APPROVAL SHEET

Title of Dissertation: Multiplicative Data Perturbation for Privacy Preserving Data Mining

Name of Candidate: Kun Liu
Doctor of Philosophy, 2007

Dissertation and Abstract Approved: _____

Dr. Hillol Kargupta
Associate Professor
Department of Computer Science and Electrical Engineering

Date Approved: _____

Curriculum Vitae

Name: Kun Liu.
Permanent Address:
Degree and date to be conferred: Doctor of Philosophy, 2007.
Date of Birth:
Place of Birth:

Collegiate institutions attended:

- University of Maryland Baltimore County (UMBC), Baltimore, MD, USA
  Doctor of Philosophy, Computer Science, 2007.
- Nankai University, Tianjin, China,
  Bachelor of Science, Computer Science, 2001.

Major: Computer Science.

Professional publications:

Refereed Journals

[1] **K. Liu**, C. Giannella, and H. Kargupta, "An attacker's view of exact and approximate distance preserving perturbations for privacy preserving data mining," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, January 2007 (in preparation).

[2] **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)*, vol. 18, no. 1, pp. 92-106, January 2006.

[3] S. Bandyopadhyay, C. Giannella, U. Maulik, H. Kargupta, **K. Liu**, and S. Datta, "Clustering distributed data streams in peer-to-peer environments," *Information Sciences*, vol. 176, no. 14, pp. 1952-1985, July 2006.

Refereed Conference Proceedings

[4] **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)*, Berlin, Germany, September 2006, pp. 297-308.

[5] C. Giannella, **K. Liu**, T. Olsen, and H. Kargupta, "Communication efficient construction of decision trees over heterogeneously distributed data," in *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, Brighton, UK, November 2004, pp. 67-74.

[6] H. Kargupta, R. Bhargava, **K. Liu**, M. Powers, P. Blair, S. Bushra, J. Dull, K. Sarkar, M. Klein, M. Vasa, and D. Handy, "Vedas: A mobile and distributed data stream mining system for real-time vehicle monitoring," in *Proceedings of the 2004 SIAM International Data Mining Conference (SDM'04)*, Orlando, FL, April 2004, pp. 300-311. **Best application paper nomination.**

[7] H. Kargupta, **K. Liu**, and J. Ryan, "Privacy sensitive distributed data mining from multi-party data," in *Proceedings of the First NSF/NIJ Symposium on Intelligence and Security Informatics*, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, Tucson, AZ, June 2003, pp. 336-342.

[8] H. Kargupta, **K. Liu**, S. Datta, J. Ryan, and K. Sivakumar, "Homeland security and privacy sensitive data mining from multi-party distributed resources," in *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, vol. 2, St. Louis, MO, May 2003, pp. 1257-1260.

Refereed Workshop Proceedings

[9] **K. Liu**, K. Bhaduri, K. Das, P. Nguyen, and H. Kargupta, "Client-side web mining for community formation in peer-to-peer environments," in *Proceedings of KDD Workshop on Web Mining and Web Usage Analysis (WebKDD'06)*, Philadelphia, PA, August 2006, pp. 130-139, held in conjunction with the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06).

[10] H. Kargupta, **K. Liu**, and J. Ryan, "Random projection and privacy preserving correlation computation from distributed data," in *Proceedings of the 6th International Workshop on High Performance Data Mining: Pervasive and Data Stream Mining (HPDM:PDS'03)*, San Francisco, CA, May 2003, held in conjunction with the third International SIAM Conference on Data Mining (SDM'03).

[11] H. Kargupta, **K. Liu**, S. Datta, J. Ryan, and K. Sivakumar, "Link analysis, privacy preservation, and random perturbations," in *Proceedings of KDD Workshop on Link Analysis for Detecting Complex Behavior (LinkKDD'03)*, Washington D.C., July 2003, held in conjunction with the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03).

Professional positions held:

- Research Assistant 01/2003 – 01/2007
  Distributed Adaptive Discovery and Computation Lab, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County (UMBC).
- Student Intern 05/2005 – 08/2005
  Microsoft Research Asia (MSRA), Beijing
- Teaching Assistant 08/2001 – 05/2003
  Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County (UMBC).

ABSTRACT

Title of Dissertation: Multiplicative Data Perturbation for Privacy Preserving Data Mining

Kun Liu, Doctor of Philosophy, 2007

Dissertation directed by: Dr. Hillol Kargupta
Associate Professor
Department of Computer Science and Electrical Engineering

Recent interest in the collection and monitoring of data using data mining technology for the purpose of security and business-related applications has raised serious concerns about privacy issues. For example, mining health care data for the detection of disease outbreaks may require analyzing clinical records and pharmacy transaction data of many individuals over a certain area. However, releasing and gathering such diverse information belonging to different parties may violate privacy laws and eventually be a threat to civil liberties. Privacy preserving data mining strives to provide a solution to this dilemma. It aims to allow useful data patterns to be discovered without compromising privacy.

In 2000, Agrawal and Srikant proposed the addition of i.i.d. white noise for privacy protection. However, Kargupta *et al.* pointed out that additive noise can be easily filtered off revealing a good approximation of the private data. This makes one wonder about the possibility of using multiplicative noise. This dissertation systematically investigated different multiplicative data perturbation techniques for privacy preserving data mining. These types of perturbation distort the private data by multiplying some random noise and only the perturbed version is released for data mining analysis. Extensive theoretical and experimental results were provided to support the following primary contributions.

First, we examined the security issues of distance preserving data perturbation. This technique is potentially very useful in that some important data mining algorithms can be efficiently applied to the perturbed data and produce exactly the same results as if applied to the original data. However, the issue of how well the original data is hidden had not been carefully studied. We took a step in this direction by considering three types prior knowledge an attacker may have and use to design attack techniques to recover the original data. Our results offered insight into the vulnerabilities of distance preserving perturbation.

Second, we explored a random projection-based data perturbation that preserves the inner products and Euclidean distances in the original data with high probabilities. We proposed a maximum a posteriori probability (MAP) estimate-based Bayes privacy model to quantify the privacy. Guidelines were offered for the data owner to control the privacy/accuracy tradeoff when perturbing the data. Theoretical analysis showed that this perturbation provides higher privacy protection than distance preserving perturbation, but with little loss of accuracy.

MULTIPLICATIVE DATA PERTURBATION FOR PRIVACY PRESERVING DATA MINING

by

Kun Liu

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

To my parents and grandparents.

ACKNOWLEDGMENTS

During the hard years passed for completing my doctoral research, many great people have given continuous and generous support that I really want to acknowledge here.

First, I thank my advisor, Dr. Hillol Kargupta, for his continuous guidance and support during the last five years. He teaches me how to think creatively and gives me room to develop my own interests. Without him, this work would not be possible. I am also grateful to Dr. Chris Giannella, who has been working with me closely and always shares with me his motivations, thoughts and expertise. I have learned a lot from him. Thanks are also due to my doctoral examination committee members, Dr. Yun Peng, Dr. Tim Oates, Dr. Anupam Joshi and Dr. Aryya Gangopadhyay, for their encouragement and intellectual contributions to this work. This research is supported by the U.S. NSF Grant IIS-0329143, and partially supported by NSF CAREER award IIS-0093353.

Furthermore, it is a pleasure to express my thanks to my writing advisor Mr. John Rollins, who helped proofreading the whole dissertation, chapter by chapter and word by word. To my lab mates, Jessica Ryan, Todd Olsen, Souptik Datta, Haimonti Dutta, Kanishka Bhaduri, Kamalika Das, Sourav Mukherjee and Phuong Nguyen. Thank you for spending these years working with me and experiencing the same frustration and excitement in our research. I wish all your hard work will finally bear fruit. To my dear friends, Xu Wang, Jing Zhang, Qianjun Xu, Xiaojie Zhou, Huifen Li, Yan Liang, Ling Tang, Cha Li, Zhongli Ding, Li Ding, Yongmei Shi, Rong Pan, Yang Yu, Yu Zhou, Yong Rao, Gang Wu, Jun Wang, Ying Wang, Hua Ling, Xiang Li... I am sorry I can’t list all of your names here, but I thank you for your priceless friendship and help in my hardest time.

I want to acknowledge my special thanks to my mom Zhao, Huifen and dad Liu, Xinmin for bring me up and for their lifelong love. They are the peaceful harbor for me

whenever I feel tired and frustrated; they are the sanctuary where my soul can find sweet rest from the struggle and the tension of the life.

Finally, I want to give my thanks to Ting Zhong, the most beautiful girl in the world, for her continued encouragement, caring, sharing and love.

December 19, 2006

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background

MOST of our daily activities are now routinely recorded and analyzed by a variety of governmental and commercial organizations for the purpose of security and business related applications. From telephone calls to credit card purchases, from Internet surfing to medical prescription refills, we generate data with almost every action we take. Collecting and analyzing such data are causing a major concern about our privacy. A Forbes cover story in November 1999, *I Know What You Did Last Night*, highlights the way that different slices of consumer data can now be pulled together to create a vivid picture of any individual's life [1]. Privacy has been gaining more attention since September 11. To handle the terrorism, the government needed to examine, using data mining technology, more information about individuals to detect unusual disease outbreaks, financial fraudulent behaviors, network intrusions, etc. While all of these applications of data mining can benefit our society, there is also a negative side to this technology because it could be a threat to the individuals' privacy. Recently, we have heard much about national security vs. privacy in newspapers, magazines, research articles, and on television talk shows [2]. In 2003, concerns over the U.S. Total Information Awareness (also known as Terrorism Information Awareness) project even led to the introduction of a bill in the U.S. Senate that

would have banned any data mining programs in the U.S. Department of Defense. To eliminate the misguided impression, SIGKDD, an ACM's special interest group on knowledge discovery and data mining, even sent out a letter to claim *"Data Mining" is NOT Against Civil Liberties* [3]. However, as the letter pointed out that:

> the best (and perhaps only) way to overcome the "limitations" of data mining techniques is to do more research in data mining, including areas like data security and privacy preserving data mining, which are actually active and growing research areas.

In 2000, Agrawal and Srikant [4] published their early work on privacy preserving data mining. They proposed an additive data perturbation technique for decision tree construction in a client/server scenario. In their work, each client has a numerical private attribute x_i and the server wants to learn the distribution of these attributes to build a classification model. The clients mask their attributes x_i by adding random noise r_i drawn independently from a known distribution. The server collects the values of $x_i + r_i$ and reconstructs x_i's distribution. However, Kargupta *et al.* [5] later questioned the use of random additive noise and pointed out that additive noise can be easily filtered out in many cases. Their work was further extended by Huang *et al.* [6], Guo *et al.* [7] and many else.

The drawback of additive noise makes one wonder about the possibility of using multiplicative noise for protecting the data privacy. In this type of perturbation, the private data is distorted by multiplying some random noise and only the perturbed version is released for data mining analysis. To our best knowledge, this technique has not been carefully studied in the literature. This dissertation specifically investigates different multiplicative data perturbations for PPDM. It presents extensive theoretical and experimental results on the accuracy and privacy of each of the multiplicative data perturbation techniques. Thus, valuable information is gained into effectiveness of multiplicative perturbations for PPDM.

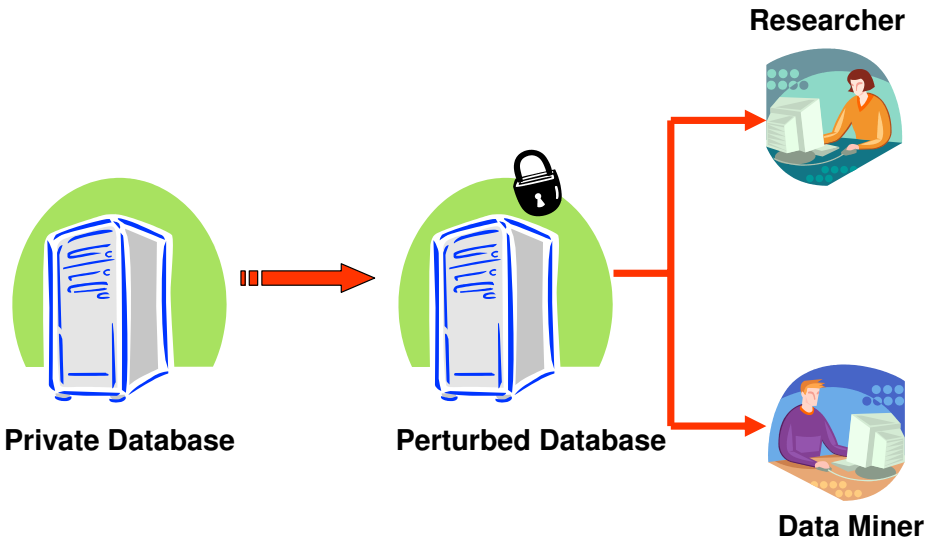


FIG. 1.1. Census Model.

1.2 Problem Statement

The problem we are interested in can be stated as follows. An organization has a private database and wishes to make it publicly available for data analysis while keeping the original data records private. To achieve this goal, this organization transforms its database into another form and only release that. A third party data miner or a researcher can analyze and discover useful patterns of the original data from only the transformed data. This is generally referred to as the census model, as illustrated by Figure 1.1.

1.3 Contributions of this Dissertation

This dissertation has systematically studied multiplicative data perturbation techniques for privacy preserving data mining. It has made the following main contributions.

1. We examined the effectiveness of distance preserving perturbations in privacy preserving data mining. These techniques are potentially very useful in that some important data mining algorithms can be efficiently applied to the transformed data and

produce exactly the same results as if applied to the original data, *e.g.*, distance-based clustering and k-nearest neighbor classification. However, the issue of how well the original data is hidden has, to our knowledge, not been carefully studied. We took a step in this direction by assuming the role of an attacker armed with three types of prior information regarding the original data. We studied how well the attacker can recover the original data from the transformed data and prior information. Three different attack techniques were developed. The first one was based on linear algebra and statistical theory, the second on principal component analysis (PCA), and the third on independent component analysis (ICA). Our results offered insight into the advantages and vulnerabilities of distance preserving perturbations.

2. We further proposed a random projection-based data perturbation that preserves distance with high probabilities, and derived the analytic error bounds for the accuracy. We proposed a maximum a posteriori probability (MAP) estimate-based Bayes privacy model to quantify the privacy offered by the perturbation technique. Our analysis showed that, under mild assumptions, random projection-based data perturbation did not offer the attacker more information about the private data than what had been implied by the distance preserving property of random projection itself. In addition, guidelines were offered for the data owner to control the privacy/accuracy tradeoff when perturbing the data. Our theoretical analysis and experimental results provided valuable information about the characteristics of this perturbation.

1.4 Dissertation Organization

This dissertation is organized as follows.

Chapter 1: This chapter presents the background of this research, the problem definition, the contributions, and the organization of this dissertation.

Chapter 2: This chapter offers an overview of various techniques and methodologies that have been developed in the privacy preserving data mining area. It notes that the main consideration in privacy preserving data mining is two fold: 1) *data hiding*: sensitive raw data should be modified or trimmed out from the original database while the important underlying patterns of the data should still be preserved, and 2) *rule hiding*: sensitive knowledge which can be discovered from the data should be filtered out. The objective of privacy preserving data mining is to allow meaningful patterns to be identified while keeping private information private during and after the mining process.

Chapter 3: This chapter briefly reviews two multiplicative data perturbation techniques that have been studied in the statistics community. These perturbations distort each data element independently, and they are primarily used to mask the private data while allowing summary statistics (*e.g.,* sum, mean, variance) of the original data to be estimated. This chapter notes that these perturbation schemes are equivalent to the additive perturbation after a logarithmic transformation, and therefore, they are vulnerable to many attacks designed for additive perturbation. Moreover, the Euclidean distances among data records are generally not preserved after perturbation.

Chapter 4: This chapter discusses a new multiplicative perturbation technique called distance preserving data perturbation. The perturbed data preserves inner products and Euclidean distances. Many important data mining algorithms can be efficiently applied to the perturbed data and produce exactly the same results as if applied to the original data. This chapter first talks about the basic mathematical properties of this perturbation. Then, it addresses the security issues of this technique by studying how well an attacker can recover the original data from the perturbed data and other prior knowledge. Three attack algorithms are designed. The first is based on basic properties of linear algebra, the second on principal component analysis (PCA), and the third on independent component analysis (ICA). As such, valuable information is gained into the effectiveness of distance preserving

transformation for privacy preserving data mining.

Chapter 5: This chapter proposes a random projection-based multiplicative data perturbation technique. This technique maps the data onto a lower dimensional space while maintaining, with high probabilities, the pairwise Euclidean distances and the inner products of the original data. This chapter first derives some analytic error bounds for the accuracy of the distances preserved by random projection. Then, it offers a Bayes privacy model to measure the privacy provided by the perturbation. To be more specific, it considers the use of maximum a posteriori probability (MAP) estimate to recover the original data, and to quantify the privacy. A closed-form expression about the (upper bound of the) privacy breach is derived, which can be used together with the error bounds to guide the perturbation in practice. Next, this chapter examines several privacy disclosure scenarios and analyzes the efficacy of the corresponding attacks.

Chapter 6: This chapter concludes this dissertation and outlines the directions for future research.

Chapter 2

BACKGROUND AND RELATED WORK

Recent interest in the collection and monitoring of data using data mining technology for the purpose of security and business-related applications has raised serious concerns about privacy issues. Sometimes, individual or organizational entities may not be willing to divulge the sensitive raw data; sometimes, the knowledge and/or patterns detected by a data mining system may be used in a counter-productive manner that violates the privacy policy. The main objective of privacy preserving data mining is to develop algorithms for modifying the original data or modifying the computation protocols in some way, so that during and after the mining process, the private data and private knowledge remain private while other underlying data patterns or models can still be effectively identified.

There exists a growing body of literature on privacy preserving data mining. This chapter presents a classification and an extended description of the various techniques and methodologies that have been developed in this area (see Table 2.1 for a brief overview of the categories).

2.1 Data Hiding

The main objective of data hiding is to transform the data or to design new computation protocols so that the private data remains private during and/or after data mining

Data Hiding	Data Perturbation { Value Distortion { Additive Perturbation; Multiplicative Perturbation; Data Microaggregation; Data Anonymization; Data Swapping; Other Randomization Techniques } ; Probability Distribution { Sampling Method; Analytical Method } } Secure Multi-Party Computation (SMC) / Cryptographic Protocols Distributed Data Mining (DDM)
Rule Hiding	Association Rule Hiding { Data Perturbation; Data Blocking } Classification Rule Hiding { Parsimonious Downgrading

Table 2.1. A brief overview of privacy preserving data mining techniques.

operations while the underlying data patterns or models can still be discovered.

2.1.1 Data Perturbation

Data perturbation techniques can be grouped into two main categories, which we call the value distortion technique and probability distribution technique. The value distortion technique perturbs data elements or attributes directly by either additive noise, multiplicative noise or some other randomization procedures. On the other hand, the probability distribution technique considers the private database to be a sample from a given population that has a given probability distribution. In this case, the perturbation replaces the original database by another sample from the same (estimated) distribution or by the distribution itself.

Note that there has been extensive research in the area of statistical databases (SDB) on how to provide summary statistical information without disclosing individuals' confidential data (*e.g.,* [8–10]). The privacy issues arise when the summary statistics are derived from data of very few individuals. A popular disclosure control method is data perturbation,

which alters individual data in a way such that the summary statistics remain approximately the same. However, problems in data mining become somewhat different from those in SDBs. Data mining techniques, such as clustering, classification, prediction and association rule mining, are essentially relying on more sophisticated relationships among data records or data attributes, but not just simple summary statistics. This dissertation specifically focuses on data perturbation for privacy preserving data mining. In the following, we will primarily discuss different perturbation techniques in the data mining area. Some important perturbation approaches in SDBs are also covered for the sake of completeness.

Additive Perturbation The work in [4, 11] proposed an additive data perturbation technique for building decision tree classifiers. In this technique, each client has a numerical attribute x_i and the server (or data miner) wants to learn the distribution of these attributes to build a classification model. The clients randomize their attributes x_i by adding random noise r_i drawn independently from a known distribution such as a uniform distribution or a Gaussian distribution. The server (or data miner) collects the values of $x_i + r_i$ and reconstructs x_i's distribution using a version of the Expectation-Maximization (EM) algorithm. This algorithm provably converges to the maximum likelihood estimate of the desired original distribution [11].

Kargupta *et al.* [5] questioned the use of random additive noise and pointed out that additive noise can be easily filtered out in many cases that will possibly compromise the privacy. To be more specific, they proposed a random matrix-based Spectral Filtering (SF) technique to recover the original data from the perturbed data. Their empirical results have shown that the recovered data can be reasonably close to the original data. However, two important questions remain to be answered: 1) What are the theoretical lower bound and upper bound of the reconstruction error; and 2) What are the key factors that influence the accuracy of the data reconstruction?

Guo and Wu [7] further investigated the Spectral Filtering technique and derived an

upper bound for the Frobenius norm of the reconstruction error using matrix perturbation theory. They also proposed a Singular Value Decomposition (SVD)-based reconstruction method and derived a lower bound for the reconstruction error [12]. They then proved the equivalence between the SF and SVD approach, and as a result, the lower bound of SVD approach can also be considered as the lower bound of the SF approach.

Huang *et al.* [6] pointed out that the key factor that decides the accuracy of data reconstruction is the correlation among the data attributes. Their results have shown that when the correlations are high, the original data can be reconstructed more accurately, that is, more private information can be disclosed. They further proposed two data reconstruction methods based on data correlations: one used the Principal Component Analysis (PCA), and the other used the Bayes Estimate (BE) technique, which in essence is a maximum a posterior probability estimation. To improve privacy, they designed a modified additive perturbation scheme, in which they let the correlation of random noise *similar* to the original data. This approach is similar with many data perturbation approaches used in the statistics community (*e.g.,* [13, 14]). Their results have shown that the reconstruction accuracy of both PCA and BE techniques get worse as the similarity increases.

Given the large body of existing signal-processing literature on filtering random additive noise, the utility of random additive noise for privacy preserving data mining is not quite clear.

Multiplicative Perturbation Two basic forms of multiplicative noise have been studied in the statistics community [15]. One multiplies each data element by a random number that has a truncated Gaussian distribution with mean one and small variance. The other takes a logarithmic transformation of the data first, adds multivariate Gaussian noise, then takes the exponential function *exp(.)* of the noise-added data. Neither of these perturbations preserve

pairwise distance among data records. [1]

To facilitate large scale data mining applications, Liu *et al.* [16] proposed an approach where the data is multiplied by a randomly generated matrix – in effect, the data is projected into a lower dimensional random space. This technique preserves distance on expectation. Oliveira and Zaiane [17], Chen and Liu [18] discussed the use of random rotation for privacy preserving clustering and classification. These authors observed that the distance preserving nature of random rotation enables a third party to produce exactly the same data mining results on the perturbed data as if on the original data. However, they did not analyze the privacy limitations of random rotation. Liu *et al.* [19] addressed the privacy issues of distance preserving perturbation (including rotation) by studying how well an attacker can recover the original data from the transformed data and prior information. They proposed two attack techniques: the first is based on basic properties of linear algebra and the second on principal component analysis. Their analysis explicitly illuminated scenarios where privacy can be breached. As such, valuable information was gained into the effectiveness of distance preserving transformation for privacy preserving data mining. Mukherjee *et al.* [20] considered the use of discrete fourier transformation (DFT) and discrete cosine transformation (DCT) to perturb the data. Only the high energy DFT/DCT coefficients were used, and the transformed data in the new domain approximately preserved the Euclidean distance. The DFT/DCT coefficients were further permutated to enhance the privacy protection level. However, the authors did not offer a rigorous analysis of the privacy. Also note that if no coefficients were dropped, their technique would be fundamentally the same as distance preserving transformation; therefore, the privacy issues could be analyzed using the model proposed by Liu *et al.* [19].

Data Microaggregation Data microaggregation is a popular data perturbation approach

[1] In Chapter 3 we will discuss these perturbation schemes in details.

in the area of secure statistical databases (SDBs). For a dataset with a single private attribute, univariate microaggregation (*e.g.*, [21]) sorts data records by the private attribute, groups adjacent records into groups of small sizes, and replaces the individual private values in each group with the group average. Multivariate microaggregation considers all the attributes and groups data using a clustering technique (*e.g.*, [22, 23]). This approach primarily considers the preservation of data covariance instead of the pairwise distance among data records.

Recently, two multivariate microaggregation approaches have been proposed by researchers in the data mining area. Aggarwal and Yu [24] presented a condensation approach to privacy preserving data mining. This approach first partitions the original data into multiple groups of predefined size. For each group, a certain level of statistical information (*e.g.*, mean and covariance) about different data records is maintained. This statistical information is used to create anonymized data that has similar statistical characteristics to the original dataset, and only the anonymized data is released for data mining applications. This approach preserves data covariance instead of the pairwise distance among data records. Li *et al.* [25] proposed a kd-tree based perturbation method, which recursively partitions a dataset into smaller subset such that data records in each subset are more homogeneous after each partition. The private data in each subset are then perturbed using the subset average. The relationships between attributes are expected to be preserved.

Data Anonymization Sweeney [26] developed the *k-anonymity* framework wherein the original data is transformed so that the information for any individual cannot be distinguished from $(k-1)$ others. Generally speaking, anonymization is achieved by suppressing (deleting) individual values from data records (*e.g.*, , name and social security numbers are removed), and/or replacing every occurrence of certain attribute values with a more general value (*e.g.*, the zip codes 21250-21259 might be replaced with 2125*). A variety of refinements of this framework have been proposed since its initial appearance. Some of the

work (*e.g.*, [26,27]) start from the original dataset and systematically or greedily generalize it into one that is k-anonymous. Some (*e.g.*, [28]) start with a fully generalized dataset and systematically specialize the dataset into one that is minimally k-anonymous.

The problem of k-anonymization is not simply to find any k-anonymization, but to, instead, find one that is "good" or even "best" according to some quantifiable cost metric. Each of the previous work provides its own unique cost metrics for modeling desirable anonymization. Cost metrics typically tally the information loss resulting from the suppression or generalizations applied. As an illustration, we will show two cost metrics here.

The first metric was proposed by Bayardo and Agrawal [28]. This metric attempts to capture in a straightforward way the desire to maintain discernibility between data records as much as is allowed by a presetting of k. This discernibility metric assigns a penalty to each data record based on how many records in the transformed dataset are indistinguishable from it. If an unsuppressed record falls into an induced equivalence class of size j, that record is assigned a penalty of j. If a record is suppressed, it is assigned a penalty of $|D|$, the size of the original dataset. This penalty reflects the fact that a suppressed record cannot be distinguished from any other record in the dataset. This metric can be mathematically expressed as follows:

$$Cost(g, k, D) \quad = \quad \sum_{\forall E \text{ s.t } |E| \geq k} |E|^2 + \sum_{\forall E \text{ s.t. } |E| < k} |D||E|,$$

where E is the equivalence classes of records in D induced by the anonymization function g. The first sum of the above expression computes penalties for each non-suppressed record, the second for suppressed records.

The second cost metric was proposed by Iyengar [29]. This metric can be applied when records are associated with categorical class labels. Thus, the anonymization can produce equivalence classes consist of records that are uniform with respect to the class

label. This classification metric assigns no penalty to an unsuppressed tuple if it belongs to the majority class within its induced equivalence class. All other tuples are penalized a value of 1. This metric can be mathematically stated as follows:

$$Cost(g, k, D) = \sum_{\forall E \text{ s.t. } |E| \geq k} (|minority(E)|) + \sum_{\forall E \text{ s.t. } |E| < k} |E|,$$

where the minority function accepts a set of class labeled records and returns the subset of records belonging to any minority class with respect to that set. The first sum of the above expression penalizes non-suppressed records, the second penalizes suppressed records. Iyengar has shown that this metric produces anonymized datasets that give better classification models than do class oblivious metrics.

Recently, Machanavajjhala *et al.* [30] pointed that simple *k-anonymity* is vulnerable to strong attacks due to the lack of diversity in the sensitive attributes. They proposed a new privacy definition called l-diversity. The main idea behind l-diversity is the requirement that the values of the sensitive attributes are well represented in each group. Other enhanced *k-anonymity* models have been proposed elsewhere [31, 32].

Data Swapping This technique transforms the database by switching a subset of attributes between selected pairs of records so that the individual record entries are unmatched, but the statistics (*e.g.*, marginal distributions of individual attributes) are maintained across the individual fields. This technique was first proposed by Dalenius and Reiss [33]. A variety of refinements and applications of data swapping have been addressed since its initial appearance. We refer readers to [34] for a thorough treatment.

Other Randomization Techniques The work in [35, 36] considered categorical data perturbation in the context of association rule mining. This work was extended in [37] where a rigorous framework for quantifying privacy breaches was introduced. This framework uses a key concept of γ-amplification and applies without any assumptions of the underlying

distribution from which the original data is drawn. The work in [38] considered this framework again and showed how to optimally set the perturbation parameters for reconstruction while maintaining γ-amplification. Along a related line, Verykios [39] considered perturbation techniques that allow the discovery of some association rules while hiding others considered to be sensitive.

Sampling Method Liew *et al.* [40] proposed a probability distribution-based approach for protecting a single confidential attribute in a private database. This approach consists of three steps: 1) estimate the underlying probability density function of the attribute; 2) generate a new sample set from the estimated density function; and 3) substitute the new sample for the original attribute in the same rank order, that is, the smallest value of the new sample should replace the smallest value in the original data, and so forth. This approach is applicable to both numeric and categorical attributes. The noise introduced by this approach is larger when the private database is small; thus, better security is achieved, but biased-query responses are provided with users. When the size of the database increases, the bias becomes smaller, but less security of confidential attribute is achieved.

Analytical Method Lefons *et al.* [41] proposed an approach for protecting multi-numerical sensitive attributes by replacing the original private database with its probability density. The key contribution of their work lies in the approximation of the data distribution by orthogonal polynomials. The coefficients used in the computation of the approximation are called canonical coefficients. These coefficients are well suited for usage in an online environment because they can be adopted easily in case of insertions and deletions of the database records. However, if the estimated probability density function is a very precise description of the original data, there is hardly any protection against partial disclosures. On the other hand, if there is large deviation between the density function and the original sensitive data, issues such as how to avoid bias and how to control the trade-off between precision and security need to be carefully addressed.

2.1.2 Secure Multi-party Computation (SMC)

Definition Secure Multi-party Computation (SMC) [42] considers the problem of evaluating a function of two or more parties' secret inputs, such that no party learns anything but the designated output of the function. Concretely, we assume we have inputs $x_1, \ldots, x_n$, where party i owns x_i, and we want to compute function $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ such that party i gets y_i and nothing more than that.

Example As an example, we may consider Yao's millionaire's problem: two millionaires meet in the street and want to find out who is richer without having to reveal their actual fortune to each other. The function computed in this case is a simple comparison between two numbers. If the result is that the first millionaire is richer, then he knows that, but this should be all the information he learns about the other guy.

Adversarial Behavior It is common to model cheating by considering adversarial parties that attempt to obtain information about the private inputs of their peers. SMC typically studies two types of adversaries: A *semi-honest* adversary (also known as *passive*, or *honest but curious* adversary) is a party who follows the protocol properly, yet attempts to learn additional information by analyzing all the intermediate results and the messages received during the protocol execution. On the other hand, a *malicious* adversary may arbitrarily deviate from the protocol specification. A malicious adversary could refuse to participate in the protocol when the protocol is first invoked, could substitute its input and enter the protocol with an input other than the one provided with it, and could abort the protocol prematurely. It is obviously easier to design a solution that is secure against semi-honest adversaries than it is to design a solution for malicious adversaries. In practice, people usually first design a secure protocol for the semi-honest scenario, and then transform it to a protocol that is secure against malicious adversaries. This transformation can be done by requiring each party to use zero-knowledge proofs to prove that each step that it is taking

follows the protocol specification.

Privacy Generally speaking, an SMC protocol *privately* computes a function if any information that a party can obtain can be essentially obtained by that party through its own inputs and outputs. An alternative definition compares the results of the actual computation to that of an *ideal* computation. Here the *ideal* computation assumes there exists a *trusted party* who does not deviate from the protocol specification at all, and does not attempt to cheat. All parties send their private inputs to the *trusted party*, who computes the function and sends the appropriate results back to all the parties. We say a protocol is secure or private if anything that an adversary can learn in the actual world can also be learned in the ideal world, namely from its own inputs and from the outputs it receives from the *trusted party*. In essence, protocols satisfying this definition prevent an adversary from gaining any extra advantage in the actual world over what it could have gained in an ideal world.

Building Blocks We describe here some representative building blocks of secure multi-party computation.

- **Oblivious Transfer** In cryptography, an oblivious transfer protocol is a protocol by which a sender sends some information to the receiver, but remains oblivious as to what is sent. Oblivious transfer is one of the most important protocols for secure computation. It has been shown by Kilian [43] that oblivious transfer is sufficient for secure computation in the sense that given an implementation of oblivious transfer it is possible to securely evaluate any polynomial time computable function without any additional primitive. A simply form of oblivious transfer called "1 out of 2 oblivious transfer,", denoted by OT_1^2, was developed later by Shimon Even, Oded Goldreich, and Abraham Lempel [44]. This protocol involves two parties, the *sender* and the *receiver*. The sender's input is a pair (x_0, x_1) and the receiver's input is a bit $\lambda \in \{0, 1\}$. At the end of the protocol the receiver learns x_λ and nothing else, and the sender learns nothing. Oblivious transfer protocols can be designed based

on virtually all known constructions of trapdoor functions, for example, public key cryptosystems. In the case of semi-honest adversaries, there exist simple and efficient protocols for oblivious transfer [44, 45].

As an illustration of the application of oblivious transfer, let us consider the following problem. Assume there are two parties. Party 1 holds $a_1 \in \{0,1\}, b_1 \in \{0,1\}$, and party 2 holds $a_2 \in \{0,1\}, b_2 \in \{0,1\}$. We are interested in computing the function $f = (a_1 + a_2) \cdot (b_1 + b_2)$ such that upon completion of the computation, Party 1 has a random number $c_1 \in \{0,1\}$; Party 2 has a random number $c_2 \in \{0,1\}$ such that $c_1 + c_2 = (a_1 + a_2) \cdot (b_1 + b_2)$. In other words, if we use the notation $(input_1, input_2) \mapsto (output_1, output_2)$ to define the result of a function, then f is the function $((a_1, b_1), (a_2, b_2)) \mapsto (c_1, c_2)$. Here $\cdot$ corresponds to a bitwise AND and $+$ corresponds to a bitwise XOR. The basic procedure for privately computing f is illustrated in Algorithm 2.1.2.1. Table 2.2 shows the values of both parties' inputs and outputs.

Algorithm 2.1.2.1 Privately Computing $c_1 + c_2 = (a_1 + a_2) \cdot (b_1 + b_2)$

Inputs: Party i holds $(a_i, b_i) \in \{0,1\} \times \{0,1\}, i = 1, 2$.
Outputs Party 1 outputs c_1, Party 2 outputs c_2, and $c_1 + c_2 = (a_1 + a_2) \cdot (b_1 + b_2)$.

1: Party 1 randomly selects $c_1 \in \{0,1\}$.
2: Party 1 and Party 2 engage in a 1-out-of-4 oblivious transfer, where Party 1 plays the sender and Party 2 plays the receiver. The input to the sender is the 4-tuple $\{c_1 + a_1 \cdot b_1, c_1 + a_1 \cdot (b_1 + 1), c_1 + (a_1 + 1) \cdot b_1, c_1 + (a_1 + 1) \cdot (b_1 + 1)\}$. The input to the receiver is $1 + 2a_2 + b_2 \in \{1, 2, 3, 4\}$.

- **Circuit Evaluation** Yao [42] presented a constant-round protocol for privately computing any probabilistic polynomial-time function. The protocol is based on expressing the function as a combinatorial circuit with gates defined over some fixed base

Party 1: (a_1, b_1)	(a_1, b_1)	(a_1, b_1)	(a_1, b_1)	(a_1, b_1)
Party 2: (a_2, b_2)	(0,0)	(0,1)	(1,0)	(1,1)
OT_1^4 Input:	1	2	3	4
OT_1^4 Output:	$c_1 +$ $a_1 \cdot b_1$	$c_1 +$ $a_1 \cdot (b_1 + 1)$	$c_1 +$ $(a_1 + 1) \cdot b_1$	$c_1 +$ $(a_1 + 1) \cdot (b_1 + 1)$
Party 2's Output (c_2):	$c_1 +$ $a_1 \cdot b_1$	$c_1 +$ $a_1 \cdot (b_1 + 1)$	$c_1 +$ $(a_1 + 1) \cdot b_1$	$c_1 +$ $(a_1 + 1) \cdot (b_1 + 1)$
Party 1's Output (c_1):	c_1	c_1	c_1	c_1
$c_1 + c_2$	$a_1 b_1$	$a_1 \cdot (b_1 + 1)$	$(a_1 + 1) \cdot b_1$	$(a_1 + 1) \cdot (b_1 + 1)$
$(a_1 + a_2) \cdot (b_1 + b_2)$	$a_1 b_1$	$a_1 \cdot (b_1 + 1)$	$(a_1 + 1) \cdot b_1$	$(a_1 + 1) \cdot (b_1 + 1)$

Table 2.2. Truth table for privately computing $c_1 + c_2 = (a_1 + a_2) \cdot (b_1 + b_2)$.

$\mathcal{B}$. For example, $\mathcal{B}$ can include all the functions $f : \{0,1\}^* \times \{0,1\}^* \mapsto \{0,1\}$ (two-party case as an example). The bits of the input are entered into input wires and are propagated through the gates. Yao's protocol works by having one of the parties (Alice for example) first generates an "encrypted" or "garbled" circuit computing f and send its representation to the other party (Bob for example). In order for Bob to obtain the garbled values of the input wires, both Alice and Bob engage, for each input wire, in a 1-out-of-2 oblivious transfer. As a result of the oblivious transfer, Bob learns the garbled value of his input bit and nothing about the garbled value of the other bit, and Alice learns nothing. Now Bob has sufficient information to compute the output of the circuit on his own. After computing f, he can send this value to Alice if she requires it. Generally speaking, Yao's protocol is inherently inefficient because it uses a circuit representation of the function. The computational complexity of the protocol is roughly linear in relation to the size of Bob's input. To be more specific, the oblivious transfer stage requires one exponentiation per bit of Bob's input. The communication complexity is linear in relation to the size of the circuit. More accurately, a table of about 320-512 bits is generated and communicated for every gate (assuming that all gates have two inputs and one output). For

more detailed analysis about the complexity, please refer to Pinkas's work [46].

- **Homomorphic Encryption** A public-key cryptosystem $\mathcal{P}(G, E, D)$ is a collection of probabilistic polynomial time algorithms for key generation, encryption and decryption. The key generation algorithm G produces a private key sk and public key pk with specified key size. Anybody can encrypt a message with the public key, but only the holder of a private key can actually decrypt the message and read it. The encryption algorithm E takes as an input a plaintext m, a random value r and a public key pk and outputs the corresponding ciphertext $E_{pk}(m, r)$. The decryption algorithm D takes as an input a ciphertext c and a private key sk (corresponding to the public key pk) and outputs a plaintext $D_{sk}(c)$. It is required that $D_{sk}(E_{pk}(m, r)) = m$. The plaintext is usually assumed to be from $\mathbb{Z}_\mu$, [2] where μ is the product of two large primes. A public-key cryptosystem is homomorphic when

$$\begin{aligned} & \forall m_1, m_2, r_1, r_2 \in \mathbb{Z}_\mu, \\ D_{sk}(E_{pk}(m_1, r_1)E_{sk}(m_2, r_2) \bmod \mu^2) &= m_1 + m_2 \bmod \mu; \\ D_{sk}(E_{pk}(m_1, r_1)^{m_2} \bmod \mu^2) &= m_1 m_2 \bmod \mu; \\ D_{sk}(E_{pk}(m_2, r_2)^{m_1} \bmod \mu^2) &= m_1 m_2 \bmod \mu. \end{aligned}$$

This feature allows a party to add or multiply plaintexts by doing simple computations with ciphertexts, without having the secret key. Several homomorphic cryptosystems (*e.g.*, [47, 48]) in the literature are proved to be secure under reasonable complexity assumptions.

A natural application of homomorphic encryption is private inner product computa-

[2] The integers modulo μ, denoted $\mathbb{Z}_\mu$, is the set of (equivalence classes of) integers $\{0, 1, \ldots, \mu - 1\}$. Addition, subtraction, and multiplication in $\mathbb{Z}_\mu$ are performed modulo μ.

tion. It considers the problem of computing the inner product of two vectors owned by two different parties (Alice and Bob for example), respectively, so that neither party should learn anything beyond what is implied by the party's own vector and the output of the computation. Here the output for a party is either the inner product or nothing, depending on what the party is supposed to learn. The algorithm described in 2.1.2.2 was proposed by Goethals *et al.* [49]. It is directly based on homomorphic encryption and has been proved to be private in a strong sense. To be more specific, no probabilistic polynomial time algorithm substituting one party can obtain a non-negligible amount of information about the other party's private input, except what can be deduced from the input and output of this party.

Algorithm 2.1.2.2 Private Inner Product

Private Input of Alice: Vector $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{Z}_\mu^d$
Private Input of Bob: Vector $\mathbf{y} = (y_1, \ldots, y_d) \in \mathbb{Z}_\mu^d$
Output of Alice: $\mathbf{x} \cdot \mathbf{y} \bmod \mu$

1: Alice generates a private and public key pair (sk, pk), and sends pk to Bob.
2: For each $i, i = 1, \ldots d$, Alice generates a random number $r_i \in Z_\mu$, and sends $c_i = E_{pk}(x_i, r_i)$ to Bob.
3: Bob computes $w = \prod_{i=1}^{d} c_i^{y_i} \bmod \mu^2$ and sends w back to Alice.
4: Alice computes $\mathbf{x} \cdot \mathbf{y} \bmod \mu = D_{sk}(w)$.

For the sake of completeness, we note that many private inner product protocols have been proposed in the literature. Generally speaking, these protocols can be classified into two categories: 1) cryptosystem-based approaches, which offer strong privacy protection, but incur high communication and computational cost (*e.g.*, [50]) and 2) data perturbation-based approaches, which provide weaker privacy protection but allow more efficient solutions for more complicated data mining tasks (*e.g.*, [51]). We refer interested readers to [49] for an overview on this topic.

- **Commutative Encryption** Simply speaking, a commutative encryption is a pair of

encryption function f and g such that $f(g(x)) = g(f(x))$. To be more concrete, we borrow the definition used in [52].

Definition 2.1.1 (Commutative Encryption) *A commutative encryption $\mathcal{F}$ is a computable polynomial time function $f : Key\ \mathcal{F} \mapsto Dom\ \mathcal{F}$, defined on finite computable domains, and satisfying all properties listed below. We denote $f_e(x) \equiv f(e, x)$, and use "$\in_r$" to mean "is chosen uniformly at random from."*

1. *Commutativity: For all $e, e' \in Key\ \mathcal{F}$, we have $f_e \circ f'_e = f'_e \circ f_e$.*
2. *Each $f_e : Dom\ \mathcal{F} \mapsto Dom\ \mathcal{F}$ is a bijection.*
3. *The inverse f_e^{-1} is also computable in polynomial time given e.*
4. *The distribution of $< x, f_e(x), y, f_e(y) >$ is computationally indistinguishable from the distribution $< x, f_e(x), y, z >$, where $x, y, z \in_r Dom\ \mathcal{F}$ and $e \in_r Key\ \mathcal{F}$.*

Property 1 says that the composition of the encryption with two different keys is the same irrespective of the order of encryption. Property 2 says that two different values will never have the same encrypted value. Property 3 says that given an encrypted value $f_e(x)$ and the encryption key e, we can find x in polynomial time. Property 4 says that given a value x and its encryption $f_e(x)$ (but not the key e) and a new value y, we cannot distinguish between $f_e(y)$ and a random value z in polynomial time. Thus we cannot encrypt y or decrypt $f_e(y)$ in polynomial time.

As an example, let $Dom\ \mathcal{F}$ be all quadratic residues modulo p, where p is a safe prime number, *i.e.*, both p and $q = (p-1)/2$ are primes. Let $Key\ \mathcal{F}$ be $\{1, 2, \ldots, q-1\}$. Then assuming the Decisional Diffie-Hellman hypothesis (DDH), the power function

$$f_e(x) \equiv x^e \bmod p$$

is a commutative encryption because

$$f_e(f_d(x)) = (x^d \bmod p)^e \bmod p = x^{de} \text{ mop } p = (x^e \bmod p)^d \bmod p = f_d(f_e(x)).$$

Based on commutative encryption, Agrawal *et al.* [52] developed several secure protocols for set intersection, equijoin, intersection size, and equijoin size. We refer interested readers to their work for more details.

Related Work The work in [45] detailed a rigorous introduction to SMC and cryptographic protocols. It has shown that any polynomial-time function can be expressed as a combinatorial circuit of polynomial size, and is therefore privately computable using a generic circuit evaluation protocol. However, the communication and computational complexity of doing so makes this general approach infeasible for large datasets. As a result, many new, more efficient SMC techniques are being developed by exploring a combination of different approaches such as data perturbation, linear transformation, and cryptographic primitives. The work in [46] offered a broad view of SMC framework and its applications to data mining. A collection of SMC tools useful for privacy preserving data mining (*e.g.*, secure sum, set union, inner product) were discussed in [53]. Several privacy preserving data mining algorithms have been developed based on these tools, *e.g.*, association rule mining from vertically partitioned data [54] and horizontally partitioned data [55], clustering with distributed EM mixture modeling [56], and K-Means clustering over vertically partitioned data [57]. A detailed overview of these techniques and applications can be found in [58]. SMC and cryptographic protocols have also been applied for statistical analysis [51], support vector machine [59], naive Bayes classification [60], privacy preserving OLAP [61], Bayesian network structure computation [50], information sharing across private databases [52], privacy preserving distributed decision tree induction [62] and many others.

2.1.3 Distributed Data Mining (DDM)

Bluntly put, distributed data mining (DDM) is data mining where the data and computation are spread over many independent sites. For some applications, the distributed setting is more natural than the centralized one because the data is inherently distributed. The bulk of DDM methods in the literature operate over an abstract architecture where each site has a private memory containing its own portion of the data. The sites can operate independently and communicate by message-passing over an asynchronous network. Typically, communication is a bottleneck. Because communication is assumed to be carried out exclusively by message-passing, a primary goal of many methods in the literature is to minimize the number of messages sent. For more information about DDM, the reader is referred to two recent surveys [63,64]. These provide a broad overview of DDM, touching on issues such as: clustering, classification, association rule mining, Bayesian network learning, basic statistics computation, and the historical roots of DDM. An online repository for DDM related publications can be found at [65].

Since DDM produces a global data mining model by exchanging only a small amount of information among the participating sites, it has been adopted for many distributed privacy preserving data mining scenarios. The work in [66] proposed a paradigm for clustering distributed privacy sensitive data in an unsupervised or a semi-supervised scenario. In this algorithm, each local data site builds a model and transmits only the parameters of the model to the central site where a global clustering model is constructed. A distributed privacy preserving algorithm for Bayesian network parameter learning is reported elsewhere [67].

2.2 Rule Hidning

The main objective of rule hiding is to transform the database such that the sensitive rules, for example, associate rules and classification rules, are masked, and all the other underlying patterns can still be discovered.

2.2.1 Association Rule Hiding

Association rule hiding considers the problem of transforming the database so that all the sensitive association rules are concealed and other non-sensitive rules can still be identified. The work in [68] gave a formal proof that finding an optimal solution to hide sensitive large item sets is an NP-hard problem. For this reason, many heuristic approaches have been proposed to address the complexity issues. For example, the perturbation-based association rule hiding techniques [39, 69] are implemented by changing a selected set of 1-values to 0-values (in a binary database) or vice versa so that the frequent item sets that generate the sensitive rules are hidden or the support of sensitive rules is lowered to a user-specified threshold. The blocking-based association rule hiding approach [70] replaces certain attributes of the data with a question mark. The introduction of this new special value in the dataset imposes some changes on the definition of the support and confidence of an association rule. In this regard, the minimum support and minimum confidence will be changed into a minimum support interval and a minimum confidence interval. As long as the support and/or the confidence of a sensitive rule lies below the middle in these two ranges, the confidentiality of data is expected to be protected.

2.2.2 Classification Rule Hiding

The work in [71] presented a framework that combines decision tree classification and parsimonious downgrading. Here the term “parsimonious downgrading” refers to the

phenomenon of trimming out sensitive information from a dataset when it is transferred from a secure environment (referred to as high) to a public domain (referred to as low). The objective of this work is to guarantee that the receiver of the data will be unable to build informative classification models for the data that is not downgraded.

2.3 Summary

Data mining technologies have enabled commercial and governmental organizations to extract useful knowledge from data for the purpose of business and security related applications. While successful applications are encouraging, there are increasing concerns about the invasions to the privacy of personal information. To address these concerns, researchers in the data mining community have proposed various solutions. This chapter presents an overview of them. It has noted that the main consideration in privacy preserving data mining is two fold: 1) *data hiding*: sensitive raw data should be modified or trimmed out from the original database while the important underlying patterns of the data should still be preserved; and 2) *rule hiding*: sensitive knowledge which can be discovered from the data should be filtered out. We refer interested readers to a recent book, a survey and an online bibliography [58, 72, 73] for more information about this booming research area.

Chapter 3

TRADITIONAL MULTIPLICATIVE DATA PERTURBATION

A statistical database (SDB) system is a database system that allows its users to retrieve aggregate statistics (*e.g.*, sample mean and variance) for a subset of the entities represented in the database and prevents the collection of information on specific individuals. In the statistics community, there has been extensive research on the problem of securing SDBs against disclosure of confidential information. This is generally referred to as *statistical disclosure control*. Statistical disclosure control approaches suggested in the literature are classified into four general groups: conceptual, query restriction, output perturbation and data perturbation [8]. The conceptual approach provides a framework for better understanding and investigating the security problem of statistical database at the conceptual data model level. It does not provide a specific implementation procedure. The query restriction approach offers protection by either restricting the size of query set or controlling the overlap among successive queries, etc. The output perturbation approach perturbs the answer to user queries while leaving the data in the database unchanged. The data perturbation approach introduces noise into the database and transforms it into another version. This dissertation primarily focuses on the data perturbation approach, and we refer interested readers to [8] for more details about other approaches.

Adding random noise to the private database is one common data perturbation approach. In this case, a random noise term is generated from a prescribed distribution, and the perturbed value takes the form: $y_{ij} = x_{ij} + r_{ij}$, where x_{ij} is the i-th attribute of the j-th private data record, and r_{ij} is the corresponding random noise. In the statistics community, this approach was primarily used to provide summary statistical information (*e.g.*, sum, mean, variance, etc.) without disclosing individuals' confidential data (*e.g.*, [74]). In the privacy preserving data mining area, this approach was considered in [4, 11] for building decision tree classifiers from private data. Recently, many researchers have pointed out that additive noise can be easily filtered out in many cases that may lead to compromising the privacy [5–7]. Given the large body of existing signal-processing literature on filtering random additive noise, the utility of random additive noise for privacy-preserving data mining is not quite clear.

The possible drawback of additive noise makes one wonder about the possibility of using multiplicative noise (*i.e.*, $y_{ij} = x_{ij} * r_{ij}$) for protecting the privacy of the data. Two basic forms of multiplicative noise have been well studied in the statistics community [15]. One multiplies each data element by a random number that has a truncated Gaussian distribution with mean one and small variance. The other takes a logarithmic transformation of the data first, adds multivariate Gaussian noise, then takes the exponential function *exp(.)* of the noise-added data. As noted in [15], the former perturbation scheme was once used by the Energy Information Administration in the U.S. Department of Energy to mask the heating and cooling degree days, denoted by x_{ij}. A random noise r_{ij} is generated from a Gaussian distribution with mean 1 and variance 0.0225. The random noise is further truncated such that the resulting number r_{ij} satisfies $0.01 \leq |r_{ij} - 1| \leq 0.6$. The perturbed data $x_{ij}r_{ij}$ were released. This approach was also discussed in [75].

This chapter gives a brief review of these two perturbation schemes.

3.1 Perturbation Scheme I

3.1.1 Perturbation Scheme

Let x_i be the i-th attribute of a private database. Let x_{ij} be the value for the i-th attribute of the j-th record in the database, $i = 1, \ldots, n$, $j = 1, \ldots, m$. Let r_{ij} denote the random noise corresponding to x_{ij}. The perturbed data y_{ij} is

$$y_{ij} = x_{ij} r_{ij},$$

where r_{ij} is independent and identically chosen from a Gaussian distribution with mean μ_i (usually $\mu_i = 1$) and variance σ_i^2. In other words, all r_{ij}'s for a given i follow the same distribution. In practice, the probability density of noise r (ignoring the subscript) is usually doubly truncated as follows:

$$\begin{aligned} f(r) &= \frac{\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(r-\mu)^2)}{\frac{1}{\sqrt{2\pi}\sigma} \int_A^B \exp(-\frac{1}{2\sigma^2}(r-\mu)^2) dr} \quad \text{for } A < r < B. \\ &= \frac{\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(r-\mu)^2)}{\Phi(\frac{B-\mu}{\sigma}) - \Phi(\frac{A-\mu}{\sigma})}, \end{aligned}$$

where A and B are the lower and upper truncation bounds and $\Phi(A)$ stands for the cumulative probability up to A. The above equation can be further simplified as

$$K \mathrm{Z}(\frac{r-\mu}{\sigma}),$$

where $K = \frac{1}{\Phi(\frac{B-\mu}{\sigma}) - \Phi(\frac{A-\mu}{\sigma})}$, and $\mathrm{Z}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2}x^2)$.

3.1.2 Statistical Properties of the Perturbed Data

It has been proved in [15] that the mean and variance of the original data attributes can be estimated from the mean and variance of the perturbed data.

Mean of x_i:

$$E(x_i) = \frac{E(y_i)}{\mu_i + K[\mathrm{Z}(\frac{A-\mu_i}{\sigma_i}) - \mathrm{Z}(\frac{B-\mu_i}{\sigma_i})]}. \tag{3.1}$$

Because the data owner will release μ_i, σ_i, A and B, the data receiver can compute the expected value of x_i.

Variance of x_i:

$$Var(x_i) = E(x_i^2) - (E(x_i))^2, \tag{3.2}$$

where $E(x_i)$ can be easily calculated following Eq. 3.1, and $(E(x_i))^2$ can be computed from the follow equations:

$$\begin{aligned}
Var[y_i] &= E(x_i^2)E(r_i^2) - (E(x_i)E(r_i))^2 \\
&= E(x_i^2)\{\sigma_i^2 + \mu_i^2 + \sigma_i^2 K[\frac{A-\mu_i}{\sigma_i}\mathrm{Z}(\frac{A-\mu_i}{\sigma_i}) - \frac{B-\mu_i}{\sigma_i}\mathrm{Z}(\frac{B-\mu_i}{\sigma_i})] \\
&\quad +2\sigma_i\mu_i K[\mathrm{Z}(\frac{A-\mu_i}{\sigma_i}) - \mathrm{Z}(\frac{B-\mu_i}{\sigma_i})]\} \\
&\quad -(E(x_i))^2\{\mu_i^2 + \sigma_i^2 K^2[\mathrm{Z}(\frac{A-\mu_i}{\sigma_i}) - \mathrm{Z}(\frac{B-\mu_i}{\sigma_i})]^2 \\
&\quad +2\sigma_i\mu_i K[\mathrm{Z}(\frac{A-\mu_i}{\sigma_i}) - \mathrm{Z}(\frac{B-\mu_i}{\sigma_i})]\}.
\end{aligned}$$

Although the original attribute's mean and variance can be estimated from the perturbed data, the inner product and Euclidean distance among the data records are not necessarily preserved after perturbation. The following lemmas depict this situation.

Lemma 3.1.1 *Let $y_{ij} = x_{ij}r_{ij}$, where each r_{ij} is independent and identically chosen from a Gaussian distribution with mean 1 and variance σ^2. Then*

$$\begin{aligned} E(\sum_{i=1}^{n} y_{ij}y_{ik} - \sum_{i=1}^{n} x_{ij}x_{ik}) &= 0; \\ Var(\sum_{i=1}^{n} y_{ij}y_{ik} - \sum_{i=1}^{n} x_{ij}x_{ik}) &= \sigma^2 \sum_{i=1}^{n} x_{ij}^2 x_{ik}^2. \end{aligned}$$

Proof:

$$\begin{aligned} E(\sum_{i=1}^{n} y_{ij}y_{ik} - \sum_{i=1}^{n} x_{ij}x_{ik}) &= E(\sum_{i=1}^{n} x_{ij}r_{ij}x_{ik}r_{ik}) - \sum_{i=1}^{n} x_{ij}x_{ik} \\ &= \sum_{i=1}^{n} E(x_{ij}r_{ij}x_{ik}r_{ik}) - \sum_{i=1}^{n} x_{ij}x_{ik} \\ &= \sum_{i=1}^{n} x_{ij}E(r_{ij})x_{ik}E(r_{ik}) - \sum_{i=1}^{n} x_{ij}x_{ik} \\ &= 0. \\ Var(\sum_{i=1}^{n} y_{ij}y_{ik} - \sum_{i=1}^{n} x_{ij}x_{ik}) &= Var(\sum_{i=1}^{n} x_{ij}r_{ij}x_{ik}r_{ik}) \\ &= \sum_{i=1}^{n} Var(x_{ij}r_{ij}x_{ik}r_{ik}) + \\ &\quad 2\sum_{p=1}^{n-1}\sum_{q=p+1}^{n} Cov(x_{pj}r_{pj}x_{pk}r_{pk}, x_{qj}r_{qj}x_{qk}r_{qk}) \\ &= \sum_{i=1}^{n} Var(x_{ij}r_{ij}x_{ik}r_{ik}) \\ &= \sum_{i=1}^{n} \{E(x_{ij}^2 r_{ij}^2 x_{ik}^2 r_{ik}^2) - (E(x_{ij}r_{ij}x_{ik}r_{ik}))^2\} \\ &= \sum_{i=1}^{n} \{(1+\sigma^2)x_{ij}^2 x_{ik}^2 - x_{ij}^2 x_{ik}^2\} \\ &= \sigma^2 \sum_{i=1}^{n} x_{ij}^2 x_{ik}^2. \end{aligned}$$

□

The above lemma shows that although the inner product is preserved on expectation, the variance of the error could be very large.

Lemma 3.1.2 *Let $y_{ij} = x_{ij}r_{ij}$, where each r_{ij} is independent and identically chosen from a Gaussian distribution with mean 1 and variance σ^2. Then*

$$E(\sum_{i=1}^{n}(y_{ij} - y_{ik})^2 - \sum_{i=1}^{n}(x_{ij} - x_{ik})^2) \quad = \quad \sum_{i=1}^{n}\sigma^2(x_{ij}^2 + x_{ik}^2).$$

Proof: Let LHS denotes the left hand side of the above equation. We have

$$\begin{aligned}
LHS &= E(\sum_{i=1}^{n}(x_{ij}r_{ij} - x_{ik}r_{ik})^2) - \sum_{i=1}^{n}(x_{ij} - x_{ik})^2 \\
&= E(\sum_{i=1}^{n}(x_{ij}^2r_{ij}^2 + x_{ik}^2r_{ik}^2 - 2x_{ij}r_{ij}x_{ik}r_{ik})) - \sum_{i=1}^{n}(x_{ij} - x_{ik})^2 \\
&= \sum_{i=1}^{n}((1+\sigma^2)x_{ij}^2 + (1+\sigma^2)x_{ik}^2 - 2x_{ij}x_{ik}) - \sum_{i=1}^{n}(x_{ij} - x_{ik})^2 \\
&= \sum_{i=1}^{n}((x_{ij} - x_{ik})^2 + \sigma^2(x_{ij}^2 + x_{ik}^2)) - \sum_{i=1}^{n}(x_{ij} - x_{ik})^2 \\
&= \sum_{i=1}^{n}(x_{ij} - x_{ik})^2 + \sum_{i=1}^{n}\sigma^2(x_{ij}^2 + x_{ik}^2) - \sum_{i=1}^{n}(x_{ij} - x_{ik})^2 \\
&= \sum_{i=1}^{n}\sigma^2(x_{ij}^2 + x_{ik}^2).
\end{aligned}$$

□

The above lemma shows that the Euclidean distance is not preserved after perturbation.

3.2 Perturbation Scheme II

3.2.1 Perturbation Scheme

Let x_{ij} be the value for the i-th attribute of the j-th record in the database as before. $i = 1, \ldots, n$, $j = 1, \ldots, m$. Let

$$u_{ij} = \ln x_{ij}.$$

We generate the random noise following the multivariate Gaussian distribution $N(0, c\Sigma_U)$, where $0 < c < 1$ and Σ_U is the covariance matrix of variables $u_1, u_2, \ldots, u_n$. We denote the noise as e_{ij}. Let

$$\begin{aligned} z_{ij} &= u_{ij} + e_{ij}, \\ y_{ij} &= \exp(z_{ij}) \\ &= \exp(\ln x_{ij} + e_{ij}) \\ &= x_{ij} \exp(e_{ij}) \\ &= x_{ij} h_{ij}. \end{aligned}$$

The perturbed data y_{ij} is released then. Note that this scheme assumes that all x_{ij} are positive.

3.2.2 Statistical Properties of the Perturbed Data

It has been proved in [15] that the mean, variance and covariance of the original data attributes can be estimated from the perturbed data.

Mean of x_i: Let $\sigma_i^2 = cVar(\ln x_i)$. We have

$$E(x_i) = \frac{E(y_i)}{\exp(\frac{\sigma_i^2}{2})}. \quad (3.3)$$

Variance of x_i:

$$\begin{aligned} Var(x_i) &= E(x_i^2) - (E(x_i))^2 \\ &= \frac{Var(u_i)}{\exp(2\sigma_i^2)} - \frac{E(x_i)^2}{\exp(\sigma_i^2)} - (E(x_i))^2. \end{aligned} \quad (3.4)$$

Covariance of x_i and x_j:

$$Cov(x_i, x_j) = \{\frac{\sum_{k=1}^m y_{ik}y_{jk}}{\exp[(\sigma_i^2 + 2\rho\sigma_i\sigma_j + \sigma_j^2)/2]} - \frac{m\frac{\sum_{k=1}^m y_{ik}}{m}\frac{\sum_{k=1}^m y_{jk}}{m}}{\exp[\sigma_i^2 + \sigma_j^2]}\}/(m-1), \quad (3.5)$$

where ρ is the correlation coefficient of x_i and x_j, and it can be obtained from the perturbed data. Because the noise was generated to maintain the same correlation structure, the correlation between the perturbed data will be on average the same as that between the original data in log-scale.

Similar to perturbation scheme I, the inner product and Euclidean distance among the data records are not preserved after perturbation. The following lemma depicts this situation.

Lemma 3.2.1 *Let $y_{ij} = x_{ij}h_{ij}$, where x_{ij} and h_{ij} are defined as before. We have*

$$\begin{aligned} E(\sum_{i=1}^n y_{ij}y_{ik} - \sum_{i=1}^n x_{ij}x_{ik}) &= \sum_{i=1}^n x_{ij}x_{ik}e^{\sigma_i^2} - \sum_{i=1}^n x_{ij}x_{ik}; \\ E(\sum_{i=1}^n (y_{ij} - y_{ik})^2 - \sum_{i=1}^n (x_{ij} - x_{ik})^2) &= \sum_{i=1}^n (e^{2\sigma_i^2}(x_{ij}^2 + x_{ik}^2) - 2x_{ij}x_{ik}e^{\sigma_i^2}) - \sum_{i=1}^n (x_{ij} - x_{ik})^2. \end{aligned}$$

Proof: Because $h_i = \exp(e_i)$ and e_i follows a Gaussian distribution with mean 0 and

variance σ_i^2 (note that $\sigma_i^2 = cVar(\ln x_i)$), we can compute the mean and variance of h_i as follows.

$$\begin{aligned} E(h_i) &= \int_{-\infty}^{+\infty} e^x \frac{1}{\sqrt{2\pi}\sigma_i} e^{\frac{-x^2}{2\sigma_i^2}} dx \\ &= e^{\frac{\sigma_i^2}{2}}; \\ Var(h_i) &= \int_{-\infty}^{+\infty} (e^x - e^{\frac{\sigma_i^2}{2}})^2 \frac{1}{\sqrt{2\pi}\sigma_i} e^{\frac{-x^2}{2\sigma_i^2}} dx \\ &= e^{\sigma_i^2}(e^{\sigma_i^2} - 1); \\ E(h_i^2) &= (E(h_i))^2 + Var(h_i) = e^{2\sigma_i^2}. \end{aligned}$$

Applying the above results to the proofs of Lemma 3.1.1 and Lemma 3.1.2, we get the expected results. □

The above lemma shows that in scheme II, the perturbed data does not preserve either inner product or Euclidean distance.

3.3 Privacy Issues

On the surface, multiplicative perturbation seems to change the data more than additive perturbation. For example, perturbing a salary of $\$100,000$ by adding $\$5000$ (5% relative change) would be considered a compromise while at the same time perturbing a salary of $\$10,000$ by $\$5000$ (50% relative change) would preserve the privacy of the data. On the other hand, perturbing $\$100,000$ and $\$10,000$ by multiplying by 2 would be accepted because both have 100% relative change. However, by taking logarithms on the perturbed data, the multiplicative perturbation turns into an additive perturbation. More specifically, for perturbation scheme I, the logarithmic transformation of y_{ij} gives us $\ln x_{ij} + \ln r_{ij}$, where the noise term $\ln r_{ij}$ is chosen independent and identically from some distribution. For perturbation scheme II, after logarithmic transformation, we have $\ln x_{ij} + e_{ij}$. The noise

term is chosen from $N(0, c\Sigma_{\ln X})$, where $\Sigma_{\ln X}$ is the covariance of the original data in log scale. As noted in [5–7], the privacy of the former "additive perturbation scheme" can be easily breached in many cases. The latter "additive perturbation scheme" generates random noise with *similar* covariance structure with the original data (in log scale), and therefore offers better privacy protection. This kind of perturbation has also been extensively investigated in the literature (*e.g.*, [6, 13, 14, 76]). In particular, the work in [6] shows that the accuracy of attacker's estimation of the original data gets worse as the similarity increases.

Before concluding this subsection, it should be noted that, traditionally, the privacy, denoted by ρ, provided by a perturbation technique for continuous data is measured as the variance of difference between the original data and perturbed data [8], that is, $Var(X-Y)$, where X represents the original data attribute and Y the perturbed attribute. This measure can be made scale invariant with respect to the variance of X as $\rho = Var(X-Y)/Var(X)$. This measure is suited to quantifying the privacy of a single attribute. In practice, an attacker may also attempt to use a linear combination of the perturbed attributes to estimate confidential information of the linear combination of the original attributes. Measuring the privacy offered for linear combinations is difficult because there are too many such combinations. A canonical correlation-based metric is used in [13] that can measure the maximum proportion of variance that an attacker can explain for any linear combination of the original attributes, using a linear combination of the perturbed and non-confidential attributes. Let λ denotes the largest eigenvalue of the following matrix $C_{XX}^{-1}C_{XY}C_{YY}^{-1}C_{YX}$, where C_{XX} denotes the covariance of X, C_{XY} the covariance of X and Y. The value of λ represents the maximum proportion of variability in any linear combination of X that can be explained by any linear combination of Y. The privacy is defined as $\rho = 1 - \lambda$. Thus, for any linear combination of X, at least $1 - \lambda$ proportion of variability will remain unexplained. These metrics do provide the data owner with meaningful information regarding the effectiveness of the perturbation method in some way. However, they do not

offer an insight on how the attackers could attack the perturbation if they had some prior knowledge about the data. Trottini *et al.* [77] tried to address this issue by developing a Bayesian attacker model to assess the performance of the perturbation techniques on continuous microdata. They specifically investigated the combination of both additive noise and multiplicative noise and allowed the attacker to use external data to enhance the chances of disclosing the identity of a target individual. Their simulation showed that the probability of the identity disclosure is a function of many key parameters like the variability amongst profiles in the original data, the amount of attacker's prior information, the amount of noise introduced in the data, etc.

3.4 Summary

This chapter briefly reviews two traditional multiplicative data perturbation techniques that have been well studied in the statistics community. These perturbations are primarily used to mask the private data while allowing summary statistics (*e.g.*, sum, mean, variance, covariance) of the original data to be estimated.

In summary, these multiplicative perturbations have the following advantages and disadvantages:

- The multiplicative perturbation is relative, that is, large values in the original data are perturbed more than smaller values.

- In practice, the first perturbation scheme is good if the data disseminator only wants to make minor changes to the original data; the second scheme assures higher security than the first one but maintains the data utility in the log-scale.

- These perturbation schemes are equivalent to additive perturbation after the logarithmic transformation. Due to the large volume of research in deriving private in-

formation from the additive noise perturbed data, the security of these perturbation schemes is questionable.

- The objective of these perturbation schemes is to mask the private data while allowing summary statistics to be estimated. However, problems in data mining are somewhat different. Data mining techniques, such as clustering, classification, prediction and association rule mining, are essentially relying on more sophisticated relationships among data records or data attributes, but not simple summary statistics. The traditional multiplicative perturbations distort each data element independently, therefore the Euclidean distance and inner product among data records are usually not preserved, and the perturbed data cannot be used for many data mining applications.

In the next chapter, we will present a new multiplicative data perturbation technique called *distance preserving data perturbation*. This technique preserves inner product and Euclidean distance among data records. Therefore, many data mining algorithms can be *efficiently* applied to the perturbed data and produce *exactly the same* results as if applied to the original data (*e.g.*, distance-based clustering, k-nearest neighbor classification). We further address the privacy issues of this technique by considering three types of prior knowledge an attacker may have and use to design attack techniques to recover the original data. As such, valuable information is gained into the effectiveness of distance preserving transformation for privacy preserving data mining.

Chapter 4

EUCLIDEAN DISTANCE PRESERVING DATA PERTURBATION

Recently, distance preserving data perturbation [16–18] has gained attention because it mitigates the privacy/accuracy trade-off by guaranteeing perfect accuracy. Many important data mining algorithms can be *efficiently* applied to the transformed data and produce *exactly the same* results as if applied to the original data. *e.g.*, distance-based clustering and k-nearest neighbor classification. However, the issue of how well the original data is hidden has, to our knowledge, not been carefully studied. In this chapter, we address this issue by studying how well an attacker can recover the original data from the transformed data and prior information. We restrict our attention to the class of distance preserving transformations that fix the origin and consider recovery of the original data in the presence of three different classes of prior information (described later). Our analysis explicitly illuminates scenarios where privacy can be breached. As such, valuable information is gained into the effectiveness of distance preserving transformation for privacy preserving data mining.

The remainder of this chapter is organized as follows. Section 4.1 discusses some basic mathematical properties of distance preserving transformations, the application of these transformations to privacy preserving data mining, and the generation of orthogonal matrices. Sections 4.2 and 4.3 defines the privacy breach metric and three classes of attacker's

prior knowledge. Sections 4.4, 4.5 and 4.6 examine in detail how knowledge in each of these classes can be used to estimate the original data from the transformed data. Finally, Section 4.7 concludes this chapter.

4.1 Distance Preserving Transformations

This section offers an overview of distance preserving transformation: its definition, application scenarios, etc. Throughout this chapter (unless otherwise stated), all matrices and vectors discussed are assumed to have real entries. All vectors are assumed to be column vectors and M' denotes the transpose of any matrix M. An $m \times n$ matrix M is said to be orthogonal if $M'M = I_n$, the $n \times n$ identity matrix. If M is square, it is orthogonal if and only if $M' = M^{-1}$ [78, pg. 17]. The determinant of any orthogonal matrix is either $+1$ or -1. Let $\mathbb{O}_n$ denote the set of all $n \times n$, orthogonal matrices.

4.1.1 Definition and Fundamental Properties

To define the distance preserving transformation, let us start with the definition of *metric space*. In mathematics, a metric space is a set $\mathcal{S}$ with a global distance function (the metric d) that, for every two points x, y in $\mathcal{S}$, gives the distance between them as a nonnegative real number $d(x, y)$. Usually, we denote a metric space by a 2-tuple $(\mathcal{S}, d)$. A metric space must also satisfy

1. $d(x, y) = 0$ iff $x = y$ (identity),

2. $d(x, y) = d(y, x)$ (symmetry),

3. $d(x, y) + d(y, z) \geq d(x, z)$ (triangle inequality).

A metric space $(\mathcal{S}_1, d_1)$ is isometric to a metric space $(\mathcal{S}_2, d_2)$ if there is a bijection $T : \mathcal{S}_1 \to \mathcal{S}_2$ that preserves distances. That is, $d_1(x, y) = d_2(T(x), T(y))$ for all $x, y \in \mathcal{S}_1$.

The metric space which most closely corresponds to our intuitive understanding of space is the Euclidean space, where the distance d between two points is the length of the straight line connecting them. In this chapter, we specifically consider the Euclidean space and define $d(x, y) = ||x - y||$, the l^2-norm of vector $x - y$. A function $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is distance preserving in the Euclidean space if for all $x, y \in \mathbb{R}^n$, $||x - y|| = ||T(x) - T(y)||$. Here T is also called a *rigid motion*. It has been shown that any distance preserving transformation is equivalent to an orthogonal transformation followed by a translation [78, pg. 128]. In other words, there exists $M_T \in \mathbb{O}_n$ and $v_T \in \mathbb{R}^n$ such that T equals $x \in \mathbb{R}^n \mapsto M_T x + v_T$. If T fixes the origin, $T(0) = 0$, then $v_T = 0$; hence, T is an orthogonal transformation. Henceforth we assume T is a distance preserving transformation which fixes the origin – an *orthogonal transformation*. Such transformations preserve the length (l^2-norm) of vectors: $||x|| = ||T(x)||$ (*i.e.*, given any $M_T \in \mathbb{O}_n$, $||x|| = ||M_T x||$). Hence, they move x along the surface of the hyper-sphere centered at the origin with radius $||x||$.

From a geometric perspective, an orthogonal transformation is either a rigid rotation or a rotoinversion (a rotation followed by a reflection). This property was originally discovered by Schoute in 1891 [79]. Coxeter [80] summarized Schoute's work and proved that every orthogonal transformation can be expressed as a product of commutative rotations and reflections. To be more specific, let Q denote a rotation, R a reflection, $2q$ the number of conjugate imaginary eigenvalues of the orthogonal matrix M, and r the number of (-1)'s in the $n - 2q$ real eigenvalues. The orthogonal transformation is expressible as $Q^q R^r (2q + r \leq n)$. Especially, in 2D space, $det(M) = 1$ corresponds to a rotation, while $det(M) = -1$ represents a reflection.

4.1.2 Generation of Orthogonal Matrix

Many matrix decompositions involve orthogonal matrices, such as QR decomposition, SVD, spectral decomposition and polar decomposition. To generate a uniformly distributed

random orthogonal matrix, we usually fill a matrix with independent Gaussian random entries, then use QR decomposition. Stewart [81] replaced this with a more efficient idea that Diaconis and Shahshahani [82] later generalized as the *subgroup algorithm*. We refer the reader to these references for detailed treatment of this subject.

4.1.3 Data Perturbation Model

Orthogonal transformation-based data perturbation can be implemented as follows. Suppose the data owner has a private database $X_{n\times m}$, with each column of X being a record and each row an attribute. The data owner generates an $n \times n$ orthogonal matrix M_T, and computes

$$Y_{n\times m} = M_{T_{n\times n}} X_{n\times m}. \tag{4.1}$$

The perturbed data $Y_{n\times m}$ is then released for future usage. As a taste of the many examples and experiments to come later in this Chapter, Figure 4.1 provides an example of how the data looks before and after perturbation.

Next we describe the privacy application scenarios where orthogonal transformation can be used to hide the data while allowing important patterns to be discovered *without error*.

4.1.4 Privacy Application Scenarios

Many data perturbation approaches pay a price in terms of the accuracy of the estimated patterns for achieving the desired level of privacy protection. For example, an additive perturbation-based approach adds noise to the data in order to make sure that the data is sufficiently distorted so that the original data values cannot be identified accurately. This also introduces noise in the patterns (*e.g.*, a decision tree, association rules) that a

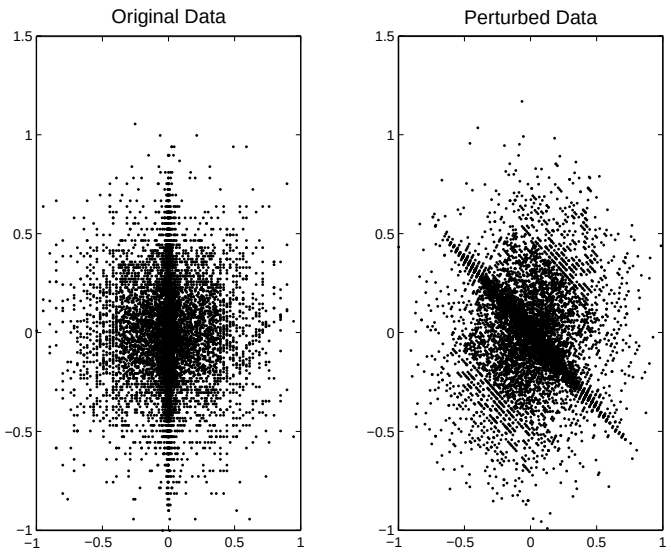


FIG. 4.1. An example of distance preserving data perturbation (with origin fixed) in 2D space.

data miner may be interested in computing. However, there are many application domains (*e.g.*, security, counter-terrorism) where losing accuracy for privacy may not be acceptable. Detecting outlier activities from a large amount of data may require highly precise data analysis capabilities. After all, we do not want the perpetrators of criminal activities to enjoy the privacy-shield offered to the law abiding individuals.

Orthogonal transformation has a nice property that it preserves vector inner product and distance in Euclidean space. Therefore, any data mining algorithms that rely on inner product or Euclidean distance as a similarity criteria are invariant to orthogonal transformation. Put in other words, many data mining algorithms can be applied to the transformed data and produce exactly the same results as if applied to the original data, *e.g.*, KNN classifier, perceptron learning, support vector machine, distance-based clustering and outlier detection. We refer the reader to [18] for a simple proof of rotation-invariant classifiers.

In practice, orthogonal transformation-based data perturbation is particularly geared towards the following privacy application scenarios:

Census Scenario (see Figure 4.2(a)) An organization has a private dataset X (each column is a data record) and wishes to make it publicly available for data analysis while keeping the original data records private. To accomplish this, $Y = M_T X$ is released to the public. The distance preserving nature of T allows a public entity to easily recovery many useful patterns from Y. For example, the cluster membership produced by a Euclidean distance-based K-means clustering on Y will be exactly the same as that produced on X. This model is widely studied in the field of security control for statistical databases. We refer the reader to [8] for an overview of this topic.

Storage Outsourcing Scenario (see Figure 4.2(b)) An organization continuously generates private data records, but does not wish to invest in the infrastructure (both personnel and hardware) needed to manage the storage. Outsourcing this job can be an attractive alternative, *i.e.,* the data records are handed over to an outside agency that manages their storage. However, the original data records are sensitive and the organization would rather avoid releasing them in the plain to the outsourcing agency. To accomplish this, the owner applies T to each data record and releases the results to the outsourcing agency. Whenever the owner wishes to retrieve records from the outsourced database, she or he transforms the query by the same T and sends it to the outsourcing agency who carries out similarity comparison on the data and, in turn, sends the results back to the owner. This scenario is closely related to work on secure database outsourcing, *e.g.,* [83].

4.2 Privacy Breach

Orthogonal transformation-based data perturbation has the nice property that many data mining algorithms can be applied to the perturbed data and produce exactly the same results as if applied to the original data. However, the issue of how well the original data is

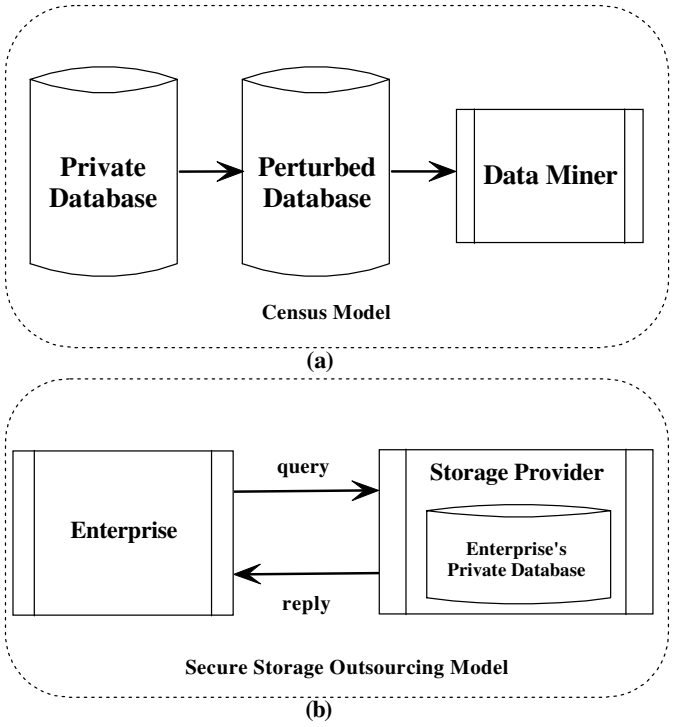


FIG. 4.2. Privacy application scenarios where orthogonal transformation can be used to hide the data while allowing important patterns to be discovered without error.

hidden has, to our knowledge, not been carefully studied. We take a step in this direction by assuming the role of an attacker armed with three types of prior information regarding the original data. We examine how well the attacker can recover the original data from the perturbed data and prior information.

Before stepping into the details of the attack algorithms, we first give the definition of *privacy breach*. We assume that an attacker will have X and Y and that Y was produced from X by an orthogonal transformation. The attacker will also have prior knowledge as described in Section 4.3. The attacker will produce $\hat{x} \in \mathbb{R}^n$ and $1 \leq \hat{i} \leq m$, where $\hat{x}$ is the attacker's estimate of $x_{\hat{i}}$, the $\hat{i}^{th}$ data tuple (column) in X.

Definition 4.2.1 (ϵ-Privacy Breach) *For any $\epsilon > 0$, we say that an ϵ-privacy breach occurs if $||\hat{x} - x_{\hat{i}}|| \leq ||x_{\hat{i}}||\epsilon$.*

Informally stated, an ϵ-privacy breach occurs if the attacker's estimate is wrong with relative error no more than ϵ. We further define the probability of privacy breach as follows:

Definition 4.2.2 (Probability of ϵ-Privacy Breach) *We define $\rho(x_{\hat{i}}, \epsilon)$ as the probability that an ϵ-privacy breach occurs given that the attacker chose $\hat{i}$, i.e., $\rho(x_{\hat{i}}, \epsilon) = Prob\{||\hat{x} - x_{\hat{i}}|| \leq ||x_{\hat{i}}||\epsilon\}$.*

4.3 Prior Knowledge

Let the $n \times m$ matrix X denote a private dataset, with each column of X being a record and each row an attribute. We assume that the attacker knows that transformation function T is an orthogonal transformation and knows the perturbed data $Y = M_T X$. In most realistic scenarios, the attacker has some additional *prior knowledge* which can potentially be used effectively for breaching privacy. We consider three types of prior knowledge.

Known input-output The attacker knows some collection of linearly independent private data records. In other words, the attacker has a set of linearly independent input-output pairs. In this scenario, we have developed an attack algorithm based on linear algebra and statistics theory.

Known sample The attacker knows that the original dataset arose as independent samples of some n-dimensional random vector V with unknown p.d.f. Also the attacker has another collection of independent samples from V. For technical reasons, we make a mild additional assumption: the covariance matrix of V has distinct eigenvalues. In this scenario, we have developed a principal component analysis (PCA)-based attack algorithm.

Independent signals Each data attribute can be thought of as a time-varying signal. All the signals, at any given time, are statistically independent and all the signals are non-Gaussian with the exception of one. In this scenario, we have developed an independent component analysis (ICA)-based attack algorithm.

Next, we describe and analyze attack techniques for *each type of* prior knowledge listed above.

4.4 Known Input-Output Attack

Consider the perturbation model

$$\begin{aligned} Y &= M_T X \Leftrightarrow \\ \left(\begin{array}{cc} Y_k & Y_{m-k} \end{array} \right) &= M_T \left(\begin{array}{cc} X_k & X_{m-k} \end{array} \right). \end{aligned}$$

Let X_k denote the first k columns of X and X_{m-k} the remainder (likewise for Y). We assume that columns of X_k are all linearly independent and X_k is known to the attacker (Y

is, of course, also known). The attacker will produce $\hat{x}$ and $1 \leq \hat{i} \leq m-k$ such that $\hat{x}$ is a good estimate of $x_{\hat{i}}$, the $\hat{i}^{th}$ column in X_{m-k} (the $(k+\hat{i})^{th}$ column in X).

If $k = n$, then the attacker can recover any column in X_{m-k} perfectly as $X_{m-k} = (Y_k X_k^{-1})' Y_{m-k}$. Thus, we assume $k < n$. Based on known information, the attacker can narrow down the space of possibilities for M_T to $\mathbb{M}(X_k, Y_k) = \{M \in \mathbb{O}_n : MX_k = Y_k\}$. Because the attacker has no additional information, any of these matrices is equally likely to have been M_T. The attacker chooses $\hat{M}$ uniformly from $\mathbb{M}(X_k, Y_k)$ and chooses index $1 \leq \hat{i} \leq m-k$ based on $\rho(x_{\hat{i}}, \epsilon)$ (the probability that an ϵ-*privacy breach* occurs given that $\hat{i}$ was chosen), then produces $\hat{x} = \hat{M}' y_{\hat{i}} = \hat{M}' M_T x_{\hat{i}}$. Later we will show how the attacker can compute $\rho(x_{\hat{i}}, \epsilon)$ for all $1 \leq \hat{i} \leq m-k$ from ϵ and Y (known information).

Note that $\mathbb{M}(X_k, Y_k)$, in most cases, is uncountable. As such, more precise definitions are needed for "choosing $\hat{M}$ uniformly from $\mathbb{M}(X_k, Y_k)$" and "the probability that $||\hat{M}' M_T x - x|| \leq ||x||\epsilon$". To do so, we first develop two key technical results.

4.4.1 Key Technical Results

Let $Col(X_k)$ denote the column space of X_k and $Col_\perp(X_k)$ denote its orthogonal complement, *i.e.*, $\{z \in \mathbb{R}^n : z'w = 0, \forall w \in Col(X_k)\}$. Because the columns of X_k are linearly independent, then the dimension of $Col(X_k)$ is k. The "Fundamental Theorem of Linear Algebra" [84, pg. 95] implies that the dimension of $Col_\perp(X_k)$ is $n-k$. Let U_k ($n \times k$) be the orthonormal basis for $Col(X_k)$, and U_{n-k} ($n \times (n-k)$) the orthonormal basis for $Col_\perp(X_k)$. Given $n \times p$ and $n \times q$ matrices A and B, let $[A|B]$ denote the $n \times (p+q)$ matrix whose first p columns are A and last q are B. Likewise, given $p \times n$ and $q \times n$ matrices A and B, let $\begin{bmatrix} A \\ B \end{bmatrix}$ denote the $(p+q) \times n$ matrix whose first p rows are A and last q are B. Let U denote $[U_k|U_{n-k}]$. Clearly, U is orthogonal.

The next Theorem provides a very useful alternate characterization of $\mathbb{M}(X_k, Y_k)$. It

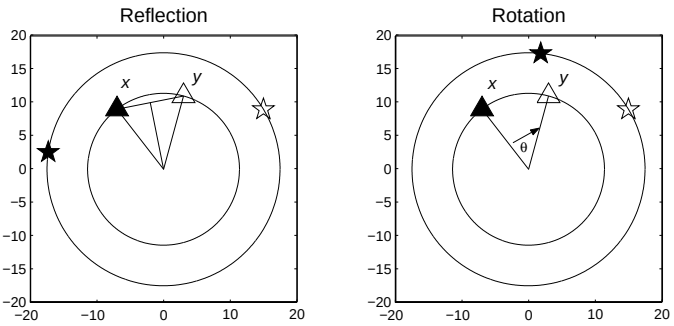


FIG. 4.3. Reflection and rotation in 2D space. Solid markers denote the original data and hollow markers denote the perturbed data.

is used critically throughout our analysis of the ϵ-privacy breach probability.

Theorem 4.4.1 *Let* $\mathbb{P}$ *denote* $\{M_T U_k U_k' + M_T U_{n-k} P U_{n-k}' : \forall P \in \mathbb{O}_{n-k}\}$, *then* $\mathbb{M}(X_k, Y_k) = \mathbb{P}$.

Proof: Please see Appendix 4.8.1 for the proof. □

This theorem shows that $\mathbb{M}(X_k, Y_k)$ has a closed-form expression:

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

For some special cases, for example, when $k = n$, $\mathbb{M}(X_k, Y_k)$ has only one element M_T; which echoes the fact that when $k = n$ the attacker can uniquely identify the perturbation matrix, and perfectly recover the private data. When $k = n-1$, $\mathbb{M}(X_k, Y_k)$ has only two elements $\{M_T U_k U_k' \pm M_T U_{n-k} U_{n-k}\}$. As an illustration, let us consider the orthogonal transformation in 2D space (shown in Figure 4.3). If we only know one data point x (solid triangle) and its perturbed counterpart y (hollow triangle) (in this case $k = 1$), we are not able to determine whether it is a rotation of a reflection. [1] If it was a rotation, the

[1] In 2D space, an orthogonal transformation is either a rotation or a reflection, depending on whether the determinant of the orthogonal matrix is $(+1)$ or (-1).

orthogonal perturbation matrix would be

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

where $\theta = \arccos(\frac{<x,y>}{||x||||y||})$. If it was a reflection, the orthogonal perturbation matrix would be

$$\frac{1}{u_x^2 + u_y^2} \begin{pmatrix} u_x^2 - u_y^2 & 2u_x u_y \\ 2u_x u_y & u_y^2 - u_x^2 \end{pmatrix},$$

where $u = (x + y)/2$, u_x is the first dimension of u, and u_y is the second dimension of u. Therefore, only if the attacker gets another data point and its perturbed version, can s/he determine the original perturbation matrix, and hence recover other private data.

Theorem 4.4.1 leads to the following corollary, which is going to be used to derive the closed-form expression of $\rho(x, \epsilon)$.

Corollary 4.4.2 *Let L be the linear mapping $M \in \mathbb{M}(X_k, Y_k) \mapsto (M_T U_{n-k})' M U_{n-k}$.*

1. *L is one-to-one and $L(\mathbb{M}(X_k, Y_k)) = \mathbb{O}_{n-k}$.*
2. *For any $x \in \mathbb{R}^n$ and any $M \in \mathbb{M}(X_k, Y_k)$, $||M' M_T x - x|| = ||L(M)' U'_{n-k} x - U'_{n-k} x||$.*

Proof: **1.** Let $M \in \mathbb{M}(X_k, Y_k)$. By Theorem 4.4.1, there exists $P_M \in \mathbb{O}_{n-k}$ such that $M = M_T U_k U'_k + M_T U_{n-k} P_M U'_{n-k}$. We have,

$$\begin{aligned} L(M) &= (M_T U_{n-k})' M_T U_k U'_k U_{n-k} + (M_T U_{n-k})' M_T U_{n-k} P_M U'_{n-k} U_{n-k} \\ &= 0 + P_M. \end{aligned}$$

Thus, $L(\mathbb{M}(X_k, Y_k)) \subseteq \mathbb{O}_{n-k}$. Let us now consider M_1, $M_2 \in \mathbb{M}(X_k, Y_k)$ such that $L(M_1) = L(M_2)$. It follows that $P_{M_1} = P_{M_2}$, so, $M_1 = M_2$. Therefore, L is one-to-one. Now consider $P \in \mathbb{O}_{n-k}$. By Theorem 4.4.1, $(M_T U_k U_k' + M_T U_{n-k} P U_{n-k}') \in \mathbb{M}(X_k, Y_k)$, and, by the above argument, L sends this element to P. Thus $\mathbb{O}_{n-k} = L(\mathbb{M}(X_k, Y_k))$.

2. Because $U' \in \mathbb{O}_n$ and any $M \in \mathbb{M}(X_k, Y_k)$ equals $M_T U_k U_k' + M_T U_{n-k} L(M) U_{n-k}'$, it follows that

$$\begin{aligned}
||M' M_T x - x|| &= ||U'(M' M_T x - x)|| \\
&= ||U'(M_T U_k U_k' + M_T U_{n-k} L(M) U_{n-k}')' M_T x - U' x|| \\
&= ||[U_k|U_{n-k}]' U_k U_k' x + [U_k|U_{n-k}]' U_{n-k} L(M)' U_{n-k}' x - [U_k|U_{n-k}]' x|| \\
&= \left|\left|\begin{bmatrix} U_k' x \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ L(M)' U_{n-k}' x \end{bmatrix} - \begin{bmatrix} U_k' x \\ U_{n-k}' x \end{bmatrix}\right|\right| \\
&= ||L(M)' U_{n-k}' x - U_{n-k}' x||.
\end{aligned}$$

□

Now we can address the issue of making precise definitions for "choosing $\hat{M}$ uniformly from $\mathbb{M}(X_k, Y_k)$" and "the probability that $||\hat{M}' M_T x - x|| \leq ||x||\epsilon$". First we define a "uniform" probability measure on $\mathbb{M}(X_k, Y_k)$. Then we describe a procedure for choosing a matrix $\hat{M}$ "uniformly" from $\mathbb{M}(X_k, Y_k)$.

Because $\mathbb{O}_{n-k}$ is a locally compact topological group [78, pg. 293], it has a Haar probability measure, denoted by μ, over $\mathbb{B}$, the Borel algebra on $\mathbb{O}_{n-k}$ [85, pg. 65]. This is commonly regarded as the standard uniform probability measure over $\mathbb{O}_{n-k}$. Its key property is *left-invariance*: for all $\mathcal{B} \in \mathbb{B}$ and all $M \in \mathbb{O}_{n-k}$, $\mu(\mathcal{B}) = \mu(M\mathcal{B})$, *i.e.*, shifting $\mathcal{B}$ by a rigid motion does not change the probability assignment. Similarly, we need a left-invariant probability measure on the Borel algebra over $\mathbb{M}(X_k, Y_k)$. Such a measure

can be regarded as the uniform probability measure on $\mathbb{M}(X_k, Y_k)$. Consider $L^{-1}(\mathbb{B}) = \{L^{-1}(\mathcal{B}) : \mathcal{B} \in \mathbb{B}\}$. From Corollary 4.4.2 part 1, it follows that $L^{-1}(\mathbb{B})$ is the Borel algebra over $\mathbb{M}(X_k, Y_k)$. Moreover, $\mu \circ L$ forms a left-invariant probability measure on the Borel algebra over $\mathbb{M}(X_k, Y_k)$. [2] Thus, $\mu \circ L$ can be regarded as the uniform probability measure on $\mathbb{M}(X_k, Y_k)$.

There are standard algorithms (*e.g.*, [86]) for generating a matrix which can be thought to have been chosen from $\mathbb{O}_{n-k}$ according to μ, *i.e.*, uniformly. Thus, a matrix $\hat{M}$ can be chosen uniformly from $\mathbb{M}(X_k, Y_k)$ as follows: (i) generate $P \in \mathbb{O}_{n-k}$ according to [86] and (ii) set $\hat{M}$ to $L^{-1}(P)$.

Now we give a precise definition of $\rho(x, \epsilon)$, the probability that $||\hat{M}' M_T x - x|| \leq ||x||\epsilon$ where $\hat{M}$ is chosen uniformly from $\mathbb{M}(X_k, Y_k)$. Let $\mathbb{M}(x, \epsilon)$ denote $\{M \in \mathbb{M}(X_k, Y_k) : ||M' M_T x - x|| \leq ||x||\epsilon.\}$ From Corollary 4.4.2 part 2, it follows that $L(\mathbb{M}(x, \epsilon)) = \{P \in \mathbb{O}_{n-k}\text{: } ||P' U'_{n-k} x - U'_{n-k} x|| \leq ||x||\epsilon\}$. Let $\mathbb{O}(x, U_{n-k}, \epsilon)$ denote this set. Because $\mathbb{O}(x, U_{n-k}, \epsilon)$ is a closed subset of $\mathbb{O}_{n-k}$, it is a Borel subset of $\mathbb{O}_{n-k}$. Thus, $\mathbb{M}(x, \epsilon)$ is a Borel subset of $\mathbb{M}(X_k, Y_k)$ (so, $\mu \circ L$ is defined on $\mathbb{M}(x, \epsilon)$). Formally then, $\rho(x, \epsilon)$ is defined to be $\mu \circ L(\mathbb{M}(x, \epsilon))$ which equals $\mu(\mathbb{O}(x, U_{n-k}, \epsilon))$.

4.4.2 A Closed-Form Expression for Privacy Breach

Let $S_{n-k}(||U'_{n-k} x||)$ denote the hyper-sphere in $\mathbb{R}^{n-k}$ centered at the origin with radius $||U'_{n-k} x||$. For any $\mathcal{A} \subseteq S_{n-k}(||U'_{n-k} x||)$, let $SA(\mathcal{A})$ denote the surface area of $\mathcal{A}$ (assuming it is defined).[3] Let $S_{n-k}(U'_{n-k} x, ||x||\epsilon)$ denote the portion of $S_{n-k}(||U'_{n-k} x||)$ whose distance from $U'_{n-k} x$ is no larger than $||x||\epsilon$, *i.e.*, $S_{n-k}(U'_{n-k} x, ||x||\epsilon) = \{z \in S_{n-k}(||U'_{n-k} x||) : ||z - U'_{n-k} x|| \leq ||x||\epsilon\}$.

[2] $\circ$ denotes a function composition.

[3] In Appendix 4.8.2 we provide a definition of surface area on a hyper-sphere.

It is shown in Appendix 4.8.2 that[4]

$$\begin{aligned}\rho(x,\epsilon) &= \frac{SA(S_{n-k}(U'_{n-k}x, ||x||\epsilon))}{SA(S_{n-k}(||U'_{n-k}x||))} \\ &= \begin{cases} \left(\frac{1}{\pi}\right) 2arcsin\left(\frac{||x||\epsilon}{2||U'_{n-k}x||}\right) & \text{if } ||x||\epsilon < 2||U'_{n-k}x||; \\ 1 & \text{otherwise.} \end{cases}\end{aligned}$$

An alternate characterization of $||U'_{n-k}x||$ yields a more intuitive form of the second right-hand side. Consider $U_kU'_kx$, the orthogonal projection of x into $Col(X_k)$. This is the closest point in $Col(X_k)$ from x. So, the distance of x from $Col(X_k)$, denoted $d(x, X_k)$, is naturally defined as $||x - U_kU'_kx||$. Observe that,

$$\begin{aligned}d(x, X_k) &= ||U'(x - U_kU'_kx)|| \\ &= \left|\left|\begin{bmatrix} U'_kx \\ U'_{n-k}x \end{bmatrix} - \begin{bmatrix} U'_kx \\ 0 \end{bmatrix}\right|\right| \\ &= ||U'_{n-k}x||.\end{aligned}$$

Thus,

$$\rho(x,\epsilon) = \begin{cases} \left(\frac{1}{\pi}\right) 2arcsin\left(\frac{||x||\epsilon}{2d(x,X_k)}\right) & \text{if } ||x||\epsilon < 2d(x, X_k); \\ 1 & \text{otherwise.} \end{cases} \tag{4.2}$$

Alternate characterizations of $d(x, X_k)$ and $||x||$ yield a right-hand side directly allowing the adversary to compute $\rho(x, \epsilon)$. Because M_T is orthogonal, $||x|| = ||M_Tx||$. Because $Col(X_k)$ has dimension k and M_T is orthogonal, then $Col(M_TX_k) = Col(Y_k)$ has dimension k. So, there exists V_k an $n \times k$ orthogonal matrix such that $Col(V_k) = Col(Y_k)$.

[4]Note that the "otherwise" case includes $x = 0$ and $||x||\epsilon \geq 2||U'_{n-k}x||$.

Because V_k is orthogonal, then $V_k V_k' M_T x$ is the projection of $M_T x$ into $Col(Y_k)$, thus, $d(M_T x, Y_k)$ is $||V_k V_k' M_T x - M_T x||$. Next we show that $d(x, X_k) = d(M_T x, Y_k)$.

Because $Col(X_k) = Col(U_k)$, then $Col(M_T X_k) = Col(M_T U_k)$, so, $Col(V_k) = Col(Y_k)$ $= Col(M_T X_k) = Col(M_T U_k)$. Thus, there exists $k \times k$ matrix P such that $V_k P = M_T U_k$. Observe that

$$\begin{aligned} P'P &= (V_k P)'(V_k P) \\ &= (M_T U_k)'(M_T U_k) \\ &= I_k, \end{aligned}$$

so, P is orthogonal. We have,

$$\begin{aligned} d(x, X_k) &= ||U_k U_k' x - x|| \\ &= ||M_T U_k U_k' M_T' M_T x - M_T x|| \\ &= ||(V_k P)(V_k P)' M_T x - M_T x|| \\ &= ||V_k V_k' M_T x - M_T x|| \\ &= d(M_T x, Y_k). \end{aligned}$$

The above results show that the attacker could compute the distance $d(x, X_k)$ using the perturbed data. Therefore, Equation 4.2 can be rewritten as

$$\rho(x, \epsilon) = \begin{cases} \left(\frac{1}{\pi}\right) 2arcsin\left(\frac{||M_T x||\epsilon}{2d(M_T x, Y_k)}\right) & \text{if } ||M_T x||\epsilon < 2d(M_T x, Y_k); \\ 1 & \text{otherwise.} \end{cases} \quad (4.3)$$

4.4.3 Known Input-Output Attack Algorithm

As stated earlier, the adversary chooses $\hat{M}$ uniformly from $\mathbb{M}(X_k, Y_k)$ and $1 \leq \hat{i} \leq m - k$ to maximize $\rho(x_{\hat{i}}, \epsilon)$. The precise details of the attack technique can be seen in Algorithm 4.4.3.1.

Algorithm 4.4.3.1 Known Input-Output Attack Technique

Inputs: X_k, an set of linearly independent columns from X known to the attacker and $Y = M_T X$, known to the attacker, where $M_T \in \mathbb{O}_n$ is an unknown, and $\epsilon \geq 0$, known to the attacker.

Outputs $1 \leq \hat{i} \leq m-k$ which maximizes $\rho(x_{\hat{i}}, \epsilon)$ and $\hat{x} \in \mathbb{R}^n$ the corresponding estimate of $x_{\hat{i}}$.

1: Compute V_k an $n \times k$, orthogonal matrix where $Col(V_k) = Col(Y_k)$ from Y_k using the Gram-Schmidt process.
2: For each $1 \leq j \leq m - k$ do
3: Compute $d(y_j, Y_k) = ||V_k V_k' y_j - y_j||$ and $||y_j||\epsilon$.
4: Compute $\rho(x_j, \epsilon)$ using Equation 4.3.
5: End For.
6: Set $\hat{i} \leftarrow \max_{1 \leq j \leq m-k}\{\rho(x_j, \epsilon)\}$.
7: Choose $\hat{M}$ uniformly from $\mathbb{M}(X_k, Y_k)$.
8: Set $\hat{x} \leftarrow \hat{M}' y_{\hat{i}}$.

4.4.4 Effectiveness of the Attack

In the previous sections, we have shown that: 1) the attacker can compute the probability of privacy breach for a given private data record and relative error bound ϵ; 2) the larger the ϵ, the higher the probability of privacy breach; 3) the closer the private record is to the column space of the known records, the higher the probability of privacy breach; and 4) the attacker could compute the distance $d(x, X_k)$ using the perturbed data.

As a concrete example, let us consider the data in Table 4.1. We assume that the attacker knows the perturbed data, the value of x_1, and also knows that y_1 comes from x_1. Because the distance of x_2 from the column space of x_1 is 0, we have $\rho(x_2, \epsilon) = 1$ for any

Private Data	x_1	x_2	x_3
	25.0000	30.0000	45.0000
	75.0000	90.0000	105.0000
Perturbed Data	y_1	y_2	y_3
	-42.0198	-50.4237	-68.5443
	66.9652	80.3582	91.3875

Table 4.1. Example of Known Input-Output Attack.

$\epsilon > 0$. On the other hand, the distance of x_3 from the column space of x_1 is 9.4868, thus $\rho(x_3, \epsilon) = \frac{1}{\pi} 2 \arcsin\left(\frac{||x_3||\epsilon}{2\times 9.4868}\right)$, *e.g.*, $\rho(x_3, 0.01) = 3.84\%$.

The maximum probability of an ϵ-privacy breach is $\rho(x_{\hat{i}}, \epsilon) = \max_{1\leq j\leq m-k} \rho(x_j, \epsilon)$. Let $\gamma(x_{\hat{i}}, \epsilon)$ denote $\frac{||x_{\hat{i}}||\epsilon}{2d(x_{\hat{i}}, X_k)}$. From Equation 4.2, the breach probability goes to zero nearly linearly with $\gamma(x_{\hat{i}}, \epsilon)$,[5] and goes to one much faster as $\gamma(x_{\hat{i}}, \epsilon)$ does. If the data owner knows that X_k is in the attacker's prior knowledge, then the owner can protect against this attack by simply not releasing $M_T x_j$ for any x_j where $\gamma(x_j, \epsilon)$ is unacceptably big. On the other hand, if the owner does not know that X_k is prior knowledge, then this attack technique can be quite damaging.

4.5 Known Sample Attack

In this scenario, we assume that each data record arose as an independent sample from a random vector V with unknown p.d.f. We also make the following mild technical assumption: the population covariance matrix Σ_V of V has all distinct eigenvalues.[6] We make this assumption because it holds in most practical situations [87, pg. 27]. Furthermore, we assume that the attacker has a collection of p samples that arose independently from V – these are denoted as the columns of matrix S.

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

[6] Given $n \times n$ matrix A, a complex number λ is an eigenvalue of A if and only if the determinate of $A - I_n\lambda$, denoted $det(A - I_n\lambda)$, is zero. Because $det(A - I_n x)$ is an n-degree polynomial, then A can have at most n distinct eigenvalues.

In this section we design a Principal Component Analysis (PCA)-based attack technique. Unlike Section 4.4, we do not attempt a rigorous analysis of the attacker's success probability. Instead, we analyze the recovery error through experiments.

4.5.1 Principal Component Analysis (PCA) Preliminaries

Let Σ_V denote the population covariance matrix of V. Because Σ_V is an $n \times n$, symmetric matrix (and we assume it has all distinct eigenvalues), it has n real eigenvalues $\lambda_1 > \ldots > \lambda_n$ [84, pg. 295]. Associated with each eigenvalue λ_i is its eigenspace, $\{z \in \mathbb{R}^n : \Sigma_V z = z\lambda_i\}$. It can be shown that because Σ_V has distinct eigenvalues, the eigenspaces are pair-wise orthogonal and each has dimension one [84, pg. 295]. As is standard practice, we restrict our attention to only a small number of eigenvectors. Let $\mathcal{Z}(V)_i$ denote the set of all eigenvectors $z \in \mathbb{R}^n$ such that $\Sigma_V z = z\lambda_i$ and $||z|| = 1$. Now consider random vector $T(V) = M_T V$ and let $\Sigma_{M_T V}$ denote its covariance matrix. The eigenspaces of Σ_V are related in a natural way to those of $\Sigma_{M_T V}$, as shown by the following theorem.

Theorem 4.5.1 *The eigenvalues of Σ_V and $\Sigma_{M_T V}$ are the same and $M_T \mathcal{Z}(V)_i = \mathcal{Z}(M_T V)_i$, where $M_T \mathcal{Z}(V)_i$ equals $\{M_T z : z \in \mathcal{Z}(V)_i\}$; and $\mathcal{Z}(M_T V)_i$ denotes the set of eigenvectors $w \in \mathbb{R}^n$ such that $\Sigma_{M_T V} w = w\lambda_i$ and $||w|| = 1$.*

Proof: First we derive an expression for Σ_V in terms of $\Sigma_{M_T V}$.

$$\begin{aligned}
\Sigma_{M_T V} &= E[(M_T V - E[M_T V])(M_T V - E[M_T V])'] \\
&= E[M_T (V - E[V])(V - E[V])' M_T'] \\
&= M_T E[(V - E[V])(V - E[V])'] M_T' \\
&= M_T \Sigma_V M_T'.
\end{aligned}$$

Now consider any eigenvalue λ_i of Σ_V. Basic properties of the matrix determinate show that $det(\Sigma_V - I_n\lambda_i)$ equals $det(M_T\Sigma_V M_T' - I_n\lambda_i)$. Therefore, λ_i is an eigenvalue of Σ_{M_TV}. [7]

We have shown that Σ_V and Σ_{M_TV} have the same eigenvalues. Now consider any non-zero $w \in \mathbb{R}^n$. We have that

$$\begin{aligned}
w \in \mathcal{Z}(M_TV)_i &\Leftrightarrow \Sigma_{M_TV}w = w\lambda_i \text{ and } ||w|| = 1 \\
&\Leftrightarrow M_T\Sigma_V M_T'w = w\lambda_i \text{ and } ||w|| = 1 \\
&\Leftrightarrow \Sigma_V(M_T'w) = (M_T'w)\lambda_i \text{ and } ||M_T'w|| = 1 \\
&\Leftrightarrow M_T'w \in \mathcal{Z}(V)_i \\
&\Leftrightarrow w \in M_T\mathcal{Z}(V)_i.
\end{aligned}$$

□

Because all the eigenspaces of Σ_V have dimension one, it can be shown that $\mathcal{Z}(V)_i$ contains only two vectors such that -1 times one equals the other. Let z_i be the lexicographically larger one. Then, $\mathcal{Z}(V)_i = \{z_i, -z_i\}$. Let Z denote the $n \times n$ eigenvector matrix whose i^{th} column is z_i. Because the eigenspaces of Σ_V are pairwise orthogonal and $||z_i|| = 1$, Z is orthogonal. Similarly, we have that $\mathcal{Z}(M_TV)_i = \{w_i, -w_i\}$ (w_i is the lexicographically larger among $w_i, -w_i$) and W is the eigenvector matrix with i^{th} column w_i (W is orthogonal). Note again that columns in both Z and W are ordered such that the i^{th} eigenvector is associated with the i^{th} eigenvalue. The following result forms the basis of the attacker's attack algorithm.

Corollary 4.5.2 *Let $\mathbb{I}_n$ be the space of all $n \times n$, matrices with each diagonal entry ± 1*

[7] This simple proof is based on the definition of eigenvalues.

and each off-diagonal entry 0 (2^n matrices in total). There exists $D_0 \in \mathbb{I}_n$ such that $M_T = WD_0Z'$.

Proof: Theorem 4.5.1 implies that for all $1 \leq i \leq n$, $M_T z_i = w_i$ or $-M_T z_i = w_i$. Therefore, for some $D_0 \in \mathbb{I}_n$, $M_T Z D_0 = W$. Because $D_0^{-1} = D_0$ and Z is orthogonal, the desired result follows. □

4.5.2 Known Sample Attack (PCA Attack) Algorithm

First assume the attacker knows the population covariance Σ_V and $\Sigma_{M_T V}$. Thus, the attacker can compute W, the eigenvector matrix of $\Sigma_{M_T V}$, and Z, the eigenvector matrix of Σ_V. By Corollary 4.5.2, the attacker knows that M_T equals WD_0Z' for some $D_0 \in \mathbb{I}_n$, and therefore, the original data would be recovered by $M_T' Y = ZD_0W'Y$. The problem is how to choose the right D from all the possible 2^n elements in $\mathbb{I}_n$. To do so, the attacker must utilize S and Y, in particular, the fact that these arose as independent samples from V and $M_T V$, respectively. For each $D \in \mathbb{I}_n$, each column of $WDZ'S$ arose as an independent sample from $WDZ'V$. If $D = D_0$, then $WDZ' = M_T$, so, $WDZ'S$ and Y should come from the same p.d.f. The attacker will choose $D \in \mathbb{I}_n$ such that $WDZ'S$ is most likely to have arisen from the same p.d.f. as Y. To make this choice, a similarity function $G(WDZ'S, Y)$ is introduced, and the D that maximizes G is chosen. There might be many ways to define this function. In this paper, we use a multivariate two-sample hypothesis test for equal distributions [88]. The two-sample problem assumes that there are two sets of independent samples $x_1, x_2, \ldots, x_{m_1}$ and $y_1, y_2, \ldots, y_{m_2}$ of independent random vectors with distributions F_1 and F_2, respectively. The goal of two-sample problem is to test $H_0 : F_1 = F_2$, versus the composite alternative $H_1 : F_1 \neq F_2$. For each $D \in \mathbb{I}_n$, we compute the p-value of the test on $WDZ'S$ and Y, denoted by $p(D)$. Here the p-value is defined as the smallest level of significance at which H_0 would be rejected on a given data set.

Small p-values suggest that the null hypothesis is unlikely to be true. The smaller it is, the more convincing is the rejection of the null hypothesis. Therefore the value of function G is nothing but the p-value, and the D matrix that is associated with the highest p-value is chosen.

In practice, the population covariance Σ_V and $\Sigma_{M_T V}$ are unknown, and will be replaced by the sample covariance Σ_S and Σ_Y from S and Y (independent samples arising from V and $M_T V$). Algorithm 4.5.2.1 shows the complete PCA-based attack procedure.

Algorithm 4.5.2.1 PCA-based Attack Technique

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

Outputs $\hat{x}$, $1 \leq \hat{i} \leq m$, an estimation of $x_{\hat{i}}$.

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

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

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

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

5: Choose $1 \leq \hat{i} \leq m$ randomly and set $\hat{x} = \hat{X}_{\hat{i}}$.

The computation cost of Algorithm 4.5.2.1 is $O(n^2(m+p) + n^3 + 2^n B)$ assuming $G(.,.)$ requires $O(B)$ computation. For the two-sample test, $B = (m+p)^2$, so, the total computation of the algorithm is $O(2^n (m+p)^2)$.

4.5.3 Experiments

To validate the PCA-based attack algorithm, we conducted experiments on both synthetic and real world data. One such synthetic dataset contains $10,000$ data points, which are generated from a three-dimensional Gaussian distribution with mean $(10, 10, 10)$ and

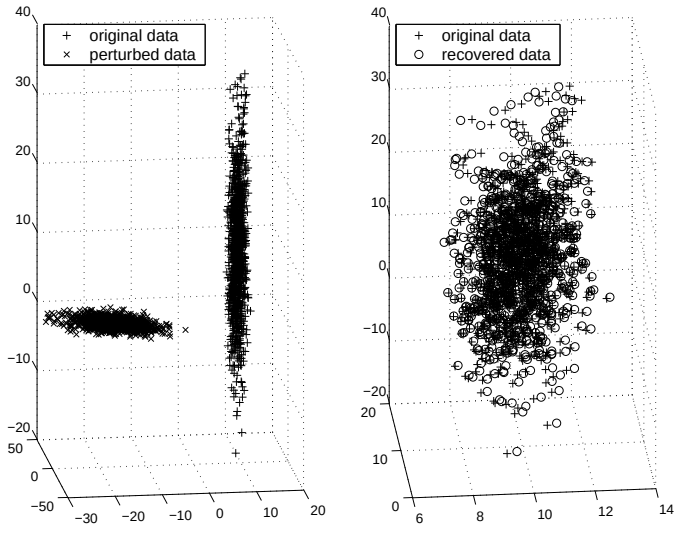


FIG. 4.4. PCA-based attack for three-dimensional Gaussian data. The average relative error of the recovered data is 0.0265. (2% sample)

covariance $\begin{pmatrix} 1 & 1.5 & 0.5 \\ 1.5 & 3 & 2.5 \\ 0.5 & 2.5 & 75 \end{pmatrix}$. The attacker has 200 sample data points (2% of the size of original data) chosen from the same distribution. Figure 4.4 shows the results of perturbation and recovery. It can be seen that although the perturbed data is very different from the original one, the recovered data almost overlaps with the original data.[8] To further examine how sample size and relative error bound ϵ affects the quality of the attack, we conducted two sets of experiments. The first set of experiments (Figure 4.5) show that when the perturbation matrix and relative error bound are fixed, the probability of privacy breach increases as the sample size increases. The second set of experiments (Figure 4.6) depict that when the perturbation matrix and the sample size are fixed, the probability of privacy breach increases as the relative error bound (that the attacker can tolerate) increases.

For the real world data, we chose the Adult Database and Letter Recognition Database

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

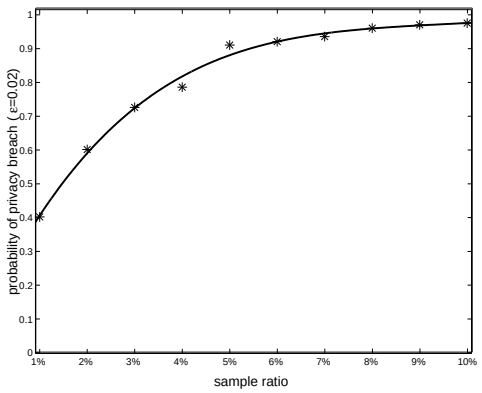


FIG. 4.5. Performance of PCA-based attack for three-dimensional Gaussian data w.r.t. sample size. The relative error bound ϵ is fixed to be 0.02. The solid line shows a best polynomial fit to the points. This line was generated with Matlab's curving fitting toolbox.

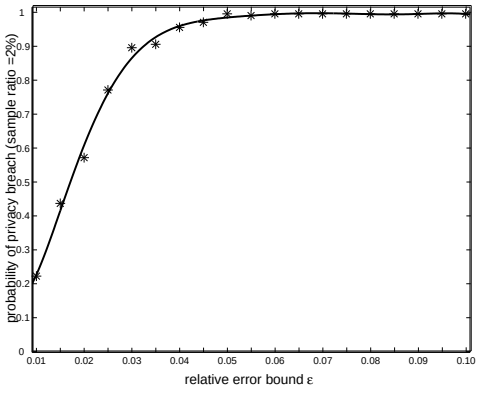


FIG. 4.6. Performance of PCA-based attack for three-dimensional Gaussian data w.r.t. relative error bound. The sample ratio is fixed to be 2%. The solid line shows a best polynomial fit to the points. This line was generated with Matlab's curving fitting toolbox.

from the UCI machine learning repository. The Adult data contains $32,561$ records, and it is extracted from the census bureau database. For the purpose of visualization, we only selected three numeric attributes: age, education-num and hours-per-week, for the experiment. The Letter Recognition data has $20,000$ instances and 16 numeric features. We chose the first 6 features (excluding the class label) for the experiments. We randomly separated each dataset into two disjoint sets. One set is viewed as the original data, and the other one is the attacker's sample data, which accounts for 2% of the original data. Figure 4.7 shows the results of perturbation and PCA attack for Adult data. Figure 4.8 and 4.9 shows the results of perturbation and PCA attack for Letter Recognition data. It can be seen that the recovered data approximates the original data very well. To examine the influence of sample size and relative error bound, we fixed the orthogonal perturbation matrix, and performed the same series of experiments as we did for Gaussian data. Figure 4.10 and 4.11 give the results for Adult data. Figure 4.12 and 4.13 give the results for Letter Recognition

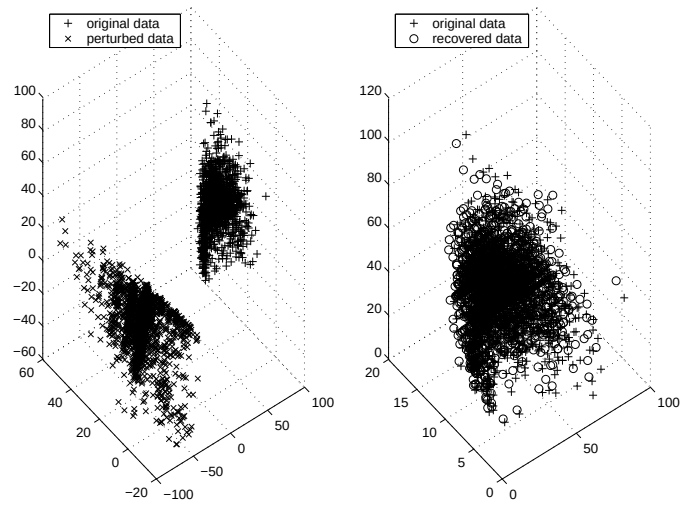


FIG. 4.7. PCA-based attack for Adult data. The average relative error of the recovered data is 0.1081. (2% sample)

data.

From the above experiments, we have the following observations: (1) the higher the relative error bound the attacker can tolerate, the higher the probability of privacy breach; (2) the larger the sample size, the better the quality of data recovery; and (3) among these three data sets, the PCA-based attack works best for Gaussian data, next Letter Recognition data, and then Adult data. The first two observations require no explainations. We will discuss the third one in the next section.

To evaluate the complexity of the PCA attack algorithm, we generated multivariate Gaussian data with dimensionality ranging from 2 to 12. Each data set contains 5250 records, 250 records of which are used as samples, and the remaining 5000 records as private data. The energy test proposed in [88] was used to quantify similarity ($G(.,.)$), The experiment was conducted on a dual-processor workstation with 3.00GHz and 2.99GHz Xeon CPUs and 3.00GB RAM. We observed that for 2-dimensional data, it took 143.1090 seconds, and for 12-dimensional data, it took 1.2442×10^5 seconds. As expected, the running time goes up rapidly with number of dimensions. However for a modest number

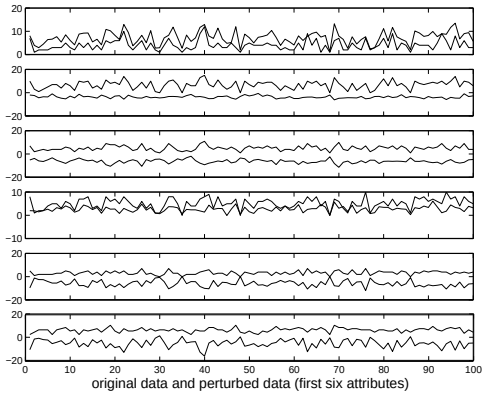


FIG. 4.8. Perturbation of the Letter Recognition data. This figure shows the first 100 records from the original and the perturbed data. Each row in the figure depicts an attribute of the data.

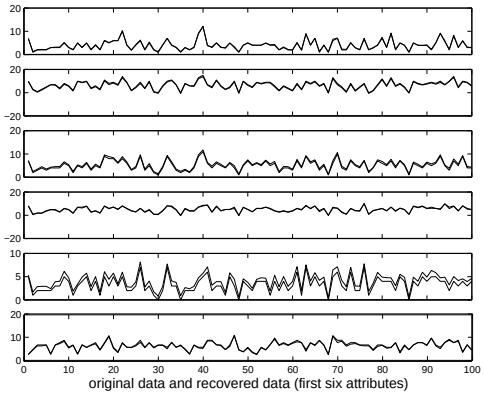


FIG. 4.9. PCA-based attack for Letter Recognition data. This figure shows the first 100 records from the original and the recovered data. Each row in the figure depicts an attribute of the data. The average relative error of the recovered data is 0.1008. (2% sample).

of dimensions, the algorithm still seems computationally feasible.

4.5.4 Effectiveness of the Attack

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

Covariance estimation quality: A great deal of work has been conducted in the statistics community on estimating the covariance matrix of a random vector based on independent samples [87, Chapter 10.4]. Generally speaking, the quality of the estimation of sample covariance is correlated with the following factors.

- **Outliers** It is usually desirable to use a robust approach for covariance estimation to downweights the disproportionate effect of any outlying records. In all the experiments we used the simple, standard sample covariance estimator: given two length m vectors x and y, $Cov(x,y) = \frac{\sum_{\ell=1}^{m}(x_\ell - \overline{x})(y_\ell - \overline{y})}{m-1}$ where $\overline{x}$ and $\overline{y}$ are the averages of

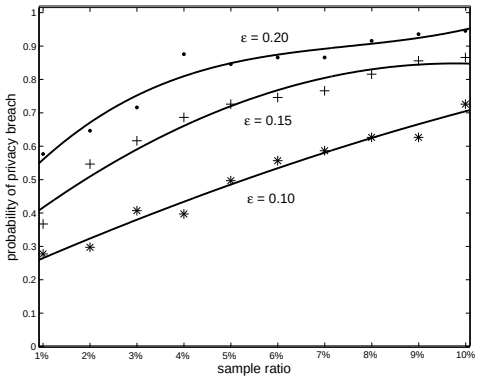


FIG. 4.10. Performance of PCA-based attack for Adult data w.r.t. sample size. The relative error bound ϵ is fixed to be 0.10, 0.15 and 0.20, respectively.

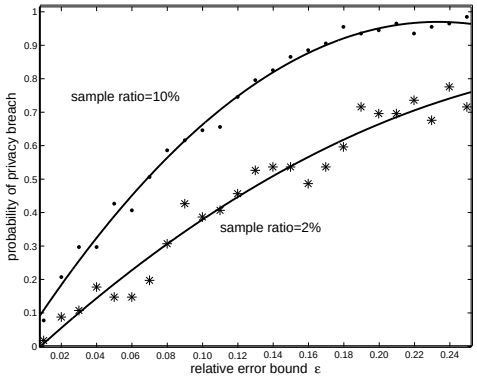


FIG. 4.11. Performance of PCA-based attack for Adult data w.r.t. relative error bound. The sample ratio is fixed to be 2% and 10%, respectively.

x and y. We note that any elaborate, robust estimation methods [87, Chapter 10.4] could be used without change by our approach.

- **Sample Size** Loosely speaking, larger samples are better than smaller samples because larger samples tend to minimize the probability of errors, maximize the accuracy of population estimates. The work in [89] investigated both sample size and the ratio of records to attributes. It showed that as the total number of samples increases, the ratio becomes less important; the converse is also true. Both factors matter in some sense, and ignoring either one can have errors of inference.

The p.d.f. of V: First, suppose the eigenvalues of Σ_V are nearly identical. For example, suppose V has a diagonal covariance matrix whose diagonal entries (from top-left to bottom-right) are d, $d-\epsilon$, $d-2\epsilon$, ..., $d-n\epsilon$ where $d-n\epsilon > 0$ and $0 < \epsilon < 1$. In this case, small errors in estimating Σ_V from sample S can produce a different ordering of the eigenvectors [9], hence, large errors in the attacker's recovery. As an extreme case, when V is the

[9] Note that the order of eigenvectors is determined by the values of eigenvalues.

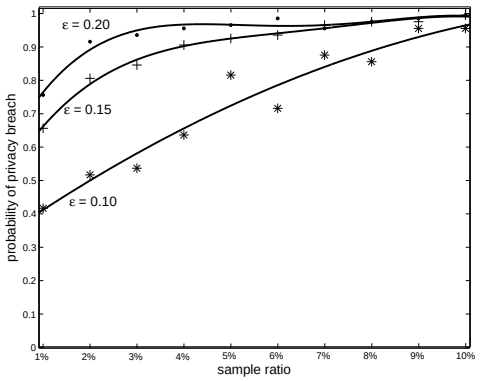


FIG. 4.12. Performance of PCA-based attack for Letter Recognition data w.r.t. sample size. The relative error bound ϵ is fixed to be 0.10, 0.15 and 0.20, respectively.

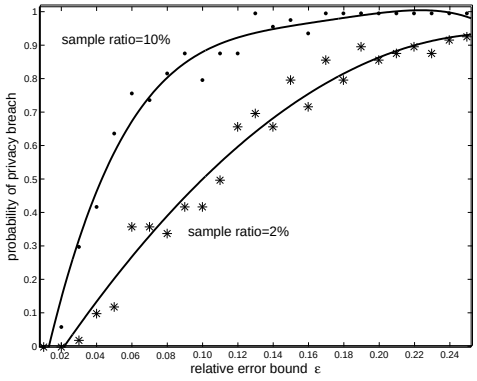


FIG. 4.13. Performance of PCA-based attack for Letter Recognition data w.r.t. relative error bound. The sample ratio is fixed to be 2% and 10%, respectively.

n-variate Gaussian with covariance matrix $I_n\gamma$ for some constant γ, all the eigenvalues are the same, and any vectors in the space can be the eigenvectors, the PCA attack algorithm will fail.

Consider the minimum ratio of any pair of eigenvalues, *i.e.*, $min\{\lambda_i/\lambda_j : \forall i \neq j; i, j = 1, \ldots, n\}$ (we call this the *minimum eigen-ratio*). We would expect that, the smaller this value, the smaller the attacker's success probability. To examine this hypothesis, we generated a three-dimensional dataset of tuples sampled independently from a Gaussian with mean $(10, 10, 10)$ and covariance $\begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & b \end{pmatrix}$. By changing the value of b from 2 to 40, we can change the minimum eigen-ratio of the covariance from 1 to 20. The original data contains $10,000$ tuples. We fixed the sample ratio to be 2% and relative error bound $\epsilon = 0.05$. Figure 4.14 shows that when all other parameters are fixed, the higher the eigen-ratio, the better the performance of the attack algorithm. This actually explains why, in our previous experiments, PCA attack works best for Gaussian data, then Letter Recognition data, and then Adult data. A simple computation shows that the minimum eigen-ratios of the Gaussian data, Letter Recognition data and Adult data are $19.6003, 1.3109, 1.2734$,

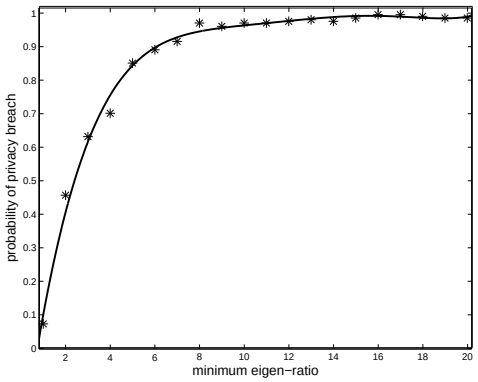


FIG. 4.14. Performance of PCA-based attack w.r.t. minimum eigen-ratio. The relative error bound ϵ is fixed to be 0.05, and the sample ratio is 2%.

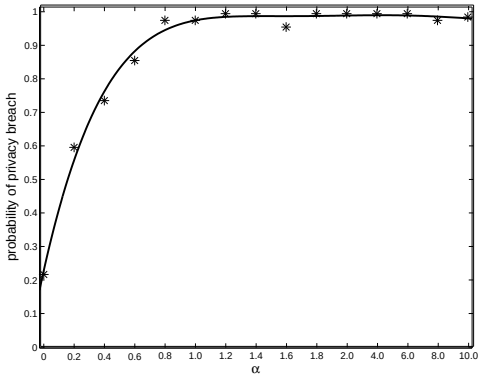


FIG. 4.15. Performance of PCA-based attack w.r.t. α. The relative error bound ϵ is fixed to be 0.05, and the sample ratio is 2%.

respectively.

Second, suppose for some $D_i \neq D_0 \in \mathbb{I}_n$, the p.d.f., f, of V is *invariant over* D_i in the sense that f_{D_i} and f_{D_0} can't be distinguished, where f_{D_i} is the p.d.f. $v \in \mathbb{R}^n \mapsto f(WD_iZ'v)$. Then, the hypothesis test could possibly conclude that $WD_0Z'S$, $WD_iZ'S$ and Y all arose from the same p.d.f., so the p-value $p(D_0)$ may not be larger than $p(D_i)$, and the attack algorithm will fail. We say that f is *invariant* if there exists some $D_i \neq D_0 \in \mathbb{I}_n$, such that f is invariant over D_i.

We would expect that the closer f is to invariance, the smaller the attacker's success probability. To examine this hypothesis we need a metric for quantifying the degree to which f is invariant. Intuitively, the invariance of f can be quantified as the degree to which f_{D_i} and f_{D_0} are distinguishable (minimized over all $D_i \neq D_0 \in \mathbb{I}_n$). To formalize this definition, we use the symmetrized Kullback-Leibler divergence $KL(g||h)+KL(h||g)$ to measure the distance between two distributions g and h. This measurement is symmetric and nonnegative, and when it is equal to zero, the distributions can be regarded as indistinguishable. The symmetrized Kullback-Leibler distance between continuous distribution g

to h is defined as

$$KL(g||h) + KL(h||g) = \int_{-\infty}^{+\infty} g(x) \log \frac{g(x)}{h(x)} dx + \int_{-\infty}^{+\infty} h(x) \log \frac{h(x)}{g(x)} dx.$$

So we quantify invariance as

$$Inv(f) \quad = \quad \min_{D_i \neq D_0 \in \mathbb{I}_n} \left\{ KL(f_{D_i}||f_{D_0}) + KL(f_{D_i}||f_{D_0}) \right\}, \tag{4.4}$$

Clearly $Inv(f) \geq 0$ with equality exactly when f is invariant. The behavior of Inv in the general case is quite complicated. However, for n-variate Gaussian distributions, Inv can be nicely simplified. First of all, for n-variate Gaussian distributions g and h with the same covariance matrix Σ (assumed to be invertible) and mean vectors μ_g and μ_h,

$$KL(g||h) + KL(h||g) = (\mu_g - \mu_h)' \Sigma^{-1} (\mu_g - \mu_h). \tag{4.5}$$

Second of all, we have the following theorem.

Theorem 4.5.3 *Let D be any matrix in $\mathbb{I}_n$.*

1. *The covariance matrix of f_D is $W \Lambda_V W'$ where Λ_V is the eigenvalue matrix of Σ_V.*
2. *The mean vector of f_D is $WDZ'\mu_V$ where μ_V is the mean vector of f (the p.d.f. of V).*
3. *If f is multivariate Gaussian, then f_D is also multivariate Gaussian.*

Proof: Follows directly from [90, Theorem 5.16]. □

This theorem along with Equations 4.4 and 4.5 allows us to simplify our invariance metric in the case where f is a multi-variate Gaussian. Let μ_0 denote the mean vector of f_{D_0} and μ_i the mean vector of f_{D_i}. We have

$$
\begin{aligned}
Inv(f) &= \min_{D_i \neq D_0 \in \mathbb{I}_n} (\mu_i - \mu_0)' \Sigma_V^{-1} (\mu_i - \mu_0) \\
&= \min_{D_i \neq D_0 \in \mathbb{I}_n} \mu_V' (ZD_iW' - ZD_0W')(W\Lambda_V W')^{-1}(WD_iZ' - WD_0Z')\mu_V \\
&= \min_{D_i \neq D_0 \in \mathbb{I}_n} \mu_V' Z(D_i - D_0)\Lambda_V^{-1}(D_i - D_0)Z'\mu_V.
\end{aligned}
$$

Clearly, $Inv(f)$ goes to zero with μ_V. And, if we consider a simple path to zero – along a straight line – the behavior of $Inv(f)$ can be nicely characterized. Consider some fixed $\mu \in \mathbb{R}^n$. Given $\alpha \geq 0$, define μ_V as $\alpha\mu$. We have that

$$
Inv(f) = \alpha^2 \min_{D_i \neq D_0 \in \mathbb{I}_n} \left(\mu' Z(D_i - D_0)\Lambda_V^{-1}(D_i - D_0)Z'\mu\right).
$$

Hence we see that $Inv(f)$ approaches zero quadratically as $\mu_V \rightarrow 0$ along the line defined by $\alpha\mu$. With this result we can carry out experiments to measure the effect of the degree to which f is invariant on the attacker's success probability. We generated a dataset by sampling each tuple independently from a three-dimensional Gaussian with covariance $\begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 40 \end{pmatrix}$ and mean vector $\mu_V = \alpha(1,1,1)'$. Note that the minimum eigen-ratio is 20, sufficiently large to isolate the effect of decreasing invariance on attacker's success probability. We vary the value of α from 0 to 10. The original dataset contains $10,000$ tuples. We fix the sample ratio to be 2%, and relative error bound $\epsilon = 0.05$. Figure 4.15 shows that as α approaches zero (the mean approaches zero accordingly), the probability of privacy breach drops to zero too; however, as α runs away from zero, the probability of privacy breach increases very fast.

4.6 Independent Signals Attack

In this scenario, we assume that the data is a collection of signals. All the signals, at any given time, are statistically independent and all the signals are non-Gaussian with the exception of one. The attacker obtains the perturbed data, and the goal is to recover the original signals. In this section, we propose an Independent Component Analysis (ICA)-based attack technique to do this job.

4.6.1 Independent Component Analysis (ICA) Preliminaries

Independent Component Analysis (ICA) [91] is a technique for discovering independent hidden factors that are underlying a set of linear or nonlinear mixtures of some unknown variables, where the mixing system is also unknown. These unknown variables are assumed to be non-Gaussian and statistically independent, and they are called the independent components (ICs) of the observed data. This technique has been widely used for separation of artifacts in MEG (Magnetoencephalography) data, image noise reduction and telecommunications [92].

A classical example of ICA is the cocktail party problem (as illustrated in Figure 4.16). Imagine you are in a cocktail party, although different kinds of background sounds are mixed together, *e.g.*, music, other people's chat, television news report, or even a siren from a passing ambulance, you still have no problem identifying the discussion of your neighbors. It is not clear how human brains can separate the different sound sources. However, ICA is able to do it, if there are at least as many 'ears' or receivers in the room as there are different simultaneous sound sources.

The basic ICA model can be defined as follows:

$$y(t) = Ax(t), \tag{4.6}$$

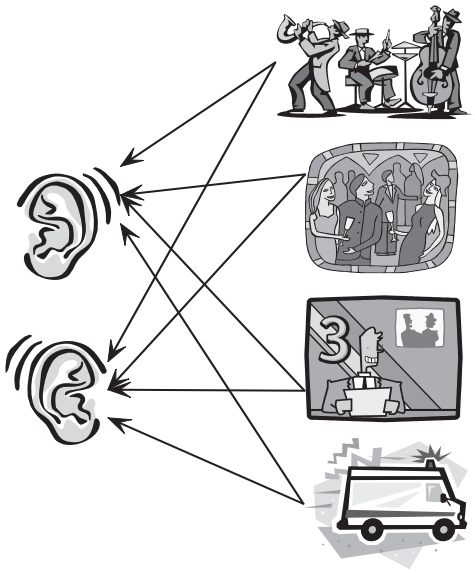


FIG. 4.16. An illustration of the cocktail party problem. What we have heard in a cocktail party are just linear (or nonlinear) combinations of different source audio signals.

where $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$ denotes an n-dimensional vector collecting the n independent source signals $x_i(t), i = 1, 2, \ldots, n$. Here t indicates the time dependence. Each signal $x_i(t)$ can be viewed as an outcome of a continuous-value random process. A is a $k \times n$ unknown mixing matrix, which can be viewed as a mixing system with k receivers. The observed mixture is $y(t) = (y_1(t), y_2(t), \ldots, y_k(t))^T$. The aim of ICA is to design a filter that can recover the original signals from only the observed mixture. Because $y(t) = Ax(t) = (A\Lambda P)(P^{-1}\Lambda^{-1}x(t))$ for any diagonal matrix Λ and permutation matrix P, the recovered signals $x(t)$ can never have completely unique representation. So, the uniqueness of the recovered signals found by ICA can only be guaranteed up to permutation and scaling ambiguities.

In practice, a linear filter is designed to get the recovered signals $\hat{x}(t) = (\hat{x}_1(t), \hat{x}_2(t),$

$\ldots, \hat{x}_l(t))^T$ from a k-dimensional input $y(t) = (y_1(t), y_2(t), \ldots, y_k(t))^T$. In other words,

$$\hat{x}(t) = By(t), \tag{4.7}$$

where B is an $l \times k$ dimensional separating matrix. Combining Eq. 5.12 and Eq. 5.13, we get

$$\hat{x}(t) = BAx(t) = Zx(t), \tag{4.8}$$

where $Z = BA$ is an $l \times n$ matrix. Each element of $\hat{x}(t)$ is thus a linear combination of $x_i(t)$ with weights given by $z_{i,j}$, where $z_{i,j}$ denotes the (i, j)-th entry of Z.

Many ICA algorithms start with whitening the data, *i.e.*, removing any correlations in the observed data $y(t)$. The source signals can then be found by an orthogonal transformation of the whitened signals. The appropriate transformation is sought by maximizing the independence of the signals. A review of different metrics for measuring independence can be found in [92].

In general, by imposing the following fundamental restrictions [92], all the source signals can be separated out up to scaling and permutation ambiguities:

- The source signals are statistically independent, *i.e.*, their joint probability density function (PDF) $f_{x(t)}(x_1(t),$
  $x_2(t), \ldots, x_n(t))$ is factorizable in the following way:

  $$f_{x(t)}(x_1(t), x_2(t), \ldots, x_n(t)) \;=\; \prod_{i=1}^{n} f_{x_i(t)}(x_i(t)),$$

  where $f_{x_i(t)}(x_i(t))$ denotes the marginal probability density of $x_i(t)$.

- All the signals must be non-Gaussian with possible exception of one signal.

- The number of observed signals k must be at least as large as the independent source

signals, *i.e.*, $k \geq n$.

- Matrix A has full column rank.

These restrictions have actually exposed the potential dangers of orthogonal transformation-based perturbation where the mixing matrix is square and has full column rank. The next section gives the ICA attack algorithm.

4.6.2 Independent Signal Attack (ICA Attack) Algorithm

We assume the data is a collection of signals, where each row of X, denoted by $x_i, i = 1, \ldots n$, represents one signal. Each signal can be viewed as an outcome of a continuous-value random process $x_i(t)$, where t indicates the time dependence. The data owner publishes $Y = M_T X$ where M_T is an unknown orthogonal matrix. The attacker obtains Y, and the goal is to recover X.

The attacker has some additional prior knowledge as follows: 1) The signals are statistically independent, *i.e.*, $\forall t$, the joint p.d.f. $f(x_1(t), \ldots, x_n(t)) = \prod_{i=1}^{n} f_(x_i(t))$, where $f(x_i(t))$ denotes the marginal probability density of $x_i(t)$. This assumption makes sense in situations where each signal arises from unrelated sources, *e.g.*, voice audio signals from people in different conversations or pixel vectors from unrelated pictures. 2) All the signals must be non-Gaussian with the possible exception of one signal. Algorithm 4.6.2.1 gives the basic procedure of ICA-based attacks. The next section demonstrates the performance of ICA-based attack in experiments.

4.6.3 Experiments

To demonstrate how ICA could attack the orthogonal transformation-based perturbation when data is statistically independent and not Gaussian, we chose both image and audio data for the experiments.

Algorithm 4.6.2.1 ICA-based Attack Technique

Inputs: $Y = M_T X$ where M_T is an unknown orthogonal matrix; X is an unknown matrix where each row represents one signal. All the signals are statistically independent. All the signals are non-Gaussian with possible exception of one signal.

Outputs The recovered data $\hat{X}$.

1: The attacker uses independent component analysis (ICA) to recover the original signals up to a scaling factor and row permutation.

First, we considered a dataset X consisting of four signals (four rows). Each is a picture of a natural scene represented by a 450×338 pixel grid – the top row of Figure 4.17. Each grid is stretched out into a length $152,100$ row vector. The perturbed versions, rows of $Y = M_T X$ for a randomly generated orthogonal matrix M_T, can be seen in the middle row of Figure 4.17. These appear to disguise the originals quite well. However, after applying ICA, the attacker produces estimates as seen in the bottom row of Figure 4.17. Due to the scaling factor, the colors do not match, and due to the row permutation, the estimated figures appear in a different order than the originals. However, the content of the original figures can be seen quite well.

FIG. 4.17. Performance of ICA on image data. The first row – original images; the second row – perturbed images; and the third row – recovered images.

Second, we considered four statistically independent audio signals, denoted as a $4 \times$

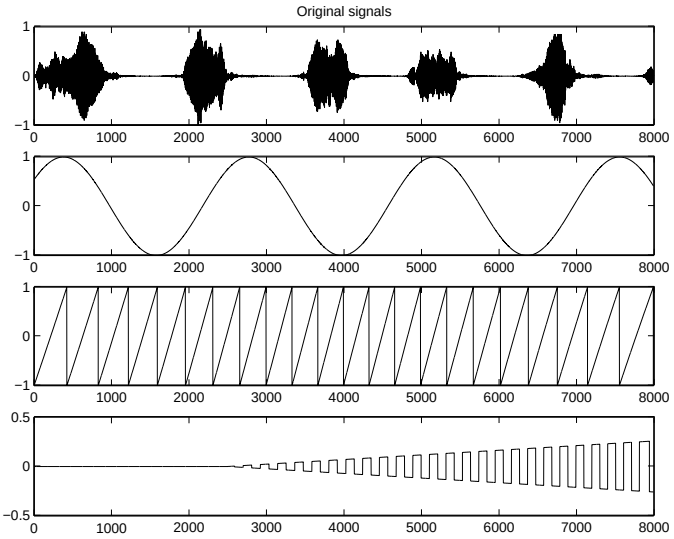


FIG. 4.18. A plot of four independent audio signals.

$13,129$ matrix X (shown in Figure 4.18). A perturbation of these signals (shown in Figure 4.19) is generated by pre-multiplying a 4×4 orthogonal matrix to X. The goal of ICA is to recover the original signals using only the perturbed data. Figure 4.20 gives the estimated signals through ICA. It can be seen that although the order and amplitude of the recovered signals are not necessarily the same as those of the original ones, the basic structure of the original signals are recovered very well.

4.6.4 Effectiveness of the Attack

Because $Y = M_T \Lambda P P^{-1} \Lambda^{-1} X$ for any diagonal matrix Λ and permutation matrix P, ICA can only recover the original signals up to permutation and scaling ambiguities. However, in many application scenarios, *e.g.*, when the data are natural images or audio signals, these ambiguities do not cause significant trouble identifying the contents of original signals, and the recovered data might be sufficient to breach privacy. The experiments in the last section validate the effectiveness of ICA attack.

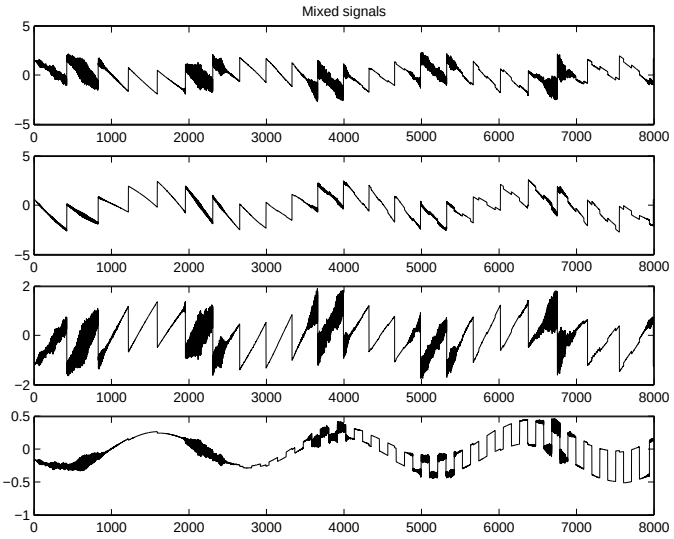


FIG. 4.19. Perturbation of the original signals using a orthogonal matrix.

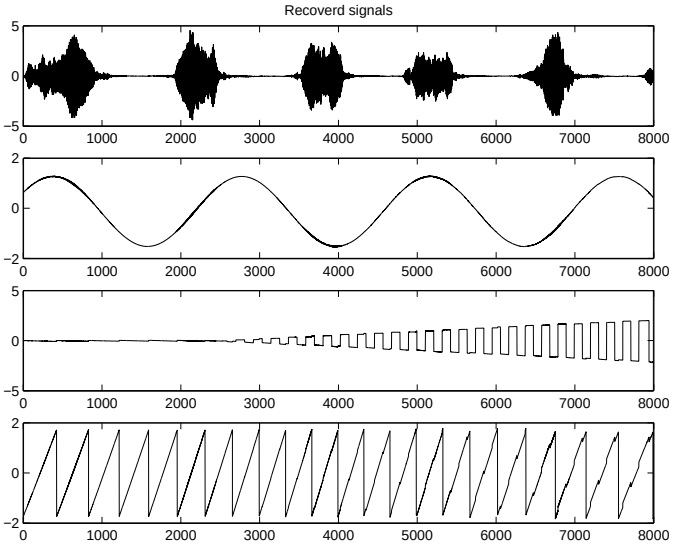


FIG. 4.20. Recovered signals using ICA.

Note that if some of the source signals are correlated, they may be lumped in the same group and can never be separated out. If there is more than one Gaussian signal, the problem becomes more complicated. The output of the filter may be either individual non-Gaussian signals, individual Gaussian signals, or a mixture of Gaussian signals. Detailed analysis can be found elsewhere [93].

4.7 Summary

In this chapter, we considered the use of distance-preserving maps (with origin fixed) as a data perturbation technique for privacy preserving data mining. On the one hand, this technique is quite useful as it allows many interesting data mining algorithms to be applied directly to the perturbed data and produce an error-free result, *e.g.*, K-means clustering and K-nearest neighbor classification. On the other hand, the privacy offered by distance preserving transformations has, to our knowledge, not been well-studied. We take a step in this direction by considering three types of prior knowledge an attacker may have and use to design attack techniques to recover the original data. The first is based on basic properties of linear algebra, the second on principal component analysis, and the third on independent component analysis. Our analysis explicitly illuminates scenarios where privacy can be seriously breached. As such, valuable information is gained into the effectiveness of distance preserving transformation for privacy preserving data mining.

4.8 Appendix

4.8.1 Appendix I

Theorem 4.4.1: Let $\mathbb{P}$ denote $\{M_T U_k U_k' + M_T U_{n-k} P U_{n-k}' : \forall P \in \mathbb{O}_{n-k}\}$. We have that $\mathbb{M}(X_k, Y_k) = \mathbb{P}$.

Proof: First we show that $\mathbb{M}(X_k, Y_k) = \mathbb{M}(U_k, M_T U_k)$, *i.e.*, any orthogonal matrix M that

satisfies condition $MX_k = Y_k$ also satisfies $MU_k = M_T U_k$, and vice versa. Because U_k is the orthonormal basis of $Col(X_k)$, there exists an invertible $k \times k$ matrix B such that $X_k B = U_k$. For any $M \in \mathbb{O}_n$, we have

$$\begin{aligned} M \in \mathbb{M}(X_k, Y_k) &\Leftrightarrow MX_k = Y_k \\ &\Leftrightarrow MX_k B = M_T X_k B \\ &\Leftrightarrow MU_k = M_T U_k \\ &\Leftrightarrow M \in \mathbb{M}(U_k, M_T U_k). \end{aligned}$$

We conclude that $\mathbb{M}(X_k, Y_k) = \mathbb{M}(U_k, M_T U_k)$.

Now we complete the proof by showing that $\mathbb{M}(U_k, M_T U_k) = \mathbb{P}$. We first show that $\forall M_P \in \mathbb{P}$, $M_P \in \mathbb{M}(U_k, M_T U_k)$. After that we will prove $\forall M \in \mathbb{M}(U_k, M_T U_k)$, $M \in \mathbb{P}$.

(1) For any $M_P \in \mathbb{P}$, we have:

$$\begin{aligned} M_P' M_P &= U_k U_k' M_T' M_T U_k U_k' + U_k U_k' M_T' M_T U_{n-k} P U_{n-k}' \\ &+ U_{n-k} P' U_{n-k}' M_T' M_T U_k U_k' + U_{n-k} P' U_{n-k}' M_T' M_T U_{n-k} P U_{n-k}' \\ &= U_k U_k' + 0 + 0 + U_{n-k} U_{n-k}' \\ &= [U_k | U_{n-k}] \begin{bmatrix} U_k' \\ U_{n-k}' \end{bmatrix} \\ &= UU' = I_n. \end{aligned}$$

The above equations reply on the fact that $U_{n-k}' U_k = U_k' U_{n-k} = 0$. Therefore, M_P is orthogonal. Also observe that

$$\begin{aligned} M_P U_k &= M_T U_k U_k' U_k + M_T U_{n-k} P U_{n-k}' U_k \\ &= M_T U_k + 0. \end{aligned}$$

Hence, $M_P \in \mathbb{M}(U_k, M_T U_k)$, so, $\mathbb{P} \subseteq \mathbb{M}(U_k, M_T U_k)$.

(2) Now consider $M \in \mathbb{M}(U_k, M_T U_k)$. We assert that $Col(M_T U_{n-k}) = Col(M U_{n-k})$ (to be proved later). Based on this assertion there exists $(n-k) \times (n-k)$ matrix P with $M_T U_{n-k} P = M U_{n-k}$. Observe that

$$\begin{aligned} P'P &= P'(M_T U_{n-k})'(M_T U_{n-k})P \\ &= (M_T U_{n-k} P)'(M_T U_{n-k} P) \\ &= (M U_{n-k})'(M U_{n-k}) \\ &= I_{n-k}. \end{aligned}$$

Thus, P is orthogonal. Moreover,

$$\begin{aligned} MU &= M[U_k | U_{n-k}] \\ &= [M_T U_k | M U_{n-k}] \\ &= [M_T U_k | M_T U_{n-k} P]. \end{aligned}$$

Thus,

$$\begin{aligned} M &= [M_T U_k | M_T U_{n-k} P] U' \\ &= [M_T U_k | M_T U_{n-k} P] \begin{bmatrix} U'_k \\ U'_{n-k} \end{bmatrix} \\ &= M_T U_k U'_k + M_T U_{n-k} P U'_{n-k}. \end{aligned}$$

Therefore, $M \in \mathbb{P}$, so, $\mathbb{M}(U_k, M_T U_k) \subseteq \mathbb{P}$.

All that remains is to prove the assertion: $Col(M_T U_{n-k}) = Col(MU_{n-k})$. Because $(MU_{n-k})'(MU_k) = 0$, then $Col(MU_{n-k}) \subseteq Col_\perp(MU_k)$. Because MU_{n-k} and MU_k are orthogonal, then the Fundamental Theorem of Linear Algebra implies that $Col_\perp(MU_k)$ and $Col(MU_{n-k})$ have the same dimension ($n-k$), thus, $Col(MU_{n-k}) = Col_\perp(MU_k)$. By replacing "$M$" with "$M_T$" in the previous two sentences, we also conclude that $Col(M_T U_{n-k}) = Col_\perp(M_T U_k)$. Finally, because $M \in \mathbb{M}(U_k, M_T U_k)$, then $Col(MU_k) = Col(M_T U_k)$, thus, $Col_\perp(MU_k) = Col_\perp(M_T U_k)$. It follows that $Col(M_T U_{n-k}) = Col_\perp(M_T U_k) = Col_\perp(MU_k) = Col(MU_{n-k})$. □

4.8.2 Appendix II

Preliminaries: Recall some definitions. For real number $\alpha \geq 0$, and integer $p \geq 1$, let $S_p(\alpha)$ denote the hyper-sphere in $\mathbb{R}^p$ centered at the origin with radius α *i.e.* $\{x \in \mathbb{R}^p : ||x|| = \alpha\}$. For any $\mathcal{A} \subseteq S_p(\alpha)$, $SA(\mathcal{A})$ denotes the surface area of $\mathcal{A}$ (assuming it is defined). To define surface area recall that a point $(x_1, \ldots, x_p) \in S_p(\alpha)$ can be written in hyper-spherical coordinates $0 \leq \theta_i \leq \pi$ (for $1 \leq i \leq p-2$) and $0 \leq \theta_{p-1} \leq 2\pi$ such that $x_1 = \alpha cos(\theta_1)$, $x_2 = \alpha sin(\theta_1) cos(\theta_2)$, ..., $x_{p-1} = \alpha sin(\theta_1) \cdots sin(\theta_{p-2}) cos(\theta_{p-1})$, and $x_p = \alpha sin(\theta_1) \cdots sin(\theta_{p-2}) sin(\theta_{p-1})$.

Let $\Pi_i(\mathcal{A})$ denote the projection of $\mathcal{A}$ onto the i^{th} hyper-spherical coordinate (for

$1 \leq i \leq p-1$). The surface area, $SA(\mathcal{A})$, of (A) is defined to be

$$\alpha^{p-1} \int_{\theta_1 \in \Pi_1(\mathcal{A})} \cdots \int_{\theta_{p-1} \in \Pi_{p-1}(\mathcal{A})} \delta(z) sin^{p-2}(\theta_1) sin^{p-3}(\theta_2) \cdots sin(\theta_{p-2}) d\theta_1 \cdots d\theta_{p-1},$$

(provided the integral exists) where z denotes $(\alpha, \theta_1, \ldots, \theta_{p-1})$ and $\delta(z)$ equals one if $z \in \mathcal{A}$; zero otherwise.

Results: For $w \in \mathbb{R}^p$ and $d \geq 0$, let $S_p(w,d)$ denote the portion of $S_p(||w||)$ whose distance from w is no larger than d, *i.e.* $S_p(w,d) = \{z \in S_p(||w||) : ||z-w|| \leq d\}$. For any $\mathcal{A} \subseteq S_p(||w||)$, let $\mathbb{O}(\mathcal{A})$ denote $\{P \in \mathbb{O}_p\text{: } P'w \in \mathcal{A}\}$. In this section, we prove the following two statements.

1. $\mu(\mathbb{O}(S_p(w,d))) = \frac{SA(S_p(w,d))}{SA(S_p(||w||))}$.
2. $\frac{SA(S_p(w,d))}{SA(S_p(||w||))} = \left(\frac{1}{\pi}\right) 2arcsin\left(\frac{d}{2||w||}\right)$ if $d \leq 2||w||$; 1 otherwise.

Because $S_1(||w||)$ equals two points (one if $||w|| = 0$), the results are obvious. Assume $p \geq 2$.

Statement 1: The proof of this fact is follows directly from basic properties of measure theory. Because it is a tangent from the primary focus of the paper, it is omitted.

Statement 2: Because $SA(z_1, d)$ equals $SA(z_2, d)$ for any $z_1, z_2 \in S_p(||w||)$, then it suffices to prove the desired result for $w_1 = ||w||e_1$ where e_1 is the first unit vector $(1, 0, \ldots, 0)'$. If $d > 2||w_1||$, all of $SA(||w_1||)$ is within d of w_1. Thus, the surface area ratio equals one as desired.

Assume $0 \leq d \leq ||w_1||\sqrt{2}$. Consider the hyper-plane $\{(x, y, 0, \ldots, 0) \in \mathbb{R}^p\}$. Figure 4.21 depicts the intersection of this hyper-plane with $S_p(||w_1||)$. It can be shown that $\frac{SA(S_p(w_1,d))}{SA(S_p(||w_1||))}$ equals $\frac{a}{\pi}$. Moreover, consider triangle ABC. It is a right triangle with hy-

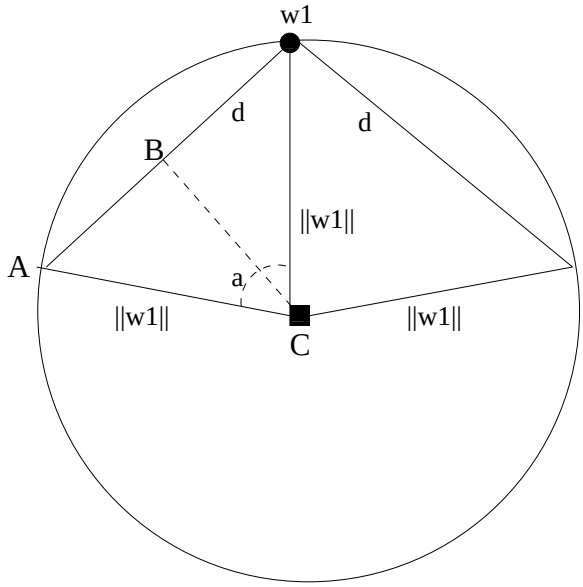


FIG. 4.21. Hyper-plane intersection with $S_p(||w||)$.

potenuse length $||w_1||$ and an angle $\frac{a}{2}$ with opposite side length $\frac{d}{2}$. Therefore, $sin(\frac{a}{2}) = \frac{d}{2||w_1||}$. So, $a = 2arcsin(\frac{d}{2||w_1||})$, yielding the desired result.

Finally, assume $\sqrt{2}||w_1|| < d \leq 2||w_1||$. Figure 4.22 depicts the intersection of the hyper-plane $\{(x, y, 0, \ldots, 0) \in \mathbb{R}^p\}$ with $S_p(||w_1||)$. It can be shown that $\frac{SA(S_p(w_1,d))}{SA(S_p(||w_1||))}$ equals $1 - \frac{a}{\pi}$ and $a = \pi - b$. Consider the right triangle ABC; it has hypotenuse length $||w_1||$ and an angle $\frac{b}{2}$ with opposite side length $\frac{d}{2}$. So, $b = 2arcsin(\frac{d}{2||w_1||})$ leading to $a = \pi - 2arcsin(\frac{d}{2||w_1||})$. Thus, the surface area ratio equals

$$1 - \frac{\pi - 2arcsin(\frac{d}{2||w_1||})}{\pi} = \frac{2arcsin(\frac{d}{2||w_1||})}{\pi}.$$

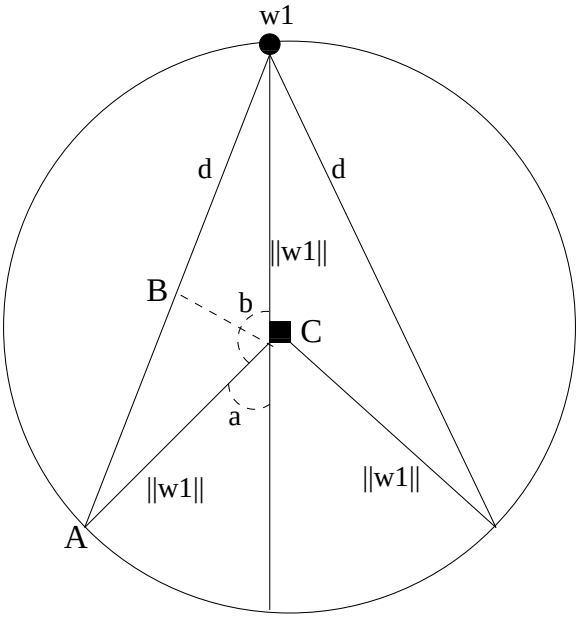


FIG. 4.22. Hyper-plane intersection with $S_p(||w||)$.

Chapter 5

RANDOM PROJECTION-BASED DATA PERTURBATION

This chapter considers a randomized multiplicative data perturbation technique for privacy preserving data mining. It is motivated by the work presented elsewhere [5–7, 19] that pointed out some security problems of additive perturbation and distance preserving perturbation. Specifically, this chapter explores the possibility of using multiplicative random projection matrices for constructing a new representation of the data. It can be proved that the inner product and Euclidean distance are preserved in the new data in the expectation. This approach is fundamentally based on the Johnson-Lindenstrauss lemma [94], which notes that any set of m points in n-dimensional Euclidean space can be embedded into an $O(\frac{\ln m}{\epsilon^2})$ dimensional space such that the pairwise distance of any two points is maintained with a high probability. Therefore, by projecting the data onto a lower dimensional random space, we can dramatically change its original form while preserving much of its distance-related characteristics. This chapter presents extensive theoretical analysis and experimental results on the accuracy and privacy of the random projection-based data perturbation technique.

The remainder of this chapter is organized as follows. Section 5.1 discusses the basic mathematical properties of random projection. It derives some error bounds for the ac-

curacy of the distances preserved by random projection. Section 5.2 demonstrates some privacy preserving data mining applications of the random projection-based data perturbation. Section 5.3 introduces a *Bayes privacy model* to measure the privacy offered by a perturbation technique. To be more specific, it considers the use of maximum a posteriori probability (MAP) estimate to recover the original data and to quantify the privacy. A closed-form expression about the (upper bound of the) privacy breach is derived, which can be used together with the error bounds to guide the perturbation in practice. Section 5.4 examines several privacy breach scenarios (some of which have been investigated in Chapter 4) and analyzes the efficacy of the corresponding attack techniques. Finally, Section 5.5 concludes this chapter.

5.1 Random Projection

This section gives the basic definition of random projection and its statistical properties.

5.1.1 Definition and Fundamental Properties

Random projection refers to the technique of projecting a set of data points from a high dimensional space to a randomly chosen lower dimensional space. Mathematically, let $X \in \mathbb{R}^{n\times m}$ be m data points in n-dimensional space. The random projection method multiplies X by a random matrix $R \in \mathbb{R}^{k\times n}$, reducing the n dimensions down to just k. It is well known that random projection preserves pairwise distances in the expectation. This technique has been successfully applied to a number of applications, for example, VLSI layout [95], nearest-neighbor search [96, 97], image and text clustering [98], distributed decision tree construction [99], motifs in bio-sequences [100] discovery, high-dimensional Gaussian mixture models learning [101], half spaces and intersections of half spaces learn-

ing [102].

The key idea of random projection arises from the Johnson-Lindenstrauss Lemma [94].

Lemma 5.1.1 (Johnson-Lindenstrauss Lemma) *[94] For any ϵ such that $0 < \epsilon < \frac{1}{2}$, and any set of points S in $\mathbb{R}^n$, with $|S| = m$, upon projection to a uniform random k-dimensional subspace where $k \geq \frac{9 \ln m}{\epsilon^2 - \frac{2}{3}\epsilon^3} + 1$, the following property holds: with probability at least $\frac{1}{2}$, for every pair $x, y \in S$,*

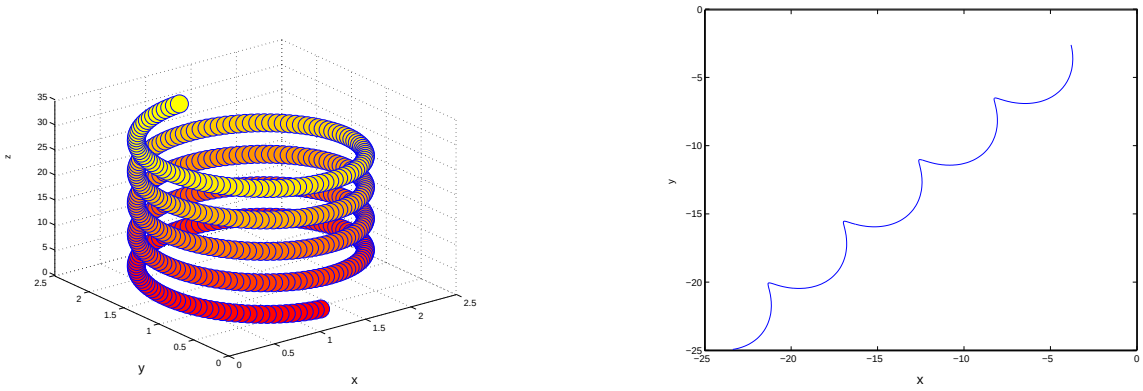
$$(1-\epsilon)||x-y||^2 \leq ||f(x)-f(y)||^2 \leq (1+\epsilon)||x-y||^2,$$

where $f(x)$, $f(y)$ are the projections of x and y.

This lemma shows that any set of m points in n-dimensional Euclidean space can be embedded into an $O(\frac{\ln m}{\epsilon^2})$ dimensional space such that the pairwise distance of any two points is maintained within a very small factor. This property implies that it is possible to change the data's original form by reducing its dimensionality while maintaining the pairwise inner products and Euclidean distances (see Figure 5.1(a), 5.1(b) as illustrative examples). In the next, we shall demonstrate how random matrices can be used for this kind of transformation.

Lemma 5.1.2 *Let R be a $p \times q$ random matrix such that each entry $r_{i,j}$ of R is independent and identically distributed (i.i.d.) according to some unknown distribution with mean zero and variance σ_r^2. Then,*

$$E[R^T R] = p\sigma_r^2 I, \text{ and } E[RR^T] = q\sigma_r^2 I.$$



(a) (b)

FIG. 5.1. (a) The original data. (b) The perturbed data after random projection, which maps the data from 3D space onto 2D space. The random matrix is chosen from N(0,1).

Proof: Let $r_{i,j}$ and $\epsilon_{i,j}$ be the i,j-th entries of matrix R and $R^T R$, respectively.

$$\begin{aligned} \epsilon_{i,j} &= \sum_{t=1}^{p} r_{t,i} r_{t,j}. \\ E[\epsilon_{i,j}] &= E[\sum_{t=1}^{p} r_{t,i} r_{t,j}] \\ &= \sum_{t=1}^{p} E[r_{t,i} r_{t,j}]. \end{aligned}$$

Because the entries of the random matrix are independent and identically distributed (i.i.d.),

$$E[\epsilon_{i,j}] = \begin{cases} \sum_{t=1}^{p} E[r_{t,i}]E[r_{t,j}] & \text{if } i \neq j; \\ \\ \sum_{t=1}^{p} E[r_{t,i}^2] & \text{if } i = j. \end{cases}$$

Now, note that $E[r_{i,j}] = 0$ and $E[r_{i,j}^2] = \sigma_r^2$; therefore,

$$E[\epsilon_{i,j}] = \begin{cases} 0 & \text{if } i \neq j; \\ p\sigma_r^2 & \text{if } i = j. \end{cases} \quad \text{So, } E[R^T R] = p\sigma_r^2 I.$$

Similarly, we have $E[RR^T] = q\sigma_r^2 I$. □

Intuitively, this result echoes the observation made elsewhere [103] that in a high-dimensional space, vectors with random directions are almost orthogonal. Lemma 5.1.2 can be used to prove the following results.

Lemma 5.1.3 (Random Projection) *Let $X \in \mathbb{R}^{n \times m}$ be a dataset of m data points in n-dimensional space. Let R be a $k \times n$ $(k < n)$ random matrix such that each entry $r_{i,j}$ of R is independent and identically distributed (i.i.d.) according to some unknown distribution with mean zero and variance σ_r^2. Further, let*

$$\begin{aligned} Y &= \frac{1}{\sqrt{k}\sigma_r} RX; \quad \textit{then} \\ E[Y^T Y] &= X^T X. \end{aligned} \tag{5.1}$$

This lemma shows that random projection preserves all pairwise inner products of X in the expectation. The beauty of this property is that the inner product is directly related to many other distance-related metrics. To be more specific, for any vectors $x, y \in \mathbb{R}^n$,

- The Euclidean distance of x and y is $||x - y||^2 = (x - y)^T (x - y)$.
- If the data vectors have been normalized to unity, then the cosine angle of x and y is

$$\cos\theta = \frac{x^T y}{||x|| \cdot ||y||} = x^T y.$$

- If the data vectors have been normalized to unity with zero mean, the sample correlation coefficient of x and y is

$$\rho_{x,y} = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sqrt{(\sum x_i^2 - \frac{(\sum x_i)^2}{m})(\sum y_i^2 - \frac{(\sum y_i)^2}{n})}} = x^T y.$$

Thus, if the data owner reduces the number of attributes of the data by projection, the inner products and Euclidean distances among the data records are still maintained. Therefore, we can directly apply common data mining algorithms to the new data without accessing the original sensitive information.

In the next subsection, we will derive some error bounds about the inner product and Euclidean distance preserved by the random projection.

5.1.2 Accuracy Analysis

As noted by Lemma 5.1.2, the entries of R (denoted by $r_{i,j}$, $_{i=1}^{n}{}_{j=1}^{m}$) should be i.i.d. with zero mean and constant variance. In fact, this is the only necessary condition for preserving the pairwise distances [104]. However, different choices of $r_{i,j}$ can change the variance of the errors. It is often convenient to let $r_{i,j}$ follow a symmetric distribution about zero with a constance variance. A simple distribution is the Gaussian distribution, *i.e.*, $r_{i,j} \sim N(0, \sigma_r^2)$. In this dissertation, unless stated otherwise, we will assume that the random entries follow the Gaussian distribution $N(0, \sigma_r^2)$. The following lemma gives the mean and variance of the projection error in the context of inner product computation.

Lemma 5.1.4 *Let x, y be two data vectors in $\mathbb{R}^n$. Let R be a $k \times n$ random matrix. Each entry of R is independent and identically distributed (i.i.d.) according to a Gaussian*

distribution with mean zero and variance σ_r^2. Further let

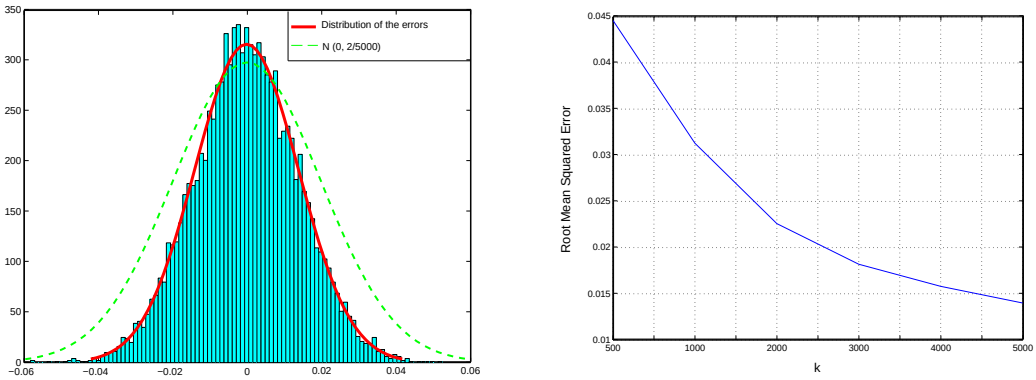
$$u = \frac{1}{\sqrt{k}\sigma_r}Rx, \quad \textit{and} \quad v = \frac{1}{\sqrt{k}\sigma_r}Ry \textit{ . Then}$$
$$E[u^T v - x^T y] = 0 \textit{ and}$$
$$Var[u^T v - x^T y] = \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 + (\sum_i x_i y_i)^2).$$

In particular, if both x and y are normalized to unity, $\sum_i x_i^2 \sum_i y_i^2 = 1$ and $(\sum_i x_i y_i)^2 \leq 1$. We have the upper bound of the variance as follows:

$$Var[u^T v - x^T y] \leq \frac{2}{k}.$$

Proof: Please see Appendix 5.6.1 for the proof. □

Lemma 5.1.4 shows that the error ($u^T v - x^T y$) of the inner product produced by the random projection-based perturbation technique is zero on average, and the variance is at most the inverse of the dimensionality of the reduced space multiplied by 2 if the original data vectors are normalized to unity. Actually, it can be proved that $\epsilon_{i,j}$ is approximately Gaussian [98]. The distortion also has an approximate Gaussian distribution with mean 0 and variance less than or equal to $2/k$. To validate the above claim, we chose a randomly generated dataset from a uniform distribution in $[0, 1]$ with $10,000$ records and 100 attributes. We normalized all the attributes to unity and compared their pairwise inner products before and after random projection. Figure 5.2(a) gives the results, which depict that even under 50% data projection rate (when $k = 5000$), the inner products still preserve very well after perturbation, and the errors approximately follow a Gaussian distribution with mean zero and variance less than $2/k$. Figure 5.2(b) shows the Root Mean Squared Error (RMSE) of the estimated inner product matrix with respect to the dimensionality of the reduced subspace. It can be seen that as k increases, the error decreases exponentially,



(a) (b)

FIG. 5.2. (a) Distribution of the error of the estimated inner products. The dataset contains $10,000$ records and 100 attributes. $k = 50\% \times 10000 = 5000$ (50% projection). The random matrix is chosen from $N(0, 2)$. Note that the variance of the error is even smaller than the variance of distribution $N(0, 2/k)$. (b) Root Mean Squared Error (RMSE) of the estimated inner products with respect to the dimensionality of the reduced subspace.

which means that the higher the dimensionality of the data, the better this technique works.

By applying Lemma 5.1.4 to the vector $(x - y)$, we have the following lemma to quantify the accuracy of the Euclidean distance preserved after random projection.

Lemma 5.1.5 *Let x, y be two data vectors in $\mathbb{R}^n$. Let R be a $k \times n$ random matrix. Each entry of R is independent and identically distributed (i.i.d.) according to a Gaussian distribution with mean zero and variance σ_r^2. Further let*

$$u = \frac{1}{\sqrt{k}\sigma_r} Rx, \quad \textit{and} \quad v = \frac{1}{\sqrt{k}\sigma_r} Ry \textit{ . Then}$$

$$E[||u - v||^2 - ||x - y||^2] = 0 \textit{ and}$$

$$Var[||u - v||^2 - ||x - y||^2] = \frac{2}{k}||x - y||^4 = \frac{2}{k}(\sum_i (x_i - y_i)^2)^2.$$

The above two lemmas show that one can compute both pairwise Euclidean distances and inner products in k-dimensional space (instead of n).

Next, we derive some formulae for the distribution of the projected data. Let u and v be k-dimensional vectors defined as before. It is easy to show that

$$\frac{u_i}{\sqrt{||x||^2/k}} \sim N(0,1), \qquad \frac{||u||^2}{||x||^2/k} \sim \chi^2_k;$$
$$\frac{u_i - v_i}{\sqrt{||x-y||^2/k}} \sim N(0,1), \qquad \frac{||u-v||^2}{||x-y||^2/k} \sim \chi^2_k,$$

where u_i and v_i are the i-th entry ($i = 1, \ldots, k$) of vector u and v, respectively, and χ^2_k is the chi-square distribution with k degrees of freedom. Knowing the distribution of the projected data enables us to derive sharp error bounds. The following lemma gives the closed-form expression of the accuracy for estimating the Euclidean distance.

Lemma 5.1.6 *Let x, y be two data vectors in $\mathbb{R}^n$. Let R be a $k \times n$ random matrix. Each entry of R is independent and identically chosen from a Gaussian distribution with mean zero and variance σ_r^2. Further let*

$$u = \frac{1}{\sqrt{k}\sigma_r} Rx, \qquad \textit{and} \qquad v = \frac{1}{\sqrt{k}\sigma_r} Ry \textit{ . Then}$$
$$Pr\{(1-\epsilon)||x-y||^2 \leq \ ||u-v||^2 \ \leq (1+\epsilon)||x-y||^2\} = \int_{k(1-\epsilon)}^{k(1+\epsilon)} f(t;k)dt,$$

where $f(t;k)$ is the probability density function of chi-square distribution with k-degrees of freedom.

$$f(t;k) \ = \ \begin{cases} \frac{1}{2^{k/2}\Gamma(k/2)} t^{k/2-1} e^{-t/2} & \textit{if } t > 0; \\ 0 & \textit{otherwise}. \end{cases}$$

Here $\Gamma(.)$ denotes the Gamma function: $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$.

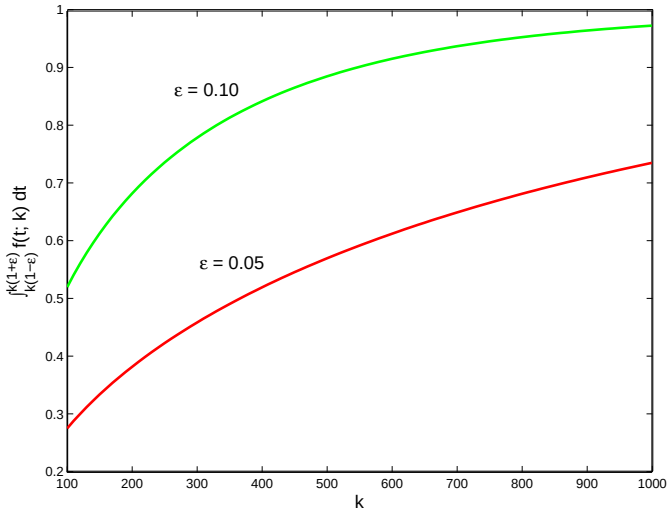


FIG. 5.3. The probability of the accuracy of random projection w.r.t. k and ϵ. Each entry of the random matrix is i.i.d., chosen from a Gaussian distribution with mean zero and constant variance.

Proof:

$$Pr\{(1-\epsilon)||x-y||^2 \leq ||u-v||^2 \leq (1+\epsilon)||x-y||^2\} = Pr\{k(1-\epsilon) \leq \frac{k||u-v||^2}{||x-y||^2} \leq k(1+\epsilon)\}.$$

The above equation implicitly assumes that $x \neq y$. Because $\frac{||u-v||^2}{||x-y||^2/k}$ follows a chi-square distribution with k degrees of freedom, we have

$$Pr\{k(1-\epsilon) \leq \frac{k||u-v||^2}{||x-y||^2} \leq k(1+\epsilon)\} = \int_{k(1-\epsilon)}^{k(1+\epsilon)} f(t;k)dt.$$

□

As an illustration, Figure 5.3 shows the actual probability of the accuracy with respect to different values of k and ϵ.

Similar results can be found elsewhere. For example, the work in [105] shows that if $k \geq \frac{4+2\gamma}{\epsilon^2/2-\epsilon^3/3}\log m$, then with probability at least $1 - m^{-\gamma}$, for any rows x, y, we have

$$(1-\epsilon)||x-y||^2 \leq ||u-v||^2 \leq (1+\epsilon)||x-y||^2.$$

The work in [104, Theorem 2] shows that

$$Pr\{(1-\epsilon)||x-y||^2 \leq \ \ ||u-v||^2 \ \ \leq (1+\epsilon)||x-y||^2\} \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}},$$

for any $0 < \epsilon < 1$. This result implies that as the reduced dimensionality k increases, the distortion drops exponentially, which echoes our previous observations that the higher the dimensionality of the data, the better the random projection works.

5.1.3 Variations of Random Projection

For the sake of completeness, we give a brief review on different variations of random projection in this section.

As we noted before, it is often convenient to let $r_{i,j}$, the entry of random matrix, follow a symmetric distribution about zero with constant variance. Roughly speaking, all such projections project the data onto a spherically random hyperplane though the origin. While this is conceptually simple, in practice, it amounts to multiplying the data matrix X with a dense matrix of real numbers. This can be a computationally expensive task in many real application scenarios. In his work, Achlioptas [105] asserted that one can replace projections onto spherically random hyperplanes with much simpler and faster operations.

Specifically, he proposed the use of the random matrix with i.i.d. entries defined as follows:

$$r_{i,j} = \sqrt{s} \begin{cases} 1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1 - \frac{1}{2}, \\ -1 & \text{with probability } \frac{1}{2s} \end{cases}$$

where Achlioptas used $s = 1$ or $s = 3$. Because the multiplication of $\sqrt{s}$ can be delayed, no floating point arithmetic is needed and all computation amounts to highly optimized database aggregation operations. When $s = 3$, one can achieve threefold speedup because only one third of the data need to be processed. Li *et al.* [106] further extended Achlioptas work by pointing out that the random entries can be chosen from $\{-1, 0, 1\}$ with probabilities $\{\frac{1}{2\sqrt{n}}, 1 - \frac{1}{\sqrt{n}}, \frac{1}{2\sqrt{n}}\}$ for achieving a significant $\sqrt{n}$-fold speedup, with little loss in accuracy.

Vempala [102] introduced a random projection technique that preserves the Hamming distance (which we will denote as $|.|_H$) among binary vectors. Mathematically speaking, let R be a $k \times n$ random matrix with each entry independently set to be 1 with probability p and 0 with probability $1 - p$. A vector x in $\mathbb{Z}_2^n$ is projected into a vector u in $\mathbb{Z}_2^k$ as $u = Rx$. Here, the arithmetic is carried out modulo 2, so we get a 0,1 vector. As the next lemma asserts, by choosing p appropriately, distance within a certain range can be preserved approximately; distances outside this range can only be distorted away from the range.

Lemma 5.1.7 *[102, Lemma 7.2] Let $0 \leq \epsilon \leq \frac{1}{2}$ and $1 \leq l \leq n$. Let each entry of a $k \times n$ matrix R be chosen independently to be 1 with probability $p = \epsilon^2/l$ and 0 with the rest. Let x, y be two vectors in $\mathbb{Z}_2^n$ and u, v be obtained as*

$$u = Rx \quad \textit{and} \quad v = Ry.$$

There is a constant C such that with probability at least $1 - 2e^{-C\epsilon^4 k}$,

- *If $|x-y|_H < \frac{l}{4}$, then $|u-v|_H < (1+\epsilon)kp\frac{l}{4}$.*
- *If $\frac{l}{4} \leq |x-y|_H \leq \frac{l}{2\epsilon}$, then $(1-\epsilon)kp \leq \frac{|u-v|_H}{|x-y|_H} \leq (1+\epsilon)kp$.*
- *If $|x-y|_H > \frac{l}{2\epsilon}$, then $|u-v|_H > (1-\epsilon)kp\frac{l}{2\epsilon}$.*

5.2 Privacy Applications of Random Projection

In this section, we demonstrate several privacy preserving data mining applications of the random projection-based perturbation technique. All the datasets we used for the experiments were chosen from the UCI Machine Learning Repository and KDD Archive without any normalization. The random matrices were generated from a Gaussian distribution with mean 0 and variance 4.

The application scenario can be defined as follows. Suppose there are N organizations $O_1, O_2, \ldots, O_N$; each organization O_i has a private transaction database DB_i. A third party data miner wants to learn certain statistical properties of the union of these databases $\bigcup_{i=1}^{N} DB_i$. These organizations are comfortable with this, but they are reluctant to disclose their raw data. This is generally referred to as the *census scenario* as we discussed in the previous chapters. Without loss of generality, we illustrate the application in both single-party-input and two-party-input scenarios (as shown in Figure 5.4).

5.2.1 Privacy Preserving Inner Product Computation from Distributed Data

Problem. Let X be an n-dimensional sensitive data vector owned by Alice and Y be an n-dimensional sensitive data vector owned by Bob. A third party wants to compute the inner product of these two vectors. None of these parties should know the others' private data.

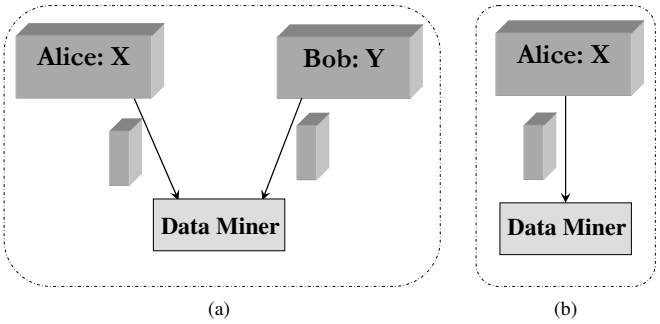


FIG. 5.4. (a) Distributed two-party-input computation model. (b) Single-party-input computation model.

Algorithm:

1. Alice and Bob cooperatively generate a secret random seed and use this seed to generate a $k \times n$ random matrix R.

2. Alice and Bob project their data onto $\mathbb{R}^k$ using R and release the perturbed version $U = \frac{1}{\sqrt{k}\sigma_r} RX$ and $V = \frac{1}{\sqrt{k}\sigma_r} RY$ to a third party.

3. The third party computes the inner product using the perturbed data U and V and gets $U^T V \approx X^T Y$.

Discussions: Similarly, the third party can compute the Euclidean distance on the perturbed data. When the data is properly normalized, the inner product matrix is nothing but the cosine angle or the correlation coefficient of X and Y.

Experiments: We considered the Adult database from the UCI Machine Learning Repository for the experiment. This data set was originally extracted from the 1994 census bureau database. Without loss of generality, we selected the first $10,000$ rows of the data with only two attributes (fnlwgt, education-num) and showed how the random projection preserves the inner product and (the square of) the Euclidean distance between these two attributes. Table 5.1 and 5.2 present the results over 20 runs. Here, k is the dimensionality

k	Mean(%)	Var(%)	Min(%)	Max(%)
100(1%)	9.91	0.41	0.07	23.47
500(5%)	5.84	0.25	0.12	18.41
1000(10%)	2.94	0.05	0.03	7.53
2000(20%)	2.69	0.04	0.01	7.00
3000(30%)	1.81	0.03	0.27	6.32

Table 5.1. Relative errors in computing the inner product of the two attributes.

k	Mean(%)	Var(%)	Min(%)	Max(%)
100(1%)	10.44	0.67	1.51	32.58
500(5%)	4.97	0.29	0.23	18.32
1000(10%)	2.70	0.05	0.11	7.21
2000(20%)	2.59	0.03	0.31	6.90
3000(30%)	1.80	0.01	0.61	3.91

Table 5.2. Relative errors in computing the square of the Euclidean distance of the two attributes.

of the perturbed vector, k also represents the percentage of the dimensionality of the original vector. It can be seen that when the vector is reduced to 30% of its original size, the relative error of the estimated inner product and (the square of) the Euclidean distance is only around 1.80%. Figure 5.5 illustrates how the original data and the perturbed data look alike.

5.2.2 Privacy Preserving K-Means Clustering from Distributed Data

Problem. Let X be an $n \times m_1$ data matrix owned by Alice and Y be an $n \times m_2$ matrix owned by Bob. A third party wants to do clustering on the union of these two data sets $(X : Y)$ without directly accessing the raw data.

Algorithm:

1. Alice and Bob cooperatively generate a secret random seed and use this seed to generate an $k \times n$ random matrix R.

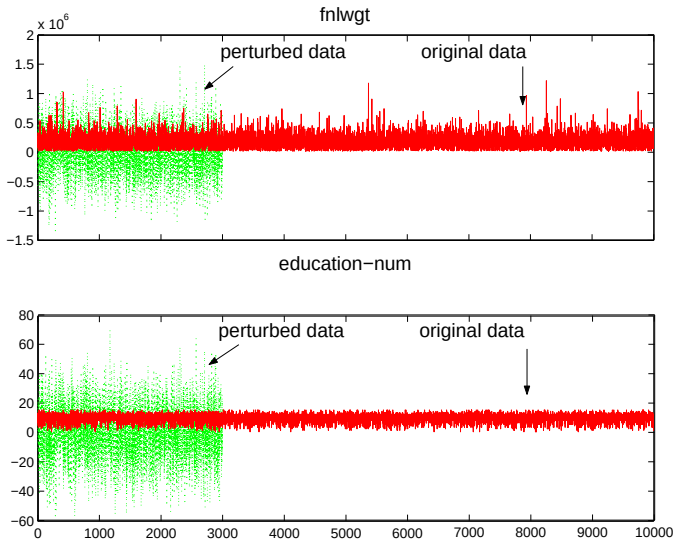


FIG. 5.5. Original data attributes and their perturbed counterparts. The random projection rate is 30 percent.

2. Alice and Bob project their data onto $\mathbb{R}^k$ using R and release the perturbed version $U = \frac{1}{\sqrt{k}\sigma_r} RX, V = \frac{1}{\sqrt{k}\sigma_r} RY$.

3. The third party does K-Means clustering over the data set $(U : V)$.

Discussions: The above algorithm is based on the fact that projection preserves the distance among vectors. Actually, random projection maps the data to a lower dimensional random space while maintaining much of its variance just like PCA. However, random projection only requires $O(mnk)(k << n)$ computations to project an $n \times m$ data matrix into $k \times m$ dimensions, while the computation complexity of estimating the PCA is $O(n^2 m) + O(n^3)$. This algorithm can be generalized for other distance-based data mining applications such as nested-loop outlier detection, k-nearest neighbor search, etc.

Experiments: For this task, we chose the Synthetic Control Chart Time Series data set from the UCI KDD Archive. This data set contains 600 examples of control charts, each with 60 attributes. There are six different classes of control charts: normal, cyclic, increas-

#Attributes	Clustered Instances						Error Rate
	1	2	3	4	5	6	
60 (Original data)	187	25	41	34	117	196	0.00%
30 (50% Projection)	188	25	40	34	117	196	0.17%
20 (33% Projection)	182	29	36	32	128	193	2.50%
10 (17% Projection)	182	19	65	36	108	190	4.33%

Table 5.3. K-Means clustering from the original data and the perturbed data.

ing trend, decreasing trend, upward shift and downward shift. We horizontally partitioned the data into two subsets, performed random projections, and then conducted K-Means clustering on the union of the projected data. Table 5.3 shows the results. It can be seen that even with a 17% projection rate (the number of attributes is reduced from 60 to 10), the clustering error rate is still as low as 4.33%.

5.2.3 Privacy Preserving Linear Classification

Problem. Given a collection of sensitive data points $x^{(i)}$ $(i = 1, 2, \ldots, m)$ in $\mathbb{R}^n$, each labelled as positive or negative, a third party data miner wants to find a weight vector w such that $w^T x^{(i)} > 0$ for all positive points $x^{(i)}$ and $w^T x^{(i)} < 0$ for all negative points $x^{(i)}$.

Algorithm:

1. The data owner generates a $k \times n$ random matrix R and projects the data to $\mathbb{R}^k$ using R such that $\hat{x}^{(i)} = \frac{1}{\sqrt{k}\sigma_r} R x^{(i)}$, $\forall i$, and releases the perturbed data.

2. The third party runs the perceptron algorithm in $\mathbb{R}^k$:

   (a) Let $\hat{w} = 0$. Do until all the examples are correctly classified

      i. Pick an arbitrary misclassified example $\hat{x}_i$ and let $\hat{w} \leftarrow \hat{w} + classlabel(\hat{x}^{(i)})\hat{x}^{(i)}$.

Discussions: Note that in this algorithm, the class labels are not perturbed. Future example x is labelled positive if $\hat{w}^T(\frac{1}{\sqrt{k}\sigma_r} R x) > 0$ and negative otherwise. This is actually

the same as checking whether $(\hat{w}^T \frac{1}{\sqrt{k}\sigma_r} R)x > 0$, namely, a linear separator in the original n-dimensional space. This also implies that $\hat{w}$ is nothing but the projection of w such that $\hat{w} = \frac{1}{\sqrt{k}\sigma_r} Rw$ and, therefore,

$$\hat{w}^T \hat{x}^{(i)} = \frac{1}{\sqrt{k}\sigma_r} w^T R^T \frac{1}{\sqrt{k}\sigma_r} Rx^{(i)} \approx w^T x^{(i)}.$$

This algorithm can be easily generalized for linear Support Vector Machine (SVM) because in the Lagrangian dual problem of the SVM task, the relationship of the original data points is completely quantified by inner product.

Experiments: We selected the Iris Plant Database from the UCI Machine Learning Repository. This is a very small data set with 150 instances and only 4 numeric attributes. Our experiments show that even for such a small data set, the algorithm still works well. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant (Iris-setosa, Iris-versicolor, Iris-virginica). We manually merged Iris-setosa and Iris-versicolor together so that we could do a binary classification on this data. The projection rate is 50%; hence, the data has only two attributes left after perturbation. We performed a voted perceptron learning on both the original data and the perturbed data. The accuracy on the original data over 10-fold cross validation is 94.67%. The classification results on the perturbed data over 10-fold cross validation are demonstrated in Table 5.4. It shows that the accuracy on the perturbed data over 10-fold cross validation is 86.67%, which is 91.55% as good as the results over the original data.

Accuracy(%)	1	2	3	4	5
	66.67	80.00	100.00	80.00	93.33
	6	7	8	9	10
	86.67	80.00	93.33	93.33	93.33
Mean(%)	86.67				
Std(%)	9.43				

Table 5.4. Classification on the perturbed iris plant data over 10-fold cross validation.

5.3 Bayes Privacy Model

In this section [1], we discuss a *Bayes privacy model* to measure the privacy offered by a perturbation technique. This model considers attacker's prior and posteriori beliefs about the data and uses Bayesian inference to evaluate the privacy. This model consists of three building blocks: 1) the definition of attacker's prior and posteriori beliefs; 2) the information non-disclosure principle; and 3) the implementation of the principle.

Attacker's Prior and Posteriori Beliefs: Let x be the unknown private data and y be the observed perturbed data. They can be viewed as the observations of two random vectors $\mathbf{x}$ and $\mathbf{y}$, respectively. Let $\boldsymbol{\theta}$ be the attacker's additional background knowledge. Further let $f_{\mathbf{x}}(x)$ be the probability density of $\mathbf{x}$ and $f_{\mathbf{x}|\mathbf{y},\boldsymbol{\theta}}(x|y,\theta)$ be the conditional probability density of $\mathbf{x}$ given $\mathbf{y} = y$ and $\boldsymbol{\theta} = \theta$. We can define the attacker's prior and posteriori belief about the private data as follows:

- Attacker's prior belief: $\alpha(x) = f_{\mathbf{x}}(x)$
- Attacker's posteriori belief: $\beta(x, y, \theta) = f_{\mathbf{x}|\mathbf{y},\boldsymbol{\theta}}(x|y,\theta)$.

Having the perturbed data and the additional background knowledge, the attacker

[1] Throughout this section, we use the **UPPER CASE BOLD LETTER** to represent a random matrix and the UPPER CASE REGULAR LETTER to represent an observation of a random matrix. We use the **lower case bold letter** to denote a random vector and the lower case regular letter to denote an observation of a random vector.

could possibly derive private information about the original data. Ideally, a secure perturbation technique should conform to the following principle.

Information Non-disclosure Principle: The perturbed data should provide the attacker with little additional information beyond the attacker's prior belief and other background knowledge.

Implementation of the Principle: This principle is universal, but, depending on the applications, it can be instantiated in several different ways to quantify the privacy offered by a perturbation technique. For example, we have the following possible choices.

1. The (ρ_1, ρ_2)-privacy breach [37] happens when $\alpha(x) < \rho_1$ and $\beta(x, y, \theta) > \rho_2$ or when $\alpha(x) > 1 - \rho_1$ and $\beta(x, y, \theta) < 1 - \rho_2$.

2. An alternate way is to measure the difference of the posteriori and the prior for a given x (*e.g.*, $\beta(x, y, \theta) - \alpha(x)$) or over all the possible x's (*e.g.*, $\max_x(\beta(x, y, \theta) - \alpha(x))$).

3. Another possible way is to compute the maximum a posteriori probability (MAP) estimate of $\mathbf{x}$ given $\mathbf{y} = y$ and $\boldsymbol{\theta} = \theta$:

$$\hat{x}_{MAP}(y, \theta) = \arg\max_x \beta(x, y, \theta) = \arg\max_x f_{\mathbf{x}|\mathbf{y},\boldsymbol{\theta}}(x|y, \theta).$$

   With this estimation, we can either compare $\hat{x}$ with the attacker's prior and background knowledge to see whether $\hat{x}$ offers any extra information. We can also compute (theoretically or empirically) the probability of an ϵ-privacy breach (see Definition 4.2.2), *i.e.*, $Prob\{||\hat{x} - \tilde{x}|| \leq ||\tilde{x}||\epsilon\}$, where $\tilde{x}$ is the original data that actually generates y through the perturbation.

The (ρ_1, ρ_2)-privacy breach [37] is a good metric to measure the information disclosure. However, it works only for discrete data. It assumes records of both private data and perturbed data are statistically independent, and it requires the transition probability (the

probability from one specific private data record to a specific perturbed data record) to be explicitly defined. These requirements make it difficult for quantifying privacy of multiplicative perturbation. In this dissertation, we use the maximum a posteriori probability (MAP) estimate to recover the original data and, therefore, to quantify the privacy offered by random projection-based perturbation. We choose MAP because 1) it has a solid statistics foundation; 2) it is closely related to maximum a posteriori probability hypothesis testing [90, Chapter 8]; 3)in the absence of a priori information, MAP estimate is equivalent to maximum likelihood estimate (MLE); 4) it often produces estimates with errors that are not much higher than the minimum mean square error; and 5) it is relatively easy to derive the conditional probability density function in the multiplicative data perturbation scenario.

Next, We will first discuss the MAP estimate with the assumption that the original data arose as a sample from a multivariate distribution. After that, we will generalize the results we have found to the matrix variate distribution scenario [107].

5.3.1 MAP Estimate for Multivariate Distribution

Let the original data have n attributes and m records. They can be considered as observations of a random vector of length n, denoted by $\mathbf{x} \in \mathbb{R}^n$. Let $\mathbf{R}$ be a $k \times n$ random matrix with each entry independent and identically chosen from $N(0,1)$. Let $\mathbf{y} = \frac{1}{\sqrt{k}}\mathbf{R}\mathbf{x}$. We also make the following assumptions:

Assumption 5.3.1 (The Attacker's Prior Belief about $\mathbf{x}$) *The attacker knows the range of each entry of* $\mathbf{x}$*, denoted by* $\mathbf{x}_i$*,* $i = 1, \ldots, n$*. In other words, the attacker knows that* $\mathbf{x}_i \in [a_i, b_i]$*. Without other information, the attacker further assumes that each entry* $\mathbf{x}_i$ *is independent and follows a uniform distribution with* $f_{\mathbf{x_i}}(x_i) = \frac{1}{b_i - a_i}$ *for* $a_i \leq x_i \leq b_i$*, and* $f_{\mathbf{x_i}}(x_i) = 0$ *otherwise.*

Assumption 5.3.2 (The Attacker's Additional Background Knowledge) *The attacker has no other background knowledge about the original data, that is,* $\boldsymbol{\theta} = \emptyset$.

We are interested in computing the maximum a posteriori probability (MAP) estimate of $\mathbf{x}$ given the observation $\mathbf{y} = y$:

$$\hat{x}_{MAP}(y) = \arg\max_x f_{\mathbf{x}|\mathbf{y}}(x|y). \tag{5.2}$$

Using the Bayesian rule, we get the following formulae:

$$\begin{aligned} \hat{x}_{MAP}(y) &= \arg\max_x f_{\mathbf{x}|\mathbf{y}}(x|y) \\ &= \arg\max_x \frac{f_{\mathbf{y}|\mathbf{x}}(y|x) f_{\mathbf{x}}(x)}{f_{\mathbf{y}}(y)} \\ &= \arg\max_x f_{\mathbf{y}|\mathbf{x}}(y|x) f_{\mathbf{x}}(x). \end{aligned} \tag{5.3}$$

Note that $\mathbf{y}$ is a k-dimensional random vector with $\mathbf{y}_i = \sum_{j=1}^{n} \mathbf{r}_{ij}\mathbf{x}_j$, $i = 1, \ldots, k$, where $\mathbf{r}_{ij}$ represents the entry on the i-th row and j-th column of $\mathbf{R}$. It can be proved that given $\mathbf{x} = x$, $\mathbf{y}$ follows a normal distribution with mean $\mu_{\mathbf{y}} = 0$ and covariance $\Sigma_{\mathbf{y}} = \frac{1}{k}\begin{pmatrix} x^T x & & \\ & \ddots & \\ & & x^T x \end{pmatrix}$. Therefore, we can write $f_{\mathbf{y}|\mathbf{x}}(y|x)$ as follows:

$$\begin{aligned} f_{\mathbf{y}|\mathbf{x}}(y|x) &= \frac{1}{(2\pi)^{k/2}|\Sigma_{\mathbf{y}}|^{1/2}} \exp\left(-\frac{1}{2}(y-\mu_{\mathbf{y}})^T \Sigma_{\mathbf{y}}^{-1}(y-\mu_{\mathbf{y}})\right) \\ &= \frac{k^{\frac{1}{2}}}{(2\pi x^T x)^{k/2}} \exp\left(-\frac{k y^T y}{2x^T x}\right). \end{aligned} \tag{5.4}$$

From Eq. 5.4 and Assumption 5.3.1, we have

$$f_{\mathbf{y}|\mathbf{x}}(y|x) f_{\mathbf{x}}(x) = \frac{k^{\frac{1}{2}}}{(2\pi x^T x)^{k/2}} \exp\left(-\frac{k y^T y}{2\sigma^2}\right) \frac{1}{b_1 - a_1}\frac{1}{b_2 - a_2} \cdots \frac{1}{b_n - a_n}.$$

Because the logarithm is a monotone one-to-one function, we can maximize the following function instead:

$$\log f_{\mathbf{y}|\mathbf{x}}(y|x) f_{\mathbf{x}}(x) = \ln \frac{k^{\frac{1}{2}}}{(2\pi x^T x)^{k/2}} + \left(-\frac{k y^T y}{2 x^T x}\right) + \ln \frac{1}{b_1 - a_1} + \cdots + \ln \frac{1}{b_n - a_n}. \quad (5.5)$$

To solve the optimization problem, let $C_1 = \frac{1}{2} \ln k - \frac{k}{2} \ln 2\pi$, let $C_2 = \Sigma_{i=1}^{n} \ln \frac{1}{b_i - a_i}$, Eq. 5.5 can be simplified as

$$-\frac{k}{2} \ln x^T x - \frac{k y^T y}{2 x^T x} + C_1 + C_2.$$

Further let $z = x^T x$, the function to be maximized becomes

$$-\frac{k}{2} \ln z - \frac{k y^T y}{2z}, \quad (5.6)$$

such that $l \leq z \leq u$, where $l = \sum_{i=1}^{n} a_i^2$ and $u = \sum_{i=1}^{n} b_i^2$.

Now we can draw a graph to see if and where this function has a maximum value in the region $[l, u]$. If it has a maximum at a point $z^* \in (l, u)$, we can set the derivative to zero in order to find z^*. In this case, it can be easily proved that $z^* = y^T y$, *i.e.*, any vector $\hat{x}$ that satisfies $\hat{x}^T \hat{x} = y^T y$ is the optimal solution. This is interesting because we know $E[\mathbf{y}^T \mathbf{y}] = x^T x$. Therefore, the maximum a posteriori estimation does not provide the attacker any more information about the private data than what has been implied by the properties of projection itself. If the function has an end-point maximum, either at $z = l$ or $z = u$, then, the derivative need not (and usually won't) vanish there. Having found z^*, our optimal solution is any point on the hyper-sphere $\sum_i^n \hat{x}_i^2 = z^*$.

In summary, under the assumptions 5.3.1 and 5.3.2, the random projection-based perturbation does not offer the attacker more information about the private data than what has been implied by the inner product preservation property itself. If the attacker has no prior

knowledge about the private data at all, the MAP estimate simply becomes a maximum likelihood estimate [90, pages 337–338]. If the prior has other distributions, we might not be able to derive a simple analytic solution to the maximization problem. In such situations, the MAP estimate must be sought using numerical methods. We will discuss that scenario in detail in Section 5.4.2.

5.3.2 Probability of ϵ-Privacy Breach

In the previous section, we proved that, under mild assumptions, any $\hat{x}$ that satisfies $\hat{x}^T\hat{x} = y^T y$ is the maximum a posteriori probability (MAP) estimate of the original data x given the perturbed data y. In other words, $\hat{x}$ can only be a point on the surface of a hyper-sphere centered at the origin with radius $||y|| = \sqrt{y^T y}$. In this section, we will compute the probability of ϵ-privacy breach $\rho(x, \epsilon)$ [2] when the attacker randomly chooses one such $\hat{x}$.

Let $S_n(||x||)$ denote the hyper-sphere in $\mathbb{R}^n$ centered at the origin with radius $||x||$. For any $\mathcal{A} \subseteq S_n(||x||)$, let $SA(\mathcal{A})$ denote the surface area of $\mathcal{A}$. Let $S_n(x, ||x||\epsilon)$ denote the portion of $S_n(||x||)$ whose distance from x is no larger than $||x||\epsilon$, *i.e.*, $S_n(x, ||x||\epsilon) = \{\hat{x} \in S_n(||x||) : ||\hat{x} - x|| \leq ||x||\epsilon\}$. The probability of privacy breach depends on the value of $x^T x$, where x is the original data point.

If $x^T x = y^T y$, then $S_n(||x||) = S_n(||y||)$ (the big hyper-sphere in Figure 5.6). The probability of privacy breach is the ratio of the surface area of the big hyper-sphere that is within the small hyper-sphere to the whole surface area of the big hyper-sphere. In Section 4.4.2 we derived the closed-form expression for the probability of ϵ-privacy breach for this

[2] Definition 4.2.2 defines the probability of ϵ-privacy breach $\rho(x, \epsilon) = Prob\{||\hat{x} - x|| \leq ||x||\epsilon\})$ for any $\epsilon > 0$.

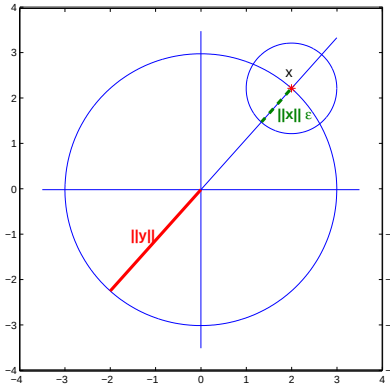


FIG. 5.6. MAP estimate when $x^T x = y^T y$.

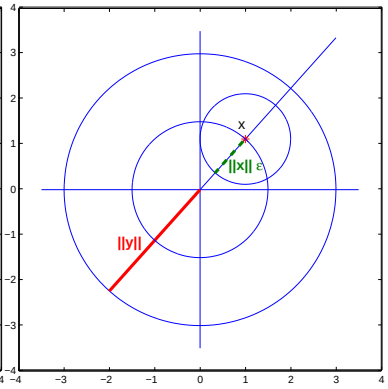


FIG. 5.7. MAP estimate when $x^T x < y^T y$.

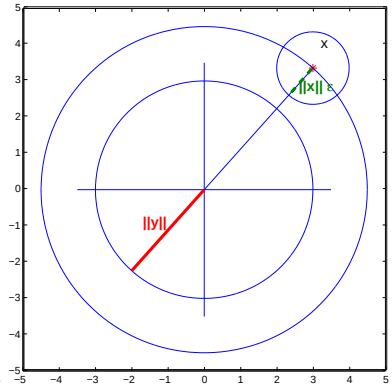


FIG. 5.8. MAP estimate when $x^T x > y^T y$.

scenario. Mathematically, we have

$$\begin{aligned} \rho(x,\epsilon) &= \frac{SA(S_n(x,||x||\epsilon))}{SA(S_n(||x||))} \\ &= \begin{cases} \left(\frac{1}{\pi}\right) 2arcsin\left(\frac{||x||\epsilon}{2||x||}\right) & \text{if } ||x||\epsilon < 2||x||; \\ 1 & \text{otherwise.} \end{cases} \end{aligned}$$

If $x^T x < y^T y$ or $x^T x > y^T y$, the ratio of the surface area is always smaller than the ratio when $x^T x = y^T y$. Specifically, for the case shown in Figures 5.7 and 5.8, we can see that the ratio is equal to 0. Therefore, the value of $\rho(x,\epsilon)$ when $x^T x = y^T y$ serves as an upper bound for the probability of ϵ-privacy breach.

5.3.3 Privacy/Accuracy Control

In the previous section, we derived the closed-form expression of the probability of ϵ-privacy breach (when $x^T x = y^T y$) and its upper bound (when $x^T x \neq y^T y$). The computation requires that the data owner knows both the original data x and the perturbed data y. However, in practice, the data owner usually wants to control the privacy and accuracy

tradeoff before actually performing the perturbation. In this section, we will discuss this possibility and offer guidelines for the data owner to perturb the data.

Privacy: As illustrated in Figure 5.9, if $||y|| < ||x|| - ||x||\epsilon$ or $||y|| > ||x|| + ||x||\epsilon$, none of the data points $\hat{x}$ on the surface of a hyper-sphere centered at the origin with radius $||y||$ will satisfy $||\hat{x} - x|| \leq ||x||\epsilon$; hence, the probability of ϵ-privacy breach will be 0. So, the question is what is the probability that $||y|| < ||x|| - ||x||\epsilon$ or $||y|| > ||x|| + ||x||\epsilon$.

In Section 5.1.2, we showed that

$$\frac{||y||^2}{||x||^2/k} \sim \chi^2_k.$$

Hence,

$$\begin{aligned}
&Prob\{||y|| < ||x|| - ||x||\epsilon \text{ or } ||y|| > ||x|| + ||x||\epsilon\} = \\
&Prob\{||y||^2 < (1-\epsilon)^2||x||^2 \text{ or } ||y||^2 > (1+\epsilon)^2||x||^2\} = \\
&Prob\{\frac{||y||^2}{||x||^2/k} < k(1-\epsilon)^2 \text{ or } \frac{||y||^2}{||x||^2/k} > k(1+\epsilon)^2\} = \\
&\int_{-\infty}^{k(1-\epsilon)^2} f(t;k)dt + \int_{k(1+\epsilon)^2}^{+\infty} f(t;k)dt, \qquad (5.7)
\end{aligned}$$

where $f(t;k)$ is the probability density function of the chi-square distribution with k degrees of freedom. Thus, Eq. 5.7 gives the probability that $\rho(x,\epsilon) = 0$ for a given ϵ and k.

Accuracy: Recall that Lemma 5.1.6 proved that for any data $x^{(1)}$, $x^{(2)}$ and their perturbed version $y^{(1)}$, $y^{(2)}$, we have

$$\begin{aligned}
&Pr\{(1-\eta)||x^{(1)} - x^{(2)}||^2 \leq ||y^{(1)} - y^{(2)}||^2 \leq (1+\eta)||x^{(1)} - x^{(2)}||^2\} = \\
&\int_{k(1-\eta)}^{k(1+\eta)} f(t;k)dt, \qquad (5.8)
\end{aligned}$$

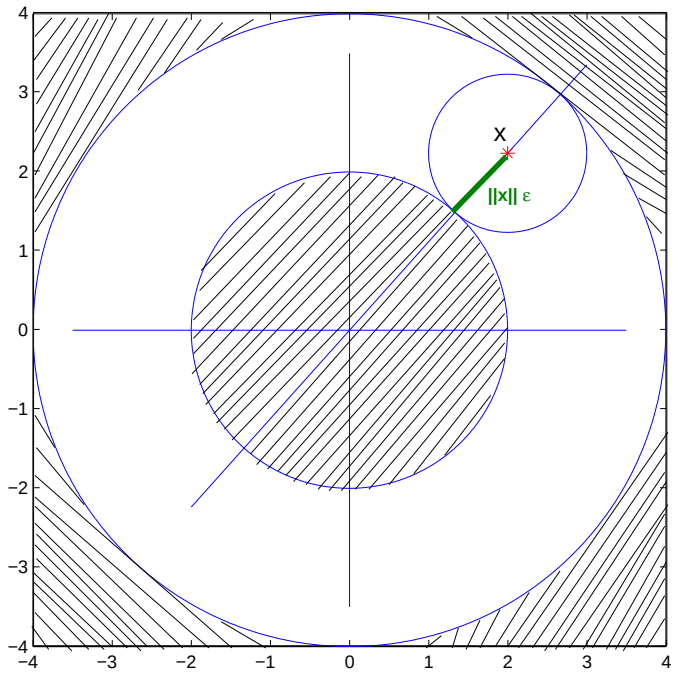


FIG. 5.9. The shaded area is $||y|| < ||x|| - ||x||\epsilon$ or $||y|| > ||x|| + ||x||\epsilon$.

where $\eta > 0$ and $f(t;k)$ is the probability density function of the chi-square distribution with k degrees of freedom. As either η or k increases, the probability increases (illustrated in Figure 5.3). Therefore, Eq. 5.8 gives the probability of the accuracy of random projection for a given η and k.

Combing Eq. 5.7 and Eq. 5.8, the data owner could setup privacy and accuracy thresholds (for a given ϵ and η) and determine the value of k such that both conditions are satisfied. As an illustration, let us look at Figure 5.10. This figure plots the probability of the accuracy (for a given $\eta = 0.10$) and the probability that $\rho(x, \epsilon) = 0$ (for a given $\epsilon = 0.01$) with respect to k. It can be seen that as k increases, the probability of the accuracy increases, but the probability of zero privacy breach decreases – a tradeoff between

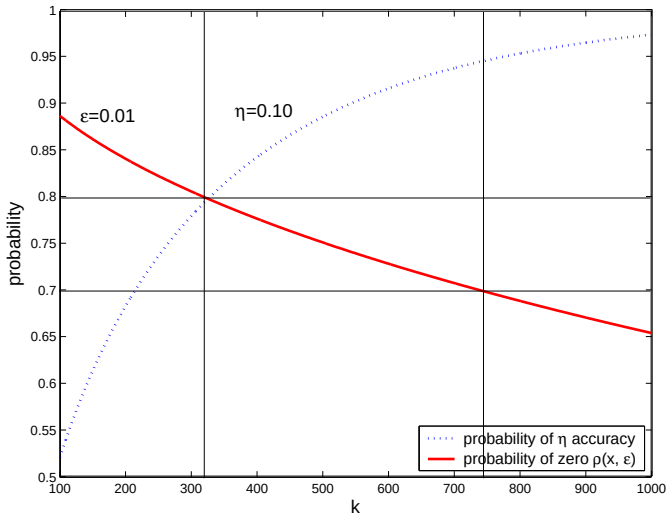


FIG. 5.10. Illustration of privacy and accuracy control.

accuracy and privacy. If, for example, the data owner wants

$$Pr\{(1-\eta)||x^{(1)}-x^{(2)}||^2 \leq ||y^{(1)}-y^{(2)}||^2 \leq (1+\eta)||x^{(1)}-x^{(2)}||^2\} \geq 80\%,\ \eta = 0.10,$$

then k should be greater than 320. If in the meantime, the data owner would like to achieve $\rho(x,\epsilon) = 0, \epsilon = 0.01$ with probability at least 70%, then k should be less than 750. Therefore, the data owner simply chooses a k in $(320, 750)$ and performs the perturbation.

5.3.4 MAP Estimate for Matrix Variate Distribution

In this section, we assume that the original data arose as a sample from a random matrix instead of a random vector. This allows the attacker to use the information of both row-wise and column-wise dependencies of the perturbed data and the original data. Next, we first present a brief introduction to some definitions and theorems from matrix algebra. Then, we will derive the closed-form expression of the MAP estimate for the matrix variate distribution.

Definition 5.3.3 *[108, pages 8] The Kronecker product of two matrices $A(m \times n) = (a_{i,j})$ and $B(p \times q) = (b_{i,j})$, denoted by $A \otimes B$, is the $mp \times nq$ matrix defined by*

$$A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,n}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1}B & a_{m,2}B & \cdots & a_{m,n}B \end{pmatrix}.$$

Definition 5.3.4 *[108, pages 9] For a matrix $X(n \times m)$, $vec(X)$ is the $nm \times 1$ vector defined as*

$$vec(X) = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{pmatrix},$$

where $x^{(i)}$, $i = 1, \ldots, m$ is the i-th column of X.

Definition 5.3.5 *[108, pages 55] The random matrix $\mathbf{R}(k \times n)$ is said to have a matrix variate Gaussian distribution with mean matrix $M(k \times n)$ and covariance matrix $\Sigma \otimes \Psi$, where $\Sigma(k \times k) > 0$ and $\Psi(n \times n) > 0$, if $vec(\mathbf{R}^T) \sim N_{kn}(vec(M^T), \Sigma \otimes \Psi)$.*

This definition tells us that if we create a single vector from matrix $\mathbf{R}$ by stacking the row vectors of $\mathbf{R}$ one after another, and if this vector follows a multivariate Gaussian distribution, this random matrix $\mathbf{R}$ has a matrix variate Gaussian distribution.

We shall use the notation $\mathbf{R} \sim N_{k,n}(M, \Sigma \otimes \Psi)$. The density of the random matrix $\mathbf{R}$ is given by the following theorem.

Theorem 5.3.6 *[108, pages 55] If $\mathbf{R} \sim N_{k,n}(M, \Sigma \otimes \Psi)$, then the probability density*

function of $\mathbf{R}$ *is given by*

$$(2\pi)^{-\frac{1}{2}kn}\det(\Sigma)^{-\frac{1}{2}n}\det(\Psi)^{-\frac{1}{2}k}etr\{-\frac{1}{2}\Sigma^{-1}(R-M)\Psi^{-1}(R-M)^T\}, \tag{5.9}$$

where $R \in \mathbb{R}^{k\times n}$, $M \in \mathbb{R}^{k\times n}$, *and* etr *is the exponential trace function* $etr\{.\} = exp(trace(.))$.

Corollary 5.3.7 *Let* $\mathbf{R}$ *be a* $k \times n$ *random matrix with each entry independent and identically distributed (i.i.d.) according to* $N(0,1)$. $\mathbf{R}$ *has a matrix variate Gaussian distribution with mean matrix* $M = 0$ *and covariance matrix* $I_k \otimes I_n$, *denoted by* $\mathbf{R} \sim N_{k,n}(0, I_k \otimes I_n)$.

Proof: Because each entry of $\mathbf{R}$ is i.i.d. and follows a $N(0,1)$ distribution, $vec(\mathbf{R}^T)$ has a multivariate Gaussian distribution with mean 0 and covariance $I_{kn} = I_k \otimes I_n$. By definition, $\mathbf{R}$ has a matrix variate Gaussian distribution. □

Theorem 5.3.8 *Let* $\mathbf{R}$ *be a* $k \times n$ *random matrix with each entry independent and identically distributed (i.i.d.) according to* $N(0,1)$. *Let* $X(n \times m)$ *be a constant matrix. Further assume* $rank(X) = m$. *Let* $\mathbf{Y} = \frac{1}{\sqrt{k}}\mathbf{R}X$. $\mathbf{Y}$ *has a matrix variate Gaussian distribution with mean matrix* 0 *and covariance* $I_k \otimes \frac{1}{k}X^TX$.

Proof: According to [90, Theorem 5.16], each row vector of $\mathbf{Y}$ has a multivariate Gaussian distribution with mean vector 0 and covariance $\frac{1}{k}X^TX$. Because each entry of $\mathbf{R}$ is statistically independent, any pairs of row vectors of $\mathbf{Y}$ are statistically independent too. We can create a single vector by stacking row vectors of $\mathbf{Y}$ one after another. According to [90, Problem 5.7.8], the new vector follows a multivariate Gaussian distribution with mean 0 and a block diagonal covariance matrix $\frac{1}{k}\begin{pmatrix} X^TX & 0 & \cdots & 0 \\ 0 & X^TX & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & X^TX \end{pmatrix}$. Therefore, by definition, we have $\mathbf{Y} \sim N_{k,m}(0, I_k \otimes \frac{1}{k}X^TX)$. Hence, the probability density function

of $\mathbf{Y}$ conditioned on $\mathbf{X}$ is given by

$$f_{\mathbf{Y}|\mathbf{X}}(Y|X) = (2\pi)^{-\frac{1}{2}km} \det(\frac{1}{k}X^TX)^{-\frac{1}{2}k} etr\{-\frac{1}{2}Y(\frac{1}{k}X^TX)^{-1}Y^T\},$$
where X has full column rank. (5.10)

□

Armed with basic matrix algebra definitions and theories, we can now compute the maximum a posteriori probability (MAP) estimate of $\mathbf{X}$ given the observation Y in the matrix form.

$$\begin{aligned}\hat{X}_{MAP}(Y,\theta) &= \arg\max_X f_{\mathbf{X}|\mathbf{Y},\boldsymbol{\theta}}(\mathbf{X}=X|\mathbf{Y}=Y,\boldsymbol{\theta}=\theta) \\ &= \arg\max_X f_{\mathbf{Y},\boldsymbol{\theta}|\mathbf{X}}(\mathbf{Y}=Y,\boldsymbol{\theta}=\theta|\mathbf{X}=X) f_{\mathbf{X}}(\mathbf{X}=X)\end{aligned}$$

To solve this maximization problem, we make the following assumptions:

Assumption 5.3.9 (The Attacker's Prior Belief about X) *The attacker assumes that $f_{\mathbf{X}}(\mathbf{X})$ is uniform within some range.*

Assumption 5.3.10 (The Attacker's Additional Background Knowledge) *The attacker has no other background knowledge about the private data, that is, $\boldsymbol{\theta} = \emptyset$.*

Assumption 5.3.11 (Independent Records) *We assume both X and Y have full column rank.* [3].

The first two assumptions are the same as the assumptions made for multivariate distributions. The third assumption allows the attacker to consider only the linearly independent records because linearly dependent records can be derived from the independent records.

[3] *Note that Y having full column rank implicitly implies that $k \geq m$.*

Under these assumptions, the MAP estimate becomes

$$\hat{X}_{MAP}(Y) = \arg\max_{X}(2\pi)^{-\frac{1}{2}km}\det(\frac{1}{k}X^TX)^{-\frac{1}{2}k}etr\{-\frac{1}{2}Y(\frac{1}{k}X^TX)^{-1}Y^T\},$$

where X and Y have full column rank. (5.11)

The following theorem gives the solution to this maximization problem.

Theorem 5.3.12 *Any X that satisfies condition $X^TX = Y^TY$ can be the optimal solution to the problem defined in Eq. 5.11.*

Proof: Please see Appendix 5.6.2 for the proof. □

Note that this result echoes the results we have for the multivariate case. If we consider $\mathbf{Y} = \mathbf{R}X$ as a random matrix, we know $E[\mathbf{Y}^T\mathbf{Y}] = X^TX$. So the optimal solution we have does not provide the attacker with more information about the private data X than what has been implied by the properties of random projection itself.

In the following part of this chapter, we will revisit some attack techniques designed in Chapter 4. We will see whether the random projection-based perturbation is vulnerable to these attacks.

5.4 Attack Techniques

In Chapter 4, we addressed the security issues of distance preserving perturbation by assuming the role of an attacker armed with three types prior information regarding the original data. We designed three different attack techniques and examined how well the attacker can recover the original data from the perturbed data and prior information. In this section, we study the privacy preserving properties of random projection along the same line. In particular, we consider the following prior knowledge the attacker could have.

5.4.1 Prior Knowledge

Known input-output The attacker knows some collection of linearly independent private data records. In other words, the attacker has a set of linearly independent input-output pairs.

Known sample The attacker knows that the original dataset arose as independent samples of some n-dimensional random vector V with unknown p.d.f. Also the attacker has another collection of independent samples from V.

Independent signals Each data attribute can be thought of as a time-varying signal. All the signals, at any given time, are statistically independent; and all the signals are non-Gaussian with the exception of one.

Random matrix is disclosed The specific realization of the random matrix is disclosed.

Next, we analyze the security of random projection-based perturbation for each of the scenarios listed above.

5.4.2 Known Input-Output Attack

Consider the perturbation model

$$\begin{aligned} Y &= \frac{1}{\sqrt{k}} RX \Leftrightarrow \\ \left(\begin{array}{cc} Y_p & Y_{m-p} \end{array} \right) &= \frac{1}{\sqrt{k}} R \left(\begin{array}{cc} X_p & X_{m-p} \end{array} \right). \end{aligned}$$

Let X_p denote the first p columns of X and X_{m-p} the remainder (likewise for Y). We assume that columns of X_p are all linearly independent and X_p is known to the attacker (Y is, of course, also known). The attacker will produce $\hat{x}$ and $1 \leq \hat{i} \leq m - p$ such that $\hat{x}$ is a good estimate of $x^{(\hat{i})}$, the $\hat{i}^{th}$ column in X_{m-p} (the $(p + \hat{i})^{th}$ column in X). Here, we also

assume $x^{(\hat{i})}$ is linearly independent of X_p because otherwise its value can be derived from a linear combination of X_p.

If $p = n$, then the attacker can recover the random matrix exactly because $R = \sqrt{k}YX_p^{-1}$. Note that even in this case, the attacker may not be able to get the exact value of the original private data. This is different from the distance preserving perturbation. We will discuss this case in Section 5.4.5 in detail. Throughout this section, we assume $p < n$. Next, we use the MAP estimate technique discussed in Sections 5.3.1 and 5.3.4 to recover the private data given the known inputs and outputs.

The MAP estimate of a data record x given its perturbed version y and known input-output pairs $RX_p = Y_p$ is

$$\begin{aligned}
\hat{x}_{MAP}(y,\theta) &= \arg\max_x f_{\mathbf{x}|\mathbf{y},\boldsymbol{\theta}}(\mathbf{x}=x|\frac{1}{\sqrt{k}}\mathbf{R}\mathbf{x}=y,\frac{1}{\sqrt{k}}\mathbf{R}X_p=Y_p)\\
&= \arg\max_x f_{\mathbf{y},\boldsymbol{\theta}|\mathbf{x}}(\frac{1}{\sqrt{k}}\mathbf{R}\mathbf{x}=y,\frac{1}{\sqrt{k}}\mathbf{R}X_p=Y_p|\mathbf{x}=x)f_{\mathbf{x}}(\mathbf{x}=x)\\
&= \arg\max_x f_{\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\mathbf{x}=x,\frac{1}{\sqrt{k}}\mathbf{R}\mathbf{x}=y,\frac{1}{\sqrt{k}}\mathbf{R}X_p=Y_p)\\
&= \arg\max_x f_{\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\frac{1}{\sqrt{k}}\mathbf{R}x=y,\frac{1}{\sqrt{k}}\mathbf{R}X_p=Y_p)\\
&= \arg\max_x f_{\mathbf{x},\mathbf{y},\boldsymbol{\theta}}(\frac{1}{\sqrt{k}}\mathbf{R}\bar{X}=\bar{Y}),
\end{aligned}$$

where $\bar{X} = [xX_p]$ and $\bar{Y} = [yY_p]$.

The above equation can be written as

$$\hat{x}_{MAP}(y,\theta) = \arg\max_x f_{\frac{1}{\sqrt{k}}\mathbf{R}\mathbf{Z}|\mathbf{Z}}(\frac{1}{\sqrt{k}}\mathbf{R}\mathbf{Z}=\bar{Y}|\mathbf{Z}=\bar{X})f_{\mathbf{Z}}(\mathbf{Z}=\bar{X}).$$

Assuming that $f_{\mathbf{Z}}$ is uniform over some interval, we get

$$\hat{x}_{MAP}(y,\theta) = \arg\max_x f_{\frac{1}{\sqrt{k}}\mathbf{R}\mathbf{Z}|\mathbf{Z}}(\frac{1}{\sqrt{k}}\mathbf{R}\mathbf{Z}=\bar{Y}|\mathbf{Z}=\bar{X}).$$

According to Theorem 5.3.8, $f_{\frac{1}{\sqrt{k}}\mathbf{RZ}|\mathbf{Z}}(\frac{1}{\sqrt{k}}\mathbf{RZ} = \bar{Y}|\mathbf{Z} = \bar{X})$ has the following form:

$$(2\pi)^{-\frac{1}{2}k(p+1)} \det(\frac{1}{k}\bar{X}^T\bar{X})^{-\frac{1}{2}k} etr\{-\frac{1}{2}\bar{Y}(\frac{1}{k}\bar{X}^T\bar{X})^{-1}\bar{Y}^T\},$$

where $\bar{X}$ has full column rank.

Theorem 5.3.12 tells us that if we knew nothing about $\bar{X}$, we could solve the MAP problem analytically. However, in the known input-output scenario, we know all the columns of $\bar{X}$ except for only one column. It is very difficult, if it is not impossible, to find an analytic solution in that case. Instead, we turn to numerical approaches to solve the maximization problem. In our experiments, we used the Matlab implementation [4] of the Nelder-Mead simplex algorithm [109] to find the optimal solution. This is a direct search method that attempts to optimize a scalar-valued nonlinear function of n real variables using only function values, without any numerical or analytic gradients. Since its publication in 1965, the Nelder-Mead simplex algorithm has become one of the most widely used methods for nonlinear unconstrained optimization. The book [110], which contains a bibliography with thousands of references, is devoted entirely to this algorithm and variations. Each iteration of this algorithm begins with a simplex. Here, a simplex in n-dimensional space is characterized by the $n + 1$ distinct vectors that are its vertices. In 2D space, a simplex is a triangle; in 3D space, it is a pyramid. At each step of the search, a new point in or near the current simplex is generated. The function value at the new point is compared with the function's values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, giving a new simplex. This step is repeated until the diameter of the simplex is less than the specified tolerance.

To demonstrate the performance of the MAP estimate-based known input-output attack, we conducted experiments on the same Letter Recognition data used in Section 4.5.3.

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

This data has $20,000$ records and 16 numeric features. We chose the first 6 features (excluding the class label) for the experiments. The setup of the experiments is illustrated in Algorithm 5.4.2.1.

Algorithm 5.4.2.1 MAP Estimate-based Known Input-Output Attack

Inputs: Let X denote the Letter Recognition Data with 6 attributes and 20,000 records. Let $Y = RX$. Let k denote the number of rows of R. Let p denote the number of known columns of the private data.

for $k = 6$ to 3 **do**
  for $p = k - 1$ to 1 **do**
    for $i = 1$ to 100 **do**
      Randomly choose $(p+1)$ independent columns from the original data X. Label the first column to be *unknown*, and all the other columns *known*.
      Choose the corresponding $(p + 1)$ columns from the perturbed data Y.
      for $j = 1$ to 100 **do**
        Call the Nelder-Mead simplex algorithm to solve the maximization problem. The starting values for the unknown is the median of the known + a random number in $(-2, 2)$
      end for
      Choose the best estimation from the above 100 solutions.
      Compute and record the relative error.
    end for
  end for
end for

Therefore, for each fixed k and p, we have logged 100 relative errors. We report the mean, median, max, min, variance of the relative errors. We also report the probability of ϵ-privacy breach. Note that in the experiments' setup, we choose $p + 1 \leq k$ to make Y full column rank. Otherwise, the function may not have an optimal. The experimental results are shown in Tables 5.5, 5.6, 5.7, and 5.8. It can be seen that as the number of known input-output pairs decreases, the relative error increases; as the dimension of the perturbed data decreases, the relative error increases.

	median	mean	variance	min	max	$\rho(x, 0.20)$	$\rho(x, 0.30)$
p=5	0.0616	0.0833	0.0055	0.0002	0.3429	0.91	0.99
p=4	0.1459	0.1954	0.0242	0.0194	0.7520	0.65	0.80
p=3	0.2459	0.2715	0.0283	0.0289	0.8564	0.38	0.61
p=2	0.3234	0.3668	0.0496	0.0673	1.2326	0.21	0.49
p=1	0.4230	0.4905	0.0814	0.0704	1.3733	0.15	0.30

Table 5.5. Relative errors of the MAP estimate-based known input-output attack. $k = 6$

	median	mean	variance	min	max	$\rho(x, 0.20)$	$\rho(x, 0.30)$
p=4	0.1742	0.2982	0.1351	0.0149	2.3440	0.56	0.71
p=3	0.2468	0.3026	0.0552	0.0263	1.2620	0.37	0.65
p=2	0.2844	0.3588	0.0629	0.0612	1.2668	0.29	0.52
p=1	0.4144	0.4847	0.0883	0.0964	1.4718	0.13	0.29

Table 5.6. Relative errors of the MAP estimate-based known input-output attack. $k = 5$

5.4.3 Known Sample Attack

In this scenario, we assume that each data record arose as an independent sample from a random vector V with unknown p.d.f. Furthermore, we assume that the attacker has a collection of p samples that arose independently from V.

In Section 4.5 of Chapter 4, we designed a Principal Component Analysis (PCA)-based attack technique. The basic idea is that the covariance matrix of the perturbed data $\Sigma_{M_T V}$ is related to the covariance of the original data Σ_V such that $\Sigma_{M_T V} = M_T \Sigma_V M_T^T$, where M_T is the orthogonal perturbation matrix (see Theorem 4.5.1). However, it can be shown that in the random projection scenario, the randomness introduced by R kills the covariance in the perturbed data used by the PCA-based attack. Specifically, given the random vector V, it can be shown that Σ_{RV} equals $I_n \gamma$ for some constant γ. Any vector in $\mathbb{R}^k$ is an eigenvector of Σ_{RV}; therefore, the PCA-based attack will not work. The following theorem depicts this property.

Theorem 5.4.1 *Let V be a random vector in $\mathbb{R}^{n \times 1}$. Let R be a random matrix in $\mathbb{R}^{k \times n}$,*

	median	mean	variance	min	max	$\rho(x, 0.20)$	$\rho(x, 0.30)$
p=3	0.2702	0.3532	0.0901	0.0501	2.0506	0.32	0.57
p=2	0.2804	0.3270	0.0413	0.0203	0.9647	0.30	0.57
p=1	0.4376	0.4828	0.0673	0.0896	1.7386	0.05	0.23

Table 5.7. Relative errors of the MAP estimate-based known input-output attack. $k = 4$

	median	mean	variance	min	max	$\rho(x, 0.20)$	$\rho(x, 0.30)$
p=2	0.3061	0.3526	0.0463	0.0439	1.0456	0.24	0.49
p=1	0.4503	0.4747	0.0724	0.0896	1.2360	0.14	0.30

Table 5.8. Relative errors of the MAP estimate-based known input-output attack. $k = 3$

each entry of R being i.i.d. with mean 0 and variance σ_r^2. Let $Y = RV$. Let Σ_V denote the population covariance matrix of V. Let Σ_{RV} be the population covariance matrix of RV. We have $\Sigma_{RV} = I_n\gamma$, where $\gamma = \sigma_r^2 E[\sum_t v_t^2]$.

Proof:

$$\begin{aligned}\Sigma_{RV} &= E[(RV - E[RV])(RV - E[RV])^T] \\ &= E[(RV - E[R]E[V])(RV - E[R]E[V])^T] \\ &= E[RVV^TR^T].\end{aligned}$$

Here, matrix RVV^TR^T can be expressed as

$$RVV^TR^T = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{k,1} & r_{k,2} & \cdots & r_{k,n} \end{pmatrix} \begin{pmatrix} v_1^2 & v_1v_2 & \cdots & v_1v_n \\ v_2v_1 & v_2^2 & \cdots & v_2v_n \\ \vdots & \vdots & \ddots & \vdots \\ v_nv_1 & v_nv_2 & \cdots & v_n^2 \end{pmatrix} \cdot \begin{pmatrix} r_{1,1} & r_{2,1} & \cdots & r_{k,1} \\ r_{1,2} & r_{2,2} & \cdots & r_{k,2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,n} & r_{2,n} & \cdots & r_{k,n} \end{pmatrix}.$$

It can be shown that the (i, j)-th entry ($i = 1, \ldots, k$, $j = 1, \ldots, k$) of RVV^TR^T is

$$\begin{cases} \sum_{q=1}^n \sum_{p=1, p\neq q}^n r_{i,p}r_{j,q}v_pv_q + \sum_{t=1}^n r_{i,t}^2 v_t^2, & \text{if } i = j; \\ \sum_{q=1}^n \sum_{p=1}^n r_{i,p}r_{j,q}v_pv_q, & \text{if } i \neq j. \end{cases}$$

Therefore, the (i, j)-th entry of $E[RVV^T R^T]$ is

$$\begin{cases} E[\sum_{q=1}^{n} \sum_{p=1, p\neq q}^{n} r_{i,p} r_{j,q} v_p v_q + \sum_{t=1}^{n} r_{i,t}^2 v_t^2] = \sigma_r^2 E[\sum_t v_t^2], & \text{if } i = j; \\ E[\sum_{q=1}^{n} \sum_{p=1}^{n} r_{i,p} r_{j,q} v_p v_q] = 0, & \text{if } i \neq j. \end{cases}$$

This completes the proof. □

5.4.4 Independent Signals Attack

In Section 4.6 of Chapter 4, we introduced Independent Component Analysis (ICA) as a possible tool for breaching privacy of distance preserving perturbation. In this section, we revisit ICA and show how to make random projection-based perturbation invulnerable to this kind of attack.

Decomposability of ICA: Recall that the basic ICA model can be defined as follows:

$$y(t) = Ax(t), \tag{5.12}$$

where $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$ denotes an n-dimensional vector collecting the n independent source signals $x_i(t), i = 1, 2, \ldots, n$. Here t indicates the time dependence. Each signal $x_i(t)$ can be viewed as an outcome of a continuous-value random process. A is a $k \times n$ unknown mixing matrix, which can be viewed as a mixing system with k receivers. The observed mixture is $y(t) = (y_1(t), y_2(t), \ldots, y_k(t))^T$. The aim of ICA is to design a filter that can recover the original signals from only the observed mixture. Because $y(t) = Ax(t) = (A\Lambda P)(P^{-1}\Lambda^{-1}x(t))$ for any diagonal matrix Λ and permutation matrix P, the recovered signals $x(t)$ can never have completely unique representation. So, the uniqueness of the recovered signals found by ICA can only be guaranteed up to permutation and scaling ambiguities.

In practice, a linear filter is designed to get the recovered signals $\hat{x}(t) = (\hat{x}_1(t), \hat{x}_2(t),$

$\ldots, \hat{x}_l(t))^T$ from a k-dimensional input $y(t) = (y_1(t), y_2(t), \ldots, y_k(t))^T$. In other words, we need to find an $l \times k$ matrix B such that

$$\hat{x}(t) = By(t). \tag{5.13}$$

Here, B is called the separating matrix. Combining Eq. 5.12 and Eq. 5.13, we get

$$\hat{x}(t) = BAx(t) = Zx(t), \tag{5.14}$$

where $Z = BA$ is an $l \times n$ matrix. Each element of $\hat{x}(t)$ is thus a linear combination of $x_i(t)$ with weights given by $z_{i,j}$, where $z_{i,j}$ denotes the (i, j)-th entry of Z.

Ideally, when $k \geq n$ (*i.e.*, the number of receivers is greater than or equal to the number of source signals), if the mixing matrix A has full column rank, there always exists an $l \times k$ separating matrix B such that $Z = BA = I$, where I is an identity matrix. If this is the case, we can recover all the signals simultaneously up to scaling and permutation ambiguities.

When $l \leq k < n$ (*i.e.*, the number of sources is greater than the number of receivers), [5] it is generally not possible to design linear filters to simultaneously recover all these signals. This kind of separation problem is termed as *overcomplete ICA* or *under-determined source separation*. Cao and Liu [93] analyzed the conditions for the existence of the separating matrix B. Next, we first introduce two definitions (Definition 5.4.1 and 5.4.2) and one theorem (Theorem 5.4.2) from their work, which serve as important building blocks in our solutions.

Definition 5.4.1 (Partition Matrix) *[93] A set of n integers $S = \{1, 2, \ldots, n\}$ can be*

[5] This implies that the number of recovered signals will be less than or equal to the number of the original signals. This is reasonable because we cannot get more signals than the original ones.

partitioned into l ($l \leq n$) disjoint subsets S_i, $i = 1, 2, \ldots, l$. An $l \times n$ matrix Z is called a partition matrix if its i, j-th entry $z_{i,j} = 1$ when $j \in S_i$, and $z_{i,j} = 0$ otherwise. Z is called a generalized partition matrix if it is a product of an $l \times n$ partition matrix and an $n \times n$ nonsingular diagonal matrix.

When none of the subset S_i is empty, Z is simply a matrix in which each column has only one nonzero entry and each row has at least one nonzero entry.

Definition 5.4.2 (l-row Decomposable) *[93] A $k \times n$ matrix A is called l-row decomposable if there exists an $l \times k$ matrix B such that $Z = B \times A$ is an $l \times n$ generalized partition matrix.*

Therefore, if A is l-row decomposable, there exists a matrix B that enables Z to separate the source signals into l disjoint subgroups; each output $\hat{x}_i(t), i = 1, 2, \ldots, l$ is a linear combination of the source signals in one subgroup, *i.e.*,

$$\hat{x}_i \quad = \quad \sum_{j \in S_i} z_{i,j} x_j, \text{ i = 1, 2, . . . , l}$$

If for some i, $S_i = \{p\}$, then $\hat{x}_i = z_{i,p} x_p$, that is, by using Z, we can separate out one signal x_p up to scaling ambiguities. If the number of the disjoint subgroups is n (*i.e.*, $l = n$), every subset S_i ($i = 1, \ldots, l$) contains only one element, and there will be a complete separation.

Theorem 5.4.2 *[93] Matrix A is l-row decomposable if and only if its columns can be grouped into l disjoint groups such that the column vectors in each group are linearly independent of the vectors in all the other groups.*

Proof: Please see the proof of Theorem 1 in [93]. □

Cao *et al.* proved that when $k < n$, the source signals can at most be separated into k disjoint groups from the observed mixture and at most $k - 1$ signals (independent components) can be separated out.

Our claim is that if we can control the structure of the mixing matrix A such that A is not two-row decomposable, then there is no linear method that can find a matrix B for separating the source signals into two or more disjoint groups. In that cases, it is not possible to separate out any of the source signals. The following theorem characterizes this property.

Theorem 5.4.3 *Any $k \times n$ ($n \geq 2k, n \geq 2$) random matrix with entries independent and identically chosen from a continuous distribution in the real domain is not two-row decomposable with probability 1.*

Proof: For a $k \times n$ random matrix with $n \geq 2k$ and any partition of its columns into two non-empty sets, at least one set will have at least k members. Thus, this set of columns contains a $k \times k$ sub-matrix, denoted as M. If M is nonsingular, its k column vectors will span $\mathbb{R}^k$ Euclidean space. In this case, there is always at least one vector in one group belonging to the space spanned by the other group, which does not satisfy the requirements in Theorem 5.4.2.

Now let us show that M is indeed nonsingular with probability 1. It has been proved in [111, Theorem 3.3] that the probability that MM^T is positive definite is 1. [6] Because 1) a matrix is positive definite if and only if all the eigenvalues of this matrix are positive and 2) a matrix is nonsingular if and only if all its eigenvalues are nonzero [107, Theorem 1.2.2], MM^T is nonsingular with probability 1. Further note that $rank(M) = rank(MM^T) = rank(M^TM)$ [112], therefore M is nonsingular with probability 1. This completes the proof. □

The above non-singularity property of a random matrix has also been proved in [107, Theorem 3.2.1] when the random matrix is Gaussian. Thus, by letting $n \geq 2k$, there is no linear filter that can separate the observed mixtures into two or more disjoint groups;

[6] We get this result by replacing the matrix A in [111, Theorem 3.3] with an identity matrix.

therefore, it is not possible to recover any of the source signals. In Section **??** we will demonstrate this property with experiments.

The discussion in this section summarizes as:

- When $k \geq n$ (*i.e.*, the number of receiver is greater than or equal to the number of source signals), all the source signals can be separated out from their mixture up to scaling and permutation ambiguities if and only if 1) the signals are statistically independent; 2) the mixing matrix A has full column rank; and 3) at most one source signal is Gaussian.

- When $l \leq k < n$ (*i.e.*, the number of receivers is less than the number of sources), the source signals can at most be separated into k disjoint groups from the mixtures and at most $k-1$ signals can be separated out. In particular, when the mixing matrix R is not two-row decomposable ($n \geq 2k, n \geq 2$, and with i.i.d. entries chosen from a continuous distribution), there is no linear method that can find a matrix B to separate out any of the source signals.

Recent Work on Overcomplete ICA: Recently, overcomplete ICA ($k < n$) has drawn much attention. It has been found that even when $k < n$, if all the sources are non-Gaussian and statistically independent, it is still possible to identify the mixing matrix such that the it is unique up to a right multiplication by a diagonal and a permutation matrix [113, Theorem 3.1]. If it is also possible to determine the distribution of $x(t)$, we could reconstruct the source signals in a probabilistic sense. However, despite its high interest, the overcomplete ICA problem has only been treated in particular cases. Lewicki *et al.* [114] proposed a generalized method for learning overcomplete ICA in which the source signals were assumed to have a sparse distribution, *e.g.*, Laplacian distribution. Several other similar solutions to the separation of independent components from their overcomplete mixtures have been proposed [115–117]. However, if any Gaussian signals were allowed, the mixing

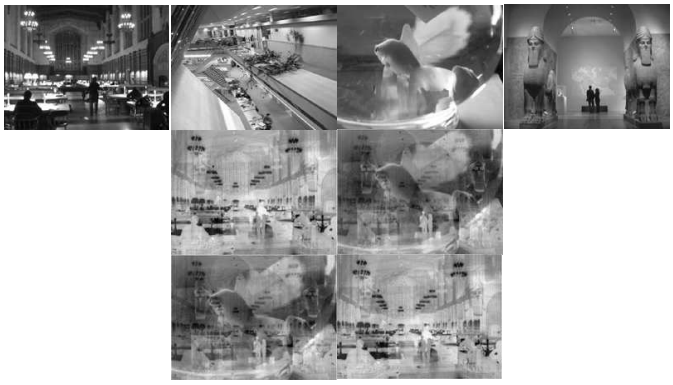


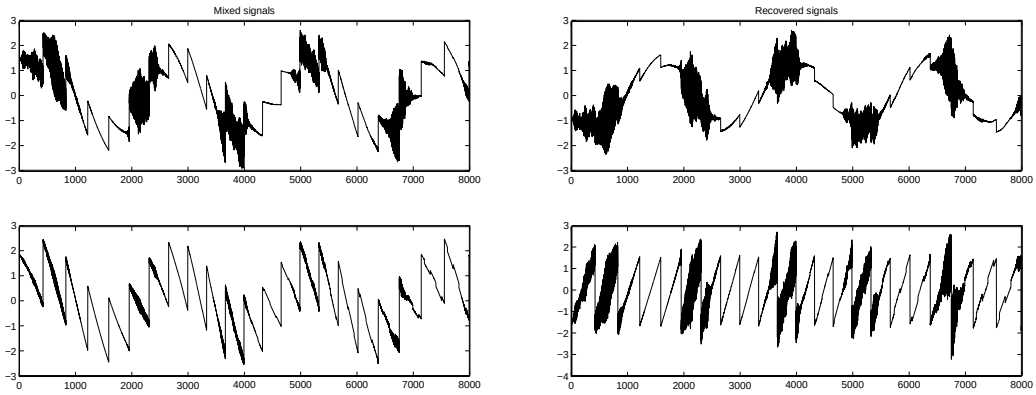
FIG. 5.11. Performance of ICA attack on random projection perturbed image data. The first row – original images; the second row – perturbed images; and the third row – recovered images.

matrix would not be identifiable [118] and the distribution of the source signals would not be unique [113, Example 2 and 4]. Again, if the sources were correlated, they would cluster in the same group and only the real independent components hidden behind them could possibly be found.

Experiments: To demonstrate that ICA attack cannot effectively breach the privacy of random projection-based perturbation, we chose both image and audio data for the experiments.

First, we considered the same image dataset used in Section 4.6.3 of Chapter 4. The dataset consists of four natural scene pictures represented by a 450×338 pixel grid – the top row of Figure 5.11. Each grid is stretched out into a length $152,100$ row vector. The perturbed versions, rows of $Y = \frac{1}{\sqrt{k}\sigma_r} RX$, can be seen in the middle row of Figure 5.11. Here, the random projection compressed four pictures into only two. After applying ICA, the attacker produced estimates as seen in the bottom row of Figure 5.11. It can be seen that ICA can only produce two pictures and each of them is still a mixture of the original four pictures. Thus, no pictures can be separated out in this scenario.

Second, we considered the same four audio signals used in Section 4.6.3 of Chapter



(a) (b)

FIG. 5.12. (a) Linear mixture of the original four source signals (as shown in Figure 4.18) with a 50% random projection rate. ($n = 4, k = 2$). (b) The recovered signals. It can be observed that none of the original signals can be reconstructed and at most $k = 2$ independent components can be found by ICA.

4 (shown in Figure 4.18). A perturbation of these signals is generated by pre-multiplying a 2×4 random matrix to them (shown in Figure 5.12(a)). The recovered signals after applying ICA is shown in Figure 5.12(b). It can be seen that after a 50% random projection, the original four signals are compressed into two and ICA cannot recover any of them.

5.4.5 Random Matrix is Disclosed

In this scenario, we assume that the random matrix itself is disclosed. This can be viewed as the worst case. Recall that for the distance preserving perturbation, if the orthogonal matrix is known, the attacker can recover the original data exactly. In this section, we analyze whether this perfect recovery also happens in random projection-based perturbation.

Consider the model $Y = RX$, where $R \in \mathbb{R}^{k \times n}$ with $k < n$, and $X \in \mathbb{R}^{n \times m}$. This model can be viewed as a set of underdetermined systems of linear equations (more unknowns than equations), each with the form $y = Rx$, where x is an $n \times 1$ column vector

from X and y is the corresponding column vector from Y. For each such linear system, assuming both R and y are known, we can prove that the solution is never unique.

In practice, the underdetermined system can be analyzed using the QR factorization [119, 120]:

$$R^T = \mathcal{Q} \begin{pmatrix} \mathcal{R} \\ 0 \end{pmatrix},$$

where $\mathcal{Q} \in \mathbb{R}^{n \times n}$ is orthogonal and $\mathcal{R} \in \mathbb{R}^{k \times k}$ is upper triangular. If R has full rank, *i.e.*, $rank(R) = k$, there is a unique solution x_{min_norm} that minimizes $||x||_2$ [7]:

$$\begin{aligned} x_{min\_norm} &= \mathcal{Q} \begin{pmatrix} \mathcal{R}^{-T} y \\ 0 \end{pmatrix} \\ &= \mathcal{Q} \begin{pmatrix} \mathcal{R} \\ 0 \end{pmatrix} (\mathcal{R}^T \mathcal{R})^{-1} y \\ &= R^T (R R^T)^{-1} y \\ &= R^\dagger y, \end{aligned}$$

where $R^\dagger = R^T (R R^T)^{-1}$ is the pseudo-inverse of R. The complete solution set to the underdetermined system $y = Rx$ can be composed by adding an arbitrary vector from the null space of R to x_{min_norm}. In other words, any $\hat{x}$ satisfying the following condition can be the solution.

$$\hat{x} = x_{min\_norm} + Ab,$$

[7] This problem is referred to as finding a minimum norm solution to an underdetermined system of linear equations.

where A is the orthonormal basis for the null space of R and b is an arbitrary vector.

Remark: The above result shows that even if the random matrix R is known to the attacker, it is still impossible to find the exact values of *all the elements* of vector x. The best we can do is to find the minimum norm solution. However, one may ask whether it is possible to completely identify *some* elements in the vector x. Obviously, if we can find as many linearly independent equations as the unknown elements, we can partially solve the system. In the following, we will discuss this possibility by using the "l-secure" definition introduced in [51, Definition 4.1].

Definition 5.4.3 (l-secure) *A matrix R is said to be l-secure if by removing any l columns from R, the remaining sub-matrix still has a full row rank.*

This property guarantees that any non-zero linear combination of the row vectors of R contains at least $l + 1$ non-zero elements. To prove this, let us assume that some linear combination of the row vectors has at most l non-zero elements. If we remove these l corresponding columns from R, then apply the same linear combination on all the row vectors of the remaining sub-matrix, we will get a zero vector. This implies that the row vectors of this sub-matrix are linearly dependent and the rank of this sub-matrix is not of full row rank, which contradicts the l-secure definition.

If the coefficient matrix of a linear equations system is l-secure, each unknown variable in a linear equation is disguised by at least l other unknown variables no matter what kind of non-zero linear combination produces this equation. Now the question is whether we can find $l+1$ linearly independent equations that just involve these $l+1$ unknowns? The answer is *No*. The following theorem (which can be viewed as a generalization of [51, Theorem 4.3]) proves that any $l + 1$ non-zero linear combinations of the equations contains at least $2l + 1$ unknown variables if these $l + 1$ vectors are linearly independent.[8]

[8] If these $l + 1$ vectors are not linearly independent, the $l + 1$ equations contain $\Gamma + l$ unknown variables.

Theorem 5.4.4 *Let Υ be an $(l+1) \times n$ matrix, where each row of Υ is a nonzero linear combination of row vectors in R. If R is l-secure, the linear equations system $y = \Upsilon x$ involves at least $2l+1$ unknown variables if these $l+1$ vectors are linearly independent.*

Proof: Since row vectors of Υ are all linearly independent, $y = \Upsilon x$ can be transformed into $y = (I : \tilde{\Upsilon})x$ through a proper Gaussian elimination. Here, I is the $(l+1) \times (l+1)$ identity matrix, $\tilde{\Upsilon}$ is a $(l+1) \times (m-(l+1))$ matrix, and $(I : \tilde{\Upsilon})$ is a vertical concatenation of I and $\tilde{\Upsilon}$. Because R is l-secure, each row of $(I : \tilde{\Upsilon})$ contains at least $l+1$ non-zero entries, which corresponds to $l+1$ unknowns. Because in each row of $(I : \tilde{\Upsilon})$, there is a single 1 from I, there are at least l non-zero entries in $\tilde{\Upsilon}$. Thus, the whole system contains at least $2l+1$ unknowns, with $l+1$ unknowns being contributed by I and at least l unknowns from $\tilde{\Upsilon}$. □

This theorem shows that if a coefficient matrix is l-secure, any linear combinations of the equations contains at least $l+1$ variables. Therefore,it is not possible to find $l+1$ linearly independent equations that just involve the same $l+1$ variables, and the solutions to any partial unknown variables are infinite.

Now let us consider the $k \times n$ random projection matrix and the restrictions of ICA we discussed in the previous sections. When $n = 2k$, after removing any k columns from mixing matrix R, according to the proof of Theorem 5.4.3, we can conclude that the remaining square matrix has a full row rank with probability 1. Therefore, the system is k-secure with probability 1. In other words, each unknown variable is disguised by at least k other variables, and we cannot find k linearly independent equations that just involve these variables, so, the solutions are infinite. When $n > 2k$, the security level is even higher because we can remove more columns while keeping the sub-matrix full row rank (however, the accuracy of the random projection will probably be compromised if k is too

Here Γ denotes the rank of the matrix formed by these $l+1$ vectors.

small).

Remark: The above result shows that even if the random matrix R is known to the attacker, if R is k-secure, each unknown variable is masked by at least k other unknown variables no matter how the equations are linear combined. So it is impossible to find the exact value of *any element* in the original data.

5.5 Summary

In this chapter, we studied a randomized multiplicative data perturbation technique for privacy preserving data mining. This technique projects the data onto a lower dimensional random space while maintaining its distance related statistics with a high probability. Theoretical and empirical results show that this technique offers higher privacy protection than the orthogonal transformation-based distance preserving perturbation, but with little loss in accuracy.

In summary, the random projection-based data perturbation has the following characteristics:

- Random projection maps the original data to a lower dimensional subspace while maintaining much of its distance-related statistics. The error of the inner product produced by random projection is zero on average, and the variance is inversely proportional to the dimensionality of the reduced space. A closed-form expression of the accuracy for estimating the Euclidean distance can be derived when the random matrix has a matrix variate Gaussian distribution.

- Under mild assumptions, any $\hat{x}$ that satisfies $\hat{x}^T\hat{x} = y^T y$ is the maximum a posteriori probability (MAP) estimate of the original data x given the perturbed data y. From this perspective, random projection does not offer the attacker more information about the private data than what has been implied by the properties of random

projection itself.

- The analytic upper bound of the probability of ϵ-privacy breach can be derived in the context of MAP estimate.
- The data owner could control the privacy and accuracy tradeoff and select an appropriate dimension for the reduced space.
- Random projection-based perturbation offers higher privacy protection than distance preserving perturbation, but with little loss in accuracy.

5.6 Appendix

5.6.1 Appendix I

Key Technical Results for the Proof of Lemma 5.1.4: Let $r_{i,j}$ and $\epsilon_{i,j}$ be the i,j-th entry of matrix $R_{k\times n}$ and $R^T R$, respectively. Each $r_{i,j}$ is independent and identically distributed (i.i.d.) according to $N(0, \sigma_r)$. Now let us prove $E[\epsilon_{i,i}] = k\sigma_r^2$, $Var[\epsilon_{i,i}] = 2k\sigma_r^4$, $\forall i$; and $E[\epsilon_{i,j}] = 0$, $Var[\epsilon_{i,j}] = k\sigma_r^4$, $\forall i, j, i \neq j$.

Proof: Note that $\epsilon_{i,i} = \sum_{t=1}^{k} r_{t,i}^2$ and $\epsilon_{i,j} = \sum_{t=1}^{k} r_{t,i}r_{t,j}$, $i \neq j$, we have $E[\epsilon_{i,i}] = E[\sum_{t=1}^{k} r_{t,i}^2] = kE[r_{t,i}^2] = k\sigma_r^2$ and $E_{i\neq j}[\epsilon_{i,j}] = E[\sum_{t=1}^{k} r_{t,i}r_{t,j}] = \sum_{t=1}^{k} E[r_{t,i}r_{t,j}] = \sum_{t=1}^{k} E[r_{t,i}]E[r_{t,j}] = 0$.

To obtain the variance of $\epsilon_{i,i}$, we first compute $E[\epsilon_{i,i}^2] = E[\sum_{t=1}^{k} r_{t,i}^4 + \sum_{p\neq q, 1\leq p,q\leq k} r_{p,i}^2 r_{q,i}^2] = kE[r_{t,i}^4] + k(k-1)E[r_{p,i}^2]E[r_{q,i}^2] = 3k\sigma_r^4 + k(k-1)\sigma_r^4 = (2k + k^2)\sigma_r^4$. The second to the last equation in the above is based on the fact that $E[r_{t,j}^4] = 3\sigma_r^4$ for random variable $r_{t,j} \sim N(0, \sigma_r)$[9]. Therefore, $Var[\epsilon_{i,i}] = E[\epsilon_{i,i}^2] - (E[\epsilon_{i,i}])^2 = 2k\sigma_r^4$. Similarly, $E_{i\neq j}[\epsilon_{i,j}^2] = E_{i\neq j}[\sum_{t=1}^{k} r_{t,i}^2 r_{t,j}^2 + \sum_{p\neq q, 1\leq p,q\leq k} r_{p,i}r_{p,j}r_{q,i}r_{q,j}] = kE[\sum_{t=1}^{k} r_{t,i}^2 r_{t,j}^2] + 0 = k\sigma_r^4$, hence, $Var_{i\neq j}[\epsilon_{i,j}] = k\sigma_r^4$. □

[9] http://mathworld.wolfram.com/NormalDistribution.html

Lemma 5.1.4: Let x, y be two data vectors in $\mathbb{R}^n$. Let R be a $k \times n$ dimensional random matrix. Each entry of the random matrix is independent and identically distributed (i.i.d.) according to a Gaussian distribution with mean zero and variance σ_r^2. Further let

$$u = \frac{1}{\sqrt{k}\sigma_r} Rx, \qquad \text{and} \qquad v = \frac{1}{\sqrt{k}\sigma_r} Ry. \text{ Then}$$
$$E[u^T v - x^T y] = 0 \text{ and}$$
$$Var[u^T v - x^T y] = \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 + (\sum_i x_i y_i)^2).$$

In particular, if both x and y are normalized to unity, $\sum_i x_i^2 \sum_i y_i^2 = 1$ and $(\sum_i x_i y_i)^2 \leq 1$. We have the upper bound of the variance as follows:

$$Var[u^T v - x^T y] \leq \frac{2}{k}.$$

Proof: Using Lemma 5.1.2, the expectation of projection distortion is

$$\begin{aligned} E[u^T v - x^T y] &= E[\frac{1}{k\sigma_r^2} x^T R^T R y - x^T y] \\ &= \frac{1}{k\sigma_r^2} x^T E[R^T R] y - x^T y \\ &= \frac{1}{k\sigma_r^2} k\sigma_r^2 x^T y - x^T y \\ &= 0. \end{aligned}$$

To compute the variance of the distortion, let us first express the inner product between the

projected vectors as

$$\begin{aligned} u^T v &= \frac{1}{\sqrt{k}\sigma_r} x^T R^T \frac{1}{\sqrt{k}\sigma_r} Ry \\ &= \frac{1}{k\sigma_r^2} x^T R^T Ry \\ &= \frac{1}{k\sigma_r^2} (\sum_i x_i \epsilon_{i,i} y_i + \sum_{i \neq j} x_i \epsilon_{i,j} y_j) \\ &= \frac{1}{k\sigma_r^2} \sum_i x_i \epsilon_{i,i} y_i + \frac{1}{k\sigma_r^2} \sum_{i \neq j} x_i \epsilon_{i,j} y_j. \end{aligned}$$

Denote $\frac{1}{k\sigma_r^2} \sum_i x_i \epsilon_{i,i} y_i$ as Φ and $\frac{1}{k\sigma_r^2} \sum_{i \neq j} x_i \epsilon_{i,j} y_j$ as Ψ. Then $Var[u^T v] = Var[\Phi] + Var[\Psi] + 2Cov[\Phi, \Psi]$.

Now, let us compute $Cov[\Phi, \Psi]$:

$$Cov[\Phi, \Psi] = E[\Phi\Psi] - E[\Phi]E[\Psi].$$

Since $E[\epsilon_{i,j}] = 0 \; \forall i, j, i \neq j$, so $E[\Psi] = 0$. Hence,

$$\begin{aligned} Cov[\Phi, \Psi] &= E[\Phi\Psi] - 0 \\ &= \frac{1}{k^2\sigma_r^4} E[\sum_i x_i \epsilon_{i,i} y_i \times \sum_{p \neq q} x_p \epsilon_{p,q} y_q]. \end{aligned}$$

It is straightforward to verify that $E[\epsilon_{i,i} \epsilon_{p,q}] = 0$ when $p \neq q$. So $Cov[\Phi, \Psi] = 0$.

The variance of Φ is

$$\begin{aligned} Var[\Phi] &= Var[\frac{1}{k\sigma_r^2}\sum_i x_i\epsilon_{i,i}y_i] \\ &= \frac{1}{k^2\sigma_r^4}Var[\sum_i x_i\epsilon_{i,i}y_i] \\ &= \frac{1}{k^2\sigma_r^4}(E[(\sum_i x_i\epsilon_{i,i}y_i)^2] - (E[\sum_i x_i\epsilon_{i,i}y_i])^2) \\ &= \frac{1}{k^2\sigma_r^4}(E[\sum_i x_i^2\epsilon_{i,i}^2y_i^2 + \sum_{p\neq q} x_py_p\epsilon_{p,p}x_qy_q\epsilon_{q,q}] - (E[\sum_i x_i\epsilon_{i,i}y_i])^2). \end{aligned}$$

Since $E[\epsilon_{i,i}] = k\sigma_r^2$, $E[\epsilon_{i,i}^2] = (2k+k^2)\sigma_r^4$ and $E[\epsilon_{p,p}\epsilon_{q,q}] = k^2\sigma_r^4$, we have

$$\begin{aligned} Var[\Phi] &= \frac{1}{k^2\sigma_r^4}(2k+k^2)\sigma_r^4\sum_i x_i^2y_i^2 + \sum_{p\neq q} x_py_px_qy_q - (\sum_i x_iy_i)^2 \\ &= (\frac{2}{k}+1)\sum_i x_i^2y_i^2 + \sum_{p\neq q} x_py_px_qy_q - (\sum_i x_iy_i)^2. \end{aligned}$$

The variance of Ψ is

$$\begin{aligned} Var[\Psi] &= \frac{1}{k^2\sigma_r^4}Var[\sum_{i\neq j} x_i\epsilon_{i,j}y_j] \\ &= \frac{1}{k^2\sigma_r^4}(E[(\sum_{i\neq j} x_i\epsilon_{i,j}y_j)^2] - (E[\sum_{i\neq j} x_i\epsilon_{i,j}y_j])^2) \\ &= \frac{1}{k^2\sigma_r^4}(E[(\sum_{i\neq j} x_i\epsilon_{i,j}y_j)^2] - 0 \\ &= \frac{1}{k^2\sigma_r^4}\sum_{i\neq j}\sum_{p\neq q} x_iy_jx_py_qE[\epsilon_{i,j}\epsilon_{p,q}]. \end{aligned}$$

Since $E[\epsilon_{i,j}\epsilon_{p,q}] = 0$ unless $i = p$ and $j = q$, or $i = q$ and $j = p$, we have,

$$\begin{aligned}
Var[\Psi] &= \frac{1}{k^2\sigma_r^4}(\sum_{i\neq j} x_i^2 y_j^2 + \sum_{i\neq j} x_i y_j x_j y_i) E_{i\neq j}[\epsilon_{i,j}^2] \\
&= \frac{1}{k^2\sigma_r^4}(\sum_i x_i^2 \sum_{j\neq i} y_j^2 + \sum_i x_i y_i \sum_{j\neq i} x_j y_j) k\sigma_r^4 \\
&= \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 - \sum_i x_i^2 y_i^2 + (\sum_i x_i y_i)^2 - \sum_i x_i^2 y_i^2) \\
&= \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 + (\sum_i x_i y_i)^2 - 2\sum_i x_i^2 y_i^2).
\end{aligned}$$

Thus,

$$\begin{aligned}
Var[u^T v] &= Var[\Phi] + Var[\Psi] + 0 \\
&= (\frac{2}{k} + 1)\sum_i x_i^2 y_i^2 + \sum_{p\neq q} x_p y_p x_q y_q - (\sum_i x_i y_i)^2 \\
&\quad + \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 + (\sum_i x_i y_i)^2 - 2\sum_i x_i^2 y_i^2) \\
&= \frac{1}{k}(\sum_i x_i^2 y_i^2 + (\sum_i x_i y_i)^2) + (\sum_i x_i^2 y_i^2 \\
&\quad + \sum_{p\neq q} x_p y_p x_q y_q - (\sum_i x_i y_i)^2) \\
&= \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 + (\sum_i x_i y_i)^2).
\end{aligned}$$

This gives the final result $Var[u^T v - x^T y] = \frac{1}{k}(\sum_i x_i^2 \sum_i y_i^2 + (\sum_i x_i y_i)^2)$. □

5.6.2 Appendix II

Theorem 5.3.12: Any X that satisfies condition $X^TX = Y^TY$ can be the optimal solution to the problem defined as below (also in Eq. 5.11.)

$$\hat{X}_{MAP}(Y) = \arg\max_X (2\pi)^{-\frac{1}{2}km} \det(\frac{1}{k}X^TX)^{-\frac{1}{2}k} etr\{-\frac{1}{2}Y(\frac{1}{k}X^TX)^{-1}Y^T\},$$

where X and Y have full column rank.

Proof: Let $Z = (\frac{1}{k}X^TX)^{-1}$. The maximization problem can be written as

$$\hat{X}_{MAP}(Y) = (2\pi)^{-\frac{1}{2}km} \det(Z)^{\frac{k}{2}} etr\{-\frac{1}{2}YZY^T\}.$$

Further let $A = YZY^T$. Since Y has full column rank, without loss of generality, we can assume that Y is invertible, *i.e.*, $k = m$. Therefore, A is also invertible and $Z = Y^{-1}AY^{T-1}$. The maximization problem can be written as

$$\begin{aligned}\hat{X}_{MAP}(Y) &= (2\pi)^{-\frac{1}{2}km} \det(Y^{-1}AY^{T-1})^{\frac{k}{2}} etr\{-\frac{1}{2}A\} \\ &= (2\pi)^{-\frac{1}{2}km} \det(Y^{-1}Y^{T-1})^{\frac{k}{2}} \det(A)^{\frac{k}{2}} etr(-\frac{1}{2}A).\end{aligned}$$

Since $(2\pi)^{-\frac{1}{2}km}$ is constant and Y is also fixed, we only need to maximize

$$\det(A)^{\frac{k}{2}} etr(-\frac{1}{2}A),$$

s.t. R is positive definite.

Let λ_j be the eigenvalue of A, we have

$$\begin{aligned} \det(A)^{\frac{k}{2}} &= (\prod_j \lambda_j)^{\frac{k}{2}} = \prod_j \lambda_j^{\frac{k}{2}}; \\ etr(-\frac{1}{2}A) &= exp(trace(-\frac{1}{2}A)) = exp(-\frac{1}{2}\sum_j \lambda_j) = \prod_j exp(-\frac{1}{2}\lambda_j). \end{aligned}$$

Therefore

$$\begin{aligned} \det(A)^{\frac{k}{2}} etr(-\frac{1}{2}A) &= \prod_j \lambda_j^{\frac{k}{2}} \prod_j exp(-\frac{1}{2}\lambda_j) \\ &= \prod_j \lambda_j^{\frac{k}{2}} exp(-\frac{1}{2}\lambda_j). \end{aligned}$$

The function $g(w) = w^{\frac{k}{2}} exp(-\frac{1}{2}w)$ has its maximum for $w > 0$ at $w = k$. So the maximum of $\det(A)^{\frac{k}{2}} etr(-\frac{1}{2}A)$ is obtained when all $\lambda_j = k$.

Thus, we can take $A = kI$, where I is identity matrix, so

$$\begin{aligned} Z &= Y^{-1} A Y^{T-1} \\ &= Y^{-1} k I Y^{T-1} \\ &= k Y^{-1} Y^{T-1} \\ &= k(Y^T Y)^{-1}. \end{aligned}$$

Because $Z = (\frac{1}{k} X^T X)^{-1}$, we have

$$(\frac{1}{k} X^T X)^{-1} = k(Y^T Y)^{-1},$$

which implies that

$$X^T X = Y^T Y.$$

Therefore, any X that satisfies condition $X^T X = Y^T Y$ can the the optimal solution.

□

Chapter 6

CONCLUSIONS AND FUTURE WORK

Privacy is becoming an increasingly important issue in many data mining applications that deal with health care, security, finance, behavior and other types of sensitive data. It is particularly becoming important in counter-terrorism and homeland security-related applications. These applications may require creating profiles, constructing social network models, and detecting terrorists' communications. All of them involve the collection and analysis of private sensitive data. For example, mining health care data for the detection of bio-terrorism may require analyzing clinical records and pharmacy transactions data of certain off-the-shelf drugs. However, releasing and combining such diverse data sets belonging to different parties may violate privacy laws. Although health organizations are allowed to release the data as long as the identifiers (e.g., name, SSN, address, etc.,) are removed, it is not considered safe enough because re-identification attacks may be constructed for linking different public data sets to identify the original subjects [26]. This calls for well-designed techniques that pay careful attention to hiding privacy sensitive information while preserving the inherent patterns of the original data. Privacy preserving data mining (PPDM) strives to provide a solution to this problem. It aims to allow useful data patterns to be extracted without compromising privacy.

This dissertation specifically investigates the characteristics of different multiplica-

tive data perturbation techniques for PPDM. First, we have briefly reviewed two traditional multiplicative data perturbation techniques that have been well studied in the statistics community. We have shown the following.

- These perturbations are primarily used to mask the private data while allowing summary statistics (*e.g.*, sum, mean, variance, covariance) of the original data to be estimated. Because each data element is distorted independently, the Euclidean distances and inner products among the original data records are usually not preserved.

- These perturbation schemes are equivalent to additive perturbation after the logarithmic transformation. Due to the large volume of research in deriving private information from the additive noise perturbed data, the security of these perturbation schemes is questionable.

Next, we have examined the effectiveness of distance preserving perturbation. Theoretical and experimental results have shown the following.

- This type of perturbation is essentially a series of rotations and reflections of the data. It exactly preserves the Euclidean distances and inner products in the original data. Therefore, many interesting data mining algorithms can be applied directly to the perturbed data and produce an error-free result.

- However, this perturbation is vulnerable to many attacks such as known input-output attacks, known sample attacks and independent signals attacks.

Finally, we have explored a random projection-based perturbation. This technique projects the data onto a lower dimensional subspace while maintaining the pairwise distances of the original data records with high probabilities. We have shown that

- From the perspective of maximum a posteriori probability (MAP) estimate, random projection-based perturbation does not offer the attacker more information about the private data than what has been implied by the properties of random projection itself.
- The analytic bounds of the probability of ϵ-privacy breach (in the context of MAP estimate) and the accuracy of the distance preservation can be derived. These bounds can be used to guide the data owner to control the privacy/accuracy tradeoff when perturbing the data.
- This perturbation offers higher privacy protection than distance preserving perturbation, with little loss of accuracy.

We believe that the privacy issues are intrinsically complex because they represent an intersection of legal, governmental, commercial, ethical and personal positions. It is not easy to produce one universal solution that addresses all these perspectives when the very definition of privacy is still open to debate. But the pressure is on to take more positive steps to encourage privacy protection while doing data mining to benefit the society. Many different PPDM techniques are now being proposed, questioned, and improved by researchers and technologists. Sociologists, policy experts, and legal experts are also encouraged to work together to articulate and enforce responsible data mining practices. We believe a good balance between the benefits in collecting and analyzing the data and the demand for privacy protection can be finally achieved. It takes time and effort, but it is worthwhile.

As an extension of this dissertation, we propose the following possible directions for future research.

Large scale distributed PPDM: Advances in computing and communication over wired and wireless networks have resulted in many pervasive distributed computing environments. Many of these environments deal with different distributed sources of

voluminous data, multiple compute nodes, and distributed user communities. Participating parties in such an environment may not all be ideal. Some may decide to behave like a "leech" to exploit the benefit of the system without contributing much. Some may intentionally try to collude with other parties to expose the private data of a specific individual. We believe that PPDM in a distributed scenario essentially looks like a game where each participant tries to maximize his/her benefit by optimally choosing the strategies during the entire PPDM process. Therefore, it is necessary to develop a game theoretic foundation of distributed PPDM, formulate PPDM algorithms based on that, and perform equilibrium analyzes.

Combination of Secure Multi-Party Computation and Perturbation Techniques Secure multi-party computation uses cryptographic protocols for privacy preserving distributed data mining. It offers strong privacy protection, but with high communication and computational complexity. On the other hand, data perturbation can efficiently distort the data, but with lower privacy guarantees. It would be ideal if we could combine these two techniques to achieve both efficiency and privacy.

REFERENCES

[1] A. L. Penenberg, "The end of privacy," *Forbes Magazine*, vol. Number 13, November 29 1999.

[2] B. Thuraisingham, "Data mining, national security, privacy and civil liberties," *SIGKDD Explorations*, vol. 4, no. 2, pp. 1–5, 2002.

[3] S. E. Committee, ""Data Mining" is NOT against civil liberties," http://www.acm.org/sigs/sigkdd/civil-liberties.pdf, June 30, 2003.

[4] R. Agrawal and R. Srikant, "Privacy preserving data mining," in *Proceedings of the ACM SIGMOD Conference on Management of Data*, Dallas, TX, May 2000, pp. 439–450.

[5] 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*, Melbourne, FL, November 2003.

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

[7] 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*, Dijon, France, April 2006, pp. 622–626.

[8] N. R. Adam and J. C. Worthmann, "Security-control methods for statistical databases: a comparative study," *ACM Computing Surveys (CSUR)*, vol. 21,

no. 4, pp. 515–556, 1989. [Online]. Available: http://portal.acm.org/citation.cfm?id=76895

[9] G. T. Duncan and S. Mukherjee, "Optimal disclosure limitation strategy in statistical databases: Dterring tracker attacks through additive noise," *Journal of The American Statistical Association*, vol. 95, no. 451, pp. 720–729, 2000.

[10] R. Gopal, R. Garfinkel, and P. Goes, "Confidentiality via camouflage: The cvc approach to disclosure limitation when answering queries to databases," *Operations Research*, vol. 50, no. 3, pp. 501–516, 2002.

[11] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems*, Santa Barbara, CA, 2001, pp. 247–255. [Online]. Available: http://portal.acm.org/citation.cfm?id=375602

[12] 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)*, Berlin, Germany, 2006.

[13] K. Muralidhar, R. Parsa, and R. Sarathy, "A general additive data perturbation method for database security," *Management Science*, vol. 45, no. 10, pp. 1399–1415, 1999.

[14] K. Muralidhar and R. Sarathy, "A theoretical basis for perturbation methods," *Statistics and Computing*, vol. 13, no. 4, pp. 329–335, 2003.

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

[16] 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)*, vol. 18, no. 1, pp. 92–106, January 2006. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.14

[17] 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*, Toronto, Canada, August 2004, pp. 67–82.

[18] K. Chen and L. Liu, “Privacy preserving data classification with rotation perturbation,” in *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM’05)*, Houston, TX, November 2005, pp. 589–592.

[19] 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)*, Berlin, Germany, September 2006, pp. 297–308.

[20] S. Mukherjee, Z. Chen, and A. Gangopadhyay, “A privacy preserving technique for euclidean distance-based mining algorithms using fourier-related transforms,” *The VLDB Journal*, p. to appear, 2006.

[21] S. L. Hansen and S. Mukherjee, “A polynomial algorithm for optimal univariate microaggregation,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 15, no. 4, pp. 1043–1044, 2003.

[22] J. Domingo-Ferrer and J. M. Mateo-Sanz, “Practical data-oriented microaggregation for statistical disclosure control,” *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 14, no. 1, pp. 189–201, 2002.

[23] J. Domingo-Ferrer and V. Torra, “Ordinal, continuous and heterogeneous k-anonymity through microaggregation,” *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 195–212, 2005.

[24] C. C. Aggarwal and P. S. Yu, “A condensation based approach to privacy preserving data mining,” in *Proceedings of the 9th International Conference on Extending Database Technology (EDBT’04)*, Heraklion, Crete, Greece, March 2004, pp. 183–199.

[25] 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)*, vol. 18, no. 9, pp. 1278–1283, 2006.

[26] L. Sweeney, “k-anonymity: a model for protecting privacy,” *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002. [Online]. Available: http://privacy.cs.cmu.edu/people/sweeney/kanonymity.html

[27] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, “Incognito: Efficient full-domain k-anonymity,” in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD’05)*, Baltimore, MD, June 2005, pp. 49–60.

[28] R. J. Bayardo and R. Agrawal, “Data privacy through optimal k-anonymization,” in *Proceedings of the 21st International Conference on Data Engineering (ICDE’05)*, Tokyo, Japan, April 2005, pp. 217–228.

[29] V. S. Iyengar, “Transforming data to satisfy privacy constraints,” in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD’02)*, Edmonton, Alberta, Canada, August 2002, pp. 279–288.

[30] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” in *Proceedings of the 22nd International Conference on Data Engineering (ICDE’06)*, Atlanta, GA, April 2006, p. 24.

[31] R. Chi-Wing, J. Li, A. W.-C. Fu, and K. Wang, “(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD’06)*, Philadelphia, PA, August 2006, pp. 754–759.

[32] N. Li and T. Li, “t-closeness: Privacy beyond k-anonymity and l-diversity,” in *Proceedings of the 23rd International Conference on Data Engineering (ICDE’07)*, Istanbul, Turkey, April 2007.

[33] T. Dalenius and S. P. Reiss, “Data-swapping: A technique for disclosure control,” *Journal of Statistical Planning and Inference*, vol. 6, pp. 73–85, 1982.

[34] S. E. Fienberg and J. McIntyre, “Data swapping: Variations on a theme by dalenius and reiss,” National Institute of Statistical Sciences, Research Triangle Park, NC, Tech. Rep., 2003.

[35] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy preserving mining of association rules,” in *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’02)*, July 2002.

[36] S. J. Rizvi and J. R. Haritsa, “Maintaining data privacy in association rule mining,” in *Proceedings of the 28th VLDB Conference*, Hong Kong, China, August 2002.

[37] 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.

[38] S. Agrawal and J. R. Haritsa, “A framework for high-accuracy privacy-preserving mining,” in *Proceedings of the 21st International Conference on Data Engineering (ICDE’05)*, Tokyo, Japan, April 2005, pp. 193–204.

[39] V. S. Verykios, A. K. Elmagarmid, B. Elisa, Y. Saygin, and D. Elena, “Association rule hiding,” in *IEEE Transactions on Knowledge and Data Engineering*, 2003.

[40] C. K. Liew, U. J. Choi, and C. J. Liew, “A data distortion by probability distribution,” *ACM Transactions on Database Systems (TODS)*, vol. 10, no. 3, pp. 395–411, 1985. [Online]. Available: http://portal.acm.org/citation.cfm?id=4017

[41] E. Lefons, A. Silvestri, and F. Tangorra, “An analytic approach to statistical databases,” in *Proceedings of the 9th International Conference on Very Large Data Bases*. Florence, Italy: Morgan Kaufmann Publishers Inc., November 1983, pp. 260–274. [Online]. Available: http://portal.acm.org/citation.cfm?id=673617

[42] A. C. Yao, “How to generate and exchange secrets,” in *Proceedings 27th IEEE Symposium on Foundations of Computer Science*, 1986, pp. 162–167.

[43] J. Kilian, “Founding cryptography on oblivious transfer,” in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, Chicago, IL, May 1988, pp. 20–31.

[44] S. Even, O. Goldreich, and A. Lempel, “A randomized protocol for signing contracts,” *Communications of the ACM*, vol. 28, pp. 637–647, 1985.

[45] O. Goldreich, *The Foundations of Cryptography*. Cambridge University Press, 2004, vol. 2, ch. 7. [Online]. Available: http://www.wisdom.weizmann.ac.il/~oded/foc-vol2.html

[46] B. Pinkas, “Cryptographic techniques for privacy preserving data mining,” *SIGKDD Explorations*, vol. 4, no. 2, pp. 12–19, 2002. [Online]. Available: http://portal.acm.org/citation.cfm?id=772865

[47] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *Advances in Cryptology - RUROCRYPT’99*, ser. Lecture Notes in Computer Science, J. Stern, Ed., vol. 1592, 1999, pp. 223–238.

[48] Ivan Damgård and M. Jurik, “A generalisation, a simplification and some applications of paillier’s probabilistic public-key system,” in *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, ser. Lecture Notes In Computer Science, vol. 1992. Springer-Verlag, 2001, pp. 119–136.

[49] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen, “On private scalar product computation for privacy-preserving data mining,” in *Proceedings of the The 7th Annual International Conference in Information Security and Cryptology (ICISC 2004)*, ser. Lecture Notes in Computer Science, Springer-Verlag, vol. 3506, Seoul, Korea, December 2004, pp. 104–120.

[50] R. Wright and Z. Yang, “Privacy-preserving bayesian network structure computation on distributed heterogeneous data,” in *Proceedings of the Tenth ACM SIGKDD*

Conference (SIGKDD'04), Seattle, WA, August 2004, pp. 713–718. [Online]. Available: http://www.cs.stevens.edu/~rwright/Publications/

[51] W. Du, Y. S. Han, and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," in *Proceedings of 2004 SIAM International Conference on Data Mining (SDM04)*, Lake Buena Vista, FL, April 2004. [Online]. Available: http://www.cis.syr.edu/~wedu/Research/paper/sdm2004_privacy.pdf

[52] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD'03)*, San Diego, CA, June 2003, pp. 86–97.

[53] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu, "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explorations*, vol. 4, no. 2, 2003.

[54] J. S. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, July 2002.

[55] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," in *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02)*, June 2002.

[56] X. Lin, C. Clifton, and Y. Zhu, "Privacy preserving clustering with distributed em mixture modeling," 2004, international Journal of Knowledge and Information Systems. To appear.

[57] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, D.C., August 2003.

[58] J. Vaidya, C. Clifton, and M. Zhu, *Privacy Preserving Data Mining*, ser. Series: Advances in Information Security. Springer, 2006, vol. 19.

[59] S. Laur, H. Lipmaa, and T. Mielikäinen, "Cryptographically private support vector machines," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'06)*, Philadelphia, PA, August 2006, pp. 618–624.

[60] M. Kantarcoglu and J. Vaidya, "Privacy preserving naive bayes classifier for horizontally partitioned data," in *IEEE ICDM Workshop on Privacy Preserving Data Mining*, Melbourne, FL, November 2003, pp. 3–9.

[61] R. Agrawal, R. Srikant, and D. Thomas, "Privacy preserving olap," in *Proceedings of the 2005 ACM SIGMOD International conference on Management of Data (SIGMOD'05)*, Baltimore, MD, June 2005, pp. 251–262.

[62] W. Du and Z. Zhan, "Building decision tree classifier on private data," in *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining*. Maebashi City, Japan: Australian Computer Society, Inc., December 2002, pp. 1–8. [Online]. Available: http://portal.acm.org/citation.cfm?id=850784

[63] H. Kargupta and K. Sivakumar, "Existential pleasures of distributed data mining," in *Data Mining: Next Generation Challenges and Future Directions*, H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha, Eds. MIT/AAAI press, 2004.

[64] B.-H. Park and H. Kargupta, "Distributed data mining," in *The Handbook of Data Mining*, ser. Human Factors and Ergonomics, N. Ye, Ed. Lawrence Erlbaum Associates, Inc., 2003, pp. 341–358. [Online]. Available: http://www.cs.umbc.edu/~hillol/PUBS/review.pdf

[65] K. Liu, H. Kargupta, J. Ryan, and K. Bhaduri, “Distributed data mining bibliography,” August 2004. [Online]. Available: http://www.cs.umbc.edu/~hillol/DDMBIB/

[66] S. Merugu and J. Ghosh, “Privacy-preserving distributed clustering using generative models,” in *Proceedings of the Third IEEE International Conference on Data Mining ICDM’03*, Melbourne, FL, November 2003.

[67] D. Meng, K. Sivakumar, and H. Kargupta, “Privacy sensitive bayesian network parameter learning,” in *Proceedings of The Fourth IEEE International Conference on Data Mining (ICDM’04)*. Brighton, UK: IEEE Press, November 2004. [Online]. Available: http://www.cs.umbc.edu/~hillol/pubs.html

[68] M. J. Atallah, E. Bertino, A. K. Elmagarmid, M. Ibrahim, and V. S. Verykios, “Disclosure limitation of sensitive rules,” in *Proceedings of the IEEE Knowledge and Data Engineering Workshop*, 1999, pp. 45–52.

[69] S. Oliveira and O. R. Zaiane, “Privacy preserving frequent itemset mining,” in *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining*. Maebashi City, Japan: Australian Computer Society, Inc., 2002, pp. 43–54. [Online]. Available: http://portal.acm.org/citation.cfm?id=850789

[70] Y. Saygin, V. S. Verykios, and C. Clifton, “Using unknowns to prevent discovery of association rules,” *SIGMOD Record*, vol. 30, no. 4, pp. 45–54, December 2001. [Online]. Available: http://portal.acm.org/citation.cfm?id=604271

[71] L. Chang and I. S. Moskowitz, “Parsimonious downgrading and decision tree applied to the inference problem,” in *Proceedings of the 1998 New Security Paradigms Workshop*, Charlottesville, VA, September 1998, pp. 82–89. [Online]. Available: http://citeseer.ist.psu.edu/376053.html

[72] 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*, vol. 3, no. 1, pp. 50–57, March 2004.

[73] K. Liu, "Privacy preserving data mining bibliography," October 2006. [Online]. Available: http://www.csee.umbc.edu/~kunliu1/research/privacy_review.html

[74] J. F. Traub, Y. Yemini, and H. Wozniakowski, "The statistical security of a statistical database," *ACM Transactions on Database Systems*, vol. 9, no. 4, pp. 672–679, 1984.

[75] K. Muralidhar, D. Batrah, and P. J. Kirs, "Accessibility, security, and accuracy in statistical databases: The case for the multiplicative fixed data perturbation approach," *Management Science*, vol. 41, no. 9, pp. 1549–1584, 1995.

[76] J. Kim, "A method for limiting disclosure in microdata based on random noise and transformation," in *Proceedings of the Survey Research Method Section*. Alexandria, Va: American Statistical Association, 1986, pp. 370–374.

[77] 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*, vol. 4, pp. 5–16, 2004.

[78] M. Artin, *Algebra*. Prentice Hall, 1991.

[79] P. H. Schoute, "Le déplacement le plus général dans l'espace à n dimensions," *Annales de l'École Polytechnique de Delft*, vol. 7, pp. 139–158, 1891.

[80] H. S. M. Coxeter, *Regular Polytopes*, 2nd ed., 1963, ch. XII, pp. 213–217.

[81] G. W. Stewart, "The efficient generation of random orthogonal matrices with an application to condition estimation," *SIAM Journal of Numerical Analysis*, vol. 17, no. 3, pp. 403–409, 1980.

[82] P. Diaconis and M. Shahshahani, "The subgroup algorithm for generating uniform random variables," *Probability in Engineering and Information Sciences*, vol. 1, pp. 15–32, 1987.

[83] B. Hore, S. Mehrotra, and G. Tsudik, "A privacy-preserving index for range queries," in *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04)*, Toronto, Canada, August 2004, pp. 720–731.

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

[85] L. Nachbin, *The Haar Integral*. Princeton, NJ: D. Van Nostrand Company, Inc., 1965.

[86] R. Heiberger, "Algorithm as 127: Generation of random orthogonal matrices," *Applied Statistics*, vol. 27, no. 2, pp. 199–206, 1978.

[87] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. Springer, 2002.

[88] G. J. Szekély and M. L. Rizzo, "Testing for equal distributions in high dimensions," *InterStat*, vol. November, no. 5, 2004.

[89] J. W. Osborne and A. B. Costello, "Sample size and subject to item ratio in principal component analysis," *Practical Assessment, Research and Evaluation*, vol. 9, no. 11, 2004. [Online]. Available: http://pareonline.net/getvn.asp?v=9&n=11

[90] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*, 2nd ed. John Wiley & Sons, INC., May 2004.

[91] P. Common, "Independent component analysis: A new concept?" *IEEE Transactions on Signal Processing*, vol. 36, pp. 287–314, 1994.

[92] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, no. 4, pp. 411–430, June 2000.

[93] X.-R. Cao and R.-W. Liu, "A general approach to blind source separation," *IEEE Transactions on Signal Processing*, vol. 44, pp. 562–571, 1996.

[94] W. B. Johnson and J. Lindenstrauss, "Extensions of lipshitz mapping into hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[95] S. Vempala, "Random projection: A new approach to vlsi layout," in *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS'98)*, Palo Alto, CA, 1998, pp. 389–395.

[96] J. M. Kleinberg, "Two algorithms for nearest-neighbor search in high dimensions," in *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*. ACM Press, 1997, pp. 599–608. [Online]. Available: http://portal.acm.org/citation.cfm?id=258653

[97] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the 30th Symposium on Theory of Computing*. Dallas, TX: ACM Press, 1998, pp. 604–613. [Online]. Available: http://portal.acm.org/citation.cfm?id=276876

[98] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," in *Proceedings of International Joint Conference on Neural Networks (IJCNN'98)*, vol. 1, Piscataway, NJ, 1998, pp. 413–418. [Online]. Available: http://citeseer.ist.psu.edu/kaski98dimensionality.html

[99] C. Giannella, K. Liu, T. Olsen, and H. Kargupta, "Communication efficient construction of decision trees over heterogeneously distributed data," in *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, Brighton, UK, November 2004, pp. 67–74.

[100] J. Buhler and M. Tompa, "Finding motifs using random projections," *Journal of Computational Biology*, vol. 9, no. 2, pp. 225–242, 2002.

[101] S. Dasgupta, "Experiments with random projection," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 143–151. [Online]. Available: http://portal.acm.org/citation.cfm?id=719759

[102] S. S. Vempala, *The Random Projection Method*, ser. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 2004, vol. 65.

[103] R. Hecht-Nielsen, "Context vectors: General purpose approximate meaning representations self-organized from raw data," *Computational Intelligence: Imitating Life*, pp. 43–56, 1994.

[104] R. I. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*. New York, NY: IEEE Computer Society,

October 1999, pp. 616–623. [Online]. Available: http://www-math.mit.edu/~vempala/papers/robust.ps

[105] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[106] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'06)*, Philadelphia, PA, August 2006, pp. 287–296.

[107] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*, H. Brezis, R. G. Douglas, and A. Jeffrey, Eds. Chapan & Hall/CRC, 1999.

[108] ——, *Matrix Variate Distributions*. CRC Press, September 1999.

[109] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[110] F. H. Walters, L. R. Parker, S. L. Morgan, and S. N. Deming, *Sequential Simplex Optimization*. Boca Raton, FL: CRC Press, 1991.

[111] M. L. Eaton and M. D. Perlman, "The non-singularity of generalized sample covariance matrices," *The Annals of Statistics*, vol. 1, no. 4, pp. 710–717, 1973.

[112] W. H ardle and L. Simar, *Applied Multivariate Statistical Analysis*. Springer, 2003, ch. 2.1, pp. 57–63.

[113] J. Eriksson and V. Koivunen, "Identifiability and separability of linear ica models revisited," in *Proceedings of the Fourth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, Nara, Japan, April 2003.

[114] M. S. Lewicki and T. J. Sejnowski, “Learning overcomplete representations,” *Neural Computation*, vol. 12, no. 2, pp. 337–365, 2000.

[115] P. Bofill and M. Zibulevsky, “Underdetermined blind source separation using sparse representation,” *IEEE Transactions on Signal Processing*, vol. 81, no. 11, pp. 2353–2362, 2001.

[116] F. J. Theis and E. W. Lang, “Geometric overcomplete ica,” in *Proceedings of the 10th European Symposium on Artificial Neural Networks (ESANN’2002)*, Bruges, Belgium, April 24-26 2002, pp. 217–223.

[117] K. Waheed and F. M. Salam, “Algebraic overcomplete independent component analysis,” in *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Source Separation*, Nara, Japan, April 1-4 2003.

[118] A. Taleb and C. Jutten, “On underdetermined source separation,” in *Proceedings of 24th International Conference on Acoustics, and Signal Processing*, vol. 3, Phoenix, AZ, March 1999, pp. 1445–1448.

[119] J. W. Demmel and N. J. Higham, “Improved error bounds for underdetermined system solvers,” Computer Science Department, University of Tennessee, Knoxville, TN, Tech. Rep. CS-90-113, August 1990. [Online]. Available: http://www.netlib.org/lapack/lawnspdf/lawn23.pdf

[120] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, ser. Prentice-Hall Series in Automatic Computation. Englewood Cliffs, New Jersey: Prentice-Hall, 1974, ch. 13.