2013

CONTEXT AWARE PRIVACY PRESERVING CLUSTERING AND CLASSIFICATION

Nirmal Thapa
University of Kentucky, nirmalthapa@uky.edu

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained and attached hereto needed written permission statements(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine).

I hereby grant to The University of Kentucky and its agents the non-exclusive license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless a preapproved embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's dissertation including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Nirmal Thapa, Student

Dr. Jun Zhang, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

2013

CONTEXT AWARE PRIVACY PRESERVING CLUSTERING AND CLASSIFICATION

Nirmal Thapa
University of Kentucky, nirmalthapa@uky.edu

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained and attached hereto needed written permission statements(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine).

I hereby grant to The University of Kentucky and its agents the non-exclusive license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless a preapproved embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's dissertation including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Nirmal Thapa, Student

Dr. Jun Zhang, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

CONTEXT AWARE PRIVACY PRESERVING CLUSTERING AND CLASSIFICATION

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Engineering at the
University of Kentucky

By
Nirmal Thapa
Lexington, Kentucky

Director: Dr. Jun Zhang, Professor of Computer Science
Lexington, Kentucky 2013

ABSTRACT OF DISSERTATION

CONTEXT AWARE PRIVACY PRESERVING CLUSTERING AND CLASSIFICATION

Data are valuable assets to any organizations or individuals. Data are sources of useful information which is a big part of decision making. All sectors have potential to benefit from having information. Commerce, health, and research are some of the fields that have benefited from data. On the other hand, the availability of the data makes it easy for anyone to exploit the data, which in many cases are private confidential data. It is necessary to preserve the confidentiality of the data. We study two categories of privacy: Data Value Hiding and Data Pattern Hiding. Privacy is a huge concern but equally important is the concern of data utility. Data should avoid privacy breach yet be usable. Although these two objectives are contradictory and achieving both at the same time is challenging, having knowledge of the purpose and the manner in which it will be utilized helps. In this research, we focus on some particular situations for clustering and classification problems and strive to balance the utility and privacy of the data.

In the first part of this dissertation, we propose Nonnegative Matrix Factorization (NMF) based techniques that accommodate constraints defined explicitly into the update rules. These constraints determine how the factorization takes place leading to the favorable results. These methods are designed to make alterations on the matrices such that user-specified cluster properties are introduced. These methods can be used to preserve data value as well as data pattern. As NMF and K-means are proven to be equivalent, NMF is an ideal choice for pattern hiding for clustering problems. In addition to the NMF based methods, we propose methods that take into account the data structures and the attribute properties for the classification problems. We separate the work into two different parts: linear classifiers and nonlinear classifiers. We propose two different solutions based on the classifiers. We study the effect of distortion on the utility of data.

We propose three distortion measurement metrics which demonstrate better characteristics than the traditional metrics. The effectiveness of the measures is examined on different benchmark datasets. The result shows that the methods have the desirable properties such as invariance to translation, rotation, and scaling.

KEYWORDS: Privacy, Matrix Factorization, Clustering, Classification

Author's signature: Nirmal Thapa

Date: December 17, 2013

CONTEXT AWARE PRIVACY PRESERVING CLUSTERING AND CLASSIFICATION

By
Nirmal Thapa

Director of Dissertation: Jun Zhang

Director of Graduate Studies: Miroslaw Truszczynski

Date: December 17, 2013

For my Mom, Dad and, Pragati.

ACKNOWLEDGMENTS

The work in the dissertation has been very intensive. I am thankful for so many hands helping and supporting me all the way.

First and foremost, I would like to thank my advisor Dr. Jun Zhang who gave me the opportunity to do this research and supported me for five years of PhD program. It was a great honor for me to work under his guidance.

I would like to thank my dissertation committes: Dr. Jerzy W. Jaromczyk (Department of Computer Science), Dr. Jinze Liu (Department of Computer Science) and Dr. Sen-ching Samson Cheung (Department of Electrical and Computer Engineering). I am thankful to my outside examiner Dr. Aaron Cramer (Department of Electrical and Computer Engineering) for his time and effort.

Thanks to Dr. Jie Wang at the Indiana University Northwest, and Mr. Lian Liu for help and advice with the Nonnegative Matrix Factorization.

I am grateful for being able to collaborate with Mr. Juergen Heit and Dr. Soundararajan Srinivasan of Robert Bosch RTC.

I would like to thank all my fellow officemates in the HiPSCCS : Dr. Changjinag Zhang, Dr. Danwei Han, Ruxin Dai, Xiwei Wang, and Pengpeng Lin.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Introduction

Data mining is an emerging field. It borrows ideas from different fields such as databases, artificial intelligence, and statistics. People have tried to study the connection between these fields [21, 68]. These techniques are the way of changing the data into useful information. In other words, data mining aims at extracting "knowledge" from certain data. The result of the process can be good or bad depending upon who gets the information and what is done with that information. Information provides support for the decision making of institutions. Scientists can verify important findings based on the data. On the flip side, data can potentially give away more information than needed, resulting in cases where the confidential and private information is given away. Personal medical data and information about products' trends are examples of private and confidential information. We list a couple of cases from [52] to show that privacy breach is a common but serious issue:

- A Michigan-based health system accidentally posted the medical records of thousands of patients on the Internet (The Ann Arbor News, February 10, 1999).
- An employee of the Tampa, Florida health department took a computer disk containing the names of 4,000 people who had tested positive for HIV, the virus that causes AIDS (USA Today, October 10, 1996).

Sweeney [58] found that 87% (216 million of 248 million) of the population in the United States had reported characteristics that likely made them unique based only on {5-digit zip, gender, date of birth}. Such data cannot be considered anonymous.

There are many health and other personal data that are publicly available. Information like those mentioned above can be derived from the data mining techniques as well. Regardless of the ways the disclosure of personal information takes place, the harm to the individual or the organization can be far reaching. We cite a couple of examples among several cases:

- A physician was diagnosed with AIDS at the hospital in which he practiced medicine. His surgical privileges were suspended. Estate of Behringer v. Medical Center at Princeton, 249 N.J. Super. 597.

- A 30-year FBI veteran was put on administrative leave when, without his permission, his pharmacy released information about his treatment for depression. Los Angeles Times, September 1, 1998.

The need for privacy is sometimes due to law (e.g., for medical databases) or can be motivated by business interests. However, there are situations where the sharing of data can lead to mutual gain. Large databases are mostly used in research, some of which are for scientific purposes while others can be for markets. The medical field can gain knowledge by utilizing data for research; so can many other businesses. Despite the potential gains, this is often not possible due to the confidentiality issues which arise.

When the threat comes from within the companies, the problem is more complicated. One way to handle such cases is preprocessing the data to ensure that all the confidential data are either taken out or hidden. The dissertation is applicable to two parts of this problem: the internal threats from within the company, and threat when data is made public. Integrating the privacy concern when applying different data mining techniques would enable wider acceptance for data mining into new services

and applications.

As mentioned earlier, when the data is preprocessed to remove some components or to hide confidential information, there is another contradictory concern about the utility of the data. Utility is the usefulness of the data. Does the processed data represent the same level of usefulness that it had originally? The simple answer to this is "NO". A practical requirement from the above described privacy concerns is a trade-off between sharing confidential information for analysis and keeping individual, corporate and national privacy. Usable data should have the characteristics of having good privacy and being usable.

Finding the right balance between data sharing and the privacy of the data has caught attention of many researchers from different fields. Researchers have tried to answer questions such as how different parties with data can cooperate with one another to achieve data mining without violating the privacy concerns, how the data mining techniques can be made aware of the privacy concern, what relations exist between different data distortion measure and the data mining techniques, and how we can preserve the data patterns with different set of values.

The result of all these efforts have led to a new field of data mining more well-known as Privacy Preserving Data Mining (PPDM). Researchers have paid attention to incorporating privacy protection mechanism into the data mining techniques that do not result in privacy breach. There have mainly been two types of work: perturbing the sensitive data and modifying the data mining techniques. Since the primary task in data mining is the development of models for decision making, developing accurate models without access to precise information in the original data is a natural objective for PPDM.

1.2 Privacy Preserving Data Mining

Privacy preserving can mainly be classified into the following two broad categories:

- **Private Data Access:** The threats to data are possible from inside as well as outside. The Wall Street Journal reported that companies were finding that insiders pose as much risk to computer security as outside attackers [60]. In recent years, companies with highly sensitive data have done a fairly good job of securing the network perimeter with firewalls and intrusion prevention systems, which have pushed attackers into looking for insiders to help them bypass these controls. Attacks from inside can be controlled by having some access control mechanism that has different levels of rights to different users in order to control data disclosure. Banks, as well as media companies are leading the way in adopting a trust-but-verify model of security to balance data protection against inside attack.

- **Public Data Access:** In many cases, data are easily available to public. The publicly available data can be used in conjunction with the background information the attacker has about the domain to breach the privacy. One obvious way to handle these data is to publish only nonconfidential parts of the data to the partners or the public.

Data can be divided into different types such as numerical-valued data vs. categorical-valued data or mixed-type data based on types of data, centralized data vs. distributed data based on the location of data. In our work data disclosure control is emphasized. We mainly concentrate on modifying the data. "Data" is understood as an abstract word for a combination of attribute values. In the dissertation, the follow-

Table 1.1: Example dataset

Name	Sex	Age	Zip-code	Occupation	Disease
Micheal	M	32	6500	Teacher	Sleepless
Nicole	F	25	62001	Govenment Employee	Back-pain
George	M	30	61025	Lawyer	Depression

ing two categories are used to describe the characteristics of our privacy-preserving methods:

- **Data Value Hiding:** Data Value Hiding (DVH) is to protect sensitive data values, but maintain data patterns in order to prevent improper use of data. We can take example from the data as in Table 1.1, which is a small subset of larger healthcare dataset A of patient profiles with attributes $\{Name,\ Sex,\ Age,\ Zip-code,\ Occupation,\ Disease\}$. This is a simple data that any medical institute can collect from its patients. $Name$ is the direct identifier of the individual and $\{Occupation, Disease\}$ are variables containing sensitive information of the individual. If this information is made public without any preprocessing, then George who is suffering from depression can have trouble convincing any clients to hire him. A simple approach might be to remove the name attribute, but this dataset set has another problem, the subset of $\{Sex, Age, Zip-code\}$ can provide inference on individual identification. The subset is also known as quasi identifiers. Let P, $\tilde{P}$ be the knowledge we want to learn from the original and the perturbed data.

  The goal of DVH is to hide those sensitive data usually by modifying the dataset. There are numerous ways proposed by different research works where attribute values are typically modified so that disclosure risks of sensitive/confidential attributes are minimized and the associated negative impact of data modification on data mining results are minimized [2, 3, 12]. A graphical representation is shown in Figure 1.2 where the aim is to maximize the difference between

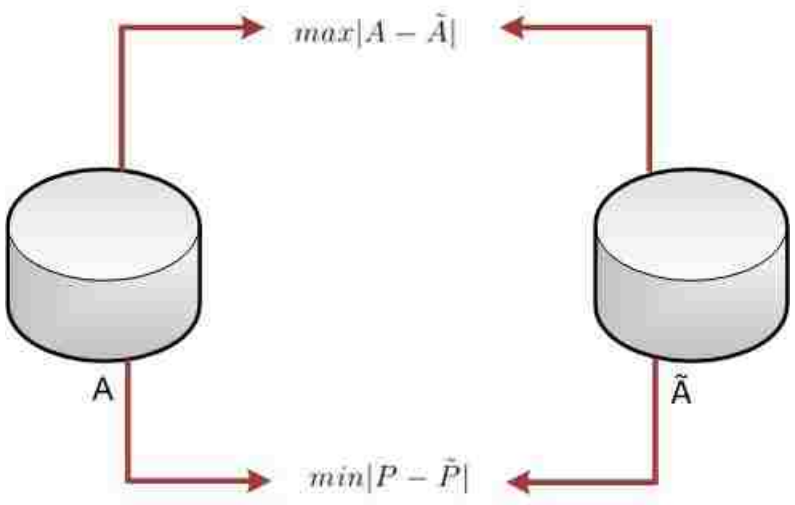


Figure 1.1: Data value hiding

an original dataset A and its modified version $\tilde{A}$ and minimize the difference between the data mining results on A and $\tilde{A}$.

Whenever a release of dataset A is required with sensitive attributes for different data mining purposes like clustering, prediction, regression, and classification; the common practice is to release $\tilde{A}$ which is a modified version of the data. A simple perturbation in this case can be the removal of the identifier columns through mechanisms like omission, generalization or anonymization. Generalization and anonymization give better utility than omission. Example utility of the data can be the ability of the data to correctly verify if the health issue can be tied with the profession. Tradeoff is to be considered between the utility and the perturbation.

- **Data Pattern Hiding:** PPDM also deals with Data Pattern Hiding (DPH). The result of data mining activities itself can compromise the privacy of individuals. Let us consider an example as in the Figure 1.2. The medical record of the individuals from the earlier example is published by taking out the sensitive attribute with some additional medical information. It is still possible to per-

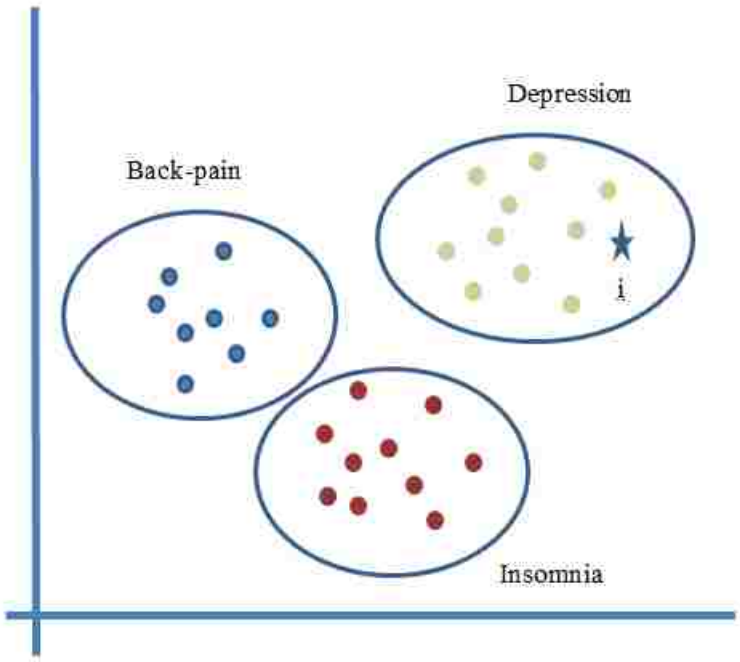


Figure 1.2: Data pattern hiding

form different data mining operations like clustering to gain more information. Clustering tries to group a set of objects and find whether there is some relationship between the objects; details of which are discussed in the later section. The following example shows the clustering performed on individuals based on the distance dividing them into different groups. With little background information such as the disease that another person in the same cluster has, it can be inferred what the subject i is suffering from.

Privacy concern like this is particularly true in the context of collaboration where multiple parties share data with each other; focus is gaining as much knowledge as possible from the combined dataset and at the same time hiding any sensitive data that can hamper their own business. Suppose $P1$ and $P2$ want to share data to gain further information both of them would like to make sure that other party does not gain any additional knowledge sensitive to their business model. $P2$ can carry out some data clustering techniques to group the

existing customers of $P1$ into two clusters: high potential valued customers and low potential valued customers; or a ranking algorithm can be executed to rank customer value. In either case, $P2$ can take advantage of the outcome of data mining and design a marketing strategy to win over the customers having a high possibility of future purchasing behavior. Probably, $P1$ will lose her customers and her business as well.

[61] addressed the problem of association rule mining where transactions are distributed across sources. Each site holds some attributes of each transaction, and the sites wish to collaborate to identify globally valid association rules. A well designed scenario is provided in [12] and Verykios et al. analyzed it to indicate the need not only to hide data attribute values, but also to prevent data mining techniques from discovering sensitive knowledge [63].

1.3 Related Work

A number of different works have been done in data privacy protection in data mining; some of the most popular techniques include randomization and k-anonymity. Some researchers have tried to apply cryptographical approaches while others have tried to use statistical measures. The database community has their own set of techniques. These techniques can mainly be divided into the following categories:

- **Data Perturbation**

  This is one of the more popular segments in PPDM. Common approaches to perturbation include additive data perturbation [3], multiplicative data perturbation [9, 43]. Evfimievski presents a technique that perturbs categorical data using random perturbation [17], but the aggregate information can be extracted with certain precision. Lin proposed a privacy preserving technique

for vertically partitioned data that uses randomized rotation [42]. Several reconstruction- or randomization-based methods adding some noise to the original data have been widely used for privacy protection [18, 50]. Cano suggested the use of synthetic data for privacy preserving data mining [7].

Xu pioneered the work on matrix decomposition based techniques for fulfilling the need of privacy preserving data mining through her work in[69, 70]. Wang worked on the same course and produced some exceptional work in [65, 66, 67]. She worked with the decomposition techniques like SVD and Nonnegative Matrix Factorization (NMF) for both the DVH and DPH.

- **Data Anonymization and Swapping**

  Work from [58] deals with a technique that can protect the privacy using k-anonymization. The work has been further extended in [47], it shows some shortcomings of k-anonymization. Gomatam et. al [24] proposed a decision-theoretic formulation of data swapping in which quantitative measures of disclosure risk and data utility are employed where decision variables are the swap rate, swap attribute(s) and possibly, constraints on the unswapped attributes.

- **Cryptographic/Secure Multi-Party Computation (SMC)**

  [55] showed that non-trusting parties can jointly compute functions of their different inputs while ensuring that no party learns anything but the defined output of the function. These results were shown using generic constructions that can be applied to any function that has an efficient representation as a circuit. Other works include: [16, 4].

- **Privacy Preserving based on the Data Mining Techniques**

  In addition to these methods based on distorting the original data values, Clifton et al. proposed another class of approaches to modify data mining algorithms so that they allow data mining operations on distributed datasets without knowing

the exact values of the data or without directly accessing the original data [11]. The data mining techniques studied are: classification [3], association rule mining [61], clustering [62], Bayes classifier [29], collaborative filtering [56], and data stream mining [53].

- **Privacy Preserving in Distributed System**

  With the size of data growing rapidly, efforts have been put into techniques that can be implemented in distributed systems. [23] defines a new privacy model "k-privacy" by means of the accepted trusted third party model. This allows implementing cryptographically secure efficient primitives for real-world large-scale distributed systems.

1.4 Applications of PPDM

The consequences that an individual or an institution has to bear due to privacy breach are huge, which has led to greater attention. There are numerous areas where PPDM can be employed. Below we list a few areas of applications [64]:

- Homeland Security Applications:

  Homeland Security has to be cautious regarding privacy while matching the subject of credential and the person presenting the credential. For example, the theft of social security numbers presents a serious threat to homeland security. The credential validation approach tries to exploit the semantics associated with the social security number to determine whether the person presenting the social security number credential truly owns it. Two commonly used case scenarios are credential validation problem and web camera surveillance.

- Video Surveillance

  A significant threat to privacy is face recognition software, which can automat-

ically identify known people from a drivers license photo database, and thereby track people regardless of suspicion.

- Genomic Privacy

  Recent advances in genomics prompt a formidable privacy challenge: As the price of a complete genome profile has declined for genome-wide genotyping, wide-spread usage of genomic information is about to become a reality. DNA data is considered extremely sensitive since it contains almost uniquely identifying information about an individual. As in the case of multidimensional data, simple removal of directly identifying data such as social security numbers is not sufficient to prevent re-identification.

  The impact of increased availability of genomic information on privacy, however, is unprecedented, for obvious reasons: First, genetic conditions and predisposition to specific diseases (such as Alzheimer's) can be revealed. Second, one′s genomic information leaks substantial information about one′s relatives. Third, complex privacy issues can arise if DNA analysis is used for criminal investigations, epidemiological research, and personalized medicine purposes.

- Multi-party Computation

  In many cases, data are distributed across different parties. It makes sense to be able to mine the data from both of the parties without each party knowing the exact underlying data from other party. Secure multi-party communication refers to the computation protocols that make sure no party involved knows anything but its own inputs and the results, i.e., the view of each party during the execution can be effectively simulated by the input and output of the party. In the late 1980s, work on secure multi party communication demonstrated that a wide class of functions can be computed securely under reasonable assumptions without involving a trusted third party.

- Social Networks

  With the rise in everyday use of social networks like Facebook and Twitter, it has become even more important to be cautious of any private information. A while back, Facebook launched a new advertising campaign called Sponsored Stories that incorporated users′ "like" into advertisements. It offered peer endorsement of products and a way for Facebook to make money. In addition to the benefits of using social network sites, there may be risks associated with using such services. For example, research has begun exploring what kinds of personally identifiable information (e.g., phone numbers, email address, postal address, social security numbers, etc.) people share through services such as Facebook and MySpace [34, 37]. The misuse of personally identifiable information obtained online can raise many privacy concerns, such as identity theft or even discrimination [46].

- Research

  Research on various healthcare data requires that proper care is taken when the data is used for experiments or studies as healthcare data are highly sensitive. There are measures in place that ensure that individuals will be informed of uses and disclosures of their medical information for research purposes, and their rights to access information.

1.5 Data Mining Techniques

Data context is a major part of this dissertation. Especially in the case of privacy preserving data mining, it is important that we define the context: how the data is utilized, what information is to be preserved, and what are the data mining techniques to be used on the data. Among the different mining techniques, this section briefly discusses the methods on which we concentrate in the following chapters:

1.5.1 Clustering

Clustering is an important technique in the data mining community. A cluster is a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. Patterns that are hidden can be found using clustering techniques. The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. Clustering algorithms can be applied in many fields, for instance [49]:

- Marketing: finding groups of customers with similar behavior given a large database of customer data containing their information and past buying records;
- Insurance: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
- Biology: classification of plants and animals given their features;
- City-planning: identifying groups of houses according to their house type, value and geographical locations;
- Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones;
- WWW: document classification; clustering weblog data to discover groups of similar access patterns.

K-means

There are many clustering algorithms, like hierarchical clustering and density based clustering. K-means is one of the most popular clustering algorithms. As we are concerned with clustering problem, we employ k-means for the experiments.

The basic objective of k-means is to cluster the n data items that can be represented by $(x_1, x_2, ..., x_n)$, into k sets $(k \leq n)$ such that $S = (S_1, S_2, ..., S_k)$ so as to

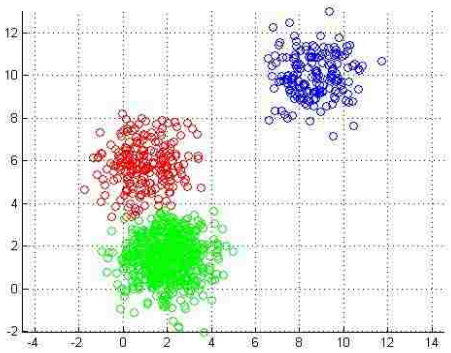


Figure 1.3: K-means Clustering

minimize the distance within the clusters:

$$\min(\sum_{i=1}^{k}\sum_{x_j \epsilon S_i} \|x_j - \mu_i\|^2) \tag{1.1}$$

where μ_i is the mean of points in S_i. Euclidean distance is often used as the metric.

K-means Algorithm

Algorithm 1: K-means Clustering Algorithm

input : $k, data$
output: $Clusters$
Initialize k centroids
while (*elements change cluster*) **do**
  Assign each point to the nearest mean
  Move "mean" to center of its cluster.

1.5.2 Regression Analysis

Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. More specifically, regression analysis helps one understand how

the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables. Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning.

Regression models

Regression models involve the following variables:

- The unknown parameters, denoted as β, which may represent a scalar or a vector.
- The independent variables, X.
- The dependent variable, Y.

In various fields of applications, different terminologies are used in place of dependent and independent variables.

A regression model relates Y to a function of X and β.

$$Y \approx f(X, \beta) \tag{1.2}$$

The approximation is usually formalized as $E(Y|X) = f(X, \beta)$. To carry out regression analysis, the form of the function f must be specified. Sometimes the form of this function is based on knowledge about the relationship between Y and X that does not rely on the data. If no such knowledge is available, a flexible or convenient form for f is chosen.

1.6 The Contributions of the Dissertation

This dissertation is focused on studying the privacy aspects of data mining and designing methods that protect privacy in the process of data mining. Privacy can be

defined differently depending on the situation and the technique that are being applied. It can be an attribute value of a particular subject or the cluster membership of the subject or the relationship between attributes. In our work, we put context into the center of discussion. We follow it with the privacy concerns that the context might have and how it can be answered. In terms of the contributions of the dissertation, our research can be divided into five parts.

- For the first part, my objective is to employ NMF for pattern hiding. NMF has found applications in many fields, specifically in clustering applications. We try to exploit the particular property of NMF for membership hiding for some particular subjects so that the real cluster membership of the subjects is not revealed. We explicitly define the constraint on the update rule, so that the matrix factorization results in the membership hiding. I call this part of my work Constrained NMF for Data Membership Protection.

- The second part of my work deals with Data Privacy protection without taking the membership into consideration. We propose two types of constraints which achieve our objective of distorting the data, but at the same time retaining the subject membership. The chapter tries to balance the two difficult and contradictory objectives between privacy and utility. As mentioned in the first part we provide explicit constraints on both methods. This part of my research is called as Constrained NMF for Data Protection.

- The third part of my work tailors novel approaches to privacy by utilizing the knowledge about the type of classifier used on the data. We study linear classifiers and their dependence on the correlation between the data attributes. We analyze how distortion in a particular way leads to better utility and larger distortion of the data. We present our results with different linear classifiers in Context Aware Privacy Preserving Data Mining for Linear Classifiers.

- We answer privacy concerns with linear classifiers in Chapter 3, but my method lacks the same performance with non-linear classifiers. We study why the technique does not extrapolate the same level of performance with non-linear classifiers. We take different types of Support Vector Machines (SVM) as a base to understand how the non-linear classifiers work. Neighborhood Aware Privacy Preserving Data Mining for Non-Linear Classifiers addresses classification with non-linear classifiers.

- Our final research called Distortion Measurement Techniques, deals with developing better distortion metrics. Some of the metrics that have been in use have some issues that need to be addressed. In this research, we focus on three techniques that have properties of better distortion metrics. We provide some theoretical analysis on how they work better than the other methods that we have used.

Chapter 2 Preliminaries

In this dissertation we consider datasets with n subjects and m attributes. Our study deals with methods for classification and clustering. We study the effect of different perturbation methods on different datasets and compare the utility and distortion level. We interchangeably use distortion and perturbation to mean the same.
This chapter describes concepts used in our research: definitions, preprocessing steps, metrics, and the datasets for experiments.

2.1 Definitions

We present a few definitions related to our study.

2.1.1 Data Model D

Given a dataset D consisting of n independent subjects in an m-dimensional feature space, with each subject having m numerical features, if we denote the ith subject of D as D_i, then

- $D = \{D_i\}_{i=1}^{n}$
- $D_i = \{d_{i1}, d_{i1}, ..., d_{im}, \}_{i=1}^{n}$

2.1.2 Vector Space Data Model A

Given a data model D, which can be represented by a matrix A, $A \in \mathbb{R}^{n \times m}$, with the rows corresponding to the n subjects and the columns to the m features, if the ith row is denoted by A_i, then A_i represents D_i. The jth feature is represented by the jth column of A, denoted by A_j.

$$A = \begin{bmatrix} A_1 \\ A_2 \\ . \\ . \\ A_n \end{bmatrix}$$

2.1.3 Data Modification

Given two datasets D and $\tilde{D}$ with the corresponding matrix models of A and $\tilde{A}$, and a modification scheme M, a sequence of modifications is a function ψ to transform A into $\tilde{A}$, where F indicates the subjects to be modified.

$\psi : (A, F, M) \rightarrow \tilde{A}$

2.2 Data Preprocessing

In data mining, data often contains noisy results and in some cases, data might be in different units or formats. It is important to convert those data to meaningful states for further analysis. Normalization is one of the important steps in data mining.

2.2.1 Normalization

There are several normalization techniques and the choice is problem-specific. Assuming A as the dataset we have used a couple of normalization steps

- **Range adjustment**. It is common that the attributes have different value ranges. We can normalize their value ranges to a unit range. Each attribute is normalized by its value range as

$$A_{ij} \leftarrow A_{ij} \times \frac{A_j - min(A_j)}{max(A_j) - min(A_j)} \tag{2.1}$$

  where A_{ij} is an element at position i,j and A_j represents the j^{th} column of dataset A. This is necessary when working with NMF techniques where data has to be positive.

- **Unit-length normalization**. Each attribute column vector can be normalized to the unit length as

$$A_j \leftarrow \frac{A_j}{||A_j||} \tag{2.2}$$

where $||A_j||$ is the length of A_j, i.e., the 2-norm of A_j.

2.3 Data Utility Metrics

It is not only important to quantify the distortion level(s) of the data but also the utility. In the case of the clustering technique, we have used the results of the k-means algorithm to calculate the accuracy of the techniques while performing clustering. The ground truth is established by running k-means clustering on the original data. All the comparisons on distorted matrices are based on the ground truth. We distort the original data and then run k-means again to cluster the resulting data. We check how different the result is from the k-means run on the original data. In our resarch, we call that similarity as *accuracy* which is the percentage of the distorted data that are correctly classified based on the ground truth established. For the classification tasks, we use the class label as the ground truth. Several techniques have been used, such as linear regression, decision trees, and naive Bayes.

- **Clustering accuracy (*Accuracy*)**:

$$Accuracy = 1 - \frac{kmeans(A) - kmeans(\tilde{A})}{SizeOf(A)} \tag{2.3}$$

- **Misclassification rate (*Misrate*)** :

  Misclassification rate is the ratio of difference between actual class and the predicted class to the total size of items in the prediction.

$$Misrate = \frac{actual\ class - predicted\ class}{SizeOf(A)} \tag{2.4}$$

  Misrate is similar to accuracy except for the fact that Misrate is calculated for classification accuracy.

- **Area Under Curve (AUC)**:

  Plotting receiver operating characteristic (ROC) curves are a popular way of displaying the discriminatory accuracy of a diagnostic test for detecting whether a particular incident happened.

  Let us define an experiment from P positive instances and N negative instances. The four outcomes can be formulated in a 2×2 contingency table or confusion matrix, as follows:

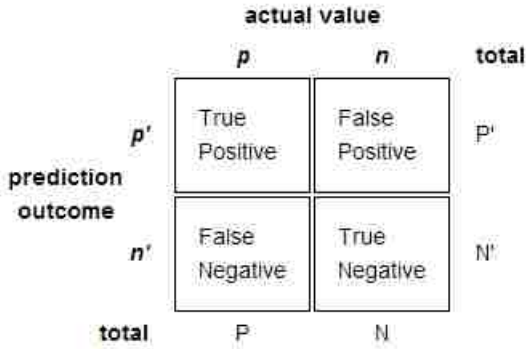


Figure 2.1: Confusion Matrix

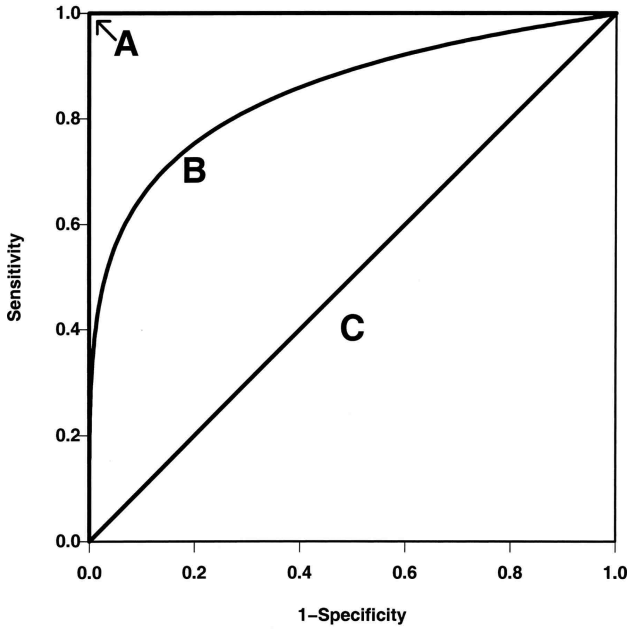


Figure 2.2: ROC Curve

The most commonly used global index of classification accuracy is the area under the ROC curve (AUC). When using normalized units, the AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. The AUC gives a number representing the accuracy of the method for the ROC curve. Since the AUC of 0.5 represents an ineffective method, favorable tests should result in AUC value close to 1.

Sensitivity is the true positive rate, while specificity is the true negative rate, calculated as

$$Sensitivity = \frac{True\ Positive}{Positive} \tag{2.5}$$

$$Specificity = \frac{True\ Negative}{Negative} \tag{2.6}$$

As is shown in Figure 2.2, A represents ideal classification with AUC=1, C represents ineffective classification as it has AUC=0.5 while B is somewhere in between.

2.4 Value Distortion Metrics

The privacy protection measure should indicate how closely the original value of an item can be estimated from the distorted data [56]. Some privacy metrics have been proposed in the literature [69], [2], and [18]. Some data distortion measures defined in [69] are used here to assess the level of data distortion which only depends on the original matrix A and its distorted counterpart $\tilde{A}$.

This section discusses data value distortion metrics used in this dissertation.

2.4.1 Value Difference

After a data matrix is distorted, the values of its elements change. The Value Difference (VD) of a dataset is represented by the relative value difference in the Frobenius norm. Thus, VD is the ratio of the Frobenius norm of the difference of A from $\tilde{A}$ to the Frobenius norm of A, given as

$$VD = \frac{\|A - \tilde{A}\|_F}{\|A\|_F} \tag{2.7}$$

2.4.2 Rank Position

The Rank Position (RP) is used to denote the average change of rank for all attributes. After the elements of an attribute are distorted, the rank of each element in an ascending order of its value changes. Assume that dataset A has n data objects and m attributes. $Rank_j^i$ denotes the rank of the j^{th} element in attribute i, and $\tilde{Rank}_j^i$ denotes the rank of the distorted element A_{ji} . Then RP is defined as

$$RP = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}|Rank_j^i - \tilde{Rank}_j^i|}{n \times m} \tag{2.8}$$

2.4.3 Rank Maintainance

The Rank Maintainance (RM) represents the percentage of elements that keep their value ranks in each column after the distortion. It is computed as

$$RM = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} Rk_j^i}{n \times m} \tag{2.9}$$

where Rk_j^i means whether an element keeps its position in the order of values.

$$Rk_j^i = \begin{cases} 1 & \text{if } Rank_j^i = \tilde{Rank}_j^i \\ 0 & otherwise \end{cases}$$

If an element keeps its position in the order of values, $Rk_j^i = 1$, otherwise, $Rk_j^i = 0$.

Table 2.1: Data Perturbation Metrics

Metric Formula	Parameter Description
$VD = \frac{\|A-\tilde{A}\|_F}{\|A\|_F}$	where $A \in \mathbb{R}^{n\times m}$
$RP = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\|Rank_j^i - \tilde{Rank}_j^i\|}{n\times m}$	$\tilde{Rank}_j^i$ is the rank for perturbed data
$RM = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} Rk_j^i}{n\times m}$	$Rk_j^i = \begin{cases} 1 & \text{if } Rank_j^i = \tilde{Rank}_j^i \\ 0 & otherwise \end{cases}$

2.4.4 Attribute Rank Change

Content of an attribute can be inferred from its relative value difference compared with other attributes. It is desirable that the rank of the average value of each attribute vary after the data distortion. The Attribute Rank Change (CP) can be calculated as

$$CP = \frac{\sum_{i=1}^{m} |RAV_i - \tilde{RAV}_i|}{m} \tag{2.10}$$

2.4.5 Attribute Rank Maintenance (CK)

Similarly to RK, CK is defined to measure the percentage of the attributes that keep their ranks of average value after the distortion. So, it is calculated as

$$CK = \frac{\sum_{i=1}^{m} Ck_i}{m} \tag{2.11}$$

where Ck_i is computed as

$$Ck_i = \begin{cases} 1 & \text{if } RAV_i = \tilde{RAV}_i, \\ 0 & otherwise \end{cases}$$

2.5 Datasets

The following datasets from UCI machine learning repository have been used for our study. We have used box plots to display the distribution of data since it provides

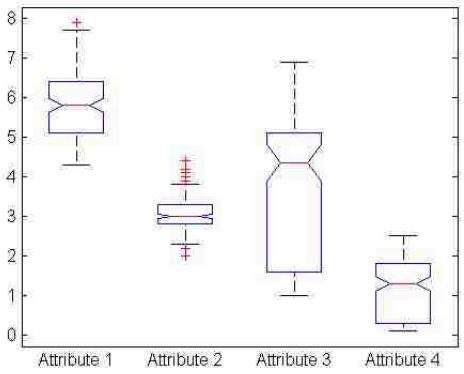


Figure 2.3: Boxplot for Iris Dataset

information like minimum, first quartile, median, third quartile, and maximum. In the simplest box plot the central rectangle spans the first quartile to the third quartile. It also simplifies the observation of outliers present in the dataset.

2.5.1 IRIS Dataset

IRIS is a very simple dataset with 150 instances in a 4-dimensional attribute space. This is perhaps the best known dataset to be found in the pattern recognition literature. The four attributes are sepal length, sepal width, petal length, and petal width. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant: Setosa, Versicolour and Virginica. Setosa is linearly separable from the other two; the latter two are not linearly separable from each other.

Figure 2.4 shows the boxplots of four attributes grouped by three classes. This figure demonstrates that the 3rd or 4th attributes are highly related to the class labels; either one can accurately filter the Setosa out.

2.5.2 YEAST Dataset

The YEAST is a real-valued data set having 1484 instances and 8 attributes. It is used to predict the localization site of protein, which has 10 predications in Table

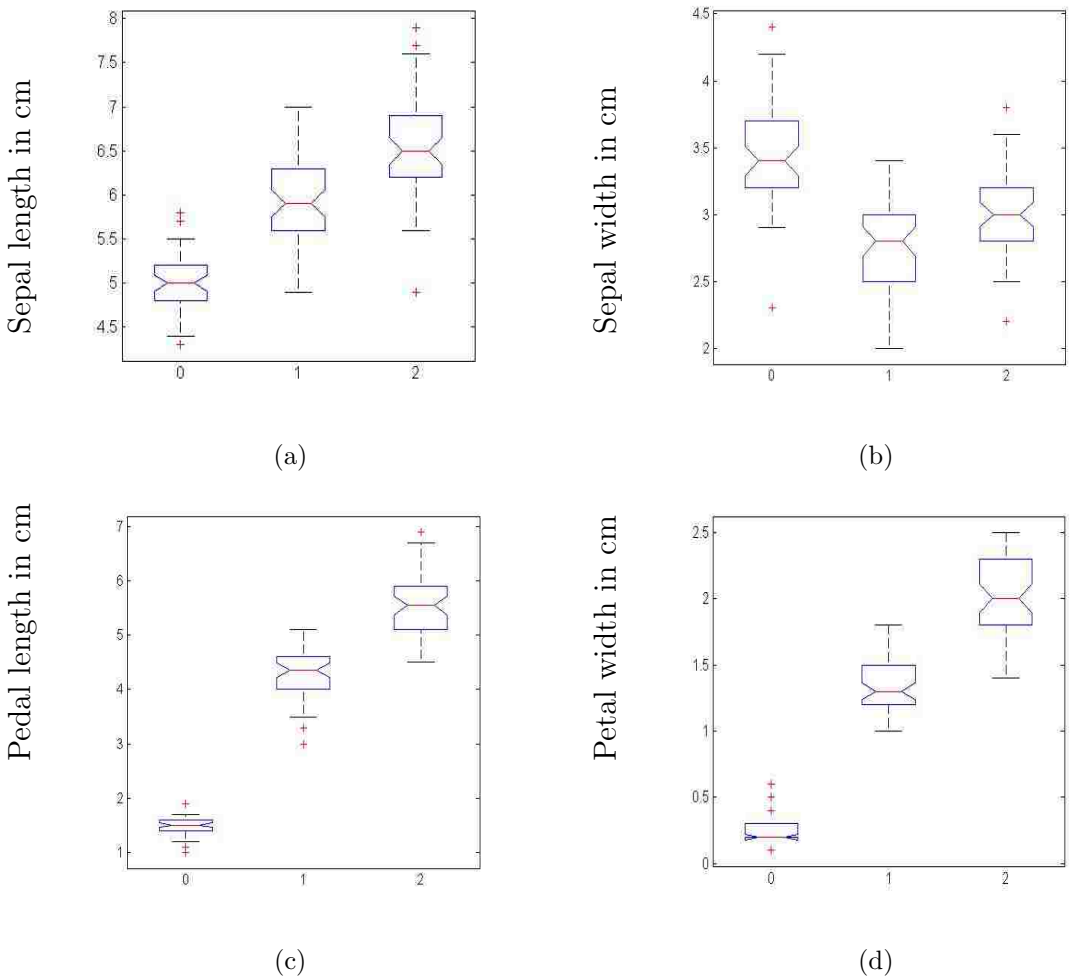


Figure 2.4: Boxplots of 4 attributes of the IRIS data set grouped by 3 classes.

2.2.

From the Figure 2.5, we can observe that attributes 5 and 6 do not have as much variance as the other attributes. All the attributes are in [0, 1].

2.5.3 Connectionist Bench (Sonar Mines vs. Rocks) Dataset

The Connectionist Bench Dataset contains 111 patterns for mines obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions

Table 2.2: YEAST Dataset

Number	1	2	3	4	5	6	7	8	9	10
Class	CYT	NUC	MIT	ME3	ME2	ME1	EXC	VAC	POX	ERL
Size	463	429	244	163	51	44	35	30	20	5

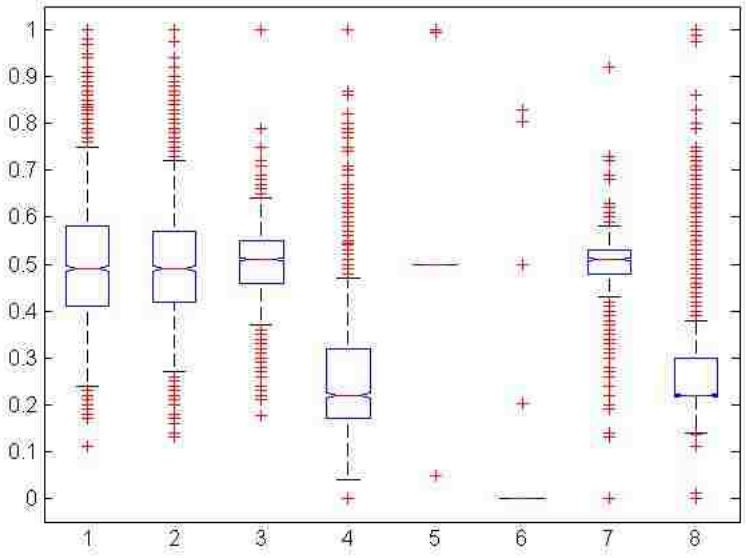


Figure 2.5: Boxplot for YEAST Dataset

and 97 patterns for rocks obtained from rocks under similar conditions. The transmitted sonar signal is a frequency-modulated chirp, rising in frequency. The dataset contains signals obtained from a variety of different aspect angles, spanning from 90 degrees for the cylinder to 180 degrees for the rock.

Each pattern is a set of 60 numbers in [0.0, 1.0]. Each number represents the energy within a particular frequency band, integrated over a certain period of time. The integration aperture for higher frequencies occurs later in time, since these frequencies are transmitted later during the chirp.

The label associated with each record contains the letter "R" if the object is a rock and "M" if it is a mine (metal cylinder). The numbers in the labels are in increasing order of aspect angle, but they do not encode the angle directly.

2.5.4 Wine Quality Dataset

The Wine Quality Dataset is related to red and white variants of the Portugese "Vinho Verde" wine [14]. It has 11 independent attributes (inputs) which are mostly

physicochemical data. Wine has the quality score (output) of [0,10] based on the sensory data.

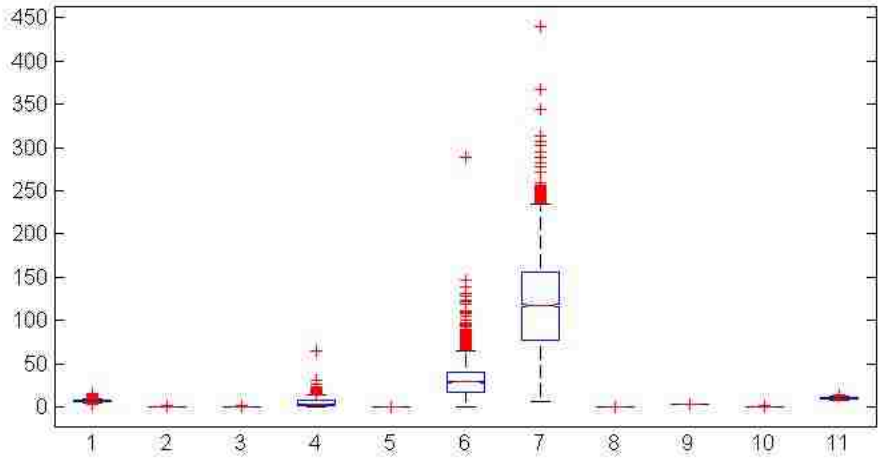


Figure 2.6: Boxplot for Wine Quality Dataset

The Figure 2.6 shows that the attributes have large difference in the range. The dataset needs to be normalized before applying data mining techniques.

2.5.5 Spambase Dataset

The concept of "spam" is diverse: advertisements for products/web sites, get rich quick schemes, and chain letters. This dataset consists of different attributes that

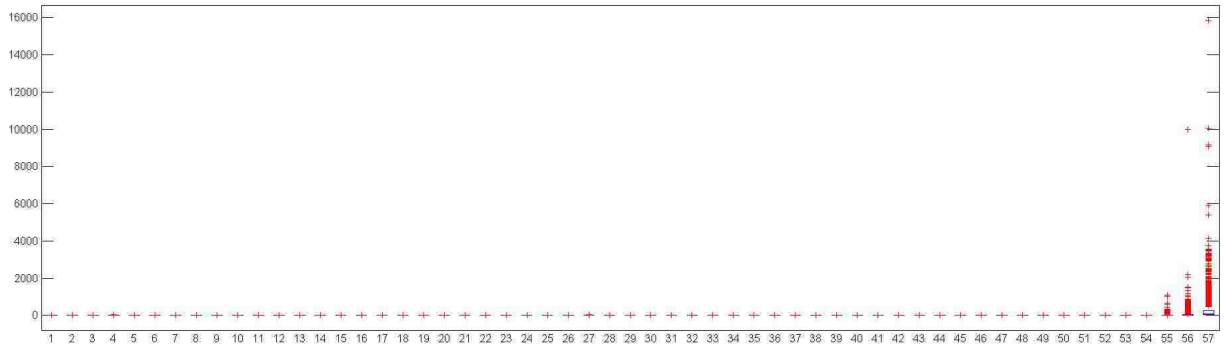


Figure 2.7: Boxplot for Spambase Dataset

classify emails as either spam or not. The collection of spam e-mails came from postmaster and individuals who had filed spam. Similarly, the collection of non-spam e-mails came from filed work and personal e-mails.

From the Figure 2.7, it is difficult to observe the range of the attributes, but we can see that attributes 55 - 57 have more variance compared to the other attributes. Moreover, those attributes have larger magnitudes.

Table 2.3: Summary of different datasets

Dataset	Items	Attributes	No. of classes
IRIS	150	4	3
YEAST	1484	8	10
SONAR	208	60	3
WINE	4898	12	2
SPAM	4601	57	2
MAGIC	19020	11	3

2.5.6 Magic Gamma Telescope Dataset

The Magic Gamma Telescope Dataset consists of two types: gamma(g) and hadron(h). The dataset was generated by a Monte Carlo program, Corsika as mentioned in [19].

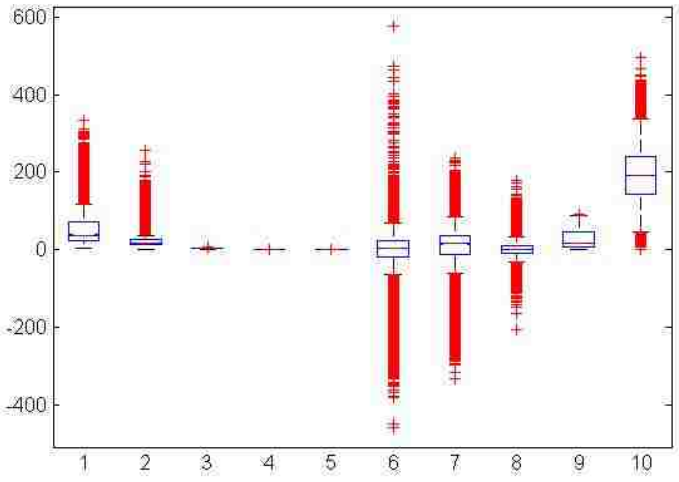


Figure 2.8: Boxplot for Magic Gamma Dataset

The Figure 2.8 reveals attributes 3 - 5 have lower variance, other attributes have lots of outliers as well. The Table 2.3 summaries the size and number of classes present in different datasets.

Chapter 3 Constrained Nonnegative Matrix Factorization for Data Pattern Hiding

3.1 Constrained Nonnegative Matrix Factorization for Hiding Cluster Membership of Data

Clustering is a very widely studied topic that has been used in different areas including machine learning, data mining, pattern recognition, image analysis, information retrieval, etc. There are many algorithms available for clustering. Among them, k-means is one of the most popular and widely used techniques. Work utilizing NMF for clustering is not a new idea but [15] goes one step further and presents the idea of similarity between the k-means and NMF. In this chapter, we present our idea of combining clustering and NMF for the purpose of membership hiding by imposing additional constraint on NMF. NMFs with additional constraints like orthogonality constraint [39] and sparseness constraint [27] have been applied to various fields. Our study uses constrained nonnegative matrix factorization for the purpose of hiding particular membership in a data analysis task. Some initial works in this field include applying NMF for privacy protection, which was done by Wang et al. [66, 65]. The work by Wang et al. [66] applies NMF in the first phase and then tries to suppress the data pattern using different ad-hoc algorithms. This chapter proposes explicit incorporation of the additional constraint in order to suppress the data patterns in the process of performing the matrix factorization. The advantage being the factorization and the suppression is a single stage operation. We start with the brief introduction of Nonnegative Matrix Factorization.

3.1.1 Overview of NMF

There are many kinds of matrix factorizations like principal component analysis (PCA), singular value decomposition (SVD), and NMF. Nonnegative matrix factorization is a way in linear algebra where a nonnegative valued matrix A is decomposed into the product of two nonnegative valued matrices H and W. NMF imposes additional constraint that none of the elements of the factor matrix H and the basis matrix W can be negative. Another notable property about NMF is that, results are non-unique which provides even better ground for it to be used for data protection. The following equations explain the relations as

$$nmf(A) \Rightarrow H \times W$$

$$A = H \times W + R, A \approx H \times W = \tilde{A}$$

where R is the residual since $H \times W$ may not be equal to A. [64] defined NMF as "*Given a nonnegative data model A(n×m), find two nonnegative matrices* $H^{n\times k}$ *and* $W^{k\times m}$ *with k being the number of clusters in A, that minimize Q, where Q is an objective function defining the nearness between the matrices A and HW. The modified version of A is denoted as* $\tilde{A}$*= H× W* ". Generally, $k < \min(m, n)$, which may reduce the rank of the original matrix. In other words, the original matrix will be compressed. There are two main aspects, one is the *objective function* and the other is the *update rule*. The objective function quantifies the quality of factorization usually in terms of distance between the two matrices A and HW. The Euclidean distance or the Frobenius norm is the common function to consider. The objective for NMF is to minimize the distance between A and HW.

$$\min_{H\geq 0, W\geq 0} f(A, H, W) = \|A - HW\|_F^2 \tag{3.1}$$

Since, NMF is an iterative technique; there is the need to update matrices H and W in each iteration. The Rule to do so is termed as *update rule*. We will discuss more on that in the following sections.

3.2 NMF and K-means Clustering

In the distance-based hard clustering, subject A_i is assigned to cluster C_k if it is closest to the centroid, c_k. Variation on its distances to the K centroids might incur a shift of A_i from its old cluster to a new cluster.

Ding et al. [15] showed that there is some connection between k-means clustering and NMF. Based on their relationship, a data pattern hiding approach [65] is proposed to change the cluster membership.

The clustering solution can be represented by a nonnegative cluster indicator matrix $D \in \mathbb{R}^{n \times K}$ as in [15], $D = (D_1, D_2, ..., D_K)$, c_k represents the center of k^{th} cluster. $|C_k|$ is the size of the kth cluster. For the hard membership, we set

$$D_{ik} = \begin{cases} \frac{1}{\sqrt{|C_k|}} & \text{if } A_i \in C_k \\ 0 & otherwise \end{cases}$$

Each D_k is normalized to unit length so that $D^T D = I$.

The elements of D are between 0 and 1 and the sum of the elements in each row of D is equal to 1. The significance of D_{ik} is that it denotes the membership of A_i or for the soft clustering, it reflects the degree to which A_i associates with cluster C_k.

Especially, the centroids $\{c_1, c_2, ..., c_K\}$ can be represented as

$$(c_1\sqrt{|C_1|}, c_2\sqrt{|C_2|}, ..., c_k\sqrt{|C_K|})^T = D^T A \tag{3.2}$$

We use $\tilde{C}$ to denote $D^T A$.

For the kth cluster, the sum of all the members in C_k can be represented in terms of the kth row of $D^T A$ as

$$\sum_{A_i \in C_k} A_i = \sqrt{|C_k|}(D^T A)_k = \sqrt{|C_k|}(\tilde{C})_k \tag{3.3}$$

Now if we use D as a representation of the clustering solution, then the objective function for seeking a D, given A can be encoded with a symmetric convex coding (SCC) model J that is built on S [45].

$$\min_{D \in \mathbb{R}_+^{n \times K}, B \in \mathbb{R}_+^{K \times K}, D^T D = I, B^T = B} J = ||S - DBD^T||^2 \tag{3.4}$$

where, B is $K \times K$ symmetric matrix. S is defined as

$$S = (S_{ij})_{i \in [1,n], j \in [1,n]} = AA^T \tag{3.5}$$

In [45], it is shown that the minimization of the objective function J in Equation(3.4) is equivalent to

$$\max_{B^T = B, B \in \mathbb{R}_+^{K \times K}} tr(BB) \tag{3.6}$$

where tr(BB) represents the trace of BB. The trace of a matrix is defined to be the sum of the elements on the main diagonal.

3.2.1 K-means Clustering

In the K-means clustering, the objective function L using Euclidean distance is used to minimize within-cluster dissimilarities, given as

$$\min_{C_k} L = \sum_{k=1}^{K} \sum_{A_i \in C_k} \|(A_i - c_k)^T\|_F^2 \tag{3.7}$$

In [15], it is shown that the minimization of Equation(3.7) is equivalent to the maximization.

$$\max_{D^T D = I, D \in \Re^{n \times K}} L(D) = tr(D^T S D) \tag{3.8}$$

In order to understand this equivalence, the proof in [15] is presented here:

$$\begin{aligned} L &= \sum_{k=1}^{K} \sum_{A_i \in C_k} \|(A_i - c_k)^T\|_F^2 \\ &= \sum_{k=1}^{K} \sum_{A_i \in C_k} [(A_i - c_k)(A_i - c_k)^T] \\ &= \sum_{k=1}^{K} \sum_{A_i \in C_k} A_i A_i^T - 2 \sum_{k=1}^{K} \sum_{A_i \in C_k} A_i c_k^T + \sum_{k=1}^{K} \sum_{A_i \in C_k} c_k c_k^T \end{aligned} \tag{3.9}$$

For the k^{th} cluster, the sum of all the members in C_k can be represented in terms of the k^{th} row of $D^T A$ as

$$\sum_{A_i \in C_k} A_i = \sqrt{|C_k|}(D^T A)_k = \sqrt{|C_k|}(\widetilde{C})_k \tag{3.10}$$

We simplify the three terms in Equation(3.9) as follows:

$$\sum_{k=1}^{K} \sum_{A_i \in c_k} A_i A_i^T = \|A\|_F^2 = tr(AA^T) \tag{3.11}$$

$$\begin{aligned} \sum_{k=1}^{K} \sum_{A_i \in c_k} A_i c_k^T &= \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{A_i \in c_k} (A_i \sum_{A_i \in c_k} A_i^T) \\ &= \sum_{k=1}^{K} (\frac{1}{|C_k|} \sum_{A_i \in c_k} A_i \sum_{A_i \in c_k} A_i^T) \end{aligned} \tag{3.12}$$

$$\begin{aligned} \sum_{k=1}^{K} \sum_{A_i \in c_k} c_k c_k^T &= \sum_{k=1}^{K} |C_k| c_k c_k^T \\ &= \sum_{k=1}^{K} (\frac{1}{|C_k|} \sum_{A_i \in c_k} A_i \sum_{A_i \in c_k} A_i^T) \end{aligned} \tag{3.13}$$

Substituting Equation(3.10) into Equation (3.12) and Equation(3.13) the second and third terms of L in Equation(3.9) become

$$\begin{aligned} -\sum_{k=1}^{K}(D^TA)_k(D^TA)_k^T &= -tr((D^TA)(D^TA)^T) \\ &= -tr(D^TAA^TD) \end{aligned} \tag{3.14}$$

Now L in Equation(3.9) becomes

$$\begin{aligned} L &= -tr(AA^T) - tr(D^TAA^TD) \\ &= tr(S) - tr(D^TSD) \end{aligned} \tag{3.15}$$

Since $tr(S)$ is a constant, the $minL$ becomes $maxL(D) = tr(D^TSD)$

3.2.2 Example of Clustering with NMF

When NMF is used for clustering, the elements in H represent the clusters to which subjects belong. Work done on application of NMF for clustering advocates that each of the columns of H represents each cluster, where as the matrix W represents the cluster centroid. Let us consider an example of the data matrix A, which has a rank 4 is factorized into two matrices of rank 3.

$$A = \begin{bmatrix} 0.81 & 0.63 & 0.95 & 0.95 \\ 0.90 & 0.09 & 0.96 & 0.48 \\ 0.12 & 0.27 & 0.15 & 0.80 \\ 0.91 & 0.54 & 0.97 & 0.14 \end{bmatrix} \begin{array}{|c|} \hline \text{Item 1} \\ \hline \text{Item 2} \\ \hline \text{Item 3} \\ \hline \text{Item 4} \\ \hline \end{array}$$

The factorization of A produces H and W. $C1$, $C2$, and $C3$ are the cluster indicator for clusters 1, 2, and 3 repectively. Each row of W represents the cluster centroid for three clusters. From the factorization, item 1 is in cluster 3 as H_{13} has the largest magnitude among all the elements of H_1.

$$H = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \text{C1} \quad \text{C2} \quad \text{C3} \\ \begin{bmatrix} 0.66 & 0.45 & 0.76 \\ 0.00 & 1.22 & 0.24 \\ 0.63 & 0.15 & 0.00 \\ 0.07 & 0.12 & 1.17 \end{bmatrix} \end{array} \qquad W = \begin{bmatrix} 0.03 & 0.44 & 0.11 & 1.16 \\ 0.59 & 0.00 & 0.63 & 0.39 \\ 0.70 & 0.43 & 0.75 & 0.00 \end{bmatrix}$$

3.3 Data Pattern Hiding

We define Data Pattern Hiding as *the process of changing the data with the aim of hiding the confidential data pattern while minimizing the alteration to the non-confidential data pattern.* This chapter mainly focuses on the problem of confidentiality in terms of clustering. We want the information about the cluster membership of some particular data not to be disclosed.

As said earlier, NMF generates two matrices H and W for a nonnegative data matrix A, which are nonnegative factor matrices generated by minimizing the objective functions. The matrix W represents coefficients for clusters and has size of $k \times m$ which defines basis vectors. The matrix H has size of $n \times k$, and contains cluster membership indicators representing additive combination for each subject. To apply this idea to data pattern hiding, we can find out cluster membership of data by finding the largest element in the factor vector from H, provided factor vectors are related to the cluster property of the subjects [65]. The shift of a subject from one cluster to another cluster occurs whenever the factors are modified. This is the essence on which data pattern hiding is based on.

Let us say, we have n items in total with k clusters, we want to change the cluster membership of an item X which was originally in cluster C_i. In such a case, there are two ways in which we change the membership:

- Change the membership of item X to a particular cluster C_j, such that $i \neq j$.
- Change the membership of item X to any cluster other than cluster C_i.

We discuss about how to explicitly specify that information into the NMF in later section. One important aspect Wang et al. [65] mentioned in their work is the issue of side effect, which is discussed in the following section.

3.3.1 Side Effect

Side effects are the unwanted changes that are introduced after applying the constrained nonnegative matrix factorization. In our case, it is the cluster membership of the data. As it is directly related to the utility of the data, it is necessary to keep the changes in cluster membership of non-confidential data to a minimum. Any perturbation technique should have the property to keep the side-effect to the minimum level in order for the modified data to be useful. Ideally, all the confidential data are changed and nothing else is altered. In our method, we strive to achieve this goal.

There must be some measure of side effect and for that we propose to compare the k-means result on distorted data against the k-means result that we run on the original data. Hence, the number of subjects whose cluster membership are changed unexpectedly by the application of the method can be taken as the measure of side effect.

3.4 Constraint on Nonnegative Matrix Factorization

NMF is an unsupervised learning algorithm that has shown its applicability in various fields. It has been found that adding knowledge explicitly to the algorithm can produce significant improvement in the learning accuracy; this additional knowledge is commonly known as constraints, making the NMF algorithm semi-supervised. Researchers have come up with different constraints to be incorporated in NMF for solving different tasks. Some works are based on orthogonality [39] and sparseness [27].

Wang et al. [65] used three different techniques to change the position of the elements in the matrix H with the aim of changing the cluster to which the element lies.

$$\tilde{H} = \begin{bmatrix} 0.66 & 0.76 & 0.45 \\ 0.00 & 1.22 & 0.24 \\ 0.63 & 0.15 & 0.00 \\ 0.07 & 0.12 & 1.17 \end{bmatrix} \qquad W = \begin{bmatrix} 0.03 & 0.44 & 0.11 & 1.16 \\ 0.59 & 0.00 & 0.63 & 0.39 \\ 0.70 & 0.43 & 0.75 & 0.00 \end{bmatrix}$$

Using Wang's methods on the earlier factorization would result in the new matrix $\tilde{H}$ where the two element in the first column are swapped. This operation results in new change in membership for the element.

However the methods suggested are manual methods, where we select the element with the largest magnitude and try to substitute in place of some other element. We would like to add a constraint which we called the *clustering constraint* that results in the matrix H that will either have one of the elements significantly large compared to others which represents the new cluster for the item or one of the elements insignificant in terms of magnitude so as to make sure that the item does not fall in that cluster. We would have to explicitely define the clusters where we want our items to lie. We design a matrix C of size $n \times k$, and the elements of C are such that;

- If the item is not to be changed then, its contents will be 1 on the index representing its cluster and the rest of them are 0.
- If the item is to be changed to another particular cluster, then contents will be 1 on the index representing destination cluster and the rest of them are 0, which we refer to as *in a cluster change*.
- If the item is to be changed to any other cluster, then contents will be 0 on the index representing source cluster and the rest of them are some random number

in the range [0-1], referred to as *not in a cluster change*.

A typical example of it is

C1	C2	C3	
0	0	1	Item in Cluster 3
1	0	0	Item in Cluster 1
0.45	0.55	0	Item in any cluster other than Cluster 3

This is what our H should look like at the end of the factorization process. One way to explicitely define that is by incorporating it into the update rule as follows.

$$f(A, H, W) = \alpha \|A - HW\|_F^2 + \beta \|H - C\|_F^2 \tag{3.16}$$

We call the new term the *penalty term*. α and β assign weights to the conventional factorization and the clustering constraint. When β=0, the factorization is the conventional NMF, $\alpha = 0$ results in factorization where the data are factorizated towards the cluster centroid. We can see the effect of varying values of α and β from the Figures 4.8 to 4.17 in Chapter 4.

3.4.1 Update Formula

The gradient of the functions $f(A, H, W)$ consists of two parts:

$$\frac{\partial f(A,H,W)}{\partial H} \text{ and } \frac{\partial f(A,H,W)}{\partial W}$$

which are respectively partial derivatives to elements in H and W. From the Karush-Kush-Turcker (KKT) optimality condition, (W, H) is a stationary point if and only if

$$H_{ia} \geq 0 \text{ and } W_{bj} \geq 0$$

$$\frac{\partial f(A,H,W)_{ia}}{\partial H} = 0 \text{ and } \frac{\partial f(A,H,W)_{bj}}{\partial W} = 0$$

Optimization methods for NMF produce a sequence $\{H^k, W^k\}_{k=1}^{\infty}$ of iterations.

Mathematical derivation for update formula

Let,

$$\begin{aligned} Q &= \|A - HW\|_F^2 \\ &= tr((A - HW)^T(A - HW)) \\ &= tr(A^TA - A^THW - W^TH^TA + W^TH^THW) \\ &= tr(A^TA) - 2tr(A^THW) + tr(W^TH^THW) \end{aligned} \tag{3.17}$$

also let,

$$\begin{aligned} L &= \|H - C\|_F^2 \\ &= tr((H - C)^T(H - C)) \\ &= tr(H^TH - H^TC - C^TH + C^TC) \\ &= tr(H^TH - 2H^TC + C^TC) \end{aligned} \tag{3.18}$$

- H fixed and W changing,

$$\frac{\delta f(A, H, W)}{\delta W}$$

$$\begin{aligned} &= \frac{\delta(\alpha\|A - HW\|_F^2 - \beta\|H - C\|_F^2)}{\delta W} \\ &= \alpha\frac{\delta(Q)}{\delta W} - \beta\frac{\delta(\|H - C\|_F^2)}{\delta W} \\ &= -2\alpha\frac{\delta(tr((A^THW)))}{\delta W} + \alpha\frac{\delta(tr((W^TH^THW)))}{\delta W} \\ &= -2\alpha H^TA + 2\alpha H^THW \end{aligned} \tag{3.19}$$

- W fixed and H changing

$$\frac{\delta f(A, H, W)}{\delta H}$$

$$
\begin{aligned}
&= \frac{\delta(\alpha\|A - HW\|_F^2 - \beta\|H - C\|_F^2)}{\delta H} \\
&= \alpha\frac{\delta(Q)}{\delta H} - \beta\frac{\delta(\|H - C\|_F^2)}{\delta H}
\end{aligned} \tag{3.20}
$$

We know, the first term gives,

$$
\alpha\frac{\delta Q}{\delta H} = -2\alpha AW^T + 2\alpha HWW^T \tag{3.21}
$$

The second term gives,

$$
\begin{aligned}
\beta\frac{\delta\|H - C\|_F^2}{\delta H} &= \beta 2H - 2C + 0 \\
&= 2\beta H - 2\beta C
\end{aligned} \tag{3.22}
$$

Combining (3.21) and (3.22) in (3.20),

$$
\begin{aligned}
\tfrac{\delta f(A,H,W)}{\delta H} &= -2\alpha AW^T + 2\alpha HWW^T + 2\beta H - 2\beta C \\
&= 2\alpha HWW^T + 2\beta H - 2\alpha AW^T - 2\beta C
\end{aligned} \tag{3.23}
$$

For optimal solution $\frac{\delta f(A,H,W)}{\delta W}$=0 and $\frac{\delta f(A,H,W)}{\delta H}$=0. Hence,

$$H^T A \oslash H^T HW = I$$

$$H(\alpha WW^T + \beta) \oslash (\alpha AW^T + \beta C) = I$$

where, $\oslash$ represents element-wise division, I denotes identity matrix. This gives rise to the update formulas for W and H as

$$W_{i,j} = W_{i,j}\frac{[H^T A]_{i,j}}{[H^T HW]_{i,j}} \tag{3.24}$$

$$H_{i,j} = H_{i,j}\frac{[\alpha AW^T + \beta C]_{ij}}{[H(\alpha WW^T + \beta)]_{ij}} \tag{3.25}$$

3.4.2 Objective Function

As mentioned earlier, the objective function needs to be changed to incorporate the constraint. Let us start with our initial formula:

$$f(A, H, W) = \alpha \|A - HW\|_F^2 + \beta \|H - C\|_F^2 \tag{3.26}$$

The Objective here is to not only make $\|A - HW\|_F^2$ smaller but to make the sum of both terms in the above equation small, which gives rise to,

$$\min_{H \geq 0, W \geq 0} (\alpha \|A - HW\|_F^2 + \beta \|H - C\|_F^2) \tag{3.27}$$

This is the objective function that will be used to check the convergence. If the value is below a certain threshold the NMF process is considered to have converged.

3.5 Convergence of the method

Convergence of the method is discussed in the next chapter (Section 4.6).

3.6 Algorithm

In this section, we present the Constrained NMF algorithm for the data pattern hiding. The algorithm is as in Algorithm 2:

Original data matrix A, k, C, tol, $maxIter$, $mainIter$, α, β are the input to the algorithm. The tol provides the stopping criterion, in other words measurement of convergence, $maxIter$ limits the number of updates to perform in H and W before stopping an NMF if convergence is not achieved. The output from the algorithm is the two matrices H and W, such that $\tilde{A} = H \times W \approx A$, where all the confidential data are hidden. The constrained NMF algorithm is run for a certain number of iterations and checked each time if the desired pattern hiding is achieved. If there are any side effects the algorithm continues to perform NMF other wise it stops. It does not show how the side-effect is calculated in the algorithm above. It does so by

Algorithm 2: Constrained NMF

input : $A \in \mathbb{R}_{+}^{n \times m}, 0 < k \ll min(n, m), C \in \mathbb{R}_{+}^{n \times k}, mainItr, tol, maxItr, \alpha, \beta$
output: $H \in \mathbb{R}_{+}^{n \times k}, W \in \mathbb{R}_{+}^{k \times m}$
Initialize H and W with the random initial estimates
$H_{i,j}^{(0)} \Leftarrow nonnegativevalue, 1 \leq i \leq n, 1 \leq j \leq k$
$W_{i,j}^{(0)} \Leftarrow nonnegativevalue, 1 \leq i \leq k, 1 \leq j \leq m$
for $i \leftarrow 1$ **to** $mainItr$ **do**
  for $j \leftarrow 1$ **to** $maxItr$ **do**
    $H_{i,j} \leftarrow H_{i,j} \frac{[\alpha A W^T + \beta C]_{ij}}{[H(\alpha W W^T + \beta)]_{ij}}$
    $W_{i,j} \leftarrow W_{i,j} \frac{[H^T A]_{i,j}}{[H^T H W]_{i,j}}$
    Calculate new $\hat{A}$
    if *value(Objective Function)*$\leq$ *tol* **then**
      break
  if *sideeffect=0* **then**
    break
  Change value of α
  Change value of β

comparing the k-means result on the modified data with the k-means result on the original data for the non-confidential data and comparing against what we wanted in the beginning for the confidential data.

3.7 Complexity

The computational complexity of CNMF can be broken down into two parts: k-means phase and the NMF phase. One simple rule of thumb to set the number of clusters for any dataset is $k \approx \sqrt{n/2}$ with n as the number of objects (data points) [48]. Regarding computational complexity, finding the optimal solution to the k-means clustering problem for observations in m dimensions is NP-hard. However, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. If k and m are fixed, the problem can be solved in time $O(n^{mk+1} \log n)$, where n is the number of entities to be clustered [28].

Let r be the number of iterations then computational complexity of multiplicative

NMF is given by [41] as $r * O(nmk)$. Hence, the computational complexity of our method will be

$$Time\ Complexity = O(rnmk) + O(n^{mk+1} \log n).$$

3.8 Experimental Results

The following sections present our experimental results with different datasets.

3.8.1 Experiment 1

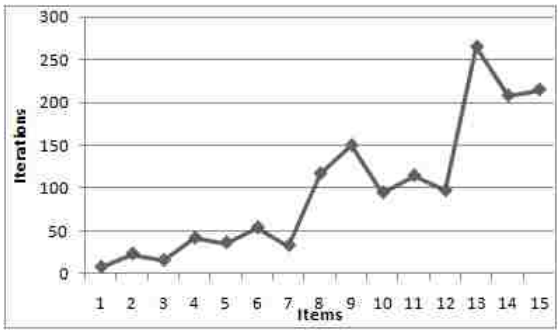


Figure 3.1: Not in a cluster (IRIS)

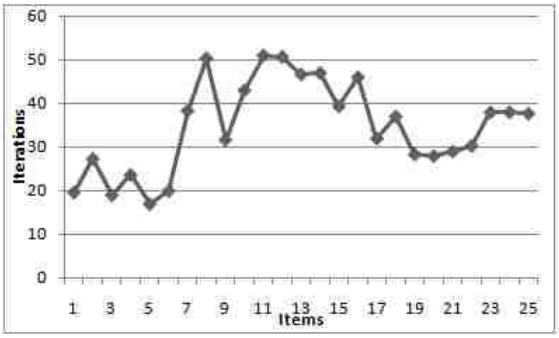


Figure 3.2: In a cluster (IRIS)

Two types of changes were made, the first was *in a cluster change* with graphs as shown in Figures 3.2 and 3.4 while the second was *not in a cluster change* shown in Figures 3.1 and 3.3 . We kept the α (0.5) and β (0.5) equal for this experiment. Experiment was done to observe the number of iterations it took to make all of the changes. The algorithm was unable to get convergence for more than 15 items for *not in a cluster change* while the number goes beyond 25 for *in a cluster change* in the case of IRIS data. Similar observation was made for the YEAST data. It can be concluded from the experiment that it takes a lot fewer iterations to make the *in a cluster change* compared to not *in a cluster change*. It can be attributed to the fact that one element in a row of the matrix H needs to be significantly large compared to the others for it to work efficiently.

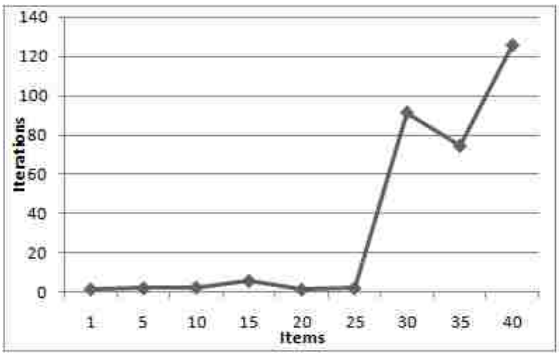


Figure 3.3: Not in a cluster (YEAST)

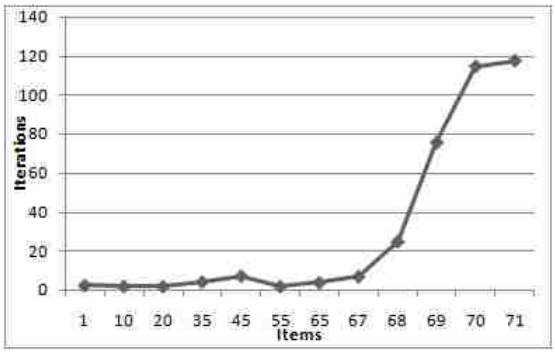


Figure 3.4: In a cluster (YEAST)

3.8.2 Experiment 2

Table 3.1: Classes for α and β

Class	α	β
1	0.1	0.9
2	0.2	0.8
3	0.3	0.7
4	0.4	0.6
5	0.5	0.5
6	0.6	0.4
7	0.7	0.3
8	0.8	0.2
9	0.9	0.1

Next experiment was to study the relation between the values of α and β with the number of confidential subjects in order to achieve convergence. We performed this experiment both with the IRIS and the YEAST data. For each dataset, we ran experiment for small p (total number of changes) and larger p. One thing to remember is that the number of *in a cluster changes* were equal to the number of *not in a cluster changes*.

Initially, $\alpha = 0.9$ and $\beta = 0.1$ and then the value of α was decreased by 0.1 and value of β was increased by 0.1, the aim is to keep $(\alpha + \beta) = 1$, so that our estimated solution does not diverge from the actual solution. Experiment was repeated again, but this time with the initial value of $\alpha = 0.1$ and $\beta = 0.9$ and increase the value

of α by 0.1 and decrease β by 0.1. Each of the experiment was performed 100 times and to see what region in terms of values of α and β gives the most convergence.

We can see from Figure 3.5 that, when p is small, we get most convergence in that class of α and β combination where we start the iteration from, but as we increased p to 26 as in Figure 3.6, we can see that most convergence occurs in the region where $\beta > \alpha$.

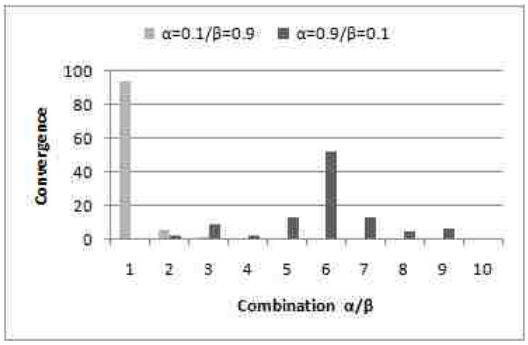


Figure 3.5: IRIS data with p=10

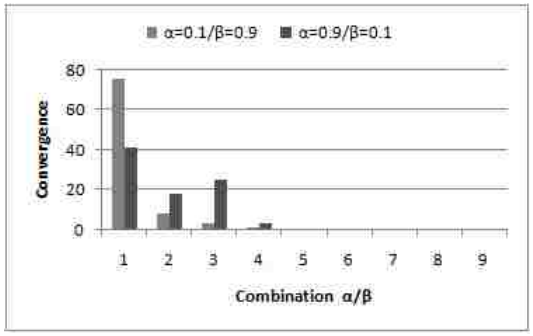


Figure 3.6: IRIS data with p=26

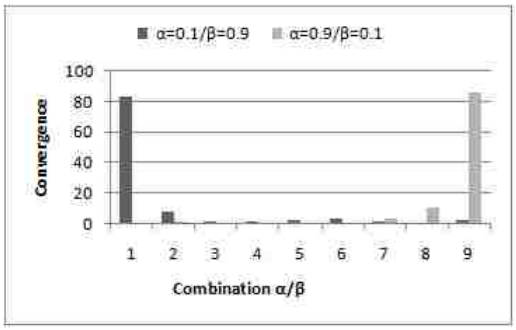


Figure 3.7: YEAST data with p=10

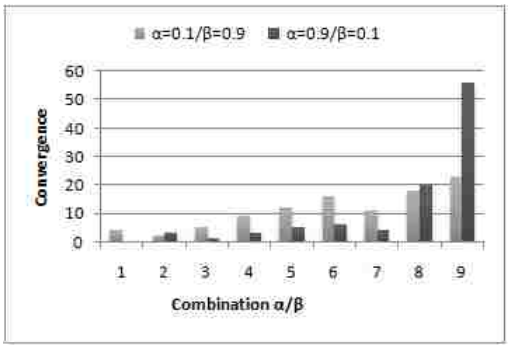


Figure 3.8: YEAST data with p=38

Similar observation was made for the YEAST data from Figures 3.7 and 3.8. When p is small values of α and β do not play an important part in the convergence. When p grows large then the distribution shifts towards the region with smaller β and greater α, indicating that to change larger number of data the values of α and β should be in this particular range. The value of α and β basically depends upon the data. As in the case of IRIS data it was $\beta > \alpha$ but for YEAST data it was more $\beta < \alpha$ region.

3.8.3 Experiment 3

In the previous experiment we saw that, convergence depends on the value of α and β depending upon p. In this experiment, we tried to study the relation between the total number of changes (change in cluster membership of the data) and the given size of data. The following tables show that total number of changes that were made successfully with different amount of data.

Table 3.2: IRIS Changes and data size

Data Size	Changes
60	10
90	14
120	20
150	24

Table 3.3: YEAST Changes and data size

Data Size	Changes
150	50
180	70
240	90
300	110
360	140
420	140
450	150
480	150

It can be seen from the Tables 3.2 and 3.3 that the maximum number of patterns that could be hidden depends on the size of the dataset we are operating on. The higher the number of elements we seek to hide in the given dataset, the more difficult it gets.

3.9 Conclusion

The chapter proposed a novel method of DPH through the use of NMF. The proposed technique provides a way to explicitly integrate the clustering constraint into the objective function of NMF, which results in the factorization of the matrix such that data members cluster membership are as defined by the clustering matrix. The membership of a confidential data can be changed by defining different cluster membership through the clustering matrix. The second benefit of the proposed method is that it not only changes the membership of the confidential data but also changes

the values of the other members and its attributes which is always a desirable feature for PPDM. The third benefit comes from the fact that each factorization is unique depending up on the initialization of matrices H and W.

As the experiments have shown, NMF does have some issues like the limitation in the number of membership changes that could be done in a fixed dataset and the varying value of α and β that provides convergence. Hence, one needs to be aware of these issues to use NMF to the full potential. NMF is a computationally expensive process but improvement can be achieved by technique like better initialization of the matrices H and W.

Chapter 4 Constrained Nonnegative Matrix Factorization for Data Value Hiding

4.1 Introduction

Most privacy-preserving data mining methods apply a transformation which reduces the effectiveness of the underlying data when data mining methods or algorithms are applied. In fact, there is a natural tradeoff between privacy and accuracy, though this tradeoff is affected by the particular algorithm which is used for privacy-preservation. A key issue is to maintain maximum utility of the data without compromising the underlying privacy constraints.

Since the perturbed data may often be used for mining and management purposes, its utility needs to be preserved. Therefore, the data mining and privacy transformation techniques need to be designed effectively, so as to preserve the utility of the results.

In the earlier chapter, we introduced CNMF based data pattern hiding technique. The same technique can be used for data distortion. In this chapter we concentrate on data value hiding and study the utility of the distorted data. Work of [70, 65] has been on hiding exact data value using SVD and NMF based techniques. Based on some mathematical derivations, we propose a few novel data distortion strategies. The first technique is called the Constrained Nonnegative Matrix Factorization (CNMF) and the second one is Sparsified CNMF. We study the distortion level of each of these algorithms with the other matrix based techniques like SVD and NMF. K-means is used to study the data utility of the two proposed methods.

4.2 Motivation

Matrix H resulting from the NMF represents additive combination for each subject which is the indicators for the cluster membership, while matrix W represents coefficients for clusters. Wang et al. [65] tried to apply this idea for DPH, they find out cluster membership of data by finding the largest element in the factor vector from H, provided factor vectors are related to the cluster property of the subjects. In such cases, we can improve the result of clustering from the NMF algorithm, if we could somehow make one of the element of vectors in H significantly larger than the other elements. We need to introduce additional constraint that will be implicitly defined into the update rule to achieve this goal, that is where the idea of introducing constrained NMF for distortion comes from. This can be a very effective method for data distortion while still maintaining the data cluster property.

4.3 Constrained Nonnegative Matrix Factorization (CNMF)

This technique is similar to the one we discussed in the earlier section with the exception that the matrix C is entirely initialized based on the k-means result on the original data.

4.4 Sparsified Constrained Nonnegative Matrix Factorization (SCNMF)

The work in [22] performs sparsification on SVD while the work in [59] performs sparsification on NMF both with the aim of removing the noise as well as reducing the storage space. The whole objective of Constrained Nonnegative Matrix Factorization is to have the factor vectors with one of the elements significantly large and other elements with insignificant magnitude. Ideally, one of the elements should be 1 and the rest of them will be zero but in practice one element is large in magnitude and the rest of them are small numbers. In this method, we plan to change the numbers that

are insignificant to 0. What is a significant value?, it can be a whole new research area to determine the magnitude of the significant number. For our research purpose, we fix the value of what we consider a significant number and try to see the output.

$$h_{i,j} = \begin{cases} h_{i,j} & \text{if } |h_{i,j}| \geq \delta \\ 0 & otherwise \end{cases}$$

δ is a threshold value against which we check the elements of matrix H.

4.5 Cluster-Aware Compression based Constrained Nonnegative Matrix Factorization

Cluster-Aware Compression based Constrained Nonnegative Matrix Factorization (ComNMF) is the third constraint based method that we propose in this chapter. Drawback of random perturbation is that the added noise distorts the distances between the data points which leads to poor clustering accuracy. The situation can be explained by Figure 4.1 and Figure 4.2, the figures demonstrate how the subjects tend to shift to different cluster as the noise is introduced. Higher level of distortion results in higher level of subject movement between the clusters.

We can proceed by adding noise in the direction of the cluster centroid, so that the items remain in their original cluster. We need a way to embed that information into the NMF algorithm itself. One extreme view is if all the points collapse to their

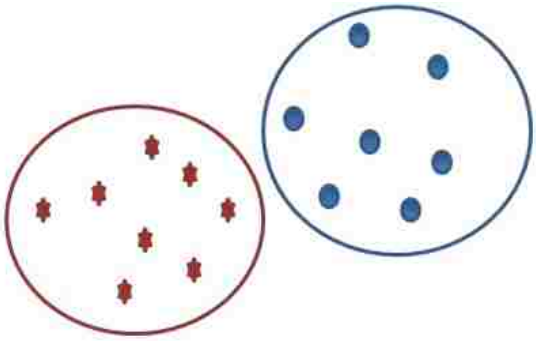


Figure 4.1: Original Clusters

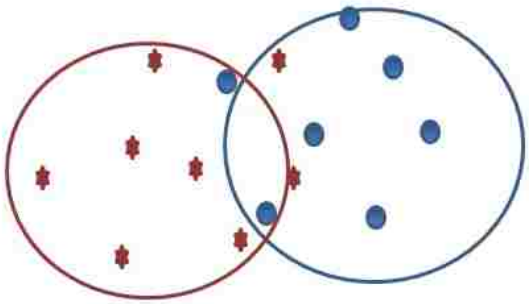
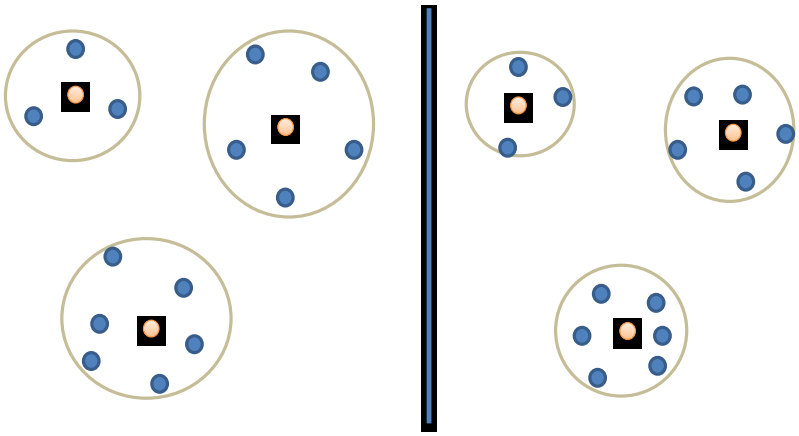


Figure 4.2: Clusters after perturbation



(a) Initial clusters (b) Transformed clusters

Figure 4.3: Compressed Clusters

corresponding cluster centroids, it would result in perfectly accurate cluster but the overall utility of the data is lost. So, we need a way to add noise in the direction of the centroid which reduces the distance between two points in the same cluster and between a given point and the centroid of the cluster it belongs to.

The Figure 4.3 shows the case where perturbation is performed towards the cluster centroid. The method is more useful for the cases where the distance between the cluster centroid is close to the sum of radius of two centroids. The situation can be explained from the Figure 4.4 where r is the radius of the first cluster, R is the radius of the second cluster, while D is the shortest distance between the two clusters. As

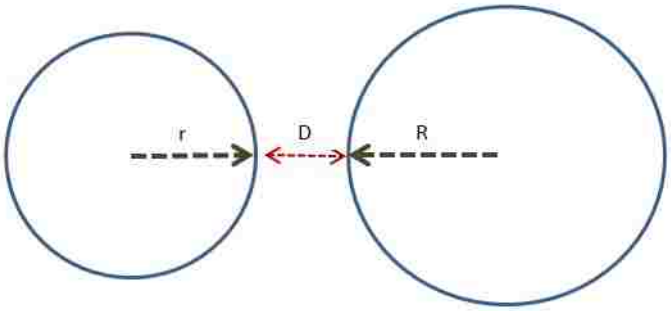


Figure 4.4: Distance between clusters

the D decreases, even a slight distortion is enough to make change in the cluster membership.

Objective Function

Let the dataset A has k cluster centroids represented as $c_1, c_2, ..., c_k$. Similar to the previous chapter, we have two objectives: traditional factorization and compression towards center. Compression towards center can be achieved by having subjects that tend towards the cluster centroids. We define the second part of the objective by defining a matrix C_L which is the matrix A where each subject is replaced by their respective cluster centroid. Let $centroid(i)$ gives centriod of the cluster to in which the item i belongs. Then C_L can be represented as;

$$\begin{bmatrix} centroid(A_1) \\ centroid(A_2) \\ . \\ centroid(A_n) \end{bmatrix}$$

Then the objective function can be written as

$$f(A, H, W) = \alpha \|A - HW\|_F^2 + \beta \|HW - C_L\|_F^2 \tag{4.1}$$

$\|HW - C_L\|_F^2$ represents the compression contrainst in the above equation. The function not only makes $\|A - HW\|_F^2$ smaller but also makes the sum of both terms in the above equation small, which gives rise to

$$\min_{H \geq 0, W \geq 0} (\alpha \|A - HW\|_F^2 + \beta \|HW - C_L\|_F^2) \tag{4.2}$$

This is the objective function that will be used to check the convergence. The value below certain threshold indicates the convergence of the method.

Update Formula

Mathematical derivation for update formula if provided in the following section. Let,

$$\begin{aligned} Q &= \|A - HW\|_F^2 \\ &= tr((A - HW)^T(A - HW)) \\ &= tr(A^TA - A^THW - W^TH^TA + W^TH^THW) \\ &= tr(A^TA) - 2tr(A^THW) + tr(W^TH^THW) \end{aligned} \tag{4.3}$$

also let,

$$\begin{aligned} L &= \|HW - C_L\|_F^2 \\ &= tr((HW - C_L)^T(HW - C_L)) \\ &= tr(W^TH^TH - W^TH^TC_L - C_L^THW + C_L^TC_L) \\ &= tr(H^TH - 2W^TH^TC_L + C_L^TC_L) \end{aligned} \tag{4.4}$$

- H fixed and W changing,

$$\frac{\delta f(A,H,W)}{\delta W}$$

$$\begin{aligned} &= \frac{\delta(\alpha\|A - HW\|_F^2 - \beta\|HW - C_L\|_F^2)}{\delta W} \\ &= \alpha\frac{\delta(Q)}{\delta W} - \beta\frac{\delta(\|HW - C_L\|_F^2)}{\delta W} \\ &= -2\alpha\frac{\delta(tr((A^THW)))}{\delta W} + \alpha\frac{\delta(tr((W^TH^THW)))}{\delta W} \\ &= -2\alpha H^TA + 2\alpha H^THW \end{aligned} \tag{4.5}$$

We know, the first term gives,

$$\alpha\frac{\delta Q}{\delta W} = -2\alpha H^TA + 2\alpha H^THW \tag{4.6}$$

The second term gives,

$$\begin{aligned} \beta\frac{\delta\|HW - C_L\|_F^2}{\delta W} &= 2\beta(H^THW - 2H^TC_L) \\ &= 2\beta H^THW - 2\beta H^TC_L \end{aligned} \tag{4.7}$$

Combining (4.10))] and (4.11)) in (4.9),

$$\begin{aligned} \tfrac{\delta f(A,H,W)}{\delta H} = & -2\alpha H^TA + 2\alpha H^THW + 2\beta H^THW - 2\beta H^TC_L \\ = & \quad 2\alpha H^THW + 2\beta H^THW - 2\alpha H^TA - 2\beta H^TC_L \end{aligned} \tag{4.8}$$

- W fixed and H changing

$$\frac{\delta f(A,H,W)}{\delta H}$$

$$\begin{aligned} &= \frac{\delta(\alpha\|A - HW\|_F^2 - \beta\|HW - C_L\|_F^2)}{\delta H} \\ &= \alpha\frac{\delta(Q)}{\delta H} - \beta\frac{\delta(\|HW - C_L\|_F^2)}{\delta H} \end{aligned} \tag{4.9}$$

We know, the first term gives

$$\alpha\frac{\delta Q}{\delta H} = -2\alpha AW^T + 2\alpha HWW^T \tag{4.10}$$

The second term gives

$$\begin{aligned} \beta\frac{\delta\|HW - C_L\|_F^2}{\delta H} &= 2\beta(HWW^T - 2C_LW^T) \\ &= 2\beta HWW^T - 2\beta C_LW^T \end{aligned} \tag{4.11}$$

Combining (4.10) and (4.11) in (4.9),

$$\begin{aligned} \tfrac{\delta f(A,H,W)}{\delta H} = & -2\alpha AW^T + 2\alpha HWW^T + 2\beta HWW^T - 2\beta C_LW^T \\ = & \quad 2\alpha HWW^T + 2\beta HWW^T - 2\alpha AW^T - 2\beta C_LW^T \end{aligned} \tag{4.12}$$

For optimal solution $\frac{\delta f(A,H,W)}{\delta W}$=0 and $\frac{\delta f(A,H,W)}{\delta H}$=0. Hence,

$$H^T HW(\alpha+\beta) \oslash (\alpha H^T A + \beta H^T C_L) = I$$

$$HWW^T(\alpha+\beta) \oslash (\alpha AW^T + \beta C_L W^T) = I$$

where, $\oslash$ represents element-wise division, I denotes identity matrix. This gives rise to the update formulas for W and H as,

$$W_{i,j} = W_{i,j} \frac{[\alpha H^T A + \beta H^T C_L]_{i,j}}{[H^T HW(\alpha+\beta)]_{i,j}}$$

$$H_{i,j} = H_{i,j} \frac{[\alpha AW^T + \beta C_L W^T]_{ij}}{[(\alpha+\beta)HWW^T]_{ij}}$$

Since α+β=1, it can be simplified to,

$$W_{i,j} = W_{i,j} \frac{[\alpha H^T A + \beta H^T C_L]_{i,j}}{[H^T HW]_{i,j}} \tag{4.13}$$

$$H_{i,j} = H_{i,j} \frac{[\alpha AW^T + \beta C_L W^T]_{ij}}{[HWW^T]_{ij}} \tag{4.14}$$

4.6 Convergence

The discussion in this section applies to both CNMF and ComNMF. We consider ComNMF for the discussion on convergence. The process is similar for CNMF.

4.6.1 Convergence of ComNMF

We start with the Equation (4.13) (update rule for W), given as

$$\begin{aligned} W_{i,j} &= W_{i,j} \frac{[\alpha H^T A + \beta H^T C_L]_{i,j}}{[H^T HW]_{i,j}} \\ &= W_{i,j} \frac{[H^T(\alpha A + \beta C_L)]_{i,j}}{[H^T HW]_{i,j}} \end{aligned} \tag{4.15}$$

where C_L is the cluster matrix composed of the centroids. Let a constant G be defined such that

$$G = \alpha A + \beta C_L \tag{4.16}$$

which results to

$$W_{i,j} = W_{i,j} \frac{[H^T G]_{i,j}}{[H^T H W]_{i,j}} \tag{4.17}$$

The above equation is the same form as the original multiplicative NMF update rules proposed by Lee and Seung [57]. So, the convergence of our constraints based NMF is as good as the multiplicative update itself. We can proceed similarly for the H matrix.

However, it has been pointed out in the literature [41, 25] that such multiplicative update properties do not guarantee the convergence to a stationary point. Gonzales and Zhang [25] numerically showed that multiplicative udpate may fail to converge to a stationary point. Lin [41] claimed that due to possible numerical inaccuracy, a mathematical example is desired before drawing conclusions. Thus the convergence issue remains open [40].

4.7 Algorithm

The algorithm consists of two phases; the first is the Constrained NMF phase which results in the modified version of the original data by imposing the clustering constraint. As can be seen, the update rule and the objective functions are different from the conventional NMF algorithm. The second phase is basically the sparsification of the modified data. Threshold value for the sparsification process is passed as the parameter to the algorithm. Algorithm 3, shows the details of the algorithm.

Algorithm 3: Sparsified and Constrained NMF

input : $A \in \mathbb{R}_+^{n \times m}, 0 < k \ll min(n,m), C \in \mathbb{R}_+^{n \times k}, tol, maxItr, \alpha, \beta, stol$
output: $H \in \mathbb{R}_+^{n \times k}, W \in \mathbb{R}_+^{k \times m}$
Initialize H and W with the random initial estimates
$H_{i,j}^{(0)} \Leftarrow nonnegativevalue 1 \leq i \leq n, 1 \leq j \leq k$
$W_{i,j}^{(0)} \Leftarrow nonnegativevalue 1 \leq i \leq k, 1 \leq j \leq m$
$\%--------------------$
$\%constrained\ NMF\ process$
for $p \leftarrow 1$ **to** $maxItr$ **do**
  $H_{i,j} \leftarrow H_{i,j} \frac{[\alpha AW^T + \beta C]_{ij}}{[H(\alpha WW^T + \beta)]_{ij}}$
  $W_{i,j} \leftarrow W_{i,j} \frac{[H^T A]_{i,j}}{[H^T HW]_{i,j}}$
  Calculate new $\tilde{A}$
  if *value(Objective Function)*$\leq$ *tol* **then**
    break
$\%--------------------$
$\%Sparsification\ Process$
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** k **do**
    if $|H_{i,j}| \leq stol$ **then**
      $H_{i,j} = 0$

4.8 Complexity

The computational complexities of CNMF, ComNMF methods remain the same as discussed in Section 3.7. With regard to SNMF, there is an extra step of sparsification in addition to regular CNMF process. As sparsification is performed for all the items in the H matrix, the complexity is $O(mn)$.

$$\begin{aligned} TimeComplexity &= O(rnmk) + O(n^{mk+1} \log n) + O(mn) \\ &= O(rnmk) + O(n^{mk+1} \log n) \end{aligned} \tag{4.18}$$

where r is the number of iterations for the factorization.

4.9 Experiments

We conduct our experiments on two data sets namely Connectionist Bench Dataset and Yeast Dataset. We show the effectiveness of our method by comparing with following standard distortion techniques;

- Noise Additive perturbation

  In our study we observe the characteristics of two additive perturbation methods: Random Noise (RD) and Normal Noise (ND). Work on additive noise was first publicized by Kim [32] with the general expression that

  $$Z = X + \epsilon \tag{4.19}$$

  Where Z is the transformed data point, X is the original data point and ϵ is the noise.

- Singular Value Decomposition perturbation (SVD)

  Wang et al. in [67] proposed use of Truncated SVD (SVD) method for perturbation of data. The first step is to create a rank-k approximation A_k to the matrix A by defining

  $$A_k = U_k \sum_k V_k^T$$

  where U_k contains the first columns of U , $\sum_k$ contains the k largest nonzero singular values of A , and V_k^T contains the first k rows of V^T. It has been proven that the distance between A and its rank-k approximation is minimized by the approximation A_k in the sense of the Frobenius norm.

  Wang defined $E_k = A - A_k$ as the noise in the original matrix A. Hence, using A_k instead of A may yield better mining accuracy. The distorted data A_k can preserve privacy because of the difference between A and A_k, as it is difficult to figure out the values of A from those of A_k without the knowledge of E_k.

Table 4.1: Yeast Dataset

Distortion	Methods	VD	RP	RM	Accuracy (%)
Level 1	RD	0.140404	254.925303	0.005215	76.177658
	ND	0.128468	166.087315	0.016487	90.24226
	SVD	0.146576	287.686575	0.004038	64.872140
	NMF	0.148205	291.939435	0.002608	60.632571
	CNMF	0.138215	345.782301	0.002608	99.865410
	SCNMF(δ=0.03)	0.138496	347.839166	0.002187	99.865410
	ComNMF	0.140179	315.110027	0.006393	100.00
Level 2	RD	0.170296	266.961642	0.004206	72.274563
	ND	0.164019	172.601615	0.013627	87.146703
	SVD	0.182865	342.139637	0.002187	45.423957
	NMF	0.177955	342.825370	0.002019	52.960969
	CNMF	0.191650	352.731326	0.001851	88.963661
	SCNMF(δ=0.03)	0.191680	352.373822	0.001682	89.703903
	ComNMF	0.180322	352.660498	0.001935	90.57

Hence, A_k can be seen as a distorted copy of A and a faithful representation of the original data.

- NMF based distortion

  Wang et al. in [66] proposed the use of NMF for data distortion. NMF provides compact representation with reduced-rank while preserving dominant data patterns. Wang proposed to use rank reduced matrix as distorted data.

4.9.1 Experiment 1

In the first experiment, we tried to observe the difference in terms of accuracy between different algorithms. We kept the VD same and analyzed the the utility of the data. As mentioned earlier, we used k-means to validate the results. The results as demonstrated by the experiments clearly show the Constrained NMF and its sparsified version clearly have an edge over the other methods. From Table 4.1, we can clearly see that the CNMF based algorithm performs far better than other methods if we look at the VD, RM and accuracy.

Table 4.2: Connectionist Bench Dataset

Distortion	Methods	VD	RP	RM	Accuracy (%)
Level 1	RD	0.509603	24.637981	0.053205	95.673077
	ND	0.500168	24.524840	0.054487	96.153846
	SVD	0.333235	51.879006	0.008013	98.076923
	NMF	0.333409	51.900962	0.007131	97.596154
	CNMF	0.347916	53.728205	0.006330	97.596154
	SCNMF(δ=0.03)	0.347916	53.728205	0.006330	97.596154
	ComNMF	0.355190	52.416667	0.007853	100.00
Level 2	RD	80.446735	39.030288	0.014423	47.115385
	ND	65.882012	41.075000	0.013301	78.365385

The Table 4.2 provides some extra insight. We can see that distortion level in terms of VD for RD and ND are higher than for the factorization based techniques; the accuracy does reflect the distortion level. If we consider level 2, then we were not able to distort the data to match the same level of distortion as RD and ND.

4.9.2 Experiment 2

In the second experiment, we examined how the accuracy of CNMF changes as we change the convergence threshold for the NMF algorithm. We calculate the convergence using Equation(3.1). Figure 4.5 is the graph plotted between the threshold level set and the clustering accuracy. This is however the result observed for NMF without sparsification. As can be seen from the Figure 4.5, it is required to get as

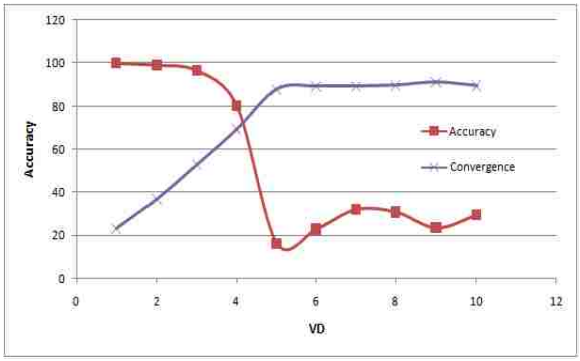


Figure 4.5: Accuracy Vs Convergence (Yeast)

Table 4.3: Accuracy with different sparsification threshold value

No Sparsification		$\delta = 0.03$		$\delta = 0.06$	
VD	Accuracy (%)	VD	Accuracy (%)	VD	Accuracy (%)
0.137521	100.00	0.137521	100.00	0.137521	100.00
0.138215	99.86541	0.138496	99.86541	0.138851	99.86541
0.14698	96.096904	0.146691	96.366083	0.147567	96.702557
0.183637	76.648721	0.183433	76.9179	0.183431	77.254374
0.19165	88.963661	0.19168	89.703903	0.19214	91.453567
0.242631	30.215343	0.242752	29.946164	0.243098	32.368775

close as possible to the convergence region for the clustering result to be accurate. As the threshold becomes much larger than the convergence value the accuracy of the method drops drastically. Thus it is important that we have the convergence to gain higher utility level.

4.9.3 Experiment 3

Our third experiment was to test the change in accuracy as a result of change in threshold value for sparsification.

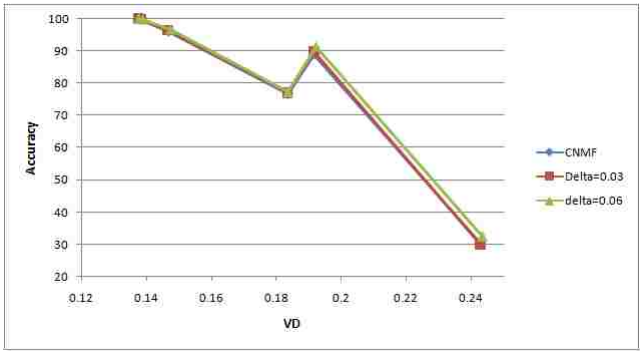


Figure 4.6: Change in Accuracy due to sparsification (Yeast)

From the Table 4.3 and the Figure 4.6, there is slight increase in accuracy of the clustering if we use the sparsification, for convenience we showed $\delta = 0.03$ and $\delta = 0.06$ only. Right now, we do not have a way to determine which value of δ will perform better, it is something that has to be determined empirically. Plot for

CNMF and δ=0.03 are highly overlapping, making it difficult to observe the plot in Figure 4.6.

4.9.4 Experiment 4

Our fourth experiment was to observe how the compression based NMF performs on the real dataset. We show our result with IRIS dataset as it has fewer number of attributes resulting in easier visualization.

The Figure 4.7 shows the difference in result we get from compressed NMF against the regular NMF. Each of the clusters is more dense in case of compressed NMF compared to conventional NMF.

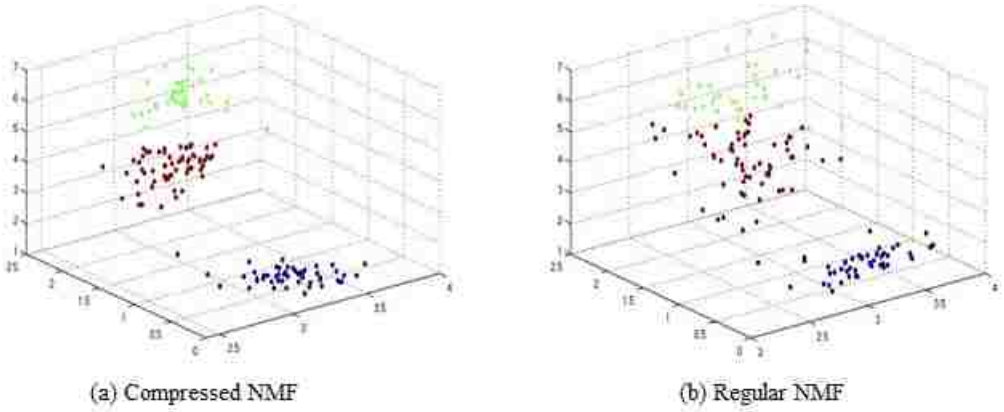


Figure 4.7: Compressed NMF Vs Regular NMF

We extend this experiment to find out if the cluster radius can be controlled implicitely through NMF. As can be seen, by varying the values of α and β we can control the radius of the cluster. As β increased from 0.1 to 1, the cluster almost shrunk to a point. This can be a useful feature in cases where we lack clear separation between two different clusters.

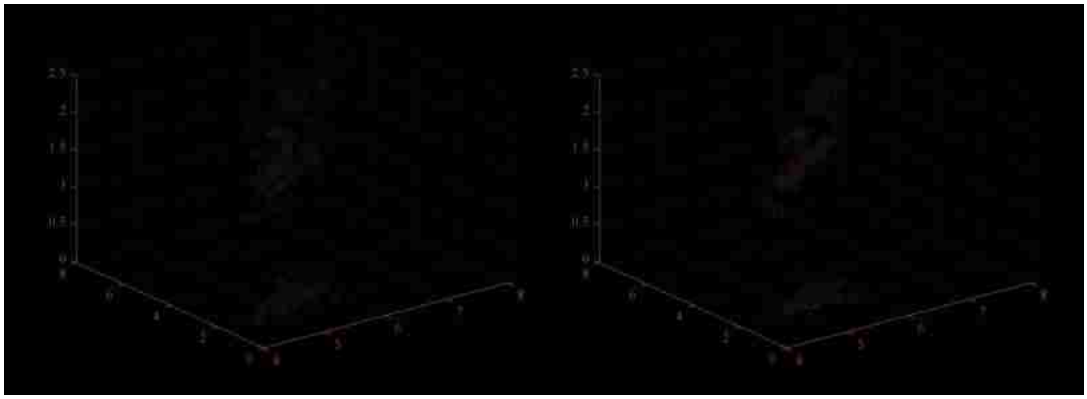


Figure 4.8: $\alpha = 0.9, \beta = 0.1$

Figure 4.9: $\alpha = 0.8, \beta = 0.2$

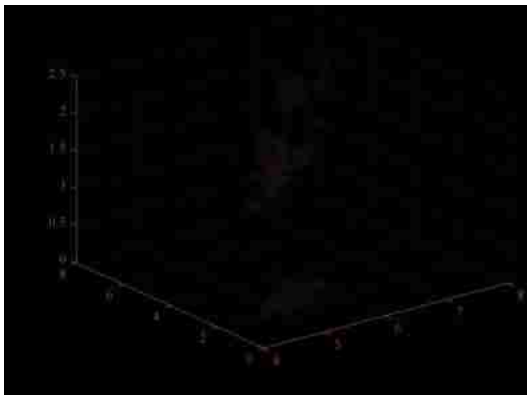


Figure 4.10: $\alpha = 0.7, \beta = 0.3$

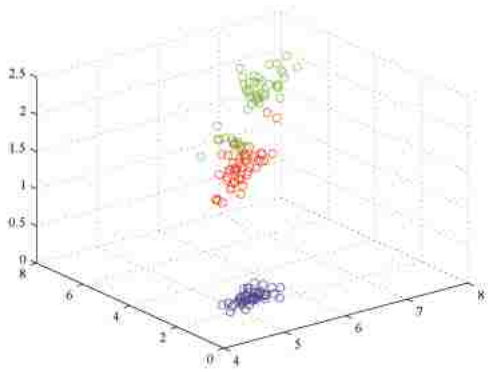


Figure 4.11: $\alpha = 0.6, \beta = 0.4$

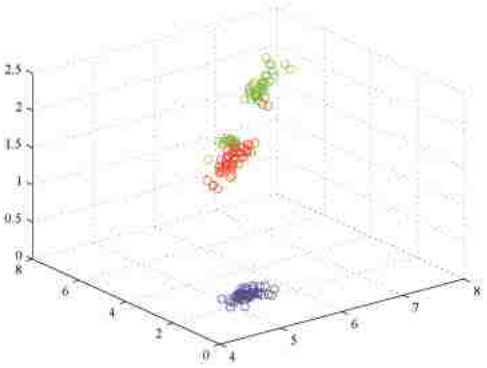


Figure 4.12: $\alpha = 0.5, \beta = 0.5$

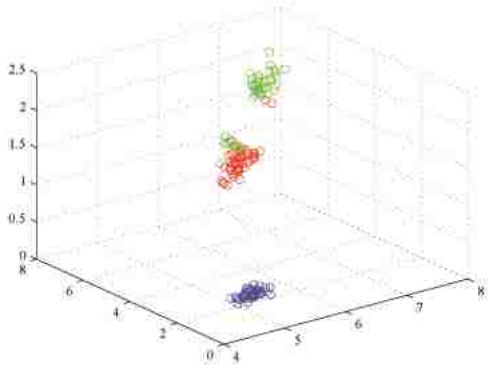


Figure 4.13: $\alpha = 0.4, \beta = 0.6$

4.10 Conclusion

In this chapter, we introduced new methods to distort the original data matrix: Constrained NMF, Sparsified CNMF, and Compressed NMF. We compared the per-

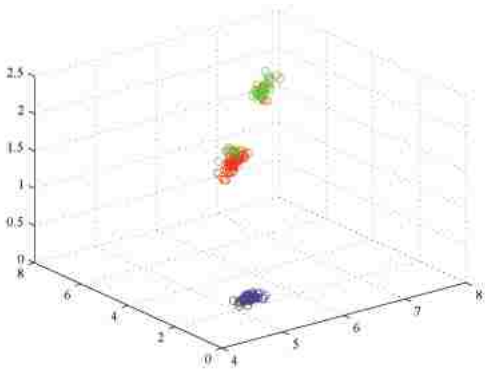


Figure 4.14: $\alpha = 0.3, \beta = 0.7$

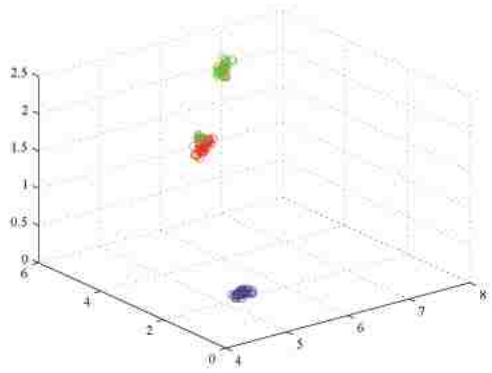


Figure 4.15: $\alpha = 0.2, \beta = 0.8$

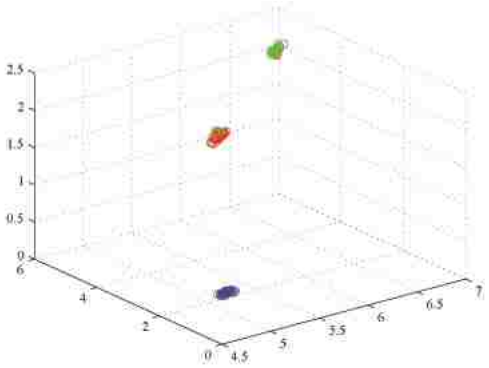


Figure 4.16: $\alpha = 0.1, \beta = 0.9$

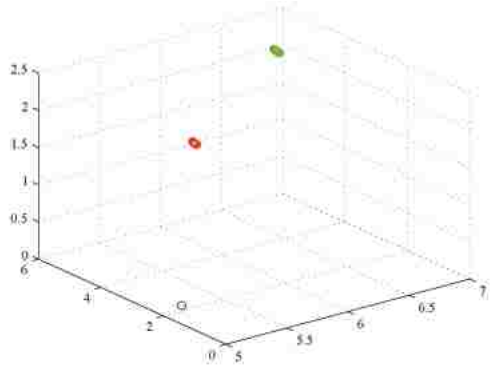


Figure 4.17: $\alpha = 0.0, \beta = 1.0$

formance of these methods both on data privacy level and data utility level to several existing strategies in privacy-preserving data mining, including two noise additive techniques, SVD, and NMF method. The experimental results demonstrate that the proposed strategies can perform much better than the existing methods in terms of clustering accuracy. Another observation was that matrix factorization based methods do not permit the same level of distortion as the additive random perturbation based methods.

The methods based on NMF are much more robust when it comes to privacy attack. When the methods like RD and RN are used, the random noise can be filtered out from the perturbed data and the privacy can be compromised. The accuracy of the cluster with the methods like RD and RN depends on the distance between the clus-

ters. If the clusters are well separated then accuracy can be fairly good but in case where the clusters are close, the accuracy decreases rapidly. Accuracy of ComNMF does not depend on the distance between the clusters.

Chapter 5 Correlation-Aware Data Perturbation for Linear Classifiers

Some of the methods in PPDM are suited for particular data mining techniques [20], while others are generic techniques that do not take into consideration the data mining techniques that need to be used [3], [9].

Our research work in this chapter falls into the former category, where perturbing the data is based on the technique that is used for classification. Our efforts address perturbation for linear regression (LR) which is one of the most popular classification tools.

5.1 Correlation-Aware Data Perturbation for Logistic Regression

LR depends on the correlation between the attributes and the class label. The approach accomodates that aspect. We start with LR followed by L1-logistic regression and discuss the need for it. Very little work has been carried out on privacy preserving LR. Chaudhuri et al. [8] provide a privacy-preserving regularized logistic regression based on a privacy preserving technique. Difference between the two methods is that, their work modifies the technique in itself while our method aims at perturbing the data. Li et al. [38] proposed adding auto-correlated noise to the streams of data based on principal component analysis (PCA). It falls into the category of generic perturbation. We call the method proposed in this capter as Context-Aware Perturbation also referred to as CAP in the following sections.

5.2 Introduction

Typical use-case scenario addressed in our work is as follows:

Let P1 and P2 be parties owning private databases D1 and D2 repectively. D1 is a large database compared to D2 which makes D1 ideal for learning knowledge. P2

wants to perform data mining and create a model based on D1 which can be applied to D2 for the prediction problem. P1 does not trust P2 and wants to make sure that P2 is not given any private information. In the rest of the text we represent D1 by T and D2 by V.

The chapter is organized as follows: Section 5.3 presents background about LR and the prediction problem, Section 5.4 formulates our problem statement while Section 5.5 discusses a couple of methods that achieve higher correlation with perturbation. The Section 5.5.2 presents our overall process and some properties of CAP. Problem with non-regularized LR leads to our selection of L1-regularized logistic regression which is discussed in Section 5.6. Experimental observations are presented in Section 5.7. We discuss extension to multiclass problem in Section 5.8.

5.3 Preliminaries

Before we define the privacy model, we will note a few preliminary points. We assume that each subject in the database is a real vector. Database contains several attributes where each has values $x_1, ..., x_n$, where $x_i \in \mathbb{R}$.

5.3.1 Logistic Regression

Logistic regression is a linear classifier that has been widely used in data mining for the purpose of prediction and classification [31]. A model is defined as logistic if the expression for probability of $output = 1$, given x can be expressed as;

$$P(y = 1|x_1, x_2, ..., x_k) = \frac{1}{1 + e^{-(\alpha + \sum \beta_i x_i)}} \tag{5.1}$$

where, α, β_i are the unknown parameters. β_i are the regression coefficients. As in Figure 5.1 output is always in between 0 and 1. This and S-shaped description of the combined effect make logistic regression particularly useful [33]. Representing

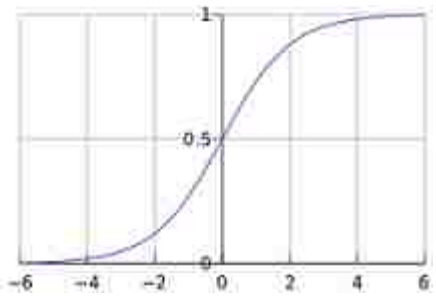


Figure 5.1: The logistic function with $\beta_0 + \beta_1 x_1 + e$ on the horizontal axis and $P(x)$ on the vertical axis

$P(y = 1|x_1, x_2, ..., x_k)$ as $P(y)$

$$P(y) = \frac{1}{1 + e^{-(\alpha + \sum \beta_i x_i)}} \tag{5.2}$$

Let us define a function such that:

$$logit^{-1}(a) = \frac{1}{1 + e^{-a}} \tag{5.3}$$

where $logit^{-1}(a)$ is the inverse logit function. From Equation (5.2) and Equation (5.3), we have

$$logitP(y) = \alpha + \sum \beta_i x_i \tag{5.4}$$

From above equation, we can see that $logitP(y)$ and x_i are linearly dependent.

5.3.2 Predictor Attributes Vs Nonpredictor Attributes

Logistic regression assumes that *no extraneous variables are included*, which in a real-world dataset is hard to confirm. Extraneous variables refer to the independent attributes which do not contribute to the dependent attribute or the class attribute. Generally dataset consists of both the important and unimportant predictor variables. Hence, it becomes necessary that the perturbation techniques do not increase the correlation between the independent attributes and class label that are unrelated.

We define a scalar value η called *correlation threshold*, which is the minimum absolute value confirming that label does depend on the attribute. It is especially

Nonpredictors	..	*Predictors*	..	*Class label*

Figure 5.2: Real World Dataset

true for real-world dataset as several attributes cannot be directly related. Different dataset will have different value for η. Domain knowledge is needed for a proper choice of δ. *P1* would determine η. Using feature selection to choose between the attributes can be a good research direction for the future.

5.4 Privacy Model

Most of the privacy preserving techniques have to address two key concerns: Perturbation and classification. We formulate our problem as;

Problem 1. *(Perturbation). Given a dataset A and the output label l_c. How to obtain the perturbed dataset $\tilde{A}$ where $||A-\tilde{A}||$ is as large as possible. In our implementation, we want to obtain larger $||A^s-\tilde{A}||$, where A^s is the ordered A based on the class label.*

Problem 2. *(Classification). Given a perturbed dataset $\tilde{A}$, the output label l_c and testset V. How to correctly predict the label for the testset.*

Our second problem formulation is different from most of the other techniques. Others consider reconstruction of original data to make the classification whereas we perturb the test data before the classification, requiring knowledge of η for classification. These problems are conflicting, the first one tries to perturb the data; the second one tries to learn from the perturbed data. We achieve absolute privacy by publishing nothing but it gives no utility, and publishing everything gives away all the privacy.

5.5 Approach

Increasing the perturbation results in lesser correlation between the attributes, which leads to lesser utility as predictions are less accurate. Our idea is to increase the correlation between the input attributes and class label or at least maintain it while perturbing the data. There are a couple of ways that result in increased correlation between attribute and class label which can be best illustrated by the following examples.

5.5.1 Example

Approach 1

Shifting the attribute values away from the mean value as in Figure 5.4 increases the correlation.

x	y
45	0
35	0
55	1
64	1

Figure 5.3: Original dataset, $corr(x, y) = 0.8988$

	x	y
↑	50	0
↑	42	0
↓	65	1
↓	74	1

Figure 5.4: Perturbed dataset, $corr(x, y) = 0.9402$

Approach 2

Another way of increasing the correlation is to move the values towards the mean value as in Figure 5.5 and Figure 5.6.

The second method has better utility, as utility is directly related with variance. More the variance less is the utility.

x	y
45	0
35	0
55	1
64	1

Figure 5.5: Original dataset, $corr(x, y) = 0.8988$

	x	y
↓	42	0
↑	39	0
↓	58	1
↑	61	1

Figure 5.6: Perturbed dataset, $corr(x, y) = 0.9878$

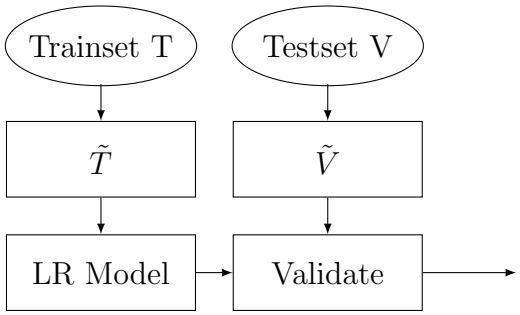


Figure 5.7: Process

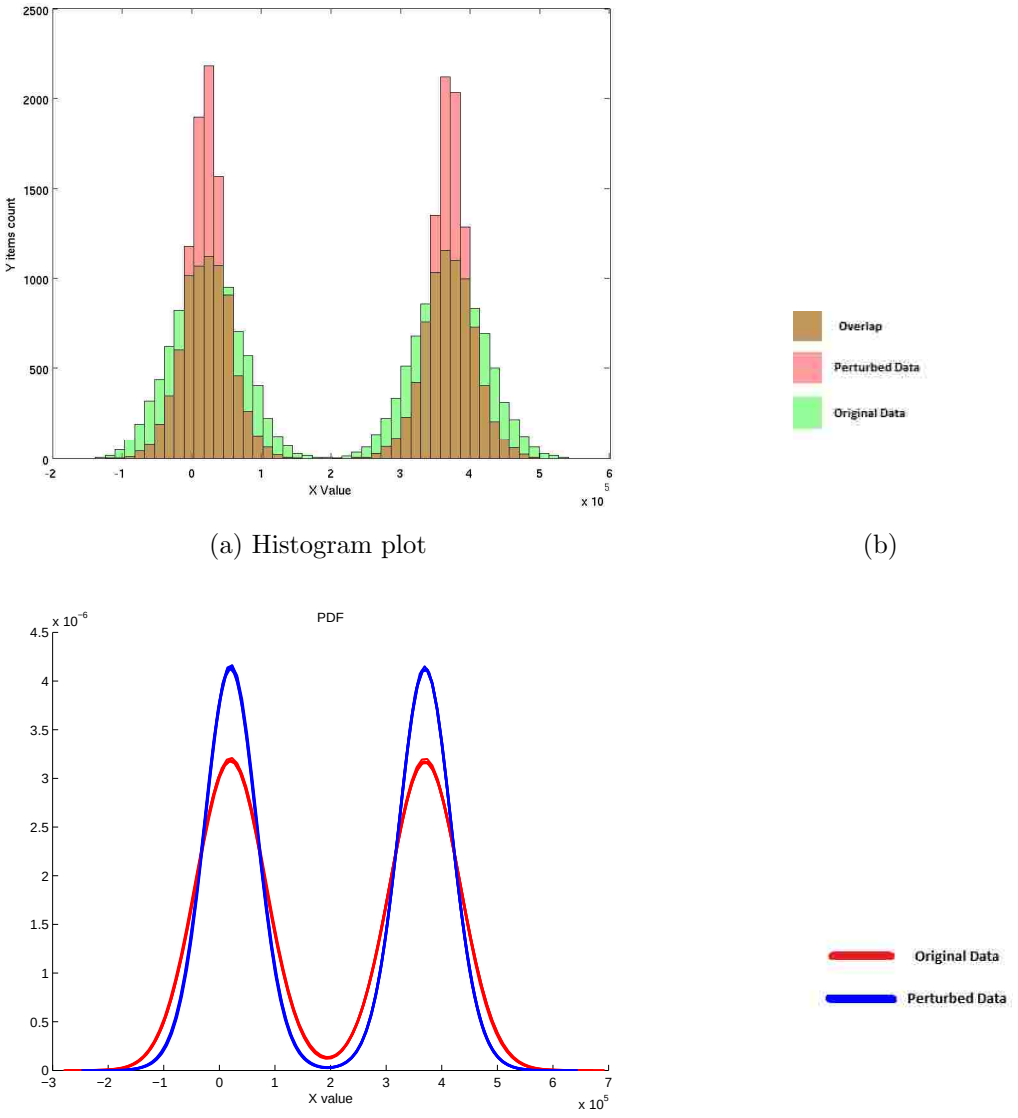
5.5.2 Overall Process

Different to other perturbation methods, CAP distorts both the training set (T) and the test set (V) before the classification as shown in Figure 5.7. Distortion of V requires knowledge of η and uses T to compare the correlation. Algorithm 4 helps in understanding the problem. If the correlation between the input attributes and output label is more than threshold in T, we apply CAP to the particular attribute in V. Knowledge of *percent* is not required to perturb V. As we want to preserve the range of each of the attribute values in a class, random percentage of distortion in between [0,100] is used for perturbing V.

Plot with the synthetic data

To illustrate the idea, the following figures show the histogram plot and the density function of the synthetic data which has normal distribution centered at two different centers. These figures demonstrate the behaviour of CAP for an attribute.

CAP makes the data more compact, which shares the idea proposed in [1, 54].



(a) Histogram plot (b)

(c) Density function (d)

Figure 5.8: Original data and the Correlation-Aware Perturbed data

We follow the scheme only if the attributes and class label have correlation above the threshold (η).

5.5.3 Algorithm

Algorithm 4 provides the details of the method. l_c^s represents the sorted class label where class 0 is followed by class 1. We perturb each of the attributes separately

Algorithm 4: Correlation-Aware Perturbation

input : A of size $n \times m$, l_c of size $n \times 1$, η, percent
output: $\tilde{A}$ of size $n \times m$

```
// Sort rows of A based on l_c, c is the number of items in class 0
```
A^s, l_c^s , c = Sort(A, l_c) ;
for $j \leftarrow 1$ **to** m **do**
    `// find mean for each class`
    $\mu_0 = mean(1 : c, i)$;
    $\mu_1 = mean(c + 1 : end, i)$;
    if $abs(corr(A^s(:, j), l_c^s(:, 1))) > \eta$ **then**
        for $i \leftarrow 1$ **to** n **do**
            `// Increase the correlation between` A^s`(:, j),` l_c^s
            if *$A^s(i, j)$ is in class 0* **then**
                `// move towards mean of class 0`
                $range = (\mu_0 - A^s(i, j))$;
                $noise = rand * range * percent$;
                $\tilde{A}(i, j) = A(i, j) + noise$;
            else
                `// move towards mean of class 1`
                $range = (\mu_1 - A^s(i, j))$;
                $noise = rand * range * percent$;
                $\tilde{A}(i, j) = A^s(i, j) + noise$;
    else
        `// Randomly perturb the data`
        $range = max(A^s(:, j) - min(A^s(:, j)))$;
        $noise = (rand - 0.5) * range * percent$;
        $\tilde{A}(:, j) = A^s(:, j) + noise$;

based on the correlation between the attributes and the output label. Simple random perturbation is performed if correlation is less than η. In each of the class, attributes are perturbed so that they move towards the mean of the class if the correlation is greater than or equal to η.

LR makes a fundamental assumption on the independence of attributes. The independence assumption licenses the classifier to collect the evidence for a class from individual attributes separately.

Lemma 1. Properties of CAP

- Let μ and σ^2 be the mean and variance of original data respectively. Similarly, let $\hat{\mu}$ and $\hat{\sigma^2}$ be that for the perturbed dataset $\hat{x}_{i=1}^{N}$ then, statistically the following relations hold: $\hat{\mu} = \mu$ and $\hat{\sigma^2} \leq \sigma^2$.

  Proof. The proof can be easily formulated by considering class 0 and class 1 separately. Since,

  $$\mu = \frac{\mu_0 \times n_1 + \mu_1 \times n_2}{n}$$

  where, class 0 has mean = μ_0 and n_1 items. Similarly, class 1 has mean = μ_1 and n_2 items. From the Figure 5.8c, we can observe the effect of the method on the distribution of the data. As long as the distortion percent $\leq$100% the distorted value will be contained within the range defined by the highest and the lowest values; hence, the method will statisfy the relation. This implies that the perturbation method do not increase the variance of the important attributes. □

- Let us define the upper dataset extent values $d_{max} = \max_j abs(x_j - \mu)$. Similarly, for the distorted data $\hat{d}_{max} = \max_j abs(\hat{x}_j - \hat{\mu})$. The extend of the distorted dataset is not increased, i.e., $\hat{d}_{max} \leq d_{max}$.

  Proof. Let us consider one class at a time. Then for class 0, we can see

  $$\begin{aligned} \hat{d}_{max0} &= \max_j abs(\hat{x}_j - \hat{\mu_0}) \\ &\leq \max_j abs(x_j - \mu_0) \\ &= d_{max0} \end{aligned} \tag{5.5}$$

  Same goes for class 1. For correlation preservation, it is desirable that the range of attributes do not expand within the class. □

Let x, y, z be the vectors such that $corr(x, y) = \rho_{xy}$, $corr(y, z) = \rho_{yz}$, $corr(x, z)$=ρ_{xz}. Little can be said of ρ_{xz} based on ρ_{xy} and ρ_{yz}. [36] showed that:

$$\rho_{xy}\rho_{yz} - \sqrt{(1-\rho_{xy}^2)(1-\rho_{yz}^2)} \leq \rho_{xz}$$

$$\rho_{xz} \leq \rho_{xy}\rho_{yz} + \sqrt{(1-\rho_{xy}^2)(1-\rho_{yz}^2)}$$

Thus, nothing can be inferred regarding the sign and magnitude of the correlation ρ_{xz} although ρ_{xy}, ρ_{yz} both are positive. Which means ρ_{xz} can be anywhere from [-1, 1] unless ρ_{xy} and ρ_{yz} are significantly high.

Theorem 1. Let us consider that $\rho_{xy}, \rho_{yz} > 0$, then the increment in ρ_{xy} and ρ_{yz} leads to the increment in ρ_{xz} .

Proof. Proof can be formulated with *Spearman's rank correlation coefficient (ρ)* [51] which is given by

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{5.6}$$

where

$$d_i = x_i - y_i \tag{5.7}$$

ρ_{xy} can be expressed as

$$\rho_{xy} = 1 - \frac{6\sum (x_i - y_i)^2}{n(n^2-1)} \tag{5.8}$$

Similarly,

$$\begin{aligned} \rho_{yz} &= 1 - \frac{6\sum (y_i - z_i)^2}{n(n^2-1)} \\ &= 1 - \frac{6\sum (z_i - y_i)^2}{n(n^2-1)} \end{aligned}$$

ρ will increase only with the decrease in d_i, which means increase in ρ_{xy} is because of decrease in distance between x_i and y_i. Similarly, increment in ρ_{yz} is a result of decrease in distance between z_i and y_i and it leads to decrease in distance between x_i and z_i, as both the values are approaching y_i. □

Theorem 2. Let x, y, z be the vectors such that $corr(x, y) = \rho_{xy}$, $corr(y, z) = \rho_{yz}$, $corr(x, z)$=ρ_{xz} and $\rho_{xy}, \rho_{yz} < 0$, then when both ρ_{xy} and ρ_{yz} decrease it leads to decrement in ρ_{xz} .

Proof. Similar to earlier proof.

The theorem implies that the correlation among the independent attributes can change even with the change in correlation between the independent attributes and dependent attribute performed separately and independently. We show the problem of colinearity in the following sections. □

5.6 L1-regularized Logistic Regression

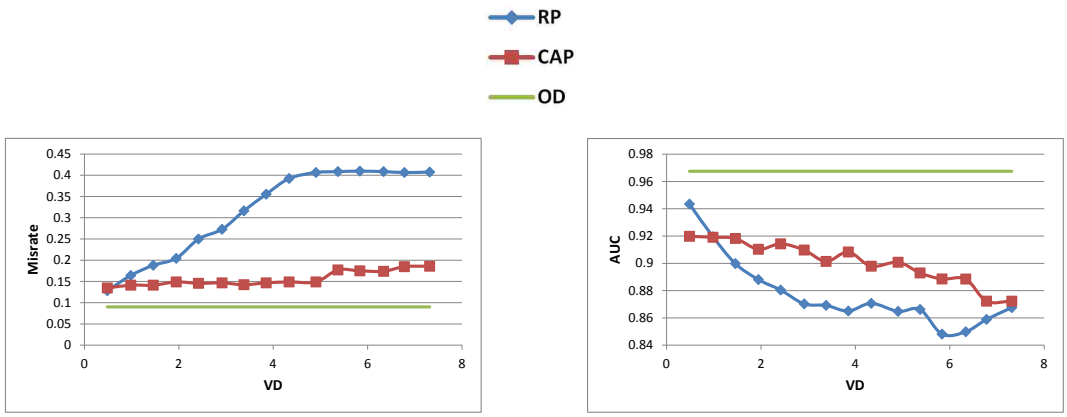
Logistic regression makes a few assumptions, one that is important to us is;

- *The independent variables are not linear combinations of each other.*

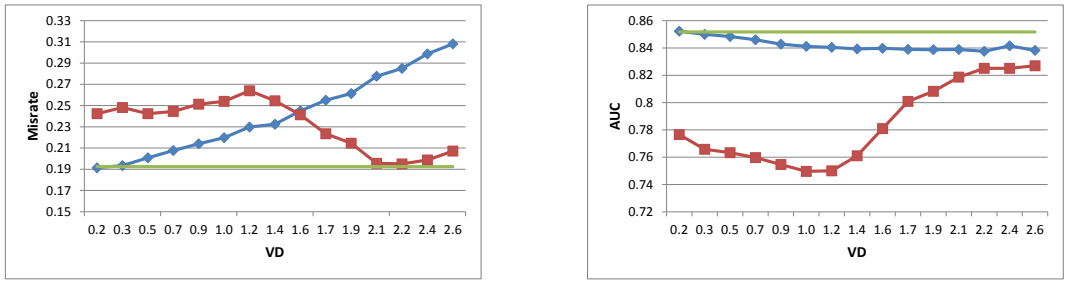
A careful observation reveals that changing the correlation between the attributes and class label results in increased correlation between the input attributes, which is a problem for conventional logistic regression. The problem is termed as *collinearity* which *refers to the condition of very high correlations among independent attributes*, leading to features being very much alike. A principal danger of such redundancy is *overfitting* which occurs when a model captures idiosyncrasies of the input data, rather than generalizing. There are a couple of work arounds to handle *collinearity*, which are as follows:

- Collect more data.
- Remove features that are redundant.

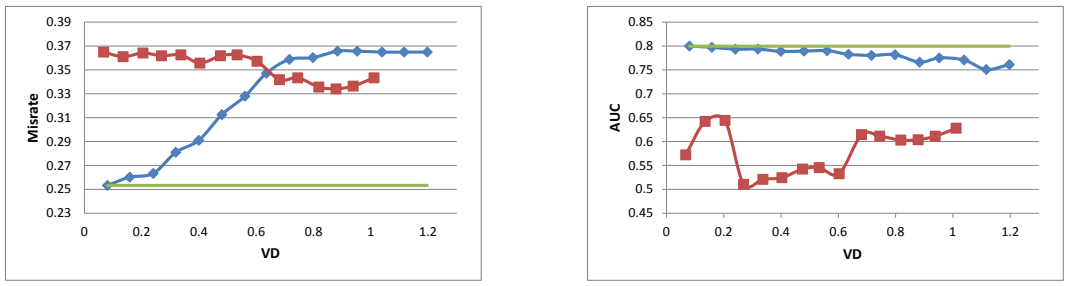
For our purpose, the first option does not work as CAP with added data will also lead to collinearity. This issue can be handled by removing redundant features. One



Spambase Dataset



Magic Gamma Dataset



Wine Quality Dataset

Figure 5.9: Experiment 1: Misrate Vs VD, AUC Vs VD with LR

convenient method is to perform regularized logistic regression. We work with L1-logistic regression which approximates feature selection and regularizes the function. L1 logistic regression is the optimization problem, expressed as

$$\min \sum_{n=0}^{N-1} -\log P(y_n|x_n) + \lambda||\beta||_1 \tag{5.9}$$

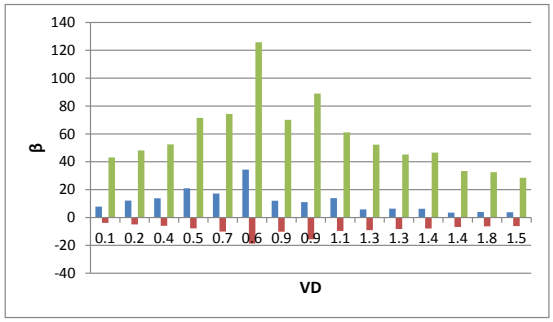


Figure 5.10: Spambase Dataset

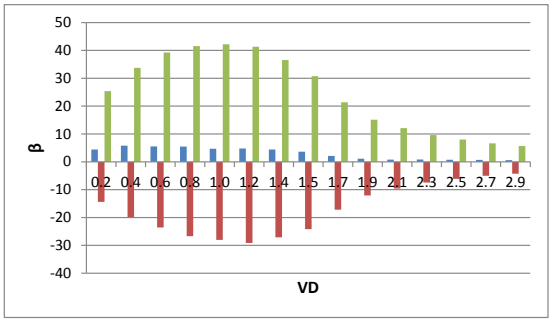


Figure 5.11: Magic Gamma Dataset

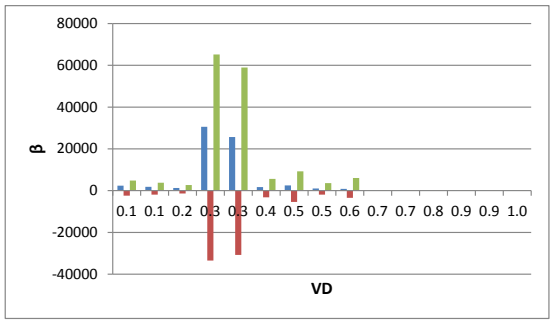


Figure 5.12: Wine Quality Dataset

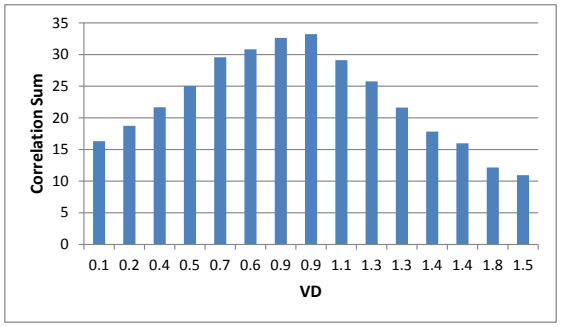
Figure 5.13: Legend

Figure 5.14: Experiment 2: Values of β_i

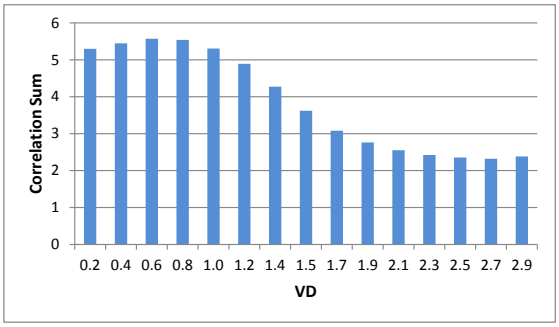
where β is the regularization term, which forces prameters to be small. λ is a scalar value that keeps the value of β under control.

5.7 Experiments

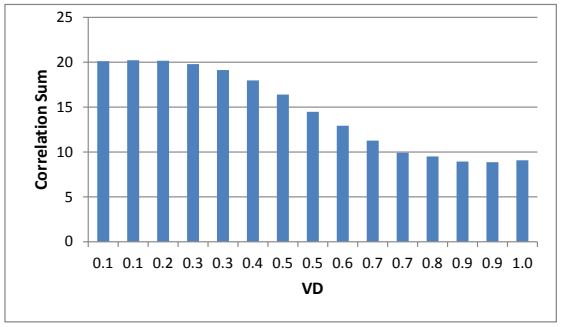
For the experimental purpose we have used VD as the perturbation metric. Misclassification Rate and AUC are the utility metrics. We compare CAP with Random Perturbation. Random perturbation is one of the most popular perturbation techniques that can produce wide range of perturbations. Some of the matrix based perturbations can only produce smaller amount of perturbation. In addition, the level of confidentiality associated with random perturbation and CAP are comparable. Methods like rotation pertubation are generally considered weaker methods as



Spambase Dataset



Magic Gamma Dataset



Wine Quality Dataset

Figure 5.15: Experiment 2: Change in correlation between predictor attributes

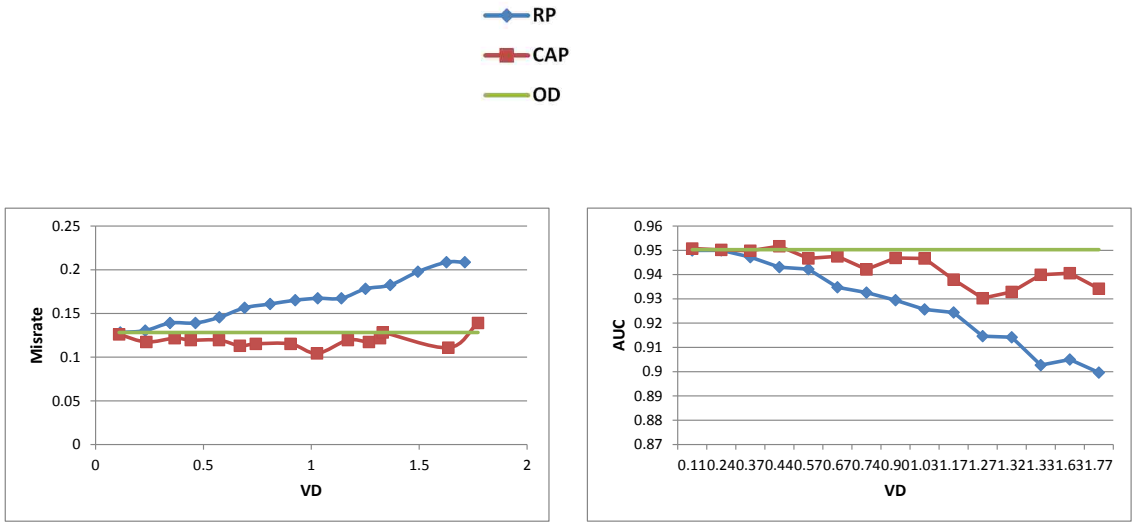
privacy can be breached by knowing the angle of rotation. These reasons make random perturbation an ideal choice for the comparision. We represent original dataset with OD, CA perturbed dataset with CAP and Random Perturbed dataset with RP.

5.7.1 Datasets

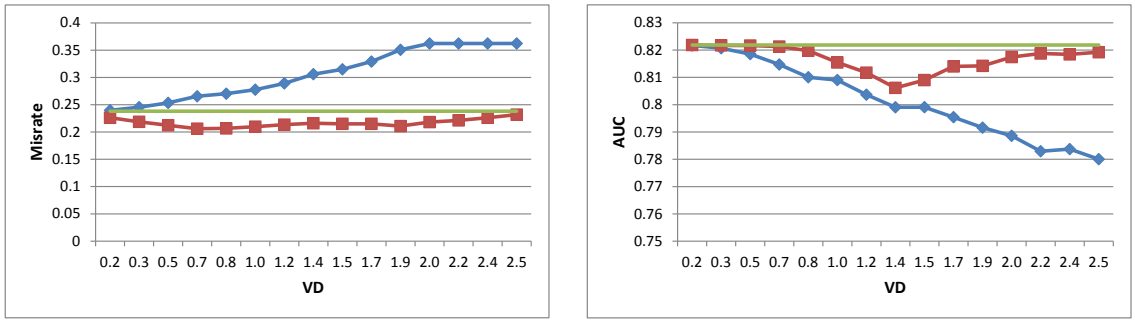
The following datasets have been used for our study: Wine Quality Dataset, Spambase Dataset, and Magic Gamma Telescope Dataset.

5.7.2 Experiment 1: Non-regularized Logistic Regression

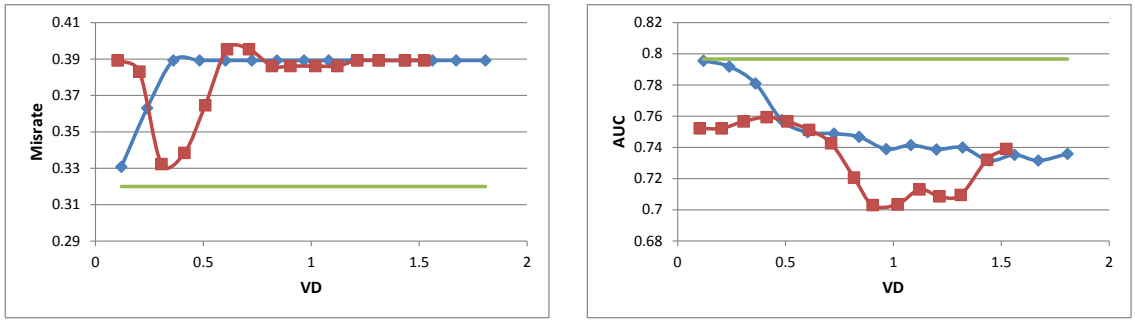
Figure 5.9 shows the results of our method with non-regularized LR. Although the results for Spambase dataset is in-line with our expectation, it does not behave as



Spambase Dataset



Magic Gamma Dataset



Wine Quality Dataset

Figure 5.16: Experiment 3: Misrate Vs VD. AUC Vs VD with *regularized* LR

well with other datasets. CAP has higher misrate and gradually starts to get closer to the results of random perturbation. Wine dataset has even worse AUC plot. The following experiment provides the understanding of the observations that were made.

5.7.3 Experiment 2: Relation between Colinearity and CAP performance

In Figure 5.14, we plot minimum, maximum and the sum of the absolute value of β_i for each of the run with different VD. Experiment with β_i of Wine Quality dataset reveals why our method fails far behind the random perturbation. Values of β_i are extremely large; the phenomenon which we discussed earlier as *collinearity*.

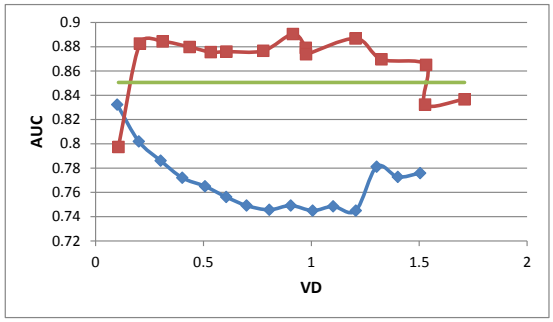
Second observation is depicted in Figure 5.15; it is a plot of the sum of correlation coefficients between the attributes having correlation with class label $> \eta$. Ideally, CAP should maintain the correlation up to certain distortion level as with Spambase dataset and once it passed the threshold the correlation sum should decrease. In case of Wine and Magic Gamma datasets the method fails to maintain the sum of the correlation as there is only decreasing trend.

5.7.4 Experiment 3: L1-Regularized Logistic Regression

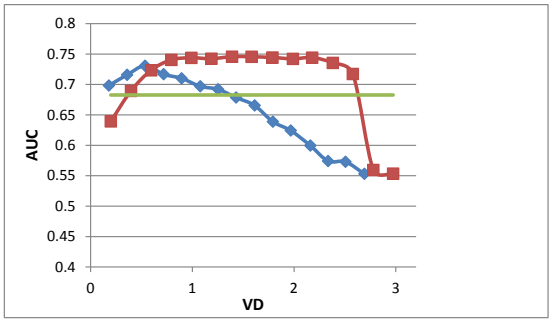
From Experiments 1 and 2 we can observe the problem with using LR. To mitigate the problem of LR, we employ regularized LR. The results are depicted in Figure 5.16. The regions that are in Figure 5.9 no longer exist. The AUC plot and the misrate plot for Wine Quality dataset are better than with non-regularized LR. L1-regularized LR does give better overall performance.

5.7.5 Experiment 4: Naive Bayes

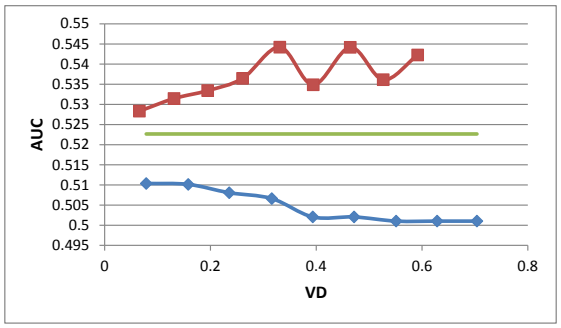
In addition to LR, we observed the performance of CAP with another linear classifier: naive Bayes (NB). It is interesting to observe that in Figure 5.17 CAP produces better results than the original data with every dataset. The result can be attributed to the



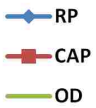
Spambase Dataset



Magic Gamma Dataset



Wine Quality Dataset



Legend

Figure 5.17: Experiment 4: AUC Vs VD with *naive Bayes*

strong independence assumption NB makes between the attributes. As CAP treats every attribute separately, it results in the favorable observations.

5.7.6 Experiment 5: Decision Trees

In our last experiment, the aim was to validate if the same level of performance is observed with non-linear classifiers. Our experiment used decision tree for the classification. We did not observe same level of performance; Accuracy was significantly lower. This is the foundation for our research presented in next chapter where we present detailed reasoning for the failure.

5.8 Discussions

- ***Multi-class prediction problem*** Our analysis and observations have been limited to two class classification problem. Extension to multi-class problem is an obvious future step. Based on the type of class label, we can extend CAP to cover multi-class problem.

  - *Case I: Class label with ordering*: If the class label has some sort of ordering like 1, 2, and 3 or bad, average, and great, in that case CAP should be easily extendable. Each class can be assigned a numerical value and the computation shall be done taking multi-class in consideration.

  - *Case II: Class label without any ordering*: It can be a problem in the case of yeast dataset where classes are nuclear, cytosolic, mitochondrial and so forth. In such a case, it is hard to give a particular value to a particular group. One likely solution is to compare two classes at a time, and give those two classes number of either 0 or 1. However, practical feasibility of the solution remains to be seen.

5.9 Conclusion

We compared our result with that of random perturbation. The proposed data perturbation technique outperforms random perturbation as our method uses statistical analysis in addition to performing random perturbation. Our experiment on naive Bayes shows that the method can be effectively applied to other linear classifiers. CAP is a robust method compared to rotation based perturbation as privacy does not depend on just the rotation value. We discussed issues like collinearity and non-linear classification. Finally, possible approaches to handle multiclass classification were presented.

Experiment with decision tree failed to extend the effectiveness shared with linear

classifers. The next chapter of this dissertation is the extension for the non-linear classification.

Chapter 6 Neighborhood-Aware Data Perturbation for Non-linear Classifiers

This chapter is the continuation of where we left off in the previous chapter. In our previous work, we proposed a data distortion technique that considered the correlation between attributes and class label. The idea was to maintain the correlation of the more important attributes with the class label even with distortion. Although this method was highly effective with linear classifiers, it was ineffective while handling non-linear data classification. Our aim in this chapter is to address that important issue. With the focus being shifted to non-linear classifiers, our idea still remains the same; to take into consideration the relation between the different dimensions. In this chapter we present a novel approach that can handle non-linear classification.

6.1 Motivation

The direct implementation of the CAP for classification using decision tree underperforms. We closely obeserved the scenario to find the reason that can be explained by the Figure 6.1.

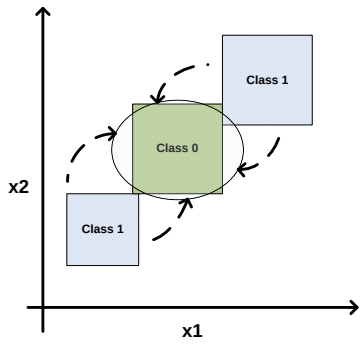
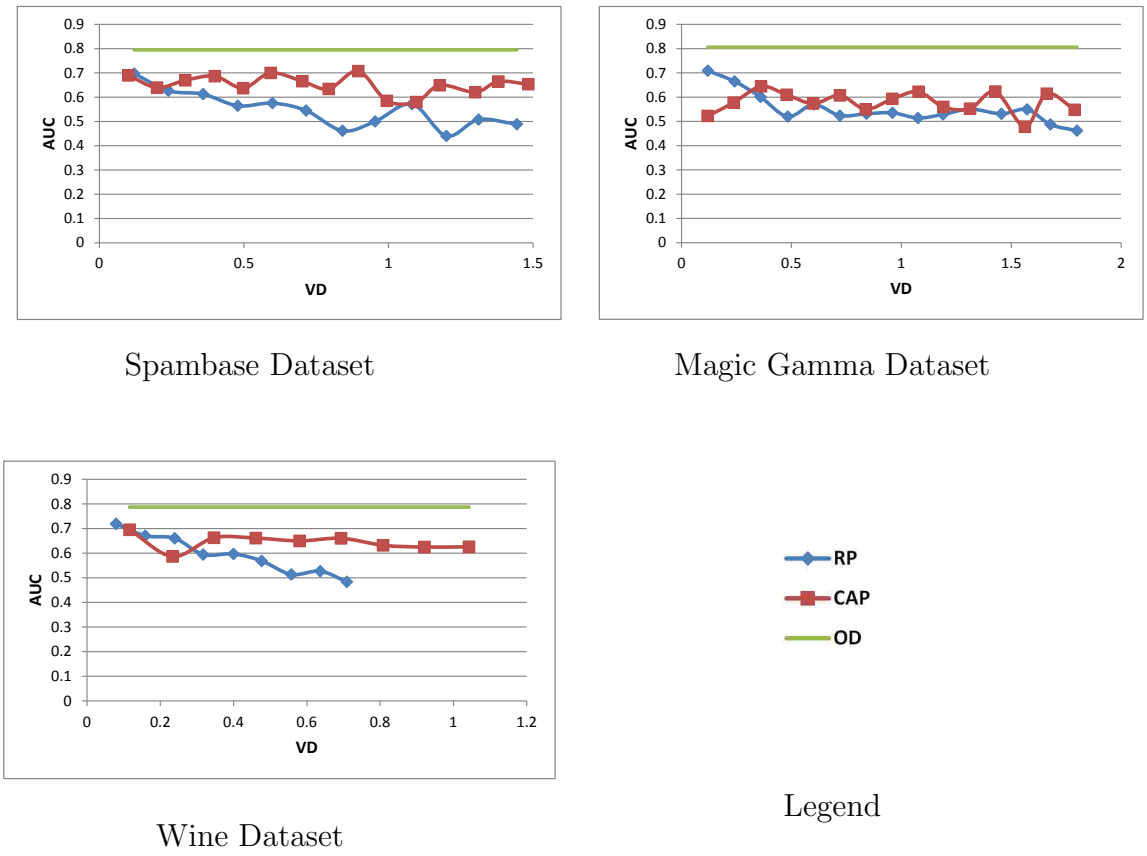


Figure 6.1: Problem with non-linear classification

Decision tree partitions the data into different regions based on the tree that it generates as in the Figure 6.1 where one class is sandwiched between the other classes. The mean of class 1 will move closer to the mean of class 0, resulting in intermixing of data from two separate classes. Hence, our method needs some alternations. A modification can be to perform our method for each of the regions than for the class itself. So, the mean will tend towards the mean of the region. We present our observation with the modified CAP in the Figure 6.2.



Spambase Dataset

Magic Gamma Dataset

Wine Dataset

Legend

Figure 6.2: Modified CAP with Decision Trees

Although we can see that the technique works for the decision tree, but it fails to achieve similar result with other non-linear classifiers especially SVMs. SVM operates by implicitly changing lower dimentional data into higher dimension. Hence, a closer and detailed study is presented in this chapter.

The typical use-case scenario addressed in our work is similar to the earlier chapter: *Let P1 and P2 be parties owning private databases D1 and D2 repectively. D1 is a large database compared to D2 which makes D1 ideal for learning knowledge. P2 wants to perform data mining and create a model based on D1 which can be applied to D2 for the prediction problem. P1 does not trust P2 and wants to make sure that P2 is not given any private information.*
In the rest of the text we represent $D1$ by T and $D2$ by V. Our experimental studies demonstrate the flexibility of our approach for privacy preserving as it works as good with other non-linear classifer as it does with SVMs. The anonymized data closely match the statistical characteristics of the original data.

The chapter is organized as follows: Section 6.2 presents background on SVM and Covariance; Section 6.3 formulates our problem statement, while Section 6.4 discusses the approach we devise. Section 6.5 and Section 6.6 present the properties and algorithm for the proposed method. Section 6.7.1 presents perturbation and utility metrics. Experimental observations are presented in Section 6.7.

6.2 Preliminaries

6.2.1 Support Vector Machines

SVM finds an optimal hyperplane that classifies linearly separable patterns with as large margin as possible. In order to handle non-linear data, kernel functions are required. These functions provide implicit mapping of points into a higher dimensional space so called feature space (H), where they can be linearly separable. It allows SVM models to perform separations even with very complex boundaries.

Kernel functions

It is a function Φ, which maps the data points x_i of the data space L to the feature space H where a linear separation is possible.

$\Phi : \mathbb{R}^n(L) \rightarrow H$

where L is lower dimensional space and H is higher dimensional space. Common kernel functions are:

- Linear Kernel (LK) $k(x_i, x_j) = \langle x_i, x_j \rangle$
- Radial Basis Kernel (RBK) $k(x_i, x_j) = \exp(\frac{-||x_i - x_j||^2}{c\sigma_0^2})$
- Combination of Kernels (CK) $k(x_i, x_j) = \lambda_1 k_1(x_i, x_j) + \lambda_2 k_2(x_i, x_j)$
- Multilayer Perceptor Kernel (MLK) $k(x_i, x_j) = \tanh(s\langle x_i, x_j \rangle + c)$

The feature space H must be a Hilbert space, which is a vector space in which a dot product (scalar product) is defined. The positive aspect about kernel function is that we do not need to know what the feature space H actually looks like; we only need the kernel function, which returns a measure of similarity. From the equations above, we can observe that for LK, CK, and MLP kernel functions depend on the dot product.

6.2.2 Covariance

Covariance gives the measure of how much two random variables change together. Positive values of covariance indicate the tendency to show similar behavior, while the negative values indicate the opposite behavior.

The covariance between two jointly distributed real-valued random variables x and y is given as

$$\sigma(x, y) = E[(x - E[x])(y - E[y])] \tag{6.1}$$

where $E[x]$ is the expected value of x also known as the mean of x.

6.2.3 Covariance Matrix

The covariance matrix captures the variance and linear correlation in multivariate/multidimensional data. Eigenvectors of the covariance matrix can be used to discover structural information about the data.

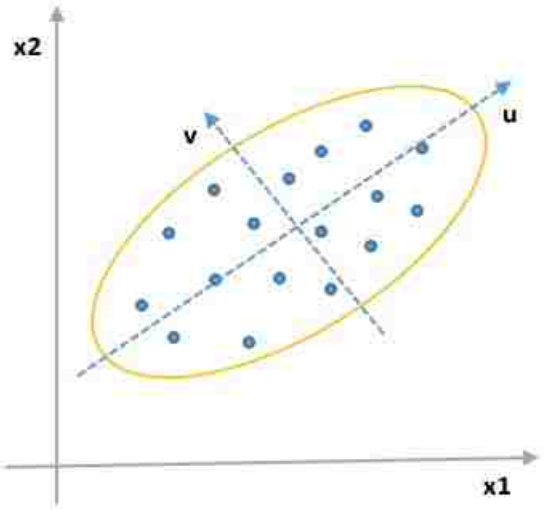


Figure 6.3: Eigenvectors of Covariance Matrix

The eigenvectors of the covariance matrix provides the important relations in a dataset. As in Figure 6.3, u and v are two eigenvectors of the covariance matrix for the 2-dimensional data with attributes x_1 and x_2. The first eigenvector(u) is the direction of greatest variance, the second eigenvector (v) is the direction of greatest variance among those that are orthogonal to the first eigenvector. Since the data is 2-dimensional, the data does not have any other eigenvector. In this case, eigenvector(u) is more important for the data than the eigenvector(v). We can preserve the data utility by preserving the more important eigenvectors of the covariance matrix of the distorted dataset.

6.2.4 Covariance and Inner Product

The covariance is closely related to the concept of inner product in the theory of vector spaces. If x and y are real-valued random variables, define the inner product of x and y by

$$\langle x, y \rangle = E(xy). \tag{6.2}$$

An important property of expectation is the linearity property, which states

Definition 1. For any constants $a, b \in \mathbb{R}$, $y = ax + b$ is a random variable whose expectation is $E(y) = aE(x) + b$.

Simplification of Equation (6.1) using Equation (6.2) combined with Definition 1 yields

$$\begin{aligned} \sigma(x, y) &= E[(x - E[x])(y - E(y)] \\ &= E[xy - xE[y] - E[x]y + E[x]E[y] \\ &= E[xy] - E[x]E[y] - E[x]E[y] + E[x]E[y] \\ &= E[xy] - E[x]E[y] \end{aligned}$$

Equation (6.3) expresses covariance as the difference between expected value of product of x, y and the product of expected value of x and the expected value of y.

Figure 6.4: Inner Product

6.2.5 Singular Value Decomposition

Data distortion techniques have their own advantages and disadvantages, which makes it hard to compare one method with another. Comparisions are meaningful when two methods behave or operate on similar principles. In our study we have made the comparision with Truncated SVD (TSVD) as introduced in [67]. A good choice for the rank of SVD could capture the main structure of a data collection and ignore the irrelevant noise. TSVD eliminates the smallest singular vector first perserving the dominating singular vectors and singular values that matter much to the dataset, which corresponds to our methods when perturbation is performed in small locality hence, preserving the singular vectors. Let A be a matrix of dimension $n \times m$ representing the original dataset. The rows of the matrix correspond to data objects and the columns to attributes. The SVD of the matrix A can be written as

$A = U \sum V^T$ where U is an $n \times n$ orthonormal matrix having the left singular vectors of A as its columns, $\sum$ is an $n \times m$ diagonal matrix whose nonnegative diagonal entries are the singular values in a descending order and V is an $m \times m$ orthonormal matrix.

We can create a rank-k approximation A_k to the matrix A by defining, $A_k = U_k \sum_k V_k^T$ where U_k contains the first k columns of U , $\sum_k$ contains the k largest nonzero singular values of A , and V_k^T contains the first k rows of V^T.The truncated rank k singular value decomposition represents the best minimum-variance linear estimate $\tilde{A}$ to A among all possible k-dimensional subspaces.

A theorem proven by Eckart and Young shows that the error in approximating a matrix A by A_k can be written as

$$||A - A_k||_F \leq ||A - B||_F \tag{6.3}$$

where B is any matrix with rank k. F stands for Frobenius norm. [10] mentions that truncated singular value decomposition A not only is the best approximation to A in

the sense of norm, but also is the closest approximation to A in the sense of statistics.

6.3 Privacy Model

As we are trying to solve the same problem but for non-linear classifiers, the privacy model can be referenced from Section 5.4.

6.4 Approach

As data distortion techniques change the structure of the data leading to change in the eigenvalues and eigenvectors. Eigenvectors tell about the variance in data that has highest importance. It is important to maintain the eigenvectors even in the distorted dataset.

The only way we can preserve the covariance between x and y from Equation (6.3) is to have the right hand side constant between the original matrix and distorted matrix. This is possible if we were to operate in a small neighborhood, in that case the covariance structure does not change much. This will lead to preservation

Algorithm 5: Neighborhood-Aware Perturbation

input : A of size $n \times m$, l_c of size $n \times 1$, r
output: $\tilde{A}$ of size $n \times m$

A^s, l_c^s , c = nclassify(A, r) ;
$index = 1$;
n_c = unique(l_c^s) ;
for $j \leftarrow 1$ **to** n_c **do**
    $tempdata = A(l_c^s == i, :)$;
    $[n_t, m_t] = size(tempdata)$;
    `// find mean for each class`
    $\mu_0 = tempdata(1, :)$;
    for $i \leftarrow 1$ **to** n_t **do**
        for $k \leftarrow 1$ **to** m_t **do**
            $\tilde{A}(index, k) = A(index, k) + 2 * rand * (\mu_0(k) - tempdata(i, k))$;
        $index = index + 1$;

of covariance matrix that will eventually result in a similar eigenvector as will the orginal matrix. Unlike Figure 6.1, the issue can be handled if neighborhood is taken into consideration.

Hence, we define a threshold for neighborhood called neighborhood radius (r). We perform in a group that lies within the radius of r. Domain expertise is needed to decide on r as it depends on the data. We call the new method Neighborhood-Aware Perturbation referred to as NAP.

6.4.1 Overall Process

Figure 5.7 shows the process in which both the trainset and the testset are perturbed, which contrasts with the general approach. In our experimental results we have considered both appraoches: one with perturbed testset and one without it.

6.5 Properties of NAP

Let $\hat{f}_X$ represent a classifier $\hat{f}$ trained with dataset X and $\hat{f}_X(Y)$ be the classification result on dataset Y. Let $T(X)$ be any transformation function, which transforms the dataset X to another dataset X_0. We use $Err(\hat{f}_X(Y))$ to denote the error rate of classifier $\hat{f}_X$ on testing data Y and let ϵ_1 be some small real number, $\epsilon_1 < 1$.

Definition 2. A classifier $\hat{f}$ is invariant to some transformation T if and only if $Err(\hat{f}_X(Y)) = Err(\hat{f}_{T(X)}(T(Y))) + \epsilon_1$ for any trainset X and testset Y.

We present some of the properties of NAP as follows:

Property 1. Value of r should be chosen in such a way that $cov(x_1, y) + cov(x, y_1) < \epsilon_2$ can be decided from the covariance between the attributes and the random noise, where x, y are the attributes and x_1 and y_1 are the noise added to the attribute x

and y respectively. ϵ_2 is a small real number that represents the change in covariance between x and y.

Proof. Since x_1, y_1 are random and independent, $Cov(x_1, y_1) = 0$. From the properties of covariance, we know

$$\begin{aligned} &= Cov(x^*, y^*) \\ &= Cov(x + x_1, y + y_1) \\ &= Cov(x, y + y_1) + Cov(x_1, y + y_1) \\ &= Cov(x, y) + Cov(x, y_1) + Cov(x_1, y) + Cov(x_1, y_1) \\ &= Cov(x, y) + Cov(x, y_1) + Cov(x_1, y) \end{aligned} \tag{6.4}$$

Algorithm 6: Algorithm to classify small neighborhood $nclassify$

input : A of size $n \times n$, r
output: A_c of size $n \times m$, l_c

```
// initialize variables
class_c = 0;
A_c = [];
l_c = [];
index = 0;
// Do until any items are left
while A ≠ [] do
    // select a item from the matrix
    item_0 = A(0, :);
    m = size(A) − 1;
    class_c = class_c + 1;
    l_c(index) = class_c;
    A_c(index) = A(0, :);
    // Do it for every remaining item
    for i ← 1 to m do
        // Check if the two items are close
        if dist(item_0, A(i, :)) < r then
            l_c(index) = class_c;
            // add that item to the new matrix
            A_c(index) = A(i, :);
            index = index + 1;
```

Hence, from Equation (6.4) we can see that r should be chosen in such a way that $Cov(x, y_1) + Cov(x_1, y) < \epsilon_2$. □

Property 2. When the $r = 0$, $A = \tilde{A}$ and $||A - \tilde{A}||$ is maximum when r is big enough to contain all the elements.

Proof. When $r = 0$, each item is contained within its own sub-cluster and the mean of attributes is equal to the attribute itself, which proves the first part.
For the second part, Let A be the original dataset and A_{sub} be a sub cluster of radius r_{sub}, where A_{sub}=$[A_1, A_2, .., A_k]$. If any item $A_j \notin A_{sub}$ is added to A_{sub}, then the new radius $r'_{sub} \geq r_{sub}$. The proof follows immediately.

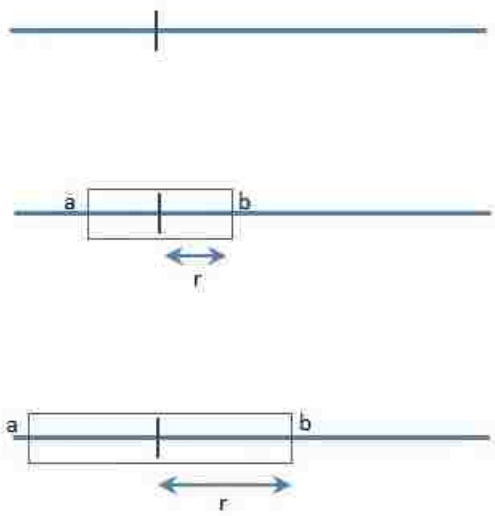


Figure 6.5: 1D distortion

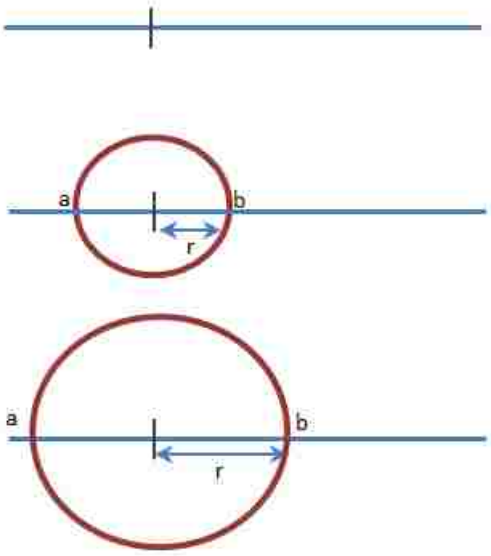


Figure 6.6: 2D distortion

Figure 6.7: Dependence of distortion on r

From Figure 6.7, we can see that distortion level is dependent on the value of r. The perturbation method can change the value in the range of $[a, b]$. The larger the

range, the greater is the distortion. The range of the distortion is dependent on the value of r. □

Property 3. Let A be a normalized matrix, r_{sub} be the radius of the sub-cluster on the original matrix, and r'_{sub} be the radius of the sub-cluster in the perturbed matrix $\tilde{A}$. The relation between r_{sub} and r'_{sub} can be expressed as:

$$|r_{sub} - r'_{sub}| = r\sqrt{n} \tag{6.5}$$

where n is the dimension of A.

Proof. We can easily see that the proof holds for 1-D case. We provide the proof for the 2-D case which can be generalized to the n dimensions.

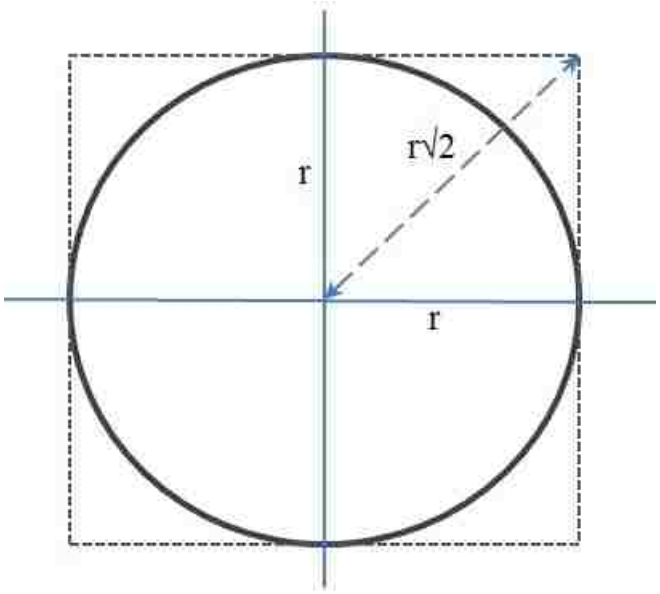


Figure 6.8: Radius of sub-cluster in distorted dataset

Since the maximum value in any dimension can be at most r, the farthest point is either of the cornors on the square presented in Figure 6.8, which is at a distance of $r\sqrt{2}$. For the 3 dimensional case, farthest distance would be $r\sqrt{3}$, which can be extended to n dimensions. □

[30] showed that original data can be accurately estimated from the perturbed dataset using spectral filter that exploits some theoretical properties of random matrix. So, the noise distribution $F(v)$ has to be completely known. The method proposed in this chapter divides the data into sub-clusters and performs random perturbation on each of those smaller clusters. Since the perturbation added to each of the cluster has different distribution based on the cluster radius, it becomes difficult to filter the noise. The method provides user with the flexibility of selecting the cluster size which eventually leads to selecting privacy.

6.6 Algorithm

Provided dataset A, our objective is to achieve perturbed dataset $\tilde{A}$. In the algorithm 5, l_c is the vector representing the label for the class. The first task in this process is to classify data into smaller cluster defined by neighborhood radius r. After the classification is done, we perform random additive perturbation on the dataset. The techniques like CAP discussed in the earlier chapter and rotation perturbation method are also possible.

6.7 Experiments

Our experiments have compared NAP with TSVD. There are two versions of NAP. The first one is NBP which perturbs the trainset with the original testset and the second one is NBPT which uses perturbed trainset and perturbed testset.

6.7.1 Metrics

VD has been used as the perturbation metric while AUC and Misrate are used for measuring utility.

6.7.2 Experiment 1: Support Vector Machines

In our first experiment, we observed the performance of our method with different datasets compared with the TSVD based distortion. For all the experiments shown in Figure 6.1 to Figure 6.9 we have performed NBP and NBPT on the same run determined by fixed r. Hence, the results in the same row for NBP an NBPT are comparable.

- **MLP:**

  From Tables 6.1, 6.2, and 6.3, we can see that both versions of NAP, i.e., NBP and NBPT outperform TSVD. Some of the rows lack data for TSVD because the experiments could not produce more distortion than the last row. Experiments show that TSVD does not produce the same level of classification accuracy even with lower distortion.

- **RBF:**

  Similar results can be drawn from Tables 6.4, 6.5, 6.6. NAP based method out performs TSVD.

- **Combined:**

  Figures 6.7, 6.8, 6.9 point to similar conclusions as the above results.

6.7.3 Experiment 2: Decision Tree

Our first experiment dealt with SVM. Although our result seems to favor our method when used with SVM, it is equally important that the techniques generalize to other techniques. In this section we present our result when NAP was used with decision tree.

In all of these experiments, we observed that it is hard to get higher VD between the original set and distorted set, even when the rank of the distorted matrix is

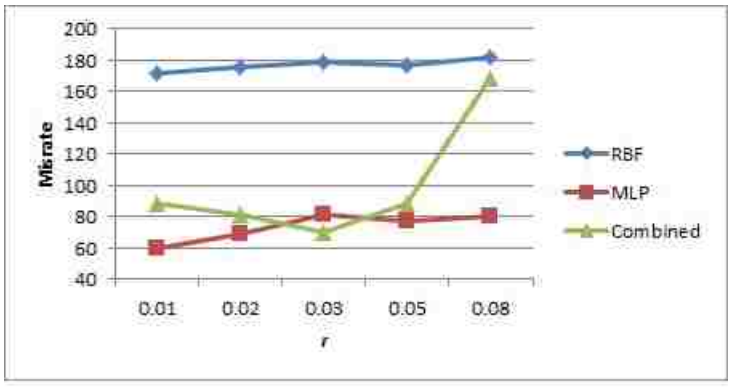


Figure 6.9: Spambase Dataset

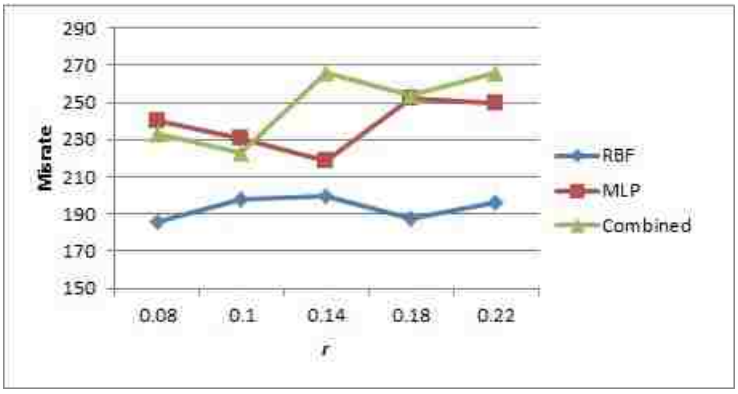


Figure 6.10: Wine Quality Dataset

Figure 6.11: Experiment 3: Effect of r on utility

reduced to 1, which is why not all the rows for SVD distortion are used in the tables presented. Utility of the data is greatly reduced by lowering the rank.

6.7.4 Experiment 3: Effect of r on utility

Our final experiment was to see the effect of r on the distortion and utility. As explained as *Property* 2 in earlier section, higher r should lead to greater amount of distortion.

From the Figure 6.11, we can see that as the value of r is increased it leads to the lower level of utility as the level of data distortion increases.

6.8 Conclusion

This chapter answered the important question which was unanswered in the previous chapter. Although correlation is a strong statistical attribute for linearly separable data, non-linearly separable data needs more than the correlation between the input and output. In this chapter, we proposed a new perturbation solution for non-linear data. NAP based methods take neighborhood into account, which gives rise to statistical property of covariance being the same in a close neighborhood. The smaller the size of the neighborhood, the lesser is the distortion, leading to preserving the covariance of the data. Although we randomly perturb the data in the close neighborhood, having several neighborhoods makes it difficult to filter out the random noise. Our theoretical analysis of NAP is backed up by our experimental results. Spambase dataset, Magic Gamma dataset, and Wine Quality dataset were used for the experiments. We observed results for SVM with different kernel functions and decision tree.

Table 6.1: Spambase Dataset with MLP

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.781286	139	0	0.651667	139	0.651667	139	0.000030	0.779398	105
0.825212	81	0.446811	0.825212	43	0.850233	68	0.001319	0.746266	122
0.749505	111	0.881611	0.764022	50	0.884358	49	0.004710	0.685063	133
0.779561	99	0.646821	0.876916	54	0.884074	51	0.007475	0.599048	155
0.791218	94	0.570660	0.849799	63	0.859423	60			
0.816826	82	0.514222	0.573370	157	0.584239	153			

Table 6.2: Wine Quality Dataset with MLP

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.593117	262	0.081136	0.592490	273	0.592490	273	0.000199	0.564750	278
0.548174	285	0.465067	0.589164	254	0.670710	209	0.001303	0.557500	288
0.562595	283	0.448604	0.605005	253	0.686844	202	0.002965	0.574250	295
0.562718	280	0.450574	0.638490	234	0.734228	175	0.009959	0.494500	318

Table 6.3: Magic Gamma Dataset with MLP

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.596279	752	0.232444	0.607011	699	0.609428	693	0.000086	0.500000	693
0.593060	745	0.249256	0.601548	738	0.595265	756	0.000788	0.606791	714
0.599925	663	0.265589	0.589049	762	0.587335	767	0.003250	0.631179	728
0.627379	628	0.285163	0.603227	739	0.592123	763	0.045718	0.712023	604
0.607836	709	0.306246	0.602467	742	0.608679	730	0.155736	0.528378	952
0.592003	763	0.313825	0.507037	690	0.508468	688			

Table 6.4: Spambase Dataset with RBF

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.888073	70	0	0.794402	70	0.794402	70	0.000030	0.791867	74
0.886262	74	0.420113	0.761303	95	0.734707	105	0.000563	0.640430	126
0.877262	81	0.719925	0.665209	132	0.651085	137	0.001532	0.528308	164
0.889352	68	0.724170	0.660669	136	0.692718	114	0.001921	0.514451	168
0.900404	61	0.785419	0.731752	103	0.711110	111	0.007501	0.500000	173
0.896012	67	0.680908	0.696578	112	0.681146	117			

Table 6.5: Wine Quality Dataset with RBF

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.789228	127	0.079080	0.797872	131	0.797872	131	0.000196	0.781984	133
0.768966	133	0.448297	0.690260	157	0.805885	107	0.000837	0.738721	156
0.788146	125	0.449604	0.661376	182	0.784886	125	0.001742	0.615226	213
0.814388	112	0.451724	0.647207	195	0.820074	104	0.009909	0.525004	385
0.744884	156	0.462940	0.604974	218	0.793176	124			
0.799124	123	0.458567	0.651528	192	0.838044	93			

Table 6.6: Magic Gamma Dataset with RBF

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.851478	258	0.231465	0.835638	297	0.836111	295	0.000086	0.847221	264
0.855946	250	0.250442	0.837432	298	0.841315	291	0.000787	0.718858	639
0.846397	260	0.264955	0.850593	286	0.847828	292	0.003240	0.746416	563
0.837028	283	0.284462	0.828058	319	0.828055	314			
0.840805	279	0.304470	0.819007	352	0.815908	351			
0.850443	267	0.315481	0.832684	323	0.838066	308			

Table 6.7: Spambase Dataset with Combined

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.747350	139	0	0.645127	139	0.645127	139	0.000394	0.798293	93
0.787659	98	0.405186	0.892121	47	0.890939	48	0.000692	0.791199	96
0.808081	86	0.587512	0.872310	54	0.880519	50	0.001514	0.612381	153
0.797510	93	0.806440	0.901399	42	0.886982	51	0.002166	0.710699	134
0.807094	105	0.534549	0.847301	65	0.853296	63	0.003546	0.740332	125
0.750432	113	0.675638	0.886014	45	0.865622	60	0.007467	0.536386	180

Table 6.8: Wine Quality Dataset with Combined

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.576855	273	0.077351	0.586729	270	0.586729	270	0.000199	0.614176	249
0.620362	239	0.444629	0.628723	233	0.746330	167	0.000839	0.621969	241
0.608138	252	0.458995	0.598108	257	0.699011	195	0.001750	0.561933	287
0.601967	257	0.441793	0.621596	238	0.733534	172	0.009937	0.484927	367
0.596469	257	0.453082	0.639399	241	0.732869	180			
0.612485	239	0.451707	0.628934	220	0.739493	161			

Table 6.9: Magic Gamma Dataset with Combined

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.629692	676	0.227920	0.634924	683	0.637584	679	0.000086	0.640687	682
0.637964	661	0.247002	0.627202	703	0.629297	697	0.003251	0.661015	701
0.655962	628	0.269103	0.656979	644	0.663279	631	0.098314	0.517695	955
0.630025	681	0.285756	0.630034	701	0.635590	689			
0.627668	683	0.304043	0.617233	724	0.619851	721			
0.634236	659	0.316028	0.635097	687	0.635377	679			

Table 6.10: Spam Dataset with Decision Tree

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.935963	59	0.000000	0.935963	59	0.935963	59	0.000031	0.929661	68
0.930936	62	0.393708	0.904747	87	0.977008	24	0.001449	0.852224	129
0.953801	49	0.796267	0.892598	106	0.891275	107	0.002669	0.777113	201
0.944959	52	0.567530	0.893437	104	0.923507	74	0.004846	0.504344	362
0.945256	69	0.619634	0.899921	101	0.901665	98			
0.933331	60	0.816122	0.880028	96	0.888034	92			

Table 6.11: Wine Quality Dataset with Decision Tree

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.834661	215	0.077052	0.825748	223	0.820128	229	0.000198	0.718424	248
0.786629	237	0.450207	0.708092	243	0.821195	168	0.000317	0.712679	258
0.786032	234	0.453113	0.671726	294	0.825911	196	0.001299	0.664304	313
0.768266	212	0.454185	0.684239	264	0.805766	183	0.009946	0.496757	484

Table 6.12: Magic Gamma Dataset with Decision Tree

Original Dataset		NBP			NBPT		SVD		
AUC	Misrate	VD	AUC	Misrate	AUC	Misrate	VD	AUC	Misrate
0.836687	491	0.233647	0.817467	564	0.815711	563	0.000087	0.828251	521
0.843696	496	0.249325	0.832941	517	0.827079	524	0.003247	0.790130	643
0.837125	481	0.269234	0.818710	557	0.826789	551	0.045506	0.691181	854
0.838033	502	0.288334	0.799300	589	0.797284	595	0.155401	0.532058	1352
0.833156	482	0.311757	0.819998	571	0.816631	570			

Chapter 7 Data Distortion Measurement

7.1 Introduction

Measurements are a major part of any research. It is important that the measurements performed have rationale. There is no standard measuremeant that can be used to measure the distortion until now. One of the fundamental difficulties is quantifying the amount of information concealed intentionally. In many cases, method of choice depends on the objective that we are trying to fulfill. In this chapter, we look into some metrics that have been used for the experiments and propose three novel measurement techniques that address some of the limitations with the measurement we have been using. Some of the frequently used data distortion metrics are presented in Table 7.1.

Table 7.1: Common Data Perturbation Metrics

Metric Formula	Parameter Description
$VD = \frac{\|A-\tilde{A}\|_F}{\|A\|_F}$	where $A \in \mathbb{R}^{n \times m}$
$RP = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\|Rank_j^i - \tilde{Rank}_j^i\|}{n \times m}$	$\tilde{Rank}_j^i$ is the rank for perturbed data
$RM = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} Rk_j^i}{n \times m}$	$Rk_j^i = \begin{cases} 1 & \text{if } Rank_j^i = \tilde{Rank}_j^i \\ 0 & otherwise \end{cases}$

In our previous work, we have used VD along with RP and RM for measuring the distance/difference between the original matrix and the distorted matrix. These methods present intuitive ways of measuring the difference between two matrices. However, these simple techniques have some serious problems associated with them. Some of the motivating reasons for our current research are as follows:

- It just takes values into consideration not the information that is available in the data. For instance, a translated distortion can have much larger VD but

Table 7.2: Dataset with name, height and salary

Name	Height (cm)	Salary($)
John	172	45,000
Tim	190	55,000
Jason	185	60,000

the actual difference between the matrices could be very little.

For a dataset A, and the translation matrix P.

$$A = \begin{bmatrix} 4.6650 & 2.4613 & 2.2746 & 2.3070 \\ 1.2332 & 2.4494 & 0.5590 & 2.8891 \\ 4.8532 & 2.3264 & 4.0698 & 2.0717 \\ 1.3951 & 4.0674 & 4.6456 & 2.0602 \end{bmatrix} \qquad P = \begin{bmatrix} 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} 104.6650 & 102.4613 & 102.2746 & 102.3070 \\ 101.2332 & 102.4494 & 100.5590 & 102.8891 \\ 104.8532 & 102.3264 & 104.0698 & 102.0717 \\ 101.3951 & 104.0674 & 104.6456 & 102.0602 \end{bmatrix}$$

Although VD between A and $\tilde{A}$ is large enough, it does not translate into the same level of privacy preservation. VD will keep increasing as long as the value of the elements of P grows.

- It is hard to explain the significance of the value VD. A value of 1 or a value of 0.0085 is hard to interpret. An even more difficult problem is when data have different units in each of the attributes as in the table below. As can be seen from Table 7.2, distortion of 1000 might not be that huge for the salary column, but it is extremely big for the height column.

- It fails to address the dissimilarity between the dataset as a whole because most of the data mining techniques are based on the principle of finding patterns in

the data and just taking the value does not make for a good metric.

The metrics like RP and RM also have their own problems. They just take the rank of the element in the row. It can be further illustrated using the following example.

Table 7.3: Perturbed dataset with name, height and salary

Name	Height (cm)	Salary($)
John	160	47,231
Tim	200	53,222
Jason	195	65,452

Although the Tables 7.2 and 7.3 are different, use of RP and RM would suggest that there is no distortion at all. So, this is our attempt to propose three novel methods which can address these issues we discussed before. The properties we focus on are mainly "geometric"; then we show these metrics behave in an intuitive and desirable way on data that are related by operations like translation, rotation, and scaling.

7.2 Proposed Techniques

The following sections introduce the techniques for measuring data distortion and also discuss their individual properties:

7.2.1 Correlation Measure

Let A be the original matrix and $\tilde{A}$ the distorted matrix. A_i represents the ith column of matrix A and $\tilde{A}_i$ represents the ith column of matrix $\tilde{A}$. The Correlation Measure (CM) calculates the average correlation between the attributes in the original matrix with those in the distorted matrix.

$$CM = \frac{\sum(Corr(A_i, \tilde{A}_i))}{n} \tag{7.1}$$

CM would be in the range of [-1, 1]. Larger distortion should give CM that tends towards 0.

Properties of CM:

- Property 1: Translation invariance of CM

$$CM(A, \tilde{A}) = 1 \text{ such that } \tilde{A} = A + b,\ b \text{ is the translation factor.} \tag{7.2}$$

Proof. Let x and y be random variables. Let z be any constant number. Here z is 'noise' that will contribute to $y' = y + z$.

Let $Cov(x, z)$ be the covariance between x and z then, we notice that $Cov(x, z) = 0$ and

$$\begin{aligned} Cov(x, y') &= Cov(x, y) + Cov(x, z) \\ &= Cov(x, y) \end{aligned} \tag{7.3}$$

since z is independent of x. If variance of y is denoted by $Var(y)$, then $Var(y') = Var(y) + Var(z) = Var(y)$ as $Var(z) = 0$. We conclude that

$$\begin{aligned} |Corr(x, y')| &= \left| \frac{Cov(x, y')}{\sqrt{Var(x)Var(y')}} \right| \\ &= \left| \frac{Cov(x, y)}{\sqrt{Var(x)Var(y)}} \right| \\ &= |Corr(x, y)| \end{aligned} \tag{7.4}$$

□

- Property 2: Addition of any zero-expectation, independent noise of finite variance will decrease the correlation between two variables x and y. $Corr(x, y) \leq Corr(x, y')$, where $e = y' - y$ and is independent of x and y with $\mu_e = E(e) = 0$, then $\mu_{y'} = E(y') = E(y) = \mu_y$.

Proof.

$$Corr(x, y') = \frac{E((x - \mu_x)(y' - \mu_y))}{\sigma_x \sigma_{y'}} \tag{7.5}$$

$$= \frac{E((x - \mu_x)(y - \mu_y)) + E((x - \mu_x)e)}{\sigma_x \sigma'_y} \tag{7.6}$$

$$= Corr(x, y)\frac{\sigma_y}{\sigma_{y'}} \tag{7.7}$$

$E((x - \mu_x)e) = E(x - \mu_x)E(e) = 0$ since x and e are independent.

Now, $\sigma_{y'} = \sqrt{\sigma_y^2 + \sigma_e^2}$, again by independence, so:

$$Corr(x, y') = Corr(x, y)\frac{1}{\sqrt{1 + \left(\frac{\sigma_e}{\sigma_y}\right)^2}} \tag{7.8}$$

We conclude that the addition of any zero-expectation, independent noise of finite variance will diminish the correlation. □

7.2.2 Canonical Correlation Analysis

The Canonical Correlation Analysis (CCA) measures the linear relationship between two multi-dimensional variables. More than one canonical correlation will be found each corresponding to a different set of basis vectors/canonical variates. Correlations between successively extracted canonical variates become gradually small. If we have two sets of variables $x_1, x_2, ..., x_n$ and $y_1, y_2, ..., y_m$ and there are correlations among the variables, then canonical correlation analysis will enable us to find linear combinations of the xs and the ys that have maximum correlation with each other. Then one seeks vectors maximizing the same correlation subject to the constraint that they are to be uncorrelated with the first pair of canonical variables.

Advantages of using CCA are:

- CCA is not dependent on the coordinate system of variables.
- CCA finds direction that yields maximum correlations.

Let $X : n \times p1$ and $Y : n \times p2$ denote centered and standardized data matrices. The aim is to construct $w = Xa_1$ and $z = Ya_2$ so that the correlation between w and z is maximal. The vectors w and z are the canonical variates. These canonical variates are standardized: $w^T w = z^T z = 1$: The vectors a_1 and a_2 are often referred to as canonical weights. Additional variates may be constructed that are orthogonal with respect to the previous ones. Then: $W = XA_1$, $Z = YA_2$, and $W^T W = Z^T Z = I$. Canonical variates can be obtained by solving singular value decomposition:

$$R_{11}^{-\frac{1}{2}} R_{12} R_{22}^{-\frac{1}{2}} = U \Lambda V^T \tag{7.9}$$

where, R_{11} denotes the correlation matrix for the first set of variables: $R_{11} = X^T X$, R_{22} is the correlation matrix for the second set: $R_{22} = Y^T Y$; and R_{12} gives the between sets correlation matrix: $R_{12} = X^T Y$. The canonical weights are calculated as

$$A_1 = R_{11}^{-\frac{1}{2}} U \text{ and } A_2 = R_{22}^{-\frac{1}{2}} V$$

so that

$$W^T W = A_1^T R_{11} A_1 = U^T U = I$$
$$Z^T Z = A_2^T R_{22} A_2 = V^T V = I$$

and

$$W^T Z = Z^T W = \Lambda$$

If $Y = X$, i.e., the column vectors are identical then the correlation between the attributes will always be 1. If they differ, then CCA will decrease gradually. The smaller the correlation, the lesser the similarity is between the two vectors. For our distortion metric, we use A in place of X and $\tilde{A}$ instead of Y. Mathematically, it can be represented as

$$CCA = \frac{\sum C_{corr}(A, \tilde{A})_i}{k} \tag{7.10}$$

where k is the number of attributes in the dataset. Similar to the earlier metrics, we can prove the following properties.

Properties of CCA:

- Property 1: Translation invariance of CCA

$$CCA(A, \tilde{A}) = 1 \text{ such that } \tilde{A} = A + b, \ b \text{ is the translation factor.} \quad (7.11)$$

Proof. As we can see from Equation(7.9), R_{11}, R_{22} and R_{12} would not change as correlation is independent of translation, resulting in the same result as the original dataset. □

- Property 2:

Another advantage of CCA is that it is independent of the units.

7.2.3 KNN Join Measure

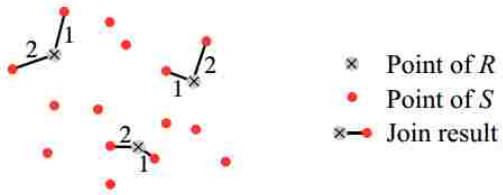


Figure 7.1: kNN Join

Basic principle behind kNN join is to find for each of the item in R its neighboring item from S. As we can treat the set R as the original dataset and dataset S as the distorted dataset, which means that we should be able to find the general structure. The idea can be used for measuring the distortion metrics; we call it kNN distortion metric.

The method works when the data are distorted in the same range, but what happens if the data are translated or rotated? In such a case, the approach fails to correctly measure the distortion in the data. The issue can be addressed if we compare the cluster to itself. The idea is to measure the similarity in the cluster structure and the cluster members between the items in two clusters. As in the figure, comparing cluster similarity will reveal the distortion level.

The kNN Join Operation

Yu et al. defined kNN join in [71] as, " Given a query point p, an integer k and a set S, the kNN query returns the set, $kNN(p) \subset S$, that contains k points from S and satisfies: $\forall q \in kNN(p), \forall q' \in S - kNN(p) : ||p - q|| \leq ||p - q'||$".
Comparison is performed in the following steps:

- Pick an item at random from the dataset, remove it from the original dataset.
- Select k nearest neighbors to the item from step 1, remove all of them from the original dataset.
- Repeat the process until none of the items remain.

$$knnM = \frac{\sum Rank(A)_i^c - Rank(\tilde{A})_i^c}{n} \tag{7.12}$$

where,

- k is the total number of clusters.
- $Rank(A)_i^c$ represents the rank of items that are neighbors of the randomly selected item in the original dataset A. Ranking is based on the distance of the items from the randomly selected item.
- $Rank(\tilde{A})_i^c$ represents the rank of items that are neighbors of the same item in the distorted dataset $\tilde{A}$.

The number will represent change in item's rank per cluster. Higher numbers of knnM indicate the higher distortion level. Similar properties to the above can be provided for this method.

- Property 1: knnM is invariant to uniform orthonormal rotation expressed as

$$knnM(A, \tilde{A}) = 0 \tag{7.13}$$

such that $\tilde{A} = QA$, where Q is the orthonormal random rotation matrix.

Proof. Let u and v be two vectors. Using the properties of orthogonal matrix, we can show that

$$||Qu||^2 = \langle Qu, Qu \rangle = (Qu)^T Qu = u^T Q^T Qu = u^T u = ||u||^2 \tag{7.14}$$

Similarly, the angle θ between Qu and Qv can be given as

$$cos\theta = \frac{\langle Qu, Qv \rangle}{||u||||v||} = \frac{(Qu)^T Qv}{||u||||v||} = \frac{u^T Q^T Qv}{||u||||v||} = \frac{u^T v}{||u||||v||} = \frac{\langle u, v \rangle}{||u||||v||} \tag{7.15}$$

From the Equations(7.14) and (7.15), we can see that the rotation preserves the angle and length between the vectors. This implies that the rank of the subject should remain unchanged. □

KnnM lacks from not being invariant to the scale, as the change of scale of two different attributes can result into different clusters.

7.3 Experimental Setup

Our experimental setup tried to observe the general behavior of our measure, as well as the properties we have mentioned in the earlier sections. In the Figure 7.6 and Table 7.5, $knnM(n)$ represents the kNN join based measure with n representing the size of the cluster.

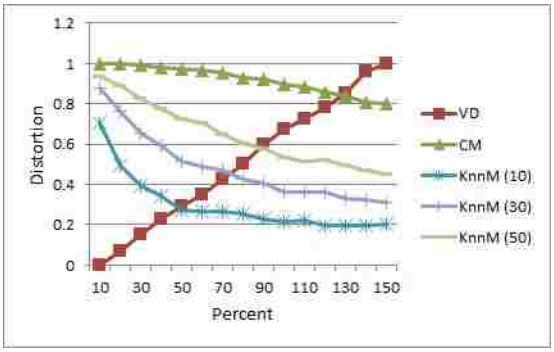


Figure 7.2: IRIS Dataset

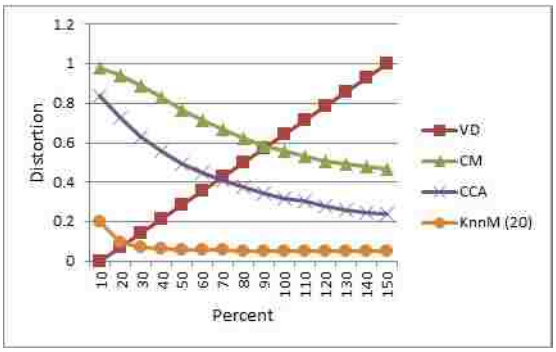


Figure 7.3: Magic Gamma Dataset

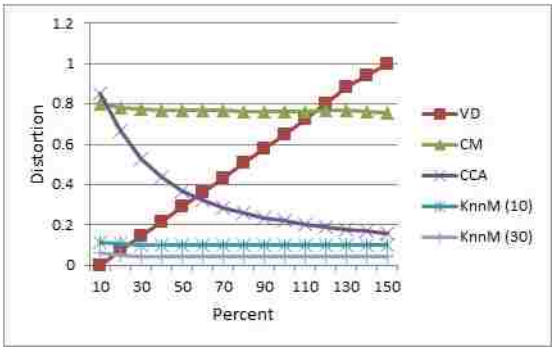


Figure 7.4: Spambase Dataset

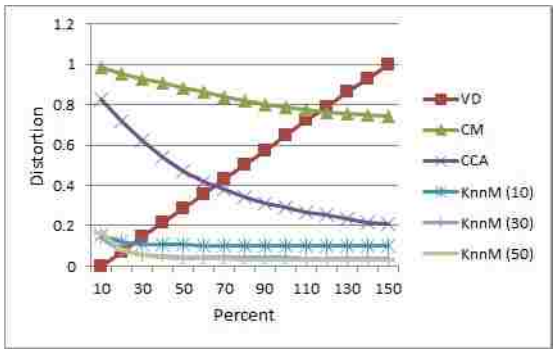


Figure 7.5: Wine Quality Dataset

Figure 7.6: Experiment 1

7.3.1 Experiment 1: Metrics behavior

As can be seen in Figure 7.6, as the percentage of distortion increases for each of the datasets the new measure starts decreasing in value. It can be noticed that the change in the value is not as constant as in the case of VD. This phenomenon leads to the fact that continually increasing the distortion does not make the dataset more private.

7.3.2 Experiment 2: Unit Independence

The second experiment dealt with observing how the new measures behaved while measurement was performed with datasets with different units. Results are presented

in Table 7.4.

Table 7.4: Experiment 2: Unit Independence

Dataset	**CM**	**CCA**
IRIS	1.00	1.00
Wine Quality	1.00	1.00
Spambase	1.00	1.00
Magic Gamma	1.00	1.00

The value of 1 in the observation shows that the original dataset has the highest degree of similarity with the perturbed dataset. This indicates the fact that these measures are invariant towards the scale used. As mentioned earlier, knnM is dependent on the scale used, hence the measure is missing from the table.

7.3.3 Experiment 3: Rotation Invariance

In this section, we present our observation when datasets were perturbed using orthogonal random rotation. We have used CCA to measure the perturbation level. The Neighborhood sizes were 5, 10, and 15. The number in the Table 7.5 show the similarity of the neighborhood of the perturbed dataset with that of the original dataset. The closer the numbers are to the size of the neighborhood, the more

Table 7.5: Experiment 3: Rotation Invariance

Dataset	**KnnM(5)**	**KnnM(10)**	**KnnM(15)**
IRIS	4.833333	9.36000	13.72340
Wine Quality	4.994767	9.982300	14.960135
Spambase	4.996522	9.980656	14.953054
WBC	4.952548	9.813708	14.597540

similar the data are. From Table 7.5, we can see that the measurements are not exactly equal to the neighborhood size. This can be attributed to the fact that we have used Gram-Schmidt orthogonalization to produce the orthonormal matrix and in real world arithmetic, this orthogonality is not perfect, and computations show a difference in the resulting matrix.

7.4 Conclusion

As seen from the theoritical properties and the experimental results, CM, CCA, and KnnM have better properties in terms of their usefulness as distortion measures. They are able to address some of the issues that plague measures that were used in our research earlier. The proposed methods aim at measuring the similarity in the overall structure of the datasets rather than just on the data values as separate entity. The CM measure separates each of the attributes and calculates the average correlation between the attributes.

One immediate avenue of future work would be to consider the information entropy as the distortion measure. There are several papers that deal with probability distribution of random variable X, but in our case we need to consider not just a variable but the multiple variables together. We may consider the Von Neumann entropy of a matrix, which deals with the entropies of the eigenvalues.

Chapter 8 Conclusions and Future Directions

In this dissertation, I have proposed several distortion techniques. My efforts have been on both Data Value Hiding and Data Pattern Hiding. The first part of my work implemented matrix factorization for clustering applications while the second part used statistical analysis for classification applications. In all of my work, I have tried to focus on context by defining what data mining techniques are going to be applied in the perturbed dataset. Having the knowledge gives the ability to tailor methods which leads to increase utility of the data. Generic methods like the random perturbation do suffer from low utility.

This NMF generalization provides greater insight into the data patterns and presents an opportunity to develop new algorithms to discover inherent data patterns by imposing suitable constraints. Constraint based NMFs are a relatively new and unexplored field, especially in privacy preserving data mining. I introduced additional constraints to preserve the privacy. The two of the constraints I defined are clustering constraint and compressing constraint. Being able to define constraints on the objective function helped achieve the privacy preserving factorization. Since our method is based primarily on the multiplicative update, it inherits the properties of multiplicative NMF. The desirable properties include simplicity and ease with which we can define the additional constraints. Although Multiplicative updates are inefficient compared to other methods, we can improve it by using better initialization of the matrices H and W. I provided the computational complexity for the proposed methods.

In addition to NMF, I proposed methods based on statistical analysis for linear and nonlinear classifications. I studied the relation between the independent and dependent variables for the linear classification problem. In my research, I studied

method to maintain the relation between the independent and dependent variables which is important for accurate classification. I studied how the proposed method can give rise to the problem of colinearity and the techniques that can be used to minimize the effect. The study included why the proposed method cannot be applied to the nonlinear classifiers. I proposed alternative solution for the problem. I compared the methods with different standard perturbation methods.

The final piece of my dissertation dealt with perturbation measurement. There is no standard measurement technique for distortion. Commonly used measurement techniques like VD, RM, and RP do not always represent the actual distortion in the data. I proposed three different measurement techniques that have more desirable properties than the methods mentioned earlier. Properties like invariant to translation, rotation, and scaling make our methods more reflective of the actual change to the data.

There are several directions which I have in my future plan. I am interested in implementing NMF as a technique to protect privacy in social network. The following sections briefly explain my idea and the different directions that I would like to pursue.

8.1 Using Constrained NMF for Privacy Preserving Data Mining in Social Network

Current efforts on PPDM for Social Networks have been on node de-identification and link protection [5, 13, 26, 35, 44]. Work has been done on anonymizing social network data based on grouping the entities into classes, and masking the mapping between entities and the nodes that represent them in the anonymized graph [6]. In [72], Zou et al. proposed a framework called k-automorphism to protect against multiple structural attacks. They also looked at the dynamic release of data.

The Figure 8.1 depicts a simple scenario where there is a network between individuals. Each individual connects to some other people with some certain weights.

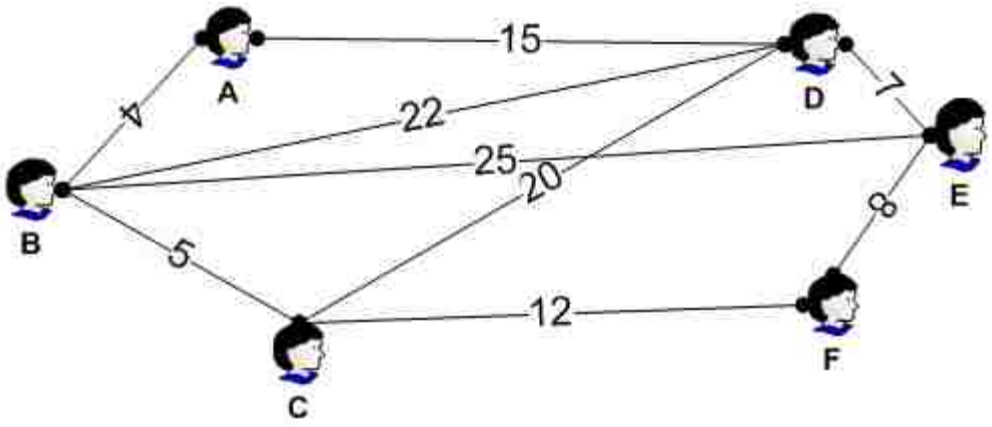


Figure 8.1: Social Network

Weights determines how far away or dissimilar they are to their connecting nodes. From the graph theory, there are ways to represent graphs in matrix forms; some common ones include: distance matrix, adjacency matrix, and incidence matrix. The above network as a distance matrix is given as

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix} \qquad \begin{bmatrix} A & B & C & D & E & F \end{bmatrix}$$

$$\begin{bmatrix} 0 & 4 & 9 & 15 & 22 & 30 \\ 4 & 0 & 5 & 19 & 25 & 17 \\ 9 & 5 & 0 & 20 & 20 & 12 \\ 15 & 19 & 20 & 0 & 7 & 15 \\ 22 & 25 & 20 & 7 & 0 & 8 \\ 30 & 17 & 12 & 15 & 8 & 0 \end{bmatrix}$$

This flexibility to represent graph by matrix makes it an ideal candidate for matrix factorization. I plan to use constraint based NMF for protecting the weights between the nodes. Application of constraint to preserve the monotonicity of the distance between the nodes is an interesting problem. We can further use relations like the hobby, interest, and field of work to explicitly define the rules for perturbation.

8.2 Efficient Computation of Constrained NMF

NMF is computationally expensive and adding constraints to NMF adds complexity to its computation. For the NMF to be more useful we need techniques that can enhance the efficiency of the method. There are two main ways: Distributed Computing and Incremental NMF. I plan to study Distributed Computing. There have been efforts in the past to compute NMF using distributed computing platforms like MPI and Hadoop. There has been no implementation of the Constrained NMF. The challenge would be reducing the NMF process into separable sub-tasks that can later be merged for the final solution. Consideration should be taken for the cost it can take for the communication between different nodes during computation.

8.3 Multi-party Computation of Proposed Techniques

As I have taken context as the primary focus in our previous methods, the effects are unknown when the data components are from different partners, and different partners have used different data distortion methods to preprocess their datasets for privacy-preserving purposes. I plan to study the properties that would or would not make the collaborative analysis difficult.

Bibliography

[1] Charu C Aggarwal and S Yu Philip. *A condensation approach to privacy preserving data mining*. Springer, 2004.

[2] Dakshi Agrawal and Charu 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*, pages 247–255. ACM, 2001.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.

[4] Mikhail J Atallah and Wenliang Du. Secure multi-party computational geometry. In *Algorithms and Data Structures*, pages 165–179. Springer, 2001.

[5] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190. ACM, 2007.

[6] Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. Class-based graph anonymization for social network data. *Proceedings of the VLDB Endowment*, 2(1):766–777, 2009.

[7] Isaac Cano, Susana Ladra, and Vicenç Torra. Evaluation of information loss for privacy preserving data mining through comparison of fuzzy partitions. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.

[8] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems*, pages 289–296, 2008.

[9] Keke Chen and Ling Liu. Privacy preserving data classification with rotation perturbation. In *Fifth IEEE International Conference on Data Mining*, pages 589–592. IEEE, 2005.

[10] Moody T Chu. On the statistical meaning of truncated singular value decomposition. *Citeseer*, 2001.

[11] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y Zhu. Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations Newsletter*, 4(2):28–34, 2002.

[12] Chris Clifton and Don Marks. Security and privacy implications of data mining. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 15–19. Citeseer, 1996.

[13] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. Anonymizing bipartite graph data using safe groupings. *Proceedings of the VLDB Endowment*, 1(1):833–844, 2008.

[14] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[15] Chris HQ Ding, Xiaofeng He, and Horst D Simon. On the equivalence of non-negative matrix factorization and spectral clustering. In *SDM*, volume 5, pages 606–610, 2005.

[16] Wenliang Du and Mikhail J Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *Proceedings of the 2001 workshop on New security paradigms*, pages 13–22. ACM, 2001.

[17] Alexandre Evfimievski. Randomization in privacy preserving data mining. *ACM Sigkdd Explorations Newsletter*, 4(2):43–48, 2002.

[18] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222. ACM, 2003.

[19] Andrew Frank and Arthur Asuncion. Uci machine learning repository, 2010. *URL http://archive. ics. uci. edu/ml*, 15:22, 2011.

[20] Nikolaos M Freris, Michail Vlachos, and Deepak S Turaga. Cluster-aware compression with provable k-means preservation. In *SDM*, pages 82–93, 2012.

[21] Jerome H Friedman. Data mining and statistics: What's the connection? *Computing Science and Statistics*, 29(1):3–9, 1998.

[22] Jing Gao and Jun Zhang. Sparsification strategies in latent semantic indexing. In *Proceedings of the 2003 Text Mining Workshop*, pages 93–103, 2003.

[23] Bobi Gilburd, Assaf Schuster, and Ran Wolff. k-ttp: a new privacy model for large-scale distributed environments. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 563–568. ACM, 2004.

[24] Shanti Gomatam, Alan F Karr, and Ashish P Sanil. Data swapping as a decision problem. *Journal of Official Statistics*, 21(4):635–655, 2005.

[25] Edward F Gonzalez and Yin Zhang. Accelerating the lee-seung algorithm for non-negative matrix factorization. *Dept. Comput. & Appl. Math., Rice Univ., Houston, TX, Tech. Rep. TR-05-02*, 2005.

[26] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment*, 1(1):102–114, 2008.

[27] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[28] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the tenth annual symposium on Computational geometry*, pages 332–339. ACM, 1994.

[29] Murat Kantarcıoglu and Jaideep Vaidya. Privacy preserving naive bayes classifier for horizontally partitioned data. In *IEEE ICDM workshop on privacy preserving data mining*, pages 3–9, 2003.

[30] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 99–106. IEEE, 2003.

[31] Andrew H Karp. Using logistic regression to predict customer retention. *The Northeast SAS User Group (NESUG)*, 1998.

[32] Jay J Kim. A method for limiting disclosure in microdata based on random noise and transformation. In *Proceedings of the section on survey research methods*, pages 303–308, 1986.

[33] DG Kleimbaum and M Klein. Logistic regression. *New York, NY, Springer-Verlag*, page 237, 1994.

[34] Ethan A Kolek and Daniel Saunders. Online disclosure: An empirical examination of undergraduate facebook profiles. *NASPA Journal*, 45(1):1–25, 2008.

[35] Aleksandra Korolova, Rajeev Motwani, Shubha U Nabar, and Ying Xu. Link privacy in social networks. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 289–298. ACM, 2008.

[36] Eric Langford, Neil Schwertman, and Margaret Owens. Is the property of being positively correlated transitive? *The American Statistician*, 55(4):322–325, 2001.

[37] A Lenhart and M Madden. Teens, privacy and online social networks. *Washington DC: Pew Internet and American Life Project*, 2007.

[38] Feifei Li, Jimeng Sun, Spiros Papadimitriou, George A Mihaila, and Ioana Stanoi. Hiding in the crowd: Privacy preservation on evolving streams through correlation tracking. In *ICDE 2007. IEEE 23rd International Conference on Data Engineering, 2007.*, pages 686–695. IEEE, 2007.

[39] Hualiang Li, Tülay Adal, Wei Wang, Darren Emge, and Andrzej Cichocki. Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 48(1-2):83–97, 2007.

[40] Chih-Jen Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *Neural Networks, IEEE Transactions on*, 18(6):1589–1596, 2007.

[41] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[42] Zhenmin Lin, Jie Wang, Lian Liu, and Jun Zhang. Generalized random rotation perturbation for vertically partitioned data sets. In *CIDM'09. IEEE Symposium on Computational Intelligence and Data Mining, 2009.*, pages 159–162. IEEE, 2009.

[43] Kun Liu, Hillol Kargupta, and Jessica Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):92–106, 2006.

[44] Lian Liu, Jie Wang, Jinze Liu, and Jun Zhang. Privacy preservation in social networks with sensitive edge weights. In *SDM*, pages 954–965, 2009.

[45] Bo Long, Zhongfei Mark Zhang, Xiaoyun Wu, and Philip S Yu. Relational clustering by symmetric convex coding. In *Proceedings of the 24th international conference on Machine learning*, pages 569–576. ACM, 2007.

[46] David Lyon. Surveillance, power, and everyday life. *Handvbook of ICTs*, 2007.

[47] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[48] Kantilal Varichand Mardia, John T Kent, and John M Bibby. Multivariate analysis. 1980.

[49] M Matteucci. Clustering: An introduction. *http* : *//home.deib.polimi.it/matteucc/Clustering/tutorial_html/*, December 2009.

[50] Krishnamurty Muralidhar and Rathindra Sarathy. Security of random data perturbation methods. *ACM Transactions on Database Systems (TODS)*, 24(4):487–493, 1999.

[51] Jerome L Myers, Arnold D Well, and Robert Frederick Lorch. *Research design and statistical analysis*. Routledge, 2010.

[52] Office of the Assistant Secretary for Planning and DHHS Evaluation. *Standards for Privacy of Individually Indentifiable Health Information*, volume 65. 2000.

[53] Rafail Ostrovsky and William E Skeith III. Private searching on streaming data. In *Advances in Cryptology–CRYPTO 2005*, pages 223–240. Springer, 2005.

[54] Rupa Parameswaran and D Blough. A robust data obfuscation approach for privacy preservation of clustered data. In *Workshop Proceedings of the 2005 IEEE International Conference on Data Mining,(Houston, Texas)*, pages 18–25. Citeseer, 2005.

[55] Benny Pinkas. Cryptographic techniques for privacy-preserving data mining. *ACM SIGKDD Explorations Newsletter*, 4(2):12–19, 2002.

[56] Huseyin Polat and Wenliang Du. Svd-based collaborative filtering with privacy. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 791–795. ACM, 2005.

[57] D Seung and L Lee. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001.

[58] Latanya Sweeney. K-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[59] Botond Szatmáry, Barnabás Póczos, Julian Eggert, E Korner, and András Lorincz. Non-negative matrix factorization extended by sparse code shrinkage and weight sparsification. In *ECAI*, pages 503–507, 2002.

[60] Michael Totty. The dangers within. *The Wall Street Journal*, 247, 2006.

[61] Jaideep Vaidya and Chris 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*, pages 639–644. ACM, 2002.

[62] Jaideep Vaidya and Chris Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM, 2003.

[63] Vassilios S Verykios, Ahmed K Elmagarmid, Elisa Bertino, Yücel Saygin, and Elena Dasseni. Association rule hiding. *Knowledge and Data Engineering, IEEE Transactions on*, 16(4):434–447, 2004.

[64] Jie Wang. *Matrix Decomposition for Data Disclosure Control and Data Mining Applications*. PhD thesis, University of Kentucky, 2008.

[65] Jie Wang, Jun Zhang, Lian Liu, and Dianwei Han. Simultaneous pattern and data hiding in unsupervised learning. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 729–734. IEEE, 2007.

[66] Jie Wang, Weijun Zhong, and Jun Zhang. Nnmf-based factorization techniques for high-accuracy privacy protection on non-negative-valued datasets. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 513–517. IEEE, 2006.

[67] Jie Wang, Weijun Zhong, Jun Zhang, and Shuting Xu. Selective data distortion via structural partition and ssvd for privacy preservation. In *IKE*, pages 114–120. Citeseer, 2006.

[68] Xindong Wu. Data mining: An ai perspective. *Intelligent Informatics*, page 23, 2003.

[69] Shuting Xu, Jun Zhang, Dianwei Han, and Jie Wang. Data distortion for privacy protection in a terrorist analysis system. In *Intelligence and Security Informatics*, pages 459–464. Springer, 2005.

[70] Shuting Xu, Jun Zhang, Dianwei Han, and Jie Wang. Singular value decomposition based data distortion strategy for privacy protection. *Knowledge and Information Systems*, 10(3):383–397, 2006.

[71] Cui Yu, Rui Zhang, Yaochun Huang, and Hui Xiong. High-dimensional knn joins with incremental updates. *Geoinformatica*, 14(1):55–82, 2010.

[72] Lei Zou, Lei Chen, and M Tamer Özsu. K-automorphism: A general framework for privacy preserving network publication. *Proceedings of the VLDB Endowment*, 2(1):946–957, 2009.

Vita

Personal Data:

Name: Nirmal Thapa
Country of Birth: Nepal

Educational Background:

- **University of Kentucky, Lexington, KY**

  College of Engineering, Ph.D. Computer Science, [Aug 2009- Present]

  – Dissertation Title: Context Aware Privacy Preserving Clustering and Classification

- **Tribhuvan University, Kathmandu, Nepal**
  Institute of Engineering, B.E. Computer Engineering, [Dec 2003 - Feb 2008]

Research Interest:

- Matrix decomposition techniques and its applications
- Data visualization
- Information privacy

Experience

- **University of Kentucky, Lexington, KY**

  Teaching Assistant (Python, C++), [Aug 2009 – Present]

  – Fully responsible for conducting lectures and labs for CS115 (Introduction to Programming) and CS215 (Introduction to Program Design, Abstraction, and Problem Solving). Interacted with the students to help them learn th materials in an effective way.

- **Jaspersoft Corporation, San Francisco, CA**

  Infrastructure Intern, [May 2013 - Aug 2013]

  – Responsible for resolving different issues with clustered computing. Focus currently has been on session and cache management across different nodes in a cluster.

- **Robert Bosch Research and Technology Center, Palo Alto, CA**

  Research Intern, [May 2012 - Aug 2012]

  - Analyzed healthcare data collected from the Health Buddy System and created model to represent the data that could be used for the mining purpose but prevent users from knowing the actual data which were considered to be sensitive information. Compared its performance with benchmark methods.

- **Computer Services, Inc., Lexington, KY**

  Engineering Intern, [May 2010 - Aug 2010]

  - NuPoint RMS (Risk Management System): Worked on the administrative module of Nupoint.
  - Analyzed the existing Nupoint System and proposed ways to enhance the system for the future.

- **ITOffshore Nepal, Kathmandu, Nepal**

  Software Engineer, [Feb 2008 - Jul 2009]

  - Campagne CMS (.NET/DNN): Extension to the DNN where the portals were in different server.
  - Guide System (.NET): A web application used by multiple companies to keep track of the employees performance and growth.
  - Hesdo ERP (.NET): Created a module for generating flyers for products in .doc format from templates.

- **Milestone Systems, Kathmandu, Nepal**

  Engineering Intern, [May 2007- Nov 2007]

  - Developed Web Application for manpower agencies where applicants could log in and apply for different positions posted and keep track of the progress. It incorporated chat feature.

Publications

Referred Conferences and Journal papers

- **Constrained Nonnegative Matrix Factorization based Data Distortion Techniques: Study of Data Privacy and Utility**. Nirmal Thapa, Pengpeng Lin, Lian Liu, Jie Wang, and Jun Zhang. *In the Proceedings of the International Conference on Data Technologies and Applications (DATA), Rome, Italy, Jul 25 - 27, 2012.*

- **Constrained Nonnegative Matrix Factorization for Feature Selection**. Nirmal Thapa and Jun Zhang. *In the Proceedings of the International Conference on Data Mining (DMIN), Las Vegas, Nevada, Jul 17-21, 2012.*

- **Constrained Non-negative Matrix Factorization for Data Privacy**. Nirmal Thapa, Lian Liu, Pengpeng Lin, Jie Wang, and Jun Zhang. *In The Proceedings of the International Conference on Data Mining (DMIN), Las Vegas, Nevada, Jul 17-21, 2011.*

- **Feature Selection: A Preprocess for Data Perturbation**. Pengpeng Lin, Nirmal Thapa, Ingrid St. Omer, Lian Liu, and Jun Zhang. *IAENG International Journal of Computer Science, 2011.*

- **Sparsified SVD Based Feature Selection Approach** . Pengpeng Lin, Nirmal Thapa, Jun Zhang, Ingrid St. Omer, and Huanjing Wang. *International Journal of Information Technology and Decision Making (IJITDM), 2011.*

- **CPU Scavenging through the Swarm Intelligence of Autonomous Agents: Employing Artificial Ants**. Prajwol Kumar Nakarmi, Nirmal Thapa, and Narendra Maharjan. *Lambart Academic Publishing, 2011* [Book].

- **Correlation-Aware Data Perturbation for Logistic Regression**. Nirmal Thapa, Juergen Heit, Soundar Srinivasan, and Jun Zhang.

- **Multiclass Data Perturbation for Support Vector Machines**. Nirmal Thapa, Juergen Heit, Soundar Srinivasan, and Jun Zhang.

Paticipation and Awards

- 25th Annual EKU Symposium in Mathematical, Statistical and Computer Sciences, Apr 2011 - [**First Position**].

- 24th Annual EKU Symposium in Mathematical, Statistical and Computer Sciences, Apr 2010.

- Open Software Competition, Locus 06/07 - [**3rd Place**].

- Paper Presentation on Adaptive Character Recognition using Neural Networks at NASCOIT, organized by NCIT, 2007 - [**Best Paper Award**].

- Member (Software committee) Locus 2004.

- Member (Hardware committee) Locus 2005.

Academic Projects

- **Clara - CPU Scavenging through the swarm intelligence of autonomous agents** (*Major project*) (JAVA) Project utilized the available processor resources in a network by using the swarm of ants.

- **GoBliki** (*Database project*) (PHP, MYSQL) Web 2.0 application combining the features of blog and wiki.
- **Nat Studio** (*Minor Project*) (.NET) This tool was mainly built for intrution dectection in networks. Main components of NAT Studio included: packet analyzer, firewall and port usage dectector.
- **Total Data Analyzer**(.NET, MSSQL): A research based project for hydrological analysis of millions of data available. Neural Network was an integral part for the pattern recognition.

Academic Honors

- Student Sponsorship - The 7th International Conference on Data Mining in 2011.
- Student Travel Support - Graduate School Fellowship 2011.
- Student Travel Support - Graduate School Fellowship 2012.
- The College Fellowship Scholarship III/II and IV/II (Year/Semester)(Undergrad).
- Freeship II/II,III/I and IV/I (Year/Semester)(Undergrad).