DISPARATE INFORMATION FUSION IN THE DISSIMILARITY FRAMEWORK

by

Zhiliang Ma

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

October, 2010

Abstract

We study the problem of combining multiple disparate types of data to improve the performances in various inferential tasks, and we propose the dissimilarity framework, which contains two steps: (1) calculate one or more dissimilarity matrices for each data source; and (2) combine all the dissimilarity matrices for the inferential purpose. In the first step, we take advantage of the knowledge of experts in each area, and unify disparate types of data into the dissimilarity space. In this dissertation, we focus on developing methods for combining multiple dissimilarity matrices.

One of the most widely used approach for using dissimilarity data involves converting the dissimilarity matrix into a configuration of points (called the embedding) through multidimensional scaling, and then building statistical models based on the embedding. To use later collected observations, called the out-of-sample data, one could re-do the embedding and modeling process, but it is not efficient. We study the alternative of out-of-sample embedding, and develop the out-of-sample embedding approach, OOSIM, to insert the out-of-sample objects into the existing embedding by minimizing sum of squared differences between dissimilarities and the corresponding Euclidean distances. Iterative majorization

is used to minimize the criterion function. The simulation experiment suggests that OOSIM is a natural extension to de Leeuw's multidimensional scaling procedure, SMACOF, which minimizes the raw stress.

We develop the J-function approach to combine multiple dissimilarity matrices in the space of the Cartesian product of the embeddings. Due to the high dimensionality of this space, we introduce a novel supervised dimensionality reduction method. The simulation and real data results show that our approach can improve classification accuracy compared to the alternatives of principal components analysis and no dimensionality reduction at all.

We also consider information fusion from a different perspective. Suppose that objects are measured under multiple conditions—e.g., indoor lighting versus outdoor lighting for face recognition, multiple language translation for document matching, etc.—the challenging task is to perform data fusion and utilize all the available information for inferential purposes. We consider two exploitation tasks: (1) how to determine whether a set of feature vectors represent a single object measured under different conditions; and (2) how to create a classifier based on training data collected under one condition in order to classify objects measured in other conditions. The key to both problems is to transform all sets of feature vectors into one commensurate space, where the (transformed) feature vectors are comparable and would be treated as if they were collected under the same condition. Toward this end, we study Procrustes analysis and develop a new approach. We illustrate our methodology on English and French documents collected from Wikipedia, demonstrating superior performance compared to that obtained via standard Procrustes transformation.

ABSTRACT

We introduce a way to generate a collection of 3D shapes of different groups, and study the problem of combining multiple dissimilarity matrices derived from the same set of shapes for classification purpose. Experiment results show that different dissimilarity measures may capture different aspects of information and consequently combining all the dissimilarity matrices in an optimal way results in a higher classification accuracy than using each single dissimilarity matrix alone.

Advisor: Dr. Carey E. Priebe

Readers: Dr. Carey E. Priebe and Dr. David J. Marchette

Acknowledgments

First, I am grateful to my advisor, Professor Carey E. Priebe. This dissertation would not have been possible without the expert guidance of Professor Priebe. The knowledge I have gained during all these years of working under his supervision is invaluable. It has been a great experience for me to be a student of his. I owe a lot of gratitude to Professor Priebe for his guidance, encouragement, and his patience and kindness during this period of time.

I am thankful to my dissertation committee—Professor Donniell E. Fishkind, Dr. David J. Marchette and Professor Nam Lee—for their interest in my research, reading and providing valuable comments on earlier versions of this dissertation. Dr. Marchette has been extremely supportive over the years, from the Ph.D. candidacy exam in 2007, the graduate board oral examination in 2008, to the dissertation defense in 2010. Dr. Marchette is a researcher in Naval Surface Warfare Center in Dahlgren, VA. It was extremely nice of him to come up and be a part of my dissertation defense. I would also like to thank Professor Daniel Q. Naiman for being a part of my Ph.D. candidacy exam committee, and thank Professors Sanjeev P. Khudanpur, Brian Caffo and Robert P.W. Duin for being a part of my

graduate board oral examination committee. Many thanks to Professor Michael W. Trosset for providing me with a number of helpful suggestions on my research. My thanks also go to Younger Park, who has helped me enormously when I was learning R.

I also would like to thank Adam Cardinal-Stakenas, Elizabeth Beer, Ming Ye, Xugang Ye, Ting Yang, Jun Ma and Kan Jiang for being my classmates or officemates. Collaborating or studying with them has helped me learn more from my coursework or research.

Finally, I am extremely grateful to my parents, for their dedication to me and my brother. I am thankful to my elder brother, Xinli Ma, for his love, support and encouragement. At last but not least, I would like to thank my dearest wife, Haitang, for her love, understanding, and faith in me. She was always there backing me up, and stood by me through the good times and bad.

Dedication

This thesis is dedicated to my parents, for their unconditional love and support.

Contents

List of Tables

List of Figures

Introduction

In the present era, massive amount of data are being generated or collected—especially on the Internet. Google VP Marissa Mayer mentioned in her presentation at PARC 2009 entitled "The Physics of Data" that there were 5 exabytes of data online in 2002, which had risen to 281 exabytes in 2009 (1 exabytes = 10^{18} bytes = 1 billion gigabytes). That's a growth rate of 56 times over seven years. These data are of disparate types, such as text, image, audio and/or video, social network, location (longitude and latitude), etc. The challenge problem is to fuse all the available information to achieve superior performances in inferential tasks than using each single source of data alone. Information fusion has been a hot research field with various applications [1–6].

In general, the most often used information fusion approaches can be summarized into two categories: feature level fusion and decision level fusion. In the feature level fusion, feature vectors extracted from different data sources are combined into the Cartesian product space [1, 3]. The decision level fusion involves combining results obtained separately from all data sources. An ensemble of classifiers is one such example, as is track fusion [7]. The advantage of these two types of information fusion stems from the fact that multiple

sets of feature vectors extracted from the same set of objects usually reflect different characteristics of patterns. By fusing multiple data sources, one obtains a more comprehensive representation of the space in which the objects live, and hence has more information for inferential tasks such as estimation, hypothesis testing, classification, etc.

Traditional feature level fusion approaches become inadequate in learning from multiple *disparate* types of data, such as text documents, images and graphs. These complex data often result in a high-dimensional Cartesian product, which usually suffers from the "curse of dimensionality," the phenomenon that the number of data points needed to learn a classifier increases exponentially with the dimension of the representation space [8, 9]. The decision level fusion approaches are often suboptimal because, at least in principle, the joint distribution usually provides more information than the product of the marginals.

We propose a novel framework based on the dissimilarity representation for fusing multiple disparate types of data: (i) compute one or more interpoint dissimilarity matrices for each source of data; and then (ii) achieve information fusion by combining all resulted dissimilarity matrices. An obvious advantage of this framework is that we can utilize the knowledge of the experts in each domain to develop dissimilarity matrices, and achieve fusion in the unified dissimilarity space. The challenge is how to combine the multiple dissimilarity matrices in a way that is beneficial for inferential purposes. In this dissertation, we investigate the dissimilarity representation and introduce some constructive methods to properly combine multiple dissimilarity matrices.

In Chapter 1, we present definitions and background on the dissimilarity representation,

as well as theoretical foundations of dissimilarity analysis for statistical pattern recognition.

In Chapter 2, we briefly review the multidimensional scaling techniques, which are widely used to obtain a feature representation from a dissimilarity matrix. We also study the out-of-sample embedding extension for classical multidimensional scaling and introduce a novel out-of-sample embedding algorithm—OOSIM (out-of-sample embedding by iterative majorization).

In Chapter 3, we introduce an approach of combining dissimilarity matrices in the Cartesian product space, along with a supervised dimensional reduction method.

In Chapter 4, we study information fusion from a different perspective and introduce a method on fusion and inference from multiple data sources in a commensurate space.

In Chapter 5, we give a case study about fusion in the dissimilarity framework, including a method of generating a collection of phantom shapes of multiple groups, procedures of obtaining multiple dissimilarity matrices from the same set of shapes, and classification studies based each dissimilarity matrix and fusions of them.

In Chapter 6, we summarize the main achievements of this dissertation, and discus open problems and directions for future work.

Chapter 1

Dissimilarity Representation and Analysis

1.1 Definitions

Definition 1.1.1. A **metric** on a set $\mathcal{X} \subset \mathbb{R}^p$ $(p \in \mathbb{Z}_+)$ is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+ \cup \{0\}$. For all $\boldsymbol{x},\ \boldsymbol{y},\ \boldsymbol{z} \in \mathcal{X}$, this function d, usually called the distance function or simply **distance**, is required to satisfy the following conditions:

1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
2. $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
4. $d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$ (triangle inequality)

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, some examples of metrics are:

1. The *discrete metric*: if $\boldsymbol{x} = \boldsymbol{y}$ then $d(\boldsymbol{x}, \boldsymbol{y}) = 0$. Otherwise, $d(\boldsymbol{x}, \boldsymbol{y}) = 1$.

2. The *Euclidean distance*: $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2$.

3. The *Manhattan distance*:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{p} |x_i - y_i|, \ (\boldsymbol{x} = [x_1, \ldots, x_p]^t, \boldsymbol{y} = [y_1, \ldots, y_p]^t).$$

4. The *Minkowski metric*:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{p} |x_i - y_i|^p \right)^{1/p}, \ p \geq 1, \ p \neq 2.$$

5. The *Mahalanobis distance*:

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^t \mathbf{C}^{-1} (\boldsymbol{x} - \boldsymbol{y})}, \ \mathbf{C} \text{ is positive-semidefinite.}$$

A dissimilarity measure is a metric without the requirement for satisfying the triangle inequality. Formally,

Definition 1.1.2. a **dissimilarity measure** on a set $\mathcal{X} \subset \Xi$ is a function $\delta : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+ \cup \{0\}$. For all $\boldsymbol{x}, \ \boldsymbol{y}, \ \boldsymbol{z} \in \mathcal{X}$, the dissimilarity measure δ is required to satisfy the following conditions:

1. $\delta(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
2. $\delta(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$ (identity of indiscernibles)
3. $\delta(\boldsymbol{x}, \boldsymbol{y}) = \delta(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)

Notice that in most cases $\Xi = \mathbb{R}^p$. However we wish to leave open the possibility for applications where the original data are infinite dimensional, graph-valued, or occupying some other nonstandard space. A **pseudo-dissimilarity** measure is a dissimilarity measure that allows $\delta(\boldsymbol{x}, \boldsymbol{y}) = 0$ for $\boldsymbol{x} \neq \boldsymbol{y}$, though we do not make a distinction between pseudo-dissimilarity and dissimilarity in this dissertation.

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, some examples of dissimilarity measures are:

1. The *cosine dissimilarity*:
$$\delta(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\boldsymbol{x}^t \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|}.$$

2. The *correlation measure*:
$$\delta(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{(\boldsymbol{x} - \bar{\boldsymbol{x}})^t (\boldsymbol{y} - \bar{\boldsymbol{y}})}{\|\boldsymbol{x} - \bar{\boldsymbol{x}}\| \|\boldsymbol{y} - \bar{\boldsymbol{y}}\|}.$$

3. The *divergence*:
$$\delta(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{p} \sum_{i=1}^{p} \frac{(x_i - y_i)^2}{(x_i + y_i)^2}.$$

4. *Soergel*:
$$\frac{\sum_{i=1}^{p} |x_i - y_i|}{\sum_{i=1}^{p} \max(x_i, y_i)}.$$

In some applications, the dissimilarities between two objects are not necessarily symmetric. That is, $\delta(\boldsymbol{x}, \boldsymbol{y}) \neq \delta(\boldsymbol{y}, \boldsymbol{x})$. In these scenarios, we can define a new symmetric dissimilarity measure δ' from the given dissimilarities. Some potential methods are:

1. $\delta'(\boldsymbol{x}, \boldsymbol{y}) = \big(\delta(\boldsymbol{x}, \boldsymbol{y}) + \delta(\boldsymbol{y}, \boldsymbol{x})\big)/2$

2. $\delta'(\boldsymbol{x}, \boldsymbol{y}) = \max\{\delta(\boldsymbol{x}, \boldsymbol{y}), \delta(\boldsymbol{y}, \boldsymbol{x})\}$

3. $\delta'(\boldsymbol{x}, \boldsymbol{y}) = \min\{\delta(\boldsymbol{x}, \boldsymbol{y}), \delta(\boldsymbol{y}, \boldsymbol{x})\}$

4. $\delta'(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\delta(\boldsymbol{x}, \boldsymbol{y})^2 + \delta(\boldsymbol{y}, \boldsymbol{x})^2}$

Definition 1.1.3. A **dissimilarity representation** for a set of n objects is expressed as a symmetric and non-negative matrix Δ, whose diagonal elements are all equal to zero. The matrix Δ is obtained by applying a dissimilarity measure δ on every pair of objects.

Most traditional statistical pattern recognition techniques rely on objects represented by points in a feature (vector) space. In this space, classifiers are developed to best separate the objects of different classes. As an alternative to the feature-based representation, the dissimilarity representation describes objects by their interpoint comparisons. The dissimilarity representation has attracted substantial interest in various areas [10–13].

1.2 Foundations of the Dissimilarity Representation

Let X, X_1, X_2 be three multivariate random variables that are independent and identically distributed as F, and Y, Y_1, Y_2 be three multivariate random variables that are independent and identically distributed as G. For a univariate function h, $h : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}_+ \cup \{0\}$ and $h(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$, Maa, Pearl and BartoszyńSki [14] show that

$$h(X, X_1) =_{\mathscr{L}} h(Y, Y_1) \text{ and } h(X, X_1) =_{\mathscr{L}} h(X, Y)$$

if and only if

$$F = G,$$

where the symbol $=_{\mathscr{L}}$ stands for the equality of distributions. That is, the equality of the three interpoint comparison distributions is equivalent to the equality of the two multivariate distributions.

In the next section, we study the foundation of using dissimilarity representation in statistical pattern recognition. For concreteness, we consider the inferential task to be classification.

1.3 A Probabilistic Foundation of Dissimilarity-Based Classification

Let (X, Y) be a random pair distributed as F_{XY}, where X is $\mathbb{R}^p$-valued and Y is (say) $\{0, 1\}$-valued. More specifically, X is a random vector representing an observation, and Y is a binary random variable representing its class label. Devroye et al. [15] describes the framework of statistical pattern recognition based on the feature representation. A classifier is a function $g(\cdot) : \mathbb{R}^p \to \{0, 1\}$. An error occurs if $g(X) \neq Y$ and the probability of error for g is defined by $L(g) = P\{g(X) \neq Y\}$. The best possible classifier is the Bayes rule, g^*, which is defined by

$$g^* = \underset{g:\mathbb{R}^d \to \{0,1\}}{\operatorname{argmin}} P\{g(X) \neq Y\}.$$

The corresponding probability of error is called the Bayes error and denoted by L^*. Consider (training) data $\mathcal{T}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, a sequence of random pairs that are independent and identically distributed (*i.i.d*) as F_{XY}. A classifier constructed on the basis of $\mathcal{T}_n$ is denoted by $g_n(\cdot\,; \mathcal{T}_n)$ and its performance is measured by $L_n = L(g_n) = P\{g_n(X; \mathcal{T}_n) \neq Y\}$.

Suppose one does not have access to the data $\mathcal{T}_n$, but the pairwise dissimilarities of $\mathcal{T}_n$ are available, represented by an $n \times n$ dissimilarity matrix Δ_n. We are concerned with constructing a framework to govern pattern classification of data represented only by dissimilarities. Duin et al. [16] describe a such a framework using the class of polynomial classifiers. We expand on this work and present a general context for dissimilarity-based pattern classification.

1.3.1 Dissimilarity Based Classifiers

Consider dissimilarity measure $\delta : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^+ \cup \{0\}$ (the set of nonnegative real numbers). Let $\mathcal{D} = \{[\delta(X, X')] : X, X' \in \mathbb{R}^p\}$ be the space of dissimilarity matrices. Let $\Delta \in \mathcal{D}$, then Δ is square, the entries of Δ are real-valued and nonnegative, $\Delta = \Delta^t$, and the diagonal elements of Δ are all 0s. Notice that in certain applications, some of these conditions may be relaxed; and in other cases, we may wish to impose more restrictions on $\Delta(i, j)$, for example that $\Delta(i, j) = 0$ if and only if $i = j$. The dissimilarity matrix for the training data is given by $\Delta_n = [\delta_{ij} = \delta(X_i, X_j) : \ X_i, X_j \in \mathcal{T}_n] \in \mathcal{D}_n$, where $\mathcal{D}_n$ is the space of $n \times n$ dissimilarity matrices. We may think of Δ as a $\mathcal{D}_n$-valued random variable,

however, because of the dependency between the dissimilarities, the joint distribution of Δ may be difficult to obtain.

We are concerned with dissimilarity-based classifiers $g : \mathcal{D}_{1,n} \to \{0, 1\}$, where $\mathcal{D}_{1,n}$ denotes the space of the dissimilarities between X and training data $\mathcal{T}_n$. The classification error of g is then defined by $L(g) = P\{g(d_X) \neq Y\}$, where $d_X \in \mathcal{D}_{1,n}$.

The classifiers constructed on the bases of (training) dissimilarity matrix Δ_n are of the form

$$g : \mathcal{D}_{n+1} \times \{0, 1\}^n \to \{0, 1\}, \text{ or}$$

$$g : \mathcal{D}_{1,n} \times \left(\mathcal{D}_n \times \{0, 1\}^n\right) \to \{0, 1\}.$$

Consequently, the classification error $L(g)$ becomes

$$L\left(g_n(d_X)\right) = P\left\{g_n(d_X; \Delta_n) \neq Y | \Delta_n\right\}.$$

1.3.1.1 The Best Classifier g^*_δ for (F_{XY}, δ) and Its Error L^*_δ

Consider the dissimilarity measure δ and a fixed observation x. Let $\delta(x, X')|Y' = j$ be the (random) dissimilarities between x and a random observation X' of class j, $j \in \{0, 1\}$. Let $h_{x,j}$ be the probability density function of the random variable $\delta(x, X')|Y' = j$, and let $\mathcal{H}_j = \{h_{x,j} : x \in \mathbb{R}^p\}$ be the space of all such probability density functions. Then the best possible classifier g^* based on dissimilarities is defined by

$$g^*_\delta = \inf_{g \in \mathcal{G}} P[g(h_{X,0}, h_{X,1}) \neq Y], \tag{1.1}$$

where $\mathcal{G} = \{g : \mathcal{H}_0 \times \mathcal{H}_1 \to \{0, 1\}\}$, Y is the true class label of X. The minimal probability of error for dissimilarity is defined by $L^*_\delta = P[g^*_\delta(X) \neq Y]$.

1.3.2 Relation Between L_δ^* and L^*

1.3.2.1 $L_\delta^* \geq L^*$

It is known that data processing destroys information [15, Problem 2.1]. Applying a dissimilarity measure on observations to generate a dissimilarity representation is a data processing procedure. Hence

$$L_\delta^* \geq L^*.$$

1.3.2.2 $L_\delta^* = L^* + \epsilon(F_{XY}, \delta)$

It is clear that the probability of classification error L_δ^* depends on the dissimilarity measure δ, because two different dissimilarity measures—e.g. the discrete distance and the Euclidean distance—result in different dissimilarity representations and hence leads to different performance in classification.

We illustrate by an example that L_δ^* depends on the distribution F_{XY}, too. Consider a two-class classification problem. Consider a family of joint distributions of (X, Y):

$$\mathcal{F} = \{F_{XY} : Y \sim \text{Bernoulli}(\pi);\ F_j = F_{X|Y=j} = \text{Uniform}\left(B((-1)^{j+1}, r)\right)\},$$

where $B(c, r)$ denotes the ball in $\mathbb{R}^2$ centered at $(c, 0)$ with radius r. Let

$$S = \{X : P[g^*(X) \neq Y] = \frac{1}{2}\},$$

which is the dark gray area in Figure 1.1. Consider a dissimilarity measure δ. Let

$$S_\delta^* = \{X : P[g_\delta^*(X) \neq Y] = \frac{1}{2}\},$$

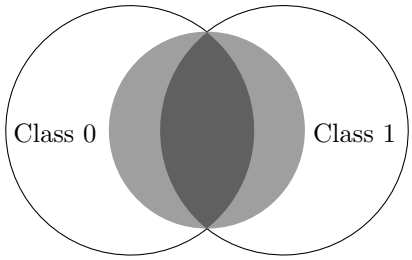


Figure 1.1: $\mathcal{F} = \{F_{XY} : Y \sim \text{Bernoulli}(1/2);\ f_j = f_{X|Y=j} = \text{Uniform}\,(B((-1)^{j+1}, r))\}$

which is the light and dark gray area in Figure 1.1.

Let

$$\mathcal{F}' = \{F_{XY} : F_{XY} \in \mathcal{F}, X \in S; \text{the rest mass of } X \text{ is uniformly distributed on } S_\delta - S\}.$$

Then

$$L^*(\mathcal{F}) = L^*(\mathcal{F}') = L^*,$$

where $L^*(\mathcal{F})$ denotes the Bayes error associated with a distribution $F_{XY} \in \mathcal{F}$. However,

$$L^*_\delta(\mathcal{F}_1) = L^* + \epsilon(\delta, \mathcal{F}) < \frac{1}{2},$$

while

$$L^*_\delta(\mathcal{F}') = L^* + \epsilon(\delta, \mathcal{F}') = \frac{1}{2}.$$

1.3.3 A Special Case of (F_{XY}, δ) : $L^*_\delta = L^*$

In this section, we investigate a special case where the best classifier based on a dissimilarity representation and the Bayes classifier lead to the same classification errors. That

is, $L^*_\delta = L^*$. We hope to shed some light on when and why a dissimilarity representation is useful in statistical pattern recognition. In the rest of the section, for concreteness we assume that X is a discrete random variable.

Theorem 1.3.1. Let $\{(X_i, Y_i)\}_{i=1}^n$ be independent observation and class label pairs so that the X_i's are discrete random variables with class conditional probability mass functions $p_{kj} = P[X_1 = a_k | Y = j]$, and marginal probability mass function $p_k = P[X_1 = a_k],\ j = 0, 1; k = 1, 2, \ldots$, where $a_k \in \mathbb{R}^d$. And let $D = \{\delta : P[\delta(X_1|Y_1 = j_1, X_2|Y_1 = j_2) = 0] = 0,\ j_1 \neq j_2\}$.

(a) If $D \neq \emptyset$, then the Bayes error $L^* = 0$,

(b) For every $\delta \in D$,

$$\lim_{n\to\infty} \inf_{g_n} L(g_n(\Delta) = 0.$$

Proof. It is not hard to see that (b) implies (a), because $L^* \leq L(g_n(\Delta))$ for any n and g_n. If (b) is true, i.e., $L(g(\Delta)) \to 0$, then $L^* = 0$.

The proof for part (b) is exceptionally straightforward. It suffices to show that there exists a classifier $\widetilde{g}_n$ such that $L(\widetilde{g}_n(\Delta)) \to 0$. We define such a classifier as follows. For any new observation, X_{n+1} calculate the $d_{X_{n+1}} = [\delta(X_i, X_{n+1})]_{i=1}^n$. If $\delta(X_i, X_{n+1}) = 0$ then assign to X_{n+1} class Y_i—notice that when there are multiple such i's, by the property of D, all corresponding Y_i's are the same; if there is no such i, then assign a class randomly. Hence for finite n the probability of error is at most the probability that we have not seen

X_{n+1} before. That is,

$$\begin{aligned} L(\widetilde{g}_n(\Delta)) \leq & P[\delta(X_i, X_{n+1}) \neq 0, i = 1, \ldots, n] \\ \leq & P[X_1 \neq X_{n+1}, \ldots, X_n \neq X_{n+1}] \\ = & \sum_k P[X_1 \neq a_k, \ldots, X_n \neq a_k | X_{n+1} = a_k] P[X_{n+1} = a_k] \\ = & \sum_k \left(P[X_{n+1} = a_k] \prod_{i=1}^{n} P[X_i \neq a_k] \right) \\ = & \sum_k p_k (1 - p_k)^n \end{aligned}$$

Now, we **claim** that as $n \to \infty$, $\sum_k p_k(1-p_k)^n \to 0$ which implies that $L(\widetilde{g}_n(\Delta)) \to 0$. □

Proof of claim: Since $0 < p_k \leq 1$, $\sum_k p_k = 1$ and for every n,

$$0 \leq P[X_1 \neq X_{n+1}, \ldots, X_n \neq X_{n+1}] = \sum_k p_k (1 - p_k)^n \leq 1,$$

for any $\epsilon > 0$, there exists $M > 0$ such that

$$\sum_{k>M} p_k (1 - p_k) < \epsilon/2$$

and hence

$$\sum_{k>M} p_k (1 - p_k)^n < \epsilon/2, \quad \text{for any } n.$$

In addition, there exists $N > 0$ such that when $n > N$,

$$\sum_{k=1}^{M} p_k (1 - p_k)^n < \epsilon/2.$$

Hence for $n > N$,

$$\sum_k p_k (1 - p_k)^n = \sum_{k=1}^{M} p_k (1 - p_k)^n + \sum_{k>M} p_k (1 - p_k)^n \leq \epsilon/2 + \epsilon/2 = \epsilon.$$

Therefore $\sum_k p_k(1-p_k)^n \to 0$ as $n \to \infty$.

Theorem 1.3.2. Let $\{(X_i, Y_i)\}_{i=1}^n$ be independent observation and class label pairs so that the X_i's are discrete random variables with class conditional probability mass functions $p_{kj} = P[X_1 = a_k | Y = j]$, and marginal probability mass function $p_k = P[X_1 = a_k]$, $j = 0, 1; k = 1, 2, \ldots$, where $a_k \in \mathbb{R}^d$, the sample space. And let $D = \{\delta : \delta(a, b) = 0 \text{ if and only if } a = b\}$. For every $\delta \in D$, let $\Delta = [\delta(X_i, X_j)]_{n\times n}$, then

$$\lim_{n\to\infty} \inf_{g_n} L(g_n(\Delta)) = L^*.$$

Proof. It suffices to show that there exists a classifier $\widetilde{g}_n$ such that $L(\widetilde{g}_n(\Delta)) \to L^*$. We define such a classifier as follows. For any new observation, X_{n+1} calculate the $d_{X_{n+1}} = [\delta(X_i, X_{n+1})]_{i=1}^n$. Let $A = \{i \in [n] : \delta(X_i, X_{n+1}) = 0\}$. If $|A| > 0$ then assign to X_{n+1} class

$$\widehat{Y}_{n+1} = I\left\{\frac{\sum_{i=1}^n Y_i}{n}\frac{\sum_{i\in A} Y_i}{\sum_{i=1}^n Y_i} > \left(\frac{n - \sum_{i=1}^n Y_i}{n}\right)\frac{|A| - \sum_{i\in A} Y_i}{n - \sum_{i=1}^n Y_i}\right\} \tag{1.2}$$

$$= I\left\{\frac{1}{|A|}\sum_{i\in A} Y_i > \frac{1}{2}\right\}, \tag{1.3}$$

if $|A| = 0$, then randomly assign X_{n+1} a class. Then,

$$L(g(\Delta)) = P[Y_{n+1} \neq \widehat{Y}_{n+1} || A| > 0]P\{|A| > 0\} + P_e P\{|A| = 0\}, \tag{1.4}$$

where P_e is the probability of misclassification when $P[|A| = 0]$. By the proof of theorem 1.3.1, we know that

$$P\{|A| = 0\} \to 0, \text{ as } n \to \infty. \tag{1.5}$$

Therefore

$$P\{|A| > 0\} \to 1, \text{ as } n \to \infty. \tag{1.6}$$

Let $p_1(x) = P[X = x|Y = 1]$ and $p_0(x) = P[X = x|Y = 0]$. The Bayes classifier is

$$g^*(x) = \begin{cases} 1 & \text{if } \pi p_1(x) > (1-\pi)p_0(x) \\ 0 & \text{otherwise,} \end{cases}$$

and the corresponding plug-in decision function is given by (1.2). By [15, theorem 2.1, 2.2; Problem 2.11]. we can show that

$$P[\widehat{Y}_{n+1} \neq Y_{n+1}] \to L^*. \tag{1.7}$$

Hence, (1.4 – 1.7) together imply that

$$L(g(\Delta)) \to L^*.$$

□

Chapter 2

Multidimensional Scaling and Out-of-Sample Embedding

When using a dissimilarity representation, the most widely used approach in statistical pattern recognition is the k-nearest neighbors rule. The k-nearest neighbors rule (k-NN) is a method for classifying objects based on the closest training examples in the training data set.

Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be the n training observation and label pairs—we assume that the class labels Y_i are {0,1}-valued; let X be the new observation to be classified; and let $d_X = [\delta(X, X_1), \ldots, \delta(X, X_n)]$ be the vector of distances/dissimilarities between X and the training observations. Then the k-NN rule is defined by

$$g_n(X) = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i I\{Y_i = 1\} > \sum_{i=1}^n w_i I\{Y_i = 0\}, \\ 0 & \text{otherwise,} \end{cases}$$

where $w_i = 1/k$ if X_i is among the k nearest neighbors of X—$\delta(X, X_i)$ is less than or equal to the k smallest dissimilarities among d_X; $w_i = 0$ elsewhere. That is, X is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors.

We notice that the k-NN rule does not make use of the dissimilarities among training observations, which usually contain useful information about the space where X lives. Figure 2.1 depicts the phenomenon.

$$d_X = \begin{bmatrix} \delta_{11} & \delta_{12} & \delta_{13} & \cdots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \delta_{23} & \cdots & \delta_{2n} \\ \delta_{31} & \delta_{32} & \delta_{33} & \cdots & \delta_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta_{n1} & \delta_{n2} & \delta_{n3} & \cdots & \delta_{nn} \\ \hline \delta_1 & \delta_2 & \delta_3 & \cdots & \delta_n \end{bmatrix}$$

Figure 2.1: Give n training observations and their class labels, the k-NN rule classifies a new observation X by a majority vote of its neighbors. Only the distances/dissimilarities between X and the training observations are used by k-NN. The interpoint dissimilarities among the training observations are not used.

2.1 Multidimensional Scaling

One way to take advantage of the information contained by the dissimilarities among training observations is by means of multidimensional scaling. Consider the dissimilarity matrix $\Delta_n = [\delta_{ij}]_{n \times n}$, obtained by measuring the pairwise dissimilarities of n objects

$O_n = \{o_1, \ldots, o_n\}$. Suppose we do not have the access to the feature representation of O_n. The techniques collectively known as multidimensional scaling (MDS) attempt to construct a configuration of points $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^t$ in a normed linear space (typically Euclidean space) such that the interpoint distances $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ approximate the dissimilarities δ_{ij}. The configuration of points $\mathbf{X}$ is usually referred to as an *embedding* of Δ_n (or O_n).

2.1.1 Criterion Function

Mathematically, a multidimensional scaling procedure aims to find

$$\mathbf{X} = \arg\min c(\Delta, D_{\mathbf{X}}),$$

where $D_{\mathbf{X}} = [d_{ij}(\mathbf{X}) = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|]_{n\times n}$ and c is a criterion function. Usually,

$$d_{ij}(\mathbf{X}) = \sqrt{\sum_k (\boldsymbol{x}_{ik} - \boldsymbol{x}_{jk})^2},$$

and consequently $D_{\mathbf{X}}$ is the Euclidean distance matrix. Examples of widely used criterion functions include:

1. raw Stress:

$$\sum_{i<j} (\delta_{ij} - d_{ij}(\mathbf{X}))^2$$

2. Stress:

$$\frac{\sum_{i<j} (\delta_{ij} - d_{ij}(\mathbf{X}))^2}{\sum_{i<j} \delta_{ij}^2}$$

3. SStress:

$$\frac{\sum_{i<j} (\delta_{ij}^2 - d_{ij}^2(\mathbf{X}))^2}{\sum_{i<j} \delta_{ij}^2}$$

4. Stress-1:

$$\sqrt{\frac{\sum_{i<j}(\delta_{ij} - d_{ij}(\mathbf{X}))^2}{\sum_{i<j} d_{ij}^2(\mathbf{X})}}$$

5. Sammon:

$$\frac{1}{\sum_{i<j}\delta_{ij}} \sum_{i<j} \frac{(\delta_{ij} - d_{ij}(\mathbf{X}))^2}{\delta_{ij}}$$

2.1.2 Classical Multidimensional Scaling

One of the mostly widely used multidimensional scaling techniques, the Classical Multidimensional Scaling (CMDS) [17–19], uses a very different criterion function, which is sometimes called Strain:

$$\|\mathbf{X}\mathbf{X}^t - \mathbf{B}_\Delta\|,$$

where $\mathbf{B}_\Delta = \tau(\Delta^{(2)}) = -\mathbf{J}\Delta^{(2)}\mathbf{J}/2$ and $\mathbf{J} = \mathbf{I} - \mathbf{1}\mathbf{1}^t$. The $\Delta^{(2)}$ is the entry-wise square of the dissimilarity matrix Δ. The operation $\tau(\cdot)$ is usually called double centering. Notice that if the provided dissimilarity matrix Δ is indeed a Euclidean distance matrix derived from $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times p}$, then $\mathbf{B}_\Delta = \tilde{\mathbf{X}}_c\tilde{\mathbf{X}}_c^t$, where $\tilde{\mathbf{X}}_c$ is column-centered $\tilde{\mathbf{X}}$. Factoring $\mathbf{B}_\Delta$ by eigen-decomposition

$$\mathbf{B}_\Delta = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^t = (\mathbf{Q}\mathbf{\Lambda}^{1/2})(\mathbf{Q}\mathbf{\Lambda}^{1/2})^t,$$

yields $\tilde{\mathbf{X}}_c = \mathbf{Q}\mathbf{\Lambda}^{1/2}$. We recover the original feature representation (up to rotation and translation). In case that the original feature is high-dimensional and one wants to find a low-dimensional representation, one can just take the first d columns of $\mathbf{Q}$ and first $d \times d$ block of $\mathbf{\Lambda}$, and the resulting embedding does in fact minimize the Strain.

If the given dissimilarity matrix Δ is not a Euclidean distance matrix, one can still use a similar procedure to obtain an embedding that minimizes the Strain. The procedure can be summarized in the following steps.

1. Calculate $\Delta^{(2)}$.
2. Apply double centering to this matrix to obtain $\mathbf{B}_\Delta = \tau(\Delta^{(2)})$.
3. Perform eigen-decomposition on $\mathbf{B}_\Delta = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^t$.
4. Choose positive eigenvalues in $\mathbf{\Lambda}$ and corresponding columns in $\mathbf{Q}$, yielding $\mathbf{\Lambda}_+$ and $\mathbf{Q}_+$. Then the embedding is given by $\mathbf{Q}_+\mathbf{\Lambda}_+$.

The classical multidimensional scaling has two nice properties: (1) it is closed-form and hence very fast; (2) the dimensions of the resulting embedding are nested—if one embeds Δ respectively into $\mathbf{X}_1 \in \mathbb{R}^{n\times d}$ and $\mathbf{X}_2 \in \mathbb{R}^{n\times(d+l)}$, $l > 0$, then the first d-dimensions of $\mathbf{X}_2$ are of the same as those of $\mathbf{X}_1$.

2.1.3 Out-of-Sample Extension for Classical Multidimensional Scaling

Suppose that inference methodologies (testing, estimation, etc.) are based on the embedding configuration $\mathbf{X}$ as well as objects observed in future $\tilde{O}_m = \{\tilde{o}_1, \ldots, \tilde{o}_m\}$, which are referred to as the *out-of-sample* objects. Analogously, O_n are called the *within-sample*

objects. Assume that we have access only to the dissimilarities from each of the m out-of-sample objects to each of the n original objects, $\Delta_{mn} = [\dot{\delta}_{ij}]_{m\times n}$, and the pairwise dissimilarities among the out-of-sample objects, $\Delta_{mm} = [\ddot{\delta}_{ij}]_{m\times m}$. In this dissertation, when necessary we use $\dot{\delta}_{ij}$ to denote the dissimilarity between an out-of-sample object and a within-sample object, and use $\ddot{\delta}_{ij}$ to denote the dissimilarity between two out-of-sample objects. In order to use the out-of-sample objects in the inference, we need to find a configuration of points $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m]^t$ in relation to the representation space specified by $\mathbf{X}$. We could, of course, (re-)embed O_n and $\tilde{O}_m$ together. However, for large n, re-embedding O_n and $\tilde{O}_m$ together will be computationally prohibitive in many applications.

Assuming the within-sample embedding was obtained through the classical multidimensional scaling, Trosset and Priebe [20] formulate the out-of-sample extension (hereafter referred to as T&P) as an unconstrained nonlinear least-squares problem:

$$\min \left\| \mathbf{B} - \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} [\mathbf{X}^t \ \mathbf{Y}^t] = \right\|^2 = \min \ 2\|\mathbf{B}_{xy} - \mathbf{X}\mathbf{Y}^t\|^2 + \|\mathbf{B}_{yy} - \mathbf{Y}\mathbf{Y}^t\|^2. \qquad (2.1)$$

The matrix

$$\mathbf{B} = \tau_w(\Delta^{(2)}) = \begin{bmatrix} \tau(\Delta_n^{(2)}) & \mathbf{B}_{xy} \\ \mathbf{B}_{xy}^t & \mathbf{B}_{yy} \end{bmatrix},$$

where

$$\Delta^{(2)} = \begin{bmatrix} \Delta_n^{(2)} & \Delta_{nm}^{(2)} \\ \Delta_{mn}^{(2)} & \Delta_m^{(2)} \end{bmatrix},$$

$$\tau_w(\Delta^{(2)}) = -\frac{1}{2}\left(\mathbf{I} - \frac{ew^t}{e^t w}\right)\Delta^{(2)}\left(\mathbf{I} - \frac{we^t}{e^t w}\right),$$

and w is the vector whose first n entries are all 1's and the rest are all 0's, $e = (1, \ldots, 1)^t \in \mathbb{R}^{n+m}$. Equation (2.1) is solved numerically by standard gradient-based methods.

Notice that if O_n is embedded through a procedure other than CMDS, T&P may not be appropriate. Ma and Priebe [21] introduce another out-of-sample embedding procedure—Out-of-Sample Embedding by Iterative Majorization (OOSIM), which we review in Section 2.2. OOSIM is developed as an extension to the multidimensional scaling procedure that uses the raw Stress.

2.2 Out-of-Sample Embedding by Iterative Majorization

The out-of-sample problem arises in various ways. Distance and dissimilarity measures are often the observed values in psychology experiments in which subjects are asked questions about which stimuli are closest [22]. In studying the hippocampus, diffeomorphic distances are measured in [23] to distinguish between certain types of disorder. Applications such as these, wherein the observations take the form of dissimilarities rather than features, are often attacked via embedding into a space in which inference is performed. Subsequent observations then require either an out-of-sample embedding approach or a full re-analysis of the original data. Out-of-sample embedding also arises as an artifact of the computational complexity of some statistical approaches, such as in using V-fold cross-validation to estimate a classifier's error rate [20]. Gower [24] provides a method to

embed only one out-of-sample object, given its *Euclidean* distances to previously specified configuration of fixed points. Trosset and Priebe [20] develop the out-of-sample extension for classical multidimensional scaling (CMDS) [17–19].

In this work, we consider the following out-of-sample problem: find a configuration of points $\mathbf{Y}$ to minimize the sum of squared differences, specified by the criterion function

$$\sigma_{\mathbf{X}}(\mathbf{Y}) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\dot{\delta}_{ij} - d_{ij}(\mathbf{Y},\mathbf{X})\right)^2 + \sum_{1\le i<j\le m}(\ddot{\delta}_{ij} - d_{ij}(\mathbf{Y}))^2, \tag{2.2}$$

where

$$d_{ij}(\mathbf{Y},\mathbf{X}) = \|\boldsymbol{y}_i - \boldsymbol{x}_j\|_2 \quad \text{and} \quad d_{ij}(\mathbf{Y}) = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2.$$

Following de Leeuw's lead, we minimize $\sigma_{\mathbf{X}}(\mathbf{Y})$ by Iterative Majorization (IM). IM is an elegant method to minimize a function, and it is based on the work of [25].

2.3 Minimizing Stress via Iterative Majorization

The Iterative Majorization method finds a minimizer of the original complicated function $f(x)$ by means of iteratively minimizing an auxiliary function $h(x,z)$, which is always greater than or equal to $f(x)$ and is easier to minimize. The z in $h(x,z)$ is some fixed value called the *supporting point*, at which the auxiliary function h should touch the surface of f. That is, $f(z) = h(z,z)$. To see how IM works, consider a sequence of supporting points $\{z_1,\ldots\}$ defined as follows. Given z_t, let $z_{t+1} := \arg\min_x h(x,z_t)$. Then,

$$f(z_t) = h(z_t,z_t) \ge h(z_{t+1},z_t) \ge f(z_{t+1}).$$

Performing this procedure repeatedly yields a monotonically non-increasing sequence of function values. Since (by assumption) the optimization of h is easier than that of f, the sequence of supporting points can be found relatively quickly. This is the general concept of IM.

Borg and Groenen [19, Section 8.6], Leeuw and Mair [26] elaborate using IM to minimize the stress function in MDS. We briefly review their work in the remainder of this section.

Let $\Delta_n = [\delta_{ij}]$ be the $n \times n$ dissimilarity matrix, and $\mathbf{X}$ be the $n \times d$ configuration matrix. Consider the raw stress [27]

$$\begin{aligned}
\sigma_r(\mathbf{X}) &= \sum_{i<j} w_{ij} \left(\delta_{ij} - d_{ij}(\mathbf{X})\right)^2 \\
&= \sum_{i<j} w_{ij}\delta_{ij}^2 + \sum_{i<j} w_{ij} d_{ij}^2(\mathbf{X}) - 2\sum_{i<j} w_{ij}\delta_{ij} d_{ij}(\mathbf{X}) \\
&= \eta_\delta^2 + \eta^2(\mathbf{X}) - 2\rho(\mathbf{X}),
\end{aligned} \tag{2.3}$$

where $\mathbf{W} = [w_{ij}]$ is an $n \times n$ weight matrix, which is symmetric and non-negative. Let $\boldsymbol{x}_{\cdot k} = [x_{ik}, \ldots, x_{nk}]$ be the kth column of $\mathbf{X}$. Let $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ be the ith and jth columns of the identity matrix $\mathbf{I}_n$. Then

$$\begin{aligned}
d_{ij}^2(\mathbf{X}) &= \sum_{k=1}^{d} (x_{ik} - x_{jk})^2 \\
&= \sum_{k=1}^{d} \boldsymbol{x}_{\cdot k}^t (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^t \boldsymbol{x}_{\cdot k} \\
&= \sum_{k=1}^{d} \boldsymbol{x}_{\cdot k}^t \mathbf{A}_{ij} \boldsymbol{x}_{\cdot k},
\end{aligned} \tag{2.4}$$

where $\mathbf{A}_{ij}$ is simply a matrix with all 0 elements, except these four entries: $a_{ii} = a_{jj} = 1$, $a_{ij} = a_{ji} = -1$. Hence $\eta^2(\mathbf{X})$ is a weighted sum of $d_{ij}^2(\mathbf{X})$:

$$\eta^2(\mathbf{X}) = \sum_{i<j} w_{ij} d_{ij}^2(\mathbf{X}) = \text{tr } \mathbf{X}^t \left(\sum_{i<j} w_{ij} \mathbf{A}_{ij} \right) \mathbf{X} = \text{tr } \mathbf{X}^t \mathbf{V} \mathbf{X}, \tag{2.5}$$

where $\mathbf{V} = \sum_{i<j} w_{ij} \mathbf{A}_{ij}$. In general, the (i, j)th element of $\mathbf{V}$ is given by

$$v_{ij} = \begin{cases} -w_{ij} & i \neq j, \\ \sum_{j=1, j\neq i}^{n} w_{ij} & i = j. \end{cases}$$

The matrix $\mathbf{V}$ has all non-negative eigenvalues by the Geršgorim disc theorem, and hence it is positive-semidefinite.

Let $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n]^t$ be another $n \times d$ matrix. The Cauchy-Schwarz inequality implies

$$\sum_{k=1}^{d} (x_{ik} - x_{jk})(z_{ik} - z_{jk}) \leq \left(\sum_{k=1}^{d} (x_{ik} - x_{jk})^2 \right)^{1/2} \left(\sum_{k=1}^{d} (z_{ik} - z_{jk})^2 \right)^{1/2}.$$

In matrix notation, this inequality becomes

$$\text{tr } \mathbf{X}^t \mathbf{A}_{ij} \mathbf{Z} \leq d_{ij}(\mathbf{X}) d_{ij}(\mathbf{Z}), \tag{2.6}$$

with equality if $\mathbf{Z} = \mathbf{X}$. Inequality (2.6) implies

$$-d_{ij}(\mathbf{X}) \leq \begin{cases} -\dfrac{\text{tr } \mathbf{X}^t \mathbf{A}_{ij} \mathbf{Z}}{d_{ij}(\mathbf{Z})} & d_{ij}(\mathbf{Z}) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

Multiplying both sides of (2.7) by $w_{ij}\delta_{ij}$ and summing over all $i < j$ gives

$$-\rho(\mathbf{X}) = -\sum_{i<j} w_{ij} \delta_{ij} d_{ij}(\mathbf{X}) \leq -\text{tr } \mathbf{X}^t \left(\sum_{i<j} b_{ij} \mathbf{A}_{ij} \right) \mathbf{Z} = -\text{tr } \mathbf{X}^t \mathbf{B}(\mathbf{Z}) \mathbf{Z},$$

where $\mathbf{B}(\mathbf{Z})$ has elements

$$b_{ij} = \begin{cases} -\dfrac{w_{ij}\delta_{ij}}{d_{ij}(\mathbf{Z})} & i \neq j,\ d_{ij}(\mathbf{Z}) \neq 0, \\ 0 & i \neq j,\ d_{ij}(\mathbf{Z}) = 0, \\ -\displaystyle\sum_{j=1, j\neq i}^{n} b_{ij} & i = j. \end{cases}$$

Because equality occurs if $\mathbf{Z} = \mathbf{X}$, we have obtained the majorization inequality

$$-\rho(\mathbf{X}) = -\text{tr } \mathbf{X}^t\mathbf{B}(\mathbf{X})\mathbf{X} \leq -\text{tr } \mathbf{X}^t\mathbf{B}(\mathbf{Z})\mathbf{Z}. \tag{2.8}$$

Combining (2.3), (2.5) and (2.8) gives the majorization inequality for the stress function

$$\begin{aligned} \sigma_r(\mathbf{X}) &= \eta_\delta^2 + \text{tr } \mathbf{X}^t\mathbf{V}\mathbf{X} - 2\text{tr } \mathbf{X}^t\mathbf{B}(\mathbf{X})\mathbf{X} \\ &\leq \eta_\delta^2 + \text{tr } \mathbf{X}^t\mathbf{V}\mathbf{X} - 2\text{tr } \mathbf{X}^t\mathbf{B}(\mathbf{Z})\mathbf{Z} \\ &\overset{\text{def.}}{=} \tau(\mathbf{X}, \mathbf{Z}). \end{aligned} \tag{2.9}$$

The function $\tau(\mathbf{X}, \mathbf{Z})$ is a simple quadratic function in $\mathbf{X}$. It is also convex, for the Hessian matrix $\mathbf{V}$ is positive-semidefinite. Hence first-order conditions are necessary and sufficient. Moreover, because $\tau(\mathbf{X}, \mathbf{Z}) \geq \sigma_r(\mathbf{X}) \geq 0$, the Frank-Wolfe Theorem [28] guarantees the existence of a solution. Therefore, the minimum of $\tau(\mathbf{X}, \mathbf{Z})$ can be obtained analytically by solving the equation system

$$\nabla\tau(\mathbf{X}, \mathbf{Z}) = \frac{\partial}{\partial \mathbf{X}}\tau(\mathbf{X}, \mathbf{Z}) = 2\mathbf{V}\mathbf{X} - 2\mathbf{B}(\mathbf{Z})\mathbf{Z} = 0. \tag{2.10}$$

Because $\mathbf{V}$ is not of full rank, the Moore-Penrose inverse $\mathbf{V}^+$ is used in the solution of (2.10), which is

$$\mathbf{X} = \mathbf{V}^+\mathbf{B}(\mathbf{Z})\mathbf{Z}. \tag{2.11}$$

In fact, any 1-generalized inverse matrix $\mathbf{V}^-$ such that $\mathbf{V}\mathbf{V}^-\mathbf{V} = \mathbf{V}$ will suffice. The choice of the Moore-Penrose inverse in particular is for uniqueness. In summary, the IM procedure for MDS can be described as:

1. Set $\mathbf{Z} := \mathbf{X}^{(0)}$ where $\mathbf{X}^{(0)}$ is a (random) start configuration.

2. Find an update $\mathbf{X}^{(t)}$ by (2.11) and compute $\sigma_r(\mathbf{X}^{(t)})$.

3. Stop iterating if $\sigma_r(\mathbf{X}^{(t-1)}) - \sigma_r(\mathbf{X}^{(t)}) < \epsilon$ or a certain iteration limit is reached. Otherwise, update $\mathbf{Z} := \mathbf{X}^{(t)}$ and go to step 2.

2.4 Out-of-sample Embedding

Having obtained a configuration $\mathbf{X}$ based on the $n \times n$ dissimilarity matrix Δ_n, we shall now develop the IM procedure to find a configuration $\mathbf{Y}$ for the m out-of-sample objects according to the criterion function (2.2). Let

$$\mathbf{\Delta} = [\delta_{ij}]_{(n+m)\times(n+m)} = \begin{bmatrix} \Delta_n & \Delta_{mn}^t \\ \Delta_{mn} & \Delta_m \end{bmatrix}$$

and

$$\mathbf{W} = [w_{ij}]_{(n+m)\times(n+m)} = \begin{bmatrix} \mathbf{0}_n & \mathbf{W}_{mn}^t \\ \mathbf{W}_{mn} & \mathbf{W}_m \end{bmatrix},$$

where $\mathbf{0}_n$ is the $n \times n$ matrix with all 0 entries, and $\mathbf{W}_{mn}$ and $\mathbf{W}_m$ are the $m \times n$ and $m \times m$ matrices with all 1 entries, respectively. In the case of missing dissimilarities in Δ_{mn} and

Δ_m, set the corresponding weights w_{ij} in $\mathbf{W}_{mn}$ and $\mathbf{W}_m$ to 0. Then the out-of-sample criterion function (2.2) becomes

$$\sigma_{\mathbf{X}}(\mathbf{Y}) = \sum_{1 \le i < j \le n+m} w_{ij} \left(\delta_{ij} - d_{ij} \left(\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \right) \right)^2 . \tag{2.12}$$

Using the matrix notation of raw stress introduced in Section 2.3, the criterion function (2.12) can be written as

$$\begin{aligned} \sigma_{\mathbf{X}}(\mathbf{Y}) &= \eta_\delta^2 + \eta^2 \left(\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \right) - 2\rho \left(\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \right) \\ &= \eta_\delta^2 + \text{tr} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}^t \mathbf{V} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} - 2\text{tr} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}^t \mathbf{B} \left(\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \right) \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \\ &\le \eta_\delta^2 + \text{tr} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}^t \mathbf{V} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} - 2\text{tr} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}^t \mathbf{B} \left(\begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix} \right) \begin{bmatrix} \mathbf{X} \\ \mathbf{Z} \end{bmatrix}, \end{aligned} \tag{2.13}$$

where the inequality in (2.13) is by (2.9). As the decomposition of $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{B}$ have the following block structure

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_n & \mathbf{V}_{mn}^t \\ \mathbf{V}_{mn} & \mathbf{V}_m \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \mathbf{B}_n & \mathbf{B}_{mn}^t \\ \mathbf{B}_{mn} & \mathbf{B}_m \end{bmatrix}. \tag{2.14}$$

Combining (2.13) and (2.14) gives

$$\begin{aligned}\sigma_{\mathbf{X}}(\mathbf{Y}) \leq \eta_\delta^2 + \text{tr } &\left\{\mathbf{X}^t\mathbf{V}_n\mathbf{X} - 2\mathbf{X}^t\mathbf{B}_n\mathbf{X} - 2\mathbf{X}^t\mathbf{B}_{mn}^t\mathbf{Z}\right\} + \\ &\text{tr } \mathbf{Y}^t\mathbf{V}_m\mathbf{Y} + \\ &2 \text{ tr } \left\{\mathbf{Y}^t\mathbf{V}_{mn}\mathbf{X} - \mathbf{Y}^t\mathbf{B}_{mn}\mathbf{X} - \mathbf{Y}^t\mathbf{B}_m\mathbf{Z}\right\} \\ &\stackrel{\text{def.}}{=} \tau_{\mathbf{X}}(\mathbf{Y}, \mathbf{Z}). \end{aligned} \tag{2.15}$$

Obviously, $\tau_{\mathbf{X}}(\mathbf{Y}, \mathbf{Z})$ is a quadratic function in $\mathbf{Y}$ which majorizes $\sigma_{\mathbf{X}}(\mathbf{Y})$. It is also convex, for the Hessian matrix $\mathbf{V}_m$ is positive-semidefinite. Hence first-order conditions are necessary and sufficient. Moreover, because $\tau_{\mathbf{X}}(\mathbf{Y}, \mathbf{Z}) \geq \sigma_{\mathbf{X}}(\mathbf{Y}) \geq 0$, the Frank-Wolfe Theorem guarantees the existence of a solution. Its minimum can be obtained analytically by solving the equation system

$$\nabla\tau_{\mathbf{X}}(\mathbf{Y}, \mathbf{Z}) = \frac{\partial}{\partial\mathbf{Y}}\tau_{\mathbf{X}}(\mathbf{Y}, \mathbf{Z}) = 2\mathbf{V}_m\mathbf{Y} + 2\mathbf{V}_{mn}\mathbf{X} - 2\mathbf{B}_{mn}\mathbf{X} - 2\mathbf{B}_m\mathbf{Z} = 0. \tag{2.16}$$

By using the Moore-Penrose inverse of $\mathbf{V}_m^+$, the solution of (2.16) is given by

$$\mathbf{Y} = \mathbf{V}_m^+(\mathbf{B}_{mn} - \mathbf{V}_{mn})\mathbf{X} + \mathbf{V}_m^+\mathbf{B}_m\mathbf{Z}. \tag{2.17}$$

In summary, the IM procedure for out-of-sample embedding can be described as:

1. Set $\mathbf{Z} := \mathbf{Y}^{(0)}$ where $\mathbf{Y}^{(0)}$ is a (random) start configuration.

2. Find an update $\mathbf{Y}^{(t)}$ by (2.17) and compute $\sigma_{\mathbf{X}}(\mathbf{Y}^{(t)})$.

3. Stop iterating if $\sigma_{\mathbf{X}}(\mathbf{Y}^{(t-1)}) - \sigma_{\mathbf{X}}(\mathbf{Y}^{(t)}) < \epsilon$ or a certain iteration limit is reached. Otherwise, update $\mathbf{Z} := \mathbf{Y}^{(t)}$ and go to step 2.

We call this procedure OOSIM (Out-Of-Sample embedding by Iterative Majorization).

2.5 Example: Simulated Dissimilarity Data

When using out-of-sample objects in conjunction with a configuration derived from within-sample embedding, it is implicitly assumed that the embedding of out-of-sample objects has the same distribution as the embedding of the original objects. In this section, we shall examine this assumption via simulation.

Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ be a sample from the multivariate Gaussian distribution with mean $\boldsymbol{\mu} = [0, 0, 0, 0]^t$ and covariance/variance matrix $\mathbf{I}_4$. The interpoint dissimilarity matrix is calculated as

$$\Delta_n = [\delta_{ij} = 1 - \cos(\boldsymbol{u}_i, \boldsymbol{u}_j)], \tag{2.18}$$

where $\cos(\boldsymbol{u}_i, \boldsymbol{u}_j) = \boldsymbol{u}_i \cdot \boldsymbol{u}_j / (\|\boldsymbol{u}_i\|_2 \|\boldsymbol{u}_j\|_2)$, the cosine of the angle between $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$. Cosine similarity is often used to compare documents in text mining [29].

Let $\mathbf{X}$ ($\mathbf{X} \in \mathbb{R}$) be the embedding of Δ_n. Suppose later out-of-sample $\tilde{\boldsymbol{u}}_1, \ldots, \tilde{\boldsymbol{u}}_m$ are observed, and their dissimilarities to the original objects Δ_{mn} and the dissimilarities among themselves Δ_m are computed. Out-of-sample embedding is then performed to obtain $\mathbf{Y}$. We use the Kolmogorov-Smirnov test to determine whether $\mathbf{Y}$ has the same distribution as $\mathbf{X}$. To illustrate the potential difference between the two out-of-sample embedding approaches, OOSIM and T&P, we use three different multidimensional scaling techniques to obtain the within-sample embedding $\mathbf{X}$—the classical multidimensional scaling (CMDS), Sammon's non-metric multidimensional scaling [30] (Sammon), and multidimensional scaling by minimizing raw stress (SMACOF). OOSIM and T&P are separately used to

embed out-of-sample objects. Notice that T&P is consistent with CMDS, and OOSIM is consistent with SMACOF, while neither is consistent with Sammon. The following experiments are for illustration and comparison purposes.

Let $n = m = 100$ and consider 1000 Monte Carlo replicates. For each replicate, the Kolmogorov-Smirnov test is applied to test the hypotheses

$$H_0 : \ F_{\mathbf{Y}} = F_{\mathbf{X}} \text{ vs. } H_A : \ F_{\mathbf{Y}} \neq F_{\mathbf{X}}.$$

Table 2.1 shows the average of p-values of the tests for various within-sample and out-of-sample embedding combinations. Notice that: when CMDS is used to obtain the within-sample configuration $\mathbf{X}$, T&P's out-of-sample approach leads to large p-values, indicating no evidence to reject the null hypothesis, while OOSIM leads to small p-values, suggesting OOSIM is not an appropriate out-of-sample embedding extension to CMDS; on the other hand, if $\mathbf{X}$ is obtained by SMACOF, the opposite is true for T&P and OOSIM; when Sammon is used to obtain $\mathbf{X}$, both T&P and OOSIM result in average p-values > 0.05, although OOSIM's p-values are larger giving some indication that this method may be more appropriate. For more information, consider examining the distribution of the corresponding p-values. It is known that under the null hypothesis, the p-value should be uniformly distributed on $[0, 1]$. An appropriate out-of-sample approach should result in p-values distributed approximately uniformly. Figure 2.2 gives the histograms of the p-values resulting from using the two different out-of-sample embedding approaches when within-sample embedding is obtained via Sammon. It is clear that the histogram of the p-values associated with T&P's approach skews strongly to the left, while the OOSIM histogram suggests

approximately uniformly distributed p-values. We conclude that OOSIM is more appropriate than T&P for this particular setting. A general theory indicating which out-of-sample embedding method is more appropriate (more robust) remains to be developed.

Within-sample	CMDS		Sammon		SMACOF	
Out-of-Sample	T&P	OOSIM	T&P	OOSIM	T&P	OOSIM
p-value	0.467	0.031	0.200	0.439	0.022	0.698

Table 2.1: Average p-values from the Kolmogorov-Smirnov test resulting from various within-sample and out-of-sample embedding combinations. When CMDS is used to obtain the within-sample configuration $\mathbf{X}$, T&P yields large p-values suggesting (correctly) that this method is appropriate, while OOSIM leads to small p-values. On the other hand, if $\mathbf{X}$ is obtained by SMACOF, the opposite is (correctly) true. When Sammon non-metric multidimensional scaling is used to obtain $\mathbf{X}$, both T&P and OOSIM result in large p-values, but Figure 2.2 suggests that OOSIM is more appropriate.

2.6 Conclusion

We have presented an out-of-sample embedding approach (OOSIM) to insert additional points into existing configurations by minimizing sum of squared differences between dissimilarities and the corresponding Euclidean distances. Iterative Majorization is used to minimize the criterion function. The simulation experiment suggests that OOSIM is a natural extension to de Leeuw's multidimensional scaling procedure, SMACOF, which minimizes raw stress. Moreover, we have compared two out-of-sample embedding approaches,

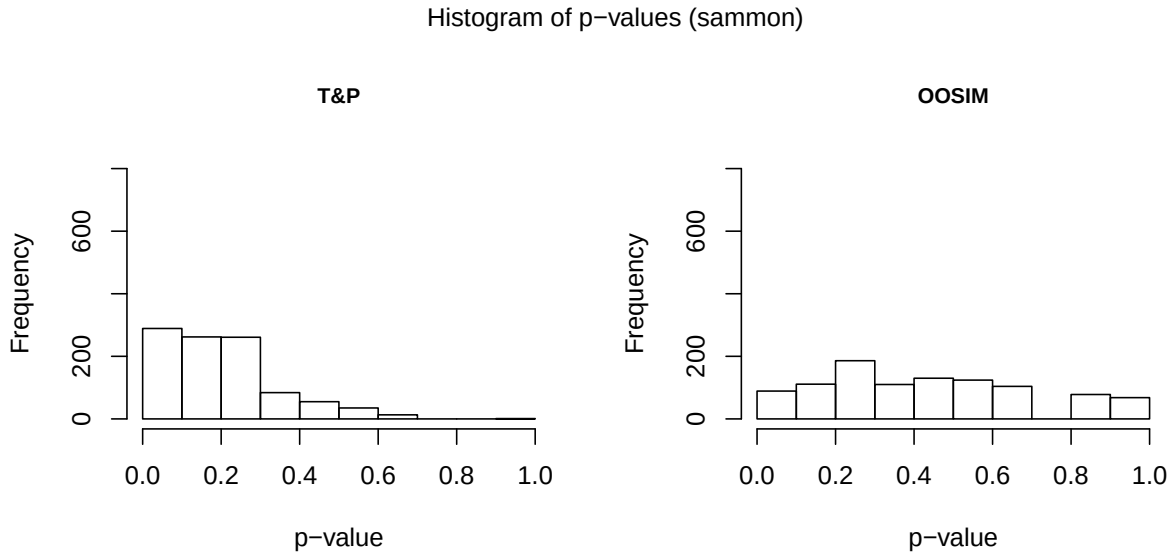


Figure 2.2: Histograms of p-values from using T&P and OOSIM for out-of-sample embedding respectively, when Sammon's non-metric multidimensional scaling is used to obtained within-sample embedding $\mathbf{X}$. OOSIM yields approximately uniformly distributed p-values, suggesting that this method is more appropriate for Sammon within-sample embedding.

OOSIM and T&P, in conjunction with different within-sample procedures. The results show T&P is consistent with CMDS, OOSIM is consistent with SMACOF; both T&P and OOSIM could be used to embed out-of-sample objects even when the within-sample embedding was not obtained by CMDS or SMACOF, and we present an example for which OOSIM is more appropriate than T&P.

We believe that these results motivate the development of a general robustness theory indicating which out-of-sample embedding method is more appropriate in cases where neither CMDS nor SMACOF is the within-sample embedding methodology.

Finally, we note that the SMACOF/OOSIM has a practical advantage over CMDS/T&P, in that SMACOF/OOSIM can easily handle missing dissimilarity values while CMDS/T&P is not directly applicable in such cases.

Chapter 3

Combining Dissimilarity Matrices In Cartesian Product Space

In this chapter, we consider the problem of combining multiple dissimilarity representations via the Cartesian product of their embeddings. For concreteness, we choose the inferential task at hand to be classification. The high dimensionality of this Cartesian product space implies the necessity of dimensionality reduction before training a classifier. We propose a supervised dimensionality reduction method, which utilizes the class label information, to help achieve a favorable combination. The simulation and real data results show that our approach can improve classification accuracy compared to the alternatives of principal components analysis and no dimensionality reduction at all.

3.1 Background

Most traditional statistical pattern recognition techniques rely on objects represented by points in a feature (vector) space. In this space, classifiers are developed to best separate the objects of different classes. As an alternative to the feature-based representation, the dissimilarity representation describes objects by their interpoint comparisons. The dissimilarity representation has attracted substantial interest in various areas [10–14].

Because there are many ways to compare two objects—for example, the L^p-distances—it is possible to construct many dissimilarity representations. Ideally, each dissimilarity representation captures different aspects of the underlying patterns. Consequently, combining multiple dissimilarity representations can be beneficial. One way to combine multiple dissimilarity representations is via the Cartesian product of their embeddings. The high dimensionality of this embedding product space implies the necessity of dimensionality reduction before training a classifier. Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote the original (or transformed) p-dimensional data matrix and let $\mathbf{X}_A$ denote a submatrix that contains only the columns of $\mathbf{X}$ with indices in $A \subseteq \{1, \ldots, p\}$. The problem of dimensionality reduction is to find an index set A of size $d \equiv |A| < p$ such that the classification error based on the d-dimensional data $\mathbf{X}_A$ is small.

Principal component analysis (PCA) is the most widely used method for dimensionality reduction, but it does not take into account the class label information, which is crucial for extracting discriminative features. Linear discriminant analysis (LDA) is also broadly used for dimensionality reduction (or as a classifier), and it uses class label information.

However relatively small sample size (compared to dimensionality p) may cause LDA's performance to decrease when adding more dimensions, even though the extra dimensions contain discriminative information. Trunk [31] affirmed this phenomenon by investigating an illuminating simple example. Chang [32], Dillon et al. [33], Kshirsagar et al. [34] all established a statistic θ_k for the kth principal component (PC) and use θ_k to decide which PCs should be used in discrimination. Jolliffe et al. [35] observed that, for two-class problem, the sample estimate $\hat{\theta}_k$ is equivalent to a t-statistic and the hypothesis test based on θ_k to decide whether or not to include the kth PC is equivalent to the t-test with null hypothesis H_{0k} that there is no difference between the two class means. The statistic θ_k is useful in determining the order of importance of PCs in separating the two populations. However, the best d individual PCs do not necessarily constitute the best subset of d PCs [36]. Takemura [37] proposed a decomposition of the Hotelling's T^2 statistic by projecting data onto the principal axes of the pooled covariance matrix, then calculating t-statistic t_k for the kth PC. Takemura suggested using the first d PCs ("... to look at $t_1^2, t_1^2 + t_2^2, \ldots$") and briefly mentioned "If one has a prior idea about the importance of various axes, a weighted sum of t_k^2, $T_w^2 = \sum_{k=1}^{d} w_k t_k^2$, might be considered." Following Takemura's framework for decomposing Hotelling's T^2, we propose choosing the d PCs that correspond to the d largest values of $J_k \equiv |t_k|$. We show that, under the assumption of mixture of two multivariate Gaussian distributions with equal covariance matrices, the best d individual PCs coincide with the best subset of d PCs. We demonstrate the use of this approach with simulation, image and caption data. The results show that, for classification, our approach outperforms

both PCA and no dimensionality reduction.

In Section 3.2, we describe the background of combining multiple dissimilarity representations, in particular, via the Cartesian product of their embeddings. Section 3.3 details the proposed supervised dimensionality reduction method. Simulation and real data examples are presented in Section 3.4. Section 3.5 provides conclusions and how to extend our approach to suit a problem with more than two classes.

3.2 Combining Multiple Dissimilarity Representations

As defined in 1.1.2, a *dissimilarity measure* is a function $\delta : \Xi \times \Xi \to \mathbb{R}_+ \cup \{0\}$ with $\delta(z_1, z_2) \geq 0,\ \delta(z_1, z_2) = \delta(z_2, z_1)$ and $\delta(z_1, z_2) = 0$ if and only if $z_1 = z_2$. It measures the magnitude of difference between two objects. Asymmetric functions are also of interest, but this is beyond the scope of the this work. Notice that in most cases $\Xi = \mathbb{R}^d$. However we wish to leave open the possibility for applications where the original data are infinite dimensional, graph-valued, or occupying some other nonstandard space. In cases where we observe only the dissimilarities, it will still be useful to imagine that they are computed from a set of Ξ-valued vectors—the "measurements" of objects. The *dissimilarity representation* of a set of objects is obtained by computing δ on each pair of objects. It is expressed as a nonnegative and symmetric matrix Δ with all zero diagonal entries.

Let $\delta_1, \ldots, \delta_K$ denote K dissimilarity measures. Given n object-label pairs $(z_i, y_i) \in \Xi \times \{0, 1\},\ i = 1, \ldots, n$, let

$$\Delta_k = [\delta_k(z_i, z_j)],\ k = 1, \ldots, K,$$

be the corresponding K dissimilarity matrices. The task is to combine these K dissimilarity matrices in order to obtain superior (compared to any one of the Δ_k alone) performance in classification.

As illustrated in Figure 3.1, there are (at least) three possible ways to combine dissimilarities: (1) "classifier ensemble" combines separate classifiers that were trained on individual dissimilarity matrices; (2) "dissimilarity combination" trains one classifier from a single combined dissimilarity matrix; (3) "embedding product" embeds each dissimilarity matrix first, then combines the embeddings to build a classifier. The process of embedding an $n \times n$ dissimilarity matrix, $\Delta = [\delta(z_i, z_j)]$, involves finding a configuration of points, $x_1, \ldots, x_n$, in a normed linear space, such that the interpoint distances, $\|x_i - x_j\|$, approximate the $\delta(z_i, z_j)$. When the normed linear space is Euclidean, embedding is widely known as multidimensional scaling. The configuration of points, here denoted $\mathbf{X}$, is called the *embedding* of Δ. (In this work, we use the bold $\mathbf{X}$ to denote an $n \times d$ data matrix, where each row corresponds to a d-dimensional observation; and we use X to denote a d-dimensional random vector.) The "classifier ensemble" is the most straightforward conceptually, and the easiest to implement. It is necessarily suboptimal, at least in principle, because the joint distribution usually provides more information than the product of the marginals. The "dissimilarity combination" seems as if it should be more natural but is the

hardest of these to implement. To combine dissimilarity matrices directly and beneficially, one has to explore the underlying dissimilarity measures. For example, if all the dissimilarity measures are the squared Euclidean distances, then the summation of all dissimilarity matrices makes perfect sense, because it is a squared Euclidean distance matrix in the joint space. However, this kind of phenomenon will not hold for most pairs of dissimilarity measures, for the dissimilarity measures used to generate the dissimilarity matrices are usually complicated and quite different from each other. These issues make the "dissimilarity combination" decidedly nontrivial. Nevertheless, a naïve, but possibly effective, overview of this method can be found in [13, Equation 10.1]. The "embedding product" approach, on the other hand, not only considers the joint distribution by means of the Cartesian product of the embeddings, but also requires no specific knowledge of the underlying dissimilarity measures. In this work, we focus on the "embedding product" approach and discuss in depth how to perform dimensionality reduction in the Cartesian product space.

The key to the "embedding product" approach is to determine the "right" embedding dimensionality d_k of each Δ_k and the dimensionality of the Cartesian product space. Miller et al. [23] gave an example in the $K = 2$ case. They embedded Δ_1 and Δ_2 into $\mathbb{R}^{d_1}$ and $\mathbb{R}^{d_2}$, ranging d_1 and d_2 from 0 to some maximum d_1^{max} and d_2^{max}, respectively. (In their case, $d_1^{max} = d_2^{max} = 15$.) They then built a classifier for each possible combination of (d_1, d_2), and obtained an estimate of the classification error L_{d_1,d_2}. In the end, they chose $(\hat{d}_1, \hat{d}_2) = \arg\min L_{d_1,d_2}$. This method is necessarily suboptimal as it includes all the first $\hat{d}_1$ and $\hat{d}_2$ PCs, but ignores higher rank PCs, which may contain discriminative information.

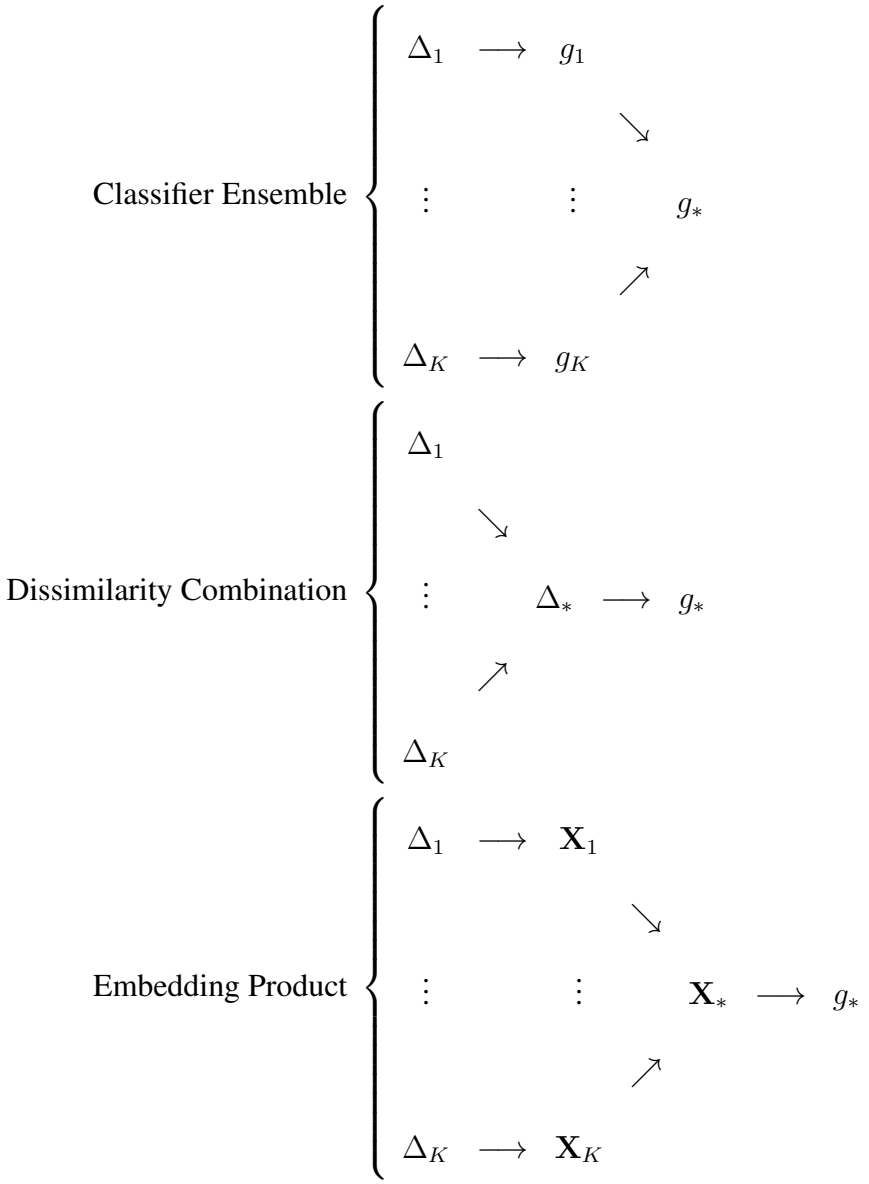


Figure 3.1: $\Delta_1, \cdots, \Delta_K$ are K dissimilarity matrices. "Classifier ensemble" combines separate classifiers g_k that were trained on individual dissimilarity matrices Δ_k; "dissimilarity combination" trains one classifier g_* from a single combined dissimilarity matrix Δ_*; "embedding product" embeds each Δ_k into $\mathbf{X}_k$ and combines those embeddings to obtain $\mathbf{X}_*$, and then a classifier is trained.

It also becomes unwieldy for $K > 2$.

An alternative way to implement the "embedding product" approach is to embed each Δ_k into $\mathbf{X}_k \in \mathbb{R}^{n \times d_k}$, and construct a classifier in the Cartesian product space $[\mathbf{X}_1, \ldots, \mathbf{X}_K]$. The dimensionality of the product space could be very high, especially when K is large. Dimensionality reduction is needed to alleviate the "curse of dimensionality," the phenomenon that the number of data points needed to learn a classifier increases exponentially with the dimension of the representation space [8, 9].

3.3 Dimensionality Reduction

PCA is widely used to create low-dimensional representations of high-dimensional data. PCA constructs a new coordinate system in such a way that the span of the first k principal components (PCs) is the k-dimensional linear subspace that best summarizes (in the sense of squared error) the data. PCA is unsupervised, so applying PCA within classes may result in different PCs than applying PCA to the entire data set. Furthermore, the PCs that best summarize variation in the data may not be the dimensions that best discriminate between classes, as in the case of "parallel cigars." In contrast to PCA, LDA uses the class labels to find the best dimensions for class discrimination. Unfortunately, LDA may perform badly in high-dimensional spaces (cf. Trunk, 1979). To address this difficulty, Belhumeur, Hespanha and Kriegman [38] proposed a two-step procedure (LDA $\circ$ PCA) in which PCA is first used to reduce dimensionality, after which LDA is used to train a linear classifier;

however, if the PCA step discards dimensions that are important for discrimination, then LDA $\circ$ PCA may also perform badly. To remedy this failing, we develop an alternative PCA step that we call the J-function procedure. The essential idea is to extract (class-conditional uncorrelated) PCs based on their ability to discriminate rather based on how much variation they contain. As explained in Figure 3.2, LDA $\circ J$ improves on LDA $\circ$ PCA.

3.3.1 J-function

Consider data matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{n \times p}$ and class vector $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ with $\{0, 1\}$ entries. The goal is to find a d-dimensional ($d < p$) representation of $\tilde{\mathbf{X}}$ that contains the most class information. The J-function procedure can be described via the following steps:

1. Compute the pooled sample covariance matrix $\mathbf{S} = \pi \mathbf{S}_1 + (1 - \pi)\mathbf{S}_0$, where $\pi = \sum_{i=1}^{n} y_i/n$ and $\mathbf{S}_j$ is the sample covariance matrix for class j.

2. Perform eigenvalue decomposition on $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ and transform $\tilde{\mathbf{X}}$ to $\mathbf{X} = \tilde{\mathbf{X}}\mathbf{U}$. (Assume that the columns of $\tilde{\mathbf{X}}$ have been centered to have mean zero.)

3. Compute the J value for the ith dimension of $\mathbf{X}$

$$J_i = \begin{cases} |\boldsymbol{m}_{\mathbf{1}_i} - \boldsymbol{m}_{\mathbf{0}_i}|/\sqrt{\lambda_i}, & \lambda_i > 0, \\ 0, & \lambda_i = 0, \end{cases}$$

   where $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ are the sample means of classes 0 and 1, respectively, and λ_i is the ith largest eigenvalue of $\mathbf{S}$.

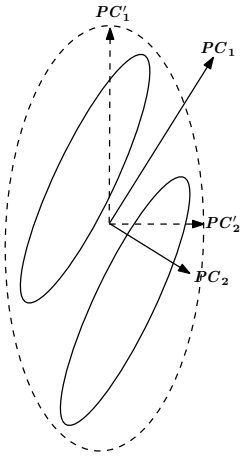


Figure 3.2: The solid ellipses represent the data from the two classes. The dashed ellipse represents the entire dataset, on which performing PCA reports PC'_1 and PC'_2 as the 1st and 2nd principal components, respectively. The J-function approach first finds the principal components PC_1 and PC_2 by performing eigenvalue decomposition on the pooled covariance matrix. It then computes the J value, a measure of discriminative power, for each PC and reorders the PCs by the J values associated with them. PCs with larger J values will have higher rank in the order. For this dataset $J_1 < J_2$ (J_i is the J value of the PC_i). Therefore the final first and second PCs generated by the J-function approach are PC_2 and PC_1, respectively. Notice that for low dimensional data, the J-function approach is essentially the same as LDA. For high dimensional data, where LDA has problems, one can use the two-step approach, LDA $\circ$ J.

4. Obtain $\mathbf{X}^J$ by reordering the dimensions of $\mathbf{X}$ according to the J values—dimensions with larger values have higher rank in the order. Let $\mathbf{X}_d^J$ be the first d dimensions of $\mathbf{X}^J$.

Then $\mathbf{X}_d^J$ is the d-dimensional representation of $\tilde{\mathbf{X}}$ obtained by the J-function approach. In summary, this approach first projects data onto the principal axes of the pooled covariance matrix to obtain conditionally uncorrelated (given class label Y) PCs, then ranks them by a quantity J, which is the absolute value of a t-statistic, and finally includes only these PCs with large J values. Devroye et al. [39, p. 566] sketched a similar idea to rank (class) independent Gaussian distributed features. We show in the following theorem that, under the assumption of mixture of two multivariate Gaussian distributions with equal covariance matrices, $\mathbf{X}_d^J$ contains the most class information among a collection of p-dimensional projections of $\tilde{\mathbf{X}}$. That is, for the transformed data $\mathbf{X}$, the best d individual PCs constitute the best subset of d PCs.

Theorem 3.3.1. Suppose that (X, Y) is distributed as F_{XY}, where $X : \Omega \to \mathbb{R}^p$, Y is Bernoulli distributed with parameter π, and that the conditional distribution of $X|Y = j$ is $N(\boldsymbol{\mu}_j, \Sigma)$. Let $f : \mathbb{R}^p \to \mathbb{R}^d$ be the function that projects X onto the space spanned by any d of the p eigenvectors of Σ, where $d < p$, and let f^* be the projection function deduced from the above J-function procedure. If $L^*_{f(X)}$ and $L^*_{f^*(X)}$ denote the Bayes error probabilities for $(f(X), Y)$ and $(f^*(X), Y)$ respectively, then

$$L^*_{f^*(X)} \leq L^*_{f(X)}. \tag{3.1}$$

Proof. We assume that $\Sigma = I_p$, where the dimensions of X are ordered (from largest to smallest) by the magnitudes of the elements of $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. To see that this assumption entails no loss of generality, first note that if Σ is of less than full rank, then X can be projected into a lower dimensional space in which the covariance matrix is of full rank. Second, we can assume that $\Sigma = I_p$ because there exists a matrix A for which $(AX|Y = j) \equiv (X_A|Y = j) \sim N(A\boldsymbol{\mu}_j, I_p)$. Hence, any linear projection function of X can be written as $f(X) = f(A^{-1}X_A) \triangleq f_A(X_A)$. Third, assuming that $\Sigma = I_p$, we can further assume that the dimensions are in any prescribed order—we simply apply the same argument with A chosen to be a suitable permutation matrix.

Then the projection $f^*(X) = T^*X$ chooses the first d dimensions of X. That is, T^* is a $d \times p$ matrix with all 1's on the diagonal of its leftmost $d \times d$ block and 0's elsewhere; and the projection $f(X) = TX$ chooses any d dimensions of X. That is, T has same columns as T^* does, but with different order. By the previous assumptions, we have

$$\|T^*\boldsymbol{\mu}_1 - T^*\boldsymbol{\mu}_0\| \geq \|T\boldsymbol{\mu}_1 - T\boldsymbol{\mu}_0\|,$$

which implies

$$L^*_{T^*X} \leq L^*_{TX}$$

and

$$L^*_{f^*(X)} \leq L^*_{f(X)}.$$

□

In practice, the sample covariance matrix $\mathbf{S}_j$ usually is not an accurate and reliable estimator of the population covariance matrix, especially when the data have a large number of dimensions but contain comparatively few samples. This will decrease J-function's power in determining discriminative dimensions. To alleviate this problem, in the following experiments section, we used Schäfer and Strimmer's [40] shrinkage estimation of covariance matrix to obtain $\mathbf{S}_j$.

3.4 Experiments

3.4.1 Simulation Experiment

To illustrate the J-function approach and its advantages, we conduct a simple simulation experiment. Let $F_{XY} = \pi N(\boldsymbol{\mu}, \Sigma) + (1-\pi)N(-\boldsymbol{\mu}, \Sigma)$, where

$$\pi = \frac{1}{2},\ \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \\ \boldsymbol{\mu}_b \end{pmatrix},\ \boldsymbol{\mu}_a = \boldsymbol{\mu}_b = (1,1,1,1,1,0,\cdots,0)^T \in \mathbb{R}^{40},$$

$$\Sigma = \begin{pmatrix} \Sigma_a & \mathbf{0} \\ \\ \mathbf{0} & \Sigma_b \end{pmatrix},\ \Sigma_a = \text{diag}(1,\ldots,40),\ \Sigma_b(i,j) = \frac{\sqrt{ij}}{2^{|i-j|}},\ i,j = 1,\ldots,40.$$

Notice that the two multivariate Gaussian distributions have the same covariance matrix Σ, and the only difference is in the means. The reason we construct Σ in this special form is that we try to simulate two different data sources, analogous to the Cartesian product of embeddings of $K = 2$ dissimilarity matrices.

We randomly draw $2n$ samples $\mathbf{X} = [x_1, \ldots, x_{2n}]^T$ from F_{XY}, with the first n samples

as training data and the rest as testing data. We then perform dimensionality reduction, build LDA based on the training data and then classify the testing observations. For comparison, we consider PCA and the J-function method in the dimensionality reduction step. In addition, we let p, the reduced dimensionality, range from 1 to 80 ($p = 80$ means no dimensionality reduction). Notice that the J-function approach is a supervised dimensionality reduction method. That is, it utilizes the class label information. We perform two experiments: in the first one we use only the class labels of the training observations, and in the second one we use the class labels of both the training and the testing observations. The following LDA step remains the same for both experiments. Because the dimensionality reduction step (although not the LDA step) in the second experiment uses the testing class labels, this experiment leads to an overly optimistic classification error. It provides a (meaningful) lower bound on the error from the first (valid) experiment. We call the dimensionality reduction method used in the first experiment the J-function approach and that used in the second experiment the $\underline{J}$-function approach. We use L_J and $L_{\underline{J}}$ to denote the classification errors corresponding to the J-function and $\underline{J}$-function.

We repeat the above process 100 times each for three different sample sizes: $n = 100$, $n = 200$ and $n = 400$. Let $\bar{L}_P(d)$, $\bar{L}_J(d)$ and $\bar{L}_{\underline{J}}(d)$ denote the means of the estimated classification errors resulting from the d-dimensional data, which are obtained through PCA, the J-function and $\underline{J}$-function procedures, respectively. Let $\bar{L}_\emptyset$ denote the mean of the estimated classification error when using LDA only, that is, no dimensionality reduction. This simulation experiment shows that (1) for classification, for fixed reduced dimension-

ality $d < 80$, the J-function procedure outperforms PCA and no dimensionality reduction: $\bar{L}_{\underline{J}}(d) < \bar{L}_J(d) < \bar{L}_\emptyset \leq \bar{L}_P(d)$, for all $d < 80$; (2) the J-function procedure works better than PCA, when both use optimal reduced dimensionalities: $\min_d \bar{L}_{\underline{J}} < \min_d \bar{L}_J < \min_d \bar{L}_P$; (3) for classification, the $\underline{J}$-function procedure provides a lower bound on the error, and the difference between the J- and $\underline{J}$-procedure, $\bar{L}_J(d) - \bar{L}_{\underline{J}}(d)$, decreases as the sample size increases. We plot the results in Figure 3.3.

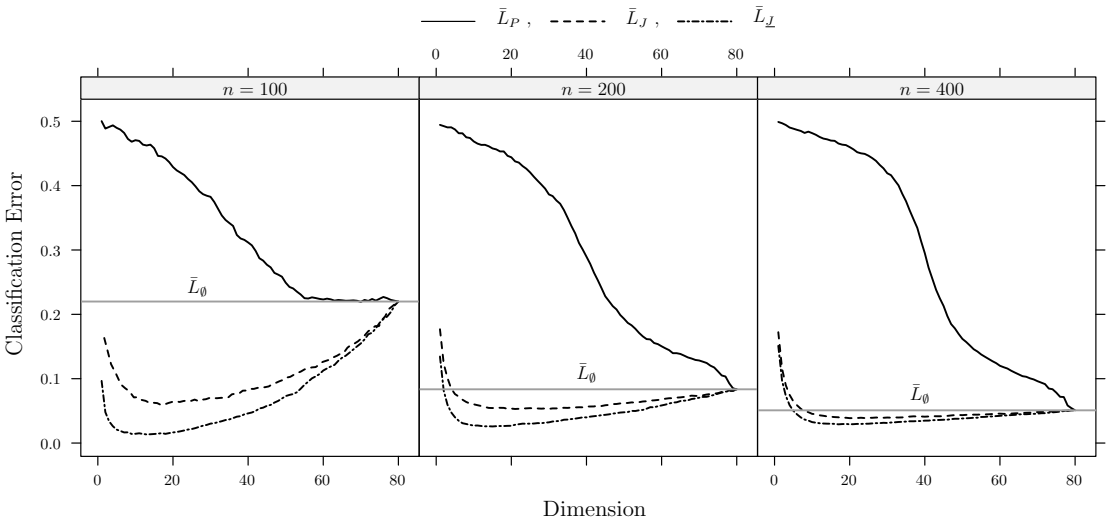
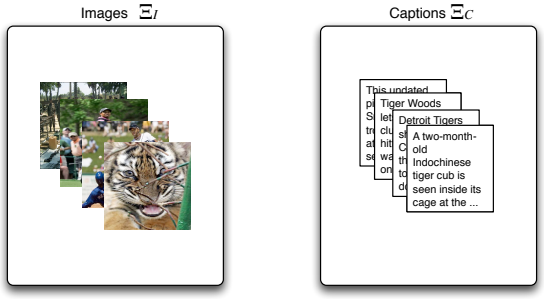


Figure 3.3: Let $\bar{L}_P(d)$, $\bar{L}_J(d)$ and $\bar{L}_{\underline{J}}(d)$ denote the mean of the estimated classification errors resulting from the d-dimensional data, which are obtained through PCA, the J- and $\underline{J}$-function procedure, respectively. Let $\bar{L}_\emptyset$ denote the mean of the estimated classification error when using LDA only, that is no dimensionality reduction. These plots depict that (1) $\bar{L}_{\underline{J}}(d) < \bar{L}_J(d) < \bar{L}_\emptyset \leq \bar{L}_P(d)$, for all $d < 80$; (2) $\min_d \bar{L}_{\underline{J}} < \min_d \bar{L}_J < \min_d \bar{L}_P$; (3) $\bar{L}_J(d) - \bar{L}_{\underline{J}}(d)$ decreases as the sample size increases.

3.4.2 The Tiger Data

In this section, we present an example of combining image and caption data. The data are 140,577 images and captions collected from the Yahoo! Photos website. We selected 1,600 pairs by using the query word "tiger" on captions. The "tiger" data were manually labeled into 6 classes based only on captions (see Figure 3.4). For simplicity we consider the problem of discriminating between the two classes of "Tiger Woods" and "Tamil Tigers".

The image, text, and joint image-text spaces are rather complicated, so there is no



Label	#
Animal tiger	148
Detroit Tigers baseball team	145
Tiger Woods the golfer	897
Tamil Tigers soldiers of Sri Lanka	330
Leicester Tigers rugby team	48
Others	32

Figure 3.4: The "tiger" data. Each observation consists of an image/caption pair.

simple way to combine them directly. We use the first and second order pixel derivatives [41, 42] on the images, and the mutual information [43] on the captions to extract features from each space. We then compute R_{ij}, the random forest proximity [44], for each pair of observations and use $1 - R_{ij}$ as the dissimilarity measure to generate two dissimilarity matrices, Δ_C and Δ_I. Classical multidimensional scaling (CMDS) [17–19] is used to embed Δ_C into $\tilde{\mathbf{X}}_C \in \mathbb{R}^{n \times p(n)}$ and Δ_I into $\tilde{\mathbf{X}}_I \in \mathbb{R}^{n \times p(n)}$. We used $p(n) = 1000$. Because the coordinates of the embedding constructed by CMDS are its PCs, it is easy to perform PCA. Figure 3.5 displays a scree plot of variances. The automatic dimensionality

selection of [45] was used to determine reduced dimensionalities of $d_C = 473$ for caption and $d_I = 152$ for image. These choices err on the side of anti-parsimony, but further dimensionality reduction in the Cartesian product space will follow. For comparison, we considered also the J-function on $\tilde{\mathbf{X}}_C$ and $\tilde{\mathbf{X}}_I$ (Zhu and Ghodsi's approach was used also to determine reduced dimensionality). For the Cartesian product, we separately performed PCA, the J-function and $\underline{J}$-function to reduce the dimensionality. A linear classifier was built on caption alone, image alone and their combination, respectively. Leave-one-out cross-validation was used to estimate classification errors. Figure 3.4.2 shows the above procedures.

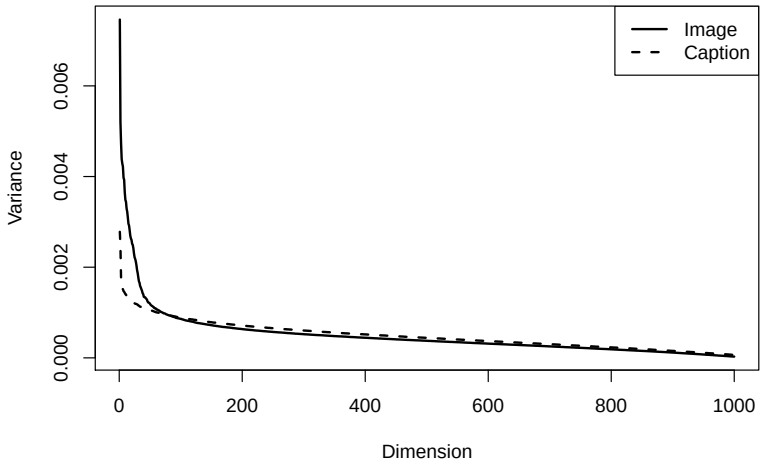


Figure 3.5: "Tiger" data. Using the classical multidimensional scaling to embed both Δ_C and Δ_I into 1000-dimensional Euclidean space. The scree plot depicts the variance for each dimension.

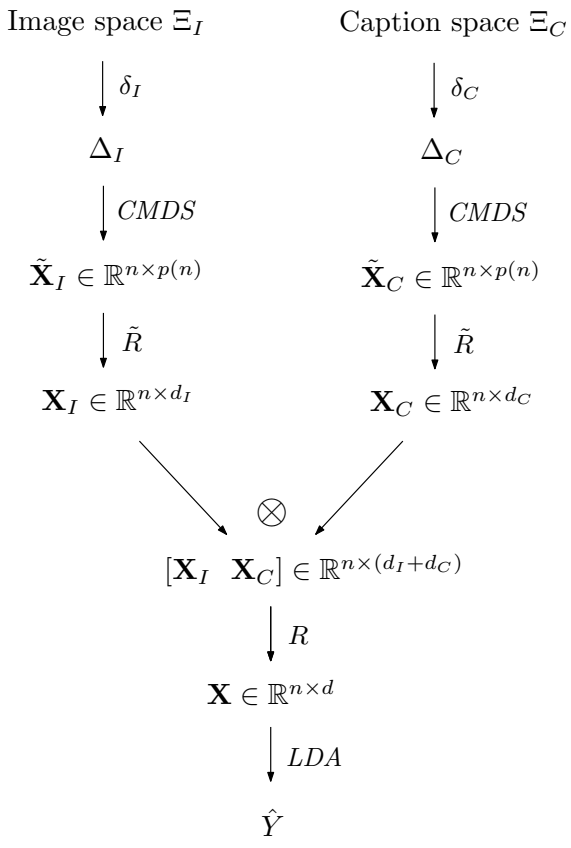


Figure 3.6: "Tiger" data. We combined image and caption data using dissimilarity representation: image and caption data were transformed into dissimilarity matrices Δ_I and Δ_C, which were then embedded into $p(n)$-dimensional Euclidean space. Dimensionality reduction procedures $\tilde{R}$ and R were performed on each embedding and then on the Cartesian product, respectively. Finally, a linear classifier was trained. We considered $\tilde{R} \in \{\text{PCA}\,, J\text{-function}\}$ and $R \in \{\text{PCA}, J, \underline{J}, \emptyset\}$, where $\emptyset$ means no dimensionality reduction.

Table 3.1 reports classification errors for several procedures. The results suggest that (i) the two step procedure LDA $\circ$ J works better than LDA only (no dimensionality reduction) and than LDA $\circ$ PCA; (ii) LDA $\circ$ J is better than LDA $\circ$ PCA$'$, which is the same as LDA $\circ$ PCA, except using the reduced dimensionalities determined by the J-function; (iii) LDA $\circ$ $\underline{J}$ is apparently better than the other procedures, but recall that $\underline{J}$ uses testing class labels (the error rate for LDA $\circ$ $\underline{J}$

We used McNemar's test to validate these statements and show the results in Table 3.2.

We also applied the two-step procedure LDA$\circ$ R ($R \in \{\emptyset, \text{PCA}, J\text{-function}\}$) on the two classes of "animal" versus "baseball". The resulted leave-one-out cross-validation classification errors are: $L(\emptyset(\mathbf{X})) = 0.0751,\ L(\text{PCA}(\mathbf{X})) = 0.0648$ and $L(J(\mathbf{X})) = 0.0444$. At level of significance $\alpha = 0.05$, McNemar's test shows that PCA is not statistically significantly different from no dimensionality reduction (p-value = 0.2249); J-function is statistically significantly better than no dimensionality reduction (p-value = 0.0038); J-function is not statistically significantly better than PCA (p-value = 0.0745).

3.5 Conclusion

The main obstacles to combining multiple dissimilarity matrices via the Cartesian product of their embeddings are the curse of dimensionality and the parallel cigars phenomenon. We have proposed a new supervised dimensionality reduction approach and show by theo-

Data	R	Low-dim. Data	Dimensionality	Error
Image: $\tilde{\mathbf{X}}_I \in \mathbb{R}^{n\times 1000}$	PCA	$\mathbf{X}_I^P$	152	0.1491
	J-function	$\mathbf{X}_I^J$	$\overline{62}$	0.1133
Caption: $\tilde{\mathbf{X}}_C \in \mathbb{R}^{n\times 1000}$	PCA	$\mathbf{X}_C^P$	473	0.1883
	J-function	$\mathbf{X}_C^J$	$\overline{384}$	0.1345
Combination: $\mathbf{X}^P = [\mathbf{X}_I^P\ \mathbf{X}_C^P]$	$\emptyset$		625	0.1557
	PCA		160	0.1125
	PCA′		$\overline{205}$	0.1182
	J-function		$\overline{205}$	0.0815
	$\underline{J}$-function		71	0.0171
Combination: $\mathbf{X}^J = [\mathbf{X}_I^J\ \mathbf{X}_C^J]$	J-function		$\overline{186}$	0.0864

Table 3.1: "Tiger" data. We use the two-step approach LDA $\circ$ R—perform dimensionality reduction procedure R and then train linear classifier on the low-dimensional data—together with leave-one-out cross validation to estimate classification error. The notation $\emptyset$ means no dimensionality reduction and PCA′ is PCA but using the dimensionalities determined by the J-function procedure. The bar on dimensionality means that the corresponding number is the average of dimensionalities used in leave-one-out cross validation by J-function.

$\boldsymbol{H_A}$	$\boldsymbol{p}$**-value**
$L(J(\tilde{\mathbf{X}}_I)) < L(\text{PCA}(\tilde{\mathbf{X}}_I))$	4.803e-07
$L(J(\tilde{\mathbf{X}}_C)) < L(\text{PCA}(\tilde{\mathbf{X}}_C))$	5.215e-04
$L(\text{PCA}(\mathbf{X}^P)) < L(\emptyset(\mathbf{X}^P))$	4.643e-05
$L(J(\mathbf{X}^P)) < L(\text{PCA}(\mathbf{X}^P))$	8.095e-05

Table 3.2: "Tiger" data. McNemar's test is used to compare the dimensionality reduction procedures $R \in \{\emptyset,\ \text{PCA}, J, \underline{J}\}$. The alternative hypothesis H_A is listed in the first column and the corresponding null hypothesis replaces "$<$" with "$\geq$". We use $L(\mathbf{X})$ to denote the LDA leave-one-out cross validation classification error based on data $\mathbf{X}$, and use $R(\mathbf{X})$ to denote the low-dimensional data obtained by the procedure R. The definitions of various forms of $\mathbf{X}$ can be found in Table 3.1. These p-values, together with Table 3.1, show that (i) LDA $\circ$ J works better than LDA only (i.e., no dimensionality reduction) and better than LDA $\circ$ PCA; (ii) LDA $\circ$ J is better than LDA $\circ$ PCA′, which is the same as LDA $\circ$ PCA except using the reduced dimensionalities determined by the J-function; (iii) LDA $\circ$ $\underline{J}$ is apparently better than the other procedures, but recall that $\underline{J}$ uses testing class labels (the error rate for LDA $\circ$ $\underline{J}$ is a meaningful lower bound on the error rate of LDA $\circ$ J).

rem, simulation and real data experiments that the J-function approach can improve classification performance compared to the alternatives of principal components analysis and no dimensionality reduction at all. The proposed approach is not specific to this type of data and can serve as a general dimensionality reduction technique. It is particularly useful when (1) the data is high-dimensional and (2) many dimensions of the data have similar variances and PCA is liable to fail in extracting discriminative dimensions.

The proposed dimensionality reduction approach has been developed for the simple

two-class problem. One way to extend it to K $(K > 2)$ classes is the following: (1) project data onto the principal axises of the pooled sample covariance matrix; (2) calculate the absolute differences between each class mean and the overall mean; (3) normalize and weight them by corresponding eigenvalues and class proportions, respectively, to obtain a $K \times p$ matrix $\boldsymbol{J}$; (4) finally use the column sums of $\boldsymbol{J}$ to rank and choose principal components. Alternatively, the two-step LDA $\circ$ J approach for $K > 2$ classes can be addressed in two other ways: (1) perform LDA $\circ$ J on each pair of classes and combine the $\binom{K}{2}$ classifiers in the end [46, 47]; or (2) perform LDA $\circ$ J on each pair of "class i versus not class i" and combine the K classifiers in the end.

Chapter 4

Fusion and Inference from Multiple Data Sources in a Commensurate Space

Given objects measured under multiple conditions—e.g., indoor lighting versus outdoor lighting for face recognition, multiple language translation for document matching, etc.—the challenging task is to perform data fusion and utilize all the available information for inferential purposes. We consider two exploitation tasks: (1) how to determine whether a set of feature vectors represent a single object measured under different conditions; and (2) how to create a classifier based on training data collected under one condition in order to classify objects measured in other conditions. The key to both problems is to transform all sets of feature vectors into one commensurate space, where the (transformed) feature vectors are comparable and would be treated as if they were collected under the same condition. Toward this end, we study Procrustes analysis and develop a new approach, which

uses the interpoint dissimilarities for each condition. We impute the dissimilarities between measurements of different conditions to create one omnibus dissimilarity matrix, which is then embedded into Euclidean space. We illustrate our methodology on English and French documents collected from Wikipedia, demonstrating superior performance compared to that obtained via standard Procrustes transformation.

4.1 Introduction

Information fusion techniques aim to merge information from multiple data sources in order to achieve more accurate inferences than using each single source alone. Information fusion has been a hot research field with various applications [1–4].

In general, the most often used information fusion approaches can be summarized into two categories: feature level fusion and decision level fusion. In feature level fusion, feature vectors extracted from different data sources are combined into the Cartesian product space, directly [5] or via some data transformation procedures [3]. Decision level fusion involves combining results obtained separately from all data sources. An ensemble of classifiers is one such example, as is track fusion [7]. The advantage of these two types of information fusion stems from the fact that multiple sets of feature vectors extracted from the same set of objects usually reflect different characteristics of patterns. By fusing multiple disparate data sources, one generates a more complete representation of the space in which the objects live, and hence has more power for inferential tasks such as hypothesis

testing, classification, etc.

In this work, we consider information fusion from a different perspective. Suppose that objects are measured under multiple conditions—e.g. indoor lighting versus outdoor lighting for face recognition, multiple language translation for document matching, etc. The challenging questions are: (1) how to determine whether a set of feature vectors represent a single object measured under different conditions? For example, whether pictures taken under different lighting conditions are the photos of the same individual or not; and (2) how to create a classifier based on training data measured under one condition, and use it to classify objects measured in other conditions? We refer the two problems as the implicit translation problem and the classification problem, respectively. A direct approach would involve finding the underlying mappings among all the spaces of measurements and transform all these measurements into one commensurate space through the derived mappings. In this commensurate space, all transformed feature vectors are treated as if they were from the same data source. The solutions to both questions will then be straightforward. In real applications, finding the mappings among all spaces of measurements is usually difficult. In fact, it is possible to fuse multiple spaces into one commensurate space without using the mappings among these spaces. (Generalized) Procrustes analysis is one potential solution. Consider a set of objects, each of which is measured under K ($K \geq 2$) conditions, yielding K feature vectors. Assuming all the feature vectors have been column centered, Procrustes solution rotates (possibly with dilation) the feature vectors to best match each other, and thereby defines a commensurate space.

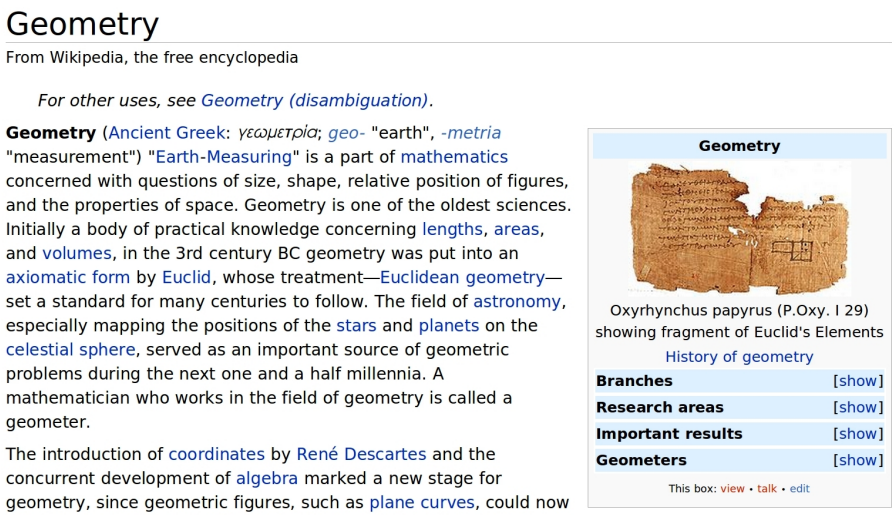
CHAPTER 4. FUSION IN A COMMENSURATE SPACE

The raw data in text or image analysis are usually high-dimensional. Dissimilarity analysis is one of the commonly applied approaches for finding low-dimensional representation of such data. Usually in dissimilarity analysis, one first calculates interpoint dissimilarities to obtain a dissimilarity matrix, and then embeds it into a low dimensional space via multidimensional scaling. We use a collection of Wikipedia documents to illustrate the two problems (implicit translation and classification) and the solutions. The two step approach, which we refer to as the P-approach, first embeds dissimilarity matrices derived from different data sources and then utilizes a Procrustes transformation on the embeddings to make them commensurate. We propose an approach that simultaneously embeds all dissimilarity matrices and finds the commensurate space. In this approach, dissimilarity matrices from different data sources are put onto the diagonal of an omnibus matrix, whose off-diagonal entries are imputed. Embedding this omnibus matrix results in feature vectors in one commensurate space. We refer this approach as the W-approach. Both approaches are studied in this work, and the results on Wikipedia example show that the W-approach leads to larger powers in testing and higher accuracy in classification, compared to the P-approach.

In Section 4.2, we describe the Wikipedia data set, the derivation of dissimilarity matrices, and the implicit transformation and classification problems. Section 4.3 details the traditional Procrustes solution and the proposed W-approach. The results are given in Section 4.4. Section 4.5 provides conclusions.

4.2 Data

Wikipedia is an open-source Encyclopedia that is written by a large community of users (everyone who wants to, basically). There are versions in over 200 languages, with various amounts of content. The full data for the Wikipediae are freely available for download. A Wikipedia document has one or more of: title, unique ID number, text—the content of the document, images, internal links—links to other documents, external links—links to other content elsewhere on the web, and language links—links to "the same" document in other languages. Figure 4.1 shows an English Wikipedia document titled "Geometry". The multilingual Wikipediae provide a good testbed for developing methods for analysis of text, implicit translation, and fusion of text and graph information.



Geometry

From Wikipedia, the free encyclopedia

For other uses, see Geometry (disambiguation).

Geometry (Ancient Greek: *γεωμετρία*; *geo-* "earth", *-metria* "measurement") "Earth-Measuring" is a part of mathematics concerned with questions of size, shape, relative position of figures, and the properties of space. Geometry is one of the oldest sciences. Initially a body of practical knowledge concerning lengths, areas, and volumes, in the 3rd century BC geometry was put into an axiomatic form by Euclid, whose treatment—Euclidean geometry—set a standard for many centuries to follow. The field of astronomy, especially mapping the positions of the stars and planets on the celestial sphere, served as an important source of geometric problems during the next one and a half millennia. A mathematician who works in the field of geometry is called a geometer.

The introduction of coordinates by René Descartes and the concurrent development of algebra marked a new stage for geometry, since geometric figures, such as plane curves, could now

Figure 4.1: "Geometry", an example of English Wikipedia documents. In general, a Wikipedia document has one or more of: title, unique ID number, text, images, internal links, external links, and language links.

4.2.1 Dissimilarities from Graph Structure and Textual Content

Let $G = (V, E)$ be a (directed) graph, where V is the set of nodes—Wikipedia documents, and E is the set of edges—the links within the documents. We consider two Wikipediae, English and French. A subset of the English and French Wikipediae is extracted such that there is an 1-1 correspondence between English documents and French documents. From the English subset, we take the (directed) 2-neighborhood of the document "Algebraic Geometry", yielding set $\boldsymbol{E} = \{\boldsymbol{x}_{1,0}, \ldots, \boldsymbol{x}_{n,0}\}$ ($n = 1382$). The associated documents in French constitute set $\boldsymbol{F} = \{\boldsymbol{x}_{1,1}, \ldots, \boldsymbol{x}_{n,1}\}$. Thus, the English graph with nodes in $\boldsymbol{E}$ is connected by construction, but the French graph with nodes in $\boldsymbol{F}$ need not be connected (and in fact it is not). We consider two types of data, both of which are given in the form of dissimilarity matrices denoted generically as $\mathbf{D}_0$ and $\mathbf{D}_1$: (1) dissimilarity matrices $\mathbf{G}_0$ and $\mathbf{G}_1$, developed from the graph structures of $\boldsymbol{E}$ and $\boldsymbol{F}$ respectively; (2) dissimilarity matrices $\mathbf{T}_0$ and $\mathbf{T}_1$, obtained from the textual contents of $\boldsymbol{E}$ and $\boldsymbol{F}$ respectively.

To get dissimilarity matrices from graph structure, the adjacency matrices $\mathbf{A}_0$ and $\mathbf{A}_1$ are first created from $\boldsymbol{E}$ and $\boldsymbol{F}$. An adjacency matrix is a square binary matrix, with 1 in position (i, j) only when the ith document contains an link to the jth document. Dissimilarity matrices $\mathbf{G}_0$ and $\mathbf{G}_1$ are then derived from $\mathbf{A}_0$ and $\mathbf{A}_1$, with (i, j) entry as the number of steps it takes to reach node j from node i. By the nature of the graphs, the elements of

$\mathbf{G}_0$ take values in $\{0, \ldots, 4\}$, while the elements of $\mathbf{G}_1$ take values in $\{0, \ldots, 1384\}$, with 1384 meaning no directed path between two nodes. Because it is computationally expensive to develop $\mathbf{G}_1$, in practice we assign the value 6 to $\mathbf{G}_1(i, j)$ if it takes more than 4 steps to reach node j from node i. Finally, $\mathbf{G}_0$ and $\mathbf{G}_1$ are symmetrized by averaging the corresponding lower- and upper-triangle entries, respectively.

For dissimilarity matrices of textual content, we use Lin & Pantel's approach [43,48] on Wikipedia documents $\boldsymbol{E}$ and $\boldsymbol{F}$ to obtain two mutual information feature matrices. Rare-word discounting [43] is then applied to reduce the impact of infrequent words. Given feature vectors of two documents $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, a dissimilarity function ρ is defined as $\rho(\boldsymbol{a}, \boldsymbol{b}) = 1 - (\boldsymbol{a} \cdot \boldsymbol{b})/(\|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2)$. Employing ρ on the two feature matrices of $\boldsymbol{E}$ and $\boldsymbol{F}$ separately results in two dissimilarity matrices $\mathbf{T}_0$ and $\mathbf{T}_1$.

When a new English document $\boldsymbol{y}_0$ and a new French $\boldsymbol{y}_1$ are provided, we have access to the dissimilarities (for both graph structure and textual content) between $\boldsymbol{y}_0$ and $\boldsymbol{x}_{i,0}$, and those between $\boldsymbol{y}_1$ and $\boldsymbol{x}_{i,1}$, $i = 1, \ldots, n$. Therefore the Wikipeida data set contains four dissimilarity matrices $\mathbf{G}_0, \mathbf{G}_1, \mathbf{T}_0$ and $\mathbf{T}_1$, and each new document $\boldsymbol{y}_k$ will be represented by a dissimilarity vector $\{\delta(\boldsymbol{y}_k, \boldsymbol{x}_{i,k})\}_{i=1}^n,\ k = 0, 1$. ($\delta$ is a dissimilarity function.)

4.2.2 Implicit Translation and Classification

An implicit translation of a document, unlike a word-level or a real translation in any normal sense, is an association with another document in a different language that is on the same topic. In our framework, we treat each topic as an object with measurements

(documents) taken under different conditions (languages). That is, topics are represented by documents of different languages. Consider the two collections of matched Wikipedia documents $\boldsymbol{E} = \{\boldsymbol{x}_{1,0}, \ldots, \boldsymbol{x}_{n,0}\}$ and $\boldsymbol{F} = \{\boldsymbol{x}_{1,1}, \ldots, \boldsymbol{x}_{n,1}\}$. Let $\boldsymbol{x}_{i,0} \sim \boldsymbol{x}_{i,1}$ denote that the English document $\boldsymbol{x}_{i,0}$ and the French document $\boldsymbol{x}_{i,1}$ are matched—they are the measurements (under $K = 2$ conditions) of the same topic. The goal of implicit translation is to determine whether a match is present between two new documents $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$. That is, we consider the hypothesis testing:

$$H_0 : \boldsymbol{y}_0 \sim \boldsymbol{y}_1 \text{ versus } H_A : \boldsymbol{y}_0 \nsim \boldsymbol{y}_1$$

Notice that we assume the two new documents represent a matched pair under H_0. This allows us to control the probability of missing a true match. This is practical in many applications where computer algorithms are used to eliminate easily rejected pairs and the remaining possibly matched pairs will be manually examined.

The second problem is to classify French documents by a classifier trained on English documents. Formally, consider two manifolds, Ξ_0 and Ξ_1. Let

$$(X, C, Z) \sim F_{X,C,Z},$$

$$C : \Omega \to \mathcal{C} \cup \tilde{\mathcal{C}},$$

$$Z : \Omega \to \{0, 1\},$$

$$X|Z = z : \Omega \to \Xi_z,$$

where $\mathcal{C}$ and $\tilde{\mathcal{C}}$ are two disjoint sets of class labels. Suppose the following training data are available

$$\mathcal{T}_0 = \{(x_i, c_i \in \mathcal{C}, z_i = 0),\ i = 1, \ldots, n_0\},$$

$$\mathcal{T}_1 = \{(x_i, c_i \in \mathcal{C}, z_i = 1),\ i = 1, \ldots, n_1\},$$

$$\tilde{\mathcal{T}}_0 = \{(x_i, c_i \in \tilde{\mathcal{C}}, z_i = 0),\ i = 1, \ldots, m_0\}.$$

That is, there are training data from all the classes $\mathcal{C} \cup \tilde{\mathcal{C}}$ in space Ξ_0, but in space Ξ_1 only training data from classes $\mathcal{C}$ are available. We are interested in creating a classifer g based on the training data and use it to classify future observations in Ξ_1 into one of the classes in $\tilde{\mathcal{C}}$. Figure 4.2 depicts the classification problem.

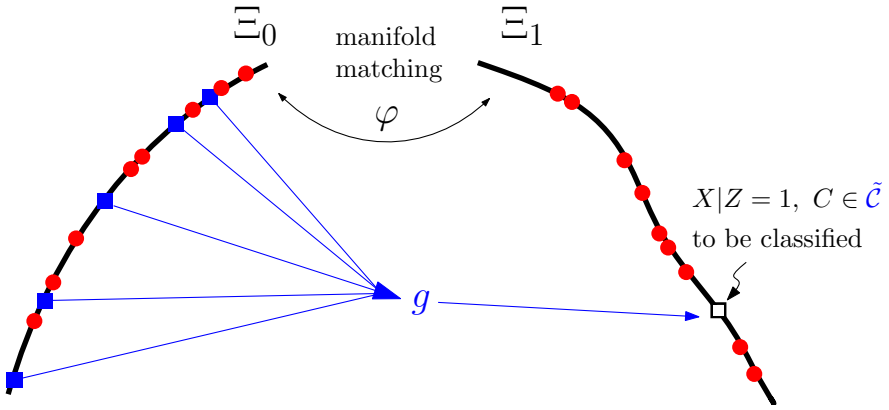


Figure 4.2: Classification problem. In space Ξ_0 training data from classes $\mathcal{C}$ (red) and $\tilde{\mathcal{C}}$ (blue) are available, while in space Ξ_1 only training data from classes $\mathcal{C}$ are available. We are interested in training a rule g to classify objects of classes $\tilde{\mathcal{C}}$ in space Ξ_1. It is impossible to directly create such a classifier in Ξ_1 due to lack of training data.

We consider Ξ_0 and Ξ_1 to be the English and French Wikipedia document space, respec-

tively. The 1382 Wikipedia documents are labeled into 5 groups. The two disjoint sets of class labels are $\mathcal{C} = \{0, 1, 2\}$ and $\tilde{\mathcal{C}} = \{3, 4\}$. We are interested in finding a way to create a classifier based on English documents and use it to classify French documents.

4.3 Methods

For the implicit translation problem, suppose that there is a way to embed $\boldsymbol{E} \in \Xi_0$ and $\boldsymbol{F} \in \Xi_1$ into a commensurate space Ξ_c, where the embeddings of English and French documents would be treated equally, as if they were collected under the same condition. We can embed the two new documents $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$, referred to as the out-of-sample documents, into the space Ξ_c. Whether a match is present is then determined by examining the distance between the embeddings of $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$, with a large distance being evidence against H_0. There are two ways to determine critical values. The naïve way is to treat the distances between the embeddings of matched pairs in $\mathbf{E}$ and $\mathbf{F}$ as the ground truth, and use the $100(1-\alpha)$th percentile as the critical value for a level α test. However, this method does not always lead to large powers, because the distribution of the distances between out-of-sample embeddings is usually slightly different from that of the original embeddings, even under the matched assumption H_0. Another way of obtaining critical values is by means of Monte Carlo simulation: (i) randomly choose a pair of matched documents $\boldsymbol{x}_{i,0}$ and $\boldsymbol{x}_{i,1}$ from $\boldsymbol{E}$ and $\boldsymbol{F}$, and treat them as out-of-sample documents; (ii) embed the selected documents into the space Ξ_c, and compute their distance; and (iii) repeat (i—ii) to obtain an empirical

distribution of such distances. The critical value for a level α test is then calculated as the $100(1-\alpha)$th percentile of this empirical distribution. We use the latter method in this work and get larger powers than using the naïve approach.

For the classification problem, suppose a commensurate space Ξ_c could be obtained through $\mathcal{T}_0$ and $\mathcal{T}_1$—English and French documents of classes $\mathcal{C}$. We can embed the training English documents $\tilde{\mathcal{T}}_0$ and the new French documents into the space Ξ_c. In the commensurate space Ξ_c, building classifier g based on English documents with labels in $\tilde{\mathcal{C}}$ and using it to classify new French documents are then straightforward.

Therefore the key to both problems is: how shall we determine the commensurate space Ξ_c and how shall we embed new documents into this space?

4.3.1 Procrustes Transformation

Procrustes analysis [49, and references contained therein] is to transform a configuration of points (source) to another (target) as closely as possible in the least-square sense. The permitted transformations are any combination of dilation (uniform scaling), rotation, reflection, and translation. We define the space where the target and the transformed source live as the commensurate space.

For the implicit translation problem, we embed $\mathbf{D}_0$ and $\mathbf{D}_1$ through multidimensional scaling to obtain $n \times d$ configurations $\mathbf{X}_0$ and $\mathbf{X}_1$ in the space $\mathbb{R}^d$ separately. The two new documents $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ are then embedded to $\tilde{\boldsymbol{y}}_0$ and $\tilde{\boldsymbol{y}}_1$ in $\mathbb{R}^d$ respectively via out-of-sample embedding [20]. Notice that the coordinates in $\mathbf{X}_0$ and $\mathbf{X}_1$ may be given in different sys-

tems. Procrustes analysis is performed to transform one of the embeddings—e.g., $\mathbf{X}_1$—to best match the other one—e.g., $\mathbf{X}_0$. The resulting transformation function t is then applied to the corresponding out-of-sample embedding $\tilde{\boldsymbol{y}}_1$ so that $t(\tilde{\boldsymbol{y}}_1)$ and $\tilde{\boldsymbol{y}}_0$ are commensurate.

For the classification problem, a similar procedure is performed. Let $\mathbf{D}_0^{\mathcal{C}}$ and $\mathbf{D}_1^{\mathcal{C}}$ denote the dissimilarity matrices among documents in $\mathcal{T}_0$ and $\mathcal{T}_1$. We embed $\mathbf{D}_0^{\mathcal{C}}$ and $\mathbf{D}_1^{\mathcal{C}}$ to $\mathbf{X}_0^{\mathcal{C}}$ and $\mathbf{X}_1^{\mathcal{C}}$ in $\mathbb{R}^d$ respectively. Then the English documents in $\tilde{\mathcal{T}}_0$ and the new French documents, whose class labels belong to $\tilde{\mathcal{C}}$, are embedded to $\boldsymbol{X}_0^{\tilde{\mathcal{C}}}$ and $\boldsymbol{X}_1^{\tilde{\mathcal{C}}}$ in $\mathbb{R}^d$ respectively via out-of-sample embedding. Procrustes transformation function $t_{\mathcal{C}}$ learned from $\mathbf{X}_0^{\mathcal{C}}$ and $\mathbf{X}_1^{\mathcal{C}}$ is then applied to $\boldsymbol{X}_1^{\tilde{\mathcal{C}}}$ so that $t_{\mathcal{C}}(\boldsymbol{X}_1^{\tilde{\mathcal{C}}})$ and $\boldsymbol{X}_0^{\tilde{\mathcal{C}}}$ are commensurate.

We refer this approach as the P-approach.

4.3.2 Our Approach

The P-approach creates the commensurate space in two steps, namely embedding and Procrustes transformation. In this section, we introduce a novel method, which defines the commensurate space in one step.

Suppose that we have access to a $2n \times 2n$ dissimilarity matrix, which consists of the pairwise dissimilarities among documents in $\boldsymbol{E} \cup \boldsymbol{F} = \{\boldsymbol{x}_{1,0}, \ldots, \boldsymbol{x}_{n,0},\ \boldsymbol{x}_{1,1}, \ldots, \boldsymbol{x}_{n,1}\}$. Then the embedding of this dissimilarity matrix is a $2n \times d$ data matrix, with the first n rows being the embedding of $\boldsymbol{E}$ and the rest the embedding of $\boldsymbol{F}$. In addition, the embeddings of $\boldsymbol{E}$ and $\boldsymbol{F}$ are in the same space—the commensurate space. The question is how we obtain the $2n \times 2n$ omnibus dissimilarity matrix. In implicit translation, we

impute $\mathbf{W}$, the dissimilarities between $\boldsymbol{E}$ and $\boldsymbol{F}$, by the entrywise average of $\mathbf{D}_0$ and $\mathbf{D}_1$. That is, the dissimilarity between the English document $\boldsymbol{x}_{i,0}$ and the French document $\boldsymbol{x}_{j,1}$ is imputed as the average of the two dissimilarities: the dissimilarity between the English documents $\boldsymbol{x}_{i,0}$ and $\boldsymbol{x}_{j,0}$, and the dissimilarity between the French documents $\boldsymbol{x}_{i,1}$ and $\boldsymbol{x}_{j,1}$. An omnibus dissimilarity matrix $\mathbf{M}$ is then constructed by putting $\mathbf{D}_0$ and $\mathbf{D}_1$ on the diagonal, and putting $\mathbf{W}$ on the off-diagonal. We embed $\mathbf{M}$ to obtain a configuration of $2n$ points $\mathbf{X}$ in $\mathbb{R}^d$. We take the first n points and the remaining n points as embeddings of $\mathbf{D}_0$ and $\mathbf{D}_1$, respectively. Notice that $\mathbf{X}_0$ and $\mathbf{X}_1$ are already in the same space Ξ_c, because the distances between matched English and French document pairs have been taken into account when embedding $\mathbf{M}$—the imputed matrix $\mathbf{W}$ has all zeros on its diagonal. For any two additional documents $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$, let $\boldsymbol{u}_0$ and $\boldsymbol{v}_1$ denote the dissimilarity vector between $\boldsymbol{y}_0$ and $\boldsymbol{E}$, $\boldsymbol{y}_1$ and $\boldsymbol{F}$, respectively. Under the null hypothesis that $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ are matched, we impute the dissimilarities between $\boldsymbol{y}_0$ and $\boldsymbol{F}$ (denoted by $\boldsymbol{v}_0$), and dissimilarities between $\boldsymbol{y}_1$ and $\boldsymbol{E}$ (denoted by $\boldsymbol{u}_1$) by entrywise average of $\boldsymbol{u}_0$ and $\boldsymbol{v}_1$. That is, $\boldsymbol{v}_0 = \boldsymbol{u}_1 = (\boldsymbol{u}_0 + \boldsymbol{v}_1)/2$. Out-of-sample embedding is used to embed $(\boldsymbol{u}_0^t, \boldsymbol{v}_0^t)^t$ and $(\boldsymbol{u}_1^t, \boldsymbol{v}_1^t)^t$ into Ξ_c. Figure 4.3 depicts the construction of the omnibus dissimilarity matrix $\mathbf{M}$.

In the classification problem, similarly we create omnibus matrix $\mathbf{M}^{\mathcal{C}}$ from $\mathbf{D}_0^{\mathcal{C}}$, $\mathbf{D}_1^{\mathcal{C}}$ and the imputed matrix $\mathbf{W}^{\mathcal{C}} = (\mathbf{D}_0^{\mathcal{C}} + \mathbf{D}_1^{\mathcal{C}})/2$. The omnibus matrix $\mathbf{M}^{\mathcal{C}}$ is then embedded into a commensurate space Ξ_c. To embed out-of-sample English documents in $\tilde{\mathcal{T}}_0$, we first impute the dissimilarity between $\boldsymbol{x}_{i,0} \in \tilde{\mathcal{T}}_0$ and $\boldsymbol{x}_{j,1} \in \mathcal{T}_1$ by the average of the dissimilarities between $\boldsymbol{x}_{j,1}$ and $\boldsymbol{x}_{i,0}$'s 3 nearest neighbors in $\mathcal{T}_0$. (These dissimilarities can be found

$$\overset{2n\times 2n}{\mathbf{M}} = \begin{array}{c} \\ \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \end{array} \begin{array}{c} \begin{array}{cc} \overset{n\times n}{\mathbf{D}_0} & \overset{n\times n}{\mathbf{W}} \end{array} \\ \begin{array}{cc} \mathbf{W}^T & \overset{n\times n}{\mathbf{D}_1} \end{array} \\ \begin{array}{cc} \boldsymbol{u}_0^t & \boldsymbol{v}_0^t \end{array} \\ \begin{array}{cc} \boldsymbol{u}_1^t & \boldsymbol{v}_1^t \end{array} \end{array} \begin{array}{c} \\ \overset{n\times 1}{\boldsymbol{u}_0}\ \overset{n\times 1}{\boldsymbol{u}_1} \\ \overset{n\times 1}{\boldsymbol{v}_0}\ \overset{n\times 1}{\boldsymbol{v}_1} \\ \\ \end{array}$$

Figure 4.3: We impute $\mathbf{W}$, the dissimilarities between $\boldsymbol{E}$ and $\boldsymbol{F}$, by $(\mathbf{D}_0 + \mathbf{D}_1)/2$ to construct $\mathbf{M}$, which is then embedded into the space Ξ_c. We impute $\boldsymbol{u}_1$ and $\boldsymbol{v}_0$ by $(\boldsymbol{u}_0 + \boldsymbol{v}_1)/2$. Finally, out-of-sample embedding is used to embed $(\boldsymbol{u}_0^t, \boldsymbol{v}_0^t)^t$ and $(\boldsymbol{u}_1^t, \boldsymbol{v}_1^t)^t$ into Ξ_c.

in $\mathbf{W}^{\mathcal{C}}$.) All the imputed dissimilarities are stored in $\mathbf{D}_{01}^{\tilde{\mathcal{C}}\mathcal{C}}$. The dissimilarities between documents in $\tilde{\mathcal{T}}_0$ and $\mathcal{T}_0$ are given by $\mathbf{D}_0^{\tilde{\mathcal{C}}\mathcal{C}}$, and the dissimilarities among $\tilde{\mathcal{T}}_0$ are given by $\mathbf{D}_0^{\tilde{\mathcal{C}}}$. Trosset and Priebe's out-of-sample embedding approach [20] is then used to embed $\tilde{\mathcal{T}}_0$ into the space Ξ_c. Similarly, new French documents of classes $\tilde{\mathcal{C}}$ are embedded into Ξ_c. Figure 4.4 depicts the construction of the omnibus dissimilarity matrix $\mathbf{M}^{\mathcal{C}}$ and how to out-of-sample embed documents in $\tilde{\mathcal{T}}_0$.

We refer this approach as the W-approach.

4.3.3 Fusion

We consider one additional step, to combine the data of textual content and graph structure. Ideally both sources of data contain complementary information so that their fusion leads to larger power in testing and higher accuracy in classification than using either textual content data or graph structure data alone. We achieve the fusion by combining the

$$\mathbf{M}^{\mathcal{C}} = \begin{array}{|c|c|c|} \hline \mathbf{D}_0^{\mathcal{C}} & \mathbf{W}^{\mathcal{C}} & \mathbf{D}_0^{\mathcal{C}\tilde{\mathcal{C}}} \\ \hline \mathbf{W}^{\mathcal{C}} & \mathbf{D}_1^{\mathcal{C}} & \mathbf{D}_{10}^{\mathcal{C}\tilde{\mathcal{C}}} \\ \hline \mathbf{D}_0^{\tilde{\mathcal{C}}\mathcal{C}} & \mathbf{D}_{01}^{\tilde{\mathcal{C}}\mathcal{C}} & \mathbf{D}_0^{\tilde{\mathcal{C}}} \\ \hline \end{array}$$

Figure 4.4: We impute $\mathbf{W}^{\mathcal{C}}$, the dissimilarities between documents in $\mathcal{T}_0$ and $\mathcal{T}_1$, by $(\mathbf{D}_0^{\mathcal{C}} + \mathbf{D}_1^{\mathcal{C}})/2$ to construct $\mathbf{M}^{\mathcal{C}}$, which is then embedded into the space Ξ_c. The dissimilarities between documents in $\tilde{\mathcal{T}}_0$ and $\mathcal{T}_0$ are given by $\mathbf{D}_0^{\tilde{\mathcal{C}}\mathcal{C}}$ ($\mathbf{D}_0^{\mathcal{C}\tilde{\mathcal{C}}}$ is the transpose of $\mathbf{D}_0^{\tilde{\mathcal{C}}\mathcal{C}}$). The dissimilarity between $\boldsymbol{x}_{i,0} \in \tilde{\mathcal{T}}_0$ and $\boldsymbol{x}_{j,1} \in \mathcal{T}_1$ are imputed by the average of the $\mathbf{W}^{\mathcal{C}}$ entries that are corresponding to $\boldsymbol{x}_{i,1}$ and $\boldsymbol{x}_{i,0}$'s 3 nearest neighbors in $\mathcal{T}_0$. All the imputed dissimilarities are stored in $\mathbf{D}_{01}^{\tilde{\mathcal{C}}\mathcal{C}}$ ($\mathbf{D}_{10}^{\mathcal{C}\tilde{\mathcal{C}}}$ is the transpose of $\mathbf{D}_{01}^{\tilde{\mathcal{C}}\mathcal{C}}$).

embeddings obtained in the P- or W-approach via the Cartesian product [5].

4.4 Results

To compute critical values and estimate powers in hypothesis testing, we randomly select two pairs of matched documents from $\boldsymbol{E}$ and $\boldsymbol{F}$. That is, we leave out four documents, two from each language, and they result in two matched pairs and two non-matched pairs. (Notice that in a real problem we only need to leave one matched pair out to get critical values; leaving two matched pairs out makes it also possible to estimate testing powers.) The approaches introduced in Section 4.3 are then applied to obtain the distances between the two matched pairs (denoted by d_0), and the distances between the two non-matched pairs (denoted by d_A). We use Classical Multidimensional Scaling (CMDS) [17, 50] in the embedding. Embedding dimension $d = 6$ is determined by Zhu and Ghodsi's automatic dimensionality selection [45]. We use ranks of the distances d_A based on 200 Monte Carlo simulations to estimate the powers for different levels of α, where the power β_α is the probability of rejecting the null hypothesis when rejection is in fact the correct decision and α is the probability of missing a true match. That is, for each $\alpha \in [0, 1]$, the critical value c_α is defined as the (100α)th percentile of d_0, and the corresponding power is the percentage of distances in d_A that are larger than the critical value c_α. The power at level α is our performance in determining that a non-match is in fact a non-match. The β against α ROC curves are shown in Figure 4.5. For example, at $\alpha = 0.05$ (missing 5% of the true matches),

we obtain a power of $\hat{\beta}_{W\text{-}fusion} = 0.560$ (correctly eliminating 56% of the false matches) via W-fusion. This is a statistical significant improvement over the results obtained sans fusion ($\hat{\beta}_{P\text{-}G} = 0.135$, $\hat{\beta}_{P\text{-}T} = 0.379$, $\hat{\beta}_{W\text{-}G} = 0.403$, $\hat{\beta}_{W\text{-}T} = 0.468$. See Figure 4.5).

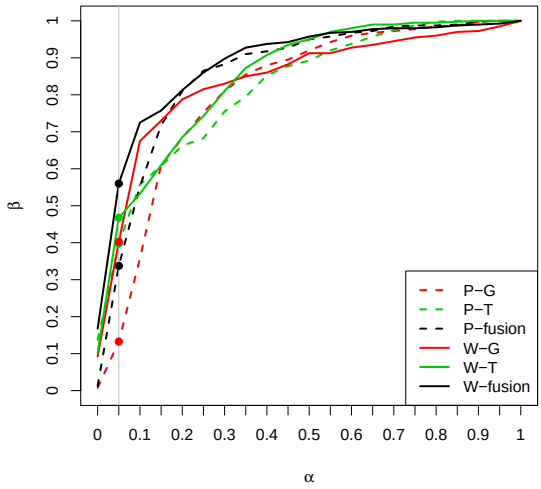


Figure 4.5: The ROC curve depicts that W-approach is generally superior to P-approach; T is generally superior to G; Fusion is generally superior to either G or T alone.

As mentioned in Section 4.3, the commensurate space Ξ_c in the classification problem is determined by $\mathbf{D}_0^{\mathcal{C}}$ and $\mathbf{D}_1^{\mathcal{C}}$. Training English documents in $\tilde{\mathcal{T}}_0$ and new French documents are then embedded into Ξ_c. We consider two types of association relations between $\mathcal{T}_0$ and $\mathcal{T}_1$, 1-to-1 association and group association. When assuming 1-to-1 association, we use the information of 1-to-1 correspondence between the training English and French documents with classes in $\mathcal{C}$; while for group association, we use only the class label information between English and French documents, but not use the 1-to-1 relationship between them. Introducing group association between $\mathcal{T}_0$ and $\mathcal{T}_1$ makes it possible to define a com-

mensurate space through non-matched English and French documents. When assuming group association, in P-approach we learn transformation matrix through the group means of embeddings, while in W-approach we impute the dissimilarities among same group by 0s and those between different groups by the dissimilarities between group means.

In the commensurate space, we train a linear classifier g based on the embedding of $\tilde{\mathcal{T}}_0$. We then apply g to the embeddings of new French documents. Classification errors are given in Table 4.1. It is clear that W-approach results in smaller classification errors than P-approach. But combining data from graph structure and text content does not, in general, improve performance.

Assocation	P-G	P-T	P-fusion	W-G	W-T	W-fusion
1 – 1	0.417	0.496	0.493	0.300	0.285	0.282
Group	0.404	0.470	0.470	0.301	0.069	0.122

Table 4.1: Given the association between the training data $\mathcal{T}_0$ and $\mathcal{T}_1$, one-to-one or group-to-group, we transform Ξ_0 and Ξ_1 into one commensurate space by P- or W-approach. A linear disscriminant classifier is then created based on $\tilde{\mathcal{T}}_0$ and then tested on $\tilde{\mathcal{T}}_1$. The symbols G and T indicate that the Graph and Text data, respectively.

4.5 Conclusion and Discussion

We have discussed two problems regarding fusion from multiple data sources in a commensurate space:

1. how to determine whether a set of feature vectors represent a single object measured

under different conditions?

2. how to create a classifier based on training data measured under one condition in order to classify objects measured in other conditions?

The key to both problems is to construct a commensurate space, where the (transformed) feature vectors of different sources are comparable and would be treated as if they were collected under the same condition. Two approaches were studied. In P-approach, embedding dissimilarity matrices and defining a commensurate space are performed separately. W-approach achieves the two procedures simultaneously, by constructing an omnibus dissimilarity matrix. Applying both approaches on Wikipedia data set showed that W-approach leads to higher hypothesis testing powers in the implicit translation problem and smaller errors in the classification problem, compared to P-approach.

4.5.1 Procrustes Transformation and Embedding with Raw Stress

In the P- and W-approaches, we have used classical multidimensional scaling to embed dissimilarity matrices. In this section, we investigate the relation between the two-step P-approach—embedding dissimilarity matrices and performing Procrustes translation to best match the embeddings—and the one-step W-approach—performing embedding and matching simultaneously—in conjunction with raw Stress as the criterion function for embedding. We show that the two-step approach and the one-step approach result in same

embeddings in the limit, and the one-step approach has an advantage—-by managing a weight parameter of the criterion function, one has more flexibility in determining the optimization problem: as the weight goes to 0, one gets embeddings the same as what he/she would get via the P-approach; as the weight increases, one put more penalty on the non-matchness of the two embeddings. We hope to shed some light on why the W-approach in conjunction with classical multidimensional scaling leads to better performances than the P-approach in the inferential tasks we introduced in this chapter.

Consider two $n \times n$ dissimilarity matrices $\Delta_1 = [\delta_{ij}^{(1)}]$ and $\Delta_2 = [\delta_{ij}^{(2)}]$. Let $\mathbf{X}_1$ and $\mathbf{X}_2$ denote the two corresponding $n \times p$ configuration matrices obtained by separately minimizing raw Stress (4.1) and (4.2) with the constraint that $\mathbf{X}_1$ and $\mathbf{X}_2$ have column means equal to 0,

$$\sigma_r(\mathbf{X}_1) = \sum_{i<j} \left[d_{ij}(\mathbf{X}_1) - \delta_{ij}^{(1)} \right]^2, \tag{4.1}$$

$$\sigma_r(\mathbf{X}_2) = \sum_{i<j} \left[d_{ij}(\mathbf{X}_2) - \delta_{ij}^{(2)} \right]^2. \tag{4.2}$$

Let

$$\mathbf{Q} = \underset{\mathbf{P}^t\mathbf{P}=\mathbf{P}\mathbf{P}^t=\mathbf{I}}{\arg\min} \|\mathbf{X}_1 - \mathbf{X}_2\mathbf{P}\|^2, \tag{4.3}$$

$$\tilde{\mathbf{X}}_2 = \mathbf{X}_2\mathbf{Q}, \tag{4.4}$$

$$\mathbf{X} = \left[\mathbf{X}_1^t \ \tilde{\mathbf{X}}_2^t \right]^t \in \mathbb{R}^{2n \times p}. \tag{4.5}$$

For $\epsilon > 0$, let $\mathbf{Y}_\epsilon = [\mathbf{Y}_1^t \ \mathbf{Y}_2^t]^t \in \mathbb{R}^{2n \times p}$ be a configuration obtained by minimizing

$$L(\mathbf{Y}, \epsilon) = \sum_{i<j} \left[d_{ij}(\mathbf{Y}_1) - \delta_{ij}^{(1)} \right]^2 + \sum_{i<j} \left[d_{ij}(\mathbf{Y}_2) - \delta_{ij}^{(2)} \right]^2 + \epsilon \|\mathbf{Y}_1 - \mathbf{Y}_2\|^2 \tag{4.6}$$

$$= \sigma_r(\mathbf{Y}_1) + \sigma_r(\mathbf{Y}_2) + \epsilon \cdot s(\mathbf{Y}) \tag{4.7}$$

with the constraint that $\mathbf{Y}_1$ and $\mathbf{Y}_2$ have column means equal to 0. Let $\mathbf{Y}_0 = \lim_{\epsilon \to 0} \mathbf{Y}_\epsilon$.

Theorem 4.5.1. $\mathbf{Y}_0$ is equal to $\mathbf{X}$, up to rotation and reflection.[1]

Proof. Let A denote the set of $\mathbf{Y} = [\mathbf{Y}_1^t \ \mathbf{Y}_2^t]^t \in \mathbb{R}^{2n \times p}$ that minimizes

$$\sigma_r(\mathbf{Y}) \stackrel{\text{def.}}{=} \sigma_r(\mathbf{Y}_1) + \sigma_r(\mathbf{Y}_2)$$

with $\mathbf{Y}_1$ and $\mathbf{Y}_2$ having column means equal to 0.

Let $\tilde{\mathbf{Y}} = \underset{\mathbf{Y} \in A}{\arg\min}\ s(\mathbf{Y}) = \underset{\mathbf{Y} \in A}{\arg\min}\ \|\mathbf{Y}_1 - \mathbf{Y}_2\|^2$. Notice that because $\mathbf{X}$ is equal to $\tilde{\mathbf{Y}}$ up to rotation and reflection, it is sufficient to show that $\mathbf{Y}_0$ is equal to $\tilde{\mathbf{Y}}$, up to rotation and reflection.

To show that $\mathbf{Y}_0$ is the same as $\tilde{\mathbf{Y}}$, up to rotation and reflection, it is sufficient to show that

$$\lim_{\epsilon \to 0} L(\tilde{\mathbf{Y}}, \epsilon) - L(\mathbf{Y}_\epsilon, \epsilon) = 0, \tag{4.8}$$

and

$$\lim_{\epsilon \to 0} \sigma_r(\mathbf{Y}_\epsilon) - \sigma_r(\tilde{\mathbf{Y}}) = 0. \tag{4.9}$$

[1] In case that more than one global minimum configuration exist [19, 13.4], after taking into account rotation and reflection, the two sets of minimum configurations should be the same.

Consider (4.8), $L(\tilde{\mathbf{Y}}, \epsilon) - L(\mathbf{Y}, \epsilon) \geq 0$, because $\mathbf{Y}_\epsilon$ minimizes $L(\mathbf{Y}, \epsilon)$. On the other hand,

$$L(\tilde{\mathbf{Y}}, \epsilon) - L(\mathbf{Y}, \epsilon) \tag{4.10}$$

$$= \sigma_r(\tilde{\mathbf{Y}}) + \epsilon \cdot s(\tilde{\mathbf{Y}}) - [\sigma_r(\mathbf{Y}_\epsilon) + \epsilon \cdot s(\mathbf{Y}_\epsilon)] \tag{4.11}$$

$$= \epsilon[s(\tilde{\mathbf{Y}}) - s(\mathbf{Y}_\epsilon)] + [\sigma_r(\tilde{\mathbf{Y}}) - \sigma_r(\mathbf{Y}_\epsilon)] \tag{4.12}$$

$$\leq \epsilon[s(\tilde{\mathbf{Y}}) - s(\mathbf{Y}_\epsilon)] \tag{4.13}$$

$$\leq \epsilon \cdot s(\tilde{\mathbf{Y}}). \tag{4.14}$$

Notice that in (4.12), $\sigma_r(\tilde{\mathbf{Y}}) - \sigma_r(\mathbf{Y}_\epsilon) \leq 0$. This is because $\tilde{\mathbf{Y}}$ minimizes $\sigma_r(\mathbf{Y})$. Hence,

$$0 \leq L(\tilde{\mathbf{Y}}, \epsilon) - L(\mathbf{Y}, \epsilon) \leq \epsilon \cdot s(\tilde{\mathbf{Y}}).$$

Taking the limit as $\epsilon \to 0$, we have

$$0 \leq \lim_{\epsilon \to 0} L(\tilde{\mathbf{Y}}, \epsilon) - L(\mathbf{Y}_\epsilon, \epsilon) \leq \lim \epsilon \cdot s(\tilde{\mathbf{Y}}) = 0.$$

Therefore,

$$\lim_{\epsilon \to 0} L(\tilde{\mathbf{Y}}, \epsilon) - L(\mathbf{Y}_\epsilon, \epsilon) = 0.$$

Consider (4.9), because

$$0 = \lim_{\epsilon \to 0} L(\tilde{\mathbf{Y}}, \epsilon) - L(\mathbf{Y}_\epsilon, \epsilon) = \lim_{\epsilon \to 0} \epsilon[s(\tilde{\mathbf{Y}}) - s(\mathbf{Y}_\epsilon)] + \lim_{\epsilon \to 0}[\sigma_r(\tilde{\mathbf{Y}}) - \sigma_r(\mathbf{Y}_\epsilon)],$$

and

$$\lim_{\epsilon \to 0} \epsilon[s(\tilde{\mathbf{Y}}) - s(\mathbf{Y}_\epsilon)] = 0,$$

we have

$$\lim_{\epsilon \to 0} [\sigma_r(\tilde{\mathbf{Y}}) - \sigma_r(\mathbf{Y}_\epsilon)] = 0.$$

□

Chapter 5

Combining Multiple Dissimilarity Matrices in Shape Analysis

Among other areas, dissimilarity representation is widely used in shape analysis. In this chapter, we study the problem of combining multiple dissimilarity matrices derived from the same set of shapes for classification purpose.

5.1 Introduction

The Large Deformation Diffeomorphic Metric Mapping (LDDMM) techniques [51–53] compare and quantize morphometric changes in shapes. They assign dissimilarities on the space of anatomical images in Computational Anatomy. Thereby they provide a way to analyze shapes—by performing statistical analysis on the LDDMM dissimilarities. For a

set of shapes, there are three types of LDDMM dissimilarity measures, namely LDDMM-Volume, LDDMM-Surface and LDDMM-Landmark. From the names we can tell that these measures focus on different aspects of the shapes—they aim to perform a non-rigid deformation between two shapes by representation of their volume (all voxels), surface (all voxels on the surface) and a set of landmarks, respectively. Applying the three LDDMM dissimilarity measures on a set of shapes results in three dissimilarity matrices. In this chapter, we compare and combine the three dissimilarity matrices in the context of classification. Toward this end, we simulate 3 groups of 3D phantom shapes with 100 shapes in each group. A main advantage of using computer-generated phantom shapes rather than real Magnetic Resonance Imaging (MRI) images in studying LDDMM dissimilarities is that the exact anatomy and class membership of the phantom are known, thus providing a gold standard from which to evaluate the various LDDMM procedures. Other advantages include (1) phantom shapes can be quickly obtained with low cost, while to prepare a set of real (segmented) MRI images typically take months, if not years and is very expensive; (2) phantom shapes are constructed to be independent and identically distributed (i.i.d.). However, real data are messy. They could be collected or pre-processed under different conditions. This fact does not satisfy most statistical procedures that require i.i.d. data; (3) real shapes are usually not of the same scale/size. They need to be re-scaled and aligned before applying LDDMM procedures. This step could introduce additional variance to the LDDMM dissimilarities.

5.2 Data

5.2.1 3 Classes of 3D Phantom Shapes

Consider the unit cube $\Omega = [0,1]^3$ as the 3D background space. We first generate 3 base shapes with the resolution $32 \times 32 \times 32$: an ellipsoid, a broken ellipsoid and an elliptic cylinder. Random noises are then added to these base shapes. We finally smooth out the isolate points to obtain the phantom shapes that we shall apply the various LDDMM procedures on. Notice that all the 3-class objects are generated based on the first 3 base shapes.

The 3 base shapes are generated within Ω and centered at $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. The ellipsoid is parametrized by a, c, the equatorial radii along the x and z axes, and b, the polar radius along y axis. The broken ellipsoid is obtained by excluding from the ellipsoid the points that are inside a ball centering at $(0, \frac{1}{2}, \frac{1}{2})$ with radius r. And the elliptic cylinder is constructed by belting the ellipsoid. That is, the elliptic cylinder consists of all the points that are within both the ellipsoid and an elliptic column, whose major and minor radii are proportional to a and c. Figure 5.1 shows a sketch of the three base shapes. Notice that the three base shapes are of the same position, scale and orientation. The random phantom shapes generated based on them should retain same properties. Equations (5.1–5.3) are the mathematical formulas we use to generate the 3 base shapes. The specific parameters we use are

$$a = \frac{1}{6},\ b = \frac{2}{5},\ c = \frac{1}{10};\ \ r = \frac{11}{30};\ \ h = \frac{3}{10}.$$

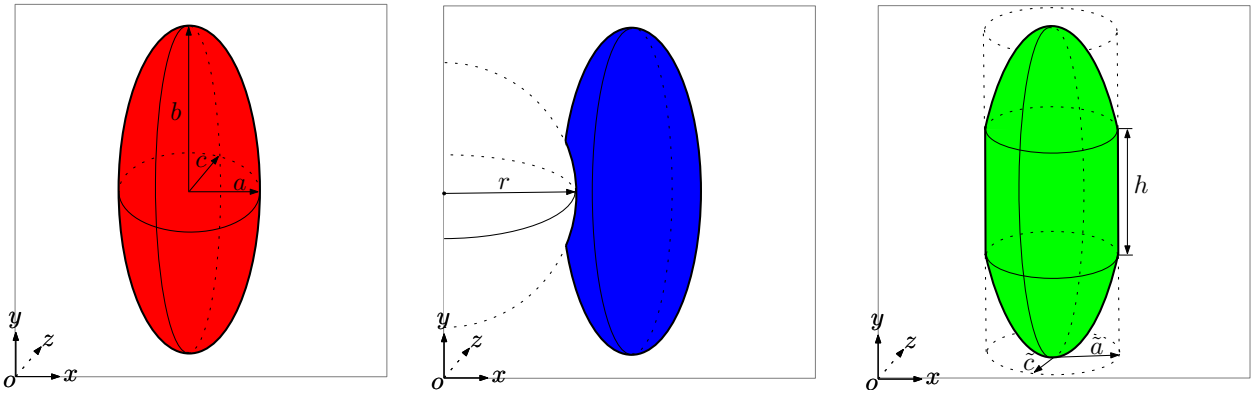


Figure 5.1: The three base shapes are all within the unit cube $\Omega = [0,1]^3$ and centered at $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. The ellipsoid (left) is parametrized by a, c, the equatorial radii along the x and z axes, and b, the polar radius along the y axis; the broken ellipsoid (middle) is obtained by excluding from the ellipsoid the points that are inside a ball centered at $(0, \frac{1}{2}, \frac{1}{2})$ with radius r; and the elliptic cylinder (right) consists of all the points that are within both the ellipsoid and an elliptic column, whose major and minor radii are proportional to a and c ($\tilde{a}/a = \tilde{c}/c = \sqrt{1-(h/2b)^2}$).

$$\text{Ellipsoid:} \qquad \frac{(x-\frac{1}{2})^2}{a^2} + \frac{(y-\frac{1}{2})^2}{b^2} + \frac{(z-\frac{1}{2})^2}{c^2} \leq 1. \tag{5.1}$$

$$\text{Broken Ellipsoid:} \begin{cases} \dfrac{(x-\frac{1}{2})^2}{a^2} + \dfrac{(y-\frac{1}{2})^2}{b^2} + \dfrac{(z-\frac{1}{2})^2}{c^2} \leq 1, \\ \\ (x+\dfrac{1}{2})^2 + (y-\dfrac{1}{2})^2 + (z-\dfrac{1}{2})^2 \geq r^2. \end{cases} \tag{5.2}$$

$$\text{Elliptic Cylinder: } \begin{cases} \dfrac{(x-\frac{1}{2})^2}{a^2} + \dfrac{(y-\frac{1}{2})^2}{b^2} + \dfrac{(z-\frac{1}{2})^2}{c^2} \leq 1, \\ \\ \dfrac{(x-\frac{1}{2})^2}{a^2\left(1-(\frac{h}{2b})^2\right)} + \dfrac{(z-\frac{1}{2})^2}{c^2\left(1-(\frac{h}{2b})^2\right)} \leq 1. \end{cases} \tag{5.3}$$

We then add random noises to the 3 base shapes to obtain n 3-class shapes $(n = 300)$, with 100 objects in each class. Conventionally this is achieved by randomly moving each point on the shapes generated basing on the formulas (5.1–5.3) toward any direction by ϵ voxels ($\epsilon \in \{0, 1\}$). Alternatively, we directly add random noises to the coordinates of grid points to obtain $(x+\epsilon_x, y+\epsilon_y, z+\epsilon_z)$, which are then used to generate shapes according to the mathematical formulas (5.1–5.3). The three random variables $\epsilon_x, \epsilon_y, \epsilon_z$ are independent and uniformly distributed on $\{-\frac{1}{32}, \frac{0}{32}, \frac{1}{32}\}$.

To avoid isolate points and harsh surfaces, a finial smoothing step is performed. We determine whether to keep a point v (e.g. the center point in Figure 5.2) by the number of points that are within the 26 nearest neighbors of v and are also on the shape. We keep the point v only if that number is greater than a threshold, which we choose to be 20. Figure 5.3 shows some of the final phantom shape examples.

5.2.2 LDDMM Dissimilarity Matrices

The essential idea of the LDDMM approaches aims to model the space of shapes as a Riemmannian manifold with metric structure, which results from the assumption that the

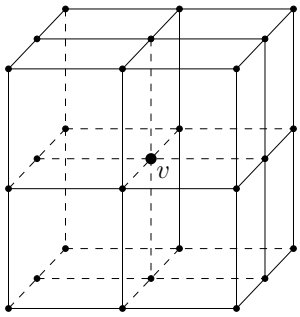
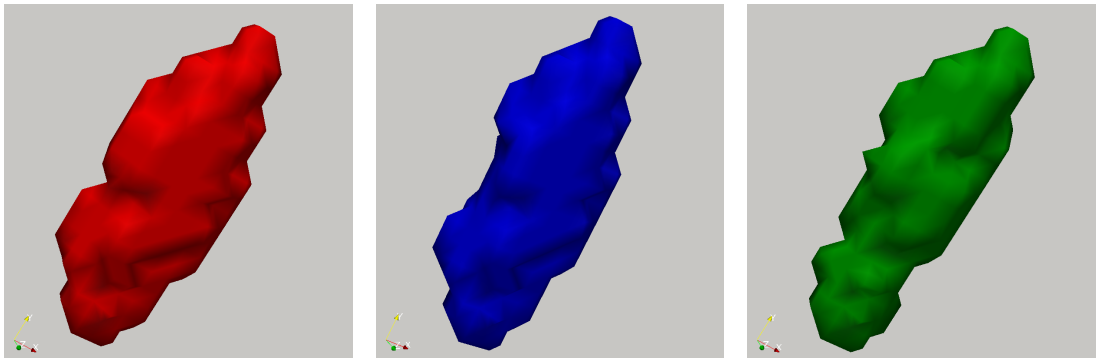


Figure 5.2: We determine whether to keep the center point v on the shape by examining the surrounding 26 black points. The point v is retained only if 20 or more out of its 26 neighbors are on the shape.



(a) Ellipsoid (b) Broken ellipsoid (c) Elliptic cylinder

Figure 5.3: An example of smoothed shapes from the three classes.

shapes are an orbit under a group of diffeomorphisms (1-1 and onto transformations with inverses that are smooth) [23, and references therein]. For any pair of shapes I and J, there exists a flow of diffeomorphisms $g_t,\ t \in [0,1]$ transforming one shape to the other $g \cdot I = J$. The diffeomorphisms are solutions of ordinary differential equations $\dot{g}_t = v_t(g_t),\ t \in [0,1]$ with $g_0 = \text{id}$ the identity map, and associated vector fields $v_t, t \in [0,1]$. The dissimilarity between I and J is given by the length of the shortest or geodesic curve through the space of shapes generated from I connecting to J. The dissimilarity takes the form

$$\rho(I,J)^2 = \inf_{v:\ \dot{g}_t = v_t(g_t),\ g_0 = \text{id}} \int_0^1 \|v_t\|_V^2 dt, \tag{5.4}$$

such that $g \cdot I = J$. In practice, a cost function, $c(g \cdot I,\ J) \overset{\text{def.}}{=} \|g \cdot I - J\|_{L^2}^2$, measuring the difference between the mapped anatomical shape I and the target shape J is introduced in calculating the diffeomorphic mapping g between I and J. The variational problem becomes

$$\underset{v:\ \dot{g}_t = v_t(g_t)}{\arg\min} \left(\int_0^1 \|v_t\|_V^2 dt + \|g \cdot I - J\|_{L^2}^2 \right), \tag{5.5}$$

and the dissimilarity is still given by (5.4). More detailed references for different versions of LDDMM procedures can be found from [53] for LDDMM-Volume, [54] for LDDMM-Surface, and [55, 56] for LDDMM-Landmark.

The landmark points on a 3D shape are defined as following: we select the head and the tail (the two points corresponding to y_{max} and y_{min}), then five slices between them, with four landmarks at four quadrants of each slice. The five slices are cut at $y = (10,\ 30,\ 50,\ 70,\ 90)$th percentiles of $(y_{\text{min}},\ y_{\text{max}})$.

5.3 Statistical Analysis of LDDMM Dissimilarity Matrices

5.3.1 Obtain Dissimilarity Matrices

Applying three LDDMM procedures (LDDMM-Volume, LDDMM-Surface and LDDMM-Landmark) on the n $(n = 300)$ phantom shapes, respectively, generates three $n \times n$ dissimilarity matrices Δ_V, Δ_S and Δ_L. Figures 5.4 shows the density estimates of the dissimilarities within and between different classes (only upper-triangle of the dissimilarity matrices are used for within-class density estimates).

5.3.2 Classification

We consider the task of classification based on dissimilarities.

For each dissimilarity matrix Δ, we consider constructing classifiers in three different ways: (1) applying 3-nearest neighbors rule on dissimilarities directly; (2) embedding Δ into $\mathbf{X} \in \mathbb{R}^{n\times d}$ and creating 3-nearest neighbors rule using $\mathbf{X}$; and (3) embedding Δ into $\mathbf{X} \in \mathbb{R}^{n\times d}$ and using the Linear Discriminant Analysis (LDA). Leave-one-out cross-validation is used to compare the performance of these classifiers. We embed Δ_V, Δ_S and Δ_L to obtain $\mathbf{X}_V \in \mathbb{R}^{n\times 100}$, $\mathbf{X}_S \in \mathbb{R}^{n\times 100}$ and $\mathbf{X}_L \in \mathbb{R}^{n\times 50}$ via classical multidimensional scaling. To alleviate the *curse of dimensionality*, we use the automatic dimensionality selection introduced in [45] to reduce the dimensionality of $\mathbf{X}$ before creating classifiers. The

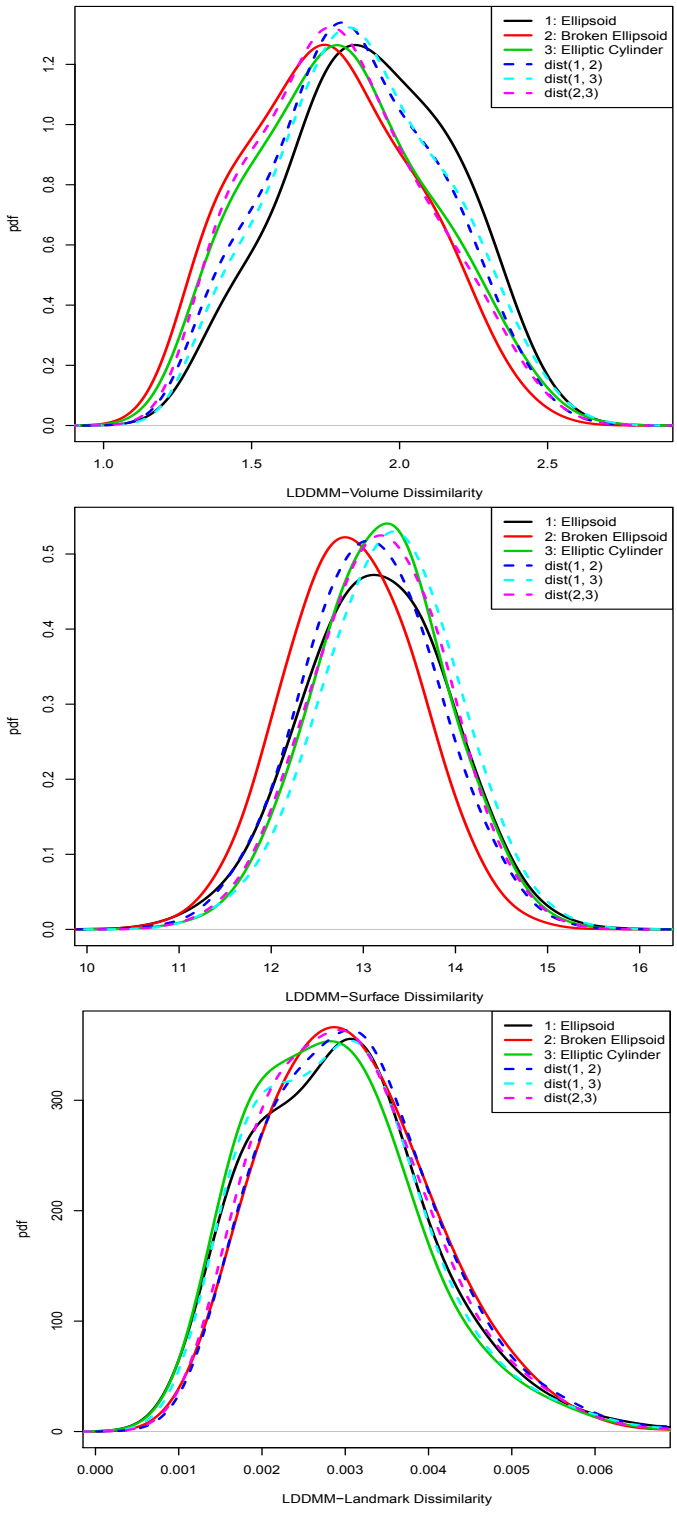


Figure 5.4: Density estimates of dissimilarities within and between different classes.

classification errors are shown in Table 5.1. We can see that (i) LDA on embeddings results in smaller errors than the other two classifiers; (ii) LDDMM-Surface dissimilarity data has more class information than LDDMM-Volume and LDDMM-Landmark dissimilarity data. We note that the two findings are based on our particular experiment settings and they may not be generally true.

	3-*NN*	3-*NN* (d)	LDA (d)
Volume	0.517	0.680 (42)	0.467 (42)
Surface	0.513	0.510 (32)	0.300 (32)
Landmark	0.603	0.627 (21)	0.490 (21)

Table 5.1: Classification errors for 3-class problem. The first column corresponds the 3-NN applied directly on dissimilarities, the second and third columns correspond to the 3-NN and LDA applied on the embeddings of dissimilarity matrices. The dimensionalities d used in building classifiers (3-NN or LDA) in the embedding space are provided in parenthesis. We can see that (i) LDA on embeddings results in smaller errors than the other two classifiers; (ii) LDDMM-Surface dissimilarity data has more class information than LDDMM-Volume and LDDMM-Landmark dissimilarity data.

5.3.2.1 Fusion

Using the approach introduced in Chapter 3, we combine Δ_S and Δ_L, as well as all the three matrices—Δ_V, Δ_S and Δ_L. The classification error from using surface and landmark data is 0.15, and the classification error from using all three matrices is 0.17. Both errors are smaller than the errors from each single data alone. However, adding volume data to

the combination of surface and landmark data clearly is not helpful.

Chapter 6

Conclusions and Future Work

This dissertation has introduced the dissimilarity framework for combining multiple disparate sources of data. This framework has the following advantages over traditional feature-level fusion techniques:

- For some types of data, such as hyperspectral images, text data, contours or shapes, data represented by trees or graphs, it is hard to extract meaningful features, while it is natural or probable to compute pairwise dissimilarities. In these scenarios, feature-level fusion techniques are not applicable, while combining in the dissimilarity space has no such problem.

- Features extracted from disparate data sources are of different types and characteristics. The resulting Cartesian product space is complicated and hence difficult to model. Combining such features for statistical inferential purposes is usually not feasible.

- Feature-level fusion techniques usually contain two steps: (1) extract meaningful features from every data source, and (2) combine all the features into a Cartesian product space. A favorable feature-level fusion technique should consider and optimize the two steps together, because combining most informative features in each space may not lead to an optimal Cartesian product. However, it is usually hard to consider feature extraction procedure and feature fusion procedure simultaneously.

- Combining in the dissimilarity space also consists of two steps: (1) calculate one or more dissimilarity matrices for every data source; and (2) combine all the dissimilarity matrices. In the first step, one takes advantage of the knowledge of experts in each area, and unifies disparate types of data into the dissimilarity space. When combining all the dissimilarity matrices, usually one does not need to know the original data spaces. In other words, the two steps can be optimized separately.

- Feature-level fusion techniques are usually developed specifically for the given problems and particular types of data, and hence are not generally applicable. While methods for combining dissimilarity matrices are usually generally applicable for all problems.

Decision-level fusion techniques combine results separately obtained from each single data source. They are simple but suboptimal in principle, because joint distribution usually contains more information than the product of the marginals. Fusion in the dissimilarity framework considers joint distribution to some extent, and therefore is favorable over

decision-level fusion in many cases.

Despite all the virtues of fusion in the dissimilarity framework, there are many challenges, too. We have investigated some of them and provided sound solutions.

In Chapter 1, we gave a theoretical foundation for using dissimilarity representations in statistical pattern recognition. We showed that the probability of error for the best dissimilarity-based classifier is greater or equal to the Bayes error. That is, $L_\delta^* \geq L^*$. Moreover, L_δ^* depends on both the joint distribution of observations and class labels, and the dissimilarity measure. That is, $L_\delta^* = L^* + \epsilon(F_{XY}, \delta)$. We also showed that for discrete X and a collection of δ's, the best dissimilarity-based classifier result in same classification error as the Bayes rule. With the explorations and findings, we wish to shed some light on why the dissimilarity representation is useful in statistical pattern recognition.

One of the most widely used methods for the dissimilarity representation is to embed the dissimilarity matrix into a configuration of points (called the embedding) in the Euclidean space via multidimensional scaling. Statistical inferences or classifiers are then created based on the embedding. Once new data (or the out-of-sample data) are observed, one can calculate the overall dissimilarity matrix and then embed it. The alternative, and arguably better, way is to utilize the out-of-sample embedding, which inserts out-of-sample observations into the space represented by the original embedding in an optimal manner. In Chapter 2, we introduced the OOSIM (out-of-sample embedding by iterative majorization) procedure. OOSIM was developed as an natural extension to the multidimensional scaling technique with the raw stress (SMACOF). We compared OOSIM with Trosset and Priebe's

out-of-sample method, an extension to classical multidimensional scaling (CMDS), and found that T&P's method is consistent with CMDS, OOSIM is consistent with SMACOF; both T&P and OOSIM could be used to embed out-of-sample objects even when the within-sample embedding was not obtained by CMDS or SMACOF. We presented an example for which OOSIM is more appropriate than T&P.

In Chapter 3, we introduced a method of combining dissimilarity matrices in Cartesian product space. The main obstacles in this method are the curse of dimensionality—due to the high dimensionality of the Cartesian product space—and the parallel cigars phenomenon—the most discriminative dimensions are not necessarily the dimensions with largest variances. We developed a new supervised dimensionality reduction approach for projecting the Cartesian product into a low dimensional space. We applied this approach on simulation, as well as image and caption data. The results showed that our approach improved classification accuracy compared to the alternatives of principal components analysis and no dimensionality reduction at all.

In Chapter 4, we considered information fusion from a different perspective and discussed two problems regarding fusion from multiple data sources in a commensurate space: (1) how to determine whether a set of feature vectors represent a single object measured under different conditions? (2) how to create a classifier based on training data measured under one condition in order to classify objects measured in other conditions? The key to both problems is to construct a commensurate space, where the (transformed) feature vectors of different sources are comparable and would be treated as if they were collected

under the same condition. We studied two approaches. The P-approach embeds dissimilarity matrices and defines commensurate space separately, while the W-approach achieves the two procedures simultaneously, by constructing an omnibus dissimilarity matrix. We applied both approaches on Wikipedia data set and the results showed that the W-approach led to higher hypothesis testing powers and smaller classification errors, compared to the P-approach.

The two problems that we considered when studying fusion in a commensurate space necessarily require data from multiple sources. Hence, fusion in a commensurate space is sensible while combining in the Cartesian product space is not. In fact, there are cases where both approaches are applicable. In such cases, which method is more appropriate is based on the nature of the data. If the multiple data sources are disparate and capture different aspects of pattern, combining in the Cartesian product space should be expected to work better. On the other hand, if the multiple data sources intrinsically contain the same information, fusion in the commensurate space denoises, hence is usually preferred.

Dissimilarity representation is widely used in shape analysis. In Chapter 5, we studied the problem of combining multiple dissimilarity matrices derived from the same set of shapes for classification purpose. We introduced a way to generate a collection of 3D shapes of different groups. Three versions of the Large Deformation Diffeomorphic Metric Mapping (LDDMM) were applied onto the generated shapes, yielding 3 dissimilarity matrices. Classification results showed that, for the given data, (1) LDDMM-Surface captures most class information, compared to the other two dissimilarity measures, namely

LDDMM-Volume and LDDMM-Landmark; (2) a linear classifier trained the embedding of a dissimilarity matrix performs better than a 3-nearest neighbors classifier trained on dissimilarities or the embedding; and (3) combining LDDMM-Surface and LDDMM-Landmark dissimilarity matrices results in more accurate classifiers, but introducing LDDMM-Volume into the fusion is not beneficial.

Many challenges in combining multiple data in the dissimilarity framework still lie open for future work.

In Chapter 1, we study the relation between L^* and L^*_δ, and especially focused on discretely distributed X. We wish to investigate this relation for continuously distributed X in the future.

In Chapter 2, motivated by the results of the two out-of-sample embedding methods, OOSIM and T&P, we wish to the develop a general robustness theory indicating which out-of-sample embedding method is more appropriate in cases where neither CMDS nor SMACOF is the within-sample embedding methodology.

The J-function approach we introduced in Chapter 3 was developed for the simple two-class problem. We briefly recommended the possible ways to extend it to $C > 2$ classes. But there are still some nontrivial challenges left for future investigation.

Also in Chapter 3, we mentioned that there are three possible ways to combine multiple dissimilarity matrices and we studied one of them. We wish to study the other two methods and investigate when and why one method works better than the other two.

Multiple kernel learning has recently been widely studied [57–59]. Kernels are often

referred to as symmetric, positive definite functions of two variables. They express similarity between objects represented in a feature space, and thereby kernel methods are very related to the methods we have studied and developed for dissimilarity representation in this dissertation. We wish to investigate on combining multiple dissimilarity matrices in context of multiple kernel learning.

Bibliography

[1] C. Liu and H. Wechsler, "A shape- and texture-based enhanced fisher classifier for face recognition," *Image Processing, IEEE Transactions on*, vol. 10, no. 4, pp. 598–608, April 2001.

[2] A. Ross and A. K. Jain, "Multimodal biometrics: An overview," in *Proceedings of 12th Signal Processing Conference (EUSIPCO)*, 2004, pp. 1221–1224.

[3] Q.-S. Sun, S.-G. Zeng, Y. Liu, P.-A. Heng, and D.-S. Xia, "A new method of feature fusion and its application in image recognition," *Pattern Recognition*, vol. 38, no. 12, pp. 2437 – 2448, 2005.

[4] J. Kludas, E. Bruno, and S. M. Maillet, *Information Fusion in Multimedia Information Retrieval*. Berlin, Heidelberg: Springer-Verlag, 2008.

[5] Z. Ma, A. Cardinal-Stakenas, Y. Park, M. W. Trosset, and C. E. Priebe, "Combining dissimilarity representations in embedding product space," *Journal of Classification*, pp. 1–15, 7 October 2010.

BIBLIOGRAPHY

[6] Z. Ma, D. J. Marchette, and C. E. Priebe, "Fusion and inference from multiple data sources in a commensurate space," *submitted for publication*, 2010.

[7] K. C. Chang, T. Zhi, and R. K. Saha, "Performance evaluation of track fusion with information matrix filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, pp. 455–466, 2002.

[8] R. E. Bellman, *Adaptive control processes - A guided tour*. Princeton, New Jersey, U.S.A.: Princeton University Press, 1961.

[9] C. M. Bishop, *Neural networks for pattern recognition*. Oxford: Clarendon Press; New York: Oxford University Press, 1995.

[10] K. R. Clarke, "Non-parametric multivariate analyses of changes in community structure," *Australian Journal of Ecology*, vol. 18, no. 1, pp. pp 117–143, 1993.

[11] C. E. Priebe, "Olfactory classification via interpoint distance analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 404–413, April 2001.

[12] M. J. Anderson and J. Robinson, "Generalized discriminant analysis based on distances," *Australian & New Zealand Journal of Statistics*, vol. 45, no. 3, pp. 301–318, September 2003.

[13] E. Pękalska and R. P. W. Duin, *The Dissimilarity Representation for Pattern Recog-*

nition: Foundations and Applications. World Scientific Publishing Company, December 2005.

[14] J.-F. Maa, D. K. Pearl, and R. Bartoszyński, "Reducing multidimensional two-sample data to one-dimensional interpoint comparison," *The Annals of Statistics*, vol. 24, no. 3, pp. 1069–1074, 1996.

[15] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. New York: Springer, 1996.

[16] R. P. Duin, D. de Ridder, and D. M. Tax, "Featureless classification," Proc. Workshop on Statistical Pattern Recognition, Prague, June 1997.

[17] W. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, December 1952.

[18] J. C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika*, vol. 53, no. 3/4, pp. 325–338, 1966.

[19] I. Borg and P. J. F. Groenen, *Modern multidimensional scaling: theory and applications*, 2nd ed. New York: Springer, 2005.

[20] M. W. Trosset and C. E. Priebe, "The out-of-sample problem for classical multidimensional scaling," *Computational Statistics & Data Analysis*, vol. 52, no. 10, pp. 4635–4642, 2008.

[21] Z. Ma and C. E. Priebe, "Out-of-sample embedding using iterative majorization," *submitted for publication*, 2010.

[22] I. Borg and D. Leutner, "Dimensional models for the perception of rectangles," *Percept Psychophys*, vol. 34, no. 3, pp. 257–267, September 1983.

[23] M. I. Miller, C. E. Priebe, A. Qiu, B. Fischl, A. Kolasny, T. Brown, Y. Park, J. T. Ratnanather, E. Busa, J. Jovicich, P. Yu, B. C. Dickerson, R. L. Buckner, and the Morphometry BIRN, "Collaborative computational anatomy: An MRI morphometry study of the human brain via diffeomorphic metric mapping," *Human Brain Mapping*, vol. 30, no. 7, pp. 2132–2141, 2009.

[24] J. C. Gower, "Adding a point to vector diagrams in multivariate analysis," *Biometrika*, vol. 55, no. 3, pp. 582–585, 1968.

[25] J. de Leeuw, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*, J. R. Barra, F. Brodeau, G. Romier, and B. van Cutsem, Eds. North Holland Publishing Company, Amsterdam, 1977, pp. 133–145.

[26] J. de Leeuw and P. Mair, "Multidimensional scaling using majorization: SMACOF in R," *Journal of Statistical Software*, vol. 31, no. 3, pp. 1–30, June 2009.

[27] J. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, no. 2, pp. 115–129, June 1964.

[28] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, pp. 95–110, 1956.

[29] G. Salton and M. J. McGill, *Introduction to modern information retrieval*. McGraw-Hill Companies, 1983.

[30] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, no. 5, pp. 401–409, 1969.

[31] G. V. Trunk, "A problem of dimensionality: A simple example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 3, pp. 306–307, July 1979.

[32] W.-C. Chang, "On using principal components before separating a mixture of two multivariate normal distributions," *Applied Statistics*, vol. 32, no. 3, pp. 267–275, 1983.

[33] W. R. Dillon, N. Mulani, and D. G. Frederick, "On the use of component scores in the presence of group structure," *The Journal of Consumer Research*, vol. 16, no. 1, pp. 106–112, 1989.

[34] A. M. Kshirsagar, S. Kocherlakota, and K. Kocherlakota, "Classification procedures using principal component analysis and stepwise discriminant function," *Communications in Statistics – Theory and Methods*, vol. 19, no. 1, pp. 91–109, 1990.

[35] I. T. Jolliffe, B. J. T. Morgan, and P. J. Young, "A simulation study of the use of principal components in linear discriminant analysis," *Journal of Statistical*

Computation and Simulation, vol. 55, no. 4, pp. 353–366, 1996. [Online]. Available: http://www.informaworld.com/10.1080/00949659608811775

[36] G. T. Toussaint, "Note on optimal selection of independent binary-valued features for pattern recognition (corresp.)," *IEEE Transactions on Information Theory*, vol. 17, no. 5, pp. 618–618, Sep 1971.

[37] A. Takemura, "A principal decomposition of hotelling's t^2 statistic," in *Multivariate Analysis VI*, P. Krishnaiah, Ed. Amsterdam: Elsevier, 1985, pp. 583–597.

[38] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, July 1997.

[39] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. New York: Springer, 1996.

[40] J. Schäfer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, 2005.

[41] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2007.

[42] A. K. Jain, *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[43] D. Lin and P. Pantel, "Concept discovery from text," in *Proceedings of the 19th international conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp. 1–7.

[44] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, October 2001.

[45] M. Zhu and A. Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 918–930, Nov. 2006.

[46] J. H. Friedman, "Another approach to polychotomous classification," Department of Statistics, Stanford University, Tech. Rep., 1996.

[47] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *The Annals of Statistics*, vol. 26, no. 2, pp. 451–471, 1998.

[48] P. Pantel and D. Lin, "Discovering word senses from text," in *In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2002, pp. 613–619.

[49] R. Sibson, "Studies in the robustness of multidimensional scaling: Procrustes statistics," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 40, no. 2, pp. 234–238, 1978.

[50] T. F. Cox and M. A. A. Cox, *Multidimensional scaling*. Boca Raton: Chapman & Hall/CRC, 2001.

[51] P. Dupuis, U. Grenander, and M. I. Miller, "Variational problems on flows of diffeomorphisms for image matching," *Quarterly of Applied Mathematics*, vol. LVI, no. 3, pp. 587–600, 1998.

[52] M. I. Miller, A. Trouvé, and L. Younes, "On the metrics and euler-lagrange equations of computational anatomy," *Annual Review of Biomedical Engineering*, vol. 4, no. 1, pp. 375–405, 2002.

[53] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms." *International Journal of Computer Vision*, vol. 61, no. 2, pp. 139–157, 2005.

[54] M. Vaillant, A. Qiu, J. Glaunés, and M. I. Miller, "Diffeomorphic metric surface mapping in subregion of the superior temporal gyrus," *NeuroImage*, vol. 34, no. 3, pp. 1149 – 1159, 2007.

[55] S. Joshi and M. Miller, "Landmark matching via large deformation diffeomorphisms," *Image Processing, IEEE Transactions on*, vol. 9, no. 8, pp. 1357–1370, Aug 2000.

[56] J. Glaunès and A. T. L. Younes, "Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching," vol. 2. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 712–718.

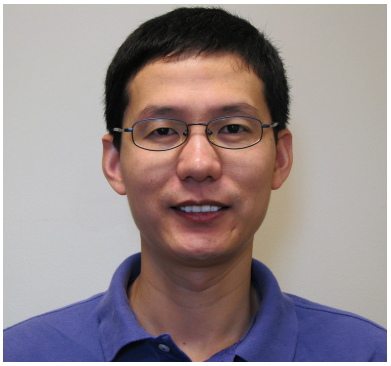
[57] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

BIBLIOGRAPHY

[58] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 41–48.

[59] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 1191–1198.

Vita



Zhiliang Ma was born on February 19, 1979 in Qing-Xian, Hebei, China. In 2002, he received his undergraduate degree—a Bachelor of Science in Applied Mathematics—from Minzu University of China, Beijing, China. He then worked as an Engineer for one year in the Advanced Mobile Communication Lab, Matsushita R&D Co. Ltd. Beijing, China. In 2003, he began his graduate studies in the Department of Mathematical Sciences at University of Cincinnati, OH, USA. In 2005, he earned a Master of Science in Applied Statistics from this department. In the same year, he went to the Department of Applied Mathematics & Statistics at Johns Hopkins University to pursue a Ph.D. in Statistics.

In 2008, he presented a poster at the Southern Regional Council on Statistics Summer Research Conference and got the R.L. Anderson award and the Clint Miller award honorable mention for best poster. He has presented his research at Interface 2008, the Joint Statistical Meeting 2009, as well as in an invited talk at Quantitative Methods in Defense

and National Security 2010.

Zhiliang currently lives in Sunnyvale, CA, and works as an Analytical Engineer at Quin-Street in Foster City, CA.