *i.e.*, $inf\{||x - x_i||{:}x$ is in the column space

of X_p}. Equation (15.6) illustrates that the sensitivity of a tuple, x_i, to breach depends upon its length relative to its distance to the column space of X_p, *i.e.*, $\frac{||x_i||}{2d(x_i,X_p)}$. Tuples whose relative length is large are particularly sensitive to breach. In particular when x_i is in the column space of X_p, the attacker's success probability equals one. Liu *et al.* also described how the attacker can compute $||x_i||$ and $d(x_i, X_p)$ for any $p \leq i \leq m$, and therefore, determine which tuple is most sensitive to breach.

Chen *et al.* [8] also discussed a known I/O attack technique. They however consider a combination of matrix multiplicative and additive perturbation: $Y = MX + R$. They considered the case when the number of linearly independent data tuples (columns in X_p) is no smaller than the data dimensionality, n (rows in X_p). They pointed out that $\hat{M}$, an estimate of M, can be produced using linear regression, then x_i estimated as $\hat{M}^{-1}y_i$.

Random Perturbation Matrix. Liu [28] developed a MAP-based known I/O attack which works under the assumption that M is an $n' \times n$ matrix whose entries were generated independently from a normal distribution with mean zero and variance σ^2 (n' may be $\leq n$ or $> n$).[5] The larger n' is, the more closely preserved are Euclidean distances between data tuples (up to constant factor $\sigma^2 n'$), but, the better the known I/O attack will work at breaching privacy. Therefore, a trade-off must be balanced in setting n'.

For simplicity, we assume that the columns of Y_p are linearly independent.[6] For any $p \leq i \leq m$, the attacker will produce $\hat{x}_i$ an estimate of x_i. If x_i is linearly dependent on the columns of X_p, the attacker can discover this as y_i will be linearly dependent on the columns of Y_p. In this case, the attacker will set $\hat{x}_i = X_p(Y_p^T Y_p)^{-1} Y_p^T y_i$ which equals x_i (perfect recovery).[7] Henceforth, we assume x_i is linearly independent of the columns of X_p. Therefore, the attacker will only consider estimates, $\hat{x} \in \mathbb{R}^n$, which are also linearly independent of the columns of X_p (for brevity, we write "l.i. $\hat{x}$" to mean that $\hat{x}$ is linearly independent of the columns of X_p). Finally, since the columns of Y_p are assumed to be linearly independent, then it follows that the columns of X_p are too.

Let $\mathcal{M}$ be an $n' \times n$ matrix of random variables each independently and identically distributed as normal with mean zero and variance σ^2. The columns of Y arose as independent samples from random vector $\mathcal{Y} = \mathcal{M}\mathcal{X}$. Using the

[5] They do assume that the original data records arose as independent samples from $\mathcal{X}$.

[6] This assumption is not essential. It can be eliminated at the cost of a more complicated attack algorithm. However, the fundamental idea remains the same.

[7] There exists $z_i \in \mathbb{R}^p$ such that $X_p z_i = x_i$ and $Y_p z_i = y_i$. Since the columns of Y_p are assumed to be linearly independent, then by [13, pg. 96], the matrix $(Y_p^T Y_p)^{-1} Y_p^T$ exists. Thus, $X_p(Y_p^T Y_p)^{-1} Y_p^T y_i = X_p(Y_p^T Y_p)^{-1}(Y_p^T Y_p) z_i = X_p z_i = x_i$.

MAP approach, the attacker will choose l.i. $\hat{x}$ so as to maximize the likelihood that $\mathcal{X}$ equals $\hat{x}$ given that $\mathcal{Y}$ equals y_i and $\mathcal{M}X_p$ equals Y_p. This analysis is based on the following key observation (whose proof follows directly from manipulating moment-generating functions). For any matrix B, let $\overline{B}$ denote the column vector which results from stacking the columns of B.

THEOREM 15.2 *For any $n \times q$ matrix A with linearly independent columns, $\overline{\mathcal{M}A}$ is distributed as an (qn')-variate Gaussian with mean vector zero and covariance matrix*

$$\Sigma_{\overline{\mathcal{M}A}} = \sigma^2 \begin{bmatrix} A^TA & 0 & 0 & \cdots & 0 \\ 0 & A^TA & 0 & \cdots & 0 \\ 0 & 0 & A^TA & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A^TA \end{bmatrix}$$

Let $[X_p, \hat{x}]$ and $[Y_p, y_i]$ denote matrices which result from attaching $\hat{x}$ and y_i as an additional right-most column onto X_p and Y_p. Observe that $[X_p, \hat{x}]$ has linearly independent columns. Let $f_{\mathcal{X}|\mathcal{Y}=y_i, \overline{\mathcal{M}X_p}=\overline{Y_p}}$ denote the *p.d.f.* of $\mathcal{X}$ conditioned on $\mathcal{Y} = y_i$ and $\overline{\mathcal{M}X_p} = \overline{Y_p}$; let $f_{\overline{\mathcal{M}[X_p,\hat{x}]}}$ denote the *p.d.f.* of $\overline{\mathcal{M}[X_p, \hat{x}]}$. Using the MAP approach, the attacker will choose

$$\hat{x}_i = argsup\{f_{\mathcal{X}|\mathcal{Y}=y_i, \overline{\mathcal{M}X_p}=\overline{Y_p}}(\hat{x}) : \text{ l.i. } \hat{x} \in \mathbb{R}^n\}.$$

Using Bayes' rule, it can be shown that

$$\hat{x}_i = argsup\{f_{\overline{\mathcal{M}[X_p,\hat{x}]}}(\overline{[Y_p, y_i]}) f_{\mathcal{X}}(\hat{x}) : \text{ l.i. } \hat{x} \in \mathbb{R}^n\},$$

thus, Theorem 15.2 implies

$$\hat{x}_i = argsup\{\phi(\overline{[Y_p, y_i]}) f_{\mathcal{X}}(\hat{x}) : \text{ l.i. } \hat{x} \in \mathbb{R}^n\}, \tag{15.7}$$

where ϕ is the $((p+1)n')$-variate Gaussian distribution with mean vector zero and covariance matrix $\Sigma_{\overline{\mathcal{M}[X_p,\hat{x}]}}$. For simplicity we assume that the attacker knows nothing about $f_{\mathcal{X}}$ and, following a common practice, uses a uniform distribution over some interval in place of $f_{\mathcal{X}}$ in (15.7).[8] Thus,

$$\hat{x}_i = argsup\{\phi(\overline{[Y_p, y_i]}) : \text{ l.i. } \hat{x} \in \mathbb{R}^n\}. \tag{15.8}$$

Producing a closed-form expression for $\hat{x}_i$ in (15.8) is desirable, but quite difficult. Instead, the attacker can turn to numerical approaches. Experiments

[8] A more complicated approach could have the attacker using the fact that the columns of X_p arose as independent samples from $\mathcal{X}$, and use X_p to inform a better substitution for $f_{\mathcal{X}}$ in (15.7).

were reported in [28] where the attacker used the Matlab implementation[9] of the Nelder-Mead simplex algorithm [35] to solve this optimization problem. The results show that the accuracy of the attack technique increases with n' or the number of known input-output pairs.

15.4.2 Known Sample Attack

The attacker is assumed to know a collection of independent samples (columns of S) from $\mathcal{X}$ (S may or may not overlap with X). Furthermore, the attacker assumes M is orthogonal.

The approach is based on the observation that the eigenvectors of $\mathcal{Y}$ are equal to those of $\mathcal{X}$ *left-multiplied by* M (up to a factor of ± 1). Therefore by estimating $\Sigma_\mathcal{Y}$ and $\Sigma_\mathcal{X}$ and matching their eigenvectors, the attacker can produce, $\hat{M}$, an estimation of M. Using this, data record x_i ($1 \leq i \leq m$) is estimated as $\hat{x}_i = \hat{M}^T y_i$.

The following results (proved in [29]) establishes the key match between the normalized eigenspaces.

THEOREM 15.3 *The eigenvalues of $\Sigma_\mathcal{X}$ and $\Sigma_\mathcal{Y}$ are the same and for all $1 \leq j \leq n$, $M\mathbb{V}^j_\mathcal{X} = \mathbb{V}^j_\mathcal{Y}$, where $M\mathbb{V}^j_\mathcal{X}$ equals $\{Mv : v \in \mathbb{V}^j_\mathcal{X}\}$.*

COROLLARY 15.4 *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 $M = V_\mathcal{Y} D_0 V_\mathcal{X}^T$.*

First assume that the attacker knows the covariance matrices $\Sigma_\mathcal{X}$ and $\Sigma_\mathcal{Y}$ and, thus, computes $V_\mathcal{X}$ and $V_\mathcal{Y}$. By Corollary 15.4, the attacker can perfectly recover M if she can choose the right D from $\mathbb{I}_n$. To do so, the attacker utilizes S and Y, in particular, the fact that these arose as independent samples from $\mathcal{X}$ and $\mathcal{Y} = M\mathcal{X}$. For any $D \in \mathbb{I}_n$, if $D = D_0$, then $V_\mathcal{Y} D V_\mathcal{X}^T S$ and Y have both arisen as independent samples from $\mathcal{Y}$. The attacker will estimate M as $\hat{M} = V_\mathcal{Y} D V_\mathcal{X}^T$, where D was chosen from $\mathbb{I}_n$ so as to maximize the likelihood that $V_\mathcal{Y} D V_\mathcal{X}^T S$ and Y arose from the same random vector. To make this choice, the attacker can use a multi-variate two-sample hypothesis test for equal distributions [42]. The smaller the p-value, the more convincingly the null hypothesis (that $V_\mathcal{Y} D V_\mathcal{X}^T S$ and Y have both arisen as independent samples from $\mathcal{Y}$) can be rejected. Therefore, $D \in \mathbb{I}_n$ is chosen to maximize the p-value.

Finally, the attacker can eliminate the assumption at the start of the previous paragraph by replacing $\Sigma_\mathcal{X}$ and $\Sigma_\mathcal{Y}$ with estimates computed from S and Y. Using the standard sample covariance matrices, the pseudo-code for the attack technique is shown in algorithm 5. A weakness lies in its computation cost, $O(2^n(m+p)^2)$. For high-dimensional data, the technique is infeasible.

[9] http://www.mathworks.com/access/helpdesk/help/techdoc/ref/fminsearch.html

Protocol 5 Eigen-Analysis Attack

Require: Y, the perturbed data matrix and S, the sample data matrix.
Ensure: $\hat{X}$, an estimate of the original data matrix X.

1: Compute standard, sample covariance matrices of S and Y and $\hat{V}_{\mathcal{X}}$ and $\hat{V}_{\mathcal{Y}}$ their normalized eigenvector matrices.
2: Choose $D \in \mathbb{I}_n$ so as to maximize the p-value of two-sample hypothesis test for equal distributions on $\hat{V}_{\mathcal{Y}} D \hat{V}_{\mathcal{X}}^T S$ and Y.
3: Set $\hat{M}$ to $\hat{V}_{\mathcal{Y}} D \hat{V}_{\mathcal{X}}^T$ and $\hat{X}$ to $\hat{M}^T Y$.

It should be noted the eigen-analysis attack does not work if each entry of M were generated independently from some distribution with mean zero and variance σ^2. In that case, $\Sigma_{\mathcal{Y}}$ will equal γI for some constant $\gamma > 0$, thereby killing any useful matching like that in Theorem 15.3.

15.4.3 Other Attacks Based on ICA

Before finishing the section, we briefly describe some attacks based on independent component analysis (ICA) [19].

ICA Overview. Given an n'-variate random vector $\mathcal{V}$, one common ICA model posits that this random vector was generated by a linear combination of independent random variables, *i.e.*, $\mathcal{V} = A\mathcal{S}$ with $\mathcal{S}$ an n-variate random vector with independent components. Typically, $\mathcal{S}$ is further assumed to satisfy the following additional assumptions: (i) at most one component is distributed as a Gaussian; (ii) $n' \geq n$; and (iii) A has rank n.

One common scenario in practice: there is a set of unobserved samples (the columns of $n \times q$ matrix S) that arose from $\mathcal{S}$ which satisfies (i) - (iii) and whose components are independent. But observed is $n' \times q$ matrix V whose columns arose as linear combination of the rows of S. The columns of V can be thought of as samples that arose from a random vector $\mathcal{V}$ which satisfies the above generative model. There are ICA algorithms whose goal is to recover S and A up to a row permutation and constant multiple. This ambiguity is inevitable due to the fact that for any diagonal matrix (with all non-zeros on the diagonal) D, and permutation matrix P, if A, S is a solution, then so is (ADP), $(P^{-1}D^{-1}\mathcal{S})$.

Other Attacks. Liu *et al.* [30] considered matrix multiplicative data perturbation where M is an $n' \times n$ matrix with each entry generated independently from the some distribution with mean zero and variance σ^2. They discussed the application of the above ICA approach to estimate X directly from Y: $\mathcal{S} = \mathcal{X}$, $\mathcal{V} = \mathcal{Y}$, $S = X$, $V = Y$, and $A = M$. They argued the approach to be

problematic because the ICA generative model imposes assumptions not likely to hold in many practical situations: the components of $\mathcal{X}$ are independent with at most one such being Gaussian distributed. Moreover, they pointed out that the row permutation and constant multiple ambiguity further hampers accurate recovery of X. A similar observation is made later by Chen *et al.* [8].

Guo and Wu [15] considered matrix multiplicative perturbation assuming only that M is an $n \times n$ matrix (orthogonal or otherwise). Further they assumed a weaker variant of the known I/O holds: the attacker knows, $\widetilde{X}$, a collection of original data columns from X but does not know to which of the columns in Y these correspond. They develop an ICA-based attack technique for estimating the remaining columns in X. To avoid the ICA problems described in the previous paragraph, they instead applied ICA *separately* to $\widetilde{X}$ and Y producing representations $(A_{\widetilde{X}}, S_{\widetilde{X}})$ and (A_Y, S_Y). They argued that these representations are related in a natural way allowing X to be estimated. Their approach is similar in spirit to the known sample attack described earlier which related S and Y through representations derived through eigen-analysis.

15.4.4 Summary

This section discussed the vulnerabilities of matrix multiplicative data perturbation to certain attacks based on prior knowledge. The primary attack techniques discussed are summarized in Table 15.2.[10]

Table 15.2. Summarization of Attacks on Matrix Multiplicative Perturbation

Categories	*Related Work*	*General Assumptions*
Linear algebra/measure theory	[29]	known I/O, M is orthogonal
MAP Estimation	[28]	known I/O, M is $n' \times n$ with entries generated independently from $\mathcal{N}(0, \sigma^2)$,
Eigen-Analysis	[29]	known sample, M is orthogonal,
ICA	[8, 30]	M has rank n, the data attributes are largely independent and at most one is Gaussian
ICA	[15]	M is $n \times n$, weak known I/O

Chen *et al.* [8] discussed a modification of matrix multiplicative data perturbation to improve its resilience to attack. They examine the combination of matrix multiplicative and additive data perturbation. They argue that this approach offers additional privacy protection, but the utility of the perturbed data

[10] All the attack techniques, except known I/O with orthogonal M, implicitly assume that the original data records arose independently from $\mathcal{X}$.

is negatively affected since additive noise does not preserve Euclidean distance well.

15.5 Attacking *k*-Anonymization

Before concluding this chapter, we briefly survey a very recent body of research aimed at analyzing the vulnerabilities of the popular k-anonymity model [38, 41]. Here, the private data X is perturbed such that each of the resulting records is identical to at least $k-1$ others with respect to a pre-defined set of attributes called *quasi-identifiers*. All of the other attributes are called *sensitive attributes* and these are not modified by the perturbation. This perturbation can be carried out by judicious *value generalization* (*e.g.*, zip 95120 $\rightarrow$ 951**) or *tuple suppression*, and it is aimed at preventing linkage attacks through the quasi-identifiers.

Recently, Machanavajjhala *et al.* [32] developed a background knowledge attack on k-anonymity which we call a *homogeneity attack*. They showed how a lack of diversity among the sensitive attribute values can be used to establish a linkage between individuals and sensitive values. To remedy this problem, they proposed a new privacy definition called l-diversity such that in each equivalence class there are at least l "well-represented" sensitive values. Along the same line, Wong *et al.* [48] proposed an (α, k)-anonymization model such that the relative frequency of the sensitive value in every equivalence class is less than or equal to α. Li *et al.* [25] later developed attacks on l-diversity (*skewness attack* and *similarity attack*), and argued that l-diversity is neither necessary nor sufficient to prevent attribute disclosure. To cope with these problems, they proposed an improved framework called t-closeness, which requires the distribution of a sensitive attribute in any equivalence class to be close to the distribution of the attribute in the original data set.

Wang *et al.* [46] considered the privacy breach caused by the attacker's data mining capabilities. They presented an approach (that combines association rule hiding and *k*-anonymity) to limit the confidence of inferring sensitive properties about the existing individuals.

Aggarwal [2] also argued the original k-anonymity model to be problematic. He considered the case of high dimensional data and pointed out that the exponential number of quasi-identifier combinations can allow precise inference attacks unless an unacceptably high amount of information loss is suffered.

15.6 Conclusion

This chapter provides a detailed survey of attack techniques on additive and matrix multiplicative perturbation. It also presents a brief overview of attacks on k-anonymization. These attacks offer insights into vulnerabilities data perturbation techniques under certain circumstances. In summary, the following

information could lead to disclosure of private information from the perturbed data.

1. Attribute Correlation: Many real world data has strong correlated attributes, and this correlation can be used to filter off additive white noise. See, *e.g.*, [14, 17, 18, 22].

2. Known Sample: Sometimes, the attacker has certain background knowledge about the data such as the *p.d.f.* or a collection of independent samples which may or may not overlap with the original data. See, *e.g.*, [28, 29, 18].

3. Known Inputs/Outputs: Sometimes, the attacker knows a small set of private data and their perturbed counterparts. This correspondence can help the attacker to estimate other private data. See, *e.g.*, [28, 15, 29].

4. Data Mining Results: The underlying pattern discovered by data mining also provides a certain level of knowledge which can be used to guess the private data to a higher level of accuracy. See, *e.g.*, [4, 9, 31, 16, 12, 46].

5. Sample Dependency: Most of the attacks (except the known I/O developed by [29]) discussed in this chapter assume the data as independent samples from some unknown distribution. This assumption may not hold true for all real applications. For certain types of data, such as the time series data, there exists auto correlation/dependency among the samples. How this dependency can help the attacker to estimate the original data is still an open problem.

Notes

The contributions of C. Giannella and K. Liu were equal.

Acknowledgements

The authors wish to thank the U.S. National Science Foundation for their support through awards IIS-0329143 and IIS-0093353. The authors also wish to thank Kamalika Das, Souptik Datta, and Ran Wolff for their assistance.

References

[1] N. R. Adam and J. C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989.

[2] Charu C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st VLDB Conference*, pages 901–909, Trondheim, Norway, 2005.

[3] Charu C. Aggarwal and Philip S. Yu. A condensation based approach to privacy preserving data mining. In *Proceedings of the 9th International*

Conference on Extending Database Technology (EDBT'04), pages 183–199, Heraklion, Crete, Greece, March 2004.

[4] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 247–255, Santa Barbara, CA, 2001.

[5] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 439–450, Dallas, TX, May 2000.

[6] R. Brand. Microdata protection through noise addition. *Lecture Notes in Computer Science - Inference Control in Statistical Databases*, 2316:97–116, 2002.

[7] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 589–592, Houston, TX, November 2005.

[8] K. Chen, G. Sun, and L. Liu. Towards attack-resilient geometric data perturbation. In *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM'07)*, Minneapolis, MN, April 2007.

[9] J. Domingo-Ferrer, F. Sebé, and J. Castellà-Roca. On the security of noise addition for privacy in statistical databases. *Privacy in Statistical Databases*, LNCS3050:149–161, 2004.

[10] A. Evfimevski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the ACM SIGMOD/PODS Conference*, San Diego, CA, June 2003.

[11] S. E. Fienberg and J. McIntyre. Data swapping: Variations on a theme by dalenius and reiss. Technical report, National Institute of Statistical Sciences, Research Triangle Park, NC, 2003.

[12] A. Friedman, R. Wolff, and A. Schuster. Providing k-anonymity in data mining. *Journal of VLDB*, 2006 (to be published).

[13] G. Strang. *Linear Algebra and Its Applications (3rd Ed.)*. Harcourt Brace Jovanovich College Publishers, New York, 1986.

[14] S. Guo and X. Wu. On the use of spectral filtering for privacy preserving data mining. In *Proceedings of the 21st ACM Symposium on Applied Computing*, pages 622–626, Dijon, France, April 2006.

[15] S. Guo and X. Wu. Deriving private information from arbitrarily projected data. In *Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'07)*, Nanjing, China, May 2007.

[16] S. Guo, X. Wu, and Y. Li. Deriving private information from perturbed data using iqr based approach. In *Proceedings of the Second International Workshop on Privacy Data Management (PDM'06)*, Atlanta, GA, April 2006.

[17] S. Guo, X. Wu, and Y. Li. On the lower bound of reconstruction error for spectral filtering based privacy preserving data mining. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'06)*, pages 520–527, Berlin, Germany, September 2006.

[18] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *Proceedings of the 2005 ACM SIGMOD Conference*, pages 37–48, Baltimroe, MD, June 2005.

[19] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4):411–430, June 2000.

[20] I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, second edition, 2002.

[21] D. Jonsson. Some limit theorems for the eigenvalues of a sample covariance matrix. *Journal of Multivariate Analysis*, 12:1–38, 1982.

[22] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'03)*, pages 99–106, Melbourne, FL, November 2003.

[23] J. Kim. A method for limiting disclosure in microdata based on random noise and transformation. In *Proceedings of the American Statistical Association on Survey Research Methods*, pages 370–374, Washington, DC, 1986.

[24] J. J. Kim and W. E. Winkler. Multiplicative noise for masking continuous data. Technical Report Statistics #2003-01, Statistical Research Division, U.S. Bureau of the Census, Washington D.C., April 2003.

[25] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE'07)*, pages 106–115, Istanbul, Turkey, April 2007.

[26] X.-B. Li and S. Sarkar. A tree-based data perturbation approach for privacy-preserving data mining. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18(9):1278–1283, 2006.

[27] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems (TODS)*, 10(3):395–411, 1985.

[28] K. Liu. *Multiplicative Data Perturbation for Privacy Preserving Data Mining*. PhD thesis, University of Maryland, Baltimore County, Baltimore, MD, January 2007.

[29] K. Liu, C. Giannella, and H. Kargupta. 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)*, pages 297–308, Berlin, Germany, September 2006.

[30] K. Liu, H. Kargupta, and J. Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18(1):92–106, January 2006.

[31] M. Kantarcioğlu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *Proceedings of the 10th ACM SIGKDD Conference (KDD'04)*, pages 599–604, Seattle, WA, August 2004.

[32] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2006.

[33] 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.

[34] K. Muralidhar and R. Sarathy. Data shuffling - a new masking approach for numerical data. *Management Science*, 52(5):658–670, May 2006.

[35] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[36] S. R. M. Oliveira and O. R. Zaïane. Privacy preserving clustering by data transformation. In *Proceedings of the 18th Brazilian Symposium on Databases*, pages 304–318, Manaus, Amazonas, Brazil, October 2003.

[37] S. R. M. Oliveira and O. R. Zaïane. Privacy preservation when sharing data for clustering. In *Proceedings of the International Workshop on Secure Data Management in a Connected World*, pages 67–82, Toronto, Canada, August 2004.

[38] P. Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, November/December 2001.

[39] J. W. Silverstein and P. L. Combettes. Signal detection via spectral theory of large dimensional random matrices. *IEEE Transactions on Signal Processing*, 40(8):2100–2105, 1992.

[40] G. W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[41] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[42] G. J. Szekély and M. L. Rizzo. Testing for equal distributions in high dimensions. *InterStat*, November(5), 2004.

[43] P. Tendick. Optimal noise addition for preserving confidentiality in multivariate data. *Journal of Statistical Planning and Inference*, 27(2):341–353, 1991.

[44] M. Trottini, S. E. Fienberg, U. E. Makov, and M. M. Meyer. Additive noise and multiplicative bias as disclosure limitation techniques for continuous microdata: A simulation study. *Journal of Computational Methods in Sciences and Engineering*, 4:5–16, 2004.

[45] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. In *IEEE Transactions on Knowledge and Data Engineering*, volume 16, pages 434–447, 2004.

[46] K. Wang, Benjamin C. M. Fung, and Philip S. Yu. Handicapping attacker's confidence: an alternative to k-anonymization. *Knowledge and Information Systems*, 11(3):345–368, 2007.

[47] E. P. Wigner. On the statistical distribution of the widths and spacings of nuclear resonance levels. *Proceedings of the Cambridge Philosophical Society*, 47:790–798, 1952.

[48] R. Chi-Wing Wong, J. Li, A. Wai-Chee Fu, and K. Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD Conference (KDD'06)*, pages 754–759, Philadelphia, PA, August 2006.

[49] Y. Zhu and L. Liu. Optimal randomization for privacy preserving data mining. In *Proceedings of the 10th ACM SIGKDD Conference (KDD'04)*, pages 761–766, Seattle, WA, August 2004.