FEATURE SELECTION FOR SUPERVISED AND UNSUPERVISED LEARNING

A Dissertation

by

XIAOPENG SUI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Tie Liu
Co-Chair of Committee,	Xiaoning Qian
Committee Members,	Aniruddha Datta
	Anxiao (Andrew) Jiang
Head of Department,	Miroslav M. Begovic

December 2018

Major Subject: Electrical Engineering

ABSTRACT

Unsupervised and semi-supervised learning are explored in convex clustering with metric learning while supervised learning is explored in a novel feature selection method. First, we evaluate the performance of convex clustering against previous clustering formulations. Moreover, we implement two metric learning schemes in convex clustering to replace the Euclidean distance used in the original convex clustering formulation. The first metric learning scheme involves using a full-rank positive definite matrix to characterize a Mahalanobis metric and the second metric learning scheme involves using a sparse compositional metric. This sparse compositional metric is a weighted sum of a set of orthonormal rank-1 basis vectors. In experimentation on both simulated data and real life data, convex clustering with metric learning, especially a sparse compositional metric, can outperform convex clustering, other methods based on convex clustering and previous popular clustering algorithms. Second, a novel feature selection method is proposed using Chow-Liu tree approximations to estimate Shannon's mutual information. In experimental analysis, this Chow-Liu tree feature selection method out performs previous feature selection method when classification accuracy is used as a performance measure.

DEDICATION

To my family.

ACKNOWLEDGMENTS

I would first like to thank my committee for guiding me through my experience at Texas A&M. In particular I would like to thank my co-advisors Dr. Tie Liu and Dr. Xiaoning Qian for supporting me and advising me every step in this journey.

I would like to thank all of the fellow graduate students and researchers that I had the joy of sharing an office, a classroom, a project, a bus ride or just a cup of coffee with. The support and friendship I experienced here at Texas A&M made me proud to be part of the Aggie family. In particular, I would like to thank my collaborator, my mentor and my friend Easton Li Xu for his exceptional guidance in my work and research.

Lastly, I want to thank my parents, my husband and my extended family, the people that had never doubted for even one moment that this day will come. To my mom and dad, thank you for standing by me through every moment of my life, and always pushing me to realize my goals. To my husband, thank you for coming into my life and for your enduring love and patience through the ups, downs and uncertainities of our life together.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Tie Liu, Professor Xiaoning Qian, and Professor Aniruddha Datta of the Department of Electrical and Computer Engineering and Professor Anxiao (Andrew) Jiang of the Department of Computer Science and Engineering.

Funding Sources

This work was made possible in part by NSF, under Grant No. 1719017, 1447235, 1547557, and 1553281.

NOMENCLATURE

ADMM	Alternating Direction Method of Multipliers
ARR	Arrythemia
CC	Convex Clustering
CCML	Convex Clustering with Metric Learning
CCSCML	Convex Clustering with Sparse Compositional Metric Learning
CLT	Chow-Liu Tree
CFS	Correlation Based Feature Selection
COIL	Columbia University Image Library
DLBCL	Diffuse Large B-Cell Lymphoma
GMM	Gaussian Mixture Model
HDR	Handwritten Digits Recognition
LDA	Linear Discriminant Analysis
mRMR	Minimum Redundancy Maximum Relevancy
MST	Maximum Spanning Tree
NB	Naive Bayesian
N-Cut	Normalized Cut
PSD	Positive Semi-Definite
RCC	Robust Convex Clustering
RF	Random Forest
SCC	Sparse Convex Clustering
SVM	Support Vector Machine

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

CONTRIBUTORS AND FUNDING SOURCES v

NOMENCLATURE vi

TABLE OF CONTENTS vii

LIST OF FIGURES ix

LIST OF TABLES xi

1. INTRODUCTION 1

    1.1 Background 1
        1.1.1 Clustering with Metric Learning 1
        1.1.2 Feature Selection 3
    1.2 Overview 4

2. CLUSTERING 6

    2.1 Introduction 6
    2.2 Problem Statement 6
    2.3 Previous Clustering Methods 6
        2.3.1 Centroid Models 7
        2.3.2 Graphical Models 8
        2.3.3 Hierarchical Models 10
        2.3.4 Distribution Models 11
    2.4 Conclusion 11

3. METRIC LEARNING 12

    3.1 Introduction 12
    3.2 Notable Works in Metric Learning 13
    3.3 Conclusion 17

4. CONVEX CLUSTERING WITH METRIC LEARNING 18

4.1 Introduction 18
4.2 Convex Clustering 19
4.3 Robust Convex Clustering 21
4.4 Convex Clustering with Full Rank Metric Learning Algorithm 22
4.4.1 Solving $\mathbf{U}$ for a Fixed $\mathbf{B}$ 22
4.4.2 Solving $\mathbf{B}$ for a Fixed $\mathbf{U}$ 27
4.4.3 Iteration between Convex Clustering and Metric Learning 28
4.5 Sparse Convex Clustering 28
4.6 Convex Clustering with Sparse Compositional Metric Learning 29
4.6.1 Structural Constraints on Metric Learning 30
4.6.2 Convex Clustering with Sparse Compositional Metric Learning 32
4.6.3 Fisher Linear Discriminant Analysis 34
4.7 Experimental Results 35
4.7.1 Synthetic data 36
4.7.2 Real-world data 40
4.8 Conclusion 44

5. FEATURE SELECTION 48

5.1 Introduction 48
5.2 Related Works 49
5.2.1 Minimum Redundancy Maximum Relevance (mRMR) 49
5.2.2 Correlation Based Feature Selection (CFS) 50
5.3 Conclusion 50

6. FEATURE SELECTION USING CHOW-LIU TREE APPROXIMATION 52

6.1 Introduction 52
6.2 Chow-Liu Tree Approximation 53
6.3 Feature Selection Algorithm 54
6.4 Experimental Results 56
6.4.1 The Data Sets 57
6.5 Conclusion 60

7. CONCLUSION 68

REFERENCES 70

LIST OF FIGURES

FIGURE Page

2.1 Example of the k-means clustering of "Iris" Data 9

2.2 Example of when normalized cut performs better than min-cut 10

4.1 Example of synthetic data before outlier features were added. 37

4.2 Gaussian GMM data: Clustering accuracy as a function of the number of outlier features. The narrow line on top of each bar indicates the standard deviation for each set of experiments. 38

4.3 Gaussian GMM data: Convergence of the clustering accuracy and the minimum value for RCC. 39

4.4 Gaussian GMM data: Convergence of the clustering accuracy and the minimum value for CCML. 40

4.5 Gaussian GMM data: Convergence of the clustering accuracy and the minimum value for CCSCML. 41

4.6 Gaussian GMM data: Cumulative running time as a function of the number of iterations. The algorithms were implemented using MATLAB R2017b on a Windows 10 PC with an Intel Core i7 2.8 GHz processor. 42

4.7 Gaussian GMM data: Intensity map and the singular values of the full rank metric $\mathbf{B}$ learned from the final iteration. 43

4.8 Real-world data sets: Intensity map and the singular values of the full rank metric $\mathbf{B}$ learned from the final iteration. 46

4.9 The ordered eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$ in the Fisher LDA. 47

6.1 Classification accuracy result of ARR data using NB classification. 58

6.2 Classification accuracy result of ARR data using SVM classification. 59

6.3 Classification accuracy result of ARR data using RF classification. 60

6.4 Classification accuracy result of HDR data using NB classification. 61

6.5 Classification accuracy result of HDR data using SVM classification. 62

6.6 Classification accuracy result of HDR data using RF classification. 63

6.7 The 20 objects of the COIL-20 data set. 64

6.8 Classification accuracy result of COIL-20 data using NB classification..... 65

6.9 Classification accuracy result of COIL-20 data using SVM classification. 66

6.10 Classification accuracy result of COIL-20 data using RF classification. 67

LIST OF TABLES

TABLE Page

4.1 The basic parameters of four real-world data sets. 40

4.2 The clustering accuracies of various clustering algorithms for four real-world data sets. 41

1. INTRODUCTION

1.1 Background

Clustering is an unsupervised machine learning algorithm that has been well researched. Clustering is the task of grouping a set of data points into clusters so that the data points in each cluster are similar to each other and the date points in different cluster are dissimilar to each other. It is considered unsupervised because no training data is required to perform the algorithm. In most clustering algorithms, the only needed input is the set of testing data and sometimes the number of clusters. By introducing metric learning to clustering algorithms, the problem then become semi-supervised. Metric learning can use partial training information or partial and preliminary clustering solutions to enhance the performance of clustering algorithms. Metric learning algorithms focus on the feature dimension of the data and has the effect of dimension reduction and feature weighing. Feature selection is a supervised learning algorithm. It uses a set of training data to train for the best subset of features that enhances clustering or classification algorithm performance accuracy. This dissertation can be divided into the following two parts: clustering with metric learning and feature selection.

1.1.1 Clustering with Metric Learning

Clustering analysis is a fundamental problem in many diverse scientific fields [1–10]. For example, clustering can be used in brain imaging where regions of the brain are clustered according to their MRI signals so that each cluster is related to certain brain functions [2]. In the economic sciences, clustering can be used in market research where consumers are clustered into groups that are similar in their shopping trends [11]. And in robotics, clustering can be used for situational awareness to track and detect objects in sensory data [12]. Many previous works have been developed in clustering, tackling the problem using various schemes. There are centroid models, such as the classical k-means method [6], graphical models, such as normalized-cut algorithm [5], hierarchical clustering [13] and distribution model clustering [14]. However, these algorithms can

be trapped at local minima, which can be sub-optimal. These traditional clustering methods take a greedy approach and also suffer from instabilities due to their nonconvex optimization formulations. Chi et al. [15] then proposed a *convex clustering* scheme that can be viewed as a convex relaxation of k-means clustering and hierarchical clustering. The convexity ensures that it achieves a global optimizer. Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a collection of N data points to be clustered, and let $\mathbf{X}$ be the data matrix for which the jth column is given by $\mathbf{x}_j$ (so each row of $\mathbf{X}$ represents a feature of the data). In [15], the convex clustering problem was formulated as the following optimization problem:

$$\underset{\mathbf{U}}{\text{Minimize}} \quad \frac{1}{2}\sum_{j=1}^{N} \|\mathbf{x}_j - \mathbf{u}_j\|_2^2 + \gamma \sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}} \|\mathbf{u}_{j_1} - \mathbf{u}_{j_2}\|_1 \tag{1.1}$$

where γ is a positive tuning constant, $w_{\{j_1,j_2\}}$ is a nonnegative weight, and the jth column $\mathbf{u}_j$ of the matrix $\mathbf{U}$ is the center of the cluster that the data point $\mathbf{x}_j$ belongs to.

Although the convex clustering problem solves many of the issues that traditional clustering algorithms had, it still uses Euclidean distance to calculate the distance between the data point and the cluster centers. Many data sets in the scientific fields have features or dimensions in the data that are noisy or irrelevant to the clustering solution. Therefore it would naturally enhance clustering performance to use a distance measure that takes feature importance into consideration. In our work we propose to add a Mahalannobis metric to the convex learning problem [16]. In an iterative procedure, the convex clustering and the metric learning problem can be both solved. The convex clustering with metric learning problem can be written as

$$\underset{\mathbf{U},\mathbf{B}}{\text{Minimize}} \quad \frac{1}{2}\sum_{j=1}^{N} (\mathbf{x}_j - \mathbf{u}_j)^T \mathbf{B} (\mathbf{x}_j - \mathbf{u}_j) + \gamma \sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}} \|\mathbf{u}_{j_1} - \mathbf{u}_{j_2}\|_1 \tag{1.2}$$

In the above formulation, the Mahalanobis metric is characterized by the matrix $\mathbf{B}$. This Mahalanobis metric is symmetric and positive semi-definite. If $\mathbf{B}$ is diagonal, then the diagonal values weigh more important features with higher coefficients and weigh less important features with lower coefficients. For a general positive semi-definite $\mathbf{B}$, this can be understood through principal component analysis. In this dissertation, we first formulate $\mathbf{B}$ to be a full rank, positive definite

matrix. With a given data set $\mathbf{X}$ and $\mathbf{U}$, the full-rank $\mathbf{B}$ can be solved in closed-form. Secondly, we formulate $\mathbf{B}$ as a sparse compositional metric. $\mathbf{B}$ is represented as a nonnegative weighted sum of s rank-1 positive semidefinite matrices:

$$\mathbf{B} = \sum_{i=1}^{s} \sigma_i \mathbf{q}_i \mathbf{q}_i^T = \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^T, \tag{1.3}$$

where $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_s)$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$. The sparse compositional metric learning also serves as a dimension reduction technique as $s < d$.

1.1.2 Feature Selection

Feature selection or variable selection is a supervised learning method. Unnecessary features in a data set decreases training speed, model interpretability, and performance in the test set. Feature selection algorithms select features from the training data by eliminating features with high percentage of missing values, highly correlated or redundant features, features with zero importance or relevance to the outcome variable, and features with low variance in any of the data set classes [17–23]. Most feature selection methods use a measure that tries to incorporate all the appropriate selection criterias and select a subset of features in a greedy fashion [24, 25].

Formally, we let $(\mathsf{X}_V, \mathsf{Y})$ be a collection of jointly distributed random variables, where $\mathsf{X}_V := (\mathsf{X}_i : i \in V)$ are the features and the outcome variable is Y. The goal of feature selection is to find a subset of features, $B \subseteq V$, of size k that are highly relevant to the outcome variable based on a give set of independent and identically drawn samples from $(\mathsf{X}_V, \mathsf{Y})$. By far, most of the previous feature selection methods rely on the heuristic of maximum relevance and minimum redundancy.

1. Minimum Redundancy Maximum Relevance (mRMR) [24]. Using Shannon's mutual information as the correlation measure between two variables X_1 and X_2, $r_{\mathsf{X}_1,\mathsf{X}_2} = I(\mathsf{X}_1, \mathsf{X}_2)$. Using $\bar{r}_{xy}(B) := \frac{1}{|B|} \sum_{i \in B} r_{\mathsf{X}_i,\mathsf{Y}}$ as the average relevance betwen the feature and the outcome and $\bar{r}_{xx}(B) := \frac{1}{|B|(|B|-1)} \sum_{(i,j) \in B^2 : i \neq j} r_{\mathsf{X}_i,\mathsf{X}_j}$ as the average redundancy between the features,

the mRMR selection criteria is:

$$\max_{X_j \in X_V - S_{m-1}} I(\mathsf{X}_i, \mathsf{Y}) - \frac{1}{m-1} \sum_{X_i \in S_{m-1}} I(\mathsf{X}_i, \mathsf{Y}). \tag{1.4}$$

2. Correlation-based Feature Selection (CFS) [25]. Using a value called symmetric uncertainty $r_{xy} = \frac{2I(\mathsf{X}_i,\mathsf{Y})}{H(X_i)+H(X_j)}$ as the correlation measure between random variables, the selection criteria for CFS is

$$\frac{|B|\bar{r}_{xy}(B)}{\sqrt{|B| + |B|(|B|-1)\bar{r}_{xx}(B)}}. \tag{1.5}$$

In this work we propose a new feature selection algorithm that instead of following the maximum relevance minimum redundancy principle, it uses a direct approximation on the Shannon mutual information $I(\mathsf{X}_B; \mathsf{Y})$ by using the well known Chow-Liu tree approximations. The algorithm is an incremental search algorithm over the entire feature set to select a subset of features greedily by solving the problem

$$\max_{i \in V \setminus B} I(\mathsf{X}_{B \cup \{i\}}; \mathsf{Y}). \tag{1.6}$$

1.2 Overview

The rest of this dissertation is as follows. In Chapter 2, the problem statement and previous works of clustering will be formally stated and explained. The types of clustering models and algorithms will be described to inspire the advantages of convex clustering. In Chapter 3, the definition of Mahalanobis metric will be stated and how it enhances the performance of clustering algorithms. Two notable works in metric learning that the work in Chapter 4 are inspired from will also be explained. In Chapter 4, the convex clustering formulation and the algorithm to solve convex clustering will be introduced. Moreover, the two works on convex clustering with full-rank metric learning and sparse compositional metric learning will be proposed. It will conclude with experimental results comparing the two proposed convex clustering with metric learning algorithms along with convex clustering and two previous works with convex clustering - robust convex clustering [26] and sparse convex clustering [27]. In Chapter 5 of the dissertation, it will

move towards supervised learning. Chapter 5 will formally introduce the feature selection problem. It will also describe two earlier works in feature selection: minimum redundancy maximum relevance [24] and correlation based feature selection [25]. In Chapter 6, we propose a novel feature selection algorithm by estimating mutual information using Chow-Liu Tree approximations. The chapter will end with experimental results comparing the novel algorithm with the two previous classic feature selection algorithms. The dissertation will conclude in Chapter 7 with some closing remarks.

2. CLUSTERING

2.1 Introduction

Clustering is a well researched machine learning technique that has many applications [1–10]. It is the grouping of various data points into groups, or clusters, thus that the data points within one cluster are similar to each other and data points in different clusters are dissimilar from each other. The goal is to divide the data set into meaningful and useful clusters. The clusters are meaningful when they capture the natural structures of the data. There are many application for clustering methods in various fields such as neuro-imaging [2], image segmentation [5], social sciences and economics [11]. The structure of the data points and the desired structure of the resulting clusters dictate the method and algorithm that is most appropriate for the application of clustering.

2.2 Problem Statement

Here we begin by defining variables in clustering and data mining. Let N denote the number of sample points in the data set and let the number of "features" be d, then let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a collection of N data points to be clustered, where $\mathbf{x}_j \in \mathbb{R}^d$ for each $j = 1, 2, \ldots, N$. Let $\mathbf{X}$ be the data matrix for which the jth column is given by $\mathbf{x}_j$ (so each row of $\mathbf{X}$ represents a feature of the data). The features of the data can be continuous or discrete variables. Some examples of the data features are color and intensity of image data, genetic markers, and grades from students in grade prediction algorithms. The problem of clustering is to group the data points into k number of clusters based on a similarity measure or distance measure based on the values of the features in the data.

2.3 Previous Clustering Methods

There has been many works in clustering - the optimization problem can be formulated in various ways and the optimization problem can be solved using various methods as well. Below is four groups of clustering methods.

2.3.1 Centroid Models

Centroid model algorithms are iterative algorithms where similarity or distance is derived from the closeness of the data points to the centroid of the individual clusters. The most well-known centroid based clustering algorithm is k-means clustering [6]. In k-means clustering, as well as other centroid models, the number of clusters k is a required aprior knowledge. These models are iterative algorithms that finds a local optimum.

The main idea of k-means clustering is to define k centroids for k clusters. In theory, the centroids should be placed far from each other to distinguish the clusters. The next step is to associate each data point to the nearest cluster centers. The algorithm is thus iterative between the two steps: updating centroids and assigning the next data points to a centroid that it is currently closest to. Formally the objective function for k-means can be written as,

$$\underset{\mathbf{U}}{\text{Minimize}}\ J(\mathbf{U}) = \underset{\mathbf{U}}{\text{Minimize}} \sum_{i=1}^{k} \sum_{j=1}^{k_j} \|\mathbf{x}_i - \mathbf{u}_j\|_2^2 \tag{2.1}$$

where u_j are the cluster centroids, k is the number of clusters, and k_j is the number of data points in the j-th cluster. Note the distance measure here is Euclidean distance or $l2$ norm.

The algorithm steps of k-means clustering can be formally described as followed. With the set of data points be $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$.

1. Randomly select k cluster centroids.

2. Find the Euclidean distance between each data point and the k cluster centroids.

3. Cluster assignment: assign each data point to a cluster by its smallest distance to the cluster centroid.

4. Recalculate the centroids with:

$$\mathbf{u}_j = \frac{1}{k_j} \sum_{i=1}^{k_j} \mathbf{x}_i$$

   where k_j are the number of data points in each cluster currently.

5. Repeat step 2 to 4 until,

6. None of the data points has been reassigned and cluster centroids are unchanged. The algorithm has come to a convergence point.

Figure 2.1 is an example of a k-means clustering solution [28]. The data set used is the "Iris" data set from the UCI Machine Learning Repository [29]. The left figure is the solution from k-means clustering while the right of the figure is the groundtruth of the data set.

The advantages of k-means clustering is that it is fast and easy to understand. The concept is intuitive to the idea of clustering. It also performs relatively well when the data points are well separated from each other and when all dimensions of the features are considered. The main disadvantage is that although this is a unsupervised learning algorithm, it still needs the apriori knowledge of the number of clusters. It also lacks the ability to separate clusters that are overlapping in some dimensions. The algorithm is also not invariant to non-linear transformations. This means that with different representation of data (such as from Cartesian plane to polar coordinates) k-means will give different results. Another big disadvantage is that the algorithm can often be trapped in a local minimum instead of achieving a global minimum.

2.3.2 Graphical Models

Another approach to solve the clustering problem is to view the data set as a graphical model. In these models, the data points are the nodes of the graph $\mathcal{V}$ and the edges of the graph, $\mathcal{E}$, are measures of similarity or distance between the data points. The graphical model is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. The goal of these clustering algorithms is to segregate the graph $\mathcal{G}$ into disconnected subgraphs with cuts along edges with low similarity or high distance measures. Each subgraph is then a cluster.

The most well-known graphical model based clustering algorithm is the normalized cut (N-cut) algorithm [5]. Prior to this work there was a pure minimum graph cut algorithm. Its criteria is to minimize a cut value

$$\text{cut}(A, B) = \sum_{\mathbf{x}_i \in A, \mathbf{x}_j \in B} w(\mathbf{x}_i, \mathbf{x}_j) \tag{2.2}$$

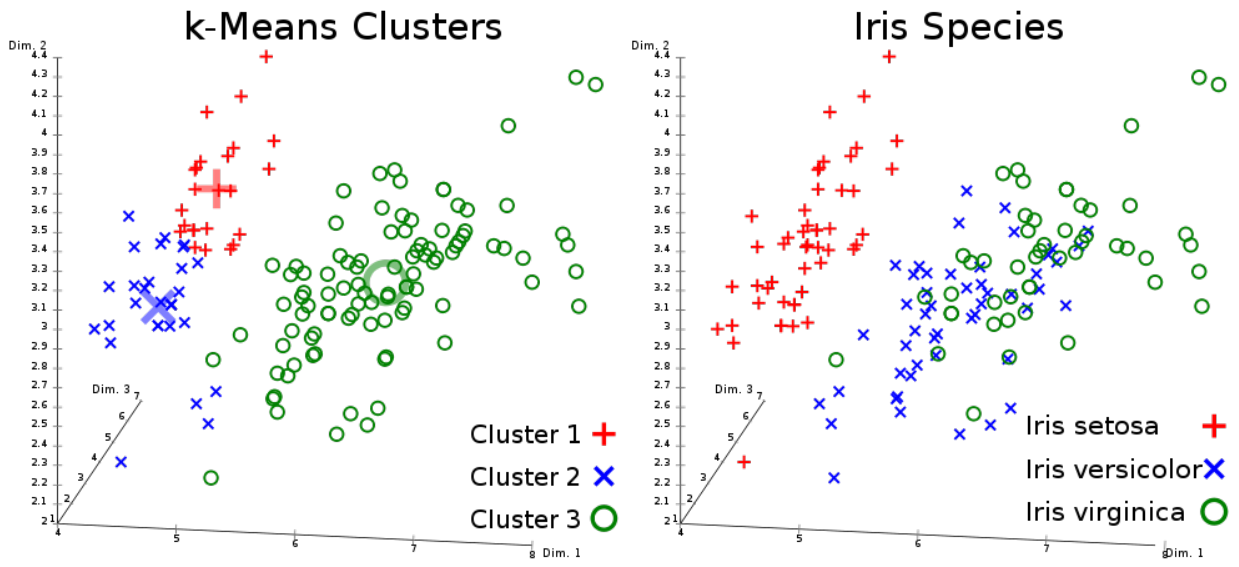


Figure 2.1: Example of the k-means clustering of "Iris" Data

where A and B is the two disjoint subgraphs and $w(\mathbf{x}_i, \mathbf{x}_j)$ is the edge weight, similarity or distance measure, between the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. The drawback of this minimum cut criteria is that it will support cutting isolated nodes in the graph due to the small values achieved by partitioning such nodes. In the work of Shi et al. [5], the *normalized* cut computes the cut cost as a fraction of the total edge connections to all the nodes in the graph. The N-cut is defined as

$$\text{N-cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \tag{2.3}$$

where $\text{assoc}(A, V) = \sum_{u \in A, t \in V} w(u, t)$. The distinct advantage is that it is an unbiased measure; the N-cut value with respect to the isolated nodes will be of a large percentage compared to the total connection from small set to all other nodes. Figure 2.2 shows an example of how normalized cut can outperform minimum cut [5]. To solve the normalized cut clustering problem, a similarity matrix, $\mathbf{W}$, of size N-by-N is first constructed where $w(i, j)$ is the similarity measure between $\mathbf{x}_i$ and $\mathbf{x}_j$. Second a diagonal matrix $\mathbf{D}$ of also size N-by-N is constructed where $\mathbf{D}(i, i) = \sum_j w(i, j)$ is the total connection weight from node i to all other nodes. The problem is then solved by constructing an eigensystem with the similarity matrix and the diagonal matrix. The

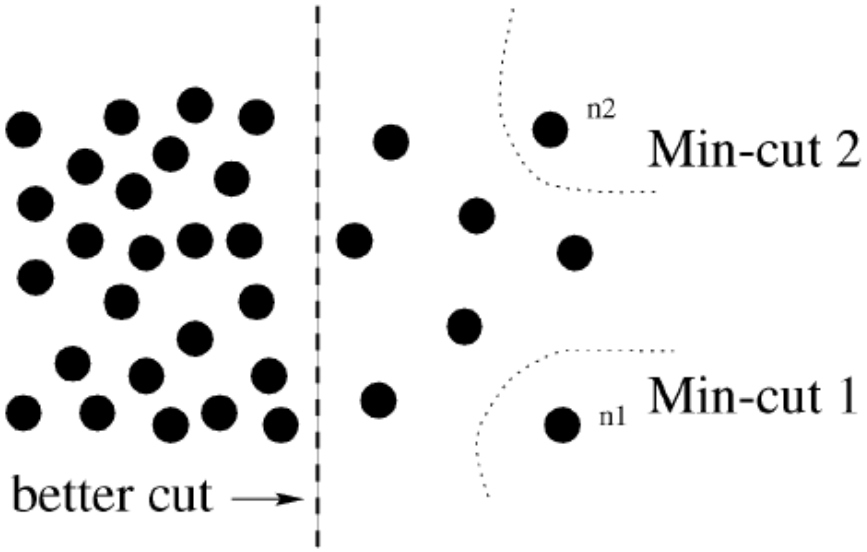


Figure 2.2: Example of when normalized cut performs better than min-cut

eigenvectors with the smallest eigenvalues for $(\mathbf{D} - \mathbf{W})\mathbf{x} = \lambda\mathbf{D}\mathbf{x}$. Due to the construction of the eigensystem, the main disadvantage of normalized cut is that it is only an approximate solution. Another disadvantage from the eigensystem is that it is limited computationally in the dimension of the data.

2.3.3 Hierarchical Models

Hierarchical clustering creates clusters in a tree-like structure. There are two types of hierarchical clustering, divisive and agglomerative [13]. In the divisive, or top-down method, the entire data set is first divided into two clusters. The procedure is repeated recursively on each sub-cluster until there is one cluster for each observation. The agglomerative, or bottom-top method, start with each data point in their own cluster, i.e. N clusters. Then the two most similar clusters are joined. This is also a recursive procedure until all the data points are in one cluster of size N. The main advantage of this clustering method is that clustering solutions of various fineness and cluster size can be obtained. The disadvantage is that hierarchical methods are usually computationally

expensive.

2.3.4 Distribution Models

Lastly, another clustering method is fitting the data into a probabilistic graphical model. These methods calculate how probable is it that all data points in the cluster belong to the same distribution, such as Gaussian Mixture Model (GMM) [14]. The disadvantage for these methods is over-fitting as well as susceptible to noise.

2.4 Conclusion

Clustering is a very useful technique in many areas of machine learning. Therefore, it is a well study subject with many previously well-known clustering methods. However, all of these previous methods have distinct disadvantages. Many of the algorithms only solve the problem approximately or is only able to find a local optimum. Graphical models for clustering requires solving eigensystem that is computationally too expensive with large dimensions. In the following sections, we propose a convex clustering scheme that solves the above disadvantages.

3. METRIC LEARNING

3.1 Introduction

As seen from the previous chapter, it is clear that all clustering algorithms relies on a measure of either distance or similarity. The most popular method of distance or similarity uses Euclidean distance. Euclidean distance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$d_2(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}.$$

It is clear that Euclidean distance weighs every feature equally in the distance calculation. That is not the case in most real world data sets. Thus, the goal of metric learning is to adapt a metric function to the problem of interest using training information. This metric function is characterized by a Mahalanobis metric:

$$d_{\mathbf{B}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{B}(\mathbf{x}_i - \mathbf{x}_j)}.$$

The Mahalanobis metric is characterized by a symmetric positive semi-definite matrix $\mathbf{B}$. To learn this matrix, most methods learn it from in a weakly-supervised way from pair or triplet based constraint of the form:

- Must-link/cannot-link training information:

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are similar } \},$$

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are dissimilar } \}.$$

- Relative constraint:

$$\mathcal{R} = \{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) : \mathbf{x}_i \text{ is more similar to } \mathbf{x}_j \text{ than to } \mathbf{x}_k\}.$$

Metric learning algorithms aim to find a Mahalanobis metric, in the form of the matrix $\mathbf{B}$, such that it best agrees with the training constraints. A general optimization problem for metric learning can be written as,

$$\underset{\mathbf{B}}{\text{Minimize}}\ \ell(\mathbf{B}, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(\mathbf{B}) \tag{3.1}$$

where $\ell(\mathbf{B}, \mathcal{S}, \mathcal{D}, \mathcal{R})$ is a loss function that penalizes when the training constraints are not met, $R(\mathbf{B})$ is a regularizer function on $\mathbf{B}$ and λ is the regularization parameter.

Metric learning is useful in many applications outside of clustering and classification as well. For example, in computer vision, there is a great need to find appropriate distance metrics not only to compare images or video in ad-hoc representations but also in pre-processing step [30]. Thus there has been many different works in computer vision problem such as image classification [31], object recognition [32], or visual tracking [33].

3.2 Notable Works in Metric Learning

The Mahalanobis distance came from Mahalanobis in 1936 [34] and originally refers to a distance measure with the matrix being the inverse of the correlation between features: $\mathbf{B} = \Omega^{-1}$. The data vectors are from the same distribution with covariance matrix Ω. The pioneer work on modern metric learning was in 2002 by Xing et al. [35] that formulated the metric learning problem as a convex optimization problem using must-link/cannot-link constraints in the training data. The convex optimization is stated as follows:

$$\begin{aligned} \underset{\mathbf{B}\in\mathbb{S}_+^d}{\text{Max}} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} d_{\mathbf{B}}(\mathbf{x}_i,\mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} d_{\mathbf{B}}(\mathbf{x}_i,\mathbf{x}_j)^2 \leq 1. \end{aligned} \tag{3.2}$$

The above optimization is then solved using a simple projected gradient approach requiring the full eigenvalue decomposition of $\mathbf{B}$ at each iteration.

Since Xing et al.'s work from 2002, many new methods of metric learning has stemmed. In particular, two works inspired the works in the following chapter. The first work was done by Hoi

et al. [36] in metric learning in the image clustering and retrieval field of research. In their work, it was proposed to follow the principles of manifold regularization for semi-supervised learning. Their formulation begins with a set of training data of size N in an d-dimensional vector space $\mathcal{C} = \{\mathbf{x}_i\}_{i=1}^N \subseteq \mathbb{R}^d$, and the must-link and cannot-link pairwise constraints:

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are similar}\},$$

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are dissimilar}\}.$$

The metric distance is then expressed as

$$d_{\mathbf{B}} = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{B}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{B}(\mathbf{x}_i - \mathbf{x}_j)} = \sqrt{\text{tr}(\mathbf{B}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T)}, \tag{3.3}$$

where $\mathbf{B}$ is the d-by-d metric and tr is the trace operator. In general, the metric matrix is valid if and only if it satisfies the nonnegativity and triangle inequality properties, meaning that it is positive semi-definite (PSD). With this formulation, Hoi et al. enhances the generalization and robustness performance of the distance metric learning problem proposed by Xing et al. by introducing the regularization principle. The regularization framework for distance metric learning was formulated as

$$\min_{\mathbf{B} \succeq 0} g(\mathbf{B}) + \gamma_s \mathcal{V}_s(\mathcal{S}) + \gamma_d \mathcal{V}_d(\mathcal{D}), \tag{3.4}$$

where $g(\mathbf{B})$ is a regularizer defined on the target metric $\mathbf{B}$, and $\mathcal{V}_s(\cdot)$ and $\mathcal{V}_d(\cdot)$ are some loss functions defined on the must-link and cannot-link constraints, respectively. Moreover, γ_s and γ_d are two regularization parameters for balancing the two sets of constraints. Intuitively, the loss functions should result in the minimization of the distances in the must-link constraints and the maximization of the distances in the cannot-link constraints. In Hoi et al. the two loss functions are chosen as

$$\mathcal{V}_s(\cdot) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{B}}^2, \qquad \mathcal{V}_d(\cdot) = - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{B}}^2. \tag{3.5}$$

To formulate the regularizer, the formulation takes advantage of the unlabeled data information in the regularization framework. For a set of unlabeled data, it uses a weight matrix $\mathbf{W}$ that encodes the similarity between pairs of points. The similarity matrix is constructed as

$$\mathbf{W}_{i,j} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \mathcal{N}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \\ 0, & \text{otherwise.} \end{cases}$$

where $\mathcal{N}(\mathbf{x}_i)$ denotes the nearest neighbor list of $\mathbf{x}_i$. Then the metric $\mathbf{B}$ can be seen as the product of a linear mapping $\mathbf{P}$ to itself: $\mathbf{B} = \mathbf{P}\mathbf{P}^T$.

$$d_{\mathbf{B}} = \|\mathbf{P}^T(\mathbf{x}_i - \mathbf{x}_j)\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{P}\mathbf{P}^T(\mathbf{x}_i - \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^2\mathbf{B}(\mathbf{x}_i - \mathbf{x}_j), \tag{3.6}$$

where $\mathbf{P}^T : \mathbb{R}^d \to \mathbb{R}^s$ and $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_s] \in \mathbb{R}^{d\times s}$. With this formulation of $\mathbf{B}$ and the weight matrix $\mathbf{W}$, a Laplacian regularizer is as follows:

$$g(\mathbf{B}) = \frac{1}{2}\sum_{i,j=1}^{N} \|\mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_j\|^2 W_{ij} = \text{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{B}), \tag{3.7}$$

where $\mathbf{L}$ is the Laplacian matrix: $\mathbf{L} = \mathbf{D} - \mathbf{W}$ ($\mathbf{D}$ is a diagonal matrix whose elements $D_{ii} = \sum_j W_{ij}$). Having defined the regularizer and the two loss function, the regularized metric learning problem in (3.4) can be rewritten as follows:

$$\begin{aligned} \min_{\mathbf{B}\succeq 0} \quad & \text{tr}(\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{B}) + \gamma_s \textstyle\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{B}} - \gamma_d \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\mathbf{B}} \\ \text{s. t.} \quad & \text{logdet}(\mathbf{B}) \geq 0 \end{aligned} \tag{3.8}$$

The constraint of $\text{logdet}(\mathbf{B}) \geq 0$ prevents trivial solutions but also ensures that $\mathbf{B}$ is full rank and thus positive definite. This constraint will also be used in our work in convex clustering and (full-rank) metric learning.

In the work of Hoi et al. the learned Mahalanobis metric was constraint to be full-rank. The disadvantages of having a full rank metric in a cluster scheme is that the metric learning gives no

dimension reduction and cannot eliminate completely irrelevant features. Thus, sparsity can be introduced in the structural constraints of the matrix $\mathbf{B}$. Shi et al. [37] proposed a new approach for metric learning that separates the Mahalanobis metric into a combination of orthonormal basis and a weighing vector. In their work they extract locally discriminative basis elements from the training data. The metric learning problem then learn the sparse combination of those elements. In this sparse compositional metric learning (SCML) scheme, the number of basis vectors in the basis set is much smaller than the number of features in the data, this reduces the computational complexity of the metric learning algorithm. SCML can also be seen a feature reduction algorithm as well as a metric learning algorithm. The metric matrix $\mathbf{B}$ can be represented as nonnegative weighted sum of s rank-1 PSD matrices:

$$\mathbf{B} = \sum_{i=1}^{s} \sigma_i \mathbf{q}_i \mathbf{q}_i^T = \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^T, \tag{3.9}$$

where $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_s)$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$. Each rank-1 basis $\mathbf{q}_i$ is a d-dimensional column vector. To solve the SCML problem given a set of training data, Shi et al. proposes a two parts algorithm. Here the training data is presents a sets of triplet constraint C where each $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in C$ indicates that $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar or a must-link set and $\mathbf{x}_i$ and $\mathbf{x}_k$ are dissimilar or a cannot-link set. The first step of the algorithm is to find the orthonormal rank-1 basis vectors. Fisher linear discriminant analysis (LDA) [38] is used to project the training data in to s-dimensional data. After having found the set of basis, the second part of the algorithm is to find $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$. The optimization problem can then be stated as

$$\min_{\mathbf{\Sigma}} \frac{1}{|C|} \sum_{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in C} L_{\mathbf{\Sigma}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) + \beta \|\mathbf{\Sigma}\|_1. \tag{3.10}$$

The first term is the margin-based hinge loss function and $L_{\mathbf{\Sigma}}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = [1 + d_{\mathbf{\Sigma}}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{\Sigma}}(\mathbf{x}_i, \mathbf{x}_k)]_+$. Note that $[\cdot]_+ = \max(0, \cdot)$. The second term is a $l1$ norm regularizer with $\beta \geq 0$ being the regularization parameter. In Shi et al. [37], this problem is convex by the linearity of both terms and bounded from below. A global minimum can be reached. Our second work on convex

clustering with sparse compositional metric learning was inspired by this work.

3.3 Conclusion

Metric learning methods have shown to enhance the results of clustering or classification. In this chapter, three methods of metric learning was introduced. In the earliest work of Xing et al. [35] the metric learning problem relies on a convex formulation with no regularization parameter. They first introduced the notion of must-link and cannot-link sets for training data. The algorithm maximizes the distance between pairs of the cannot-link set while constraining on the distances between pairs of the must-link set to be small. In Hoi et al. semi-supervised metric learning, Laplacian regularization is used to solve the metric learning problem. Moreover, to ensure a full-rank metric the optimization is constraint with the logdet$(\mathbf{B}) \geq 0$. Lastly, the sparse compositional metric learning algorithm in Shi et al. decomposes metric learning into finding orthonormal rank-1 basis and nonnegative weights for each of the basis vectors. In this work, the metric also has the effect of dimension reduction.

4. CONVEX CLUSTERING WITH METRIC LEARNING

4.1 Introduction

A common challenge for developing clustering algorithms is that many clustering formulations are inherently difficult to solve and in practice can only be approximately solved based on various heuristics [1]. The famous k-means [6] and normalized-cut [5] algorithms are two prime examples. One interesting exception is the recently proposed convex clustering (CC) formulation by Chi and Lange [15].[2] In their formulation, each data point is associated with a cluster center, and the goal is to minimize the aggregated distance between the data points and their corresponding cluster centers. A regularization term is then added to the objective function to leverage group sparsity to the clustering solution. Varying the weight of the regularization term creates a clustering path that may contain multiple meaningful solutions. More importantly, as demonstrated in [15], this formulation leads to a convex optimization problem, which can be precisely and efficiently solved using the well-known Alternating Direction Method of Multipliers (ADMM) [40–42].

One potential drawback about the CC formulation of Chi and Lange [15] is that it uses the standard Euclidean metric to measure the distance between the data points and their corresponding cluster centers. As is well known, the Euclidean metric treats each feature of the data equally, and as a result, the performance of the CC algorithm of Chi and Lange [15] deteriorates significantly in the presence of outlier features.

To address this issue, Wang et al. [26] proposed the so-called robust convex clustering (RCC) formulation, in which they introduced the so-called *robust component* to explicitly identify the outlier features of the data. By assuming that the outlier features are sparse, it was shown [26] that the robust component can be learned from the unlabeled data. However, even though RCC [26] can provide a performance boost over the CC algorithm of Chi and Lange [15], the underlying modeling assumption that the outlier features are sparse can be questionable. For example, for

[1] Part of this section is reprinted with permission from X. Sui, X. Li, X. Qian, and T. Liu, "Convex clustering with metric learning," *Pattern Recognition, vol. 81, pp. 575-584, September 2018*

[2] Using convex optimization techniques to solve clustering problems has also been previously explored in [7, 39].

many real-world data sets, it is the highly relevant features, rather than the outlier features, that are sparse. In a similar fashion, Wang et al. [27] proposed the so-called sparse convex clustering (SCC) formulation, in which they introduced sparsity constraints on the feature vector itself. SCC reformulates the convex clustering problem on the feature-level and add a sparsity constraint on the feature vectors itself. The goal of SCC is to eliminate features that shows low variance across all clusters. This sparsity of features only solves one type of feature relevancy problem. Thus, the SCC formulation does not yield the desirable effects as well. We propose to add metric learning to the distance calculation in convex clustering to solve these issues.

4.2 Convex Clustering

Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a collection of N data points to be clustered, and let $\mathbf{X}$ be the data matrix for which the jth column is given by $\mathbf{x}_j$ (so each row of $\mathbf{X}$ represents a feature of the data). In [15], the CC problem was formulated as the following optimization problem:

$$\underset{\mathbf{U}}{\text{Minimize}} \quad \frac{1}{2}\sum_{j=1}^{N} \|\mathbf{x}_j - \mathbf{u}_j\|_2^2 + \gamma \sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}} \|\mathbf{u}_{j_1} - \mathbf{u}_{j_2}\|_1 \tag{4.1}$$

where γ is a positive tuning constant, $w_{\{j_1,j_2\}}$ is a nonnegative weight, and the jth column $\mathbf{u}_j$ of the matrix $\mathbf{U}$ is the center of the cluster that the data point $\mathbf{x}_j$ belongs to. Multiple data points that belong to the same cluster will have the same cluster center vector, thus the columns of $\mathbf{U}$ are not unique: If there are k clusters, there will be k unique cluster centers, i.e. k unique columns of $\mathbf{U}$. Clearly, the goal of this convex optimization problem is to cluster the set of data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ such that the aggregated distance between the data points and their corresponding cluster centers is minimized. The second term in the objective function is a regularizer that leverages group sparsity to control the complexity (the number of clusters) of the clustering solution.

To solve the optimization problem using the aforementioned ADMM framework, let $\mathcal{E}$ be the set of edges in a complete graph with nodes $1, 2, \ldots, N$, *i.e.,* $\mathcal{E} = \{\{j_1, j_2\} : 1 \le j_1 < j_2 \le N\}$. We define a one-to-one edge-labeling mapping $\phi : \{1, 2, \ldots, \varepsilon\} \longrightarrow \mathcal{E}$ with $\varepsilon = N(N-1)/2$,

and let $\phi_1(\ell) = j_1$ and $\phi_2(\ell) = j_2$ if $\{j_1, j_2\} = \phi(\ell)$ and $j_1 < j_2$. For notational simplicity, let $w_\ell := w_{\phi(\ell)}$ for $1 \leq \ell \leq \varepsilon$. For each $1 \leq \ell \leq \varepsilon$, let $\mathbf{v}_\ell := \mathbf{u}_{\phi_1(\ell)} - \mathbf{u}_{\phi_2(\ell)}$ be the difference between the centroids $\mathbf{u}_{\phi_1(\ell)}$ and $\mathbf{u}_{\phi_2(\ell)}$. The matrix $\mathbf{V}$ is given by the collection of $\mathbf{v}_\ell$, $1 \leq \ell \leq \varepsilon$ as its columns. With this notion of the matrix $\mathbf{V}$, the convex clustering problem (4.1) can be recast as the following constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\text{Minimize}} \quad & \frac{1}{2}\sum_{j=1}^{N} \|\mathbf{x}_j - \mathbf{u}_j\|_2^2 + \gamma \sum_{\ell=1}^{\varepsilon} w_\ell \|\mathbf{v}_\ell\|_1 \\ \text{Subject to} \quad & \mathbf{u}_{\phi_1(\ell)} - \mathbf{u}_{\phi_2(\ell)} - \mathbf{v}_\ell = \mathbf{0}, \quad 1 \leq \ell \leq \varepsilon. \end{aligned} \tag{4.2}$$

Considering a vectorization of $\mathbf{U}$ and $\mathbf{V}$, the optimization problem (4.2) is a special case of the following general optimization problem:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v}}{\text{Minimize}} \quad & f(\mathbf{u}) + g(\mathbf{v}) \\ \text{Subject to} \quad & \mathbf{A}_1\mathbf{u} + \mathbf{A}_2\mathbf{v} = \mathbf{c}. \end{aligned} \tag{4.3}$$

The *augmented* Lagrangian of this general optimization problem is given by:

$$\mathcal{L}_\nu(\mathbf{u}, \mathbf{v}, \boldsymbol{\lambda}) := f(\mathbf{u}) + g(\mathbf{v}) + \langle \boldsymbol{\lambda}, \mathbf{c} - \mathbf{A}_1\mathbf{u} - \mathbf{A}_2\mathbf{v} \rangle + \frac{\nu}{2}\|\mathbf{c} - \mathbf{A}_1\mathbf{u} - \mathbf{A}_2\mathbf{v}\|_2^2, \tag{4.4}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrangian multipliers, and ν is a nonnegative tuning parameter. The ADMM minimizes the augmented Lagrangian $\mathcal{L}_\nu(\mathbf{u}, \mathbf{v}, \boldsymbol{\lambda})$ over its variables $\mathbf{u}$, $\mathbf{v}$ and $\boldsymbol{\lambda}$ separately and one block of variables at a time. This leads to the following sequential updates for $\mathbf{u}$, $\mathbf{v}$, and $\boldsymbol{\lambda}$:

$$\begin{aligned} \mathbf{u}^{m+1} &:= \arg\min_{\mathbf{u}} \mathcal{L}_\nu(\mathbf{u}, \mathbf{v}^m, \boldsymbol{\lambda}^m); \\ \mathbf{v}^{m+1} &:= \arg\min_{\mathbf{v}} \mathcal{L}_\nu(\mathbf{u}^{m+1}, \mathbf{v}, \boldsymbol{\lambda}^m); \\ \boldsymbol{\lambda}^{m+1} &:= \boldsymbol{\lambda}^m + \nu(\mathbf{c} - \mathbf{A}_1\mathbf{u}^{m+1} - \mathbf{A}_2\mathbf{v}^{m+1}). \end{aligned} \tag{4.5}$$

The CC algorithm proposed in [15] is based on calculating the updates of $\mathbf{u}^{m+1}$ and $\mathbf{v}^{m+1}$ efficiently until convergence. We shall describe these updates as a special case of our more general CC algorithm under a positive definite Mahalanobis distance metric in the next section.

4.3 Robust Convex Clustering

To improve the performance of CC in the presence of the outlier features, Wang et al. [26] proposed the following RCC problem:

$$\underset{\mathbf{U},\mathbf{Q}}{\text{Minimize}} \quad \frac{1}{2}\sum_{j=1}^{N}\|\mathbf{x}_j-(\mathbf{u}_j+\mathbf{q}_j)\|_2^2+\gamma\sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}}\|\mathbf{u}_{j_1}-\mathbf{u}_{j_2}\|_1+\beta\|\mathbf{Q}\|_{2,1} \tag{4.6}$$

where the matrix $\mathbf{Q}$ is the so-called *robust component* for which the jth column is given by $\mathbf{q}_j$, and β is a second tuning parameter in addition to γ. The penalization term $\beta\|\mathbf{Q}\|_{2,1}$ is introduced to achieve row-wise sparsity: If a feature is relevant, the corresponding row in $\mathbf{Q}$ will be zero for all elements; if a feature is an outlier, this row will be non-zero.

To solve the optimization problem (4.6), Wang et al. [26] proposed an alternating procedure that alternates between CC (minimizing over $\mathbf{U}$) and learning the robust component $\mathbf{Q}$. More specifically, for a fixed $\mathbf{Q}$, the optimization problem (4.6) reduces to the original CC problem (4.1) with the data set $\mathbf{X}$ replaced by $\mathbf{X}-\mathbf{Q}$. For a fixed $\mathbf{U}$, the optimization problem (4.6) admits a closed-form solution for $\mathbf{Q}$ whose ith row is given by [26]:

$$\max\left(0,1-\frac{\beta}{\|(\mathbf{X}-\mathbf{U})_i\|_2}\right)(\mathbf{X}-\mathbf{U})_i, \tag{4.7}$$

where $(\mathbf{X}-\mathbf{U})_i$ denotes the ith row of the matrix $\mathbf{X}-\mathbf{U}$. Thus, to solve the optimization problem (4.6), we may begin by setting the robust component $\mathbf{Q}$ as zero and perform CC. For the next iterations, one may alternate between learning the robust component according to (4.7) and CC, where learning the robust component is based on the optimal $\mathbf{U}$ obtained from the previous iteration, and CC is then based on the just-updated robust component $\mathbf{Q}$. We may continue such iterations till the solutions converge.

4.4 Convex Clustering with Full Rank Metric Learning Algorithm

To incorporate ML into the formulation of CC, let $\mathbf{B}$ be a *full rank* positive definite matrix and consider the following optimization problem:

$$\begin{array}{ll} \underset{\mathbf{U},\mathbf{B}}{\text{Minimize}} & \frac{1}{2}\sum_{j=1}^{N}(\mathbf{x}_j-\mathbf{u}_j)^T\mathbf{B}(\mathbf{x}_j-\mathbf{u}_j)+\gamma\sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}}\|\mathbf{u}_{j_1}-\mathbf{u}_{j_2}\|_1 \\ \text{Subject to} & \log\det(\mathbf{B})>0, \end{array} \tag{4.8}$$

where the choice of the constraint $\log\det(\mathbf{B}) \geq 0$ was motivated by [36] and ensures that the matrix $\mathbf{B}$ has a *full* rank. (As we shall see, maintaining the full rank of the matrix $\mathbf{B}$ is also crucial for developing the proper convex clustering algorithm.) The structure of the matrix $\mathbf{B}$ shows which features of the data are more congruent with the cluster assignment. In particular, when $\mathbf{B}$ is diagonal, the larger diagonal values of $\mathbf{B}$ correspond to the features that are of higher relevance or of lower noise corruptions. Note that for the original CC formulation [15] where $\mathbf{B}$ is an identity matrix, all features are uniformly weighted for clustering, which can be very sub-optimal in the presence of outlier features. For a general positive definite $\mathbf{B}$, its operational meaning can be understood through the standard singular value decomposition.

To solve the optimization problem (4.8), we shall consider an alternating procedure that alternates between CC (minimizing over $\mathbf{U}$) and ML (minimizing over $\mathbf{B}$).

4.4.1 Solving $\mathbf{U}$ for a Fixed $\mathbf{B}$

Fix $\mathbf{B}$ to be positive definite matrix and consider the artificial variables $\mathbf{v}_\ell := \mathbf{u}_{\phi_1(\ell)} - \mathbf{u}_{\phi_2(\ell)}$ for $1 \leq \ell \leq \varepsilon$. The optimization problem (4.8) can be equivalently written as:

$$\begin{array}{ll} \underset{\mathbf{U},\mathbf{V}}{\text{Minimize}} & \frac{1}{2}\sum_{j=1}^{N}(\mathbf{x}_j-\mathbf{u}_j)^T\mathbf{B}(\mathbf{x}_j-\mathbf{u}_j)+\gamma\sum_{\ell=1}^{\varepsilon} w_\ell\|\mathbf{v}_\ell\|_1 \\ \text{Subject to} & \mathbf{u}_{\phi_1(\ell)}-\mathbf{u}_{\phi_2(\ell)}-\mathbf{v}_\ell=\mathbf{0}, \quad 1\le\ell\le\varepsilon. \end{array} \tag{4.9}$$

Note that when $\mathbf{B}$ is an identity matrix, the optimization problem (4.9) reduces to the original CC formulation (4.2), which can be solved efficiently and precisely using the ADMM framework.

To apply the ADMM framework to solve the optimization problem (4.9), note that its aug-

mented Lagrangian is given by:

$$\begin{aligned}\mathcal{L}_\nu(\mathbf{U},\mathbf{V},\mathbf{\Lambda}) :=& \frac{1}{2}\sum_{j=1}^{N}(\mathbf{x}_j-\mathbf{u}_j)^T\mathbf{B}(\mathbf{x}_j-\mathbf{u}_j)+\gamma\sum_{\ell=1}^{\varepsilon}w_\ell\|\mathbf{v}_\ell\|_1+\\ &\sum_{\ell=1}^{\varepsilon}\boldsymbol{\lambda}_\ell^T(\mathbf{v}_\ell-\mathbf{u}_{\phi_1(\ell)}+\mathbf{u}_{\phi_2(\ell)})+\frac{\nu}{2}\sum_{\ell=1}^{\varepsilon}\|\mathbf{v}_\ell-\mathbf{u}_{\phi_1(\ell)}+\mathbf{u}_{\phi_2(\ell)}\|_2^2,\end{aligned} \tag{4.10}$$

where $\mathbf{\Lambda} := (\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\varepsilon)$. We shall update $\mathbf{U}$ and $\mathbf{V}$ in each iteration of the ADMM according to the procedure described in (4.5).

Updating $\mathbf{U}$. To update $\mathbf{U}$, we need to minimize the function

$$f(\mathbf{U}) := \frac{1}{2}\sum_{j=1}^{N}(\mathbf{x}_j-\mathbf{u}_j)^T\mathbf{B}(\mathbf{x}_j-\mathbf{u}_j)+\frac{\nu}{2}\sum_{\ell=1}^{\varepsilon}\|\tilde{\mathbf{v}}_\ell-\mathbf{u}_{\phi_1(\ell)}+\mathbf{u}_{\phi_2(\ell)}\|_2^2, \tag{4.11}$$

where $\tilde{\mathbf{v}}_l := \mathbf{v}_\ell + \nu^{-1}\boldsymbol{\lambda}_\ell$. Let $\mathbf{u} := \mathrm{vec}(\mathbf{U})$ and $\mathbf{x} := \mathrm{vec}(\mathbf{X})$. Then, the function $f(\mathbf{U})$ can be equivalently written as:

$$f(\mathbf{u}) = \frac{1}{2}(\mathbf{x}-\mathbf{u})^T\boldsymbol{\mathcal{B}}(\mathbf{x}-\mathbf{u})+\frac{\nu}{2}\sum_{\ell=1}^{\varepsilon}\|\mathbf{E}_\ell\mathbf{u}-\tilde{\mathbf{v}}_\ell\|_2^2, \tag{4.12}$$

where $\boldsymbol{\mathcal{B}} := \mathbf{I}\otimes\mathbf{B}$ and $\mathbf{E}_\ell := (\mathbf{e}_{\phi_1(\ell)}-\mathbf{e}_{\phi_2(\ell)})^T\otimes\mathbf{I}$. We can further simplify $f(\mathbf{u})$ as follows. Let

$$\mathbf{E} := \begin{pmatrix}\mathbf{E}_1\\ \vdots\\ \mathbf{E}_\varepsilon\end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{v}} := \begin{pmatrix}\tilde{\mathbf{v}}_1\\ \vdots\\ \tilde{\mathbf{v}}_\varepsilon\end{pmatrix}. \tag{4.13}$$

Then

$$f(\mathbf{u}) = \frac{1}{2}(\mathbf{x}-\mathbf{u})^T\boldsymbol{\mathcal{B}}(\mathbf{x}-\mathbf{u})+\frac{\nu}{2}(\mathbf{E}\mathbf{u}-\tilde{\mathbf{v}})^T(\mathbf{E}\mathbf{u}-\tilde{\mathbf{v}}). \tag{4.14}$$

We calculate the optimality condition for minimizing the quadratic function (4.14) as:

$$\left(\boldsymbol{\mathcal{B}} + \nu \mathbf{E}^T \mathbf{E}\right) \mathbf{u} = \boldsymbol{\mathcal{B}} \mathbf{x} + \nu \mathbf{E}^T \tilde{\mathbf{v}}. \tag{4.15}$$

Note that

$$\begin{aligned} \mathbf{E}^T \mathbf{E} &= \left[\sum_{\ell=1}^{\varepsilon} (\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)})(\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)})^T \right] \otimes \mathbf{I} && (4.16) \\ &= \left(N\mathbf{I} - \mathbf{1}\mathbf{1}^T\right) \otimes \mathbf{I} && (4.17) \end{aligned}$$

and

$$\mathbf{E}^T \tilde{\mathbf{v}} = \sum_{\ell=1}^{\varepsilon} [(\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)}) \otimes \mathbf{I}] \tilde{\mathbf{v}}_\ell. \tag{4.18}$$

Then, the optimality condition (4.15) can be written as:

$$[\mathbf{I} \otimes \mathbf{B} + \nu \left(N\mathbf{I} - \mathbf{1}\mathbf{1}^T\right) \otimes \mathbf{I}]\mathbf{u} = \boldsymbol{\mathcal{B}} \mathbf{x} + \nu \sum_{\ell=1}^{\varepsilon} [(\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)}) \otimes \mathbf{I}] \tilde{\mathbf{v}}_\ell, \tag{4.19}$$

yielding the following equivalent linear system:

$$\mathbf{B}\mathbf{U} + \mathbf{U}\mathbf{D} = \mathbf{B}\mathbf{X} + \mathbf{R}, \tag{4.20}$$

where $\mathbf{D} := \nu \left(N\mathbf{I} - \mathbf{1}\mathbf{1}^T\right)$ and $\mathbf{R} := \nu \sum_{\ell=1}^{\varepsilon} [\tilde{\mathbf{v}}_l (\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)})^T]$. Note that the system equation (4.20) is in fact a *Sylvester* equation [43].

By assumption $\mathbf{B}$ is positive definite so all eigenvalues of $\mathbf{B}$ are positive, while the eigenvalues of $-\mathbf{D}$ are $0, -N, \ldots, -N$. By the unique solution criterion [43], the Sylvester equation (4.20) must have a unique solution. To solve (4.20), note that when $\mathbf{B} = \mathbf{I}$, we simply have $\mathbf{U} = (\mathbf{X} + \mathbf{R})(\mathbf{I} + \mathbf{D})^{-1}$. This is the update procedure proposed in [15]. For a general positive definite $\mathbf{B}$, we can first transform $\mathbf{B}$ into a lower real Schur form [44] and $\mathbf{D}$ into an upper real Schur form

as follows:

$$\tilde{\mathbf{B}} := \mathbf{P}^T\mathbf{B}\mathbf{P} = \begin{bmatrix} \tilde{\mathbf{B}}_{1,1} & & \mathbf{0} & \\ \tilde{\mathbf{B}}_{2,1} & \tilde{\mathbf{B}}_{2,2} & & \\ \vdots & \vdots & \ddots & \\ \tilde{\mathbf{B}}_{d,1} & \tilde{\mathbf{B}}_{d,2} & \cdots & \tilde{\mathbf{B}}_{d,d} \end{bmatrix} \tag{4.21}$$

and

$$\tilde{\mathbf{D}} := \mathbf{Q}^T\mathbf{D}\mathbf{Q} = \begin{bmatrix} \tilde{\mathbf{D}}_{1,1} & \tilde{\mathbf{D}}_{2,1} & \cdots & \tilde{\mathbf{D}}_{r,1} \\ & \tilde{\mathbf{D}}_{2,2} & \cdots & \tilde{\mathbf{D}}_{r,2} \\ & \mathbf{0} & \ddots & \vdots \\ & & & \tilde{\mathbf{D}}_{r,r} \end{bmatrix}, \tag{4.22}$$

where $\tilde{\mathbf{B}}$ is lower quasi-triangular, $\tilde{\mathbf{D}}$ is upper quasi-triangular, the diagonal blocks $\tilde{\mathbf{B}}_{i,i}$ and $\tilde{\mathbf{D}}_{i,i}$ are order of at most two, and $\mathbf{P}$ and $\mathbf{Q}$ are both orthogonal. Then, we can solve the transformed equation

$$\tilde{\mathbf{B}}\tilde{\mathbf{U}} + \tilde{\mathbf{U}}\tilde{\mathbf{D}} = \mathbf{P}^T(\mathbf{B}\mathbf{X} + \mathbf{R})\mathbf{Q} = \tilde{\mathbf{B}}\mathbf{P}^T\mathbf{X}\mathbf{Q} + \mathbf{P}^T\mathbf{R}\mathbf{Q} \tag{4.23}$$

by *backward substitutions* [45]. The solution of the original equation (4.20) is thus given by $\mathbf{U} = \mathbf{P}\tilde{\mathbf{U}}\mathbf{Q}^T$.

Updating $\mathbf{V}$. To update $\mathbf{V}$, observe that the augmented Lagrangian $\mathcal{L}_\nu(\mathbf{U}, \mathbf{V}, \mathbf{\Lambda})$ is *separable* in the vectors $\mathbf{v}_\ell$. Thus, for any $1 \leq \ell \leq \varepsilon$, $\mathbf{v}_\ell$ can be updated as [15]:

$$\begin{aligned} \mathbf{v}_\ell &= \arg\min_{\mathbf{v}} \left[\frac{1}{2}\|\mathbf{v} - (\mathbf{u}_{\phi_1(\ell)} - \mathbf{u}_{\phi_2(\ell)} - \nu^{-1}\boldsymbol{\lambda}_\ell)\|_2^2 + \frac{\gamma w_\ell}{\nu}\|\mathbf{v}\|_1\right] \\ &= \mathcal{S}\left(\mathbf{u}_{\phi_1(\ell)} - \mathbf{u}_{\phi_2(\ell)} - \nu^{-1}\boldsymbol{\lambda}_\ell, \frac{\gamma w_\ell}{\nu}\mathbf{1}\right), \end{aligned} \tag{4.24}$$

where $\mathcal{S}$ is the element-wise soft-thresholding function given by $\mathcal{S}(\mathbf{x}, \mathbf{a}) := (\mathbf{x} - \mathbf{a})_+ - (-\mathbf{x} - \mathbf{a})_+$.

Algorithm, convergence, and complexity. Algorithm 1 summarizes the updates of $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{\Lambda}$ in the ADMM. It is straightforward to verify that the optimization problem (4.9) satisfies the Slater's condition [46] and hence that the strong duality holds. It then follows from the saddle-point property [47] that there exists a $(\mathbf{U}^*, \mathbf{V}^*, \mathbf{\Lambda}^*)$ such that the un-augmented Lagrangian $\mathcal{L}_0$

Algorithm 1 Solving $\mathbf{U}$ for a fixed $\mathbf{B}$ via the ADMM

Input: $\mathbf{X}, \mathbf{B}, \gamma, \nu$ **and** $\{w_l\}_{\ell=1}^{\varepsilon}$**.**
Output: $\mathbf{U}, \mathbf{V}$, **and** $\mathbf{\Lambda}$**.**

Set the maximum number of iterations ω.
Initialize $\mathbf{\Lambda}^{(0)}$ and $\mathbf{V}^{(0)}$.
$\mathbf{D} := \nu\left(N\mathbf{I} - \mathbf{1}\mathbf{1}^T\right)$.
Find the Schur forms $\tilde{\mathbf{B}} = \mathbf{P}^T\mathbf{B}\mathbf{P}$ and $\tilde{\mathbf{D}} = \mathbf{Q}^T\mathbf{D}\mathbf{Q}$ of $\mathbf{B}$ and $\mathbf{D}$ by (4.22), respectively.
$\tilde{\mathbf{X}} := \tilde{\mathbf{B}}\mathbf{P}^T\mathbf{X}\mathbf{Q}$.
for $m = 1, 2, 3, \ldots, \omega$ **do**
    $\mathbf{R}^{(m)} := \nu \sum_{\ell=1}^{\varepsilon} \left[(\mathbf{v}_\ell^{(m-1)} + \nu^{-1}\boldsymbol{\lambda}_\ell^{(m-1)})(\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)})^T\right]$.
    Find the solution $\tilde{\mathbf{U}}^{(m)}$ of $\tilde{\mathbf{B}}\tilde{\mathbf{U}} + \tilde{\mathbf{U}}\tilde{\mathbf{D}} = \tilde{\mathbf{X}} + \mathbf{P}^T\mathbf{R}^{(m)}\mathbf{Q}$ by backward substitution.
    $\mathbf{U}^{(m)} := \mathbf{P}\tilde{\mathbf{U}}^{(m)}\mathbf{Q}^T$.
    for $\ell = 1, 2, \ldots, \varepsilon$ **do**
        $\mathbf{v}_\ell^{(m)} := \mathcal{S}\left(\mathbf{u}_{\phi_1(\ell)}^{(m)} - \mathbf{u}_{\phi_2(\ell)}^{(m)} - \nu^{-1}\boldsymbol{\lambda}_\ell^{(m-1)}, \frac{\gamma w_\ell}{\nu}\mathbf{1}\right)$
        $\boldsymbol{\lambda}_\ell^{(m)} := \boldsymbol{\lambda}_\ell^{(m-1)} + \nu\left(\mathbf{v}_\ell^{(m)} - \mathbf{u}_{\phi_1(\ell)}^{(m)} + \mathbf{u}_{\phi_2(\ell)}^{(m)}\right)$.
    end for
end for
return $\mathbf{U} := \mathbf{U}^{(\omega)}$, $\mathbf{V} := \mathbf{V}^{(\omega)}$, and $\mathbf{\Lambda} := \mathbf{\Lambda}^{(\omega)}$.

satisfies:

$$\mathcal{L}_0(\mathbf{U}^*, \mathbf{V}^*, \mathbf{\Lambda}) \leq \mathcal{L}_0(\mathbf{U}^*, \mathbf{V}^*, \mathbf{\Lambda}^*) \leq \mathcal{L}_0(\mathbf{U}, \mathbf{V}, \mathbf{\Lambda}^*) \tag{4.25}$$

for any $\mathbf{U}, \mathbf{V}$, and $\mathbf{\Lambda}$. We may thus conclude by the convergence criterion of ADMM [40, 42] that Algorithm 1 converges to the optimal value of the optimization problem (4.9). Finally, we note that the computational complexity for solving the Sylvester equation (4.20) is $O(d^3 + d^2N + dN^2 + N^3)$ [45], where d is the number of features of the data and N is the number of data points. Considering that N is usually much larger than d, this is rather comparable to the $O(N^3)$ complexity for inverting the matrix $\mathbf{I} + \mathbf{D}$ needed for solving the original CC formulation of Chi and Lange [15].

4.4.2 Solving B for a Fixed U

Fixing $\mathbf{U}$, the optimization problem (4.8) can be equivalently written as:

$$\begin{array}{ll} \underset{\mathbf{B}}{\text{Minimize}} & \sum_{j=1}^{N}(\mathbf{x}_j - \mathbf{u}_j)^T\mathbf{B}(\mathbf{x}_j - \mathbf{u}_j) \\ \text{Subject to} & \log\det(\mathbf{B}) \geq 0. \end{array} \tag{4.26}$$

To find an optimal solution for $\mathbf{B}$, let

$$\mathbf{A} := \sum_{j=1}^{N}(\mathbf{x}_j - \mathbf{u}_j)(\mathbf{x}_j - \mathbf{u}_j)^T = (\mathbf{X} - \mathbf{U})(\mathbf{X} - \mathbf{U})^T. \tag{4.27}$$

The Lagrangian of (4.26) is given by:

$$\mathcal{L}(\mathbf{B}, \mu) = \sum_{j=1}^{N}(\mathbf{x}_j - \mathbf{u}_j)^T\mathbf{B}(\mathbf{x}_j - \mathbf{u}_j) - \mu\log\det(\mathbf{B}) \tag{4.28}$$

$$= \operatorname{tr}(\mathbf{AB}) - \mu\log\det(\mathbf{B}). \tag{4.29}$$

Its Karush-Kuhn-Tucker conditions yield:

$$\mathbf{0} = \frac{\partial\mathcal{L}}{\partial\mathbf{B}} = \mathbf{A}^T - \mu(\mathbf{B}^{-1})^T \tag{4.30}$$

$$\log\det(\mathbf{B}) \geq 0 \tag{4.31}$$

$$\mu \geq 0 \tag{4.32}$$

$$\mu\log\det(\mathbf{B}) = 0. \tag{4.33}$$

Assuming that $\mathbf{A}$ has a full rank, i.e., no features are completely redundant, a closed-form solution of (4.26) is given by:

$$\mathbf{B} = \det(\mathbf{A})\mathbf{A}^{-1}. \tag{4.34}$$

4.4.3 Iteration between Convex Clustering and Metric Learning

To solve the optimization problem (4.8), we shall begin by setting the matrix $\mathbf{B}$ as an identity matrix and perform Algorithm 1. This is equivalent to the ADMM algorithm for CC proposed in [15]. For the next iterations, we alternate between ML according to (4.34) and CC according to Algorithm 1, where ML is based on the optimal $\mathbf{U}$ obtained from the previous iteration, and CC is then based on the just-updated matrix $\mathbf{B}$ from the ML. We may continue such iterations till the solutions converge to a local minimum.

4.5 Sparse Convex Clustering

The above metric learning algorithm proposed constraints the Mahalanobis metric matrix $\mathbf{B}$ to be full rank. Although in experiments with data set of relatively small dimensions the results for convex clustering with full rank metric learning performs well, it is too computationally expensive when the dimension of the data set grows large. Moreover, in many real life applications, some data set has variable features that are very noisy, irrelevant or invariant. In these cases, it is best to eliminate meaningless dimensions in the data.

Wang et al. proposes an alternative to convex clustering that imposes sparsity to the dimensions of the data in convex clustering. They introduce a sparse convex clustering (SCC) algorithm that formulate convex clustering in a form of regularization with an adaptive group-lasso penalty term on cluster centers to encourage the sparsity.

Wang et al. reformulates the convex clustering optimization (4.1). The data matrix $\mathbf{X}$ can be rewritten in feature-level as column vector $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d)$, where $\mathbf{x}_j = (\mathbf{X}_{1j}, \ldots, \mathbf{X}_{Nj})^T$ for $j = 1, \ldots, d$. The center matrix, $\mathbf{U}$, can be rewritten as well in feature vector as column vector $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d)$. In this formulation, it is assumed the feature vectors are centered, i.e. $\sum_{i=1}^{N} \mathbf{X}_{ij} = 0$ for each $j = 1, \ldots, d$. Thus the problem (4.1) can be formulated as:

$$\min_{\mathbf{A} \in \mathbb{R}^{N \times d}} \frac{1}{2} \sum_{j=1}^{d} \|\mathbf{x}_j - \mathbf{a}_j\|_2^2 + \gamma \sum_{l \in \mathcal{E}} w_l \|A_{i_1.} - A_{i_2.}\|_q, \tag{4.35}$$

where $\mathcal{E} = \{l = (i_1, i_2) : 1 \leq i_1 \leq i_2 \leq N\}$. Denoting $\hat{\mathbf{A}} = (\hat{A_1}, \ldots, \hat{A_N})^T = (\hat{\mathbf{a}_1} - \hat{\mathbf{a}_d})$ as a solution to the convex clustering problem, if $\hat{A_{i_1.}} = \hat{A_{i_2.}}$ then the data points i_1 and i_2 belong to the same cluster (have the same cluster center). The feature-level estimate $\hat{\mathbf{a}_j}, j = 1, \ldots, d$ implies feature importance - if the components of a feature-level estimate $\hat{\mathbf{a}_j}$ are identical, then the corresponding feature j is not useful in clustering. In high dimensional data, the solution $\hat{\mathbf{A}}$ is desired to be sparse, meaning with many of the columns being exactly $\mathbf{0}$. Wang et al. then incorporate an adaptive group-lasso penalty [48] into the convex clustering optimization problem to exclude uninformative features. The sparse convex clustering is then defined as:

$$\min_{\mathbf{A} \in \mathbb{R}^{N \times d}} \frac{1}{2} \sum_{j=1}^{d} \|\mathbf{x}_j - \mathbf{a}_j\|_2^2 + \gamma_1 \sum_{l \in \mathcal{E}} w_l \|A_{i_1.} - A_{i_2.}\|_q + \gamma_2 \sum_{j=1}^{d} u_j \|\mathbf{a}_j\|_2, \tag{4.36}$$

where the tuning parameter γ_1 controls the cluster size or number of clusters and the parameter γ_2 controls the sparsity of the informative features. In the group-lasso penalty, the weight u_j adaptively penalizes the features. Wang et al. shows in [27] that this SCC problem can still be solved using the ADMM method.

Although the SCC scheme brings sparsity in the feature space to the convex clustering problem, it is only able to capture one type of uninformative feature. The SCC is formulated thus that features with low or zero variance is eliminated. This however is only one of the ways a feature can be irrelevant to clustering. The SCC formulation is not able to identify features that are highly noisy or features that are redundant. It is then inadequate in elevating performance of convex clustering by removing invariant features alone. Therefore, we propose in the below section to impose sparsity in metric learning instead of the clustering algorithm.

4.6 Convex Clustering with Sparse Compositional Metric Learning

In the formulation of the previous convex clustering with metric learning work, the Mahalanobis distance metric $\mathbf{B}$ is required to have a *full* rank (so the entire collection of features will be utilized for the purpose of clustering). While this seems to be necessary for avoiding trivial clustering solutions without any structural constraint on $\mathbf{B}$, there are two potential problems with

this choice.

- First, from the computational complexity viewpoint, when dealing with *high-dimensional* data, solving the full-dimensional convex clustering problem (4.1) using the ADMM is computationally demanding.

- Second, from the performance viewpoint, the relevant features are known to be *sparse* for many real-world data sets and forcing all features (the outliers especially) to be used may incur (significant) performance loss. We mention here that the issue of sparsity has been considered in a convex clustering formulation known as robust convex clustering [26]. However, the modeling assumption there was that it is the outlier features, rather than the relevant ones, that are sparse.

4.6.1 Structural Constraints on Metric Learning

A natural idea for addressing the above issues is to impose *structural* constraints on the Mahalanobis distance metric $\mathbf{B}$. In this work, we focus on the so-called *sparse compositional* Mahalanobis distance metric, which was first considered by Shi et al. [37]. More specifically, let $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_s)$, where $\{\mathbf{q}_i\}_{i=1}^s$ is a set of orthonormal vectors in $\mathbb{R}^d$, and let $\mathbf{\Sigma}$ be an $s \times s$ diagonal matrix with the diagonal elements given by the positive real numbers $\sigma_1, \sigma_2, \ldots, \sigma_s$. The so-called sparse compositional Mahalanobis distance metric takes the form

$$\mathbf{B} = \sum_{i=1}^{s} \sigma_i \mathbf{q}_i \mathbf{q}_i^T = \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^T, \tag{4.37}$$

where s is usually much smaller than d to justify the name "sparse". One way to interpret the above distance metric is that it first projects the data along the directions of $\{\mathbf{q}_i\}_{i=1}^s$ and then computes the weighted square Euclidean distance using $\sigma_1, \sigma_2, \ldots, \sigma_s$ as the corresponding weights. Motivated by this interpretation, let us consider the following convex clustering problem:

$$\begin{aligned} \underset{\mathbf{U}}{\text{Minimize}} \quad & \tfrac{1}{2}\textstyle\sum_{j=1}^{N}(\mathbf{Q}^T\mathbf{x}_j - \mathbf{u}_j)^T\mathbf{\Sigma}(\mathbf{Q}^T\mathbf{x}_j - \mathbf{u}_j) + \\ & \gamma \textstyle\sum_{1\le j_1<j_2\le N} w_{\{j_1,j_2\}} \|\mathbf{u}_{j_1} - \mathbf{u}_{j_2}\|_1 \end{aligned} \tag{4.38}$$

Algorithm 2 Solving $\mathbf{U}$ for a fixed $(\mathbf{Q}, \mathbf{\Sigma})$ via the ADMM

Input: $\mathbf{X}$, $\mathbf{Q}$, $\mathbf{\Sigma}$, γ, ν, and $\{w_\ell\}_{\ell=1}^{\varepsilon}$.
Output: $\mathbf{U}$,$\mathbf{V}$, and $\mathbf{\Lambda}$.

Set the maximum number of iterations ω.
Initialize $\mathbf{\Lambda}^0$ and $\mathbf{V}^0$.
for $m = 1, 2, 3, \ldots, \omega$ **do**
    $\mathbf{R}^m := \nu \sum_{\ell=1}^{\varepsilon} \left[(\mathbf{v}_\ell^{(m-1)} + \nu^{-1} \boldsymbol{\lambda}_\ell^{(m-1)}) (\mathbf{e}_{\phi_1(\ell)} - \mathbf{e}_{\phi_2(\ell)})^T \right]$.
    for $i = 1, 2, \ldots, s$ **do**
        $\mathbf{U}_i^m := (\mathbf{\Sigma} \mathbf{Q}^T \mathbf{X} + \mathbf{R}^m)_i \left(\frac{1}{\sigma_i + \nu N} \mathbf{I} + \frac{\nu}{\sigma_i (\sigma_i + \nu N)} \mathbf{1}\mathbf{1}^T \right)$.
    end for
    for $\ell = 1, 2, \ldots, \varepsilon$ **do**
        $\mathbf{v}_\ell^m := \mathcal{S} \left(\mathbf{u}_{\phi_1(\ell)}^m - \mathbf{u}_{\phi_2(\ell)}^m - \nu^{-1} \boldsymbol{\lambda}_\ell^{m-1}, \frac{\gamma w_\ell}{\nu} \mathbf{1} \right)$.
        $\boldsymbol{\lambda}_\ell^m := \boldsymbol{\lambda}_\ell^{m-1} + \nu \left(\mathbf{v}_\ell^m - \mathbf{u}_{\phi_1(\ell)}^m + \mathbf{u}_{\phi_2(\ell)}^m \right)$.
    end for
end for
return $\mathbf{U} := \mathbf{U}^\omega$, $\mathbf{V} := \mathbf{V}^\omega$, and $\mathbf{\Lambda} := \mathbf{\Lambda}^\omega$.

for a given $(\mathbf{Q}, \mathbf{\Sigma})$.

Note that by viewing $\{\mathbf{Q}^T \mathbf{x}_j\}_{j=1}^N$ as the new data set in $\mathbb{R}^s$, the convex clustering problem (4.38) reduces to the convex clustering problem (4.1) but with two important advantages. First, the dimension of the data set $\{\mathbf{Q}^T \mathbf{x}_j\}_{j=1}^N$ for (4.38) is s, which is usually much smaller than d, the dimension of the data set $\{\mathbf{x}_j\}_{j=1}^N$ for (4.1). Therefore, the ADMM described in Section 4.4 is much more efficient for solving (4.38) than for solving (4.1). Second, the *de facto* Mahalanobis distance metric for (4.38) is $\mathbf{\Sigma}$, which is diagonal. This can be taken advantage of for the update of $\mathbf{U}$ in the ADMM as follows. Note that the Sylvester equation for the convex clustering problem (4.38) can be written as:

$$\mathbf{\Sigma} \mathbf{U} + \mathbf{U} \mathbf{D} = \mathbf{\Sigma} \mathbf{Q}^T \mathbf{X} + \mathbf{R}, \tag{4.39}$$

where the dimensions of $\mathbf{U}$ and $\mathbf{R}$ are now $s \times N$ instead of $d \times N$. When $\mathbf{\Sigma}$ is diagonal, there is a simpler way for solving $\mathbf{U}$. Let $\mathbf{U}_i$ and $(\mathbf{\Sigma} \mathbf{Q}^T \mathbf{X} + \mathbf{R})_i$, $i = 1, 2, \ldots, s$ be the row vectors of $\mathbf{U}$

and $\mathbf{\Sigma Q}^T\mathbf{X} + \mathbf{R}$, respectively. From (4.39) we have

$$\sigma_i \mathbf{U}_i + \mathbf{U}_i \mathbf{D} = (\mathbf{\Sigma Q}^T \mathbf{X} + \mathbf{R})_i \tag{4.40}$$

for $i = 1, 2, \ldots, s$. It follows immediately that

$$\begin{aligned}
\mathbf{U}_i &= (\mathbf{\Sigma Q}^T \mathbf{X} + \mathbf{R})_i (\sigma_i \mathbf{I} + \mathbf{D})^{-1} && (4.41)\\
&= (\mathbf{\Sigma Q}^T \mathbf{X} + \mathbf{R})_i ((\sigma_i + \nu N)\mathbf{I} - \nu \mathbf{1}\mathbf{1}^T)^{-1} && (4.42)\\
&= (\mathbf{\Sigma Q}^T \mathbf{X} + \mathbf{R})_i \left(\frac{1}{\sigma_i + \nu N} \mathbf{I} + \frac{\nu}{\sigma_i (\sigma_i + \nu N)} \mathbf{1}\mathbf{1}^T \right) && (4.43)
\end{aligned}$$

for $i = 1, 2, \ldots, s$. Algorithm 2 summarizes the ADMM for solving the convex clustering problem (4.38).

4.6.2 Convex Clustering with Sparse Compositional Metric Learning

To incorporate metric learning into the convex clustering formulation (4.38), we shall follow [16] and consider the following optimization framework:

$$\begin{aligned}
\underset{\mathbf{U},\mathbf{Q},\mathbf{\Sigma}}{\text{Minimize}} \quad & \tfrac{1}{2} \textstyle\sum_{j=1}^{N} (\mathbf{Q}^T \mathbf{x}_j - \mathbf{u}_j)^T \mathbf{\Sigma} (\mathbf{Q}^T \mathbf{x}_j - \mathbf{u}_j) + \\
& \qquad \gamma \textstyle\sum_{1 \le j_1 < j_2 \le N} w_{\{j_1, j_2\}} \|\mathbf{u}_{j_1} - \mathbf{u}_{j_2}\|_1 \\
\text{Subject to} \quad & \textstyle\prod_{i=1}^{s} \sigma_i \ge 1, \quad \sigma_i \ge 0 \ \forall i = 1, 2, \ldots, s,
\end{aligned} \tag{4.44}$$

where the optimization is jointly over the cluster centers $\mathbf{U}$, the set of orthonormal vectors $\{\mathbf{q}_i\}_{i=1}^s$, and the nonnegative weights $\{\sigma_i\}_{i=1}^s$. The constraint $\prod_{i=1}^s \sigma_i \ge 1$ is to ensure that all weights $\{\sigma_i\}_{i=1}^s$ are in fact strictly positive.

To solve the optimization problem (4.44), we shall consider the following iterative algorithm

that sequentially updates $\mathbf{Q}$, $\mathbf{\Sigma}$, and $\mathbf{U}$ in each iteration:

$$\begin{aligned} \mathbf{Q}^{m+1} &:= \mathbf{Q}(\mathbf{U}^m) \\ \mathbf{\Sigma}^{m+1} &:= \mathbf{\Sigma}(\mathbf{Q}^{m+1}, \mathbf{U}^m) \\ \mathbf{U}^{m+1} &:= \mathbf{U}(\mathbf{Q}^{m+1}, \mathbf{\Sigma}^{m+1}). \end{aligned} \tag{4.45}$$

To update $\mathbf{Q}$ from $\mathbf{U}^m$, we shall first extract the clustering assignment from $\mathbf{U}^m$, i.e., to put the data points with the same cluster center in the same cluster. Based on this clustering assignment, we shall follow [37] and use the Fisher Linear Discriminant Analysis (LDA) to find a new set of orthonormal vectors $\{\mathbf{q}_i^{m+1}\}_{i=1}^s$. The details of the LDA are included in the next section.

To update $\mathbf{\Sigma}$ from $\mathbf{Q}^{m+1}$ and $\mathbf{U}^m$, we shall consider the following metric learning problem:

$$\begin{aligned} \underset{\mathbf{\Sigma}}{\text{Minimize}} \quad & \tfrac{1}{2}\textstyle\sum_{j=1}^N (\mathbf{Q}^T\mathbf{x}_j - \mathbf{u}_j)^T \mathbf{\Sigma} (\mathbf{Q}^T\mathbf{x}_j - \mathbf{u}_j) \\ \text{Subject to} \quad & \textstyle\prod_{i=1}^s \sigma_i \geq 1, \quad \sigma_i \geq 0 \; \forall i = 1, 2, \ldots, s \end{aligned} \tag{4.46}$$

where we fix $\mathbf{Q} = \mathbf{Q}^{m+1}$ and $\mathbf{U} = \mathbf{U}^m$. Note that the objective function of (4.46) can be equivalently written as $\sum_{i=1}^s A_i \omega_i$, where $A_i := \frac{1}{2}\sum_{j=1}^N (\mathbf{Q}^T\mathbf{x}_j - \mathbf{u}_j)_i^2$. It follows immediately from the inequality of arithmetic and geometric means [49] that the optimal solution to (4.46) is given by:

$$\sigma_i = \frac{1}{A_i} \left(\prod_{t=1}^s A_t \right)^{1/s} \tag{4.47}$$

for $i = 1, 2, \ldots, s$. We mention here that the problem of learning a sparse compositional Mahalanobis metric was also considered in [37], where the choice of the objective function was the regularized margin-based hinge loss function. By comparison, the objective function of our metric problem (4.46) follows directly from the convex clustering formulation (4.38) and admits a very simple *closed-form* solution (4.47).

Finally, to update $\mathbf{U}$ from $\mathbf{Q}^{m+1}$ and $\mathbf{\Sigma}^{m+1}$, we simply solve the convex clustering problem (4.38) by setting $\mathbf{Q} = \mathbf{Q}^{m+1}$ and $\mathbf{\Sigma} = \mathbf{\Sigma}^{m+1}$.

We conclude this section by discussing how to obtain a good initial value for $\mathbf{U}$. We propose

the following solution. First, we obtain a clustering assignment by solving the convex clustering formulation of Chi and Lange [15]. Based on this clustering assignment, next we use the Fisher LDA to obtain a set of orthonormal vectors and project the data points using these vectors. Finally, we set the the cluster centers $\mathbf{U}$ as the (arithmetic) means of the projected data points within each cluster.

4.6.3 Fisher Linear Discriminant Analysis

The first step to learning a sparse compositional metric is to find the set of orthonormal basis. We use Fisher Linear Discriminant Analysis (LDA) to find the basis. The main idea of Fisher LDA is to separate the samples of distinct clusters by projecting them onto a subspace that maximizes the inter-cluster distance while minimizing the intra-cluster distance. Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be the collection of data points in $\mathbb{R}^d$, and let $c : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, K\}$ be a cluster assignment that assigns each of the data points to one of the K clusters. The Fisher LDA considers maximizing the following objective function [38]:

$$J(\mathbf{q}) := \frac{\mathbf{q}^T \mathsf{S}_B \mathbf{q}}{\mathbf{q}^T \mathsf{S}_W \mathbf{q}}$$

where

$$\mathbf{S}_B := \sum_{k=1}^{K} (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$$
$$\text{and} \quad \mathbf{S}_W := \sum_{j=1}^{N} (\mathbf{x}_j - \boldsymbol{\mu}_{c(j)})(\mathbf{x}_j - \boldsymbol{\mu}_{c(j)})^T.$$

Here, $\boldsymbol{\mu}$ is the (arithmetic) mean of all data points $\{\mathbf{x}_j\}_{j=1}^N$, and $\boldsymbol{\mu}_k$ is the mean of the data points $\{\mathbf{x}_j : c(j) = k\}$ from the kth cluster. The matrices $\mathbf{S}_B$ and $\mathbf{S}_W$ are known as the inter-cluster and the intra-cluster scatter matrices, respectively. Setting the derivative $\frac{dJ(\mathbf{q})}{d\mathbf{q}} = 0$ leads to the

optimality condition:

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{q} = J(\mathbf{q})\mathbf{q}$$

which can be solved by computing the singular value decomposition (SVD) of $\mathbf{S}_W^{-1}\mathbf{S}_B$ and finding the largest $s < d$ eigenvectors.

To increase the robustness of the selections, it is a common practice to first pick the data points close to the cluster centers (based on the given clustering assignment) and use the selected data points to form the inter-cluster and the intra-cluster scatter matrices. In our implementations, we pick the top 75% closest data points to the center in each cluster. We consider this data selection process as part of the Fisher LDA.

4.7 Experimental Results

In this section, we use one set of synthetic data and four sets of real-world data to benchmark the performance of the proposed convex clustering with metric learning (CCML) and convex clustering with sparse compositional metric learning (CCSCML) against the convex clustering (CC) of Chi and Lange [15], the robust convex clustering (RCC) of Wang et al. [26], and sparse convex clustering (SCC) [27]. When applying the ADMM to solve the various convex clustering problems, we use the following choices for the tuning parameters:

- In our implementations, we use the k-nearest neighbor method to determine the weighting coefficients $w_{\{j_1,j_2\}}$ [15]. More specifically, we choose the weighting coefficient $w_{\{j_1,j_2\}}$ between the data points $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$ as:

$$w_{\{j_1,j_2\}} = \iota^k_{\{j_1,j_2\}} \exp\left[-\alpha\|\mathbf{x}_{j_1} - \mathbf{x}_{j_2}\|_2^2\right],$$

  where $\iota^k_{\{j_1,j_2\}}$ is 1 if both $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$ are among the kth nearest neighbors (under the Euclidean distance metric) of each other and 0 otherwise, α is a nonnegative real constant, and k is a natural number. Note that setting $\alpha = 0$ gives uniform weights between the data points

among the k-nearest neighbors of each other. In our implementations, however, we tune α as a small positive number to improve the clustering accuracy. Following [15] and [16], we choose the value of k in our numerical experiments as the expected average cluster size.

- Note if we set the tuning parameter $\gamma = 0$, this will lead to the trivial solution of partitioning the data points into singletons. On the other hand, if we set γ to be sufficiently large, this will lump all the data points into a single cluster. Varying γ in between gives rise to the entire clustering path. In our numerical experiments, we choose γ so that the results match the expected number of clusters. Even though mathematically there seems to be no guarantee that this is always possible, we were able to achieve the exact matches in all of our numerical experiments. The convergence of the ADMM algorithms does not appear to be sensitive to the choice of the augmentation parameter ν.

In addition, we use the well known Rand index [50] to measure the accuracy of the clustering results. More specifically, for each set of testing data the ground truth is known and from that we can construct an N-by-N binary ground truth adjacency matrix $\bar{\mathbf{A}}$ with entries $\bar{\mathbf{A}}_{i,j}$ = 1 if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same cluster and 0 otherwise. For a given output of a convex clustering algorithm, we look at the columns of the matrix $\mathbf{V}$, i.e., the difference variables $\mathbf{v}_\ell$. If $\mathbf{v}_\ell = \mathbf{0}$ (or close to $\mathbf{0}$ within the numerical accuracy), we set $\tilde{\mathbf{A}}_{\phi_1(\ell),\phi_2(\ell)} = \tilde{\mathbf{A}}_{\phi_2(\ell),\phi_1(\ell)} = 1$; otherwise, we set $\tilde{\mathbf{A}}_{\phi_1(\ell),\phi_2(\ell)} = \tilde{\mathbf{A}}_{\phi_2(\ell),\phi_1(\ell)} = 0$. The clustering accuracy is then calculated by counting the number of matching values in the upper triangles (excluding the diagonal entries) of $\bar{\mathbf{A}}$ and $\tilde{\mathbf{A}}$, normalized by the total number of adjacency pairs $N(N-1)/2$.

4.7.1 Synthetic data

We considered the same synthetic data set that had been used in [16], which was generated based on the standard Gaussian mixture model (GMM). More specifically, three classes of data in $\mathbb{R}^3$ were generated, with 100 data points in each class. All data points were generated using the same variance but different means for different classes. Then, outlier feature values were added to each of the data points, making each data point a higher-dimensional vector. Each outlier

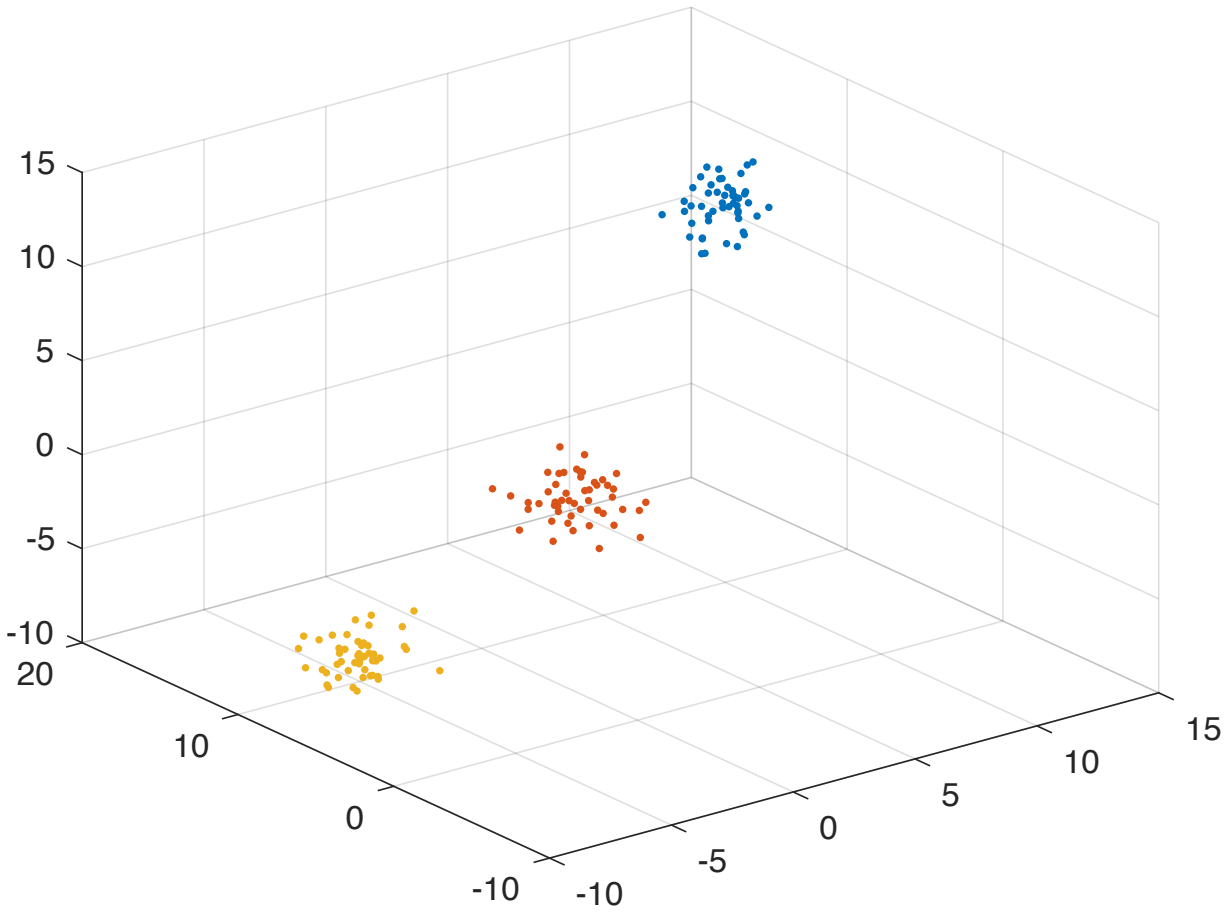


Figure 4.1: Example of synthetic data before outlier features were added. [3]

feature value was generated independently using an identical distribution across all 300 data points. Different distributions of high variance were used for different outlier features. An example of the GMM data before outlier features were added can be seen on Figure 4.1.

Figure 4.2 compares the clustering accuracies of CC, RCC, SCC, CCML, and CCSCML under different numbers of outlier features (from 0 to 7). Each data point was calculated based on the average of 50 experiments, and for each experiment the tuning parameters are tuned such that the number of clusters matches that of the ground truth and the achieved clustering accuracy is highest possible. For CCSCML, the number of orthonormal vectors obtained from the Fisher LDA

[3]Reprinted with permission from X. Sui, X. Li, X. Qian, and T. Liu, "Convex clustering with metric learning," *Pattern Recognition, vol. 81, pp. 575-584, September 2018*

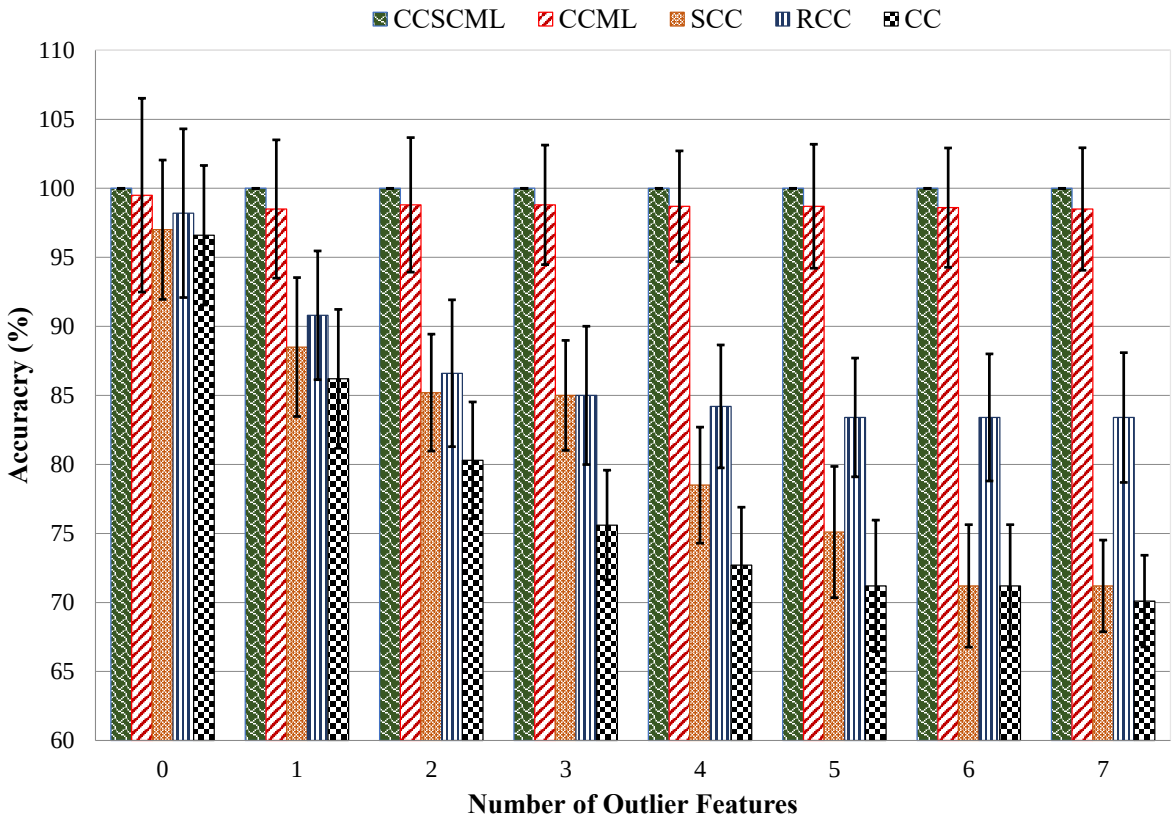


Figure 4.2: Gaussian GMM data: Clustering accuracy as a function of the number of outlier features. The narrow line on top of each bar indicates the standard deviation for each set of experiments.

s was set as three to match the number of relevant features in the ground truth. As illustrated, the accuracies of CC, RCC, SCC and CCML all decrease (to various extent) with the number of outlier features. By comparison, CCSCML appears to be very robust to outlier features and can achieve near 100% accuracies under *all* configurations.

Figure 4.3, figure 4.4, and figure 4.5 illustrates the clustering accuracy and the minimum value of the optimization problem (again averaged over 50 experiments) as a function of the number of iterations for RCC, CCML, and CCSCML respectively. Here, the number of outlier features was chosen as seven (so the total dimension of the data was ten). As illustrated, the proposed iterative algorithm appears to converge within three iterations. Under the same setting, Figure 4.6 illustrates

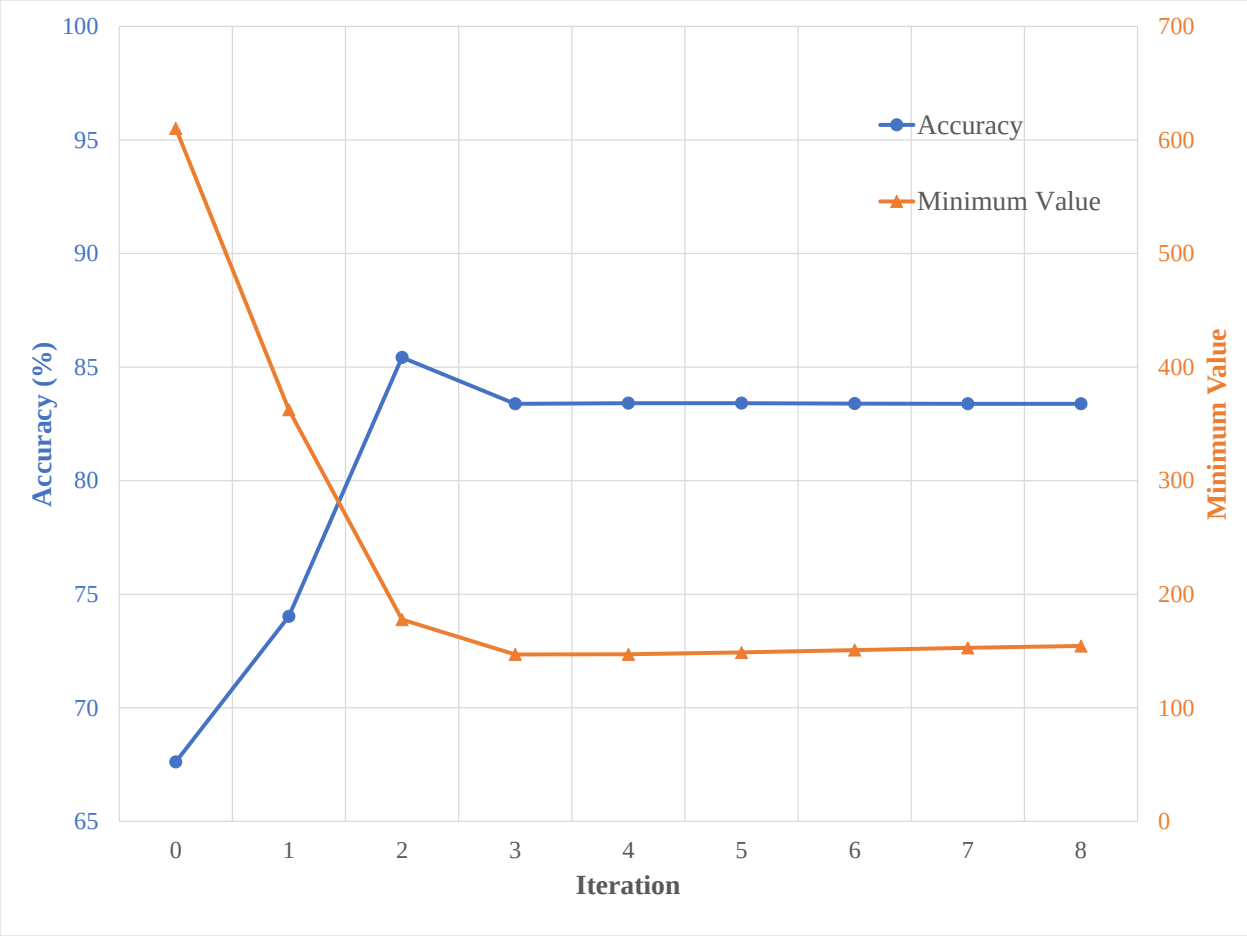


Figure 4.3: Gaussian GMM data: Convergence of the clustering accuracy and the minimum value for RCC.

the cumulative running time as a function of the number of iterations for RCC, CCML, and CC-SCML (the three iterative algorithms). All three algorithms use CC to obtain an initial clustering solution. However, the running time per iteration afterwards is much smaller for CCSCML than for RCC and CCML. This is mainly due to the fact that while RCC and CCML run full-dimensional convex clustering in each of their iterations, the convex clustering algorithm for CCSCML runs over the projected data (after initialization), which has a much smaller dimension.

Figure 4.7 illustrates the intensity map and the singular values of the full rank metric $\mathbf{B}$ learned from the final iteration for a particular experiment on a set of simulated data. As illustrated, the Mahalanobis distance metric $\mathbf{B}$ learned from the final iteration can successfully identify the three

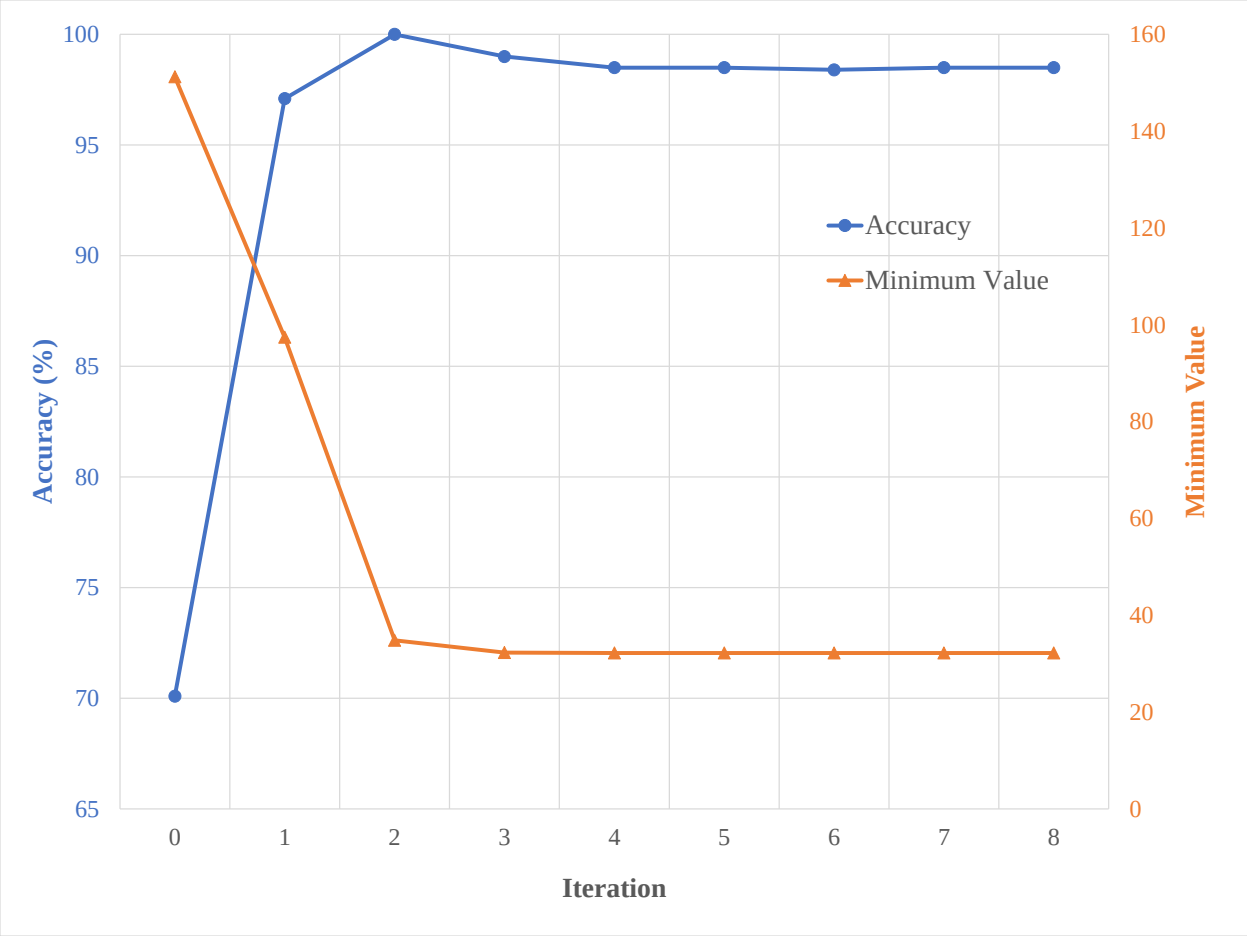


Figure 4.4: Gaussian GMM data: Convergence of the clustering accuracy and the minimum value for CCML.

highly relevant features (the first three features) of the data.

4.7.2 Real-world data

Table 4.1: The basic parameters of four real-world data sets.

	# of samples	# of features	# of clusters
“Seeds"	210	7	3
“Wine"	178	13	3
“Images"	2310	19	7
DLBCL	321	661	3

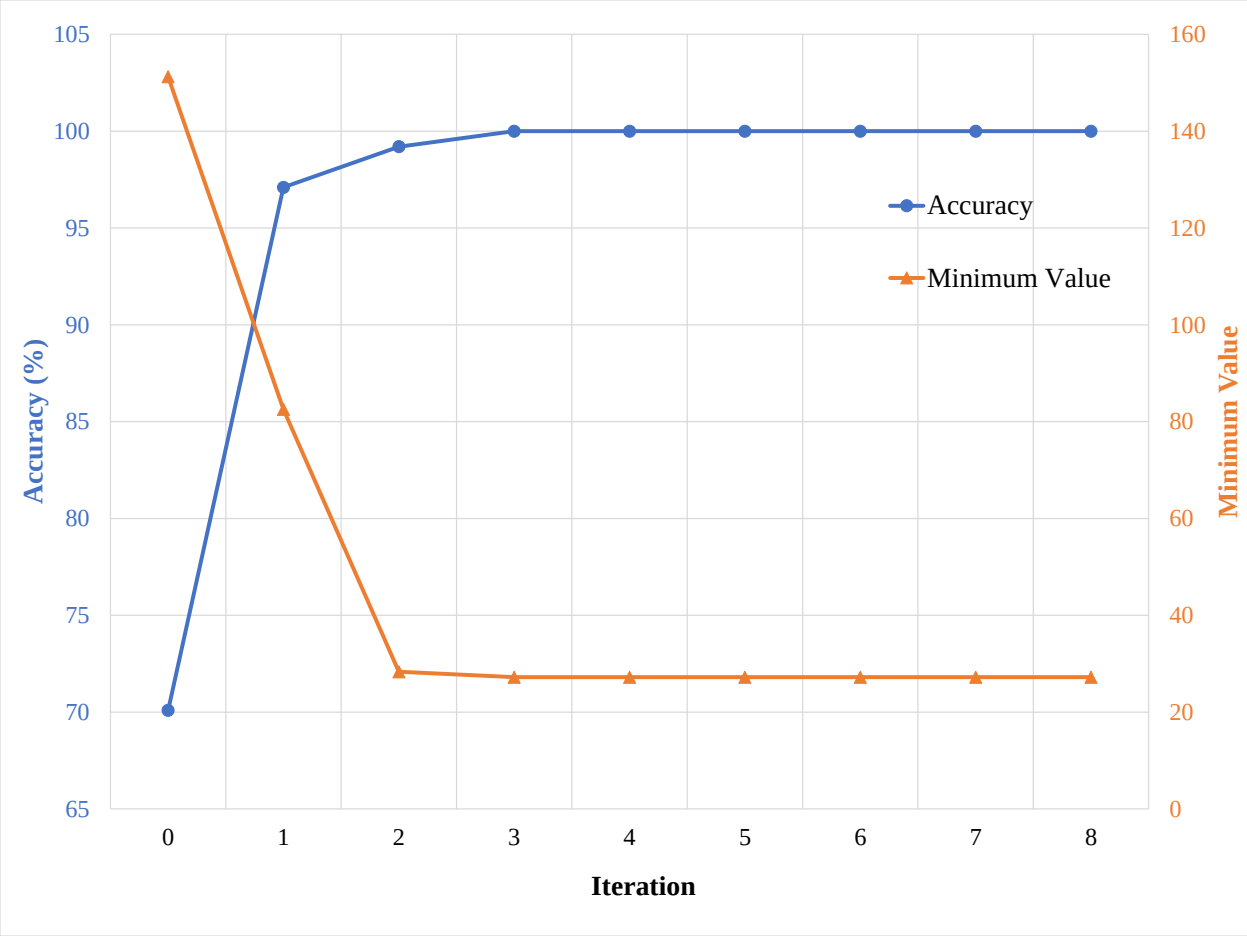


Figure 4.5: Gaussian GMM data: Convergence of the clustering accuracy and the minimum value for CCSCML.

Table 4.2: The clustering accuracies of various clustering algorithms for four real-world data sets.

	k-means	N-cut	CC	RCC	SCC	CCML	CCSCML
“Seeds"	72.5	73.3	72.0	75.4	72.5	75.6	80.5
“Wine"	70.0	70.1	65.4	67.2	69.0	71.5	72.9
“Images"	73.4	72.0	80.5	83.0	81.7	82.2	86.0
DLBCL	49.2	54.5	65.2	67.5	65.5	68.2	70.2

We also tested the performance of CC, RCC, CCML and CCSCML (and the more traditional k-means [6] and normalized-cut [5] algorithms) using the real-world data sets “seeds", “wine", and “images" from the UCI machine learning repository [29] and the higher-dimensional Diffuse

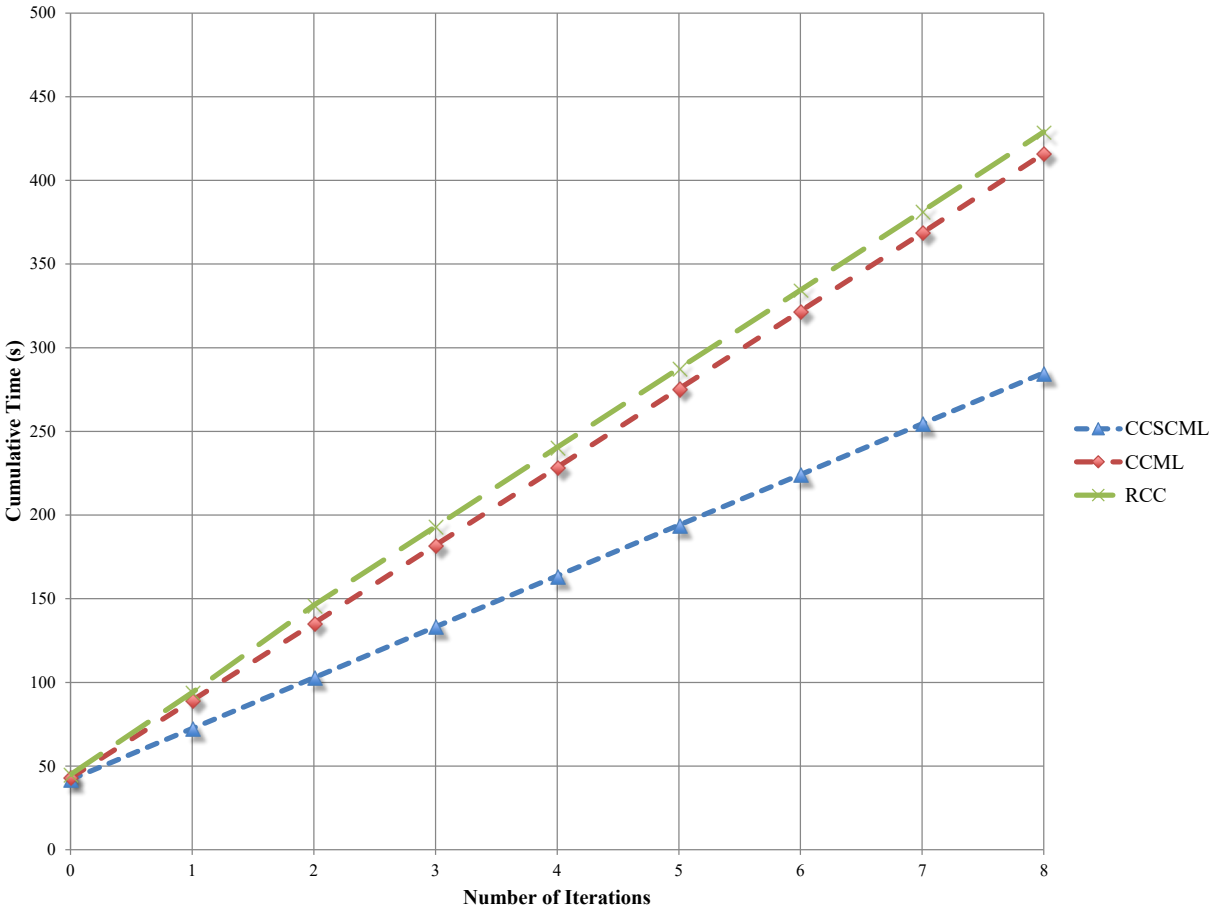


Figure 4.6: Gaussian GMM data: Cumulative running time as a function of the number of iterations. The algorithms were implemented using MATLAB R2017b on a Windows 10 PC with an Intel Core i7 2.8 GHz processor.

Large B-cell Lymphoma (DLBCL) data set [51]:

- The “seeds" data set contains the measurements of geometrical properties of seeds belonging to three different types of wheat. There are 70 samples for each of the three classes. The three classes of wheat are Kama, Rosa, and Canadian. A soft X-ray technique was used to image the seed samples, and seven real-valued features were extracted from the X-ray images.

- The “wine" data set contains the results of a chemical analysis of wines grown in the same region of Italy, but derived from three different cultivars. There are 59, 71, and 48 samples

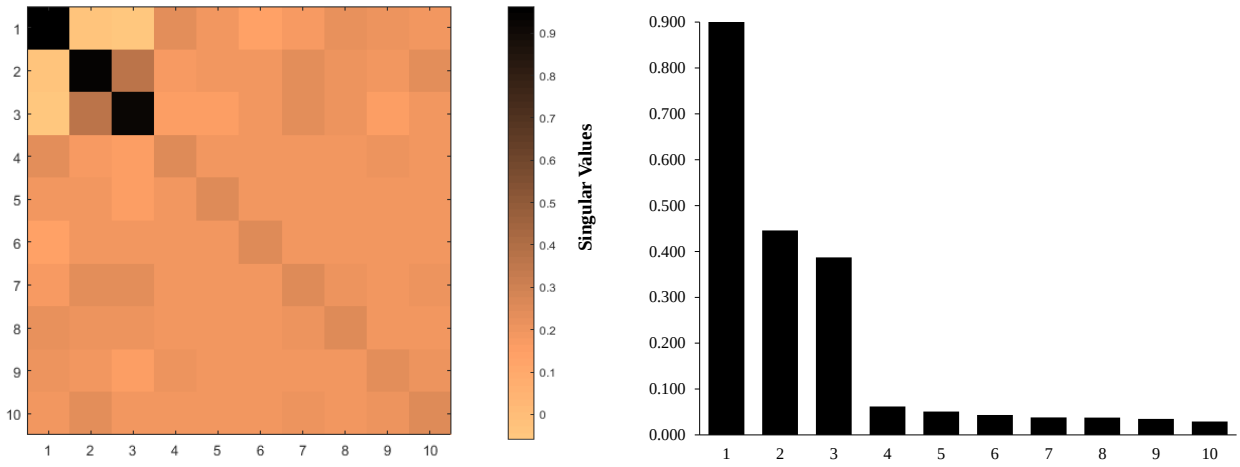


Figure 4.7: Gaussian GMM data: Intensity map and the singular values of the full rank metric $\mathbf{B}$ learned from the final iteration. [4]

in each of the three cultivars, respectively. Wines grown in the same cultivar are considered to be similar to each other. There are 13 features in this data.

- The “images" data set contains images of seven different classes of images, each with a different subject. The subjects are brick-face, sky, foliage, cement, window, path, and grass. There are 330 data points in each of the seven classes. Each image was hand-segmented into 3-by-3 regions, from which 19 features were extracted.

- The Diffuse Large B-Cell Lymphoma (DLBCL) data set contains 321 samples of gene expressions from three sub-types of Lymphoma cancer. The clustering goal is to find the clusters according to tissue and cancer types, conditioned on the generation of micro-array platforms. The three clustered sub types are designated as oxidative phosphorylation (OxPhos), B-cell response (BCR), and host response (HR) according to relevant molecular mechanisms. There are 661 features representing various gene expressions. Results from this data set is comparable to results found in recent clustering publication [52].

[4]Reprinted with permission from X. Sui, X. Li, X. Qian, and T. Liu, "Convex clustering with metric learning," *Pattern Recognition, vol. 81, pp. 575-584, September 2018*

The basic parameters of the above data sets are summarized in Table 4.1, and the exact choices of the features can be found in [29] and [51].

Table 4.2 lists the clustering accuracy of the k-means, normalized-cut, CC, RCC, SCC, CCML, and CCSCML algorithms for the four real-world data sets mentioned above. As illustrated, CCSCML performs consistently as *the* best among the algorithms considered (all accuracy numbers are state-of-the-art to the best of our knowledge).

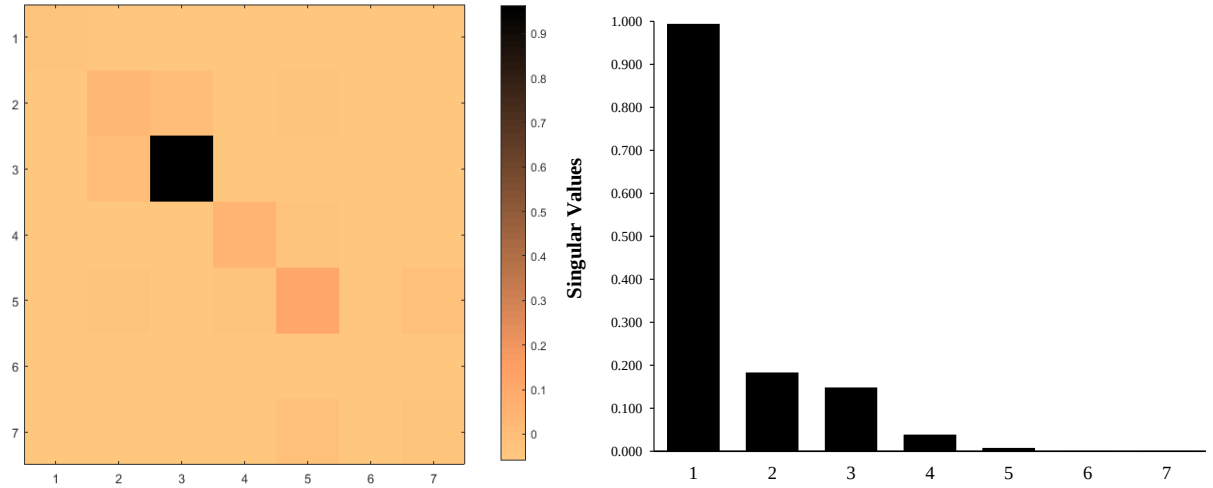
Figure 4.8 illustrates the intensity map and the singular values of the full rank distance metric $\mathbf{B}$ learned from the final iteration for "seeds", "wine", and "images" data sets. From these plots, we can identify that: 1) or the "seeds" data, the third feature "Compactness" is clearly the most relevant one to this clustering; 2) for the "wine" data, the eighth feature "Nonflavenoids phenols" is the most relevant one to this clustering, and the third and the eleventh features "Ash" and "Hue" are also highly relevant to this clustering; 3) for the "image" data, the third, tenth, eleventh, twelfth, and thirteenth features "Region-pixel-count", "Intensity-mean", "Raw-red-mean", "Raw-blue-mean", and "Raw-green-mean" are the most relevant ones to this clustering.

For CCSCML, the number of orthonormal vectors obtained from the Fisher LDA s was set as 5, 2, 5, and 200 for the "seeds", "wine", "images", and DLBCL data sets, respectively. The number of orthonormal vectors was determined by looking at the significant eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$ while performing the initial Fisher LDA (see Figure 4.9 for the ordered eigenvalues of the four data sets).

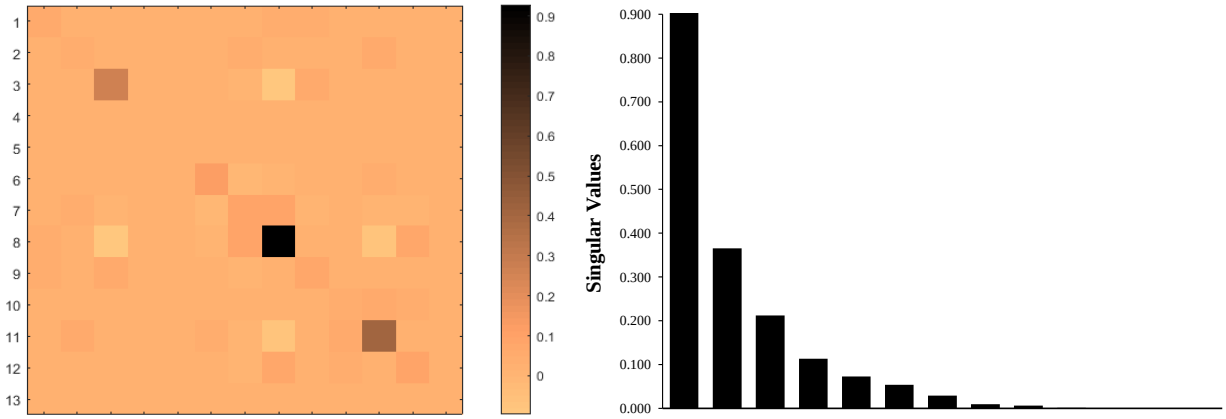
4.8 Conclusion

The first work showed that metric learning can significantly improve the performance of convex clustering. However, the use of a full-dimensional Mahalanobis distance metric can lead to high computational complexity for high-dimensional data sets and incur performance loss in the presence of outlier features. Motivated by this, in our subsequent work we show that both issues can be effectively addressed by imposing a sparse compositional structure on the Mahalanobis distance metric. Numerical experiments based on both synthetic and real-world data sets demonstrated that the proposed algorithm can significantly outperform all previous convex clustering

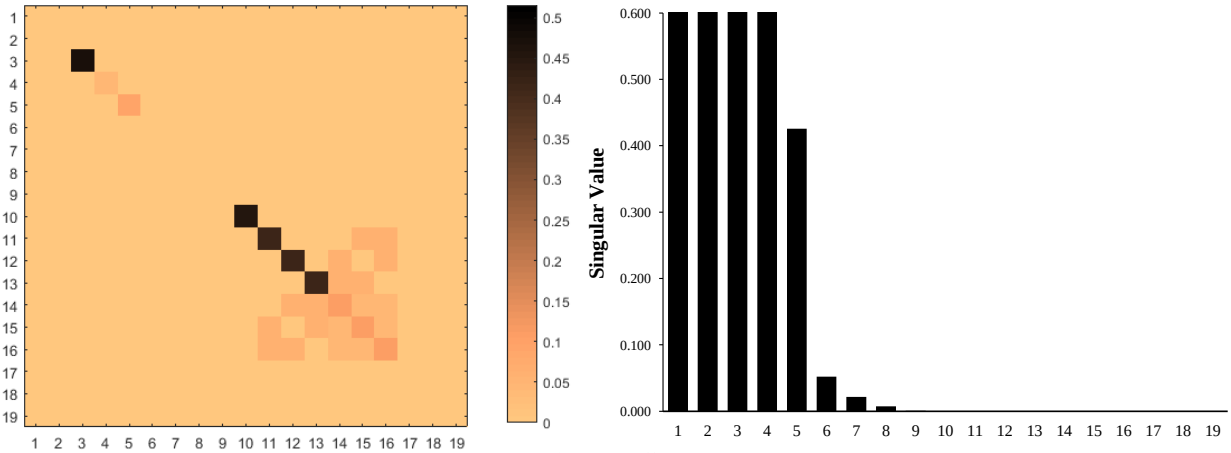
algorithms [15, 16, 26] as well as the more classical k-means [6] and normalize-cut [5] algorithms.



(a) "Seeds"



(b) "Wine"



(c) "Images"

Figure 4.8: Real-world data sets: Intensity map and the singular values of the full rank metric $\mathbf{B}$ learned from the final iteration.[5]

[5]Reprinted with permission from X. Sui, X. Li, X. Qian, and T. Liu, "Convex clustering with metric learning," *Pattern Recognition, vol. 81, pp. 575-584, September 2018*

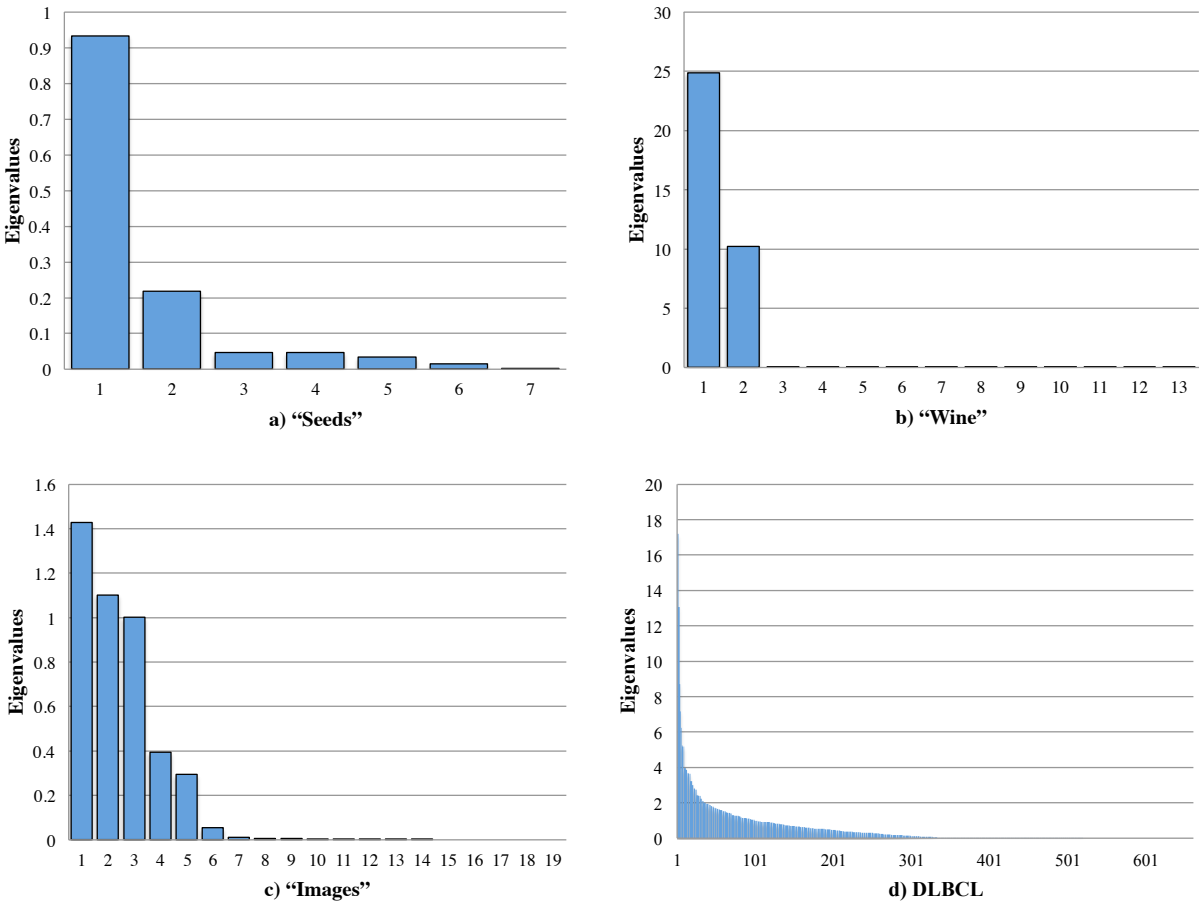


Figure 4.9: The ordered eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$ in the Fisher LDA.

5. FEATURE SELECTION

5.1 Introduction

Clustering is considered as a method in unsupervised learning. However, with the addition of metric learning and the iterative algorithm between convex clustering and metric learning, the works in the previous chapter can be consider as semi-supervised learning. In particular, in the sparse compositoinal scheme, metric learning served as a way of dimension reduction, or feature selection. In many pattern recognition applications, identifying the most meaningful features in the data set is critical in the performance of clustering and classification methods. Given data set with N data points of the dimension d denoted as $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, and the target classification variable Y, the feature selection problem aims to find from d-dimensional observation space a subspace of s features where $s < d$, that best characterizes Y. The set of all features can be denoted as V with each feature denoted as X_i, the subset of features that best represent the data in the classification scheme is denoted as B.

Given a set of training data, the feature selection algorithm is a search algorithm that searches for the best subspace that fits the training data and its classification outputs. The total number of subspaces is thus 2^d, and the number of the subspace with dimension smaller than or equal to s is $\sum_{i=1}^{d} \binom{d}{i}$, it thus difficult to exhaustively search the entire subspace. Therefore, the feature selection problem is predominantly solved using a sequential-search-based method. In a sequential-search-based method, each feature is evaluated one after another on three factors: relevancy, redundancy, and interaction. Relevancy is a measure between the feature and the outcome's variables dependency on its variance. However, some features might be redundant, as in they contribute the same information towards the outcome variable. Lastly, the effect of certain features on the outcome variable can also be dependent on its interaction with each other.

In the section below, two classical methods of feature selection are described. Both algorithms are greedy methods that searches incrementally the features with the most relevance, least redun-

dancy while taking interaction into account.

5.2 Related Works

By far, most previous works relies on the heuristic of maximum relevance and minimum redundancy. Denoting V as the full set of features and B as a small subest of feature that is desired, let r_{X_1,X_2} be a correlation measure between two discrete random variables X_1 and X_2 in V, previous literatures use averages such as:

- The average relevance between the features in a subset and the outcome:

$$\bar{r}_{xy}(B) := \frac{1}{|B|} \sum_{i \in B} r_{\mathsf{X_i},Y}$$

- The average redundancy between the features in a subset:

$$\bar{r}_{xx}(B) := \frac{1}{|B|(|B|-1)} \sum_{(i,j) \in B^2 : i \neq j} r_{\mathsf{X}_i,\mathsf{X}_j}$$

To address the algorithmic challenge, it is common to perform an incremental search over the entire feature set via a greedy algorithm.

5.2.1 Minimum Redundancy Maximum Relevance (mRMR)

The immensely popular Minimum Redundancy Maximum Relevance (mRMR) [24] algorithm is based on the following feature selection criteria:

$$\bar{r}_{xy}(B) - \bar{r}_{xx}(B) \tag{5.1}$$

where the correlation measure is chosen as Shannon's mutual information. The mRMR algorithm is an incremental search greedy algorithm. It starts with B as an empty set, and the first feature chosen is the feature with the highest mutual information between the feature and the outcome

$I(\mathsf{X}_i, \mathsf{Y})$. At the mth feature to select, the selection criteria is

$$\max_{X_j \in X_V - S_{m-1}} I(\mathsf{X}_i, \mathsf{Y}) - \frac{1}{m-1} \sum_{X_i \in S_{m-1}} I(\mathsf{X}_i, \mathsf{Y}) \tag{5.2}$$

where S_{m-1} is the set of features of size $(m-1)$ that have already been selected.

5.2.2 Correlation Based Feature Selection (CFS)

Another set of popular feature selection algorithms known as the Correlation Based Feature Selection (CFS) [25] are based on the following criteria:

$$\frac{|B|\bar{r}_{xy}(B)}{\sqrt{|B| + |B|(|B|-1)\bar{r}_{xx}(B)}} \tag{5.3}$$

where the correlation measure used is called symmetric uncertainty:

$$r_{xy} = \frac{2I(\mathsf{X}, \mathsf{Y})}{H(\mathsf{X}) + H(\mathsf{Y})} \tag{5.4}$$

The CFS algorithm is also an incremental search greedy algorithm. It starts with B as an empty set, and the first feature chosen is the feature with the highest symmetric uncertainty between the feature and the outcome $r(X_i, Y)$. At the mth feature to select, the selection criteria is

$$\max_{X_j \in X_V - S_{m-1}} \frac{m\bar{r_{cf}}}{\sqrt{m + m(m - s1)\bar{r_{ff}}}} \tag{5.5}$$

where S_{m-1} is the set of features of size $(m-1)$ that have already been selected, $\bar{r_{cf}}$ is the mean of already selected feature-to-class symmetric uncertainty, and $\bar{r_{ff}}$ is the mean of all pairwise feature-to-feature symmetric uncertainty.

5.3 Conclusion

Feature selection is a supervised learning pattern recognition method. Given a set of data with d features, denoted as V, selecting a small set of features, B, can not only reduce the dimension of the data set but can eliminate irrelevant and redundant features in the data set to produce a better

result. Previous classical feature selection methods uses a greedy incremental search method that grows the set B one feature at a time with a selection criteria. However, these selection criteria are often heuristic. In our work in the next section we use a direct approximation of Shannon's mutual information on the underlying joint distribution.

6. FEATURE SELECTION USING CHOW-LIU TREE APPROXIMATION

6.1 Introduction

Feature selection is a fundamental problem in machine learning. Formally, let $(\mathsf{X}_V, \mathsf{Y})$ be a collection of jointly distributed discrete random variables, where $\mathsf{X}_V := (\mathsf{X}_i : i \in V)$ are the features and Y is the outcome variable. The goal is to find the small subset $B \subseteq V$ of features that are highly relevant to the outcome based on a given set of identically and independently distributed drawn samples of$(\mathsf{X}_V, \mathsf{Y})$. By eliminating irrelevant and redundant features, feature selection not only helps to reduce the complexity of the training algorithm, but can also help improve the interpret-ability of the accuracy of the learned model.

This work focuses on the so-called filter method (for with the selection of features is independent of the training algorithm), which is known to be less prone to over-fitting than the so-called wrapper method (for which feature selection and training are performed jointly) [53]. Traditionally, information theory plays an important role in addressing the filter feature selection problem. In particular, Shannon's mutual information provides a well-accepted correlation measure, which can capture both linear and nonlinear dependencies between two groups of random variables. Thus, assuming that the joining distribution of $(\mathsf{X}_V, \mathsf{Y})$ is known, a natural formation of the filter feature selection problem is given by:

$$\max_{B \subseteq V : |B| = k} I(\mathsf{X}_B; \mathsf{Y}) \tag{6.1}$$

for some fixed inter k (which is usually much smaller than $|V|$), where $I(\mathsf{X}_B; \mathsf{Y})$ denotes the Shannon mutual information between the features X_B and the outcome variable Y.

By far, most of the literature relies on the heuristic of maximum relevance and minimum redundancy, such as the popular mRMR algorithm [24] and the CFS [25] algorithm. This work is a continued effort to seek an alternative feature-selection criteria that can be efficiently computed from the lower-order marginals of $(\mathsf{X}_V, \mathsf{Y})$. Instead of following the maximum relevance minimum redundancy principles, we provide a direct approximation on the Shannon mutual infor-

mation $I(\mathsf{X}_V;\mathsf{Y})$ by using the well-known Chow Liu tree (CLT) approximations [54]. This led to a new feature-selection criteria which can be efficiently computed from the pairwise mutual information between the features and the outcome variable. There are two major challenges for solving the optimization problem in (6.1):

1. From the algorithm point of view, the Shannon mutual information $I(\mathsf{X}_B;\mathsf{Y})$ as a set function on B has no known structures.

2. From the practical viewpoint, the Shannon mutual information has to be estimated from the given set of data samples, and that can be computationally challenging.

To address the algorithmic challenge, a common practice is to perform an incremental search over the entire set of features. For example, given a set $B \subseteq V$ of features that have already been chosen, a new feature can be greedily added via solving:

$$\max_{i\in V\setminus B} I(\mathsf{X}_{B\cup\{i\}};\mathsf{Y}) \tag{6.2}$$

6.2 Chow-Liu Tree Approximation

In our approach, instead of heuristic measures we adopt to use a Chow-Liu Tree approximation. For a collection of jointly distributed discrete random variables $\mathsf{X}_V = (\mathsf{X}_i, i \in V)$, consider a tree T with vertex set V. A dependency-tree approximation of X_V, denoted as X_V^T, can be written as a joint distribution of X_B where $|B| \leq 2$:

$$P_{\mathsf{X}_V^T}(x_V) := \left(\prod_{i\in V} P_{\mathsf{X_i}}(x_i)\right) \prod_{(i,j)\in\mathcal{E}(T)} \frac{P_{\mathsf{X_i},\mathsf{X_j}}(x_i,x_j)}{P_{\mathsf{X_i}}(x_i)P_{\mathsf{X_j}}(x_j)} \tag{6.3}$$

for any $x_V \in \mathsf{X}_V$ and $\mathcal{E}(T)$ is the edge set of T. Such a distribution forms a Markov tree or a Bayesian network with respect to T thus we can relabel the indices in V

$$P_{\mathsf{X}_V^T}(x_V) := \prod_{i\in V} P_{\mathsf{X_i}|\mathsf{X}_{p_i}}(x_i|x_{p_i}) \tag{6.4}$$

where $p_1 = \emptyset, p_i < i$, and $\{i, p_i\} \in \mathcal{E}(T)$ for $i > 1$.

A set of Chow-Liu tree is defined as:

$$\mathcal{T}^*(\mathsf{X}_V) := \arg\min_T D(P_{\mathsf{X}_V} || P_{\mathsf{X}_V^T}) \tag{6.5}$$

where $D(P_{\mathsf{X}_V} || P_{\mathsf{X}_V^T})$ is the divergence between X_V and a dependency-tree approximation X_V^T [54]. It can be shown that

$$D(P_{X_V} || P_{\mathsf{X}_V^T}) = D\bigg(P_{\mathsf{X}_V} || \prod_{i \in V} P_{\mathsf{X}_i}\bigg) - \sum_{\{i,j\} \in \mathcal{E}(T)} I(\mathsf{X}_i; \mathsf{X}_j) \tag{6.6}$$

where

$$D\bigg(P_{\mathsf{X}_V} || \prod_{i \in V} P_{\mathsf{X}_i}\bigg) = \sum_{i \in V} H(\mathsf{X}_i) - H(\mathsf{X_V}) \tag{6.7}$$

is known as the total correlation of X_V. In particular, for any X_V and any tree T with vertex set V:

$$D\bigg(P_{\mathsf{X}_V}^T || \prod_{i \in V} P_{\mathsf{X}_i}\bigg) = \sum_{\{i,j\} \in \mathcal{E}(T)} I(\mathsf{X}_i; \mathsf{X}_j). \tag{6.8}$$

The Chow-Liu algorithm computes a Chow-Liu tree as a maximum spanning tree with the Shannon mutual information $I(\mathsf{X}_i, \mathsf{X}_j)$ as the weight of the edge $\{i, j\}$, since the minimization in (6.5) corresponds to the maximizing the second term on the right had side of (6.6), which is the total weight of the tree T. Furthermore, by (6.8) we have,

$$D\bigg(P_{\mathsf{X}_V^T} || \prod_{i \in V} P_{\mathsf{X}_i}\bigg) = MST(\mathsf{X}_V) \tag{6.9}$$

where $MST(\mathsf{X}_V)$ is the total weight of a maximum spanning tree of X_V.

6.3 Feature Selection Algorithm

Having introduced Chow-Liu tree approximation, we can show that the information theory between a given set of features, $B \subseteq V$ and the outcome variable Y. Thus letting $T_1 \in \mathcal{T}^*(X_B)$

and $T_2 \in \mathcal{T}^*(X_B, Y)$ we can approximate $I(\mathsf{X}_B; \mathsf{Y})$:

$$
\begin{aligned}
I(\mathsf{X}_B; \mathsf{Y}) &= H(\mathsf{X}_B) + H(\mathsf{Y}) - H(\mathsf{X}_B; \mathsf{Y}); && (6.10)\\
&= \Big[\sum_{i\in B} H(\mathsf{X}_i) + H(\mathsf{Y}) - H(\mathsf{X}_B, \mathsf{Y})\Big] - \Big[\sum_{i\in B} H(\mathsf{X}_i) - H(\mathsf{X}_B)\Big] && (6.11)\\
&= D\Big(P_{\mathsf{X}_B,\mathsf{Y}} || \Big(\prod_{i\in B} P_{\mathsf{X}_i}\Big) P_{\mathsf{Y}}\Big) - D\Big(P_{\mathsf{X}_B} || \prod_{i\in B} P_{\mathsf{X}_i}\Big) && (6.12)\\
&\approx D\Big(P_{(\mathsf{X}_B,\mathsf{Y})^{T_2}} || \Big(\prod_{i\in B} P_{\mathsf{X}_i}\Big) P_{\mathsf{Y}}\Big) - D\Big(P_{\mathsf{X}_B^{T_1}} || \prod_{i\in B} P_{\mathsf{X}_i}\Big) && (6.13)\\
&= MST(\mathsf{X}_B, \mathsf{Y}) - MST(\mathsf{X}_B) && (6.14)
\end{aligned}
$$

where (6.12) follows from (6.7), (6.13) follows by approximating $P_{\mathsf{X}_B,\mathsf{Y}}$ and P_{X_B} by their respective Chow-Liu tree approximations, and (6.14) follows from (6.8). Note that the right-hand side of (6.13) can be computed from pairwise mutual information between the features and the outcome variable via standard maximum spanning tree algorithms [54].

Note that the new criteria (6.14) involves two separate Chow-Liu tree approximations: one for the joint distribution of $(\mathsf{X}_B, \mathsf{Y})$ and the other for the distribution of X_B only. By definition, both distributions are needed to compute the Shannon mutual information between X_B and Y. While the distribution of X_B can be obtain via marginalization of the distribution of $(\mathsf{X}_B, \mathsf{Y})$, it is known that the marginal distributions of a tree are not necessarily trees any more. Consider, for example, the situation where $\mathsf{X}_i, i \in B$ are independent given Y. In this case, the joint distribution (X, Y) is a star. However, the Shannon mutual information $I(\mathsf{X}_B; \mathsf{Y})$ cannot be efficiently computed from the conditional distribution $P_{\mathsf{X}_i|\mathsf{Y}}, i \in B$ and the marginal distribution of Y.

Inspired by the greedy algorithm of mRMR and of CFS and using the difference of two maximum spanning trees in (6.14) as the update parameter, we propose the feature selection algorithm in Algorithm 3. Given a set of training data with the set of features V, the outcome variable Y, as well as all the pairwise mutual information between each pair of the feature values and the features and outcome variable. The output of the feature selection algorithm is the set of most relevant and

least redundant variable set B of size k. The algorithm begins with B as an empty set, then adds features to the set one by one in a greedy fashion. In each step we construct the two maximum spanning tree and find the the different between their weight value. The feature with the largest different value is added to the set B.

Algorithm 3 Feature Selection via Chow-Liu Tree

Input: All pairwise mutual information for features and outcome, number of features to be selected k.
Output: Set of selected features B.
Initialize: $B = \emptyset$ and running maximum $M = -1$.
while $|B| < k$ **do**
    $M = -1$, $j = 0$.
    for $i \in V \backslash B$ **do**
        $T_1 = MST(X_B, X_i)$.
        $T_2 = MST(X_B, Y, X_i)$.
        if weight(T_2) $-$ weight(T_1) $> M$ **then**
            $j = i$.
            $M =$ weight(T_1) $-$ weight(T_2).
        end if
    end for
    Add X_j to B.
end while

6.4 Experimental Results

In this section, we use classification accuracy using the chosen features as an evaluation of our proposed feature selection algorithm against the well-known mRMR and CFS algorithm. The first step to the experimental tests on all three of the algorithms is to estimate the pairwise mutual information between all the features and the features and the outcome variable. To estimate the mutual information between two random variable, we first discretized the raw continuous data. Each feature variable was preprocessed to have zero mean-value and unit variance. The data is then discretized into three levels at the positions $\mu \pm \sigma$ (μ is the mean value and σ is the standard deviation): it takes the discrete value of -1 if it is less than $\mu - \sigma$, 1 if larger than $\mu + \sigma$, and 0 if it

is in between the two values. Having discretized the data, the entropies and joint entropies of each feature and the outcome is first estimated then the mutual information is calculated using

$$I(\mathsf{X}_i, \mathsf{X}_j) = H(\mathsf{X}_i) + H(\mathsf{X}_j) - H(\mathsf{X}_i, \mathsf{X}_j)$$

With a large training data set, we also did 10-fold cross validation on each of the three data sets: randomly choosing 10% of the data set as the training data and 90% of the data set as the testing data. After finding the pairwise mutual information of the training data set, we used the three feature selection algorithms to select a set B of features with the size of B varying between 5 to 50. For each set of selection features, those features are selection from the testing data. These small set of chosen features in the testing data is then used in classification algorithms to test the classification accuracy. Three classification algorithms were used: naive Bayesian (NB), support vector machines (SVM) and random forest (RF).

6.4.1 The Data Sets

Experiments were performed using three different real life data sets: arrhythmia (ARR) [29], HDR-MultiFeature (HDR) [29] and Columbia University Image Library (COIL-20) [55] data set. These three data sets are popular among other feature selection studies [24, 25].

The arrhythmia (ARR) data [29] contain 278 features and 420 samples. The attributes describe patients and the goal is to predict the presence and absence of cardiac arrhythmia. There are two classes, class 1 refers to "normal" cardiac functions and class 2 refers to having some kind of cardiac arrhythmia. The attributes or features are age, sex, height, weight, time duration of different waves in heart rhythms. The results of the ARR data using first three-level discretization, 10-fold cross validation and NB, SVM and RF classification are presented in Figure 6.1, Figure 6.2 and Figure 6.3, respectively.

The HDR-MultiFeature (HDR) data [29] contains 649 features for 2000 binary images of handwritten digits. This data set consists of images of handwritten numbers of 0 through 9, therefore there are 10 classes. The features are extracted from a collection of Dutch utility maps. There are

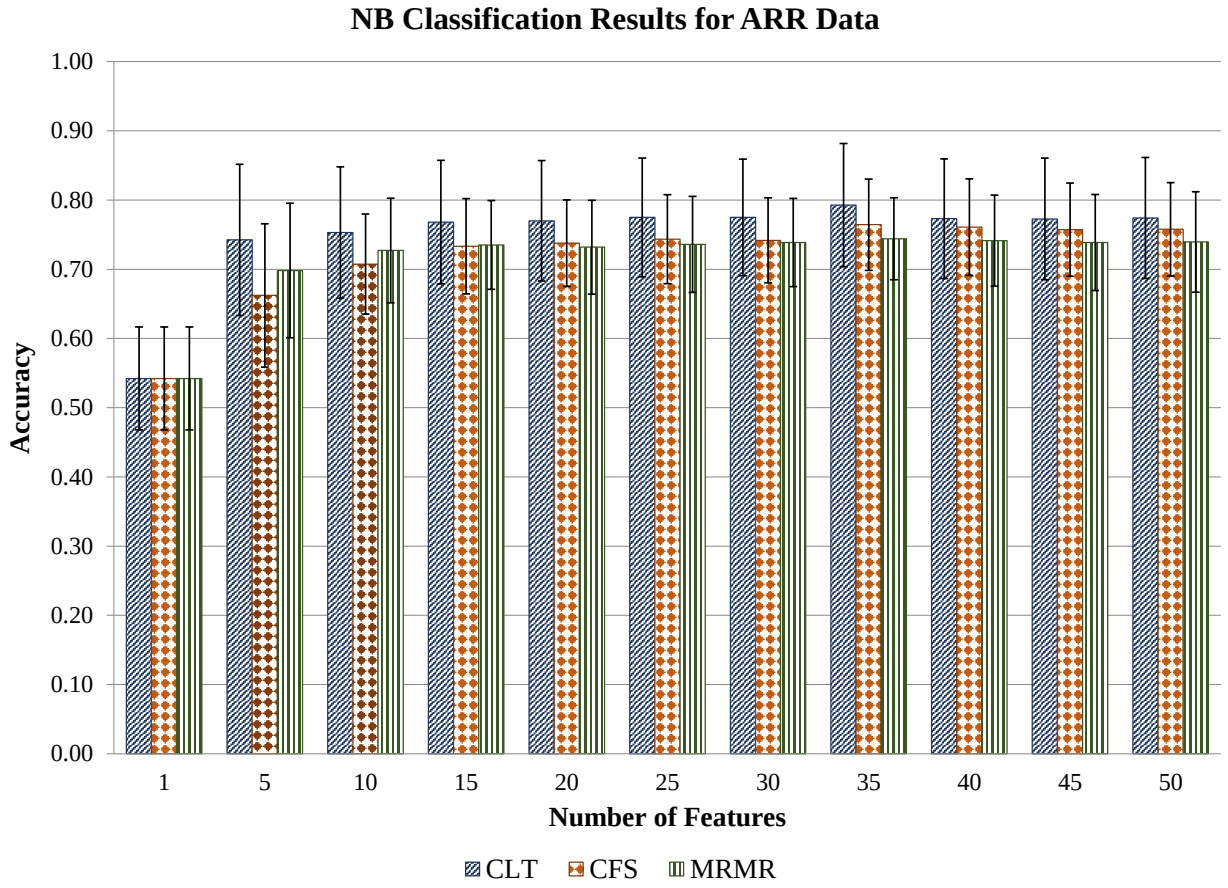


Figure 6.1: Classification accuracy result of ARR data using NB classification.

200 patterns per class with 2000 samples in total. The features include 76 Fourier coefficients of the character shapes, 216 profile correlations, 64 Karhunen-Love coefficients, 240 pixel average in a 2-by-3 windows, 47 Zernike moments, and 6 morphological features. The results of the HDR data using first three-level discritization, 10-fold cross validation and NB, SVM and RF classification are presented in Figure 6.4, Figure 6.5 and Figure 6.6, respectively.

The Columbia University Image Library (COIL-20) data set [55] is a database of gray-scale images of 20 objects. Thus, the classification has 20 classes. The objects were places on a motorized turntable against a black background. The turntable was rotated through 360 degrees to vary object pose with respect to a fixed camera. Images of the object were taken at pose intervals of 5 degrees. Therefore there are 72 images per object. Thus the total number of samples is 1440.

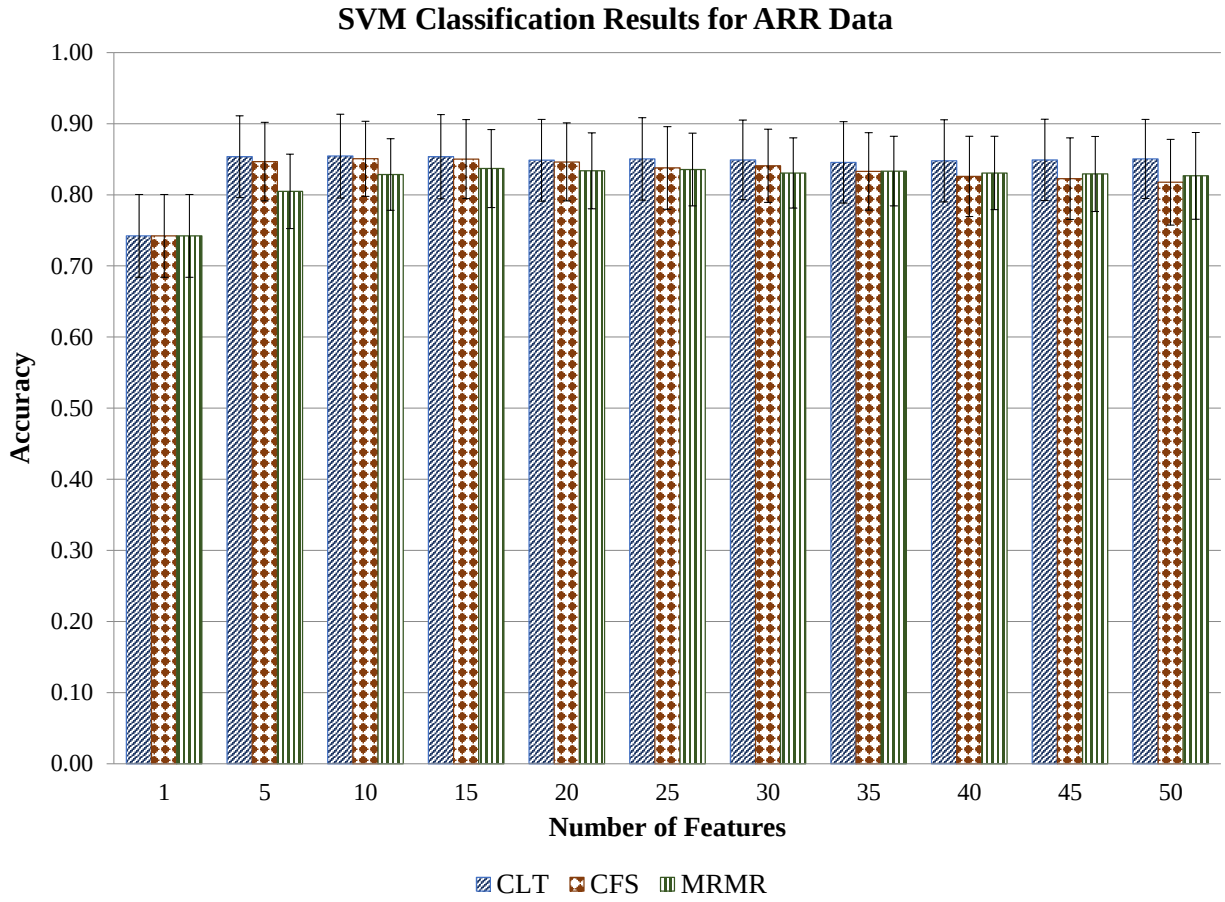


Figure 6.2: Classification accuracy result of ARR data using SVM classification.

Each image is 32-by-32 pixels, which means there are 1024 features each on a 256 gray scale level per pixel. Figure 6.7 shows a sample image from each of the 20 objects. The results of the COIL-20 data using first three-level discritization, 10-fold cross validation and NB, SVM and RF classification are presented in Figure 6.8, Figure 6.9 and Figure 6.10, respectively.

It can be seen in all 9 figures that the CLT feature selection algorithm outperforms both the mRMR and CFS feature selection algorithm. The results also show that despite having 278, 649 and 1024 features respectively in each of the three data sets, the classification accuracy stabilizes using less than 50 features in each case. This shows that feature selection is a much needed pre-processing procedure for classification.

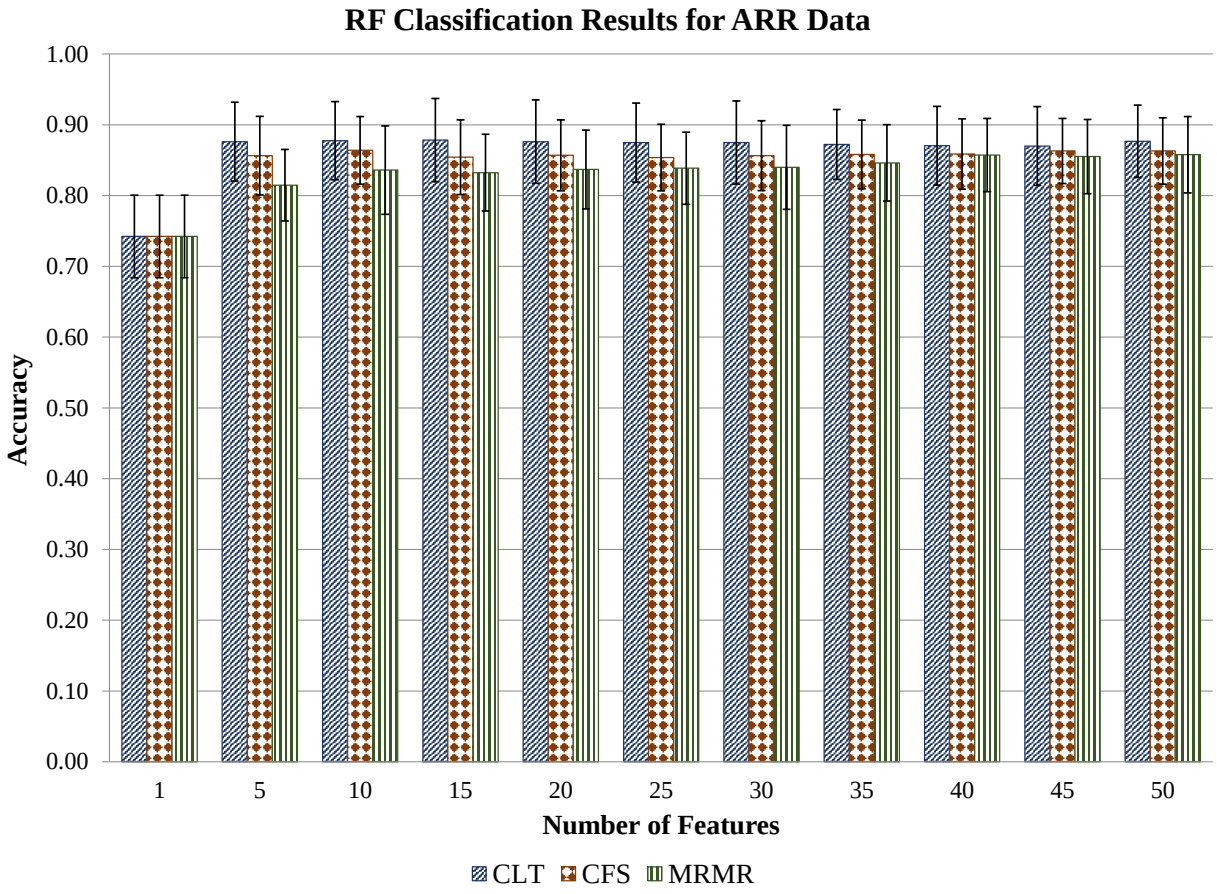


Figure 6.3: Classification accuracy result of ARR data using RF classification.

6.5 Conclusion

In this work, we proposed a new feature selection using Shannon's mutual information. The proposed algorithm is a greedy algorithm. At each iteration of the greedy algorithm the mutual information between a subset $B \subset V$ of the full set of features and the out come variable Y. This mutual information between a set of random variable and outcome random variable is computationally difficult to estimate. To estimate the mutual information, the distribution of X_B can be obtain via marginalization of the joint distribution of $(\mathsf{X}_B, \mathsf{Y})$. To achieve this, a Chow-Liu tree approximation was used. The mutual information $I(\mathsf{X}_B, \mathsf{Y})$ is estimated to be the different between two maximum spanning trees: one of X_B and one of $(\mathsf{X}_B, \mathsf{Y})$. Classification accuracy was used to test

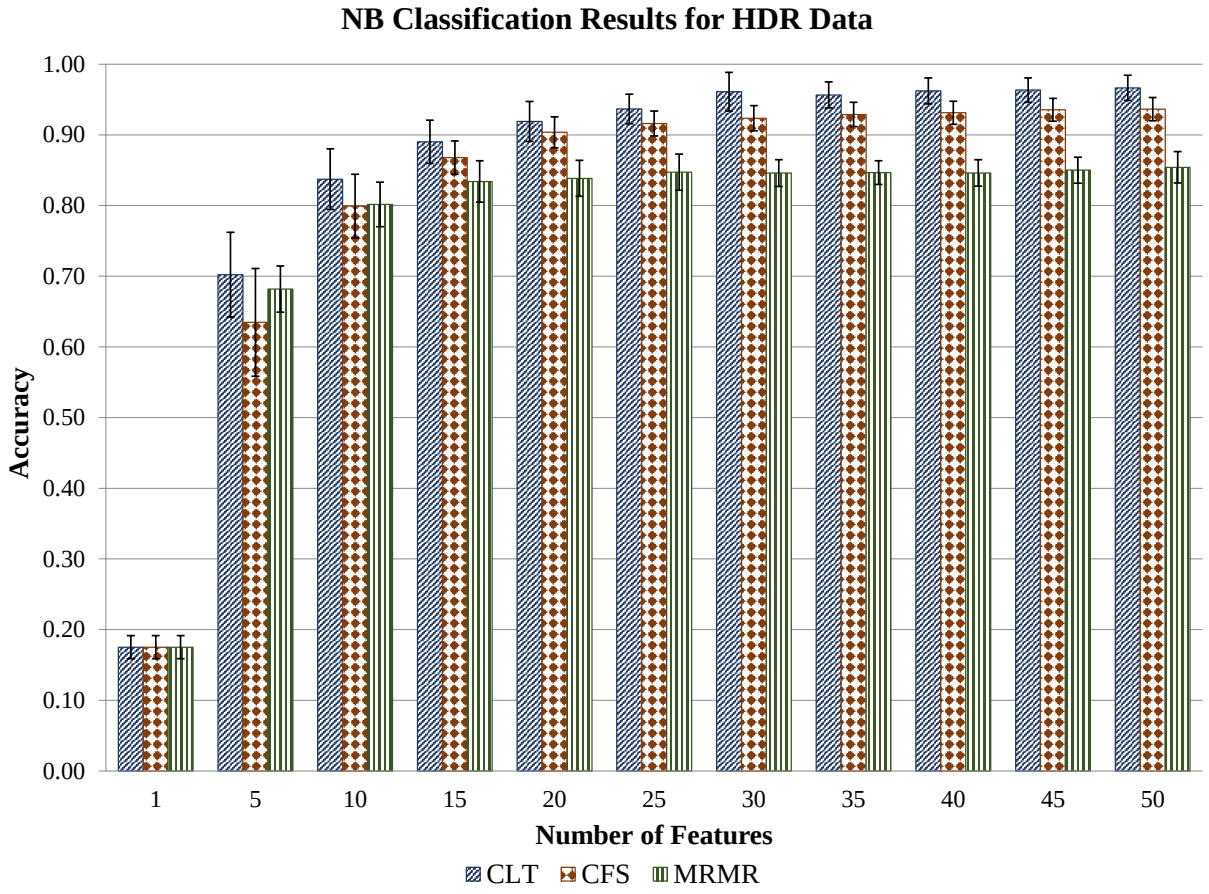


Figure 6.4: Classification accuracy result of HDR data using NB classification.

the effectiveness of the feature selection algorithm. Using three data sets and three classification algorithms, the proposed CLT algorithm outperforms previous feature selection algorithms.

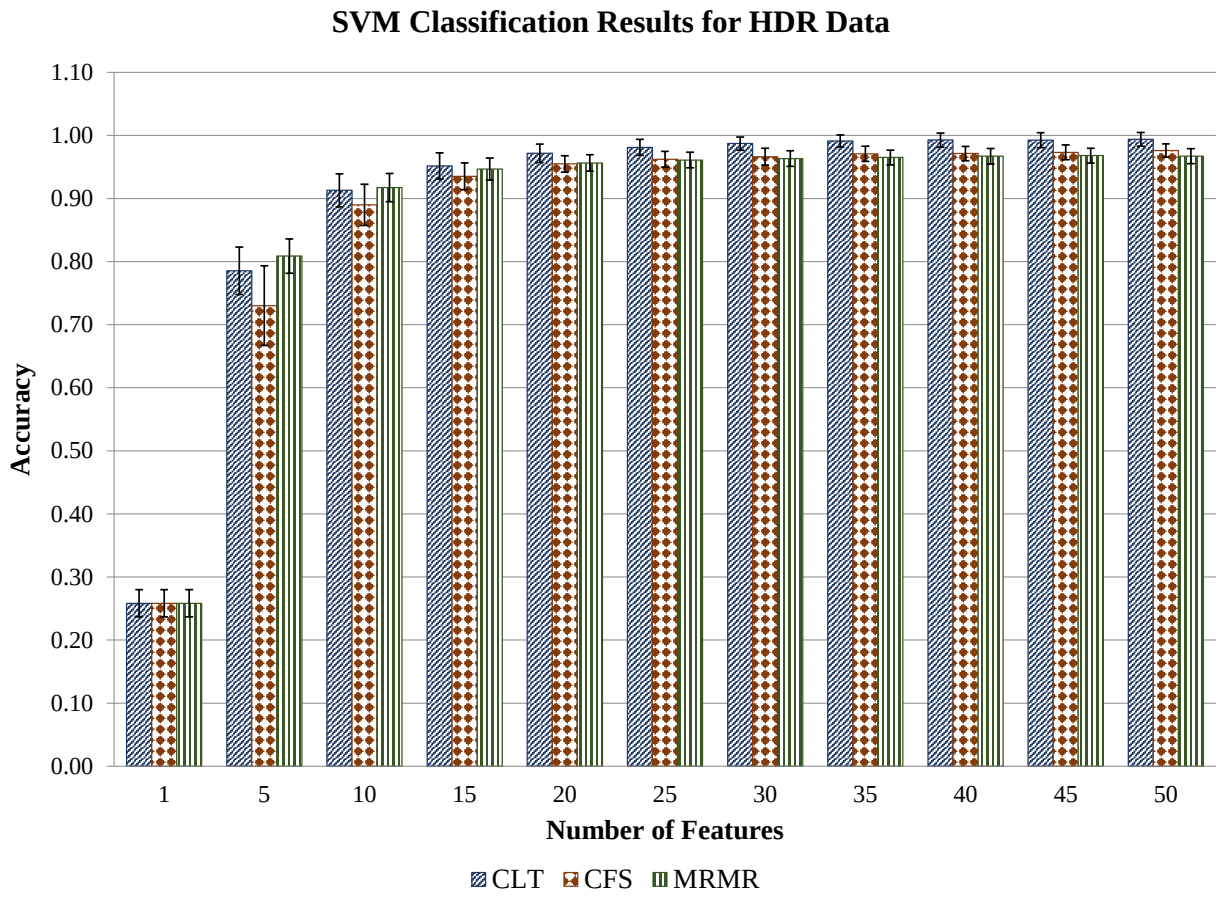


Figure 6.5: Classification accuracy result of HDR data using SVM classification.

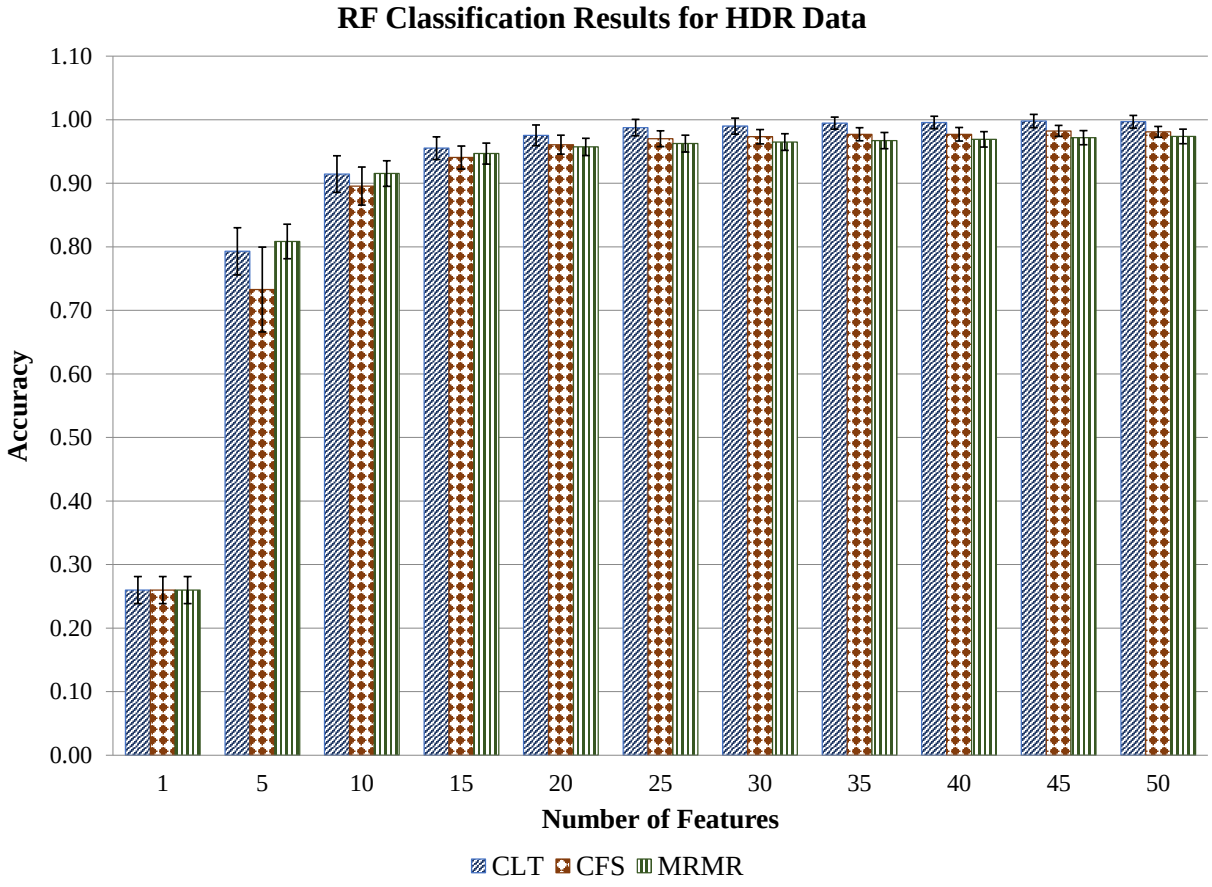


Figure 6.6: Classification accuracy result of HDR data using RF classification.

Figure 6.7: The 20 objects of the COIL-20 data set.

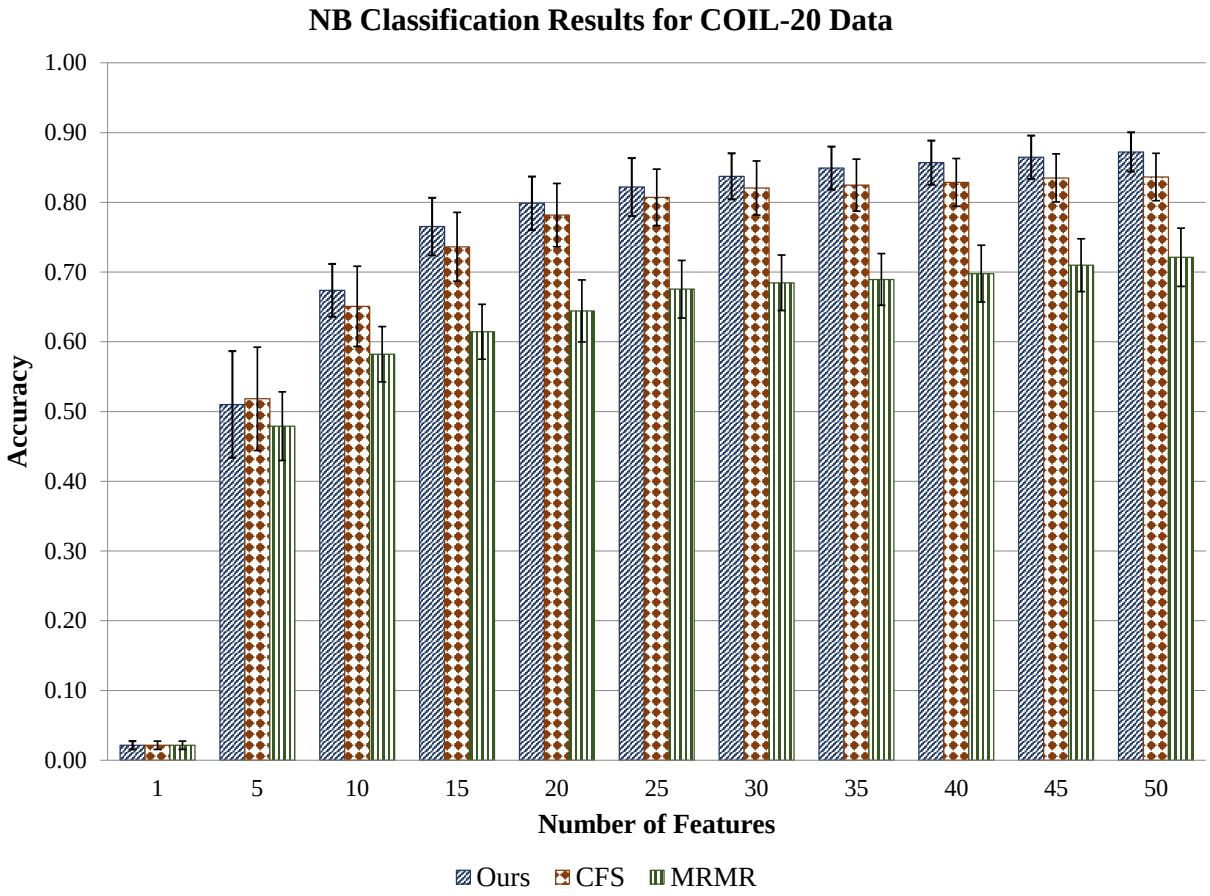


Figure 6.8: Classification accuracy result of COIL-20 data using NB classification.

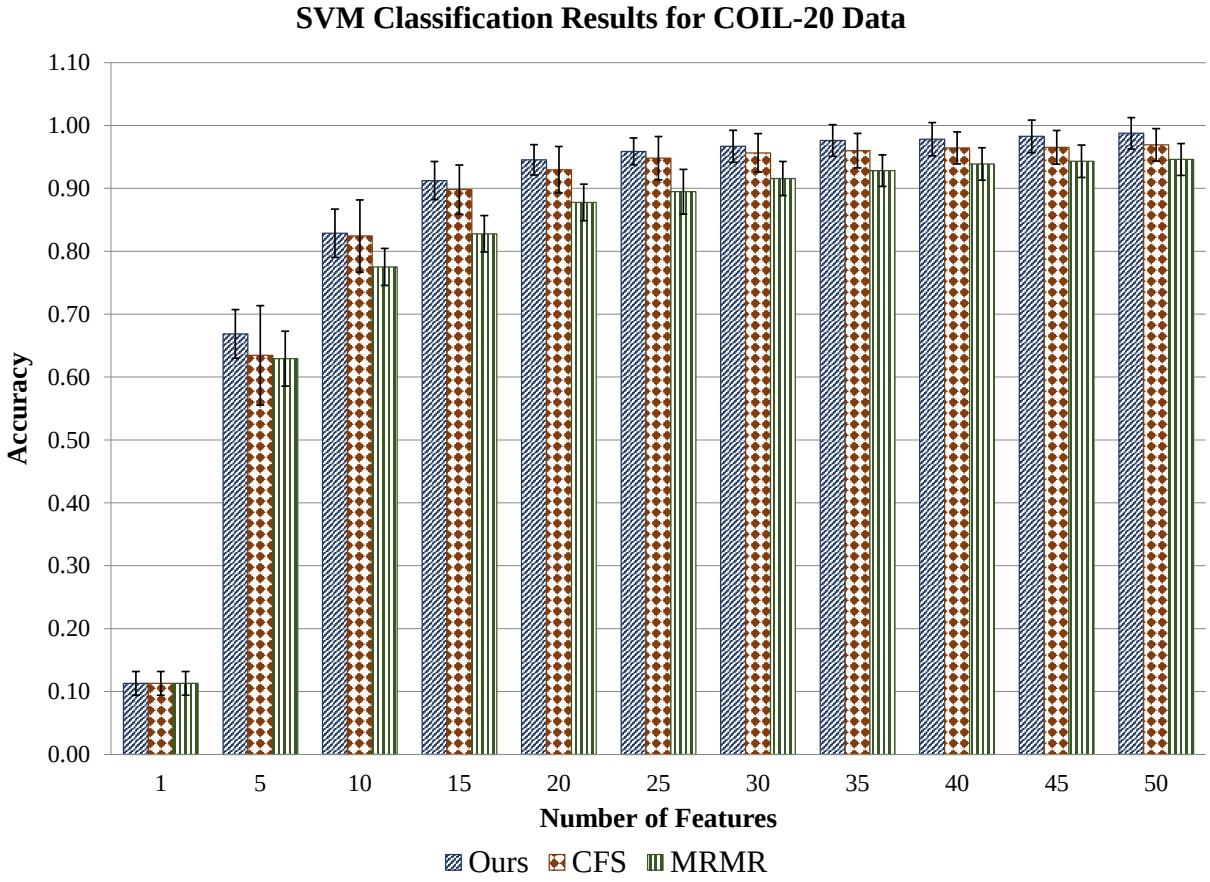


Figure 6.9: Classification accuracy result of COIL-20 data using SVM classification.

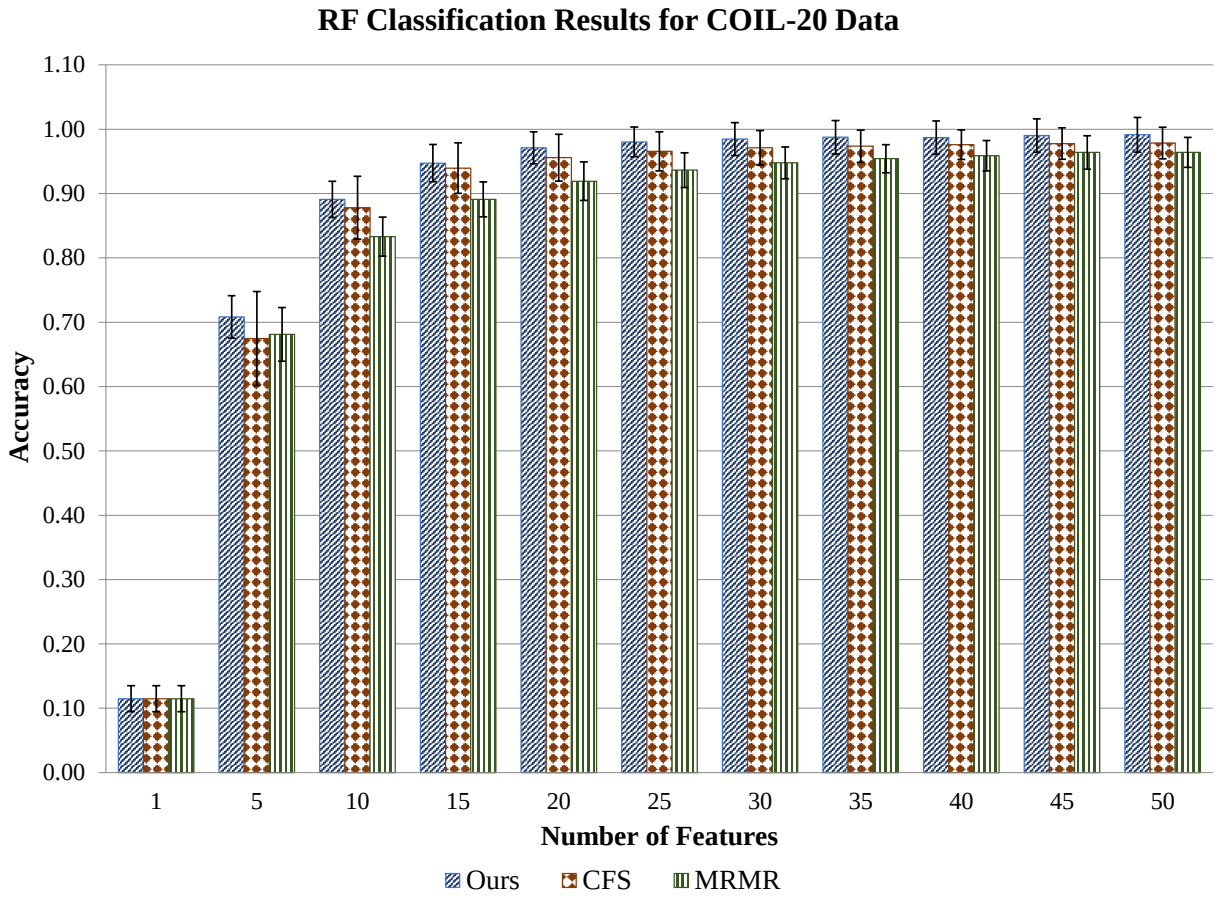


Figure 6.10: Classification accuracy result of COIL-20 data using RF classification.

7. CONCLUSION

This dissertation can be divided into two main topics, clustering with metric learning and feature selection. The flow of this dissertation is to first discuss works in the unsupervised scheme, namely in clustering, then move to a semi-supervised scheme when metric learning is added to clustering, lastly supervised learning is discussed in feature selection methods.

The previous work of Chi et al. [15] in convex clustering is first introduced. The convexity of the clustering scheme and its ability to produce multiple clustering solutions with varying size proved this new clustering optimization formulation desirable in comparison to older clustering algorithms. The convex clustering formulation aims to assign every data point to a cluster center vector, the data point with the same cluster center is considered to be in the same cluster. The optimization minimizes the aggregated distance between each point and its corresponding cluster along with a regularization parameter that controls the number of clusters. The convex clustering algorithm can be solved using ADMM. It is clear that in this cluster scheme, there is a need for an appropriate distance measure to measure the distance between two data points with d features. This dissertation proposed two metric learning algorithms that learns a Mahalanobis metric that beat the results of convex clustering using Euclidean distance in the original formulation. The first metric proposes a positive definite full-rank matrix to characterize the the metric. A principle component analysis of the full-rank matrix can reveal which features is more important in the clustering scheme. The second formulation of the Mahalanobis matrix is to characterize the matrix as a weighted sum of s rank-1 positive semidefinite matrices, which are orthonormal basis. The number of basis s can be significantly smaller than the original dimension of that data d. This produces a dimension reduction effect that projects the data into a smaller and more meaningful space and also introduces sparsity into the metric learning scheme. Experiments were conducted using both simulated data and real life data to evaluate the performance of convex clustering with both metric learning schemes and showed that metric learning greatly improve the performance of convex clustering.

With the introduction of sparsity into metric learning scheme, we moved from a unsupervised learning problem to a supervised learning problem. The sparsity of metric learning can also be seen as a method of feature selection which is the second topic of this dissertation. We proposed a new method of feature selection that uses Chow-Liu tree approximations to estimate the mutual information between a subset of features to be selected and the outcome variable. This algorithm is a incremental search greedy algorithm much like the previously popular mRMR and CFS algorithms. However, it does not rely on heuristics and thus produce much desirable effects when the selected features are used in classification algorithms. The feature selection methods were evaluated over the classification accuracy in three different classification methods. Three large real life data were used in the experiments and they showed that the new Chow-Liu tree method outperforms previous state-of-the-art methods.

REFERENCES

[1] M. Dettling and P. Bühlmann, "Supervised clustering of genes," *Genome Biology*, vol. 3, no. 12, pp. research0069.1–0069.15, 2002.

[2] X. Shen, F. Tokoglu, X. Papademetris, and R. Constable, "Groupwise whole-brain parcellation from resting-state fMRI data for network node identification," *NeuroImage*, vol. 82, pp. 403 – 415, 2013.

[3] F. Yin and C.-L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning," *Pattern Recognition*, vol. 42, no. 12, pp. 3146 – 3157, 2009.

[4] H. Liu, M. Shao, S. Li, and Y. Fu, "Infinite ensemble for image clustering," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1745–1754, ACM, 2016.

[5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[6] X. Jin and J. Han, *K-Means Clustering*, pp. 563–564. Boston, MA: Springer US, 2010.

[7] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert, "Clusterpath an algorithm for clustering using convex fusion penalties," in *28th International Conference on Machine Learning*, (United States), p. 1, June 2011.

[8] R. Lajugie, F. Bach, and S. Arlot, "Large margin metric learning for constrained partitioning problems," in *Proc. International Conference on Machine Learning*, 2014.

[9] F. R. Bach and M. I. Jordan, "Learning spectral clustering," in *Advances in neural information processing systems*, pp. 305–312, 2004.

[10] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Subspace clustering guided unsupervised feature selection," *Pattern Recognition*, vol. 66, no. Supplement C, pp. 364 – 374, 2017.

[11] I. Majerova and J. Nevima, "The measurement of human development using the ward method of cluster analysis," *Measurement*, vol. 239, p. 257, 2017.

[12] R. D. Nowak, "Distributed em algorithms for density estimation and clustering in sensor networks," *IEEE transactions on signal processing*, vol. 51, no. 8, pp. 2245–2253, 2003.

[13] C. Fraley, "Algorithms for model-based gaussian hierarchical clustering," *SIAM J. Sci. Comput.*, vol. 20, pp. 270–281, Dec. 1998.

[14] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, pp. 803–821, 1993.

[15] E. C. Chi and K. Lange, "Splitting methods for convex clustering," *Journal of Computational and Graphical Statistics*, vol. 24, no. 4, pp. 994–1013, 2015.

[16] X. Sui, X. Li, X. Qian, and T. Liu, "Convex clustering with metric learning," *Pattern Recognition*, vol. 81, pp. 575–584, September 2018.

[17] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *AAAI*, vol. 2, pp. 129–134, 1992.

[18] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper–filter feature selection algorithm using a memetic framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 70–76, 2007.

[19] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," *Expert Systems with Applications*, vol. 33, no. 1, pp. 1–5, 2007.

[20] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[21] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 856–863, 2003.

[22] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[23] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 1, pp. 1–14, 2013.

[24] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[25] M. A. Hall, "Correlation-based feature selection of discrete and numeric class machine learning," 2000.

[26] Q. Wang, P. Gong, S. Chang, T. Huang, and J. Zhou, "Robust convex clustering analysis," in *Proceedings - 16th IEEE International Conference on Data Mining, ICDM 2016*, pp. 1263–1268, January 2017.

[27] B. Wang, Y. Zhang, W. W. Sun, and Y. Fang, "Sparse convex clustering," *Journal of Computational and Graphical Statistics*, vol. 27, no. 2, pp. 393–403, 2018.

[28] W. Commons, "Iris flowers clustering kmeans," 2010.

[29] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017.

[30] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 524–531, IEEE, 2005.

[31] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *Computer Vision–ECCV 2012*, pp. 488–501, Springer, 2012.

[32] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," 2007.

[33] X. Li, C. Shen, Q. Shi, A. Dick, and A. van den Hengel, "Non-sparse linear representations for visual tracking with online reservoir metric learning," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1760–1767, IEEE, 2012.

[34] P. C. Mahalanobis, "On the generalized distance in statistics," National Institute of Science of India, 1936.

[35] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in neural information processing systems*, pp. 521–528, 2003.

[36] S. C. Hoi, W. Liu, and S.-F. Chang, "Semi-supervised distance metric learning for collaborative image retrieval and clustering," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 6, no. 3, p. 18, 2010.

[37] Y. Shi, A. Bellet, and F. Sha, "Sparse compositional metric learning.," in *AAAI*, pp. 2078–2084, 2014.

[38] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[39] F. Lindsten, H. Ohlsson, and L. Ljung, "Just relax and come clustering!: A convexification of k-means clustering," Tech. Rep. 2992, LinkÃűping University, The Institute of Technology, 2011.

[40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.

[41] R. Glowinski and A. Marroco, "Sur l'approximation, par Ãl'lÃl'ments finis d'ordre un, et la rÃl'solution, par pÃl'nalisation-dualitÃl' d'une classe de problÃĺmes de dirichlet non linÃl'aires," *ESAIM: Mathematical Modelling and Numerical Analysis - ModÃl'lisation MathÃl'matique et Analyse NumÃl'rique*, vol. 9, no. R2, pp. 41–76, 1975.

[42] D. Gabay, "Chapter ix: applications of the method of multipliers to variational inequalities," *Studies in mathematics and its applications*, vol. 15, pp. 299–331, 1983.

[43] A. Jameson, "Solution of the equation $AX + XB = C$ by inversion of an $M * M$ or $N * N$ matrix," *SIAM Journal on Applied Mathematics*, vol. 16, no. 5, pp. 1020–1023, 1968.

[44] Å. Björck, *Numerical methods in matrix computations*. Springer, 2015.

[45] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation $AX + XB = C$," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[46] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[47] A. E. Bryson, *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[48] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *arXiv preprint arXiv:1001.0736*, 2010.

[49] K. Chong, "The arithmetic geometric mean inequality: a short proof," *Internation575al Journal of Mathematical Education in Science and Technology*, vol. 12, no. 6, pp. 653–654, 1981.

[50] W. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[51] Y. Hoshida, J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Subclass mapping: identifying common subtypes in independent disease data sets," *PloS one*, vol. 2, no. 11, p. e1195, 2007.

[52] X. He, T. Gumbsch, D. Roqueiro, and K. Borgwardt, "Kernel conditional clustering," in *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 157–166, Nov 2017.

[53] N. Sánchez-Maroño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection - a comparative study," in *Intelligent Data Engineering and Automated Learning - IDEAL 2007* (H. Yin, P. Tino, E. Corchado, W. Byrne, and X. Yao, eds.), (Berlin, Heidelberg), pp. 178–187, Springer Berlin Heidelberg, 2007.

[54] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[55] S. A. Nene, S. K. Nayar, H. Murase, *et al.*, "Columbia object image library (coil-20),"