June 2020

On Sequence Clustering and Supervised Dimensionality Reduction

Tiexing Wang
Syracuse University

ABSTRACT

This dissertation studies two machine learning problems: 1) clustering of independent and identically generated random sequences, and 2) dimensionality reduction for classification problems.

For sequence clustering, the focus is on large sample performance of classical clustering algorithms, including the k-medoids algorithm and hierarchical agglomerative clustering (HAC) algorithms. Data sequences are generated from unknown continuous distributions that are assumed to form clusters according to some well-defined distance metrics. The goal is to group data sequences according to their underlying distributions with little or no prior knowledge of both the underlying distributions as well as the number of clusters. Upper bounds on the clustering error probability are derived for the k-medoids algorithm and a class of HAC algorithms under mild assumptions on the distribution clusters and distance metrics. For both cases, the error probabilities are shown to decay exponentially fast as the number of samples in each data sequence goes to infinity. The obtained error exponent bound has a simple form when either the Kolmogrov-Smirnov distance or the maximum mean discrepancy is used as the distance metric. Tighter upper bound on the error probability of the single-linkage HAC algorithm is derived by taking advantage of the simplified metric updating scheme. Numerical results are provided to validate the analysis.

For dimensionality reduction, the focus is on classification problem where label information in the training data can be leveraged for improved learning performance. A supervised dimensionality reduction method maximizing the difference of average projection energy of samples with different labels is proposed. Both synthetic data and WiFi sensing data are used to validate the effectiveness of the proposed method. The numerical results show that the proposed method outperforms existing supervised dimensionality reduction approaches based on Fisher discriminant analysis (FDA) and Hilbert-Schmidt independent

criterion (HSIC). When kernel trick is applied to all three approaches, the performance of the proposed dimensionality reduction method is comparable to FDA and HSIC and is superior over unsupervised principal component analysis.

ON SEQUENCE CLUSTERING AND SUPERVISED DIMENSIONALITY REDUCTION

By

Tiexing Wang
B.E., Beijing Institute of Technology, 2010

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical & Computer Engineering

Syracuse University
December 2020

ACKNOWLEDGMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance.

I would like to express the deepest appreciation to my advisor, Dr. Biao Chen, whose expertise was invaluable in formulating the research questions and methodology. I am also deeply indebted to him for the support during the COVID-19 pandemic.

I would also like to extend my deepest gratitude to the rest of my committee - Dr. Makan Fardad, Dr. Asif Salekin, Dr. Lixin Shen, Dr. Pramod K. Varshney and Dr. Reza Zafarani. Thank you for taking the time to review my work and offering insightful suggestions. I cannot leave Syracuse University without mentioning Dr. Yingbin Liang who provided valuable advice on my research. I am also grateful to Dr. Shuai Wang for bringing me into academic research.

I would like to extend my sincere thanks to my labmates and colleagues - Fangfang, Kapil, Pengfei, Fangrong, Yu, Shengyu, Yang and Qunwei. It was a great pleasure to work with you. I would also like to acknowledge the help from Yuexin who set up the experiment environment for my dissertation.

In addition, I am forever indebted to my parents for encouraging me to explore new directions in life. I am also extremely grateful to my girlfriend Yang Liu. Without her support, it would have been impossible to overcome difficulties during the pandemic.

Finally, I would like to acknowledge the generous support of Air Force Office of Scientific Research under Award FA9550-16-1-0077 and National Science Foun-

dation under Award CNS-1731237.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

This chapter introduces two machine learning problems that have broad applications in various fields: sequence clustering and supervised dimensionality reduction. For the former, sequences are assumed to be generated from unknown continuous distributions and the goal is to group sequences according to some well-defined distribution metrics. For the latter, dimensionality reduction is achieved by taking into account the label information to preserve maximum discriminating information for classification problems.

1.1 Sequence clustering

Sequence clustering is of interest to a broad range of applications. Examples include market segmentation [1], image clustering [2, 3], and meteorological parameters characterization [4–6]. This dissertation considers clustering of sequences generated by unknown continuous distributions. Each sequence consists of independent and identically distributed (i.i.d.) samples. The underlying distributions for the sequences are assumed to form clusters with well-defined distance metrics for distributions. Distributions belonging to the same cluster are close to each other whereas distributions belonging to different clusters are assumed to be well separated from each other. For sequence clustering, while Euclidean

distance and other vector norms have often been used [7,8], metrics that characterize distribution distances are more relevant for the intended clustering problem when the underlying generative distributions are of concern.

The above clustering problem belongs to the general problem of unsupervised learning [9, 10]. There are generally two classes of approaches: partitional and hierarchical. Partitional clustering algorithms include k-means [11–13] and k-medoids [14–16] clustering. They usually start with some initial cluster centers, often randomly chosen, assign data sequences to cluster centers, update cluster centers, and repeat the process until convergence occurs.

Hierarchical clustering algorithms include both hierarchical agglomerative clustering (HAC) algorithms and hierarchical divisive clustering (HDC) algorithms. HAC algorithms start with singletons and proceed to merge clusters having the smallest pairwise distance [17]. HDC algorithms, on the other hand, start with one cluster consisting of all data sequences and proceed to split sequences into clusters [18, 19].

While the knowledge of the number of clusters is usually required for partitional clustering algorithms, this is not necessary for hierarchical clustering algorithms. However, the threshold for merging or splitting is required for hierarchical clustering algorithms. In the following, we review existing results in both partitional and hierarchical clustering algorithms.

1.1.1 Partitional-based clustering algorithms

The partitional-based clustering algorithms usually require the knowledge of the number of clusters and they differ in how the initial centers are determined. One reasonable way is to choose a data sequence as a center if it has the largest minimum distances to all the existing centers [20–22]. Alternatively, all the initial centers can be randomly chosen [6]. With the number of clusters unknown, there are typically two alternative approaches for clustering. One starts with a small number of clusters, e.g., 1, which is an underestimate of

the true number, and proceed to split the existing clusters until convergence [22, 23]. The authors in [23] assumed a maximum number of clusters and the threshold for clustering depended on a pre-determined significance level of the two sample kolmogorov-smirnov (KS) test whereas the algorithm proposed in [22] did not assume a maximum number of clusters and the threshold for clustering was a function of the intra-cluster and inter-cluster distances. Alternatively, one may start with an overestimated the number of clusters, e.g., every sequence is treated as a cluster, and proceed to merge clusters that are deemed close to each other [22]. The algorithms in [6, 20, 23] were all validated by simulation results without carrying out an analysis of the error probability.

There are some key differences between the k-means algorithm and the k-medoids algorithm. The k-means algorithm minimizes a sum of squared Euclidean distances. Meanwhile, the k-medoids algorithm assigns data sequences as centers and minimizes a sum of arbitrary distances, which makes it more robust to outliers and noise [24, 25]. Moreover, the k-means algorithm requires updating the distances between data sequences and the corresponding centroids in every iteration whereas the k-medoids algorithm only requires the pairwise distances of the data sequences, which can be computed before hand. Thus, the k-medoids algorithm outperforms the k-means algorithm in terms of computational complexity as the number of sequences increases [26].

Most prior research focused on computational complexity analysis, whereas the error probability and the performance comparison of different clustering algorithms were typically studied through numerical experiments [15, 16, 26, 27]. This dissertation attempts to theoretically analyze the error probability for the k-medoids algorithm especially in the asymptotic region. Furthermore, in contrast to previous studies, which frequently used vector norms as the distance metric (e.g., Euclidean distance), our study adopts the distance metrics between distributions for clustering in order to capture the statistical models of data sequences considered in this dissertation. This formulation based on a distributional distance metric is uniquely suited to the proposed clustering problem, where each data point,

i.e., each sequence, represents an empirical probability distribution and each cluster is a collection of distributions that are close to each other with respect to a suitably selected distribution metric.

1.1.2 Hierarchical clustering algorithms

Hierarchical clustering algorithms include both hierarchical agglomerative clustering (HAC) algorithms and hierarchical divisive clustering (HDC) algorithms. HAC algorithms start with singletons and proceed to merge clusters having the smallest pairwise distance [17]. HDC algorithms, on the other hand, start with one cluster consisting of all data sequences and proceed to split sequences into clusters [18, 19]. The knowledge of the number of clusters is not necessary for hierarchical clustering algorithms. However, the threshold for merging or splitting is required for hierarchical clustering algorithms.

HAC algorithms can be further divided into two groups - linkage-based algorithms and centroid-based algorithms. Linkage-based algorithms determine clustering using pairwise distances between sequences; centroid-based algorithms on the other hand rely on distances between cluster centroids. Examples for linkage-based algorithms include single-linkage (SLINK) [28], complete-linkage (CLINK) [29], weighted pair group method with arithmetic mean (WPGMA) and unweighted pair group method with arithmetic mean (UPGMA) [30]. Centroid-based clustering algorithms include unweighted pair-group method centroid (UPGMC) and weighted pair-group method centroid (WPGMC) [31]. For both linkage-based and centroid-based HAC algorithms distances between clusters are updated in a recursive manner [32] called Lance-Williams Dissimilarity (LWD) update formula and the difference between these two classes are reflected by different weights in the LWD update.

There has been prior work on the consistency for sequence clustering using HAC algorithms. For example, in [33], the performance of HAC algorithms in the asymptotic regime given Gaussian mixture model is analyzed. The information-theoretic threshold for

clustering data sequences generated from Gaussian distributions with different means and identical variance is investigated in [34], where the difference between means shrinks as sample size increases. In [35], clustering time series generated from stationary ergodic distributions is considered where the sequence does not need to be independent and identically distributed (i.i.d.). The proposed clustering algorithms therein are shown to be consistent. The trade-off is that a single distribution is assumed for each cluster; this is different from the current work where each cluster consists of multiple distributions. We note that a popular approach to analyze HAC algorithms is to define and subsequently minimize a cost function [36, 37]. With sequence clustering, error probability appears to be a natural choice instead of any specialized cost functions.

1.2 Supervised Dimensionality Reduction

Principle component analysis (PCA) is a classical method for *unsupervised* data dimensionality reduction approach [38]. PCA searches a low-dimension linear subspace approximation of original data that preserves the maximum variation. However, classical PCA is inherently unsupervised; finding the best linear approximation by PCA does not take into account the label information associated with data when applied to supervised learning such as classification problems [39].

Supervised dimensionality reduction (SDR) for classification has attracted a lot of research interest in recent years [40–54]. For example, if some components of the original samples are highly correlated with labels, then a reasonable way for dimensionality reduction is to compute the correlation between every component and the labels and compare it with a pre-determined threshold [40]. Only components corresponding to the correlation exceeding the threshold are kept. PCA can be then applied to the selected components for further dimensionality reduction. The method proposed in [40] has some drawbacks. First, it does not work when the number of classes exceeds 2. Second, the components excluded

by the threshold may still contain useful information for classification. A iterative version of the method is then proposed in [41], which choose one component in each iteration. The influence of the newly selected component is then subtracted from the original samples. The next component is then selected in the same manner.

More sophisticated approaches are also proposed for SDR. Some existing works are shown to be equivalent to (generalized) eigenvalue problems. For instance, Fisher discriminant analysis (FDA) finds the subspace that preserves the maximum difference of projected empirical means with different labels normalized by the sample variance [42]. The subspace obtained by FDA is always $(L-1)$-dimensional, where L is the number of classes. The performance of FDA suffers when 1) the empirical means of different classes are close to each other or 2) some class consists disjoint clusters, i.e., data become multi-modal [43]. Local FDA is then proposed for the multi-modal case [44] which preserves the structure of local data. Another drawback of FDA is that potential information loss may occur given large sample size and small L. Alternatively, some works focus on maximizing the dependency between projected samples and labels. The author in [45] proposed an SDR approach which maximizes Hilbert-Schmidt independence criterion (HSIC) between samples and labels. The subspace obtained by HSIC-based SDR is at most L-dimensional, which implies potential information loss given large sample size and small L as well. This problem can be alleviated by modifying the kernel matrix of labels [46, 47]. However, increasing the rank of the kernel matrix of labels may reduce the dependency between samples and labels. The HSIC-based SDR is suitable for cases where the sample dimension is much larger than the sample size, e.g., biomedical image processing [48]. One advantage shared by SDR approaches equivalent to (generalized) eigenvalue problems is that the subspaces with different dimensions are obtained from the same unitary matrix obtained by eigenvalue decomposition. This enables adaptive selection of the number of features without the need to recompute the eigen-decomposition.

There are other SDR approaches that can not be formulated as eigen decomposition

problems [49–54]. They usually do not have a closed-form solution and requires repeating the SDR procedure if the number of selected features changes. In [49], the author proposed an approach which jointly considers SDR and classification for a pre-determined subspace dimension. The projection matrix is obtained through jointly minimizing the approximation error and a loss function related to classification error. In [50], the distance correlation is used to characterize the dependency between the samples and the labels. The objective function, depending on the pairwise distance of samples and labels, is also equivalent to an eigenvalue problem. However, the constraint depends on the pairwise distance matrix of the samples. In [51], a probability-based SDR approach is proposed based on the assumption that data samples follow one of the common distributions such as Gaussian, heavy-tail or linear. The cost function depends on the joint probability distributions in the projection and response spaces. The projection matrix is solved by optimization methods. In [52], a modified supervised distance preserving projection (SDPP) initially introduced in [53] is proposed which incorporates the total variance of the projection and preserves the global structure simultaneously. One can also apply neural network for SDR. A centroid-encoder which is a generalized auto-encoder is proposed in [54].

This dissertation will focus on SDR approaches that can be transformed into a (generalized) eigenvalue problem.

1.3 Scope of Dissertation and Summary of Contributions

The scope of the dissertation and its contributions are summarized in this section.

1.3.1 Scope of Dissertation

For sequence clustering, we focus on large sample performance and establish exponential consistency of a number of classical clustering algorithms including the k-medoids algo-

rithm and HAC algorithms. Data sequences are generated from unknown continuous distributions that are assumed to form clusters according to some well-defined distance metrics. The goal is to group data sequences according to their underlying distributions with little or no prior knowledge of both the underlying distributions as well as the number of clusters. Upper bounds on the clustering error probability are derived for the k-medoids algorithm and a class of HAC algorithms under mild assumptions on the distribution clusters and distance metrics. For both cases, the error probabilities are shown to decay exponentially fast as the number of samples in each data sequence goes to infinity. The obtained error exponent bound has a simple form when either the KS distance or the maximum mean discrepancy (MMD) is used as the distance metric. Tighter upper bound on the error probability of SLINK algorithm is derived by taking advantage of the simplified metric updating scheme. Numerical results are provided to validate the analysis.

For supervised dimensionality reduction, we attempt to address deficiency of several existing approaches. Specifically, a supervised dimensionality reduction method maximizing the difference of average projection energy of samples with different labels is proposed. Both synthetic data and WiFi sensing data are used to validate the effectiveness of the proposed method. The numerical results show that the proposed method outperforms existing supervised dimensionality reduction approaches based on FDA and HSIC as well as PCA. When kernel trick is applied to all these approaches, the performance of the proposed dimensionality reduction method is comparable to FDA and HSIC and is superior over unsupervised principal component analysis.

1.3.2 Summary of contributions

The contribution of this dissertation is summarized as follows.

- For data sequences generated from distributions satisfying some simple assumptions, the k-medoids algorithm is shown to be exponentially consistent. That is, the error probability of clustering algorithms decays exponentially fast as the sample size in-

creases. The exponential consistency is established with both known and unknown number of clusters.

- For both linkage-based and centroid-based HAC algorithms, exponential consistency is established when the number of clusters is unknown. While the results for these two HAC algorithms differ, the analysis is unified as both these algorithms conform to the Lance-Williams dissimilarity (LWD) update.

- A new supervised dimensionality reduction method is proposed. The new approach maximizes the difference of the average energy in the subspace between data with different labels. The proposed method is shown to outperform existing dimensionality reduction approaches on both synthetic data and WiFi sensing data [55].

- A kernelized version of the proposed SDR method is also developed. Numerical comparison of the kernelized versions demonstrate that the proposed method achieves similar performance to kernelized FDA and HSIC while significantly outperform kernelized PCA.

CHAPTER 2
SEQUENCE CLUSTERING BY K-MEDOIDS ALGORITHM

This chapter focuses on asymptotic performance study of sequence clustering using the k-medoids algorithm. Two commonly used distribution metrics, the KS distance and MMD, are introduced, along with some relevant properities. The upper bound on the error probability of k-medoids algorithm with a known number of clusters is derived, followed by parallel results of the clustering algorithms with an unknown number of clusters. The derived upper bounds are shown to decay exponentially as the same size increases, establishing the exponential consistency of the k-medoids algorithm. The simulation results of k-medoids algorithm under the KS distance and MMD are provided in Section 2.4.

2.1 System Model and Preliminaries

2.1.1 Clustering Problem

Suppose there are K distribution clusters denoted by $\mathcal{P}_k$ for $k = 1, \ldots, K$, where K is fixed but unknown. Define the intra-cluster distance of $\mathcal{P}_k$ and the inter-cluster distance

between $\mathcal{P}_k$ and $\mathcal{P}_{k'}$ for $k \neq k'$ to be

$$\begin{aligned} d(\mathcal{P}_k) &= \sup_{p_i, p_{i'} \in \mathcal{P}_k} d(p_i, p_{i'}), \\ d(\mathcal{P}_k, \mathcal{P}_{k'}) &= \inf_{p_i \in \mathcal{P}_k, p_{i'} \in \mathcal{P}_{k'}} d(p_i, p_{i'}), \end{aligned} \tag{2.1}$$

where $d(\cdot, \cdot)$ is a suitably defined distribution metric. Thus $d(\mathcal{P}_k)$ and $d(\mathcal{P}_k, \mathcal{P}_{k'})$ are respectively the diameter of $\mathcal{P}_k$ and the distance between $\mathcal{P}_k$ and $\mathcal{P}_{k'}$. Define further

$$\begin{aligned} d_L &= \max_{k=1,\ldots,K} d(\mathcal{P}_k), \\ d_H &= \min_{k \neq k'} d(\mathcal{P}_k, \mathcal{P}_{k'}), \\ \Sigma &= d_H + d_L, \\ \Delta &= d_H - d_L. \end{aligned} \tag{2.2}$$

Furthermore, when specific distance metric is used, subscript reflecting the distance metric will be added, e.g., for the KS distance, d_{ks}, $d_{L,ks}$, $d_{H,ks}$, Σ_{ks} and Δ_{ks} represent the corresponding quantities defined in (2.1) and (2.2).

Suppose M_k data sequences are generated from distributions in $\mathcal{P}_k$, hence a total of $M \coloneqq \sum_{k=1}^{K} M_k$ sequences are to be clustered. Each sequence $\mathbf{x}_{k,j_k} = [\mathbf{x}_{k,j_k}[1], \ldots, \mathbf{x}_{k,j_k}[n]]$ consists of n i.i.d. samples generated from $p_{k,j_k} \in \mathcal{P}_k$ for $k = 1, \ldots, K$ and $j_k \in \{1, \ldots, M_k\}$. Note that p_{k,j_k}'s are not necessarily distinct for the same k, i.e., $\mathbf{x}_{k,j_k}$'s can be generated from the same distribution from cluster k. Additionally, all sequences are assumed to have equal length; our analysis can be easily extended to the case with different sequence lengths by replacing n with the minimum sequence length.

We make the following assumptions on distribution clusters and on the distance metrics used in clustering.

Assumption 1. *The d_L and d_H defined in* (2.2) *satisfies*

$$d_L < d_H. \tag{2.3}$$

Therefore, inter-cluster distances are greater than intra-cluster distances, ensuring the clustering problem to be well defined.

Assumption 2. *For any distribution clusters $\{\mathcal{P}_1, \ldots, \mathcal{P}_K\}$, any length-$n$ sequences $\mathbf{x}_{k,j_k} \sim p_{k,j_k}$, $\mathbf{x}_{k,j'_k} \sim p_{k,j'_k}$ and $\mathbf{x}_{k',j_{k'}} \sim p_{k',j_{k'}}$, where $k \neq k'$, and sufficiently large n, the following inequalities hold for any d_{th} with $d_L < d_{th} < d_H$:*

$$P\left(d\left(\mathbf{x}_{k,j_k}, \mathbf{x}_{k',j_{k'}}\right) \leq d_{th}\right) \leq a_1 e^{-b_1 n}, \tag{2.4a}$$

$$P\left(d\left(\mathbf{x}_{k,j_k}, \mathbf{x}_{k,j'_k}\right) > d_{th}\right) \leq a_2 e^{-b_2 n}, \tag{2.4b}$$

$$P\Big(d(\mathbf{x}_{k,j_k}, \mathbf{x}_{k,j'_k}) \geq d(\mathbf{x}_{k,j_k}, \mathbf{x}_{k',j_{k'}})\Big) \leq a_3 e^{-b_3 n}, \tag{2.4c}$$

where a_i's are some constants independent of distributions, b_i's (> 0) is a function of d_{th} and n is the sample size. □

Assumption 2 relates to the concentration properties of the distance metric $d(\cdot, \cdot)$ and is completely independent of the distribution clusters. Eq. (2.4a) states that the probability that the distance between two sequences generated from distributions belonging to different clusters is smaller than d_H decays exponentially fast. Eq. (2.4b) states that the probability that the distance between two sequences generated from distributions belonging to the same cluster is greater than d_L decays exponentially fast. Eq. (2.4c) states that the probability that a sequence is closer to a sequence from a different cluster than to a sequence of the same cluster decays exponentially fast.

A clustering error occurs if 1) any sequences generated from different distribution clusters are assigned to the same cluster, or 2) sequences generated by the same distribution cluster are assigned to more than one cluster. A clustering algorithm is said to be *consistent*

if for any $0 \leq d_L < d_H$,

$$\lim_{n\to\infty} P_e = 0,$$

where P_e is the probability of clustering error and n is the sequence length. The algorithm is said to be *exponentially consistent* if for any $0 \leq d_L < d_H$,

$$B = \lim_{n\to\infty} -\frac{1}{n} \log P_e > 0.$$

For the case where a clustering algorithm is exponentially consistent, we are also interested in characterizing the (bound for) error exponent B.

2.1.2 Preliminaries of KS distance

Denote by F_p the cumulative distribution function (c.d.f.) of distribution p. The KS distance between distributions p and q is defined as

$$d_{KS}(p, q) = \sup_{a\in\mathbb{R}} |F_p(a) - F_q(a)|. \tag{2.5}$$

Let $\mathbf{x}$ be an i.i.d. sequence generated by the distribution p. The empirical c.d.f. induced by $\mathbf{x}$ is given by

$$F_{\mathbf{x}}(a) = \frac{1}{n} \sum_{i=1}^{n} 1_{[-\infty,a]}(\mathbf{x}[i]),$$

where $1_{[-\infty,x]}(\cdot)$ is the indicator function. The empirical KS distance between two sequences $\mathbf{x}$ and $\mathbf{y}$ is the KS distance between the corresponding empirical c.d.f., and denoted by $d_{KS}(\mathbf{x}, \mathbf{y})$ for notational convenience.

2.1.3 Preliminaries of MMD

Let $\mathcal{P}$ be a set of distributions, and $\mathcal{H}$ the reproducing kernel Hilbert space (RKHS) associated with a positive definite kernel $g(\cdot, \cdot)$ [56]. Define a mapping from $\mathcal{P}$ to $\mathcal{H}$ such that

each distribution $p \in \mathcal{P}$ is mapped into an element in $\mathcal{H}$ as follows

$$\mu_p(\cdot) = \mathbb{E}_p[g(\cdot, x)] = \int g(\cdot, x)\, dp(x),$$

where $\mu_p(\cdot)$ is the *mean embedding* of the distribution p into the Hilbert space $\mathcal{H}$. The mean embedding of distributions is guaranteed to exist for bounded kernels and satisfies the reproducing property $\mathcal{H}$, $\mathbb{E}_p[f] = \langle \mu_p, f \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$.

Additionally, with characteristic kernels such as Gaussian and Laplace, mean embedding is injective [57–60]. Many machine learning problems involving unknown distributions can thus be solved by mean embedding of probability distributions without actually estimating the distributions [61–64]. For example, distinguishing between two distributions p and q can be achieved by computing the distance between the two mean embedding functions in the RKHS

$$d_{\text{MMD}}(p, q) := \|\mu_p - \mu_q\|_{\mathcal{H}}. \tag{2.6}$$

This is precisely the definition of MMD and its most celebrated use is in the two-sample test [65] where a biased estimator of $d_{\text{MMD}}(p, q)$ based on $\mathbf{x}$ and $\mathbf{y}$ of respective sequence lengths n and m is defined to be

$$\begin{aligned} d_{\text{MMD}}(\mathbf{x}, \mathbf{y}) = \Bigg(& \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} g(\mathbf{x}[i], \mathbf{x}[j]) + \\ & \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} g(\mathbf{y}[i], \mathbf{y}[j]) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} g(\mathbf{x}[i], \mathbf{y}[j]) \Bigg)^{\frac{1}{2}}. \end{aligned} \tag{2.7}$$

Here $g(x, y)$ is the kernel function assumed to be bounded, i.e., $0 \leq g(x, y) \leq \mathbb{G} < \infty$ for all x and y. This simple two-sample test was later shown to be asymptotically optimal [66].

Finally, we remark that both the KS distance and MMD satisfy the concentration properties in Assumption 2.

Proposition 1. [67] *If Assumption 1 holds for both the KS distance and MMD, i.e.,* $d_{L,ks} <$

Algorithm 1 Initialization with known K

Input: Data sequences $\{\mathbf{y}_i\}_{i=1}^M$, number of clusters K.
Output: Partitions $\{\mathcal{C}_k\}_{k=1}^K$.
{Center initialization}
Arbitrarily choose one $\mathbf{y}_i$ as $\mathbf{c}_1$.
for $k = 2$ to K **do**
    $\mathbf{c}_k \leftarrow \arg\max_{\mathbf{y}_i} \left(\min_{l \in I_1^{k-1}} d\left(\mathbf{y}_i, \mathbf{c}_l\right) \right)$
end for
{Cluster initialization}
Set $\mathcal{C}_k \leftarrow \emptyset$ for $1 \leq k \leq K$.
for $i = 1$ to M **do**
    $\mathcal{C}_l \leftarrow \mathcal{C}_l \cup \{\mathbf{y}_i\}$, where $l = \arg\min_{l \in I_1^K} d\left(\mathbf{y}_i, \mathbf{c}_l\right)$
end for
Return $\{\mathcal{C}_k\}_{k=1}^K$

$d_{H,ks}$, $d_{L,mmd} < d_{H,mmd}$. *Then* (2.4a) - (2.4c) *hold for both the KS distance and MMD.*

2.2 Known number of clusters

In this section, we study the clustering algorithm for known K, the number of clusters. The method proposed in [20] is used for center initialization, as described in Algorithm 1. The initial K centers are chosen sequentially such that the center of the k-th cluster is the sequence that has the largest minimum distance to the previous $k-1$ centers. The clustering algorithm itself is presented in Algorithm 2. Given the centers, each sequence is assigned to the cluster for which the sequence has the minimum distance to the center. For a given cluster, a sequence is assigned as the center subsequently if the sum of its distances to all the sequences in the cluster is the smallest. The algorithm continues until the clustering result converges.

The following theorem provides the convergence guarantee for Algorithm 2 via an upper bound on the error probability.

Theorem 2.2.1. *Algorithm 2 converges after at most* $\binom{M}{K} K^{(M-K)}$ *iterations. Moreover, if the data sequences generated from distributions satisfying Assumption 1 and the distance*

Algorithm 2 Clustering with known K

Input: Data sequences $\{\mathbf{y}_i\}_{i=1}^M$, number of clusters K.
Output: Partition set $\{\mathcal{C}_k\}_{k=1}^K$.
Initialize $\{\mathcal{C}_k\}_{k=1}^K$ by Algorithm 1.
while not converge **do**
    {Center update}
    for $k = 1$ to K **do**
        $\mathbf{c}_k \leftarrow \arg\min_{\mathbf{y}_i \in \mathcal{C}_k} \sum_{\mathbf{y}_{j'} \in \mathcal{C}_k} d(\mathbf{y}_i, \mathbf{y}_{j'})$
    end for
    {Cluster update}
    for $i = 1$ to M **do**
        if $\mathbf{y}_i \in \mathcal{C}_{k'}$ and $d(\mathbf{y}_i, \mathbf{c}_k) < d(\mathbf{y}_i, \mathbf{c}_{k'})$ **then**
            $\mathcal{C}_k \leftarrow \mathcal{C}_k \cup \{\mathbf{y}_i\}$ and $\mathcal{C}_{k'} \leftarrow \mathcal{C}_{k'} \setminus \{\mathbf{y}_i\}$.
        end if
    end for
end while
Return $\{\mathcal{C}_k\}_{k=1}^K$

metric used by the algorithm satisfies Assumption 2, the error probability of Algorithm 2 after T iterations is upper bounded as follows

$$P_e \leq M^2 \left(a_1 + a_2 + (T+1)\, a_3\right) e^{-bn},$$

where a_1, a_2, a_3 and b are as defined in Assumption 2 and

$$T \leq \binom{M}{K} K^{(M-K)}.$$

Outline of the Proof. The idea of proving the upper bound on the error probability is as follows. We first prove that the error probability at the initialization step decays exponentially. Note that the event that an error occurs during the first T iterations is the union of the event that an error occurs at the t-th step and the previous $t-1$ iterations are correct for $t = 1, \ldots, T$. Thus, if we prove that the error probability at the t-th step *given* correct updates from the previous iterations decays exponentially, then so does the error probability of the algorithm by the union bound argument. See Appendix B.1 for details. □

Theorem 2.2.1 shows that for any given K and distributions satisfying Assumption 1, any distance metric satisfying Assumption 2 yields an exponentially consistent k-medoids clustering algorithm with the error exponent b.

Corollary 2.2.1.1. *Suppose the KS distance and the MMD statistic are used for Algorithms 1 and 2, then for n sufficiently large,*

$$P_e^{KS} \leq M^2 (6T + 14) \exp\left(-\frac{n\Delta_{ks}^2}{8}\right),$$
$$P_e^{MMD} \leq M^2 (4T + 8) \exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right).$$

Proof. By Propositions 1, the upper bound on the error probability of Algorithm 2 in Theorem 2.2.1 applies to the KS distance and the MMD statistic. Thus, the corollary is obtained by substituting the values specified in Lemmas A.0.3 - A.0.8 in the upper bound. □

Corollary 2.2.1.1, combined with the fact that T is finitely bounded for finite M and K, implies that Algorithm 2 is exponentially consistent under both the KS and MMD distance metrics with an error exponent no smaller than $\frac{\Delta_{ks}^2}{8}$ and $\frac{\Delta_{mmd}^2}{64\mathbb{G}}$, respectively.

2.3 Unknown number of clusters

In this section, we propose the merge- and split-based algorithms for estimating the number of clusters as well as grouping the sequences.

2.3.1 Merge Step

If a distance metric satisfies (2.4b) and two sequences generated by distributions within the same cluster are assigned as centers, then, with high probability, the distance between the two centers is small. This is the premise of the clustering algorithm based on merging centers that are close to each other.

Algorithm 3 Merge-based initialization with unknown K

Input: Data sequences $\{\mathbf{y}_i\}_{i=1}^{M}$ and threshold d_{th}.
Output: Partitions $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$.
$\{$Center initialization$\}$
Arbitrarily choose one $\mathbf{y}_i$ as $\mathbf{c}_1$ and set $\hat{K} = 1$.
while $\max_{i \in I_1^M} \left(\min_{k \in I_1^{\hat{K}}} d\left(\mathbf{y}_i, \mathbf{c}_k\right) \right) > d_{th}$ **do**
    $\mathbf{c}_{\hat{K}+1} \leftarrow \arg\max_{\mathbf{y}_i} \left(\min_{k \in I_1^{\hat{K}}} d\left(\mathbf{y}_i, \mathbf{c}_k\right) \right)$
    $\hat{K} \leftarrow \hat{K} + 1$
end while
Clustering initialization specified in Algorithm 1.
Return $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$

The proposed approach is summarized in Algorithms 3 and 4. There are two major differences between Algorithms 3 and 4 and Algorithms 1 and 2. First, the center initialization step of Algorithm 3 keeps generating an increasing number of centers until all the sequences are close to one of the existing centers. Second, an additional Merge Step in Algorithm 4 helps to combine clusters if the corresponding centers have small distances between each other.

Theorem 2.3.1. *Algorithm 4 converges after at most T_{max} iterations, where*

$$T_{max} = \sum_{\hat{K}=1}^{M} \binom{M}{\hat{K}} \hat{K}^{(M-\hat{K})}.$$

Moreover, if the data sequences generated from distributions satisfying Assumption 1 and the distance metric used by the algorithm satisfies Assumption 2,, then the error probability of Algorithm 4 after T iterations is upper bounded as follows

$$P_e \leq M^2 \left((T+1)\, a_1 + a_2 + (T+1)\, a_3 \right) e^{-bn},$$

where a_1, a_2, a_3 and b are as defined in Assumption 2 and $T \leq T_{max}$.

Proof. The proof shares the same idea as that of Theorem 2.2.1. See Appendix B.2 for details. □

Algorithm 4 Merge-based clustering with unknown K

Input: Data sequences $\{\mathbf{y}_i\}_{i=1}^{M}$ and threshold d_{th}.
Output: Partition set $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$.
Initialize $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$ by Algorithm 3.
while not converge **do**
    Center update specified in Algorithm 2.
    {Merge Step}
    for $k_1, k_2 \in \{1, \ldots, \hat{K}\}$ and $k_1 \neq k_2$ **do**
        if $d(\mathbf{c}_{k_1}, \mathbf{c}_{k_2}) \leq d_{th}$ **then**
            if $\sum_{\mathbf{y}_i \in \mathcal{C}_{k_1}} d(\mathbf{c}_{k_2}, \mathbf{y}_i) < \sum_{\mathbf{y}_i \in \mathcal{C}_{k_2}} d(\mathbf{c}_{k_1}, \mathbf{y}_i)$ **then**
                $\mathcal{C}_{k_2} \leftarrow \mathcal{C}_{k_1} \cup \mathcal{C}_{k_2}$ and delete $\mathbf{c}_{k_1}$ and $\mathcal{C}_{k_1}$.
            else
                $\mathcal{C}_{k_1} \leftarrow \mathcal{C}_{k_1} \cup \mathcal{C}_{k_2}$ and delete $\mathbf{c}_{k_2}$ and $\mathcal{C}_{k_2}$.
            end if
            $\hat{K} \leftarrow \hat{K} - 1$.
        end if
    end for
    Cluster update specified in Algorithm 2.
end while
Return $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$

Theorem 2.3.1 shows that the merge-based algorithm is exponentially consistent given distributions satisfying Assumption 1 under any distance metric satisfying Assumption 2 with the error exponent b.

Corollary 2.3.1.1. *Suppose the KS distance and the MMD statistic are used with* $d_{th} = \frac{\Sigma_{ks}}{2}$ *and* $d_{th} = \frac{\Sigma_{mmd}}{2}$. *Then for* n *sufficiently large, the error probability of Algorithm 4 after* T *iterations is upper bounded as follows*

$$P_e^{KS} \leq M^2 (10T + 14) \exp\left(-\frac{n\Delta_{ks}^2}{8}\right),$$
$$P_e^{MMD} \leq M^2 (6T + 8) \exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right).$$

Proof. By Propositions 1, the upper bound on the error probability of Algorithm 4 in Theorem 2.3.1 applies to the KS distance and the MMD statistic. Thus, the corollary is obtained by substituting the values specified in Lemmas A.0.3 - A.0.8 in the upper bound. □

Algorithm 5 Split-based clustering with unknown K

Input: Data sequences $\{\mathbf{y}_i\}_{i=1}^M$ and threshold d_{th}.
Output: Partition set $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$.
$\mathcal{C}_1 = \{\mathbf{y}_i\}_{i=1}^M$, $\hat{K} = 1$ and find $\mathbf{c}_1$ by center update specified in Algorithm 2.
while not converge **do**
    {Split Step}
    if $\max_{k \in I_1^{\hat{K}},\, \mathbf{y}_i \in \mathcal{C}_k} d(\mathbf{c}_k, \mathbf{y}_i) > d_{th}$ **then**
        $\hat{K} \leftarrow \hat{K} + 1$.
        $k = \arg\max_{k \in I_1^{\hat{K}}} \left(\max_{\mathbf{y}_i \in \mathcal{C}_k} d(\mathbf{c}_k, \mathbf{y}_i)\right)$
        $\mathbf{c}_{\hat{K}} \leftarrow \arg\max_{\mathbf{y}_i \in \mathcal{C}_k} d(\mathbf{c}_k, \mathbf{y}_i)$
    end if
    Cluster update specified in Algorithm 2.
end while
Return $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$

Corollary 2.3.1.1, combined with the fact that T is finitely bounded for finite M and K, implies that Algorithm 4 is exponentially consistent under both the KS and MMD distance metrics with an error exponent no smaller than $\frac{\Delta_{ks}^2}{8}$ and $\frac{\Delta_{mmd}^2}{64\mathbb{G}}$, respectively.

2.3.2 Split Step

Suppose a cluster contains sequences generated by different distributions and the center is generated from $p \in \mathcal{P}_k$. Then if the distance metric satisfies (2.4a), the probability that the distances between sequences generated from distribution clusters other than $\mathcal{P}_k$ and the center is small decays as the sample size increases. Therefore, it is reasonable to begin with one cluster and then split a cluster if there exists a sequence in the cluster that has a large distance to the center. The corresponding algorithm is summarized in Algorithm 5.

Definition 2.3.1.1. *Suppose Algorithm 5 obtains $\hat{K}$ clusters at the t-th iteration, where $\hat{K} < K$ and $\hat{K} = t$ or $t+1$. Then the correct clustering update result is that each cluster contains all the sequences generated from the distribution cluster that generates the center.*

Theorem 2.3.2. *Algorithm 5 converges after at most M iterations. Moreover, under Assumptions 1 and 2, the error probability of Algorithm 5 after T iterations is upper bounded*

as follows

$$P_e \leq M^2 T \left(a_1 + a_2 + a_3\right) e^{-bn},$$

where a_1, a_2, a_3 *and* b *are as defined in Assumption 2 and* $T \leq M$.

Outline of the Proof. An error occurs at the t-th iteration if and only if the $\hat{K}$-th center is generated from distribution clusters that generated the previous centers or the clustering result is incorrect. Note that the error event of the first T iterations is the union of the events that an error occurs at the t-th iteration while the clustering results in the previous $t-1$ iterations are correct for $t = 1, \ldots, T$. Similar to the proof of Theorem 2.2.1, the error probability is bounded by the union bound. See Appendix B.3 for more details. □

Theorem 2.3.2 shows that the split-based algorithm is exponentially consistent given distributions satisfying Assumption 1 under any distance metric satisfying Assumption 2 with the error exponent b.

Corollary 2.3.2.1. *Suppose the KS distance and the MMD statistic are used with* $d_{th} = \frac{\Sigma_{ks}}{2}$ *and* $d_{th} = \frac{\Sigma_{mmd}}{2}$. *Then for* n *sufficiently large, the error probability of Algorithm 5 after* T *iterations is upper bounded as follows*

$$P_e^{KS} \leq 14 M^2 T \exp\left(-\frac{n\Delta_{ks}^2}{8}\right),$$
$$P_e^{MMD} \leq 8 M^2 T \exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right).$$

Proof. By Propositions 1, the upper bound on the error probability of Algorithm 5 in Theorem 2.3.2 applies to the KS distance and the MMD statistic. Thus, the corollary is obtained by substituting the values specified in Lemmas A.0.3 - A.0.8 in the upper bound. □

Corollary 2.3.2.1, combined with the fact that T is finitely bounded for finite M, implies that Algorithm 5 is exponentially consistent under both the KS and MMD with an error exponent no smaller than $\frac{\Delta_{ks}^2}{8}$ and $\frac{\Delta_{mmd}^2}{64\mathbb{G}}$, respectively.

2.4 Numerical Results

In this section, we provide some simulation results given $K = 5$, $M_k = 5$ for $k = 1, \ldots, 5$, and $\mathbf{x}_{k,j_k}[i] \in \mathbb{R}$. Gaussian distributions $\mathcal{N}(\mu_{k,j_k}, \sigma^2)$ and Gamma distributions $\Gamma(\alpha_{k,j_k}, \beta)$ are used in the simulations. The probability density function (p.d.f.) of $\Gamma(\alpha, \beta)$ is defined as

$$f(x; \alpha, \beta) = \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} \exp\left(-\frac{x}{\beta}\right) \quad (x > 0),$$

where $\alpha > 0$, $\beta > 0$ and $\Gamma(\cdot)$ is the Gamma function, respectively. For this experiment, we set $\sigma = 1$, $\beta = 1$, and

$$\mu_{k,j_k} = (k-1) + \left(j_k - \frac{M_k + 1}{2}\right)\frac{\delta}{2},$$
$$\alpha_{k,j_k} = 2.5(k-1) + \left(j_k - \frac{M_k + 1}{2}\right)\frac{\delta}{2} + 1,$$

where $j_k = 1, \ldots, 5$, $\delta = 0$ and 0.1. Note that when $\delta = 0$, sequences belonging to the same distribution cluster are generated from a single distribution. The squared exponential kernel function is used for the MMD distance, i.e.,

$$g(x, y) = e^{-\frac{(x-y)^2}{2}}. \tag{2.8}$$

The Monte Carlo experiment for a given sample size continues until the following two conditions are both satisfied:

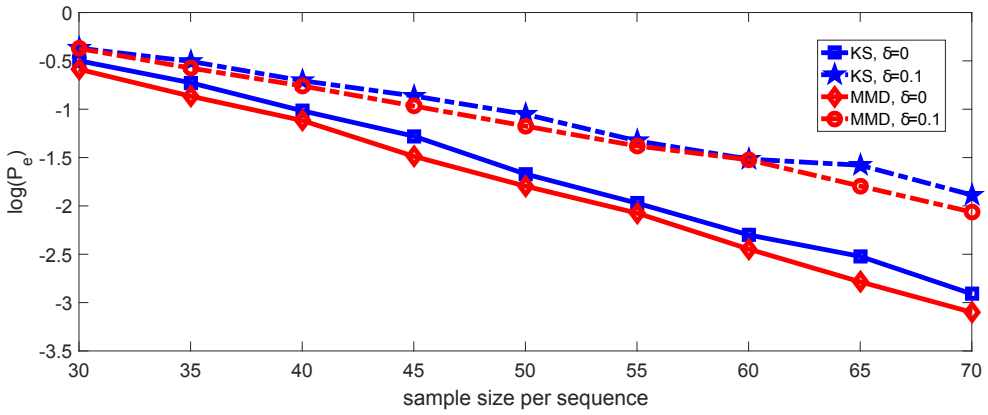
1. the number of trials that provide incorrect clustering output reaches 1000,
2. the total number of trials reaches 5×10^4.

2.4.1 Known Number of Clusters

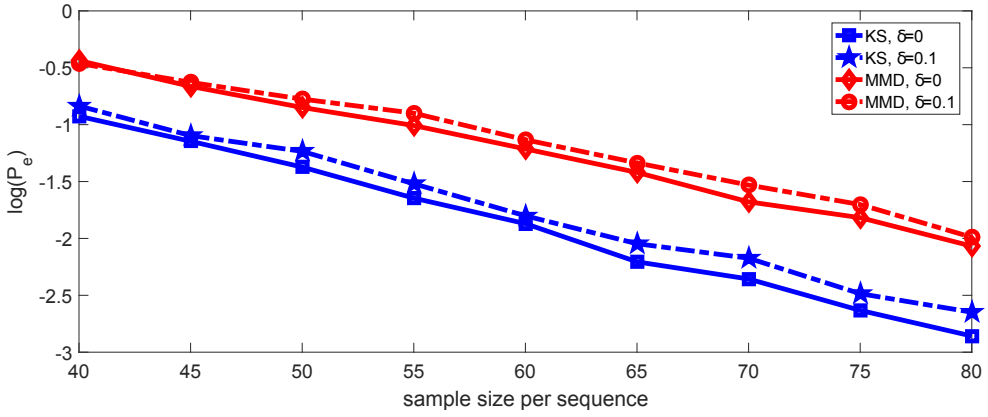
Simulation results for a known number of clusters are shown in Fig. 2.1. One can observe from the figures that by using both the KS distance and MMD, $\log P_e$ is a linear function of

the sample size, i.e., P_e is exponentially consistent. Moreover, the logarithmic slope of P_e with respect to n, i.e., the quantity $-\frac{\log P_e}{n}$, increases as δ becomes smaller, which, in the current simulation setting, implies a larger Δ.

Furthermore, a good distance metric for Algorithm 2 depends on the underlying distributions. The kernel function in (2.8) is a good choice given symmetric p.d.f.s whereas the KS distance which relates to the order statistics becomes a better choice when the p.d.f.s are skewed.



(a) Gaussian distributions



(b) Gamma distributions

Figure 2.1: Performance of Algorithm 2

2.4.2 Unknown Number of Clusters

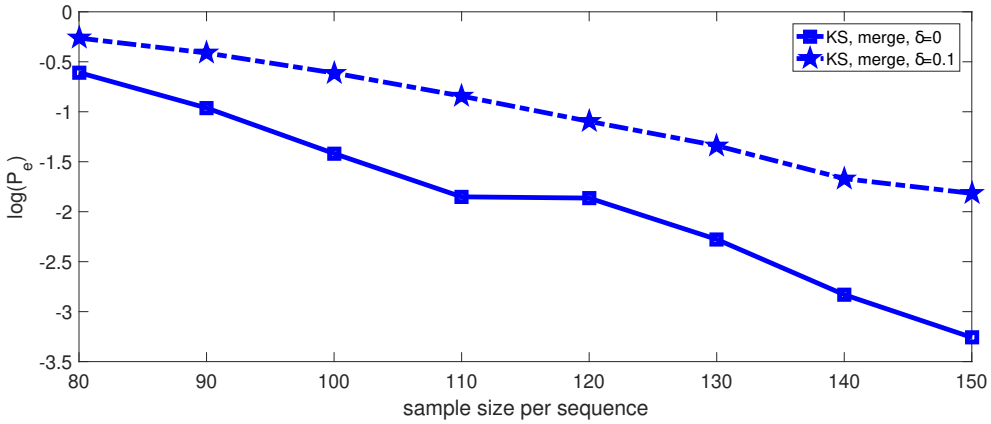
With an unknown number of distribution clusters, the threshold d_{th} specified in Corollaries 2.3.1.1 and 2.3.2.1 are used in the simulation. The performance of Algorithms 4 and 5 for the KS distance and MMD are shown in Figs. 2.2 - 2.5, respectively. Given the KS distance and MMD, $\log P_e$'s are linear functions of the sample size when the sample size is large and larger Δ implies a larger slope of $\log P_e$.

Intuitively, smaller δ implies larger Δ in the current simulation setting, thereby should result in better clustering performance for a given sample size. Figs. 2.3 and 2.5 indicates that Algorithms 4 and 5 with the KS distance and MMD performs better with $\delta = 0.1$ than that with $\delta = 0$ when the sample size is small. This is likely due to the fact that 1) the KS distance between the two sequences is always lower bounded by $\frac{1}{n}$, 2) the MMD estimator in (2.7) always has a positive bias, 3) the Gaussan kernel in (2.8) may not be a good choice for skewed p.d.f.s. Thus, with small sample sizes, Algorithms 4 and 5 are likely to overestimate the number of clusters.

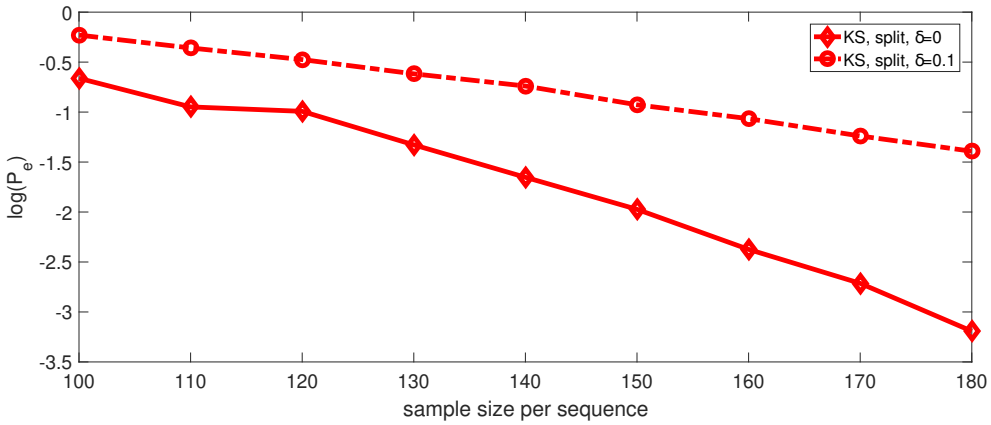
In Tables 2.1 - 2.8, the frequencies of the cases where $\hat{K} = K$ and $\hat{K} > K$ corresponding to Figs. 2.2 - 2.5 are provided. From Tables 2.1 - 2.8, we can conclude that under both KS and MMD, the algorithms tend to overestimate the number of clusters given $d_{th} = \frac{d_L + d_H}{2}$.

Table 2.1: $\hat{K} = K / \hat{K} > K$ in Fig. 2.2a

	n	80	90	100	110	120	130	140	150
$\delta = 0$	$P(\hat{K} = K)$	0.46	0.62	0.76	0.84	0.84	0.90	0.94	0.96
	$P(\hat{K} > K)$	0.54	0.38	0.24	0.16	0.16	0.10	0.06	0.04
$\delta = 0.1$	$P(\hat{K} = K)$	0.23	0.34	0.46	0.57	0.67	0.74	0.81	0.84
	$P(\hat{K} > K)$	0.77	0.66	0.54	0.43	0.33	0.26	0.19	0.16



(a) merge

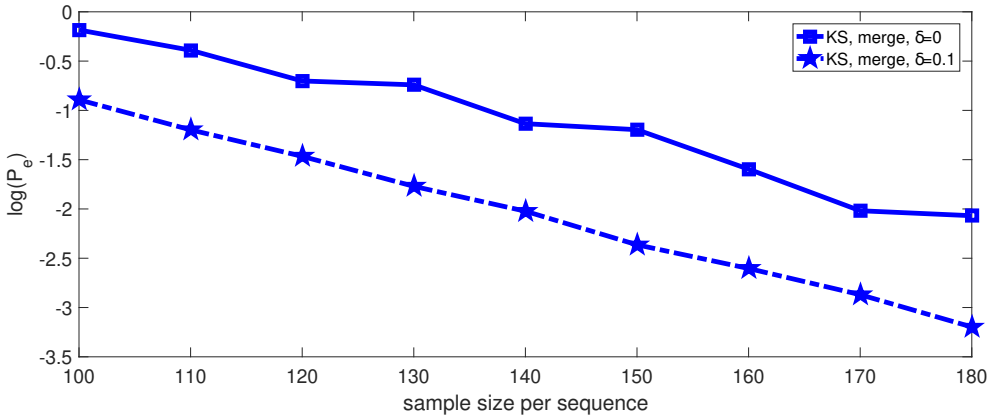


(b) split

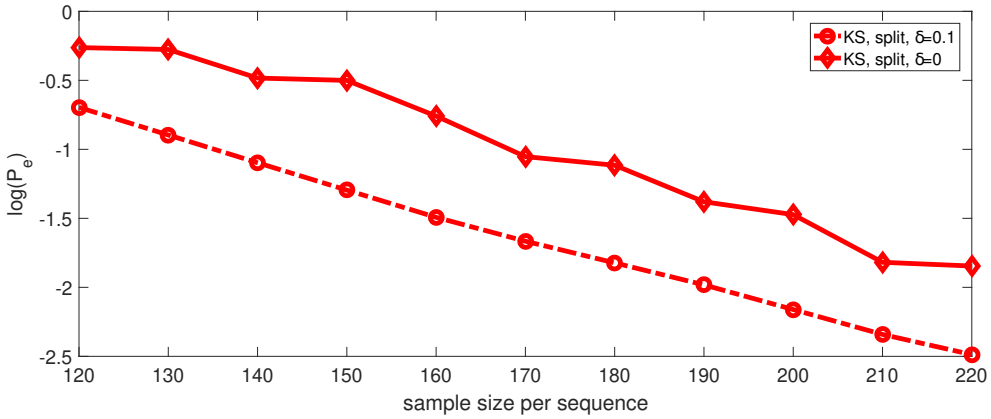
Figure 2.2: Performance of Algorithms 4 and 5 for the KS distance given Gaussian distributions

Table 2.2: $\hat{K} = K / \hat{K} > K$ in Fig. 2.2b

	n	100	110	120	130	140	150	160	170	180
$\delta = 0$	$P(\hat{K} = K)$	0.48	0.61	0.63	0.74	0.81	0.86	0.91	0.93	0.96
	$P(\hat{K} > K)$	0.52	0.39	0.37	0.27	0.19	0.14	0.10	0.07	0.04
$\delta = 0.1$	$P(\hat{K} = K)$	0.21	0.30	0.38	0.46	0.52	0.60	0.66	0.71	0.75
	$P(\hat{K} > K)$	0.79	0.70	0.62	0.54	0.48	0.40	0.34	0.29	0.25



(a) merge



(b) split

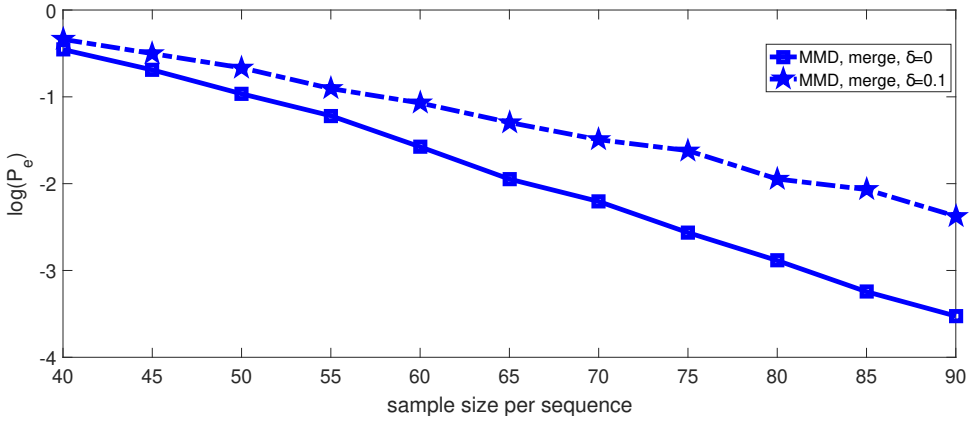
Figure 2.3: Performance of Algorithms 4 and 5 for the KS distance given Gamma distributions

Table 2.3: $\min \hat{K} / \max \hat{K}$ in Fig. 2.3a

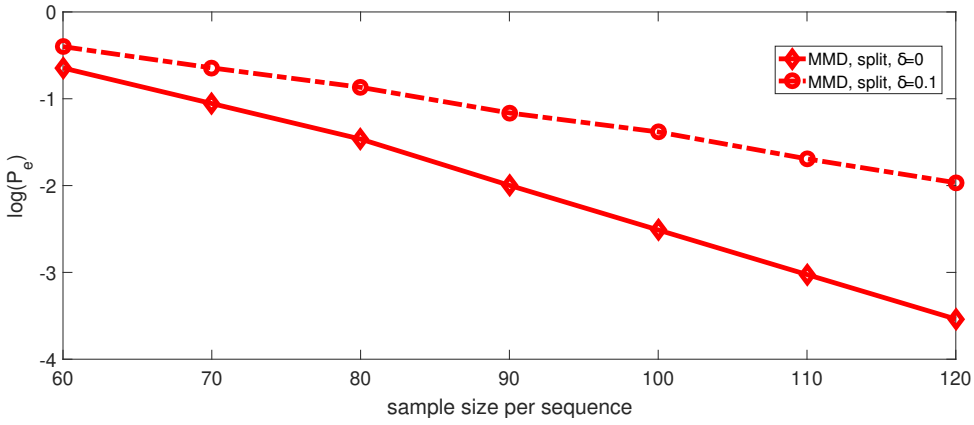
	n	100	110	120	130	140	150	160	170	180
$\delta = 0$	$P(\hat{K} = K)$	0.17	0.32	0.50	0.52	0.68	0.70	0.80	0.87	0.87
	$P(\hat{K} > K)$	0.83	0.68	0.50	0.48	0.32	0.30	0.20	0.13	0.13
$\delta = 0.1$	$P(\hat{K} = K)$	0.59	0.70	0.77	0.83	0.87	0.90	0.93	0.94	0.96
	$P(\hat{K} > K)$	0.41	0.30	0.23	0.17	0.13	0.09	0.07	0.06	0.04

Table 2.4: $\min \hat{K} / \max \hat{K}$ in Fig. 2.3b

	n	140	150	160	170	180	190	200	210	220
$\delta = 0$	$P(\hat{K} = K)$	0.38	0.39	0.53	0.65	0.67	0.75	0.77	0.84	0.84
	$P(\hat{K} > K)$	0.62	0.61	0.47	0.35	0.33	0.25	0.23	0.16	0.16
$\delta = 0.1$	$P(\hat{K} = K)$	0.67	0.73	0.78	0.81	0.84	0.86	0.88	0.90	0.92
	$P(\hat{K} > K)$	0.33	0.27	0.22	0.19	0.16	0.14	0.12	0.10	0.08



(a) merge



(b) split

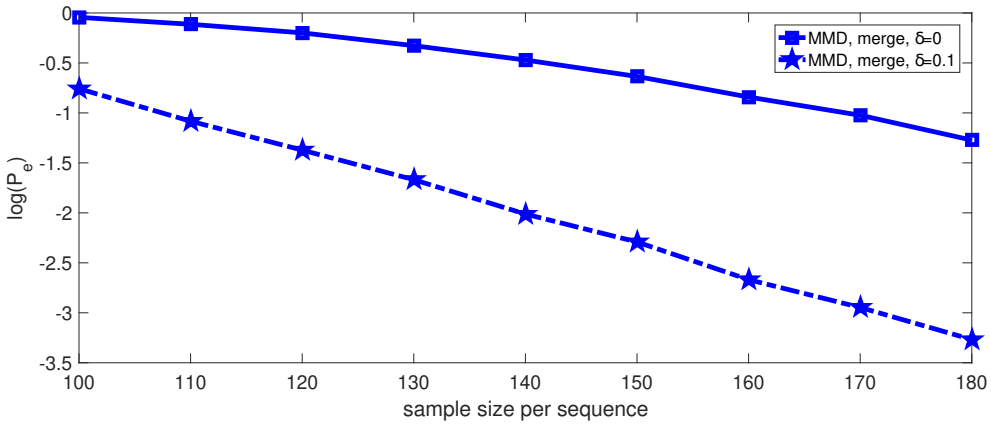
Figure 2.4: Performance of Algorithms 4 and 5 for MMD given Gaussian distributions

Table 2.5: $\min \hat{K} / \max \hat{K}$ in Fig. 2.4a

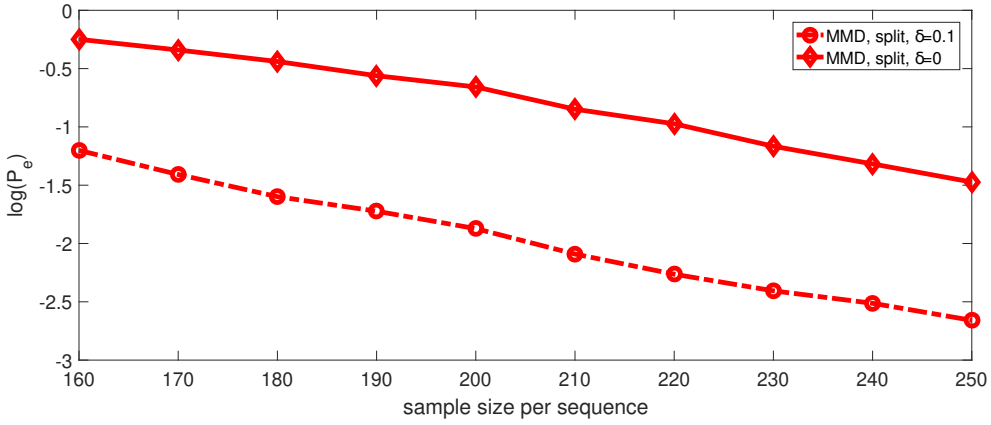
	n	50	55	60	65	70	75	80	85	90
$\delta = 0$	$P(\hat{K} = K)$	0.64	0.73	0.80	0.86	0.90	0.93	0.95	0.96	0.97
	$P(\hat{K} > K)$	0.36	0.28	0.20	0.14	0.10	0.07	0.05	0.04	0.03
$\delta = 0.1$	$P(\hat{K} = K)$	0.53	0.64	0.69	0.75	0.80	0.82	0.87	0.88	0.91
	$P(\hat{K} > K)$	0.47	0.36	0.31	0.25	0.20	0.18	0.13	0.12	0.09

Table 2.6: $\min \hat{K} / \max \hat{K}$ in Fig. 2.4b

	n	60	70	80	90	100	110	120
$\delta = 0$	$P(\hat{K} = K)$	0.49	0.66	0.77	0.87	0.92	0.95	0.97
	$P(\hat{K} > K)$	0.51	0.34	0.34	0.13	0.08	0.05	0.03
$\delta = 0.1$	$P(\hat{K} = K)$	0.36	0.50	0.60	0.70	0.76	0.82	0.87
	$P(\hat{K} > K)$	0.64	0.50	0.40	0.30	0.24	0.18	0.13



(a) merge



(b) split

Figure 2.5: Performance of Algorithms 4 and 5 for MMD given Gamma distributions

Table 2.7: $\min \hat{K} / \max \hat{K}$ in Fig. 2.5a

	n	100	110	120	130	140	150	160	170	180
$\delta = 0$	$P(\hat{K} = K)$	0.04	0.10	0.18	0.30	0.38	0.47	0.57	0.64	0.72
	$P(\hat{K} > K)$	0.96	0.89	0.82	0.72	0.62	0.53	0.43	0.36	0.28
$\delta = 0.1$	$P(\hat{K} = K)$	0.54	0.67	0.75	0.81	0.87	0.90	0.93	0.95	0.96
	$P(\hat{K} > K)$	0.46	0.33	0.25	0.19	0.13	0.10	0.07	0.05	0.04

Table 2.8: $\min \hat{K} / \max \hat{K}$ in Fig. 2.5b

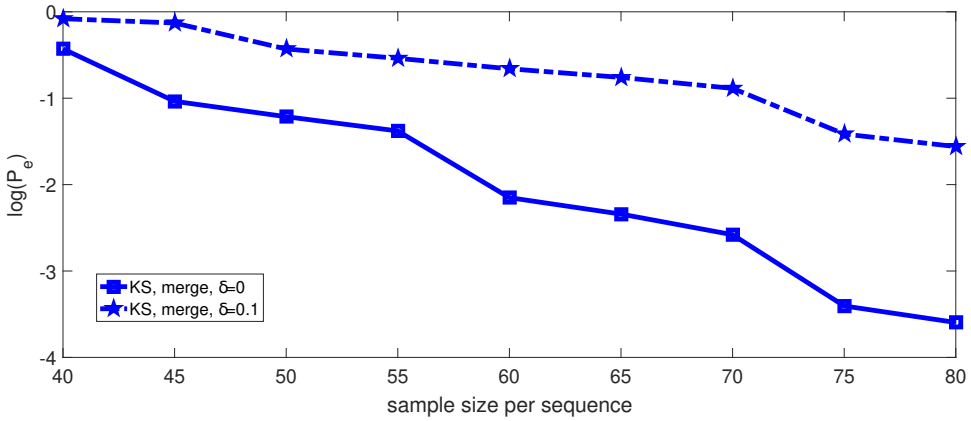
	n	170	180	190	200	210	220	230	240	250
$\delta = 0$	$P(\hat{K} = K)$	0.29	0.36	0.43	0.48	0.57	0.62	0.69	0.73	0.77
	$P(\hat{K} > K)$	0.71	0.64	0.57	0.52	0.43	0.38	0.31	0.27	0.23
$\delta = 0.1$	$P(\hat{K} = K)$	0.76	0.80	0.82	0.85	0.88	0.90	0.91	0.92	0.93
	$P(\hat{K} > K)$	0.24	0.20	0.18	0.15	0.12	0.10	0.09	0.08	0.07

2.4.3 Choice of d_{th}

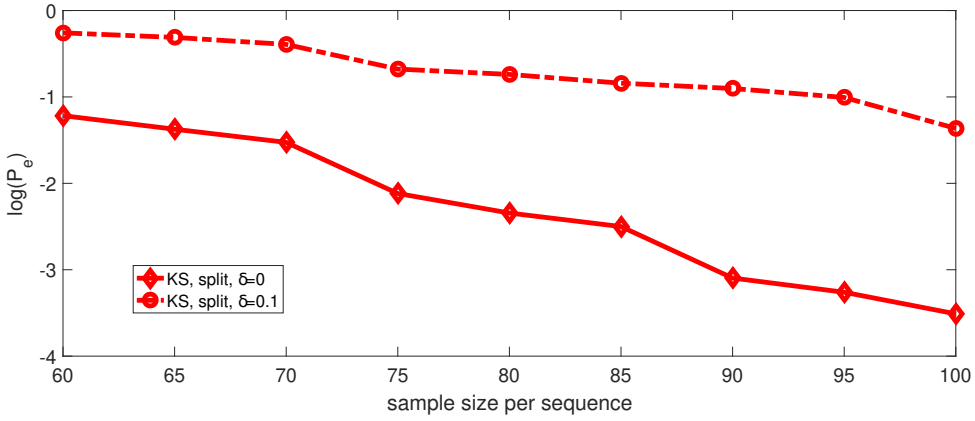
Note that in general $d_{th} = \alpha d_L + (1 - \alpha) d_H$, where $\alpha \in (0, 1)$. Theorems 2.3.1 and 2.3.2 only establish the exponential consistency of Algorithms 4 and 5, respectively. One can observe from Tables 2.1 - 2.8 that given $\alpha = 0.5$, Algorithms 4 and 5 tend to overestimate the number of clusters, which may imply larger error probability. In Figs. 2.6 - 2.9, the performance of Algorithms 4 and 5 with $\alpha < 0.5$ is provided. The performance of the two algorithms is indeed improved by choosing smaller α, i.e., larger d_{th}. The frequencies of the cases where $\hat{K} = K$ and $\hat{K} > K$ corresponding to Fig. 2.6 - 2.9 are provided in Tables 2.9 - 2.15.

Table 2.9: $\hat{K} = K / \hat{K} > K$ in Fig. 2.6a

	n	40	45	50	55	60	65	70	75	80
$\delta = 0$	$P(\hat{K} = K)$	0.39	0.71	0.75	0.78	0.91	0.92	0.94	0.97	0.98
	$P(\hat{K} > K)$	0.60	0.28	0.25	0.22	0.09	0.08	0.06	0.02	0.02
$\delta = 0.1$	$P(\hat{K} = K)$	0.09	0.13	0.39	0.45	0.51	0.55	0.60	0.78	0.80
	$P(\hat{K} > K)$	0.91	0.87	0.61	0.55	0.49	0.45	0.40	0.22	0.20



(a) merge

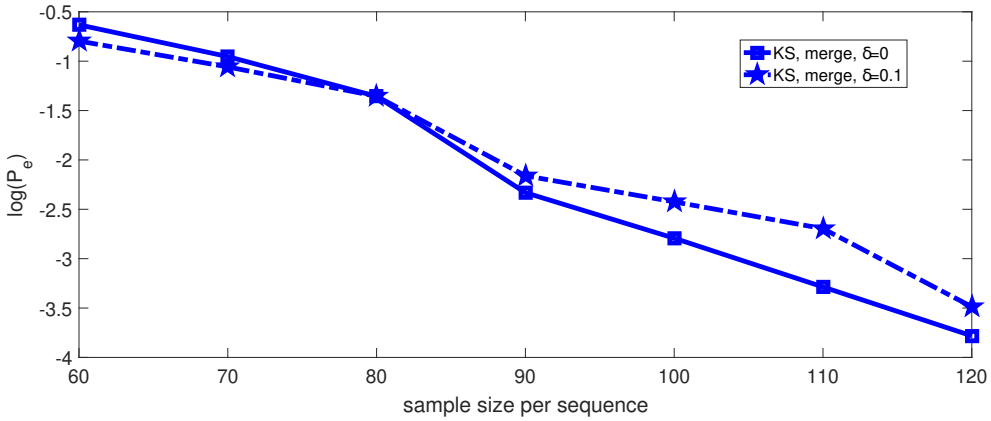


(b) split

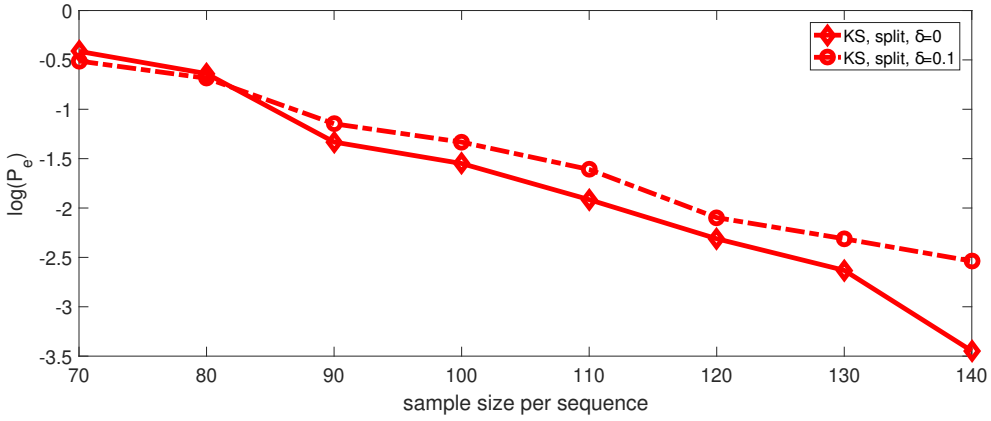
Figure 2.6: Performance of Algorithms 4 and 5 for the KS distance given Gaussian distributions with $\alpha = 0.3$

Table 2.10: $\hat{K} = K / \hat{K} > K$ in Fig. 2.6b

	n	60	65	70	75	80	85	90	95	100
$\delta = 0$	$P(\hat{K} = K)$	0.76	0.78	0.81	0.91	0.92	0.93	0.97	0.97	0.98
	$P(\hat{K} > K)$	0.24	0.22	0.19	0.09	0.08	0.07	0.03	0.03	0.02
$\delta = 0.1$	$P(\hat{K} = K)$	0.24	0.28	0.33	0.52	0.55	0.59	0.61	0.65	0.76
	$P(\hat{K} > K)$	0.76	0.72	0.67	0.48	0.45	0.41	0.39	0.35	0.24



(a) merge



(b) split

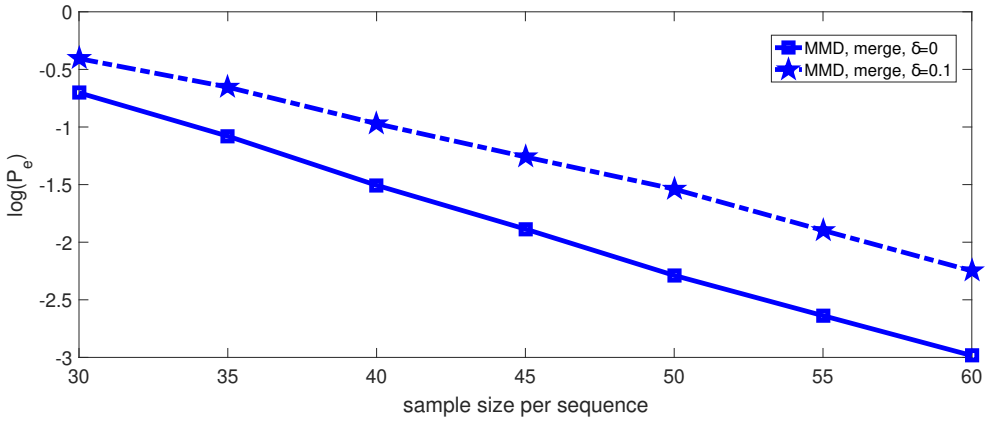
Figure 2.7: Performance of Algorithms 4 and 5 for the KS distance given Gamma distributions with $\alpha = 0.3$

Table 2.11: $\min \hat{K} / \max \hat{K}$ in Fig. 2.7a

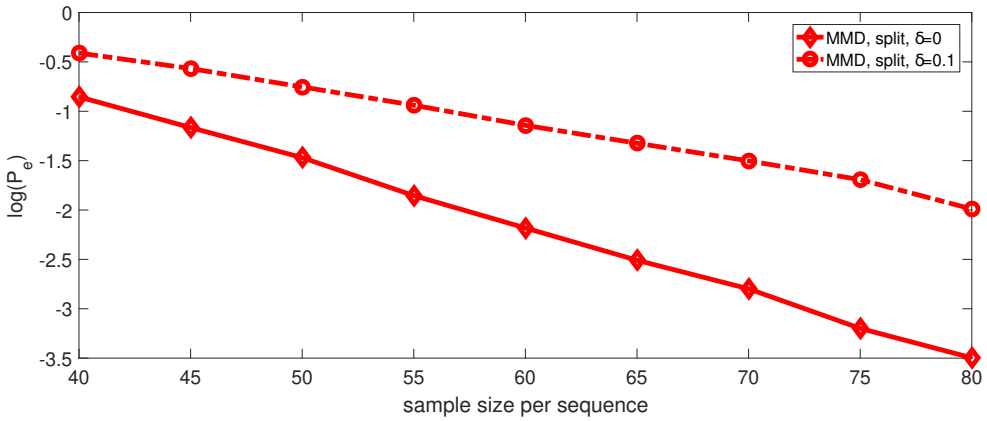
	n	60	70	80	90	100	110	120
$\delta = 0$	$P(\hat{K} = K)$	0.49	0.63	0.75	0.91	0.94	0.96	0.98
	$P(\hat{K} > K)$	0.51	0.37	0.25	0.09	0.06	0.04	0.02
$\delta = 0.1$	$P(\hat{K} = K)$	0.58	0.67	0.75	0.89	0.92	0.94	0.97
	$P(\hat{K} > K)$	0.41	0.32	0.24	0.10	0.08	0.06	0.03

Table 2.12: $\min \hat{K} / \max \hat{K}$ in Fig. 2.7b

	n	70	80	90	100	110	120	130	140
$\delta = 0$	$P(\hat{K} = K)$	0.35	0.48	0.75	0.79	0.86	0.90	0.93	0.97
	$P(\hat{K} > K)$	0.65	0.52	0.25	0.21	0.14	0.10	0.07	0.03
$\delta = 0.1$	$P(\hat{K} = K)$	0.42	0.51	0.70	0.75	0.81	0.88	0.90	0.92
	$P(\hat{K} > K)$	0.58	0.49	0.30	0.25	0.19	0.12	0.10	0.08



(a) merge



(b) split

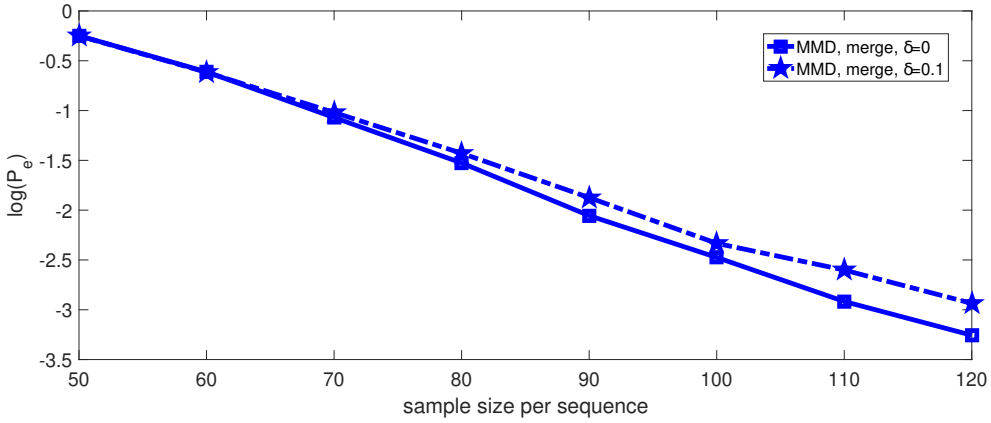
Figure 2.8: Performance of Algorithms 4 and 5 for MMD given Gaussian distributions with $\alpha = 0.3$

Table 2.13: $\min \hat{K} / \max \hat{K}$ in Fig. 2.8a

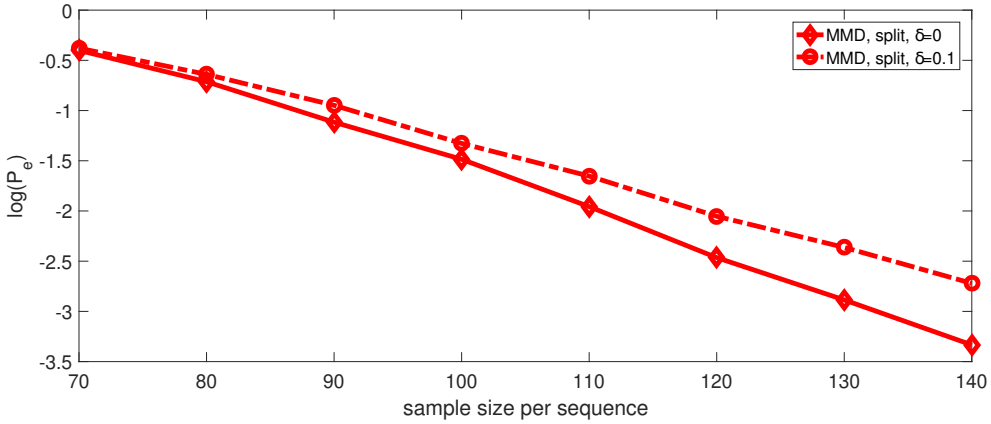
	n	30	35	40	45	50	55	60
$\delta = 0$	$P(\hat{K} = K)$	0.80	0.86	0.92	0.94	0.95	0.96	0.97
	$P(\hat{K} > K)$	0.09	0.05	0.02	0.01	0.01	0	0
$\delta = 0.1$	$P(\hat{K} = K)$	0.62	0.73	0.82	0.88	0.91	0.94	0.96
	$P(\hat{K} > K)$	0.35	0.24	0.15	0.10	0.07	0.05	0.03

Table 2.14: $\min \hat{K} / \max \hat{K}$ in Fig. 2.8b

	n	40	45	50	55	60	65	70	75	80
$\delta = 0$	$P(\hat{K} = K)$	0.87	0.93	0.95	0.97	0.98	0.99	0.99	1	1
	$P(\hat{K} > K)$	0.12	0.06	0.04	0.02	0.01	0.01	0	0	0
$\delta = 0.1$	$P(\hat{K} = K)$	0.58	0.67	0.75	0.82	0.85	0.89	0.92	0.94	0.95
	$P(\hat{K} > K)$	0.42	0.33	0.25	0.18	0.14	0.11	0.08	0.06	0.05



(a) merge



(b) split

Figure 2.9: Performance of Algorithms 4 and 5 for MMD given Gamma distributions with $\alpha = 0.2$

Table 2.15: $\min \hat{K} / \max \hat{K}$ in Fig. 2.9a

	n	50	60	70	80	90	100	110	120
$\delta = 0$	$P(\hat{K} = K)$	0.28	0.53	0.72	0.82	0.90	0.93	0.95	0.97
	$P(\hat{K} > K)$	0.70	0.44	0.25	0.15	0.07	0.04	0.02	0.01
$\delta = 0.1$	$P(\hat{K} = K)$	0.30	0.54	0.70	0.81	0.87	0.92	0.94	0.95
	$P(\hat{K} > K)$	0.68	0.42	0.25	0.15	0.08	0.04	0.02	0.02

Table 2.16: $\min \hat{K} / \max \hat{K}$ in Fig. 2.9b

	n	70	80	90	100	110	120	130	140
$\delta = 0$	$P(\hat{K} = K)$	0.22	0.29	0.36	0.43	0.48	0.57	0.62	0.69
	$P(\hat{K} > K)$	0.78	0.71	0.64	0.57	0.52	0.43	0.38	0.31
$\delta = 0.1$	$P(\hat{K} = K)$	0.70	0.76	0.80	0.82	0.85	0.88	0.90	0.91
	$P(\hat{K} > K)$	0.30	0.24	0.20	0.18	0.15	0.12	0.10	0.09

2.4.4 Modulation Clustering for Wireless Communications

In this subsection, merge based k-medoids algorithm under the KS distance is applied to an on-line data set of wireless communication signals with different modulations[1]. This data set include wireless signals corresponding to 11 modulation types, each with 1000 waveforms (magnitude only) of length 960. As no ground truth is given for the waveforms, our experiment using the digital communication signals is intended to demonstrate the exponential consistency, i.e., clustering errors decays exponentially as the sequence length increases. We consider clustering waveforms corresponding to three out of the 11 waveforms: 2 Amplitude-Shift Keying (2ASK), 4 Phase-Shift Keying (4PSK) and 16 Quadrature amplitude modulation (16QAM) with the signal-to-noise ratio (SNR) at 16dB and 20dB, respectively. A total of 100 waveforms of each modulation scheme is randomly selected in each of the 2000 trials. The clustering result is correct if and only if the 300 waveforms are correctly grouped according to their corresponding modulation schemes. As the underlying distributions of these waveforms are unknown, d_{th} is determined empirically using the waveforms for the three modulations. The performance of k-medoids algorithm under the KS distance is given in Fig. 2.10. It is clear that $\log(P_e)$ is approximately a linear function

[1]Data set is available at https://github.com/bczhangbczhang/Communication-Signal-Dataset

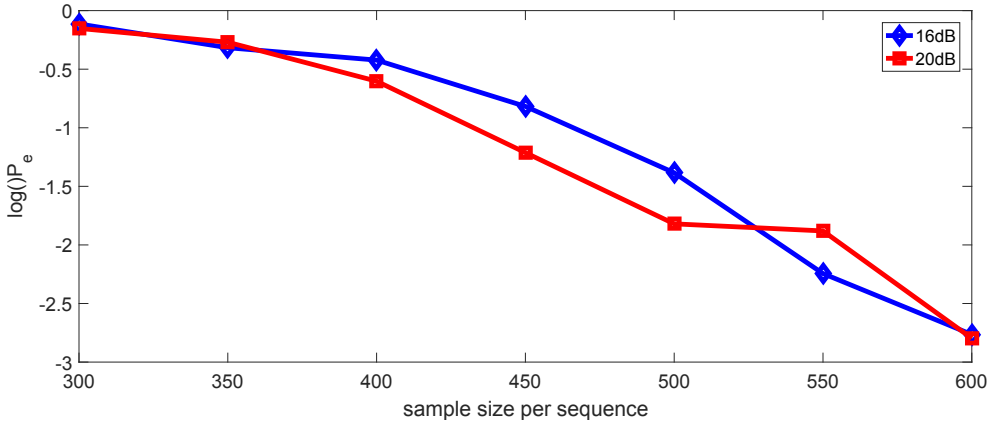


Figure 2.10: Performance of Modulation clustering by Algorithm 4 given $d_{th} = 0.1$

when sample size n becomes large.

2.4.5 Computational Complexity

Assume that the complexity of the sum and point-wise max/min operations is linear in the argument cardinality. The complexity of other operations is assumed to be $O(1)$. The computational complexities of the center initialization step and the cluster initialization step in Algorithm 1 are $O(K^2M)$ and $O(KM)$, respectively. The computational complexity of the center update step and cluster update step in Algorithm 2 are $O(KM^2)$ and $O(KM)$. Thus, the computational complexity of Algorithm 2 is $O\left(\binom{M}{K}K^{(M-K+1)}M^2\right)$

Similarly, one can verify that the computational complexities of the center initialization step and the cluster initialization step in Algorithm 3 are $O(M^3)$ and $O(M^2)$, respectively. The computational complexity of the center update step, the merge step and the cluster update step in Algorithm 4 are $O(M^3)$, $O(M^3)$ and $O(M^2)$. Thus, the computational complexity of Algorithm 4 is $O(M^3T_{max})$.

The computational complexities of the finding $\mathbf{c}_1^1$, the split step and the cluster update step in Algorithm 5 are $O(M^2)$, $O(M)$ and $O(M^2)$, respectively. Thus, the computational complexity of Algorithm 5 is $O(M^3)$.

2.5 Summary

This chapter studied the k-medoids algorithm for clustering data sequences generated from composite distributions. The convergence of the proposed algorithms and the upper bound on the error probability were analyzed for both known and unknown number of clusters. The exponential decay of error probabilities of the proposed algorithms was established for distance metrics satisfying certain properties. In particular, the KS distance and MMD were shown to satisfy the required condition, and hence the corresponding algorithms were exponentially consistent. Note that the assumption of knowing d_L and d_H (or their bounds) can be justified because the empirical KS distance and MMD can be constructed, which converge to the true KS distance and MMD. Thus these thresholds or their bounds can be obtained from historical data.

CHAPTER 3
SEQUENCE CLUSTERING BY HIERARCHICAL AGGLOMERATIVE CLUSTERING ALGORITHMS

This chapter focuses on asymptotic performance study of sequence clustering using hierarchical agglomerative clustering algorithms. The HAC algorithms with LWD update are introduced. The upper bound on the error probability of linkage-based HAC algorithms with unknown number of clusters is derived, followed by parallel results of centroid-based HAC algorithms with an unknown number of clusters. The derived upper bounds are shown to decay exponentially as the sample size increases, establishing the exponential consistency of a large set of HAC algorithms.

3.1 HAC Algorithms with LWD Update

In the previous chapter, the exponential consistency of k-medoids algorithm which updates centroid and clustering result iteratively was established. In this chapter, we consider another popular class of clustering algorithms, the hierarchical agglomerative clustering al-

gorithms, which starts with clusters containing a single data sequence and then merge two clusters with the smallest distance in every iteration until the minimum pairwise dissimilarity among remaining clusters is greater than some pre-determined threshold d_{th}. Given a dissimilarity matrix consisting of the pairwise distance between all data sequences, HAC algorithms iteratively update the dissimilarity matrix by LWD update formula.

Let $\mathcal{C}_l$, $l = 1, \cdots, L$, denote the l-th sequence cluster. The dissimilarity matrix of L clusters is defined as

$$\mathbf{D} = \begin{bmatrix} 0 & d(\mathcal{C}_1, \mathcal{C}_2) & \cdots & d(\mathcal{C}_1, \mathcal{C}_L) \\ d(\mathcal{C}_2, \mathcal{C}_1) & 0 & \cdots & d(\mathcal{C}_2, \mathcal{C}_L) \\ \vdots & \vdots & \vdots & \vdots \\ d(\mathcal{C}_L, \mathcal{C}_1) & d(\mathcal{C}_L, \mathcal{C}_2) & \cdots & 0 \end{bmatrix},$$

where $d(\mathcal{C}_l, \mathcal{C}_{l'})$ is the dissimilarity (i.e., the distance metric) between clusters $\mathcal{C}_l$ and $\mathcal{C}_{l'}$ and satisfies 1) $d(\mathcal{C}_l, \mathcal{C}_{l'}) \geq 0$, 2) $d(\mathcal{C}_l, \mathcal{C}_l) = 0$, and 3) $d(\mathcal{C}_l, \mathcal{C}_{l'}) = d(\mathcal{C}_{l'}, \mathcal{C}_l)$. In each iteration, HAC algorithms try to merge two clusters $\mathcal{C}_{l_1}$ and $\mathcal{C}_{l_2}$ if

$$d(\mathcal{C}_{l_1}, \mathcal{C}_{l_2}) = \min_{l \neq l'} d(\mathcal{C}_l, \mathcal{C}_{l'}) \leq d_{th},$$

with d_{th} a pre-determined threshold. The algorithm stops if

$$\min_{l \neq l'} d(\mathcal{C}_l, \mathcal{C}_{l'}) > d_{th}.$$

The general HAC algorithm is summarized in Algorithm 6. Note that any HAC algorithm converges within a finite number of steps which is at most M. The LWD update formula provides a unified view for dissimilarity updating after each merge step [32]. Suppose $\mathcal{C}_{l_1}$

Algorithm 6 HAC Algorithm

1: **Input**: Data sequences $\{\mathbf{y}_i\}_{i=1}^{M}$ and threshold d_{th}.
2: **Output**: Partition set $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$.
3: $\mathcal{C}_i = \{\mathbf{y}_i\}$ for $i = 1, \ldots, M$, and construct the corresponding $\mathbf{D}$.
4: **while** $\min_{\mathcal{C}_l, \mathcal{C}_{l'} \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots\}} d(\mathcal{C}_l, \mathcal{C}_{l'}) \leq d_{th}$ **do**
5: Merge $\mathcal{C}_{l_1}$ and $\mathcal{C}_{l_2}$ if
$d(\mathcal{C}_{l_1}, \mathcal{C}_{l_2}) = \min_{\mathcal{C}_l, \mathcal{C}_{l'} \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots\}} (\mathcal{C}_l, \mathcal{C}_{l'})$,
6: Update the dissimilarity matrix $\mathbf{D}$.
7: **end while**
8: Return $\{\mathcal{C}_k\}_{k=1}^{\hat{K}}$

and $\mathcal{C}_{l_2}$ are merged. Then the LWD between $\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}$ and $\mathcal{C}_{l_3}$ is given by

$$\begin{aligned} d(\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}, \mathcal{C}_{l_3}) &= \alpha_1 d(\mathcal{C}_{l_1}, \mathcal{C}_{l_3}) + \alpha_2 d(\mathcal{C}_{l_2}, \mathcal{C}_{l_3}) \\ &\quad + \beta d(\mathcal{C}_{l_1}, \mathcal{C}_{l_2}) + \gamma \left| d(\mathcal{C}_{l_1}, \mathcal{C}_{l_3}) - d(\mathcal{C}_{l_2}, \mathcal{C}_{l_3}) \right|. \end{aligned} \tag{3.1}$$

The choices of coefficients in (3.1) for typical HAC algorithms are given in Table 3.1, where $|\mathcal{C}|$ denotes the cardinality of $\mathcal{C}$ [17]. For the rest of the section, linkage-based clustering algorithms with LWD update are assumed to satisfy

$$\alpha_i \geq 0 \text{ for } i = 1, 2, \tag{3.2a}$$

$$\alpha_1 + \alpha_2 = 1, \tag{3.2b}$$

$$|\gamma| \leq \min\{\alpha_1, \alpha_2\}, \tag{3.2c}$$

$$\beta = 0. \tag{3.2d}$$

Thus $d(\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}, \mathcal{C}_{l_3})$ in (3.1) is always non-negative and

$$\begin{aligned} d(\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}, \mathcal{C}_{l_3}) &\geq \min\{d(\mathcal{C}_{l_1}\mathcal{C}_{l_3}), d(\mathcal{C}_{l_2}\mathcal{C}_{l_3})\}, \\ d(\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}, \mathcal{C}_{l_3}) &\leq \max\{d(\mathcal{C}_{l_1}\mathcal{C}_{l_3}), d(\mathcal{C}_{l_2}\mathcal{C}_{l_3})\}. \end{aligned}$$

Eq. (3.2d) is necessary for linkage-based HAC algorithms and implies that $d(\mathcal{C}_l, \mathcal{C}_{l'})$ is only a function of $d(\mathbf{y}_i, \mathbf{y}_{i'})$, where $\mathbf{y}_i \in \mathcal{C}_l$ and $\mathbf{y}_{i'} \in \mathcal{C}_{l'}$. Furthermore, centroid-based

Table 3.1: Coefficients of HAC algorithms

SLINK	$\alpha_1 = \alpha_2 = 0.5, \beta = 0, \gamma = -0.5.$
CLINK	$\alpha_1 = \alpha_2 = 0.5, \beta = 0, \gamma = 0.5.$
UPGMA	$\alpha_1 = \frac{\lvert\mathcal{C}_{l_1}\rvert}{\lvert\mathcal{C}_{l_1}\rvert+\lvert\mathcal{C}_{l_2}\rvert}, \alpha_2 = 1-\alpha_1,$ $\beta = 0, \gamma = 0.$
WPGMA	$\alpha_1 = \alpha_2 = 0.5, \beta = 0, \gamma = 0.$
WPGMC	$\alpha_1 = \alpha_2 = 0.5, \beta = -0.25, \gamma = 0.$
UPGMC	$\alpha_1 = \frac{\lvert\mathcal{C}_{l_1}\rvert}{\lvert\mathcal{C}_{l_1}\rvert+\lvert\mathcal{C}_{l_2}\rvert}, \alpha_2 = 1-\alpha_1,$ $\beta = -\frac{\lvert\mathcal{C}_{l_1}\rvert\lvert\mathcal{C}_{l_2}\rvert}{\left(\lvert\mathcal{C}_{l_1}\rvert+\lvert\mathcal{C}_{l_2}\rvert\right)^2}, \gamma = 0.$

clustering algorithms with LWD update are assumed to satisfy

$$\alpha_i \geq 0 \text{ for } i = 1, 2, \tag{3.3a}$$

$$\alpha_1 + \alpha_2 = 1, \tag{3.3b}$$

$$\gamma = 0, \tag{3.3c}$$

$$\beta \in (-1, 0). \tag{3.3d}$$

3.2 Linkage-Based Algorithms

This section presents an upper bound on the error probability of the linkage-based clustering algorithms generated from the LWD update formula with coefficients satisfying (3.2). The complete proof of the results will be provided in the Appendix.

3.2.1 General Case

Proposition 2. *If the linkage-based clustering algorithm updates* $\mathbf{D}$ *by* (3.1)*, then for* $t \geq 0$ *and* $l \neq l'$*,*

$$d\left(\mathcal{C}_l^t, \mathcal{C}_{l'}^t\right) = \sum_{i:\, \mathbf{y}_i \in \mathcal{C}_l^t} \sum_{i':\, \mathbf{y}_{i'} \in \mathcal{C}_{l'}^t} \theta_{ii'}^t\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) d\left(\mathbf{y}_i, \mathbf{y}_{i'}\right), \tag{3.4}$$

where $\theta_{ii'}^t(\mathbf{y}_i, \mathbf{y}_{i'}) \in [0,1]$ *is a function of* t, i *and* i'. *Moreover, if the LWD update satisfies* (3.2)*, then for any* $l \neq l'$,

$$\sum_{i:\, \mathbf{y}_i \in \mathcal{C}_l^t} \sum_{i':\, \mathbf{y}_{i'} \in \mathcal{C}_{l'}^t} \theta_{ii'}^t \left(\mathbf{y}_i, \mathbf{y}_{i'}\right) = 1. \tag{3.5}$$

Outline of the Proof. (3.4) can be proved by induction while (3.5) results from (3.4) and (3.2). □

Intuitively, with (3.2), the updated metric in (3.1) can be rewritten as a convex combination of $d\left(\mathcal{C}_{l_1}, \mathcal{C}_{l_3}\right)$ and $d\left(\mathcal{C}_{l_2}, \mathcal{C}_{l_3}\right)$, leading to (3.5). For simplicity, $\theta_{ii'}^t\left(\mathbf{y}_i, \mathbf{y}_{i'}\right)$ will be replaced by $\theta_{ii'}^t$ if there is no ambiguity. The choice of $\theta_{ii'}^t$ for SLINK, CLINK and UPGMA is given in (3.6) - (3.8), respectively.

$$\theta_{ii'}^t = \begin{cases} 1 & \text{if } d\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) = \min_{\mathbf{y}_j \in \mathcal{C}_l, \mathbf{y}_{j'} \in \mathcal{C}_{l'}} d\left(\mathbf{y}_j, \mathbf{y}_{j'}\right), \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

$$\theta_{ii'}^t = \begin{cases} 1 & \text{if } d\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) = \max_{\mathbf{y}_j \in \mathcal{C}_l, \mathbf{y}_{j'} \in \mathcal{C}_{l'}} d\left(\mathbf{y}_j, \mathbf{y}_{j'}\right), \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

$$\theta_{ii'}^t = \frac{1}{\left|\mathcal{C}_l^t\right| \left|\mathcal{C}_{l'}^t\right|}. \tag{3.8}$$

Theorem 3.2.1. *Suppose a linkage-based clustering algorithm uses update in* (3.1) *and satisfies* (3.2) *and data sequences are generated from distributions satisfying* (2.3)*. If the distance metric used by the algorithm satisfies* (2.4a) *and* (2.4b)*, then for* $d_{th} \in (d_L, d_H)$ *and sufficiently large* n*, the error probability upon convergence is upper bounded by*

$$P_e \leq M^2 a_1 e^{-nb_1} + M^2 a_2 e^{-nb_2}.$$

Outline of the Proof. Note that by Assumption (2), the data sequences are well separated

with probability lower bounded by $\left(1 - M^2 a_1 e^{-nb_1}\right)\left(1 - M^2 a_2 e^{-nb_2}\right)$ which is futher lower bounded by $1 - M^2 a_1 e^{-nb_1} - M^2 a_2 e^{-nb_2}$. By proposition 2, if data sequences are well separated, the clusters obtained by linkage-based clustering algorithms are still well separated. Thus, the error probability is upper bounded by $M^2 a_1 e^{-nb_1} + M^2 a_2 e^{-nb_2}$. □

Corollary 3.2.1.1. *Suppose the KS distance and MMD are used with* $d_{th} = \frac{1}{2}\Sigma_{ks}$ *and* $d_{th} = \frac{1}{2}\Sigma_{mmd}$*, where* Σ_{ks} *and* Σ_{mmd} *are defined in* (2.2)*. Then for sufficiently large* n*, the error probability of linkage-based clustering algorithms upon convergence is upper bounded by*

$$P_e^{KS} \leq 8M^2 \exp\left(-\frac{n\Delta_{ks}^2}{8}\right),$$
$$P_e^{MMD} \leq 4M^2 \exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right),$$

where Δ_{ks} *and* Δ_{mmd} *are defined in* (2.2)

Proof. By Proposition 1, the error probability upper bound of linkage-based clustering algorithms in Theorem 3.2.1 applies to KS and MMD. The corollary is obtained by substituting a_i and b_i with values specified in Lemmas A.1 - A.4. □

Corollary 3.2.1.1 implies that any linkage-based clustering algorithm satisfying (3.2) is exponentially consistent under both the KS and MMD distance metrics with an error exponent no smaller than $\frac{\Delta_{ks}^2}{8}$ and $\frac{\Delta_{mmd}^2}{64\mathbb{G}}$, respectively.

3.2.2 Tighter bounds for SLINK

Tighter upper bounds on the error probability for SLINK can be derived by taking advantage of the fact the inter-cluster distance is computed using a single pair of sequences. The entry $d\left(\mathcal{C}_l, \mathcal{C}_{l'}\right)$ in $\mathbf{D}$ for SLINK is given by

$$d_S\left(\mathcal{C}_l, \mathcal{C}_{l'}\right) = \min_{\mathbf{y}_1 \in \mathcal{C}_l, \mathbf{y}_2 \in \mathcal{C}_{l'}} d\left(\mathbf{y}_1, \mathbf{y}_2\right). \tag{3.9}$$

The following theorem provides a tighter upper bound on the error probability of SLINK.

Theorem 3.2.2. *Under Assumptions 1 and 2, the error probability of SLINK for* $d_{th} \in (d_L, d_H)$ *and sufficiently large* n *is upper bounded by*

$$P_{e,S} \leq M^2 a_1 e^{-nb_1} + M a_2 e^{-nb_2}.$$

Outline of the Proof. The idea of proving the upper bound on the error probability is the same as the proof of Theorem 3.2.1. The only difference is that the distance between two clusters for both SLINK only depends on a pair of sequences from the two clusters. □

The second term in the upper bound on error probability in Theorem 3.2.2 is $\frac{1}{M}$ of the general bound obtained in Theorem 3.2.1.

Remark: Let $\tilde{\mathbf{D}}$ be a binary matrix, where

$$\tilde{\mathbf{D}}_{i,j} = 1_{d\left(\mathcal{C}_i^0, \mathcal{C}_j^0\right) > d_{th}}.$$

Thus $\tilde{\mathbf{D}}$ is obtained by simply thresholding pairwise distances with d_{th}. Suppose an MST with weight $\hat{K} - 1$ is obtained by applying a comparison-based minimum spanning tree (MST) algorithm, e.g., Dijkstra's algorithm, Kruskal's algorithm and Prim's algorithm, to $\tilde{\mathbf{D}}$. The clustering result is then obtained by removing all the edges with nonzero weights in the MST. With probability $1 - P_e$, where P_e has the same upper bound as $P_{e,S}$ in Theorem 3.2.2, $\hat{K} = K$ and the clustering result is correct provided that Assumptions 1 and 2 are satisfied.

Corollary 3.2.2.1. *Suppose the KS distance and MMD are used with* $d_{th} = \frac{1}{2}\Sigma_{ks}$ *and* $d_{th} = \frac{1}{2}\Sigma_{mmd}$. *Then for sufficiently large* n, *the error probability of SLINK and CLINK upon convergence is upper bounded by*

$$P_{e,S}^{KS} \leq 4M\left(M+1\right)\exp\left(-\frac{n\Delta_{ks}^2}{8}\right),$$
$$P_{e,S}^{MMD} \leq 2M\left(M+1\right)\exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right).$$

Proof. The proof is the same as that of Corollary 3.2.1.1. □

3.3 Centroid-Based Algorithms

This section presents upper bounds on the error probability of centroid-based clustering algorithms with LWD update whose coefficients satisfy (3.3). The complete proof of the results will be provided in Appendix due to the space limit.

Proposition 3. *Suppose a centroid-based clustering algorithm is generated from* (3.1) *and satisfies* (3.3) *and data sequences are generated from distributions satisfying* (2.3)*. If the distance metric used by the algorithm satisfies* (2.4a) *and* (2.4c)*, then for* $d_{th} \in (d_L, d_H)$ *and sufficiently large* n*,*

$$P\left(d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_2}^t\right) \geq d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_3}^t\right)\right) \leq 2^t a_3 e^{-nb_3}, \tag{3.10a}$$

$$P\left(d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_2}^t\right) > d_{th}\right) \leq 3^t a_2 e^{-nb_2}, \tag{3.10b}$$

where $\mathcal{C}_{l_1}^t, \mathcal{C}_{l_2}^t \sim \mathcal{P}_k$ *and* $\mathcal{C}_{l_3}^t \sim \mathcal{P}_{k'}$ *for* $k \neq k'$*.*

Outline of the Proof. By induction. □

Therefore, under Assumption 2, for sequence clusters obtained after the t-th iteration by any centroid-based algorithm, any cluster pair generated from the same distribution cluster is close to each other whereas any cluster pair generated from different distribution clusters is far apart.

Theorem 3.3.1. *Suppose a centroid-based clustering algorithm is generated from* (3.1) *and satisfies* (3.3) *and data sequences are generated from distributions satisfying* (2.3)*. If the distance metric of the data sequences satisfies* (2.4a) *and* (2.4c)*, then for* $d_{th} \in (d_L, d_H)$ *and sufficiently large* n*, the error probability upon convergence is upper bounded by*

$$P_e \leq M^2 \left(2^{M+1} M a_3 e^{-nb_3} + 3^M a_2 e^{-nb_2}\right).$$

Outline of the Proof. The proof is similar to the proof of Theorem 3.2.1. □

Corollary 3.3.1.1. *Suppose the KS distance and MMD are used with* $d_{th} = \frac{1}{2}\Sigma_{ks}$ *and* $d_{th} = \frac{1}{2}\Sigma_{mmd}$*. Then for sufficiently large* n*, the error probability of centroid-based clustering algorithms upon convergence is upper bounded by*

$$P_e^{KS} \leq M^2 \left(6 \times 2^{M+1} M + 4 \times 3^M\right) \exp\left(-\frac{n\Delta_{ks}^2}{8}\right),$$
$$P_e^{MMD} \leq M^2 \left(2^{M+3} M + 2 \times 3^M\right) \exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right).$$

Proof. By Proposition 1, the upper bound on the error probability of centroid-based clustering algorithm in Theorem 3.3.1 applies to KS and MMD. Thus, the corollary is obtained by substituting a_i and b_i with values specified in Lemmas A.0.3, A.0.4, A.0.7 and A.0.8. □

Corollary 3.3.1.1 implies that any centroid-based clustering algorithm satisfying (3.3) is exponentially consistent under both the KS and MMD distance metrics with an error exponent no smaller than $\frac{\Delta_{ks}^2}{8}$ and $\frac{\Delta_{mmd}^2}{64\mathbb{G}}$, respectively.

3.4 Numerical Results

This section presents numerical results for both linkage and centroid based algorithms. The simulation setup is the same as that in Section 2.4. The Monte Carlo experiment for a given sample size continues until following two conditions are both satisfied:

1. the number of trials that provides incorrect clustering output reaches 1000,
2. the total number of trials reaches 5×10^4.

The performance of SLINK, CLINK and WPGMC is provided in the following.

3.4.1 Performance with $d_{th} = \frac{1}{2}(d_L + d_H)$

The error probabilities of SLINK, CLINK and WPGMC under the KS distance are given in Figs. 3.1 and 3.3 while the performance of these algorithms under MMD is given in Figs. 3.2 and 3.4. That $\log P_e$ is a linear function of the sample size validates exponential consistency of these algorithms. Furthermore, both SLINK and WPGMC outperform CLINK under both the KS distance and MMD in terms of the error probability. One possible reason is that the distance between two clusters estimated by

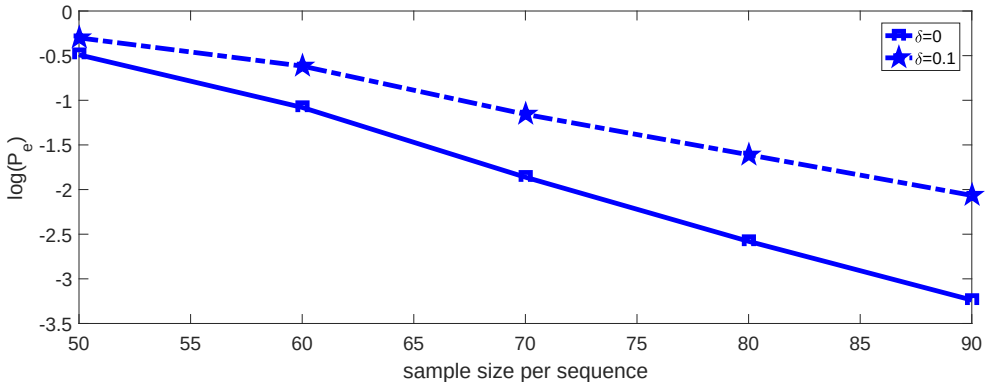
$$d_C(\mathcal{C}_l, \mathcal{C}_{l'}) = \max_{\mathbf{y}_1 \in \mathcal{C}_l, \mathbf{y}_2 \in \mathcal{C}_{l'}} d(\mathbf{y}_1, \mathbf{y}_2).$$

tends to underestimate the number of clusters. Thus, a larger d_{th} may help to improve the performance of CLINK. Moreover, the slope of $\log P_e$ with respect to n, i.e., the quantity $-\frac{\log P_e}{n}$, is non-decreasing as δ becomes smaller. In the current simulation setting, this implies a larger Δ under both the KS distance and MMD.

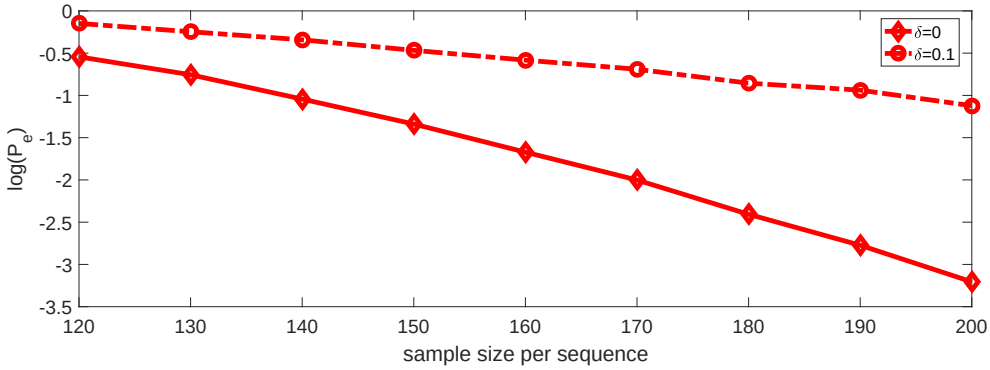
However, with Gamma distributions, $\log P_e$ with $\delta = 0$ can be larger than $\log P_e$ with $\delta = 0.1$. Possible reasons are 1) the KS distance between two sequences is always lower bounded by $\frac{1}{n}$, and 2) the MMD estimator in (2.7) has a positive bias, which has a larger impact on the clustering result when all sequences in the same cluster are generated from a single distribution.

3.4.2 Performance Given $d_{th} > \frac{1}{2}(d_L + d_H)$

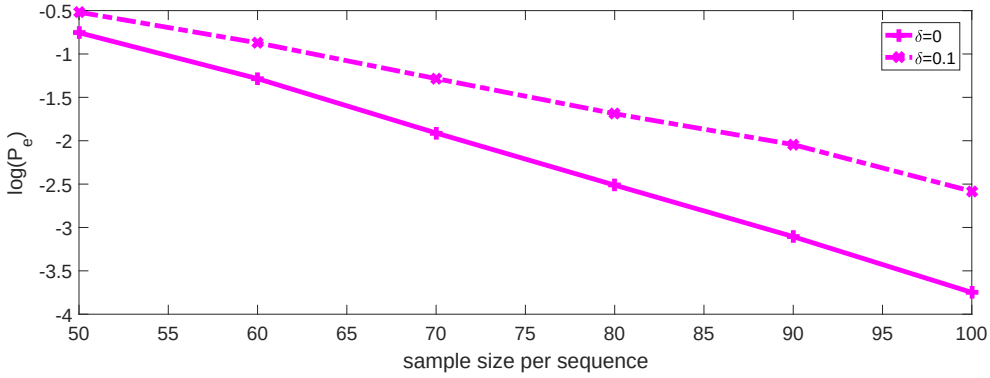
Note that Theorems 3.2.1 and 3.3.1 guarantee the exponential consistency for all $d_{th} \in (d_L, d_H)$. Let $d_{th} = \alpha d_L + (1-\alpha) d_H$, where $\alpha \in (0, 1)$. The performance of the three algorithms given $d_{th} > \frac{1}{2}(d_L + d_H)$ is provided in Figs. 3.5 and 3.6. Compare the error probability in Figs. 3.5 and 3.6 with that in Figs. 3.3 and 3.4, one can see that 1) the performance of CLINK can be significantly improved by increasing d_{th}, 2) a good choice of d_{th} for SLINK and CLINK depends on both the underlying distribution and the distance



(a) SLINK



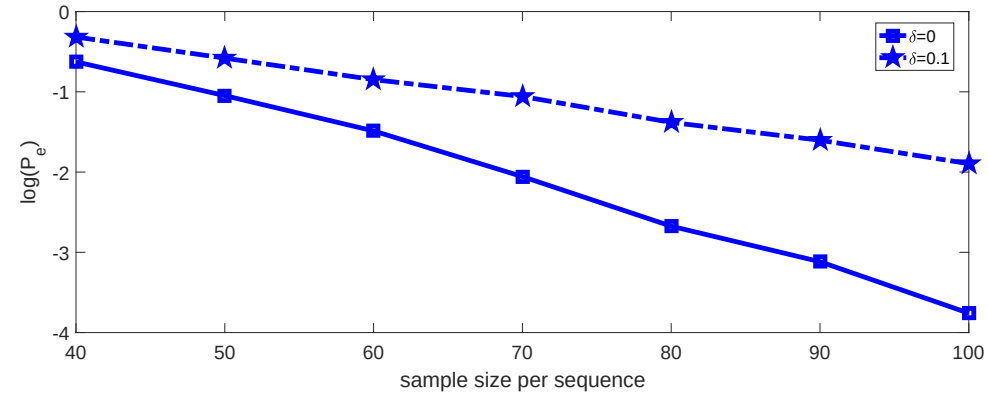
(b) CLINK



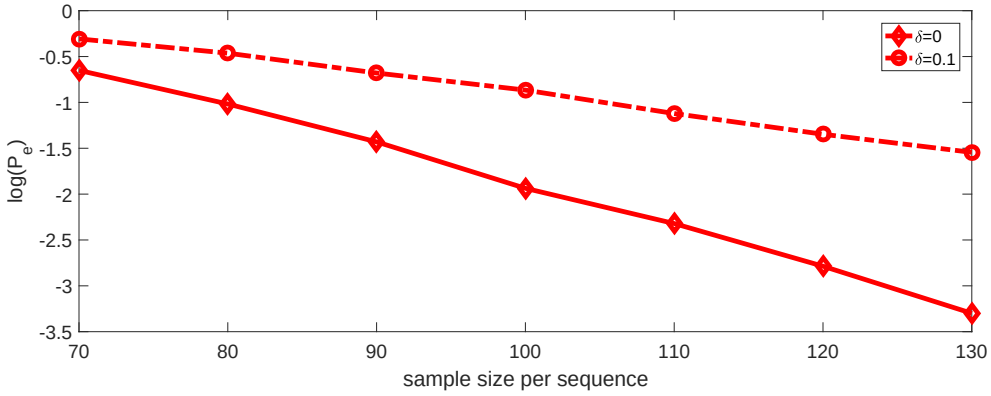
(c) WPGMC

Figure 3.1: Performance of HAC algorithms given Gaussian distributions under the KS distance

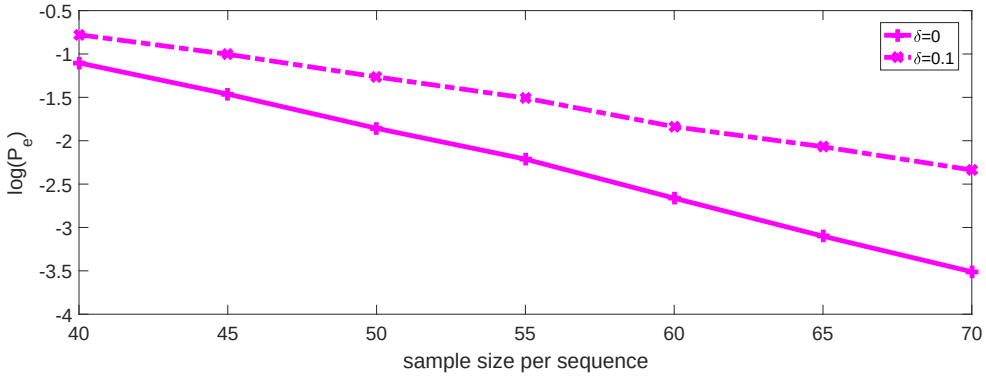
metric.



(a) SLINK



(b) CLINK

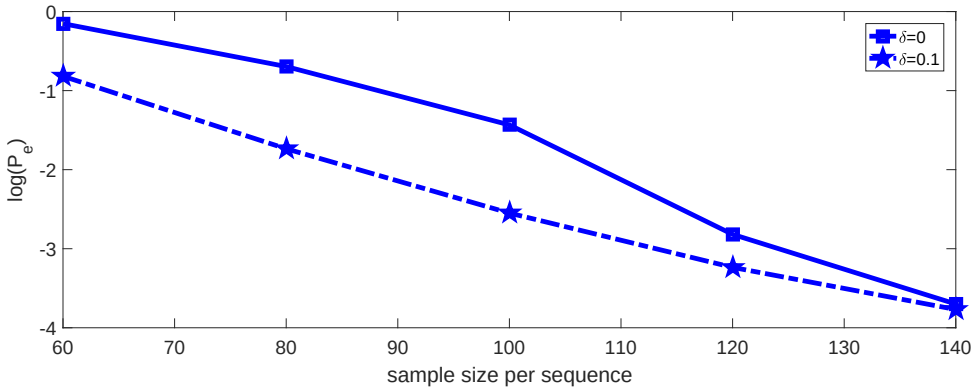


(c) WPGMC

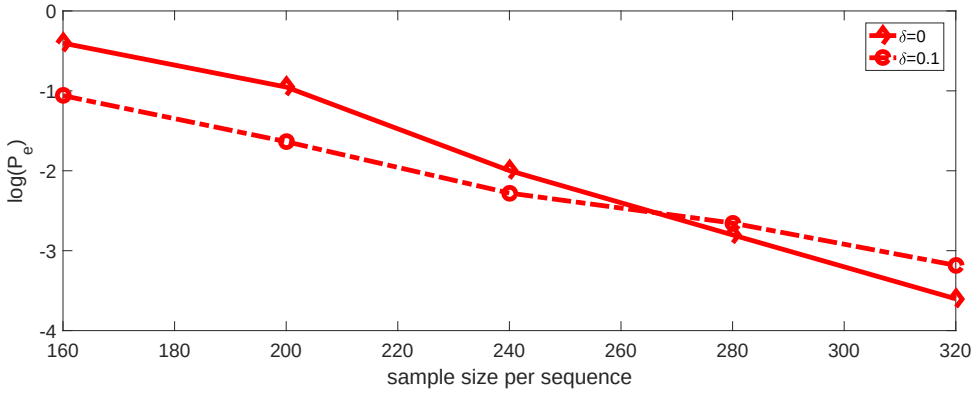
Figure 3.2: Performance of HAC algorithms given Gaussian distributions under MMD

3.4.3 Modulation Clustering for Wireless Communications

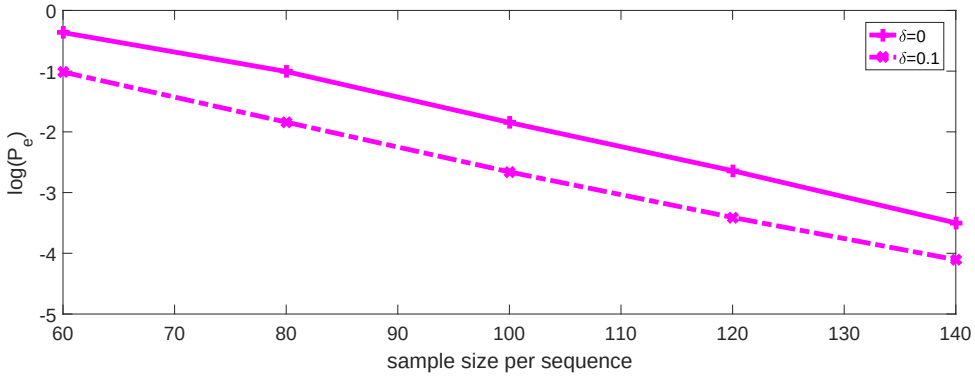
In this subsection, SLINK and UPGMA under the KS distance are applied to an on-line data set of wireless communication signals with different modulations which is introduced



(a) SLINK



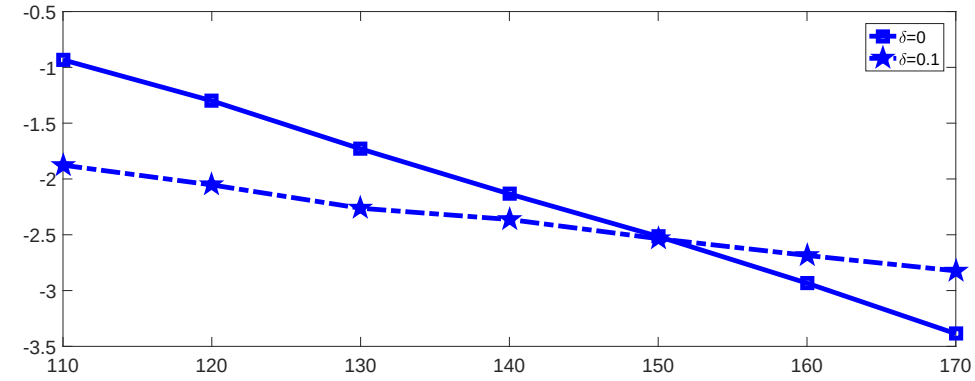
(b) CLINK



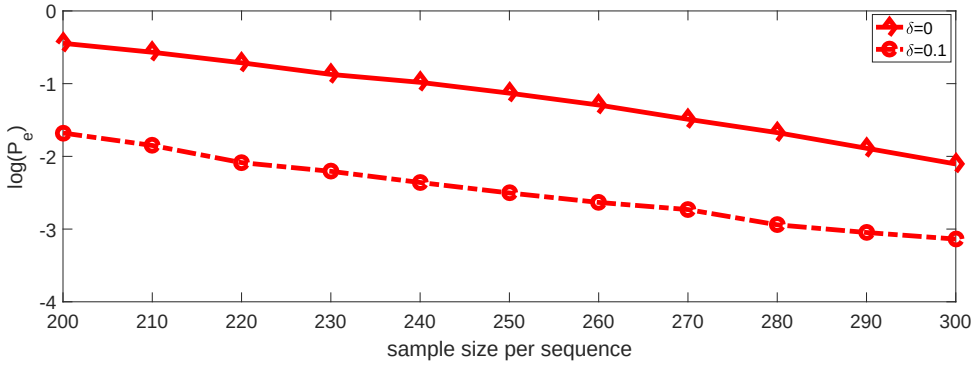
(c) WPGMC

Figure 3.3: Performance of HAC algorithms given Gamma distributions under the KS distance

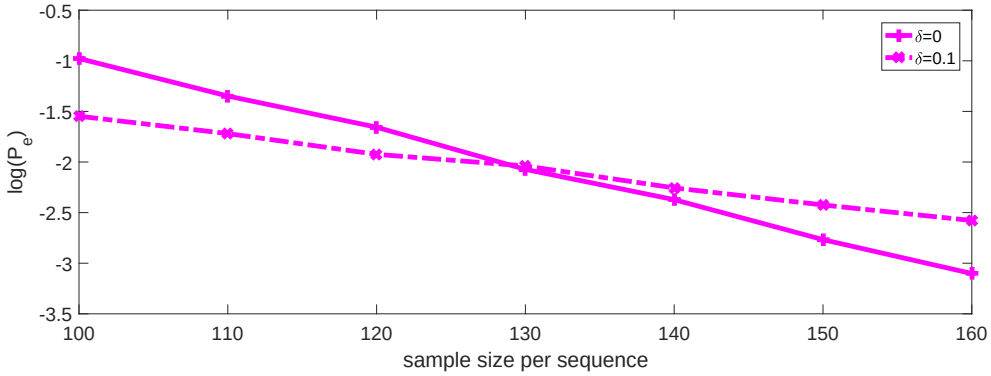
in Chapter 2.4.4. The performance of SLINK and UPGMA under the KS distance is given in Fig. 3.7. It is clear that $\log(P_e)$ is approximately a linear function when sample size n becomes large.



(a) SLINK



(b) CLINK

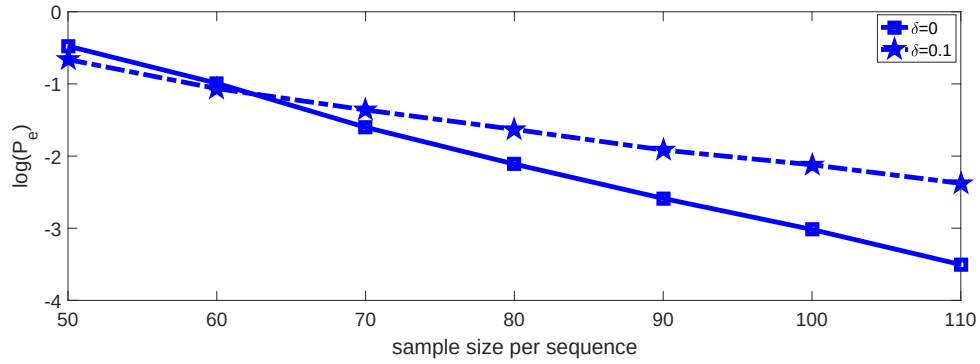


(c) WPGMC

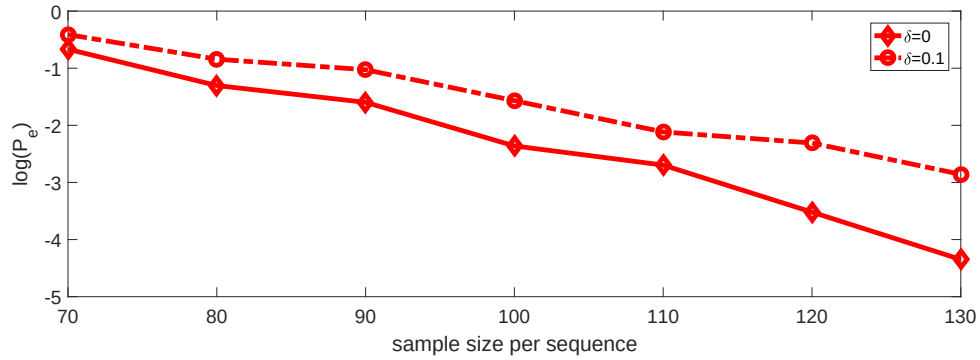
Figure 3.4: Performance of HAC algorithms given Gamma distributions under MMD

3.5 Summary

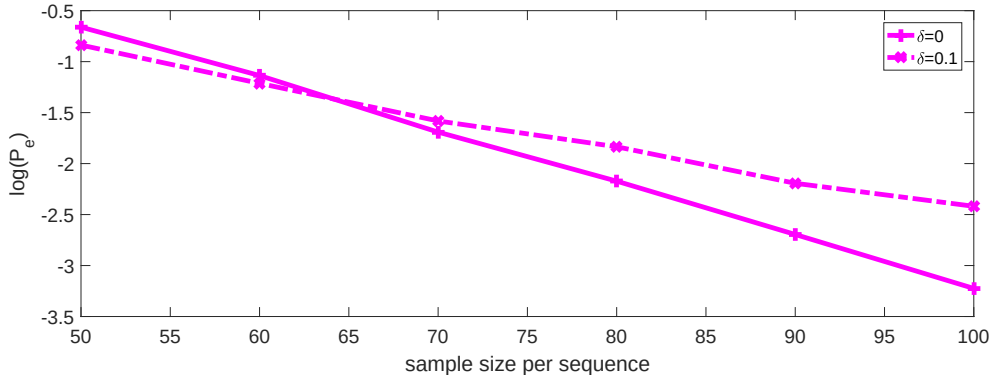
This chapter studied asymptotic performance of HAC algorithms for clustering samples generated from distribution clusters. Error probability upper bounds were derived that help establish the exponential consistency of HAC algorithms under certain conditions on the



(a) SLINK, $\alpha = 0.4$



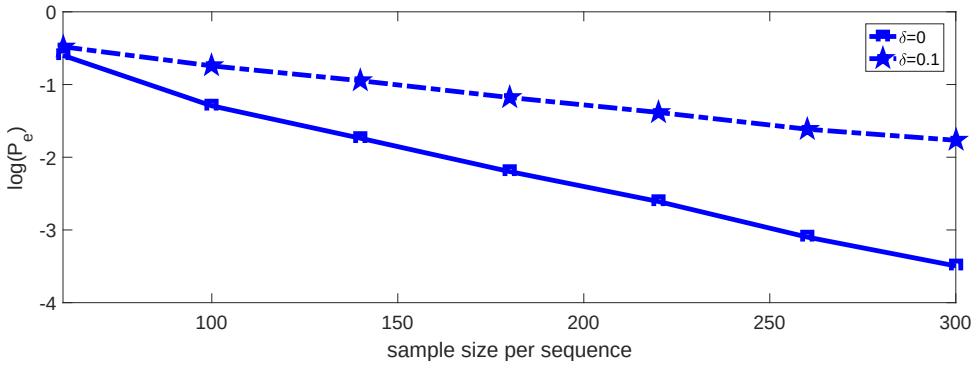
(b) CLINK, $\alpha = 0.2$



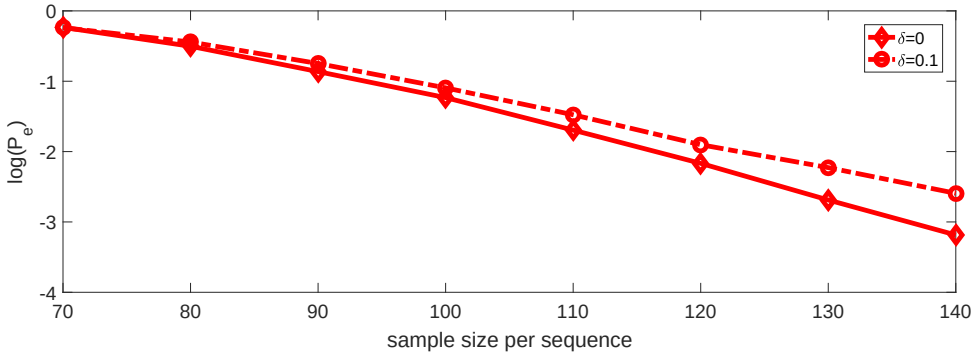
(c) WPGMC, $\alpha = 0.4$

Figure 3.5: Performance of HAC algorithms given Gamma distributions under the KS distance with different α's

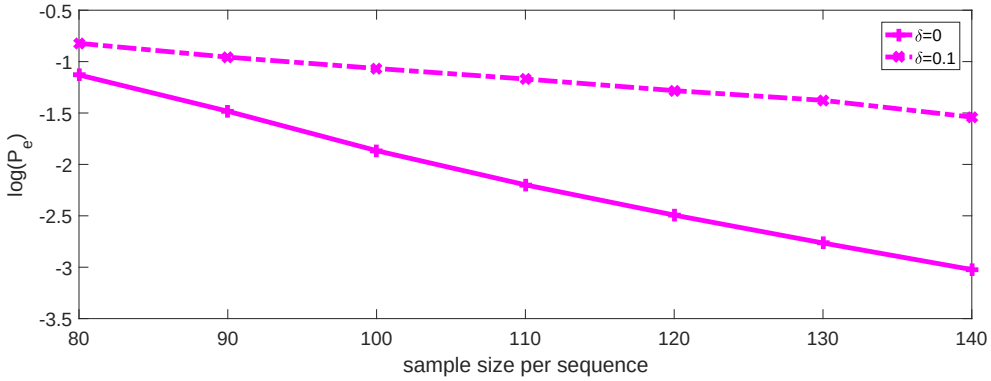
distance metrics and the underlying distribution clusters. In particular, both linkage-based and centroid-based clustering algorithms under the KS distance and MMD were shown to be exponentially consistent and lower bounds on the error exponent were characterized. While the number of sequences M is assumed to be fixed in the analysis, it is straightfor-



(a) SLINK, $\alpha = 0.3$



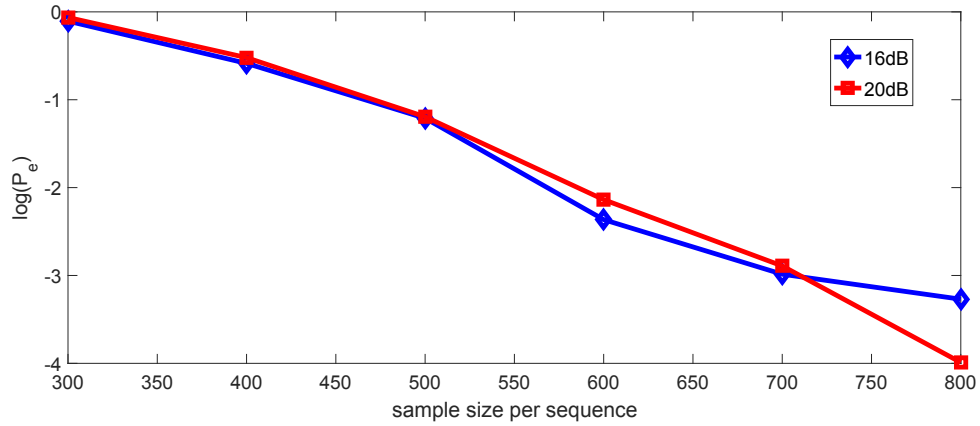
(b) CLINK, $\alpha = 0.2$



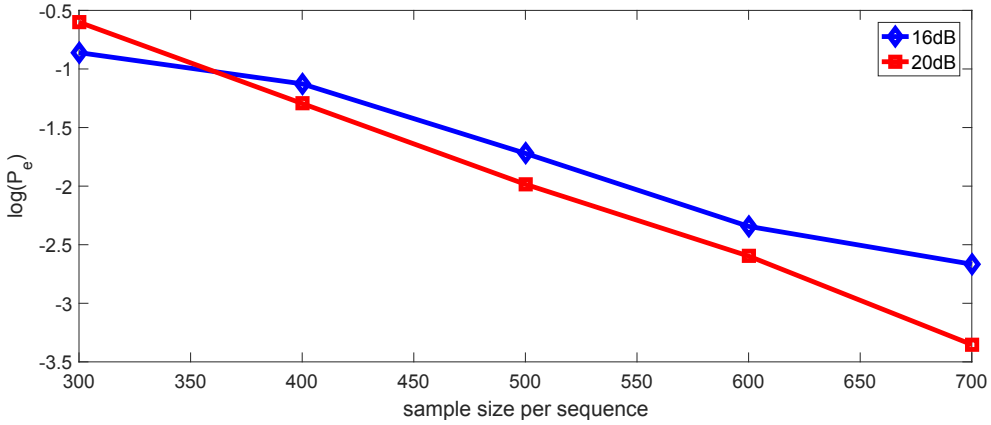
(c) WPGMC, $\alpha = 0.4$

Figure 3.6: Performance of HAC algorithms given Gamma distributions under MMD with different α's

ward to verify that exponential consistency remains valid if M grows sub-exponentially with the sample size n.



(a) SLINK, $d_{th} = 0.05$



(b) UPGMA, $d_{th} = 0.13$

Figure 3.7: Performance of Modulation clustering by HAC algorithms

CHAPTER 4
MAXIMUM DISCRIMINATING ENERGY

This chapter focuses on supervised dimensionality reduction approaches that are equivalent to (generalized) eigen-decomposition for classification problems. A new supervised dimensionarity reduction method is proposed that maximizes the difference of the average energy difference between data with different labels in the subspace. Comparison of the proposed approach with existing dimensionality reduction methods is provided to understand the relative merits among competing approaches.

4.1 Dimensionality Reduction

A common dilemma for many learning problems is the scarcity of data. The problem is particularly acute when samples are of high dimension. An example is gene sequence data whose length is often in the thousands, which may far outnumber the number of samples (i.e., the number of human subjects). Another example is in WiFi sensing where the channel state information measured in temporal, frequency, and spatial dimensions may result in a high dimensional vector when flattened.

Dimensionality reduction is an effective way to alleviate this problem. For example, the principal components are often obtained as a sequence of projection, determined by the

dataset itself, with decreasing variances. Low dimension representation can thus be obtained through simple truncation, i.e., retaining only the dominant principal components. Principal component analysis (PCA) has long been used as a de facto way for dimensionality reduction for unsupervised learning problems.

For supervised learning such as classification or regression problems, the label information or response variables were often disregarded when dimensionality reduction is carried out. This can become problematic since high variance principal components do not necessarily lead to good discriminating property for classification problems or may not have strong correlation with the response variables for regression problems.

Focusing on classification problems, we examine supervised dimensionality reduction by exploiting the label information associated with each data sample. Our goal is to preserve maximum discriminating information in the reduced dimension representation for the classification problem. We first review existing dimensionality reduction methods including the classical PCA as well as some recently proposed SDR methods.

4.2 Unsupervised PCA

Consider M data samples $\mathbf{x}_m \in \mathbb{R}^{N\times 1}$ for $m = 1, \ldots, M$. The centered data sample $\bar{\mathbf{x}}_m = \mathbf{x}_m - \hat{\boldsymbol{\mu}}$, where $\hat{\boldsymbol{\mu}} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i$. Conventionally, PCA is used as a dimensionality reduction method for *unsupervised* problems which tries to construct orthonormal basis $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K] \in \mathbf{R}^{N\times K}$, where $K < N$, such that the total variation of $\mathbf{U}^T\bar{\mathbf{x}}_m$'s is maximized. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ and $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_M]$. $\mathbf{u}_k$ is the eigenvector corresponding to the k-th largest eigenvalue of $\bar{\mathbf{X}}\bar{\mathbf{X}}^T$.

Some problems arise when PCA is used for *supervised* learning problems, e.g., classification. Since PCA tries to find a subspace that preserves the maximum variation of the centered samples regardless of the label of $\mathbf{x}_m$, the obtained subspace may ignore label information. For instance, consider a binary classification problem. Denote by $\mathbf{x}_i^{(l)}$ the i-th

sample with label l for $l = 1, 2$. Assume that $\mathbf{x}_i^{(1)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}\right)$ and $\mathbf{x}_i^{(2)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}\right)$, where $\boldsymbol{\mu}^{(1)} = [0.1, 0]$, $\boldsymbol{\mu}^{(2)} = [-0.1, 0]$, and

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.1 & 0 \\ 0 & 10 \end{bmatrix}.$$

It is easy to verify that the 1-D subspace $\mathbf{u}_1 = [1, 0]$ is sufficient for classification. However, PCA will find the 1-D subspace $\mathbf{u}_1 = [0, 1]$ since this direction contains larger variation. Furthermore, when samples with the same label form multiple clusters in the sample space, we may want to group samples according to labels in the subspace. Without utilizing the label information, it is unlikely that PCA is capable of grouping samples with the same label.

4.3 Existing Supervised PCA Methods

Before discussing the existing supervised PCA methods, we first introduce some notations to be used in the following discussion. Denote by $\mathbf{X}^{(l)} = [\mathbf{x}_1^{(l)}, \ldots, \mathbf{x}_{M_l}^{(l)}] \in \mathbb{R}^{N \times M_l}$ the matrix containing all the samples with label l for $l \in \{1, \ldots, L\}$, where $\sum_{l=1}^{L} M_l = M$. Let $\bar{\mathbf{X}}^{(l)} = [\bar{\mathbf{x}}_1^{(l)}, \ldots, \bar{\mathbf{x}}_{M_l}^{(l)}]$ be the centered samples with label l, where $\bar{\mathbf{x}}_i^{(l)} = \mathbf{x}_i^{(l)} - \hat{\boldsymbol{\mu}}$. Denote by $\mathbf{X} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(L)}]$ and $\bar{\mathbf{X}} = [\bar{\mathbf{X}}^{(1)}, \ldots, \bar{\mathbf{X}}^{(L)}]$ the matrices containing original samples and centered samples for all labels, respectively. In the following discussion, we may use either $\mathbf{x}_i^{(l)}$ or $\mathbf{x}_m$ to denote a sample depending on whether the label is of interest or not. Finally, let K be the dimension of the subspace obtained by an SDR approach.

4.3.1 FDA

Define $\hat{\boldsymbol{\mu}}^{(l)} = \frac{1}{M_l}\sum_{i=1}^{M_l}\mathbf{x}_i^{(l)}$ for $l = 1, \ldots, L$. The subspace obtained by FDA is actual the solution to the following problem [42] :

$$\begin{aligned} &\max_{\mathbf{U}} \frac{\left|\mathbf{U}^T\mathbf{S}_B\mathbf{U}\right|}{\left|\mathbf{U}^T\mathbf{S}_W\mathbf{U}\right|}, \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} \mathbf{S}_B &= \sum_{l=1}^{L} M_l \left(\hat{\boldsymbol{\mu}}^{(l)} - \hat{\boldsymbol{\mu}}\right)\left(\hat{\boldsymbol{\mu}}^{(l)} - \hat{\boldsymbol{\mu}}\right)^T, \\ \mathbf{S}_w &= \sum_{l=1}^{L}\sum_{i=1}^{M_l}\left(\mathbf{x}_i^{(l)} - \hat{\boldsymbol{\mu}}^{(l)}\right)\left(\mathbf{x}_i^{(l)} - \hat{\boldsymbol{\mu}}^{(l)}\right)^T. \end{aligned}$$

(4.1) is equivalent to the following generalized eigenvalue problem:

$$\mathbf{S}_B\mathbf{u} = \lambda\mathbf{S}_W\mathbf{u}. \tag{4.2}$$

There are two obvious drawbacks of FDA. First, FDA requires that samples with different labels should have different means. Otherwise, $\mathbf{S}_B \approx \mathbf{0}$. Second, since rank $(\mathbf{S}_B) \leq L-1$, the dimension of the subspace obtained by FDA is at most $(L-1)$, which may be too small to contain all the useful information for classification, especially when L is small.

Let Φ be a map from the sample space $\mathcal{X}$ to some feature space $\mathcal{H}$ and $k(\cdot,\cdot)$ be the kernel function associated with Φ. Denote by $\mathbf{K} \in \mathbb{R}^{M\times M}$ the matrix containing kernel function result for each pair of training samples where the (i,j)-th entry of $\mathbf{K}$ is $k(\mathbf{x}_i, \mathbf{x}_j)$. Let $\bar{\mathbf{K}}^{(l)}$ and $\bar{\mathbf{K}}$ be column vectors of length M and the m-th entry of them are $\frac{1}{M_l}\sum_{i=1}^{M_l} k(\mathbf{x}_m, \mathbf{x}_i^{(l)})$ and $\frac{1}{L}\sum_{l=1}^{L}\frac{1}{M_l}\sum_{i=1}^{M_l} k(\mathbf{x}_m, \mathbf{x}_i^{(l)})$, respectively. Furthermore, let $\mathbf{K}^{(l)} \in \mathbb{R}^{M\times M_l}$ be a matrix where its (m,i)-th entry is $k(\mathbf{x}_m, \mathbf{x}_i^{(l)})$. Then kernel FDA can

be expressed as

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \frac{\left|\boldsymbol{\beta}^T \sum_{l=1}^{L} M_l \left(\bar{\mathbf{K}}^{(l)} - \bar{\mathbf{K}}\right) \left(\bar{\mathbf{K}}^{(l)} - \bar{\mathbf{K}}\right)^T \boldsymbol{\beta}\right|}{\left|\boldsymbol{\beta}^T \sum_{l=1}^{L} \left(\mathbf{K}^{(l)}(\mathbf{I} - \frac{1}{M_l}\mathbf{1}\mathbf{1}^T)(\mathbf{K}^{(l)})^T\right) \boldsymbol{\beta}\right|}, \\ &\text{s.t. } \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} = \mathbf{I}, \end{aligned} \tag{4.3}$$

which is also a generalized eigenvalue problem.

4.3.2 HSIC

In [45], the author proposed HSIC based supervised PCA, which solves the following problem:

$$\begin{aligned} &\max_{\mathbf{U}} \text{ tr}\left(\mathbf{U}^T \bar{\mathbf{X}} \mathbf{L} \bar{\mathbf{X}}^T \mathbf{U}\right) \\ &\text{s.t. } \mathbf{U}^T \mathbf{U} = \mathbf{I}, \end{aligned} \tag{4.4}$$

where $\mathbf{L}$ is a kernel of outcome measurement matrix $\mathbf{Y} \in \mathbb{R}^{L \times M}$. e.g., $\mathbf{L} = \mathbf{Y}^T\mathbf{Y}$. One possible choice of $\mathbf{Y}$ is $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M]$, where $\mathbf{y}_m$ is a one-hot column vector depending on the label of $\mathbf{x}_m$. Given $\mathbf{L} = \mathbf{Y}^T\mathbf{Y}$, where $\mathbf{Y}$ consists of one-hot vectors, the m-th column of $\bar{\mathbf{X}}\mathbf{Y}^T \in \mathbb{R}^{N \times M}$ is $\sum_{i=1}^{M_l} \bar{\mathbf{x}}_i^{(l)}$ for some $l \in \{1, \ldots, L\}$. Therefore, (4.4) is equivalent to

$$\begin{aligned} &\max_{\mathbf{U}} \sum_{l=1}^{L} M_l \left|\left|\mathbf{U}^T \sum_{i=1}^{M_l} \bar{\mathbf{x}}_i^{(l)}\right|\right|_2^2, \\ &\text{s.t. } \mathbf{U}^T \mathbf{U} = \mathbf{I}, \end{aligned}$$

which is equivalent to PCA with L samples and the l-th sample is $\sqrt{M_l} \sum_{i=1}^{M_l} \bar{\mathbf{x}}_i^{(l)}$ for $l = 1, \ldots, L$.

The drawback of HSIC-based approach is that 1) (4.4) becomes unreliable when $\sum_{i=1}^{M_l} \bar{\mathbf{x}}_i^{(l)} \approx \mathbf{0}$ for all $l = 1, \ldots, L$, 2) unbalanced data with different labels may lead to performance degradation, and 3) the maximum number of non-zero eigenvalues obtained from (4.4) is on greater than rank$(\mathbf{L})$. Hence, the subspace obtained by HSIC-based SPCA may be too small to contain all the useful information for classification when L is small.

Kernel trick can also be applied to HSIC which is to solve the following problem:

$$\begin{aligned} &\max \text{ tr}\left(\boldsymbol{\beta}^T\mathbf{KHLHK}\boldsymbol{\beta}\right) \\ &\text{s.t. } \boldsymbol{\beta}^T\mathbf{K}\boldsymbol{\beta} = \mathbf{I}, \end{aligned} \tag{4.5}$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{M}\mathbf{1}\mathbf{1}^T$. (4.5) is also equivalent to a generalized eigenvalue problem. However, kernel trick does not increase the upper bound on the dimension of the subspace as long as $\mathbf{L} = \mathbf{Y}^T\mathbf{Y}$.

4.4 The Proposed Approach

The main idea of the proposed SDR approach named maximum discriminating energy (MDE) is to find directions that maximizes the difference of the average projection energy between different labels.

4.4.1 MDE for binary case

For binary classification problems, the subspace $\mathbf{U} \in \mathbb{R}^{N\times K}$ preserving maximum average energy difference between samples with different labels can be written as

$$\sum_{k=1}^{K}\left|\frac{1}{M_1}\sum_{i=1}^{M_1}\left|\left|\mathbf{u}_k^T\bar{\mathbf{x}}_i^{(1)}\right|\right|_2^2 - \frac{1}{M_2}\sum_{i=1}^{M_2}\left|\left|\mathbf{u}_k^T\bar{\mathbf{x}}_i^{(2)}\right|\right|_2^2\right|.$$

Let us first consider the case with

$$\frac{1}{M_1}\sum_{i=1}^{M_1}\left|\left|\mathbf{u}_k^T\bar{\mathbf{x}}_i^{(1)}\right|\right|_2^2 - \frac{1}{M_2}\sum_{i=1}^{M_2}\left|\left|\mathbf{u}_k^T\bar{\mathbf{x}}_i^{(2)}\right|\right|_2^2 > 0. \tag{4.6}$$

The subspace satisfying (4.6) can be found by solving the following problem:

$$\begin{aligned} &\max \frac{1}{M_1} \text{ tr}\left(< \mathbf{U}^T\bar{\mathbf{X}}^{(1)}, \mathbf{U}^T\bar{\mathbf{X}}^{(1)} >\right) - \frac{1}{M_2} \text{ tr}\left(< \mathbf{U}^T\bar{\mathbf{X}}^{(2)}, \mathbf{U}^T\bar{\mathbf{X}}^{(2)} >\right), \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.7}$$

The objective function in (4.7) can be further rewritten as

$$\begin{aligned} &\frac{1}{M_1} \text{ tr}\left(< \mathbf{U}^T\bar{\mathbf{X}}^{(1)}, \mathbf{U}^T\bar{\mathbf{X}}^{(1)} >\right) - \frac{1}{M_2} \text{ tr}\left(< \mathbf{U}^T\bar{\mathbf{X}}^{(2)}, \mathbf{U}^T\bar{\mathbf{X}}^{(2)} >\right) \\ &= \text{tr}\left(\frac{1}{M_1}\bar{\mathbf{X}}^{(1)T}\mathbf{U}\mathbf{U}^T\bar{\mathbf{X}}^{(1)} - \frac{1}{M_2}\bar{\mathbf{X}}^{(2)T}\mathbf{U}\mathbf{U}^T\bar{\mathbf{X}}^{(2)}\right) \\ &= \text{tr}\left(\frac{1}{M_1}\mathbf{U}^T\bar{\mathbf{X}}^{(1)}\mathbf{X}^{(1)T}\mathbf{U} - \frac{1}{M_2}\mathbf{U}^T\bar{\mathbf{X}}^{(2)}\bar{\mathbf{X}}^{(2)T}\mathbf{U}\right). \end{aligned}$$

Define $\mathbf{Q} = \frac{1}{M_1}\bar{\mathbf{X}}^{(1)}\bar{\mathbf{X}}^{(1)T} - \frac{1}{M_2}\bar{\mathbf{X}}^{(2)}\bar{\mathbf{X}}^{(2)T}$. Then (4.7) becomes

$$\begin{aligned} &\max \text{ tr}\left(\mathbf{U}^T\mathbf{Q}\mathbf{U}\right) \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.8}$$

In general, $\mathbf{Q}$ is neither positive nor negative semi-definite.

Now consider the other case with

$$\frac{1}{M_1}\sum_{i=1}^{M_1}\left|\left|\mathbf{u}_k^T\bar{\mathbf{x}}_i^{(1)}\right|\right|_2^2 - \frac{1}{M_2}\sum_{i=1}^{M_2}\left|\left|\mathbf{u}_k^T\bar{\mathbf{x}}_i^{(2)}\right|\right|_2^2 < 0.$$

It should be apparent that the eigenvectors corresponding to the negative eigenvalues obtained in (4.8) consists a solution to (4.4.1). Without loss of generality, assume that there are P ($P \geq K$) non-zero eigenvalues $\lambda_1, \ldots, \lambda_P$ such that $|\lambda_1| \geq \ldots \geq |\lambda_P| > 0$. Let $\mathbf{u}_p$ be the eigenvector corresponding to λ_p. Then the subspace obtained by (4.8) is $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$, where K can be either pre-defined or the smallest positive number such that $\sum_{p=1}^{K}|\lambda_p|$ is dominant in $\sum_{p=1}^{P}|\lambda_p|$.

Note that the method in (4.8) cannot distinguish samples generated from the following

Algorithm 7 MDE algorithm for Binary Classification

1: **Input**: Training data sequences and labels $(\mathbf{X}, \mathbf{y})$, test data $\mathbf{x}$, and hyperparameter τ.
2: **Output**: Dimension reduced training data $\mathbf{Z}$ and test data $\mathbf{z}$.
3: Compute $\bar{\mathbf{X}} = \mathbf{X} - \hat{\boldsymbol{\mu}}$, where $\hat{\boldsymbol{\mu}} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i$.
4: Split $\bar{\mathbf{X}}$ into $\bar{\mathbf{X}}^{(1)}$ and $\bar{\mathbf{X}}^{(2)}$ by labels and compute $\hat{\boldsymbol{\mu}}^{(1)} = \frac{1}{M_1}\sum_{i=1}^{M_1} \mathbf{x}_i^{(1)}$ and $\hat{\boldsymbol{\mu}}^{(2)} = \frac{1}{M_2}\sum_{i=1}^{M_2} \mathbf{x}_i^{(2)}$.
5: Compute the eigen decomposition of $\mathbf{Q} + \tau\boldsymbol{\delta}_\mu\boldsymbol{\delta}_\mu^T$ and find K eigenvalues with the largest absolute value $\lambda_1, \ldots, \lambda_K$ and the corresponding eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_K$.
6: $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$.
7: $\mathbf{Z} = \mathbf{U}^T\mathbf{X}$ and $\mathbf{z} = \mathbf{U}^T\mathbf{x}$.

case. Suppose $\mathbf{x}_i^{(1)} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ and $\mathbf{x}_i^{(2)} \sim \mathcal{N}(-\boldsymbol{\mu}, \mathbf{C})$. The correlation matrices are, for $l = 1, 2$,

$$\mathbf{R}_l = \mathbb{E}_{P_l}\left[\mathbf{x}\mathbf{x}^T\right] = \boldsymbol{\mu}\boldsymbol{\mu}^T + \mathbf{C}.$$

Assume that the sample correlation matrix is a good approximation for the true correlation matrix, i.e., $\frac{1}{M_l}\mathbf{X}^{(l)}\mathbf{X}^{(l)^T} \approx \mathbf{R}_l$. Then $\mathbf{Q} \approx \mathbf{0}$. A simple remedy is to add a penalty term relating to the mean difference to (4.8), resulting in the following problem:

$$\begin{aligned} &\max \ \mathrm{tr}\left(\mathbf{U}^T\left(\mathbf{Q} + \tau\boldsymbol{\delta}_\mu\boldsymbol{\delta}_\mu^T\right)\mathbf{U}\right) \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}, \end{aligned}$$

where $\boldsymbol{\delta}_\mu = \hat{\boldsymbol{\mu}}^{(1)} - \hat{\boldsymbol{\mu}}^{(2)}$ and $\tau \in \mathbb{R}^+$ is a tunable hyperparamter. MDE for binary case is summarized as Algorithm 7.

4.4.2 MDE for Multi-Class Case

Algorithm 7 can be directly extended to multi-class classification by viewing an L-class classification as L binary classification problems. Define $\mathbf{U}^{(l)}$ as the subspace obtained by Algorithm 7 for $\mathbf{X}^{(l)}$ and $\mathbf{X} \setminus \mathbf{X}^{(l)}$. Then the subspace for the multi-class classification is given by

$$\mathbf{U}_{\text{agg}} = \left[\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(L)}\right]. \tag{4.9}$$

Algorithm 8 MDE algorithm for Multi-Class Classification

Input: Training data sequences and labels $(\mathbf{X}, \mathbf{y})$, test data $\mathbf{x}$, and hyperparameter λ.
Output: Dimension reduced training data $\mathbf{Z}$ and test data $\mathbf{z}$.
Compute $\bar{\mathbf{X}} = \mathbf{X} - \hat{\boldsymbol{\mu}}$, where $\hat{\boldsymbol{\mu}} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i$.
$\mathbf{U}_{\text{agg}} = \emptyset$.
for $l = 1$ to L **do**
    Apply Algorithm 7 to $\bar{\mathbf{X}}^{(l)}$ and $\mathbf{X} \setminus \bar{\mathbf{X}}^{(l)}$.
    $\mathbf{U}_{\text{agg}} \leftarrow \left[\mathbf{U}_{\text{agg}},\ \mathbf{U}^{(l)}\right]$.
end for
Compute the eigen decomposition of $\mathbf{U}_{\text{agg}}\mathbf{U}_{\text{agg}}^T$ and find K largest positive eigenvalues $\lambda_1, \ldots, \lambda_K$ and the corresponding eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_K$.
$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K]$.
$\mathbf{Z} = \mathbf{U}^T\mathbf{X}$ and $\mathbf{z} = \mathbf{U}^T\mathbf{x}$.

In practice, $\mathbf{U}^{\text{agg}}$ in (4.9) may not be mutually orthogonal. By applying PCA to $\mathbf{U}_{\text{agg}}\mathbf{U}_{\text{agg}}^T$, the subspace obtained by MDE consists of the K eigenvectors corresponding to the K largest eigenvalues of $\mathbf{U}_{\text{agg}}\mathbf{U}_{\text{agg}}^T$. The proposed algorithm for multi-class classification is summarized as Algorithm 8.

4.4.3 Kernel MDE

Kernel trick can be applied to MDE. Let

$$\mathbb{I}_{M_1,M_2} = \begin{bmatrix} \frac{1}{M_1}\mathbf{I} & \mathbf{0}_{M_1\times M_2} \\ \mathbf{0}_{M_2\times M_1} & -\frac{1}{M_2}\mathbf{I} \end{bmatrix}.$$

Then $\mathbf{Q}$ in (4.8) can be rewritten as

$$\mathbf{Q} = [\bar{\mathbf{X}}^{(1)}\ \bar{\mathbf{X}}^{(2)}]\mathbb{I}_{M_1,M_2}[\bar{\mathbf{X}}^{(1)}\ \bar{\mathbf{X}}^{(2)}]^T.$$

Define

$$\Phi(\mathbf{Q}) = [\Phi(\mathbf{X}^{(1)})\ \Phi(\mathbf{X}^{(2)})]\mathbf{H}\mathbb{I}_{M_1,M_2}\mathbf{H}[\Phi(\mathbf{X}^{(1)})\ \Phi(\mathbf{X}^{(2)})]^T,$$

where $\Phi(\mathbf{X}^{(l)}) = [\Phi(\mathbf{x}_1^{(l)}), \ldots, \Phi(\mathbf{x}_{M_l}^{(l)})]$. Note that the features of the original samples rather than centered samples are computed in $\Phi(\mathbf{Q})$. Then applying kernel trick to (4.8),

we have

$$\begin{aligned} &\max \text{ tr}\left(\mathbf{U}^T\Phi(\mathbf{Q})\mathbf{U}\right) \\ &\text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.10}$$

By representation theorem [68], $\mathbf{U} = [\Phi(\mathbf{X}^{(1)})\ \Phi(\mathbf{X}^{(2)})]\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^{M\times K}$. Thus, (4.10) can be rewritten as

$$\begin{aligned} &\max_{\boldsymbol{\beta}} \text{ tr}\left(\boldsymbol{\beta}^T\mathbf{K}\mathbf{H}\mathbb{I}_{M_1,M_2}\mathbf{H}\mathbf{K}\boldsymbol{\beta}\right) \\ &\text{s.t. } \boldsymbol{\beta}^T\mathbf{K}\boldsymbol{\beta} = \mathbf{I}, \end{aligned} \tag{4.11}$$

where

$$\mathbf{K} = \begin{bmatrix} \Phi\left(\mathbf{X}^{(1)}\right)^T\Phi\left(\mathbf{X}^{(1)}\right) & \Phi\left(\mathbf{X}^{(1)}\right)^T\Phi\left(\mathbf{X}^{(2)}\right) \\ \Phi\left(\mathbf{X}^{(2)}\right)^T\Phi\left(\mathbf{X}^{(1)}\right) & \Phi\left(\mathbf{X}^{(2)}\right)^T\Phi\left(\mathbf{X}^{(2)}\right) \end{bmatrix}.$$

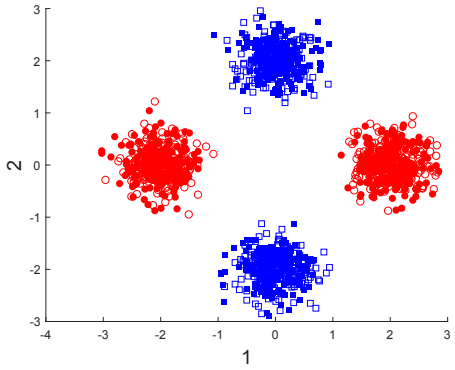
(4.11) is also equivalent to a generalized eigen-value problem. When $L > 2$, PCA is applied to $[\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(L)}]$, where $\boldsymbol{\beta}^{(l)}$ is obtained by from (4.11) given $\mathbf{X}^{(l)}$ and $\mathbf{X} \setminus \mathbf{X}^{(l)}$.

4.5 Performance Comparison

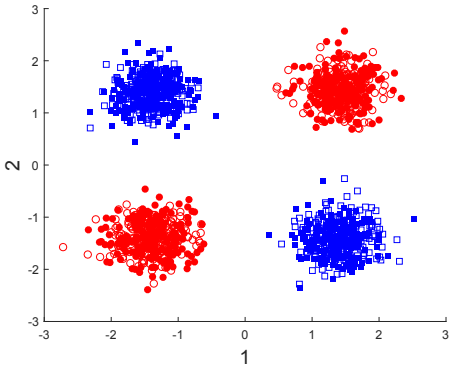
In this section, we will first provide some projection results given synthetic data. WiFi sensing data will then be used to evaluate the performance of MDE, HSIC, FDA and PCA in classification.

4.5.1 Visualization by Synthetic Data

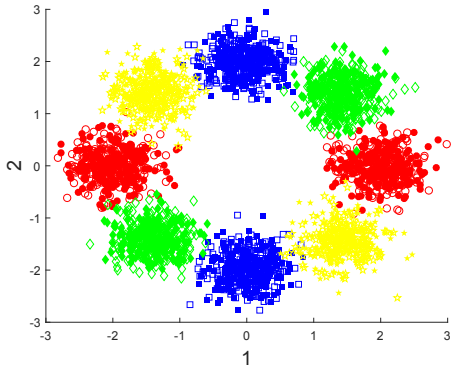
Let $\mathbf{s} \in \mathbb{R}^d$ for $d = 2, 3$ be the ground truth of the data, which is generated from some Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma}$ is identical for samples with different labels. The observation $\mathbf{x} \in \mathbb{R}^5$ is obtained by $\mathbf{x} = \mathbf{V}\mathbf{s} + \mathbf{w}$, where $\mathbf{V} \in \mathbb{R}^{5\times d}$ consists of orthonormal column vectors and $\mathbf{w} \sim \mathcal{N}(0, 0.1\mathbf{I})$ is additive noise. The ground truth of $\mathbf{s}$ in 6 cases is shown in Fig. 4.1, where data with different labels are denoted by markers with different colors. Training and test data with the same label are denoted by filled and unfilled markers,



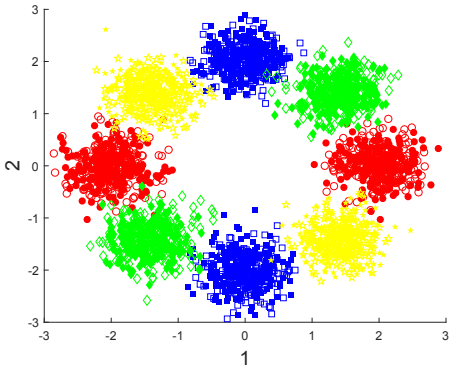
(a) Case 1: $\mathbf{\Sigma} = \text{diag}\,(0.1, 0.1)$



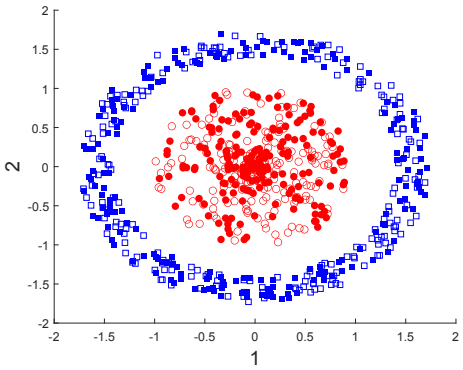
(b) Case 2: $\mathbf{\Sigma} = \text{diag}\,(0.1, 0.1, 5)$



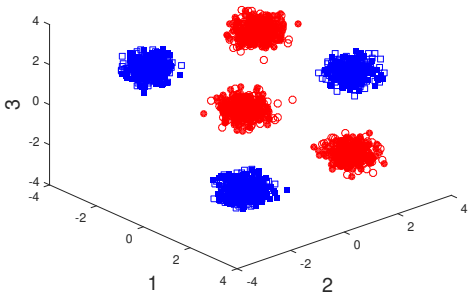
(c) Case 3: $\mathbf{\Sigma} = \text{diag}\,(0.1, 0.1)$



(d) Case 4: $\mathbf{\Sigma} = \text{diag}\,(0.1, 0.1, 5)$



(e) Case 5: Concentric Rings



(f) Case 6: $\mathbf{\Sigma} = \text{diag}(0.1, 0.1, 0.1)$

Figure 4.1: Original data s with lower dimension

respectively. In Fig. 4.1a, c and e, $\mathbf{s} \in \mathbb{R}^2$ whereas in Fig. 4.1b, d and f, $\mathbf{s} \in \mathbb{R}^3$. Furthermore, in Fig. 4.1b and d, $\boldsymbol{\mu} = [\mu_1, \mu_2, 0]$. i.e., The label information still concentrate in the first two dimensions. Only in Fig. 4.7, the third dimension of s contains label information.

For MDE, $\tau = 0.1$ and for kernel MDE, HSIC and FDA, radial basis function (RBF)

kernel is used. The RBF kernel of two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma ||\mathbf{x} - \mathbf{y}||_2^2),$$

where γ is a tunable hyper-parameter.

The projections corresponding to the largest two (or three) eigen values given MDE, HSIC and FDA are shown in Fig.s 4.2 - 4.7, respectively. From Fig.s 4.2 - 4.7, we have the following observations for the three SDR methods *without* kernel trick:

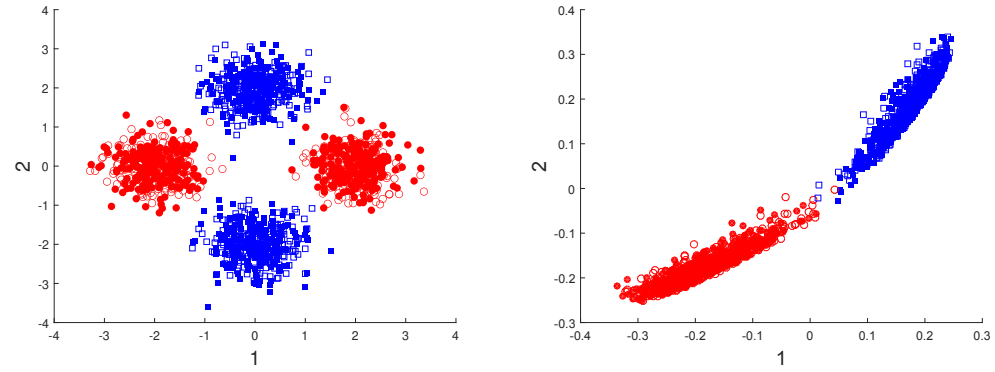
1. MDE can find a subspace that is identical to the original one subject to rotation even for data forms a concentric ring as shown in Fig.s 4.2a - 4.6a. Given label information in a 3-D space, MDE groups data according to their labels and preserve some margin between data with different labels as shown in Fig. 4.7a.

2. The projection obtained by HSIC does not always preserve the geometry of $\mathbf{s}$ given label information in 2-D space, e.g., Fig.s 4.2c and 4.3c, even though the projection obtained by HSIC preserves the concentric structure as shown in Fig. 4.6c. Furthermore, the margin of projection with different labels tends to be smaller when HSIC is used as shown in Fig.s 4.2c - 4.6c. Recall that the only difference between data in Fig.s 4.1c and 4.1d is that one dimension of $\mathbf{s}$ has large variance but without any label information. From Fig. 4.5c, one can see that HSIC is unable to find a good subspace for data in Fig. 4.1d. This implies that HSIC is sensitive to dimensions with large variance. Finally, HSIC is unable to find a good subspace for data in Fig. 4.1f when $K = 2$ as shown in Fig. 4.7c.

3. FDA without kernel trick is unable to find a subspace that can separate data with different labels in all the six cases.

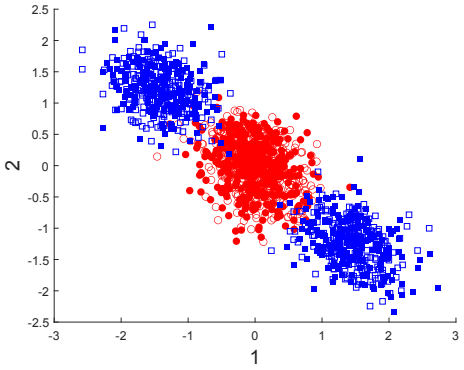
For the comparison of three methods *with* kernel trick, we have observations as follows:

1. With kernelization, both MDE and HSIC are capable of finding a good subspace for

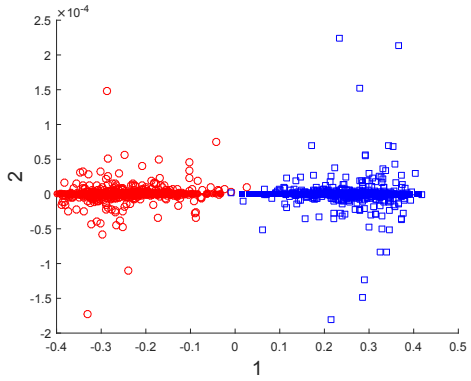


(a) MDE

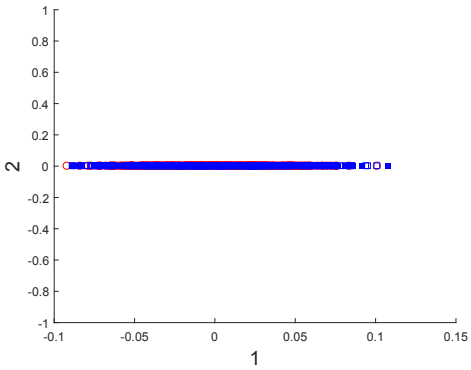
(b) MDE with RBF kernel ($\gamma = 0.1$)



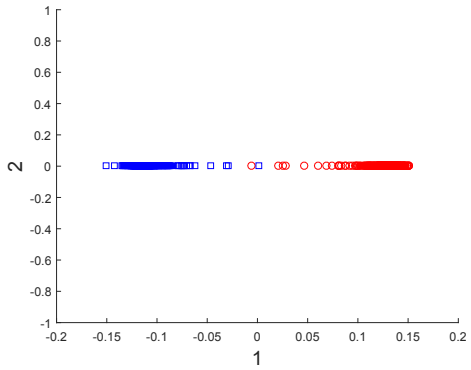
(c) HSIC



(d) kernel HSIC with RBF kernel ($\gamma = 0.1$)



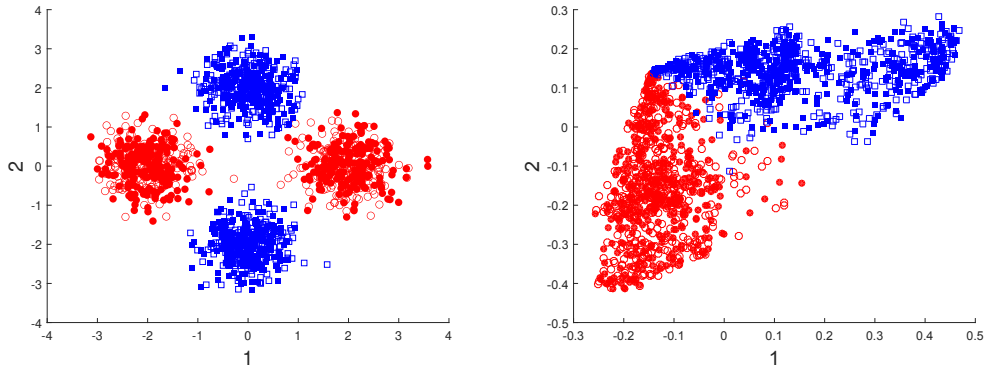
(e) FDA



(f) FDA with RBF kernel ($\gamma = 1$)

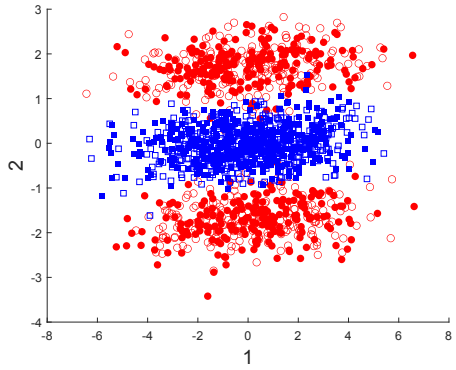
Figure 4.2: Projection result for data in Fig. 4.1a

data in Fig. 4.1. Furthermore, by properly setting γ in the RBF kernel, data with the same label are grouped in the subspace which makes classification easier. However, kernel trick does not help FDA to find a good subspace for concentric data shown in Fig. 4.1e.

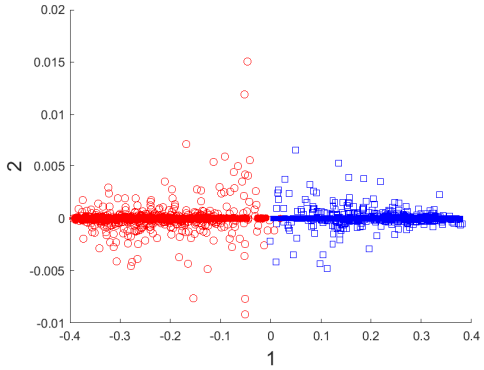


(a) MDE

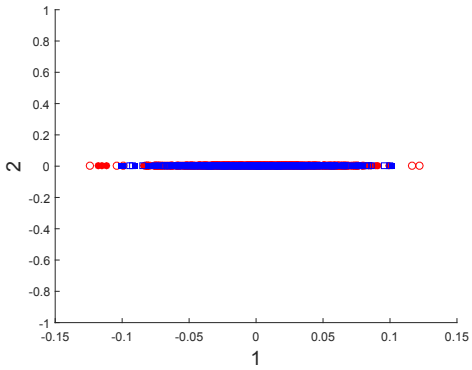
(b) MDE with RBF kernel ($\gamma = 0.14$)



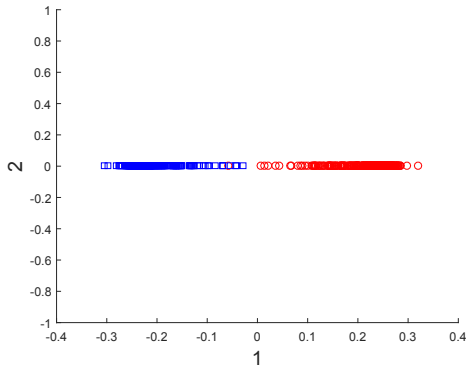
(c) HSIC



(d) HSIC with RBF kernel ($\gamma = 0.2$)



(e) FDA

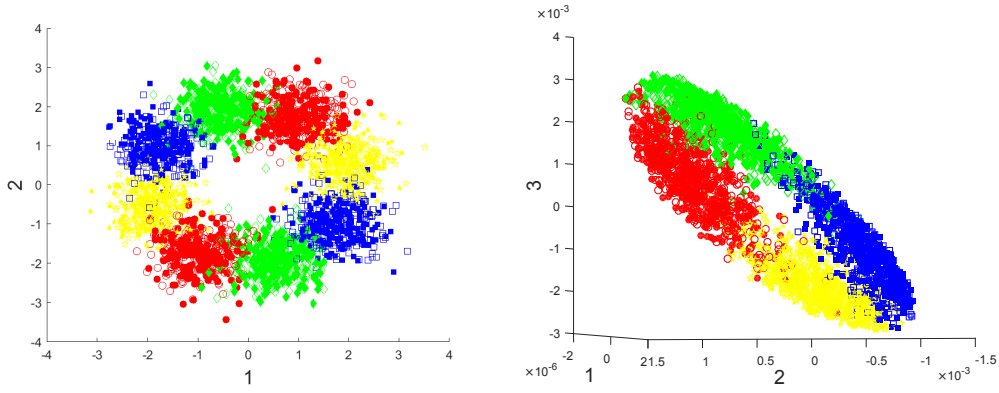


(f) FDA with RBF kernel ($\gamma = 1$)

Figure 4.3: Projection result for data in Fig. 4.1b

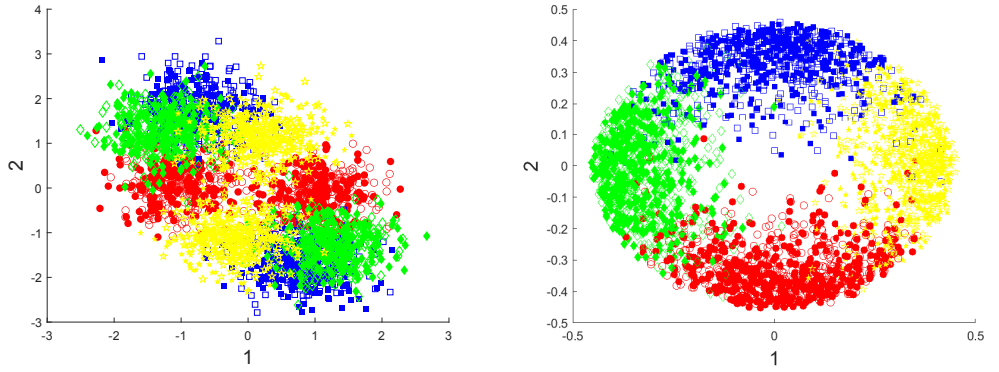
2. MDE benefit *less* from kernel trick than HSIC and FDA. As shown in Fig.s 4.4b and 4.5b, MDE requires a 3-D subspace to group all the data with the same label whereas HSIC and FDA only require a 2-D subspace.

We would like to note that the kernel trick increases the computational complexity



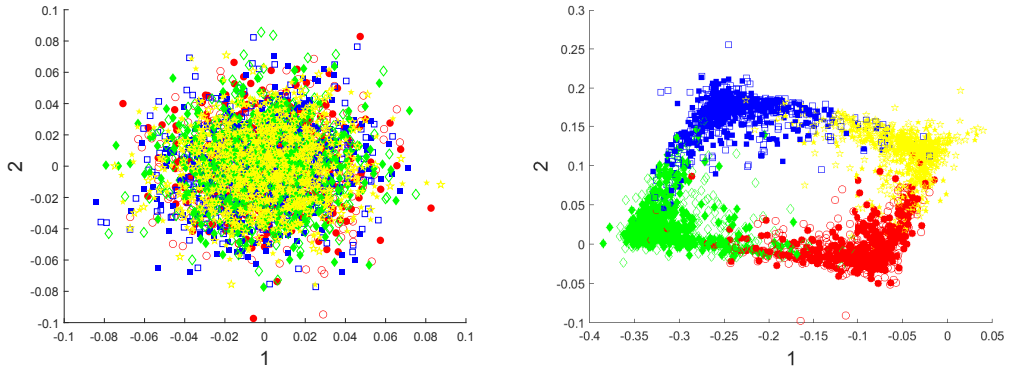
(a) MDE

(b) MDE with RBF kernel ($\gamma = 0.14$)



(c) HSIC

(d) HSIC with RBF kernel ($\gamma = 0.2$)

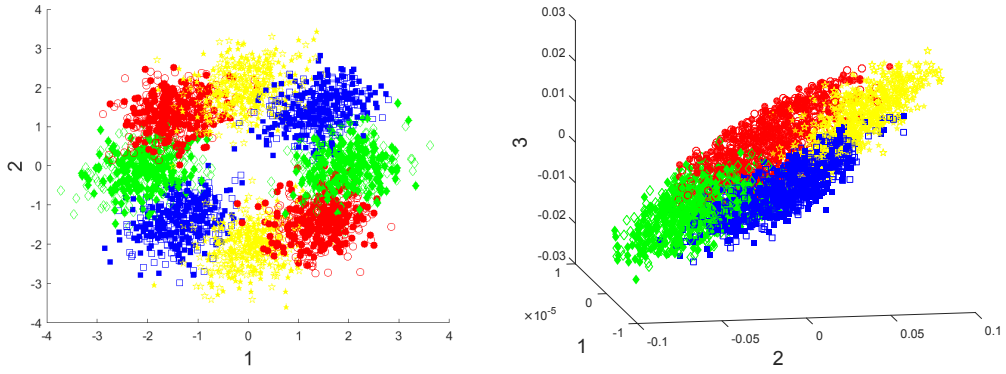


(e) FDA

(f) FDA with RBF kernel ($\gamma = 0.2$)

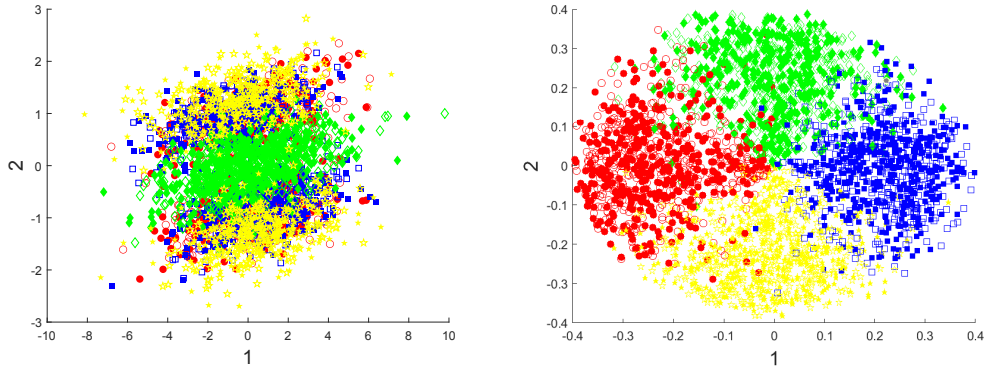
Figure 4.4: Projection result for data in Fig. 4.1c

significantly since it requires computing the kernel function for all pairs of samples.



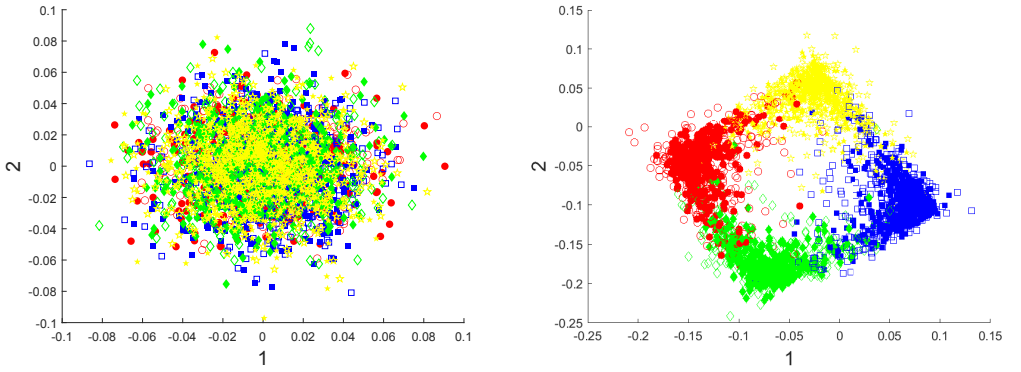
(a) MDE

(b) MDE with RBF kernel ($\gamma = 0.088$)



(c) HSIC

(d) HSIC with RBF kernel ($\gamma = 0.2$)



(e) FDA

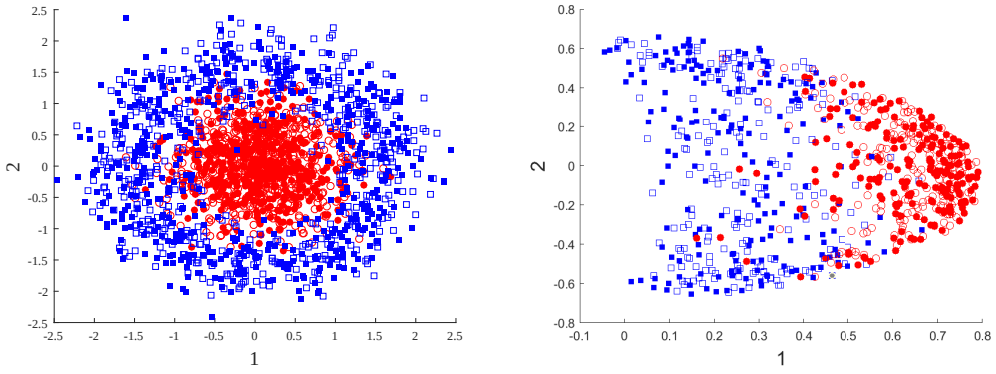
(f) FDA with RBF kernel ($\gamma = 0.2$)

Figure 4.5: Projection result for data in Fig. 4.1d

4.5.2 WiFi Sensing data

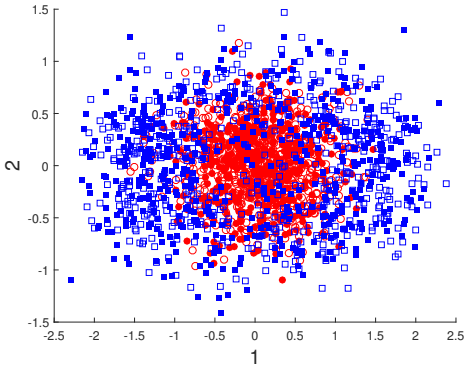
We now consider dimensionality reduction for data collected for presence detection via WiFi signal. This dataset[1] is originally collected and used in [55]. In this simulation,

[1]The dataset is available at https://github.com/bigtreeyanger/presence_detection_cnn

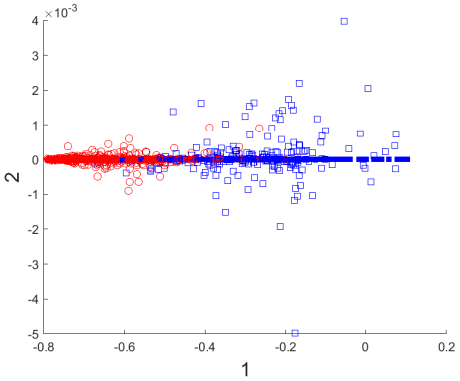


(a) MDE

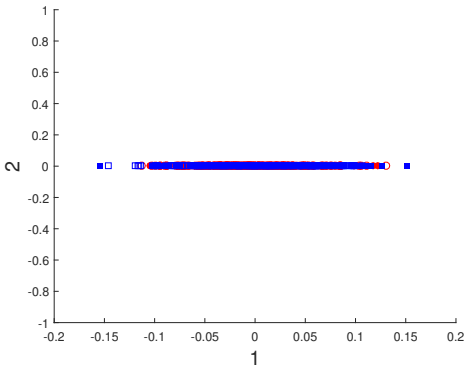
(b) MDE with RBF kernel ($\gamma = 0.2$)



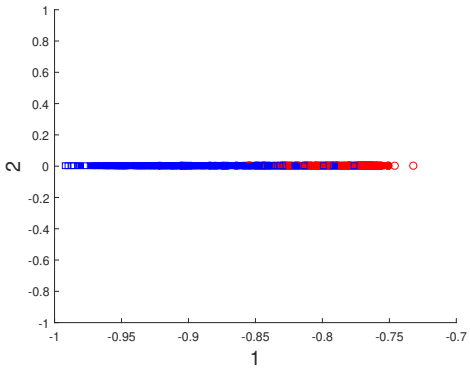
(c) HSIC



(d) HSIC with RBF kernel ($\gamma = 0.2$)



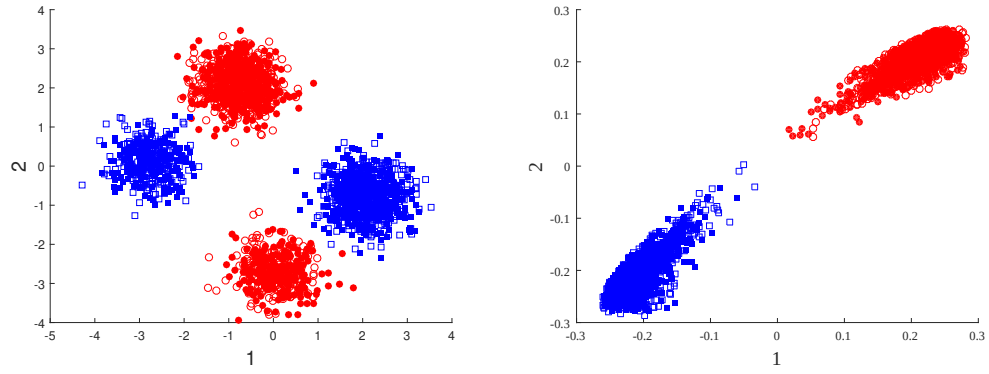
(e) FDA



(f) FDA with RBF kernel ($\gamma = 0.2$)

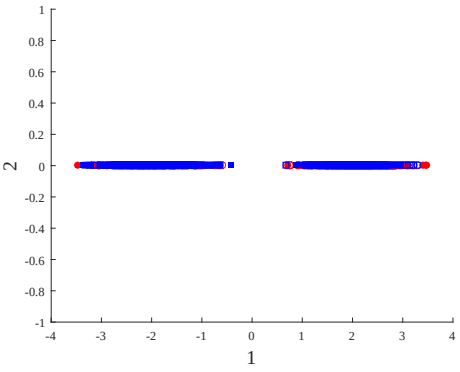
Figure 4.6: Projection result for data in Fig. 4.1e

$\tilde{\mathbf{X}}^{\text{abs-fft-crop}} \in \mathbb{R}^{64\times 7\times 9}$ is constructed in the way described in [55]. Denote by $\tilde{\mathbf{X}}^{\text{abs-fft-crop}}_{i,j,k}$ the (i,j,k)-th entry in $\tilde{\mathbf{X}}^{\text{abs-fft-crop}}$. Then the WiFi sensing data for dimension reduction is the vectorization of $\mathbf{X}$, where $\mathbf{X}_{i,j} = \frac{1}{9}\sum_{k=1}^{9} \tilde{\mathbf{X}}^{\text{abs-fft-crop}}_{i,j,k}$. Training data is randomly chosen from days 9, 11, 12 and 14 such that there are exactly 500 samples for each label. Note that

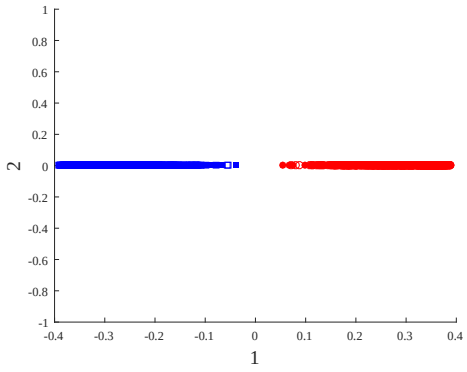


(a) MDE

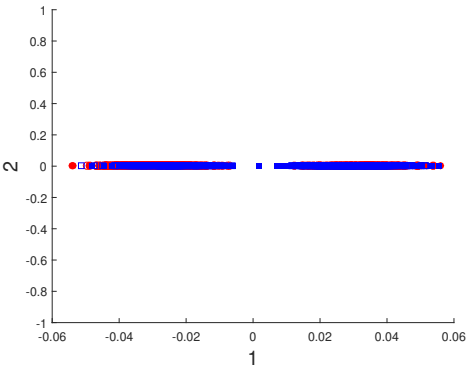
(b) MDE with RBF kernel ($\gamma = 0.1$)



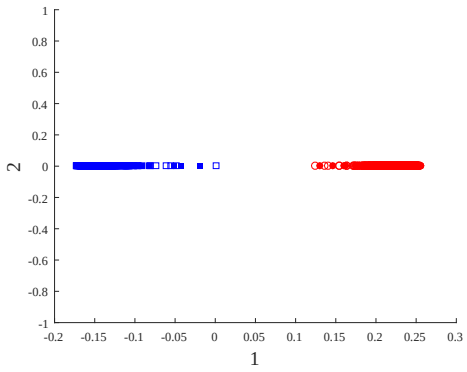
(c) HSIC



(d) HSIC with RBF kernel ($\gamma = 0.1$)



(e) FDA



(f) FDA with RBF kernel ($\gamma = 0.1$)

Figure 4.7: Projection result for data in Fig. 4.1f

the training data chosen for the four dimentionality reduction methods are identical. Test data are *all* the data collected on days 4, 7, 15, 16, 17, and 24. On each day, there are at least 5000 samples. All the data are collected in the same lab within two months.

The dimension of the subspace K is set to be 3. Both k-nearest neighbor (KNN) with

$k = 11$ and support vector machine (SVM) with RBF kernel with $\gamma = \frac{1}{3}$ are used for classification on the data after dimension reduction. The classification performance of WiFi sensing data after dimensionality reduction by MDE, HSIC, FDA and PCA with/without kernel trick is summarized in Tables 4.1 and 4.2, respectively. Since there are only two classes, FDA always provides one-dimensional results. No SVM is applied to the projection result of FDA. One can see from Table 4.1 that MDE outperforms HSIC, FDA and PCA if KNN is used for classification. Meanwhile, on day 17, the performance of PCA is much worse than MDE and HSIC with SVM used as the classifier, which implies that the use of label information helps find a better subspace for the data. The RBF kernel is used in kernel MDE, HSIC, FDA and PCA. The coefficient $\gamma \in [0.01, 1]$ in the RBF kernel is chosen by 10-fold validation on the training data. The final choices of γ for kernel MDE, HSIC, FDA and PCA are 0.01, 0.05, 0.1 and 0.01, respectively. On day 24, PCA has much larger error rate than the three SDR methods, which implies that even with kernel trick, unsupervised dimensionality reduction can lose significant label information.

Another interesting observation is that given WiFi sensing data, kernel trick does not help to improve the performance of dimensionality reduction methods whereas given synthetic data, kernel trick is shown to group data by their labels. This implies that kernel trick for dimensionality reduction methods may not be necessary for some real datasets.

4.6 Summary

This chapter studied the supervised dimensionality reduction problem. An SDR approach is proposed which maximizes the average energy difference preserved in the subspace. The projection results of synthetic data show that the proposed SDR method outperforms existing SDR approaches based in eigen-decomposition such as FDA and HSIC if no kernel trick is applied. Meanwhile, the proposed SDR method, FDA and HSIC achieve similar performance given kernel trick. WiFi sensing data is used to test the performance of the pro-

Table 4.1: Error probability given WiFi sensing data without kernel trick

day num	classification	MDE	HSIC	FDA	PCA
4	KNN	0	0.0009	0.0005	0
	SVM	0	0.0009		0
7	KNN	0.0029	0.0069	0.0132	0.0040
	SVM	0.0047	0.0049		0.0063
15	KNN	0.0007	0.0022	0.0013	0.0018
	SVM	0.0006	0.0003		0.0016
16	KNN	0.0024	0.0035	0.0048	0.0032
	SVM	0.0019	0.0039		0.0069
17	KNN	0.0005	0.0009	0.0028	0.0004
	SVM	0.0004	0.0007		0.1384
24	KNN	0.0059	0.0070	0.0231	0.0081
	SVM	0.0058	0.0064		0.0094

Table 4.2: Error probability given WiFi sensing data with kernel trick

day num	classification	kernel MDE	kernel HSIC	kernel FDA	kernel PCA
4	KNN	0	0	0	0.0015
	SVM	0	0		0.0002
7	KNN	0.0034	0.0087	0.0056	0.0015
	SVM	0.004	0.0076		0.0024
15	KNN	0.001	0.0012	0.0004	0
	SVM	0.0002	0.0027		0.0003
16	KNN	0.0031	0.0071	0.0017	0.0017
	SVM	0.0019	0.0054		0.0008
17	KNN	0.0002	0.0002	0.0002	0.0021
	SVM	0.0002	0.0007		0.0004
24	KNN	0.0077	0.0109	0.0093	0.0255
	SVM	0.0054	0.0108		0.0270

posed SDR method for classification problems. It is shown that the proposed SDR method outperforms FDA and HSIC in every case if KNN is used as the classifier. Furthermore, the proposed SDR also outperforms unsupervised PCA on average.

CHAPTER 5
CONCLUSION AND FUTURE RESEARCH

5.1 Conclusion

In this dissertation, two machine learning problems are studied. The first one is the sequence clustering problem, in which sequences are assumed to be generated from *unknown* continuous distributions and the goal is to group sequences according to some well-defined distribution metrics. The upper bound on the error probability of clustering algorithms is investigated under distribution distance metrics.

In Chapter 2, upper bounds on error probability for the k-medoids algorithm were derived that help establish the exponential consistency of the k-medoids algorithm under certain conditions on the distance metrics and the underlying distribution clusters. In particular, the exponential consistency of k-medoids is established for both known and unknwon number of clusters under the KS distance and MMD.

In Chapter 3, the asymptotic performance of HAC algorithms for clustering samples generated from distribution clusters is studied. The derived upper bounds on the error probability implies the exponential consistency of HAC algorithms under certain conditions on the distance metrics and the underlying distribution clusters. In particular, both linkage-based and centroid-based clustering algorithms under the KS distance and MMD were

shown to be exponentially consistent.

The second problem is supervised dimensionality reduction which attempts to find a lower dimensional subspace which preserves label information for data used in supervised machine learning problems.

In Chapter 4, maximum discriminant energy is proposed, which takes into account the label information to preserve maximum discriminating information for classification problems. The performance of the proposed MDE is validated by both synthetic data and WiFi sensing data. The projection results of synthetic data show that the proposed SDR method outperforms existing SDR approaches based on eigen-decomposition such as FDA and HSIC if no kernel trick is applied. Given WiFi sensing data, the proposed SDR method outperforms FDA and HSIC without kernel trick if KNN is used as the classifier. Furthermore, the proposed SDR achieves performance comparable to FDA and HSIC with kernel trick.

5.2 Future Research

We counclude the dissertation by listing several future research directions.

1. The preseted work in Chapters 2 and 3 assumes i.i.d. samples generated from distributions. However, samples are usually correlated in practice. One possible future work is to investigate the upper bound on the error probability of clustering algorithms given correlated data.

2. Another crucial assumption for the presented work in Chapters 2 and 3 is Assumption 1 which requires that the maximum intra-cluster distance between distributions is always smaller than the minimum inter-cluster distance between distributions. One possible future work is to investigate whether exponential consistency can be established while relaxing the condition in Assumption 1.

3. In Chapter 4, MDE is applied to sample of two classes since MDE tries to maximize the difference of average energy preserved in the subspace. One possible future work is to modify MDE so that it can handle multi-class case without the need for additional PCA after multiple MDEs.

4. MDE proposed in this dissertation is only validated by WiFi sensing data which is a binary classification problem. Related to 3, extending MDE to multi-class classification problems and investigating its performance are necessary to broaden its applications.

APPENDIX A
TECHNICAL LEMMAS

The following technical lemmas are used to prove Corollaries 2.2.1.1, 2.3.1.1 and 2.3.2.1. All the data sequences in Lemmas A.0.3 - A.0.8 are assumed to consist of n i.i.d. samples.

Lemma A.0.1. [Dvoretzky-Kiefer-Wolfowitz Inequality [69]] *Suppose* $\mathbf{x}$ *consists of* n *i.i.d. samples generated from* p*. Then*

$$P\left(d_{KS}\left(\mathbf{x}, p\right) > \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right).$$

Theorem A.0.2. [Theorem 7 in [65]] *Suppose* $\mathbf{x} \sim p$*,* $\mathbf{y} \sim q$*, where* $\mathbf{x}$ *and* $\mathbf{y}$ *have* m *and* n *samples, respectively. Given* $0 \leq g(x, y) \leq \mathbb{G}$*, the following inequality holds:*

$$\begin{aligned} &P\left(\left|\mathbb{MMD}(\mathbf{x}, \mathbf{y}) - \mathbb{MMD}(p, q)\right| > f(\mathbb{G}, m, n) + \epsilon\right) \\ &\leq 2\exp\left(-\frac{\epsilon^2 mn}{2\mathbb{G}\left(m+n\right)}\right). \end{aligned}$$

where $f(\mathbb{G}, m, n) = 2\left(\sqrt{\frac{\mathbb{G}}{m}} + \sqrt{\frac{\mathbb{G}}{n}}\right)$.

Lemmas A.0.3 - A.0.8 establish that the KS distance and the MMD statistic obtained by (2.7) satisfy Assumption 2 if the distribution clusters satisfy Assumption 1. Moreover, the lemmas provided in [21] are special cases of Lemmas A.0.3, A.0.5 and A.0.7 with $d_L = 0$.

Lemma A.0.3. *Suppose $\mathbf{x}_j \sim p_j$ for $j = 1, 2$, where $p_j \in \mathcal{P}$ and $d_{KS}(\mathcal{P}) \leq d_{L,ks}$. Then for any $d_0 > d_{L,ks}$,*

$$P\big(d_{KS}(\mathbf{x}_1, \mathbf{x}_2) > d_0\big) \leq 4 \exp\left(-\frac{n(d_0 - d_{L,ks})^2}{2}\right).$$

Proof. Consider

$$\begin{aligned}
&P\big(d_{KS}(\mathbf{x}_1, \mathbf{x}_2) > d_0\big) \\
&\leq P\big(d_{KS}(\mathbf{x}_1, p_1) + d_{KS}(p_1, p_2) + d_{KS}(\mathbf{x}_2, p_2) > d_0\big) \\
&\leq P\big(d_{KS}(\mathbf{x}_1, p_1) + d_{L,ks} + d_{KS}(\mathbf{x}_2, p_2) > d_0\big) \\
&\leq P\left(d_{KS}(\mathbf{x}_1, p_1) > \frac{\hat{d}}{2}\right) + P\left(d_{KS}(\mathbf{x}_2, p_2) > \frac{\hat{d}}{2}\right) \\
&\leq 4 \exp\big(-\frac{n\hat{d}^2}{2}\big),
\end{aligned}$$

where $\hat{d} = d_0 - d_{L,ks}$. The first inequality is due to the triangle inequality of the L_1-norm and the property of the supremum, and the last inequality is due to Lemma A.0.1. Therefore, we have

$$P\big(d_{KS}(\mathbf{x}_1, \mathbf{x}_2) > d_0\big) \leq 4 \exp\left(-\frac{n(d_0 - d_{L,ks})^2}{2}\right). \qquad \square$$

Lemma A.0.3 implies that the KS distance satisfies (2.4b) for $d > d_{L,ks}$.

Lemma A.0.4. *Suppose $\mathbf{x}_j \sim p_j$ for $j = 1, 2$, where $p_j \in \mathcal{P}$ and $\mathbb{MMD}(\mathcal{P}) \leq d_{L,mmd}$. Then for any $d_0 > d_{L,mmd}$ and sufficiently large n,*

$$P\big(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) > d_0\big) \leq 2 \exp\left(-\frac{n\left(d_0 - d_{L,mmd}\right)^2}{16\mathbb{G}}\right).$$

Proof. Since $\mathbb{MMD}(p_1, p_2) \leq d_{L,mmd}$, we have

$$\begin{aligned}
&P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) > d_0\right) \\
\leq\ &P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) - \mathbb{MMD}(p_1, p_2) > d_0 - d_{L,mmd}\right) \\
\leq\ &P\left(|\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) - \mathbb{MMD}(p_1, p_2)| > d_0 - d_{L,mmd}\right).
\end{aligned}$$

Choose $\epsilon = \frac{d_0 - d_{L,mmd}}{2}$ and n sufficiently large such that $f\left(\mathbb{G}, n, n\right) + \epsilon < d_0 - d_{L,mmd}$. By Theorem A.0.2, we have,

$$P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) > d_0\right) \leq 2\exp\left(-\frac{n\left(d_0 - d_{L,mmd}\right)^2}{16\mathbb{G}}\right). \qquad \square$$

Lemma A.0.4 implies that the MMD statistc satisfies (2.4b) for $d > d_{L,mmd}$.

Lemma A.0.5. *Suppose two distribution clusters $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy Assumption 1 under the KS distance. Assume that for $j = 1, 2$, $\mathbf{x}_j \sim p_j$ where $p_j \in \mathcal{P}_j$. Then for any $d_0 < d_{H,ks}$,*

$$P\big(d_{KS}(\mathbf{x}_1, \mathbf{x}_2) \leq d_0\big) \leq 4\exp\bigg(-\frac{n(d_{H,ks} - d_0)^2}{2}\bigg).$$

Proof. Similar to the proof of A.0.3, we have

$$\begin{aligned}
&P\big(d_{KS}(\mathbf{x}_1, \mathbf{x}_2) \leq d_0\big) \\
\leq\ &P\big(-d_{KS}(\mathbf{x}_1, p_1) + d_{KS}(p_1, p_2) - d_{KS}(\mathbf{x}_2, p_2) \leq d_0\big) \\
\leq\ &P\big(-d_{KS}(\mathbf{x}_1, p_1) + d_2 - d_{KS}(\mathbf{x}_2, p_2) < d_0\big) \\
\leq\ &P\bigg(d_{KS}(\mathbf{x}_1, p_1) > \frac{\hat{d}}{2}\bigg) + P\bigg(d_{KS}(\mathbf{x}_2, p_2) > \frac{\hat{d}}{2}\bigg) \\
\leq\ &4\exp\big(-\frac{n\hat{d}^2}{2}\big),
\end{aligned}$$

where $d_0 < d_2 < d_{H,ks}$, $\hat{d} = d_2 - d_0$ and $\lim_{d_2 \uparrow d_{H,ks}} = d_{H,ks} - d_0$. The last inequality is

due to Lemma A.0.1. Therefore, by the continuity of the exponential function, we have

$$P\big(d_{KS}(\mathbf{x}_1, \mathbf{x}_2) \leq d_0\big) \leq 4 \exp\bigg(- \frac{n(d_{H,ks} - d_0)^2}{2}\bigg). \qquad \square$$

Lemma A.0.5 implies that the KS distance satisfies (2.4a) for $d > d_{H,ks}$.

Lemma A.0.6. *Suppose two distribution clusters $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy Assumption 1 under MMD. Assume that for $j = 1, 2$, $\mathbf{x}_j \sim p_j$, where $p_j \in \mathcal{P}_j$. Then for any $d_0 < d_{H,mmd}$ and sufficiently large n,*

$$P\big(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) \leq d_0\big) \leq 2 \exp\left(-\frac{n\left(d_{H,mmd} - d_0\right)^2}{16\mathbb{G}}\right).$$

Proof. Similar to the proof of Lemma A.0.4, we have

$$\begin{aligned}
&P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) \leq d_0\right) \\
\leq\ &P\left(\mathbb{MMD}(p_1, p_2) - \mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) \geq d_{H,mmd} - d_0\right) \\
\leq\ &P\left(\left|\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) - \mathbb{MMD}(p_1, p_2)\right| > \hat{d}\right)
\end{aligned}$$

where $\hat{d} = d_3 - d_0$ and $d_0 < d_3 < d_{H,mmd}$. Choose $\epsilon = \frac{\hat{d}}{2}$ and n sufficiently large such that $f(\mathbb{G}, n, n) + \epsilon < \hat{d}$. By Theorem A.0.2, we have

$$P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) > d_0\right) \leq 2 \exp\left(-\frac{n\hat{d}^2}{16\mathbb{G}}\right).$$

Let $\lim_{d_3 \uparrow d_{H,ks}} = d_{H,ks} - d_0$. Then by the continuity of the exponential function, we have for n sufficiently large,

$$P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_2) \leq d_0\right) \leq 2 \exp\left(-\frac{n\left(d_{H,mmd} - d_0\right)^2}{16\mathbb{G}}\right). \qquad \square$$

Lemma A.0.6 implies that MMD satisfies (2.4a) for $d > d_{H,mmd}$.

Lemma A.0.7. [70] *Suppose two distribution clusters $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy Assumption 1 under the KS distance. Assume that for $j = 1, 2$, $\mathbf{x}_j \sim p_j$ with length n where $p_j \in \mathcal{P}_j$. Then for any $\mathbf{x}_3 \sim p_3$ with length n where $p_3 \in \mathcal{P}_1$,*

$$P\big(d_{KS}(\mathbf{x}_1, \mathbf{x}_3) \geq d_{KS}(\mathbf{x}_2,\mathbf{x}_3)\big) \leq 6 \exp\bigg(- \frac{n\Delta_{ks}^2}{8}\bigg).$$

Lemma A.0.7 implies that the KS distance satisfies (2.4c) for $d \in (d_{L,ks}, d_{H,ks})$.

Lemma A.0.8. *Suppose two distribution clusters $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy Assumption 1 under MMD. Assume that for $j = 1, 2$, $\mathbf{x}_j \sim p_j$ where $p_j \in \mathcal{P}_j$. Then for any $\mathbf{x}_3 \sim p_3$ where $p_3 \in \mathcal{P}_1$, where n is sufficiently large,*

$$P\big(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_3) \geq \mathbb{MMD}(\mathbf{x}_2, \mathbf{x}_3)\big) \leq 4 \exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right)$$

Proof. Let $\hat{\Delta} \in (0, \Delta_{mmd})$. Similar to the proof of Lemmas A.0.4 and A.0.6, we have

$$\begin{aligned}
&P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_3) \geq \mathbb{MMD}(\mathbf{x}_2, \mathbf{x}_3)\right) \\
&\leq P\big(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_3) - \mathbb{MMD}(p_1, p_3) + \mathbb{MMD}(p_2, p_3) - \mathbb{MMD}(\mathbf{x}_2, \mathbf{x}_3) \geq \Delta_{mmd}\big) \\
&\leq P\big(\big|\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_3) - \mathbb{MMD}(p_1, p_3)\big| + \big|\mathbb{MMD}(p_2, p_3) - \mathbb{MMD}(\mathbf{x}_2, \mathbf{x}_3)\big| > \hat{\Delta}\big) \\
&\leq P\big(\big|\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_3) - \mathbb{MMD}(p_1, p_3)\big| > \frac{\hat{\Delta}}{2}\big) + P\big(\big|\mathbb{MMD}(\mathbf{x}_2, \mathbf{x}_3) - \mathbb{MMD}(p_2, p_3)\big| > \frac{\hat{\Delta}}{2}\big),
\end{aligned}$$

where the last inequality is due to the union bound. Choose $\epsilon = \frac{\hat{\Delta}}{4}$ and n sufficiently large such that $f(\mathbb{G}, n, n) + \epsilon < \frac{\hat{\Delta}}{2}$. By Theorem A.0.2, we have

$$P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_3) \geq \mathbb{MMD}(\mathbf{x}_2, \mathbf{x}_3)\right) \leq 4 \exp\left(-\frac{n\hat{\Delta}^2}{64\mathbb{G}}\right).$$

Let $\hat{\Delta} \uparrow \Delta_{mmd}$. By the continuity of the exponential function, we have for n sufficiently large,

$$P\left(\mathbb{MMD}(\mathbf{x}_1, \mathbf{x}_3) \geq \mathbb{MMD}(\mathbf{x}_2, \mathbf{x}_3)\right) \leq 4 \exp\left(-\frac{n\Delta_{mmd}^2}{64\mathbb{G}}\right). \qquad \square$$

Lemma A.0.8 implies that MMD satisfies (2.4c) for $d \in (d_{L,mmd}, d_{H,mmd})$.

APPENDIX B
DETAILED PROOF OF THEOREMS IN CHAPTER 2

Define the following three events:

$$S_1(d_{th}) = \big\{\exists k, k' \in I_1^K, k \neq k', j \in I_1^{M_k}, j' \in I_1^{M_{k'}}, \text{ s.t. } d(\mathbf{x}_{k,j}, \mathbf{x}_{k',j'}) \leq d_{th}\big\},$$

$$S_2(d_{th}) = \big\{\exists k \in I_1^K, \ j, j' \in I_1^{M_k} \text{ s.t. } d(\mathbf{x}_{k,j}, \mathbf{x}_{k,j'}) > d_{th}\big\},$$

$$S_3 = \big\{\exists k, k' \in I_1^K, \ k \neq k', \ j_1, j_2 \in I_1^{M_k}, \ j' \in I_1^{M_{k'}}, \text{s.t. } d(\mathbf{x}_{k,j_1}, \mathbf{x}_{k,j_2}) \geq d(\mathbf{x}_{k,j_1}, \mathbf{x}_{k',j'})\big\},$$

where $d_{th} \in (d_L, d_H)$.

Assume that the sequences $\mathbf{x}_{k,j}$'s and the corresponding distribution clusters $\mathcal{P}_k$'s satisfy Assumption 1. By (2.4a) - (2.4c) and the union bound, we have

$$P\big(S_1(d_{th})\big) \leq \sum_{k=1}^{K} \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \sum_{j_k=1}^{M_k} \sum_{j_{k'}=1}^{M_{k'}} a_1 e^{-bn} \leq M^2 a_1 e^{-bn}, \tag{B.1a}$$

$$P\big(S_2(d_{th})\big) \leq \sum_{k=1}^{K} \sum_{j_k=1}^{M_k} \sum_{j_{k'}=1}^{M_{k'}} a_2 e^{-bn} \leq M^2 a_2 e^{-bn}, \tag{B.1b}$$

$$P\big(S_3\big) \leq \sum_{k=1}^{K} \sum_{j_k=1}^{M_k} \sum_{j_{k'}=1}^{M_{k'}} a_3 e^{-bn} \leq M^2 a_3 e^{-bn}. \tag{B.1c}$$

The main idea of the proofs of Theorems 2.2.1, 2.3.1 and 2.3.2 is to show that the error event at each iteration is a subset of $S_1(d_{th}) \cup S_2(d_{th}) \cup S_3$.

B.1 Proof of Theorem 2.2.1

The convergence of Algorithm 2 results from the design of the algorithm. Consider the $(t-1)$-th clustering step and the t-th center update step. We have for $t \geq 1$,

$$\sum_{k=1}^{K} \sum_{\mathbf{y}_i \in \mathcal{C}_k^{t-1,a}} d(\mathbf{y}_i, \mathbf{c}_k^{t-1,a}) \geq \sum_{k=1}^{K} \sum_{\mathbf{y}_i \in \mathcal{C}_k^{t-1,a}} d(\mathbf{y}_i, \mathbf{c}_k^{t,a}). \tag{B.2}$$

Moreover, for the t-th center update and the t-th cluster update, we have for $t \geq 1$,

$$\sum_{k=1}^{K} \sum_{\mathbf{y}_i \in \mathcal{C}_k^{t-1}} d(\mathbf{y}_i, \mathbf{c}_k^{t,a}) \geq \sum_{l=1}^{K} \sum_{\mathbf{y}_i \in \mathcal{C}_k^{t}} d(\mathbf{y}_i, \mathbf{c}_k^{t,a}). \tag{B.3}$$

The equalities in (B.2) and (B.3) hold if and only if $\mathcal{C}_k^{t-1} = \mathcal{C}_k^t$ and $\mathbf{c}_k^{t-1,a} = \mathbf{c}_k^{t,a}$ for $k = 1, \ldots, K$ respectively which implies the convergence of the algorithm.

Suppose there are K sequences assigned as cluster centers, and as a result $M - K$ remaining sequences are to be assigned to cluster centers. The order in which cluster centers are chosen does not matter, so there are a total of $\binom{M}{K}$ permutations of them. Since each of the remaining $M - K$ sequences can be assigned to one and only one cluster center, there are a total of $K^{(M-K)}$ possible assignments. Therefore the total number of valid partitions is $\binom{M}{K} K^{(M-K)}$. By (B.2) and (B.3), Algorithm 2 is guaranteed to visit each possible partition at most once except the one coinciding with the clustering output. Hence the maximum number of algorithm iterations is always upper bounded as

$$T \leq \binom{M}{K} K^{(M-K)}.$$

Define for $t \geq 1$,

$$E^t = \{\text{After } t\text{-th iteration, there are } K_1 \text{ centers generated from } K_2 \text{ distribution clusters}\}.$$

where

$$K_1 \begin{cases} > K_2 & \text{if } K_2 = K, \\ \geq K_2 & \text{if } K_2 < K. \end{cases}$$

Similarly, define

$$E^0 = \{\text{The center initialization obtains } K_1 \text{ centers generated from } K_2 \text{ distribution clusters}\}.$$

Then E^t for $t \geq 0$ denotes the error event that centers are incorrectly chosen at the center initialization or the t-th center update. We first consider the error occurs at the initialization step. For Algorithm 2,

$$\begin{aligned} E^0 &= \{\text{The center initialization results in } K \text{ centers generated from } K_2 \ (< K) \text{ distribution clusters centers.}\} \\ &= \{\exists k, l, l' \in I_1^K,\ l \neq l' \text{ s.t. } \mathbf{c}_l^{0,a}, \mathbf{c}_{l'}^{0,a} \sim \mathcal{P}_k\}. \end{aligned}$$

Moreover, define

$$\begin{aligned} E_1^0 &= E^0 \cap \left\{\exists l, l' \in \{1, \ldots, K\} \text{ s.t. } d(\mathbf{c}_l^{0,a}, \mathbf{c}_{l'}^{0,a}) \leq d_{th}\right\}, \\ E_2^0 &= E^0 \cap \left\{\exists l, l' \in \{1, \ldots, K\} \text{ s.t. } d(\mathbf{c}_l^{0,a}, \mathbf{c}_{l'}^{0,a}) > d_{th}\right\}. \end{aligned}$$

Then $E^0 = E_1^0 \cup E_2^0$. Without loss of generality, assume that $\mathbf{c}_1^{0,a}, \ldots, \mathbf{c}_K^{0,a}$ are chosen sequentially at the center initialization step and $l < l'$. Then E_1^0 implies that for all the

sequences $\mathbf{z} \in \{\mathbf{y}_i\}_{i=1}^{M} \setminus \{\mathbf{c}_m^{0,a}\}_{m=1}^{l'}$,

$$\min_{m\in\{1,\ldots,l'-1\}} d(\mathbf{c}_m^{0,a}, \mathbf{z}) \leq d_{th}.$$

Thus, $E_1^0 \subset S_1(d_{th})$. Then by (B.1a), we have

$$P\big(E_1^0\big) \leq P\big(S_1(d_{th})\big) \leq M^2 a_1 e^{-bn}.$$

Moreover, since $E_2^0 \subset S_2(d_{th})$, by (B.1b), we have

$$P\big(E_2^0\big) \leq P\big(S_2(d_{th})\big) \leq M^2 a_2 e^{-bn}.$$

Thus, the error probability at the center initialization step is bounded as follows

$$P\big(E^0\big) \leq M^2(a_1 + a_2)e^{-bn}. \tag{B.4}$$

We now consider the assignment step. Define for $t \geq 1$,

$$H^t = \{\text{The clustering result after the } t\text{-th cluster update is incorrect}\},$$

Moreover, define

$$H^0 = \{\text{The clustering initialization is incorrect}\}.$$

Since $E^t \subset H^{t-1}$ for $t \geq 1$, it is sufficient to obtain an upper bound on $P\big(H^t\big)$ which serves as the upper bound of $P(H^t \cup E^t)$. Define

$$\hat{H}_1^t = \begin{cases} H^0 \setminus E^0 & \text{for } t = 0, \\ H^t \setminus \left(E^0 \cup \left(\cup_{l=0}^{t-1} \left(H^l\right)\right)\right) & \text{for } t \geq 1. \end{cases}$$

Then $E^0 \cup \big(\cup_{t=1}^{T} H^t \big) = E^0 \cup \big(\cup_{t=0}^{T} \hat{H}_1^t \big)$, which is the event that Algorithm 2 makes an error before the first T iterations complete. Moreover, $\hat{H}_1^t$ implies the event that an error occurs at the t-th cluster update step *given* correct center update in the same iteration which is denoted by

$$\begin{aligned}\bar{H}_1^t = \big\{ &\exists k, k', l, l' \in I_1^K,\ k \neq k',\ j_k \in I_1^{M_k} \text{ s.t.} \\ &d(\mathbf{x}_{k,j_k}, \mathbf{c}_l^{t,a}) \geq d(\mathbf{x}_{k,j_k}, \mathbf{c}_{l'}^{t,a}) :\ \mathbf{c}_l^{t,a} \sim \mathcal{P}_k,\ \mathbf{c}_{l'}^{t,a} \sim \mathcal{P}_{k'} \big\}.\end{aligned}$$

Then $P(\hat{H}_1^t) \leq P\big(\bar{H}_1^t\big)$. Moreover, since $\bar{H}_1^t \subset S_3$, we have

$$P(\hat{H}_1^t) \leq P(\bar{H}_1^t) \leq P\big(S_3\big) \leq M^2 a_3 e^{-bn}. \tag{B.5}$$

Therefore, by (B.4), (B.5) and the union bound, the error probability of Algorithm 2 after T iterations is bounded by

$$\begin{aligned}P_e &= P\big(E^0 \cup \big(\cup_{t=0}^{T} \hat{H}_1^t \big)\big) \\ &\leq M^2 \big(a_1 + a_2 + (T+1)a_3\big) e^{-bn}.\end{aligned} \tag{B.6}$$

B.2 Proof of Theorem 2.3.1

If no merge step is executed and $\hat{K}$ clusters are found by Algorithm 3, then similar to the proof of Theorem 2.2.1 Algorithm 4 converges after at most T_0 iterations, where

$$T_0 = \binom{M}{\hat{K}} \hat{K}^{(M-\hat{K})}.$$

If the merge step is executed, the valid partitions before and after the merge step are mutually exclusive since the number of clusters is strictly decreasing. Therefore, Algorithm 4

converges after at most $T_{\max}$ iterations, where

$$T_{\max} = \sum_{\hat{K}=1}^{M} \binom{M}{\hat{K}} \hat{K}^{(M-\hat{K})}.$$

In conclusion, Algorithm 4 converges after at most $T_{\max}$ iterations since $T_0 < T_{\max}$.

We then analyze the error probability of Algorithm 4. We first consider the initialization step. Define

$$E_3^0 = E^0 \cap \big\{K_2 < K\big\},$$
$$E_4^0 = E^0 \cap \big\{K_2 = K\big\}.$$

Then $E^0 = E_3^0 \cup E_4^0$. Moreover, since

$$E_3^0 \subset \Big\{\exists k, k' \in I_1^K,\ j_k \in I_1^{M_k},\ j_{k'} \in I_1^{M_{k'}} \text{ s.t. } d(\mathbf{x}_{k,j_k}, \mathbf{x}_{k',j_{k'}}) \le d_{th}\Big\},$$
$$E_4^0 \subset \Big\{\exists k \in I_1^K,\ j_k, j_k' \in I_1^{M_k},\ \text{s.t. } d(\mathbf{x}_{k,j_k}, \mathbf{x}_{k,j_k'}) > d_{th}\Big\},$$

then $E_3^0 \subset S_1(d_{th})$ and $E_4^0 \subset S_2(d_{th})$. Thus, by (B.1a), (B.1b), we have

$$P\big(E_3^0\big) \le P\big(S_1(d_{th})\big) \le M^2 a_1 e^{-bn},$$
$$P\big(E_4^0\big) \le P\big(S_2(d_{th})\big) \le M^2 a_2 e^{-bn}.$$

Therefore, by the union bound, the probability that an error occurs at the center initialization step is bounded by

$$P\big(E^0\big) \le P\big(E_3^0\big) + P\big(E_4^0\big) \le M^2 a_1 e^{-bn} + M^2 a_2 e^{-bn}. \tag{B.7}$$

We now consider the error that occurs during iterations. $E^t \subset H^{t-1}$ for $t \ge 1$ still holds. Furthermore, define an incorrect merge as the event that the distance between two centers generated from different distribution clusters is smaller than d_{th}. Let D^t be the event that incorrect merges occur at the t-th ($t \ge 1$) merge step. Thus we only need to bound $P\big(H^t\big)$ and $P\big(D^t\big)$. Let $B_{t_1,t_2} = \big(\cup_{l=1}^{t_1} D^l\big) \cup \big(\cup_{l=0}^{t_2} H^l\big)$ for $t_1 \ge 1$ and $t_2 \ge 1$.

Define

$$\hat{D}^t = \begin{cases} D^1 & \text{for } t = 1 \\ D^t \setminus \big(E^0 \cup B_{t-1,t-1}\big) & \text{for } t > 1 \end{cases},$$

$$\hat{H}_2^t = \begin{cases} H^0 \setminus E^0 & \text{for } t = 0 \\ H^t \setminus \big(E^0 \cup B_{t,t-1}\big) & \text{for } t \geq 1 \end{cases}.$$

Then

$$\begin{aligned} & E^0 \cup \big(\cup_{t=1}^T D^t \big) \cup \big(\cup_{t=0}^T H^t \big) \\ = \; & E^0 \cup \big(\cup_{t=1}^T \hat{D}^t \big) \cup \big(\cup_{t=0}^T \hat{H}_1^t \big), \end{aligned}$$

which denotes the event that an error occurs before T iterations complete. Note that $\hat{D}^t$ implies the event that an error occurs at the t-th merge step *given* correct center update in the same iteration, which is denoted by

$$\bar{D}^t = \big\{ \exists k, k' \in I_1^K,\ k \neq k',\ l \in I_1^{\hat{K}^{t-1}},\ \text{s.t. } d(\mathbf{c}_l^{t,a}, \mathbf{c}_{l'}^{t,a}) \leq d_{th} :\ \mathbf{c}_l^{t,e} \sim \mathcal{P}_k, \mathbf{c}_{l'}^{t,e} \sim \mathcal{P}_{k'} \big\}.$$

Then $P\big(\hat{D}^t\big) \leq P\big(\bar{D}^t\big)$ and $\bar{D}^t \subset S_1(d_{th})$. Thus, by (B.1a), we have

$$P\big(\hat{D}^t\big) \leq P\big(\bar{D}^t\big) \leq P\big(S_1(d_{th})\big) \leq M^2 a_1 e^{-bn}. \tag{B.8}$$

Moreover, we have $P\big(\hat{H}_2^t\big) \leq P\big(\bar{H}_2^t\big)$, where

$$\begin{aligned} \bar{H}_2^t = \big\{ & \exists k, k' \in I_1^K,\ k \neq k',\ j_k \in I_1^{M_k},\ l, l' \in I_1^{\hat{K}^t},\ \text{s.t.} \\ & d(\mathbf{x}_{k,j_k}, \mathbf{c}_l^{t,e}) \geq d(\mathbf{x}_{k,j_k}, \mathbf{c}_{l'}^{t,e}) :\ \mathbf{c}_l^{t,e} \sim \mathcal{P}_k,\ \mathbf{c}_{l'}^{t,e} \sim \mathcal{P}_{k'} \big\}. \end{aligned}$$

Note that $P(\bar{H}_2^t)$ has the same upper bound as $P(\bar{H}_1^t)$ in (B.5). Therefore, by (B.7), (B.5)

and (B.8), the error probability after T iterations is bounded by

$$\begin{aligned} P_e &= P\big(Y^0 \cup \big(\cup_{t=0}^{T} \hat{H}_2^t\big) \cup \big(\cup_{t=1}^{T} \hat{D}^t\big)\big) \\ &\leq M^2\big((T+1)a_1 + a_2 + (T+1)a_3\big)e^{-bn}. \end{aligned} \tag{B.9}$$

B.3 Proof of Theorem 2.3.2

Note that in the extreme case, splitting results in each cluster containing only one sequence, i.e., splitting can happen at most $M-1$ times. Therefore, Algorithm 5 converges after at most M iterations. Furthermore, if $\hat{K}$ does not change from the $(t-1)$-th to the t-th iteration, then $\mathcal{C}_k^{t-1} = \mathcal{C}_k^t$ and $\mathbf{c}_k^{t-1} = \mathbf{c}_k^t$ for $k = 1, \ldots, \hat{K}$, which implies the convergence of the algorithm.

Let A^t be the event that the error occurs at the t-th split step. Then $A^t = A_1^t \cup A_2^t$, where

$$\begin{aligned} A_1^t = \big\{&\text{The algorithm fails to split any cluster containing sequences generated} \\ &\text{by diffierent distribution clusters at the } t\text{-th iteration}\big\}, \\ A_2^t = \big\{&\text{The algorithm splits a cluster containing sequences generated by} \\ &\text{one distribution clusters at the } t\text{-th iteration}\big\}. \end{aligned}$$

Let V^t denote the event that the clustering result at the t-th cluster update is incorrect. Then $A^t \cup V^t$ denotes the event that an error occurs at the t-th iteration. Define $\hat{A}^t = \hat{A}_1^t \cup \hat{A}_2^t$, where

$$\hat{A}_i^t = \begin{cases} A^1 & \text{for } t = 1, \\ A_i^t \setminus \big((\cup_{l=1}^{t-1} A^l) \cup (\cup_{l=1}^{t-1} V^l)\big) & \text{for } t > 1, \end{cases}$$

for $i = 1, 2$. Moreover, define

$$\hat{V}^t = \begin{cases} V^1 \setminus A^1 & \text{for } t = 1 \\ V^t \setminus \big((\cup_{l=1}^{t-1} V^l) \cup (\cup_{l=1}^{t} A^l)\big) & \text{for } t > 1 \end{cases}.$$

Then $\big(\cup_{t=1}^{T} A^t\big) \cup \big(\cup_{t=1}^{T} V^t\big) = \big(\cup_{t=1}^{T} \hat{A}^t\big) \cup \big(\cup_{t=1}^{T} \hat{V}^t\big)$. Since $\hat{A}_1^t \subset S_1(d_{th})$ and $\hat{A}_2^t \subset S_2(d_{th})$, then we have for $t = 1, \ldots, T$,

$$\begin{aligned} P\big(\hat{A}_1^t\big) &\leq P\big(S_1(d_{th})\big) \leq M^2 a_1 e^{-bn}, \\ P\big(\hat{A}_2^t\big) &\leq P\big(S_2(d_{th})\big) \leq M^2 a_2 e^{-bn}. \end{aligned}$$

Moreover, since $P\big(\hat{A}^t\big) = P\big(\hat{A}_1^t \cup \hat{A}_2^t\big)$, by the union bound

$$P\big(\hat{A}^t\big) \leq M^2 a_1 e^{-bn} + M^2 a_2 e^{-bn}. \tag{B.10}$$

Furthermore, by Definition 2.3.1.1, $\hat{V}^t$ implies the following event

$$\begin{aligned} \bar{V}^t = \big\{ &\exists l, l' \in I_1^{\hat{K}^t},\ k, k' \in I_1^K\ k' \neq k,\ j_k \in I_1^{M_k} \text{ s.t.} \\ &d(\mathbf{x}_{k,j_k}, \mathbf{c}_l^{t,s}) \geq d(\mathbf{x}_{k,j_k}, \mathbf{c}_{l'}^{t,s}) :\ \mathbf{c}_l^{t,s} \sim \mathcal{P}_k,\ \mathbf{c}_{l'}^{t,s} \sim \mathcal{P}_{k'} \big\}. \end{aligned}$$

Then, $P\big(\hat{V}^t\big) \leq P\big(\bar{V}^t\big)$ and $\bar{V}^t \subset S_3$. Thus, we have

$$P\big(\hat{V}^t\big) \leq P\big(\bar{V}^t\big) \leq M^2 a_3 e^{-bn}. \tag{B.11}$$

Therefore, by (B.10), (B.11) and the union bound, the error probability of Algorithm 5 after T iterations is bounded by

$$\begin{aligned} P_e &= P\Big(\big(\cup_{t=1}^{T} \hat{A}^t\big) \cup \big(\cup_{t=1}^{T} \hat{V}^t\big)\Big) \\ &\leq M^2 T\big(a_1 + a_2 + a_3\big) e^{-bn}. \end{aligned} \tag{B.12}$$

APPENDIX C
DETAILED PROOF OF THEOREMS IN CHAPTER 3

C.1 Proof of Proposition 2

Without loss of generality, assume that $\{\mathcal{C}_i^0\} = \{\mathbf{y}_i\}$ for $i = 1, \ldots, M$. Then (3.4) holds for $t = 0$ and $\theta_{ii'}^0(\mathbf{y}_i, \mathbf{y}_{i'}) = 1$. Since each $\mathcal{C}_l^0$ consists of one sequence, (3.5) holds for any $l \neq l'$ and $t = 0$.

Let $r_1^+ = \alpha_1 + \gamma$, $r_1^- = \alpha_1 - \gamma$, $r_2^+ = \alpha_2 + \gamma$ and $r_2^- = \alpha_2 - \gamma$, which are all non-negative by (3.2). Assume that (3.4) and (3.5) hold for $0 \leq t_0 \leq t$ and there are $L + 1$ clusters after the t-th iteration. Without loss of generality, we further assume that the last two clusters are combined as $\mathcal{C}_L^{t+1}$ in the $(t+1)$-th iteration, i.e., $\mathcal{C}_l^{t+1} = \mathcal{C}_l^t$ for $l < L$ and $\mathcal{C}_L^{t+1} = \mathcal{C}_L^t \cup \mathcal{C}_{L+1}^t$. Then at the $(t+1)$-th iteration, (3.4) and (3.5) hold for $l,\ l' \in I_1^{L-1}$, $l \neq l'$, since $d\left(\mathcal{C}_l^{t+1}, \mathcal{C}_{l'}^{t+1}\right) = d\left(\mathcal{C}_l^t, \mathcal{C}_{l'}^t\right)$. Furthermore, for any $l < L$, if

$d\left(\mathcal{C}_l^t, \mathcal{C}_L^t\right) \geq d\left(\mathcal{C}_l^t, \mathcal{C}_{L+1}^t\right)$, we have

$$
\begin{aligned}
& d\left(\mathcal{C}_l^{t+1}, \mathcal{C}_L^{t+1}\right) \\
&= r_1^+ d\left(\mathcal{C}_l^t, \mathcal{C}_L^t\right) + r_2^- d\left(\mathcal{C}_l^t, \mathcal{C}_{L+1}^t\right) \\
&= r_1^+ \sum_{i:\, \mathbf{y}_i \in \mathcal{C}_l^t} \sum_{i':\, \mathbf{y}_{i'} \in \mathcal{C}_L^t} \theta_{ii'}^t\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) d\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) + r_2^- \sum_{i:\, \mathbf{y}_i \in \mathcal{C}_l^t} \sum_{i':\, \mathbf{y}_{i'} \in \mathcal{C}_{L+1}^t} \theta_{ii'}^t\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) d\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) \\
&= \sum_{i:\, \mathbf{y}_i \in \mathcal{C}_l^t} \sum_{i':\, \mathbf{y}_{i'} \in \mathcal{C}_L^{t+1}} \theta_{ii'}^{t+1}\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) d\left(\mathbf{y}_i, \mathbf{y}_{i'}\right),
\end{aligned}
$$

where

$$
\theta_{ii'}^{t+1}\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) = \begin{cases} r_1^+ \theta_{ii'}^t\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) & \text{if } \mathbf{y}_{i'} \in \mathcal{C}_L^t, \\ r_2^- \theta_{ii'}^t\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) & \text{if } \mathbf{y}_{i'} \in \mathcal{C}_{L+1}^t. \end{cases}
$$

Since $r_1^+ + r_2^- = 1$ and (3.5) holds for t, then

$$
\sum_{i:\, \mathbf{y}_i \in \mathcal{C}_l^{t+1}} \sum_{i':\, \mathbf{y}_{i'} \in \mathcal{C}_L^{t+1}} \theta_{ii'}^{t+1}\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) d\left(\mathbf{y}_i, \mathbf{y}_{i'}\right) = 1.
$$

The case where $d\left(\mathcal{C}_l^t, \mathcal{C}_L^t\right) < d\left(\mathcal{C}_l^t, \mathcal{C}_{L+1}^t\right)$ can be proved in a similar manner.

C.2 Proof of Proposition 3

Note that if $\mathcal{C}_{l_1}^t = \mathcal{C}_{l_1}^{t-1}$, $\mathcal{C}_{l_2}^t = \mathcal{C}_{l_2}^{t-1}$ and $\mathcal{C}_{l_3}^t = \mathcal{C}_{l_3}^{t-1}$ for $t \geq 1$, then we can replace t by $t-1$ in (3.10). Thus, we only need to consider the case where $\mathcal{C}_{l_1}^t$ results from combining in the t-th iteration. Without loss of generality, assume that $\mathcal{C}_{l_1}^t = \mathcal{C}_{l_1}^{t-1} \cup \mathcal{C}_{l_4}^{t-1}$, where $\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_4}^{t-1} \sim \mathcal{P}_k$ and $t \geq 1$. We further assume that $\mathcal{C}_{l_2}^t = \mathcal{C}_{l_2}^{t-1} \sim \mathcal{P}_k$ and $\mathcal{C}_{l_3}^t = \mathcal{C}_{l_3}^{t-1} \sim \mathcal{P}_{k'}$, where $k \neq k'$.

Then by (3.1), we have

$$\begin{aligned}
&d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_2}^t\right) - d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_3}^t\right) \\
&= \alpha_1 d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_2}^{t-1}\right) + \alpha_2 d\left(\mathcal{C}_{l_4}^{t-1}, \mathcal{C}_{l_2}^{t-1}\right) + \beta d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_4}^{t-1}\right) \\
&\quad - \alpha_1 d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_3}^{t-1}\right) - \alpha_2 d\left(\mathcal{C}_{l_4}^{t-1}, \mathcal{C}_{l_3}^{t-1}\right) - \beta d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_4}^{t-1}\right) \\
&= \alpha_1\big[d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_2}^{t-1}\right) - d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_3}^{t-1}\right)\big] + \alpha_2\big[d\left(\mathcal{C}_{l_4}^{t-1}, \mathcal{C}_{l_2}^{t-1}\right) - d\left(\mathcal{C}_{l_4}^{t-1}, \mathcal{C}_{l_3}^{t-1}\right)\big].
\end{aligned} \tag{C.1}$$

By (C.1), we have for $t = 1$,

$$\begin{aligned}
&P\left(d\left(\mathcal{C}_{l_1}^1, \mathcal{C}_{l_2}^1\right) \geq d\left(\mathcal{C}_{l_1}^1, \mathcal{C}_{l_3}^1\right)\right) \\
&= P\big(\alpha_1\big[d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_2}^0\right) - d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_3}^0\right)\big] + \alpha_2\big[d\left(\mathcal{C}_{l_4}^0, \mathcal{C}_{l_2}^0\right) - d\left(\mathcal{C}_{l_4}^0, \mathcal{C}_{l_3}^0\right)\big] \geq 0\big) \\
&\leq P\big(d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_2}^0\right) - d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_3}^0\right) \geq 0\big) + P\big(d\left(\mathcal{C}_{l_4}^0, \mathcal{C}_{l_2}^0\right) - d\left(\mathcal{C}_{l_4}^0, \mathcal{C}_{l_3}^0\right) \geq 0\big) \\
&\leq 2a_3 e^{-nb_3},
\end{aligned}$$

where the two inequalities are due to the union bound and (2.4c), respectively. Assume that for $1 \leq t_0 \leq t$,

$$P\left(d\left(\mathcal{C}_{l_1}^{t_0}, \mathcal{C}_{l_2}^{t_0}\right) \geq d\left(\mathcal{C}_{l_1}^{t_0}, \mathcal{C}_{l_3}^{t_0}\right)\right) \leq 2^{t_0} a_3 e^{-nb_3}. \tag{C.2}$$

Then for $t_0 = t + 1$, we have

$$\begin{aligned}
&P\left(d\left(\mathcal{C}_{l_1}^{t+1}, \mathcal{C}_{l_2}^{t+1}\right) \geq d\left(\mathcal{C}_{l_1}^{t+1}, \mathcal{C}_{l_3}^{t+1}\right)\right) \\
&= P\big(\alpha_1\big[d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_2}^t\right) - d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_3}^t\right)\big] + \alpha_2\big[d\left(\mathcal{C}_{l_4}^t, \mathcal{C}_{l_2}^t\right) - d\left(\mathcal{C}_{l_4}^t, \mathcal{C}_{l_3}^t\right)\big] \geq 0\big) \\
&\leq P\big(d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_2}^t\right) - d\left(\mathcal{C}_{l_1}^t, \mathcal{C}_{l_3}^t\right) \geq 0\big) + P\big(d\left(\mathcal{C}_{l_4}^t, \mathcal{C}_{l_2}^t\right) - d\left(\mathcal{C}_{l_4}^t, \mathcal{C}_{l_3}^t\right) \geq 0\big) \\
&\leq 2^{t+1} a_3 e^{-nb_3},
\end{aligned}$$

where the two inequalities are due to the union bound and (C.2), respectively. We then prove (3.10b). For $t = 1$, we have

$$\begin{aligned}
&P\left(d\left(\mathcal{C}_{l_1}^1, \mathcal{C}_{l_2}^1\right) > d_{th}\right) \\
&= P\left(d\left(\mathcal{C}_{l_1}^1, \mathcal{C}_{l_2}^1\right) > d_{th} \text{ and } d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_4}^0\right) \leq d_{th}\right) + \\
&\quad P\left(d\left(\mathcal{C}_{l_1}^1, \mathcal{C}_{l_2}^1\right) > d_{th} \text{ and } d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_4}^0\right) > d_{th}\right), \\
&\leq P\left(\alpha_1 d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_2}^0\right) + \alpha_2 d\left(\mathcal{C}_{l_4}^0, \mathcal{C}_{l_2}^0\right) > d_{th}\right) + P\left(d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_4}^0\right) > d_{th}\right) \\
&\leq P\left(d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_2}^0\right) > d_{th}\right) + P\left(d\left(\mathcal{C}_{l_4}^0, \mathcal{C}_{l_2}^0\right) > d_{th}\right) + P\left(d\left(\mathcal{C}_{l_1}^0, \mathcal{C}_{l_4}^0\right) > d_{th}\right) \\
&\leq 3a_2 e^{-nb_2}.
\end{aligned} \tag{C.3}$$

The second inequality is due to $\alpha_1 + \alpha_2 = 1$ and the union bound. Assume that for some $1 \leq t_0 \leq t$,

$$P\left(d\left(\mathcal{C}_{l_1}^{t_0}, \mathcal{C}_{l_2}^{t_0}\right) > d_{th}\right) \leq 3^{t_0} a_3 e^{-nb_3}. \tag{C.4}$$

Then for $t_0 = t + 1$, we have

$$\begin{aligned}
&P\left(d\left(\mathcal{C}_{l_1}^{t+1}, \mathcal{C}_{l_2}^{t+1}\right) > d_{th}\right) \\
&\leq P\left(d\left(\mathcal{C}_{l_1}^{t}, \mathcal{C}_{l_2}^{t}\right) > d_{th}\right) + P\left(d\left(\mathcal{C}_{l_4}^{t}, \mathcal{C}_{l_2}^{t}\right) > d_{th}\right) + P\left(d\left(\mathcal{C}_{l_1}^{t}, \mathcal{C}_{l_4}^{t}\right) > d_{th}\right) \\
&\leq 3^{t+1} a_2 e^{-nb_2}.
\end{aligned}$$

The first inequality is due to the assumption $\beta \in (-1, 0)$ and (3.1) while the last two inequalities are due to the union bound and (C.4).

C.3 Proof of Theorem 3.2.1

Denote by F the event that sequences are *well separated*:

$$\begin{aligned}
F = \big\{ &d\left(\mathbf{y}_{j_1}, \mathbf{y}_{j_2}\right) \leq d_{th} \text{ and } d\left(\mathbf{y}_{j_1}, \mathbf{y}_{j_3}\right) > d_{th} : \mathbf{y}_{j_1}, \mathbf{y}_{j_2} \sim \mathcal{P}_k \\
&\text{and } \mathbf{y}_{j_3} \sim \mathcal{P}_{k'} \; \forall k, k' \in \{1, \ldots, K\}, k \neq k' \big\}.
\end{aligned}$$

Since there are M data sequences, by Assumption 2, $P(F)$ can be lower bounded by

$$\begin{aligned} P(F) &\geq \prod_{k=1}^{K} \left(1 - a_1 e^{-b_1 n}\right)^{M_k^2} \prod_{k=1}^{K} \prod_{\substack{k'=1 \\ k' \neq k}}^{K} \left(1 - a_2 e^{-b_2 n}\right)^{M_k M_{k'}} \\ &\geq \left(1 - a_1 e^{-b_1 n}\right)^{M^2} \left(1 - a_2 e^{-b_2 n}\right)^{M^2} \\ &\geq \left(1 - M^2 a_1 e^{-b_1 n}\right) \left(1 - M^2 a_2 e^{-b_2 n}\right) \\ &\geq 1 - M^2 a_1 e^{-b_1 n} - M^2 a_2 e^{-b_2 n}. \end{aligned} \tag{C.5}$$

The third inequality is due to Bernoulli's inequality. Without loss of generality, assume that $\mathcal{C}_{l_1}$ and $\mathcal{C}_{l_2}$ are merged. By Proposition 2, the distance $d(\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}, \mathcal{C}_{l_3})$ satisfies

$$\begin{aligned} d(\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}, \mathcal{C}_{l_3}) &\geq \min\left\{d(\mathcal{C}_{l_1}, \mathcal{C}_{l_3}), d(\mathcal{C}_{l_2}, \mathcal{C}_{l_3})\right\}, \\ d(\mathcal{C}_{l_1} \cup \mathcal{C}_{l_2}, \mathcal{C}_{l_3}) &\leq \max\left\{d(\mathcal{C}_{l_1}, \mathcal{C}_{l_3}), d(\mathcal{C}_{l_2}, \mathcal{C}_{l_3})\right\}. \end{aligned}$$

This implies that if data sequences are well separated, 1) the distances between clusters consisting of data sequences generated from the same $\mathcal{P}_k$ are always no greater than d_{th}; and the distances between clusters consisting of data sequences generated from $\mathcal{P}_k$ and $\mathcal{P}_{k'}$ for $k \neq k'$ are always no less than d_{th}. Hence, if data sequences are *well separated*, linkage-based clustering algorithms always provide correct clustering results. The error probability of a linkage-based clustering algorithm is then upper bounded by

$$P_e \leq 1 - P(F) \leq M^2 a_1 e^{-b_1 n} + M^2 a_2 e^{-b_2 n}.$$

C.4 Proof of Theorem 3.2.2

Define

$$F_1 = \left\{ \min_{\mathbf{y}_{j_1}, \mathbf{y}_{j_2} \sim \mathcal{P}_k} d\left(\mathbf{y}_{j_1}, \mathbf{y}_{j_2}\right) \leq d_{th} : \ \forall k \in \{1, \ldots, K\} \right\},$$
$$F_2 = \left\{ \min_{\mathbf{y}_{j_1} \sim \mathcal{P}_k,\, \mathbf{y}_{j_3} \sim \mathcal{P}_{k'}} d\left(\mathbf{y}_{j_1}, \mathbf{y}_{j_3}\right) > d_{th} : \ \forall k, k' \in \{1, \ldots, K\}, k \neq k' \right\}.$$

Similar to (C.5), we have

$$\begin{aligned} P\left(F_1 \cap F_2\right) &\geq \left(1 - a_1 e^{-b_1 n}\right)^{M^2} \left(1 - a_2 e^{-b_2 n}\right)^{M} \\ &\geq 1 - M^2 a_1 e^{-b_1 n} - M a_2 e^{-b_2 n}. \end{aligned}$$

The error probability of SLINK is thus upper bounded by

$$P_{e,S} \leq 1 - P\left(F_1 \cap F_2\right) \leq M^2 a_1 e^{-b_1 n} + M a_2 e^{-b_2 n}.$$

C.5 Proof of Theorems 3.3.1

Without loss of generality, assume that an HAC algorithm converges after T iterations, where $T \leq M$. Denote by E^t for $1 \leq t \leq T$ the event that after the t-th iteration, there exists at least one cluster that contains sequences generated from different distribution clusters. Due to the scheme of HAC algorithms, the clustering error can not be corrected in the following iterations. Hence, $E^1 \subset E^2 \ldots \subset E^T$. Denote by H^t the event that after the t-th iteration, there exists at least two clusters that contain sequences generated from the same distribution clusters. Then $H^1 \supset H^2 \ldots \supset H^T$. Moreover, since a clustering algorithm provides an incorrect clustering result if and only if either E^T or H^T happens, i.e., the error event is $E^T \cup H^T$, then the error probability of the HAC algorithm is given by

$$P_e = P\left(E^T \cup H^T\right). \tag{C.6}$$

Denote by E^0 the event that each cluster has exactly one data sequence. Define for $1 \leq t \leq T$, $\hat{E}^t = E^t \setminus E^{t-1}$ and $\hat{H}^T = H^T \setminus E^T$. Then (C.6) becomes

$$P_e = P\left(\left(\cup_{t=1}^{T} \hat{E}^t\right) \cup \hat{H}^T\right) \leq \sum_{t=1}^{T} P\left(\hat{E}^t\right) + P\left(\hat{H}^T\right), \tag{C.7}$$

where the inequality is due to the union bound. We then try to bound $P\left(\hat{E}^t\right)$ and $P\left(\hat{H}^T\right)$. Since after each iteration, two clusters are combined, then the number of clusters after t iterations is $M - t$. Denote by $\mathcal{C}_1^t, \ldots, \mathcal{C}_{M-t}^t$ the $M - t$ clusters. Since

$$\hat{E}^t = \Big\{ d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_2}^{t-1}\right) \leq d_{th} : \ \mathcal{C}_{l_1}^{t-1} \sim \mathcal{P}_k, \ \mathcal{C}_{l_2}^{t-1} \sim \mathcal{P}_{k'}, \ k \neq k', \ (l_1, \ l_2) = \\ \arg \min_{l, l' \in I_1^{M-t+1}, \ l \neq l'} d\left(\mathcal{C}_l^{t-1}, \mathcal{C}_{l'}^{t-1}\right), \ \forall l \in I_1^{M-t+1}, \ \mathcal{C}_l^{t-1} \sim \mathcal{P}_k \text{ for some } k \in I_1^K \Big\},$$

where $I_{k_1}^{k_2}$ denotes the integer set from k_1 to k_2, then for centroid-based clustering algorithms,

$$\hat{E}^t \subset \cup_{l_1, l_2, l_3 \in I_1^{M-t+1}} \big\{ d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_2}^{t-1}\right) \geq d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_3}^{t-1}\right) : \ \mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_2}^{t-1} \sim \mathcal{P}_k, \mathcal{C}_{l_3}^{t-1} \sim \mathcal{P}_{k'}, \\ k, k' \in I_1^K, \ k \neq k' \big\}.$$

Thus, by the union bound and (3.10a), we have for sufficiently large n,

$$\begin{aligned} & P\left(\hat{E}^t\right) \\ \leq & \sum_{l_1=1}^{M-t+1} \sum_{\substack{l_2=1 \\ l_2 \neq l_1}}^{M-t+1} \sum_{\substack{l_3=1 \\ l_3 \neq l_2 \\ l_3 \neq l_1}}^{M-t+1} P\big(\big\{ d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_2}^{t-1}\right) \geq d\left(\mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_3}^{t-1}\right) : \\ & \mathcal{C}_{l_1}^{t-1}, \mathcal{C}_{l_2} \sim \mathcal{P}_k, \ \mathcal{C}_{l_3}^{t-1} \sim \mathcal{P}_{k'}, \ k, k' \in I_1^K, \ k \neq k' \big\} \big) \\ \leq & \, 2^t M^3 a_3 e^{-n b_3}. \end{aligned} \tag{C.8}$$

Furthermore, since

$$\begin{aligned}\hat{H}^T = \Bigg\{ & \min_{l,l' \in I_1^{M-T},\, l \neq l'} d\left(\mathcal{C}_l^T, \mathcal{C}_{l'}^T\right) > d_{th}, \text{ and } \forall l \in I_1^{M-T},\ \mathcal{C}_l^T \sim \mathcal{P}_k \text{ for some } k \in I_1^K, \\ & \text{and } \exists\ \mathcal{C}_{l_1}^T,\ \mathcal{C}_{l_2}^T \sim \mathcal{P}_{k'} \text{ for } k' \in I_1^K \Bigg\} \\ \subset & \cup_{l,l' \in I_1^{M-T}} \left\{ d\left(\mathcal{C}_l^T, \mathcal{C}_{l'}^T\right) > d_{th} :\ \mathcal{C}_l^T, \mathcal{C}_{l'}^T \sim \mathcal{P}_k \right\},\end{aligned}$$

then for centroid-based clustering algorithms with sufficiently large n, $P\left(\hat{H}^T\right)$ is upper bounded by

$$P\left(\hat{H}^T\right) \leq 3^T M^2 a_2 e^{-nb_2}. \tag{C.9}$$

By (C.7) - (C.9), the upper bound on the error probability of linkage-based clustering algorithms and centroid-based clustering algorithms for n sufficiently large is given by the following two inequalities, respectively,

$$\begin{aligned}P_e &\leq \sum_{t=1}^{T} 2^t M^3 a_3 e^{-nb_3} + 3^T M^2 a_2 e^{-nb_2}, \\ &\leq M^2 \left[2^{M+1} M a_3 e^{-nb_3} + 3^M a_2 e^{-nb_2}\right].\end{aligned}$$

BIBLIOGRAPHY

[1] Y. Sakurai, L. Li, R. Chong, and C. Faloutsos, "Efficient distribution mining and classification," in *Proc. SIAM int. conf. data mining*, Atlanta, Georgia, USA, Apr. 2008, pp. 632–643.

[2] E. Spellman, B. C. Vemuri, and M. Rao, "Using the KL-center for efficient and accurate retrieval of distributions arising from texture images," in *Proc. IEEE Conf. Comput. Vision, Pattern Recognition (CVPR)*, vol. 1, San Diego, CA, USA, June 2005, pp. 111–116.

[3] C. Lin, C. Chen, H. Lee, and J. Liao, "Fast k-means algorithm based on a level histogram for image retrieval," *Expert Syst. Applicat.*, vol. 41, no. 7, pp. 3276 – 3283, 2014.

[4] M. Vrac, L. Billard, E. Diday, and A. Chédin, "Copula analysis of mixture models," *Computational Stat.*, vol. 27, no. 3, pp. 427–457, 2012.

[5] R. Moreno-Sáez, M. S. de Cardona, and L. Mora-López, "Data mining and statistical techniques for characterizing the performance of thin-film photovoltaic modules," *Expert Syst. Applicat.*, vol. 40, no. 17, pp. 7141 – 7150, 2013.

[6] R. Moreno-Sáez and L. Mora-López, "Modelling the distribution of solar spectral irradiance using data mining techniques," *Environmental Modelling, Software*, vol. 53, pp. 163 – 172, 2014.

[7] W. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *J. classification*, vol. 1, no. 1, pp. 7–24, 1984.

[8] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, "Efficient agglomerative hierarchical clustering," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2785–2797, 2015.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[10] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge: Cambridge University Press, 2012.

[11] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–137, Mar. 1982.

[12] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *KDD Workshop Text Mining*, 2000.

[13] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Trans. Pattern Anal., Mach. Intell.*, vol. 24, pp. 881–892, Jul. 2002.

[14] L. Kaufman and P. Rousseeuw, "Clustering by means of medoids," *Statistical data anal. based on the L1-norm and related methods*, pp. 405–416, 1987.

[15] M. Laan, K. Pollard, and J. Bryan, "A new partitioning around medoids algorithm," *J. Statistical Computation and Simulation*, vol. 73, no. 8, pp. 575–584, 2003.

[16] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert syst. applicat.*, vol. 36, no. 2, pp. 3336–3341, 2009.

[17] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[18] P. Macnaughton-Smith, W. T. Williams, M. B. Dale, and L. Mockett, "Dissimilarity analysis: a new technique of hierarchical sub-division," *Nature*, vol. 202, no. 4936, p. 1034, 1964.

[19] M. Chavent, Y. Lechevallier, and O. Briant, "Divclus-t: A monothetic divisive hierarchical clustering method," *Computational Stat. & Data Anal.*, vol. 52, no. 2, pp. 687–701, 2007.

[20] I. Katsavounidis, C. C. J. Kuo, and Z. Zhang, "A new initialization technique for generalized Lloyd iteration," *IEEE Signal Process. Lett.*, vol. 1, pp. 144–146, Oct. 1994.

[21] T. Wang, D. J. Bucci, Y. Liang, B. Chen, and P. K. Varshney, "Exponentially consistent k-means clustering algorithm based on Kolmogrov-Smirnov test," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Calgary, Canada, Apr. 2018, pp. 2296–2300.

[22] ——, "Clustering under composite generative models," in *Proc. Annu. Conf. Inform. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 2018, pp. 338–343.

[23] L. Mora-López and J. Mora, "An adaptive algorithm for clustering cumulative probability distribution functions using the Kolmogorov-Smirnov two-sample test," *Expert Syst. Applicat.*, vol. 42, pp. 4016 – 4021, 2015.

[24] R. T. Ng and J. Han, "Clarans: a method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep 2002.

[25] L. Kaufman and P. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. NJ: John Wiley & Sons, 2009, vol. 344.

[26] T. Velmurugan and T. Santhanam, "Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points," *J. Comput. Sci.*, vol. 6, no. 3, p. 363, 2010.

[27] W. Sheng and X. Liu, "A genetic k-medoids clustering algorithm," *J. Heuristics*, vol. 12, no. 6, pp. 447–466, Dec 2006.

[28] J. C. Gower and G. J. S. Ross, "Minimum spanning trees and single linkage cluster analysis," *J. Royal Stat. Society. Series C (Applied Stat.)*, vol. 18, no. 1, pp. 54–64, 1969.

[29] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[30] R. R. Sokal, "A statistical method for evaluating systematic relationship," *University of Kansas science bulletin*, vol. 28, pp. 1409–1438, 1958.

[31] P. H. Sneath, R. R. Sokal *et al.*, *Numerical taxonomy. The principles and practice of numerical classification.* San Francisco: W. H. Freeman, 1973.

[32] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies: 1. hierarchical systems," *Comput. J*, vol. 9, no. 4, pp. 373–380, 1967.

[33] P. Borysov, J. Hannig, and J. Marron, "Asymptotics of hierarchical clustering for growing dimension," *J. Multivariate Analysis*, vol. 124, pp. 465–479, 2014.

[34] J. Banks, C. Moore, R. Vershynin, N. Verzelen, and J. Xu, "Information-theoretic bounds and phase transitions in clustering, sparse pca, and submatrix localization," *IEEE Trans. Inform. Theory*, vol. 64, no. 7, pp. 4872–4894, 2018.

[35] A. Khaleghi, D. Ryabko, J. Mary, and P. Preux, "Consistent algorithms for clustering time series," *J Machine Learning Research*, vol. 17, no. 1, pp. 94–125, 2016.

[36] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu, "Hierarchical clustering: Objective functions and algorithms," *J. ACM*, vol. 66, no. 4, pp. 1–42, 2019.

[37] S. Dasgupta, "A cost function for similarity-based hierarchical clustering," in *Proc.the forty-eighth annual ACM symposium on Theory of Computing*, Cambridge, MA, USA, June 2016, pp. 118–127.

[38] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics, intelligent laboratory syst.*, vol. 2, no. 1-3, pp. 37–52, 1987.

[39] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York: John Wiley & Sons, 2012.

[40] E. Bair, T. Hastie, D. Paul, and R. Tibshirani, "Prediction by supervised principal components," *J. American Statistical Association*, vol. 101, no. 473, pp. 119–137, 2006.

[41] J. Piironen and A. Vehtari, "Iterative supervised principal components," *arXiv preprint arXiv:1710.06229*, 2017.

[42] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Proc. IEEE signal process. society workshop*. Madison, WI, USA: IEEE, Aug. 1999, pp. 41–48.

[43] K. Fukunaga, *Introduction to statistical pattern recognition*. San Diego, CA, USA: Elsevier, 2013.

[44] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proc. int. conf. Machine learning*, Pittsburgh, Pennsylvania, USA, June 2006, pp. 905–912.

[45] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Z. Jahromi, "Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds," *Pattern Recognition*, vol. 44, no. 7, pp. 1357–1371, 2011.

[46] Y. Wang, C. Chen, V. Watkins, and K. Ricanek, "Modified supervised kernel pca for gender classification," in *Int. Conf. Intelligent Sci. Big Data Eng.* Suzhou, China: Springer, June 2015, pp. 60–71.

[47] Z. Ma, Z. Zhan, X. Ouyang, and X. Su, "Nonlinear dimensionality reduction based on hsic maximization," *IEEE Access*, vol. 6, pp. 55 537–55 555, 2018.

[48] H. Wu, D. M. Bowers, T. T. Huynh, and R. Souvenir, "Biomedical video denoising using supervised manifold learning," in *IEEE 10th Int. Symp. Biomedical Imaging*. San Francisco, CA, USA: IEEE, Apr. 2013, pp. 1244–1247.

[49] A. Ritchie, C. Scott, L. Balzano, D. Kessler, and C. S. Sripada, "Supervised principal component analysis via manifold optimization," in *Proc. IEEE Data Sci. Workshop (DSW)*, Minneapolis, MN, USA, June 2019.

[50] L. Abdi and A. Ghodsi, "Discriminant component analysis via distance correlation maximization," *Pattern Recognition*, vol. 98, p. 107052, 2020.

[51] S. Raamadhurai, "Supervised probability preserving projection (sppp)," Master's thesis, School of Science, Aalto University, Espoo, Finland, 2014.

[52] S. Jahan, "Supervised distance preserving projection using alternating direction method of multipliers," *J. Industrial & Management Optimization*, vol. 13, no. 5, p. 1, 2019.

[53] Z. Zhu, T. Similä, and F. Corona, "Supervised distance preserving projections," *Neural process. lett.*, vol. 38, no. 3, pp. 445–463, 2013.

[54] T. Ghosh and M. Kirby, "Supervised dimensionality reduction and visualization using centroid-encoder," *arXiv preprint arXiv:2002.11934*, 2020.

[55] Y. Liu, T. Wang, Y. Jiang, and B. Chen, "Harvesting ambient rf for presence detection through deep learning," *arXiv preprint arXiv:2002.05770*, 2020.

[56] W. Rudin, *Real and Complex Analysis*. Singapore: McGraw-Hill, Inc., 1987.

[57] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf, "Kernel measures of conditional dependence," in *Proc. Advances Neural Inform. Process. Syst. (NIPS)*, Vancouver, Canada, Dec. 2008, pp. 489–496.

[58] B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf, "Injective Hilbert space embeddings of probability measures," in *Proc. Annu. Conf. Learning Theory (COLT)*, Helsinki, Finland, 2008, pp. 111–122.

[59] K. Fukumizu, B. Sriperumbudur, A. Gretton, and B. Schölkopf, "Characteristic kernels on groups and semigroups," in *Proc. Advances Neural Inform. Process. Syst. (NIPS)*, Vancouver, Canada, Dec. 2009, pp. 473–480.

[60] B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf, "Hilbert space embeddings and metrics on probability measures," *J. Mach. Learning Research*, vol. 11, pp. 1517–1561, 2010.

[61] L. Song, A. Gretton, and K. Fukumizu, "Kernel embeddings of conditional distributions," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 98–111, 2013.

[62] A. Smola, A. Gretton, L. Song, and B. Schölkopf, "A Hilbert space embedding for distributions," in *Proc. Int. Conf. Algorithmic Learning Theory*, Sendai, Japan, Oct. 2007, pp. 13–31.

[63] S. Zou, Y. Liang, H. V. Poor, and X. Shi, "Nonparametric detection of anomalous data streams," *IEEE Trans. Signal Process.*, vol. 65, pp. 5785–5797, Nov 2017.

[64] S. Zou, Y. Liang, and H. V. Poor, "Nonparametric detection of geometric structures over networks," *IEEE Trans. Signal Process.*, vol. 65, pp. 5034–5046, June 2017.

[65] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learning Research*, vol. 13, pp. 723–773, 2012.

[66] S. Zhu, B. Chen, Z. Chen, and P. Yang, "Asymptotically optimal one- and two-sample testing with kernels," *submitted to IEEE Trans. Information Theory*, 2019.

[67] T. Wang, Q. Li, D. Bucci, Y. Liang, B. Chen, and P. Varshney, "K-medoids clustering of data sequences with composite distributions," *IEEE Trans. Signal Process*, vol. 67, no. 8, pp. 2093–2106, 2019.

[68] J. L. Alperin, *Local representation theory: Modular representations as an introduction to the local representation theory of finite groups*. Cambridge University Press, 1993, vol. 11.

[69] P. Massart, "The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality," *Ann. Probability*, vol. 18, pp. 1269–1283, 1990.

[70] Q. Li, T. Wang, D. J. Bucci, Y. Liang, B. Chen, and P. K. Varshney, "Nonparametric composite hypothesis testing in an asymptotic regime," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 5, pp. 1005–1014, Oct 2018.

VITA

NAME OF AUTHOR: Tiexing Wang

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

Syracuse University, Syracuse, NY, USA, 2013-2020

Beijing Institute of Technology, Beijing, China, 2006-2010

DEGREES AWARDED:

B.E., 2010, Beijing Institute of Technology, Beijing, China

PUBLICATIONS:

Y.Liu, **T. Wang**, Y. Jiang, and B. Chen, "Harvesting Ambient RF for Presence Detection Through Deep Learning", *to appear in IEEE Trans. Neural Networks, Learning Syst.*.

T. Wang, Y. Liu, and B. Chen, "On Exponentially Consistency of Linkage-based Hierarchical Clustering Algorithm Using Kolmogrov-Smirnove Distance", *Proc. IEEE ICASSP*, Barcelona, Spain, May, 2020.

T. Wang, Q. Li, D. Bucci, B. Chen, Y. Liang and P. Varshney. "K-medoids Clustering of Data Sequences with Composite Distributions", *IEEE Trans. Signal Process.* Apr. 2019.

Q. Li **T. Wang**, D. Bucci, B. Chen, Y. Liang and P. Varshney. "Nonparametric Composite Hypothesis Testing in an Asymptotic Regime", *IEEE J. Selected Topics Signal Process.* 2018.

T. Wang, D. Bucci, Y. Liang, B. Chen, and P. Varshney, "Exponentially consistent K-means clustering algorithm based on Kolmogrov-Smirnov test", *Proc. IEEE ICASSP*, Calgary, AB, Canada, Apr. 2018.

T. Wang, D. Bucci, Y. Liang, B. Chen, and P. Varshney, "Clustering under composite generative models", *Proc. IEEE CISS*, Princeton, NJ, Mar. 2018.

T. Wang, F. Peng, and B. Chen, "Autonomous Localization and Mapping Using a Single Mobile Device", *arXiv:1612.05793[cs.RO]*, Dec. 2016

T. Wang, F. Peng, and B. Chen, "First order echo based room shape recovery using a single mobile device", *Proc. IEEE ICASSP*, Shanghai, China, Mar. 2016.

F. Peng, **T. Wang**, and B. Chen, "Room shape reconstruction with a single mobile acoustic sensor," *Proc. IEEE GlobalSIP*, Orlando, FL, Dec. 2015.