# Graph-based Estimation of Information Divergence Functions

by

Alan Wisler

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2016 by the
Graduate Supervisory Committee:

Visar Berisha, Co-Chair
Andreas Spanias, Co-Chair
Julie Liss
Daniel Bliss

ARIZONA STATE UNIVERSITY

May 2016

# ABSTRACT

Information divergence functions, such as the Kullback-Leibler divergence or the Hellinger distance, play a critical role in statistical signal processing and information theory; however estimating them can be challenge. Most often, parametric assumptions are made about the two distributions to estimate the divergence of interest. In cases where no parametric model fits the data, non-parametric density estimation is used. In statistical signal processing applications, Gaussianity is usually assumed since closed-form expressions for common divergence measures have been derived for this family of distributions. Parametric assumptions are preferred when it is known that the data follows the model, however this is rarely the case in real-word scenarios. Non-parametric density estimators are characterized by a very large number of parameters that have to be tuned with costly cross-validation. In this dissertation we focus on a specific family of non-parametric estimators, called direct estimators, that bypass density estimation completely and directly estimate the quantity of interest from the data. We introduce a new divergence measure, the $D_p$-divergence, that can be estimated directly from samples without parametric assumptions on the distribution. We show that the $D_p$-divergence bounds the binary, cross-domain, and multi-class Bayes error rates and, in certain cases, provides provably tighter bounds than the Hellinger divergence. In addition, we also propose a new methodology that allows the experimenter to construct direct estimators for existing divergence measures or to construct new divergence measures with custom properties that are tailored to the application. To examine the practical efficacy of these new methods, we evaluate them in a statistical learning framework on a series of real-world data science problems involving speech-based monitoring of neuro-motor disorders.

# ACKNOWLEDGMENTS

This dissertation would not have been possible without the support of my advisor, committee members, colleagues, family and friends.

I would first like to thank my advisor, Visar Berisha for the encouragement he has provided throuzghout my PhD studies. Visar has been instrumental in guiding me to develop my ideas and assisting me in problem-solving issues in my research. The lessons he has taught me have helped me become a better scholar and a better person.

I would also like to thank the other members of my committee. From the day I joined the university, my co-advisor Dr. Andreas Spanias has been encouraging and supportive. He helped me to find my way through this long and arduous process, and has been a wonderful mentor to me. I am also grateful to Dr. Julie Liss. She has been incredibly welcoming of me in her lab, and has remained continuously enthusiastic and receptive to my research. Furthermore, I am grateful for the support of Dr. Daniel Bliss, whose technical expertise has aided me at each stage of my Ph.D.

I would also like to thank Karthikeyan Ramamurthy, Alfred Hero, and Dennis Wei whose technical guidance has been instrumental in the development of the work contained in this dissertation.

I am very appreciative of Mahesh Banavar for his guidance and the friendship he provided in helping me through my first year at ASU.

In addition I would like to thank all of my labmates throughout my time at ASU, for their assistance and companionship. In particular, I would like to acknowledge Ming and Yishan for their continuing friendship. Your presence has made my time at ASU all the more enjoyable.

Finally, I give my upmost thanks to my family for their unconditional love and support.

## TABLE OF CONTENTS

CHAPTER Page

APPENDIX

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

# Chapter 1

# INTRODUCTION

This dissertation will focus on the task of estimating the difference, or *divergence*, between two multivariate probability distributions. Historically, estimation of information-theoretic quantities, including information divergence, has relied on plug-in estimators (methods that first estimate the probability density function, then plug it in to the desired formula). However, [56] explains the problems of plug-in estimators thusly: "on the one hand parameterizing the divergence and entropy functionals with infinite dimensional density function models is a costly over-parameterization, while on the other hand artificially enforcing lower dimensional density parametrization can produce significant bias in the estimator" [56].

In this dissertation, we investigate *direct* estimation of information divergence measures. Direct estimators, methods which bypass density estimation completely, offer an appealing alternative to plug-in estimators, particularly in scenarios where the high dimensionality of the data exacerbates the aforementioned problems. To date, most of the work in direct estimation has focussed on single distribution measures from information theory, e.g., entropy [direct estimation of entropy citations]; however direct estimates of information divergence has been under-explored [direct estimates of divergence]. These methods exploit the relationship between the convergence properties of minimal graphs and information theory. This approach enjoys a number of desirable properties: including faster asymptotic convergence rates (particularly for non-smooth densities in high-dimensional spaces), and the ability to bypass the parameter tuning required for density estimation [57]. In this Dissertation, we aim to extend methods of direct estimation to a wider range of information diver-

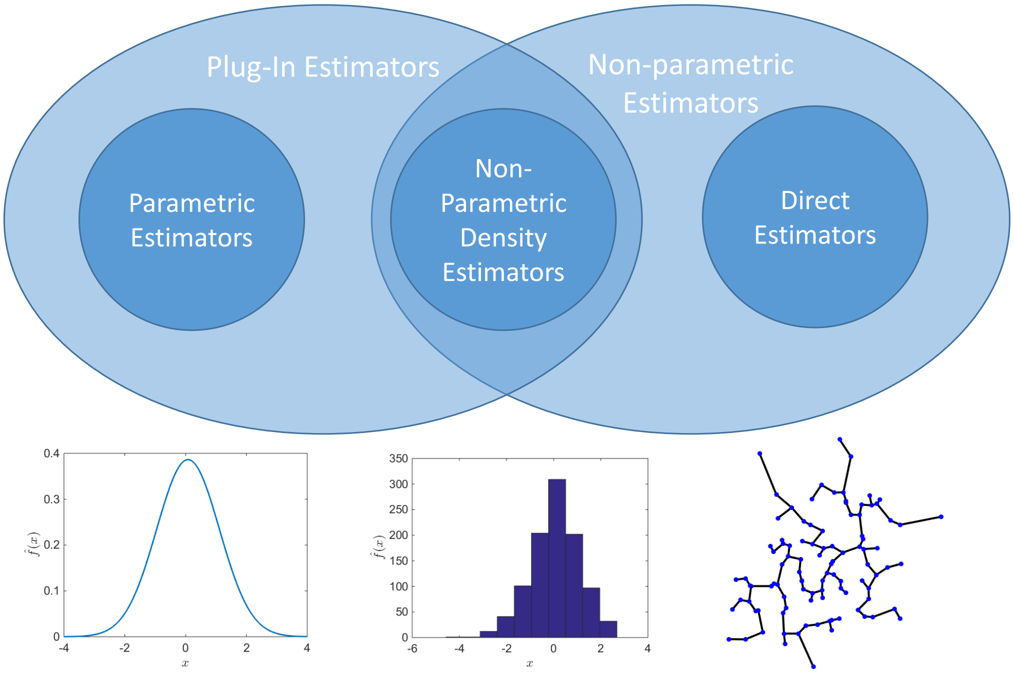


Figure 1.1: Estimation methods for information theoretic quantities.

gence functions, and to allow experimenters the freedom to design their own direct estimators for existing or new divergences.

## 1.1 Parametric Vs. Non-Parametric Vs. Direct Estimation

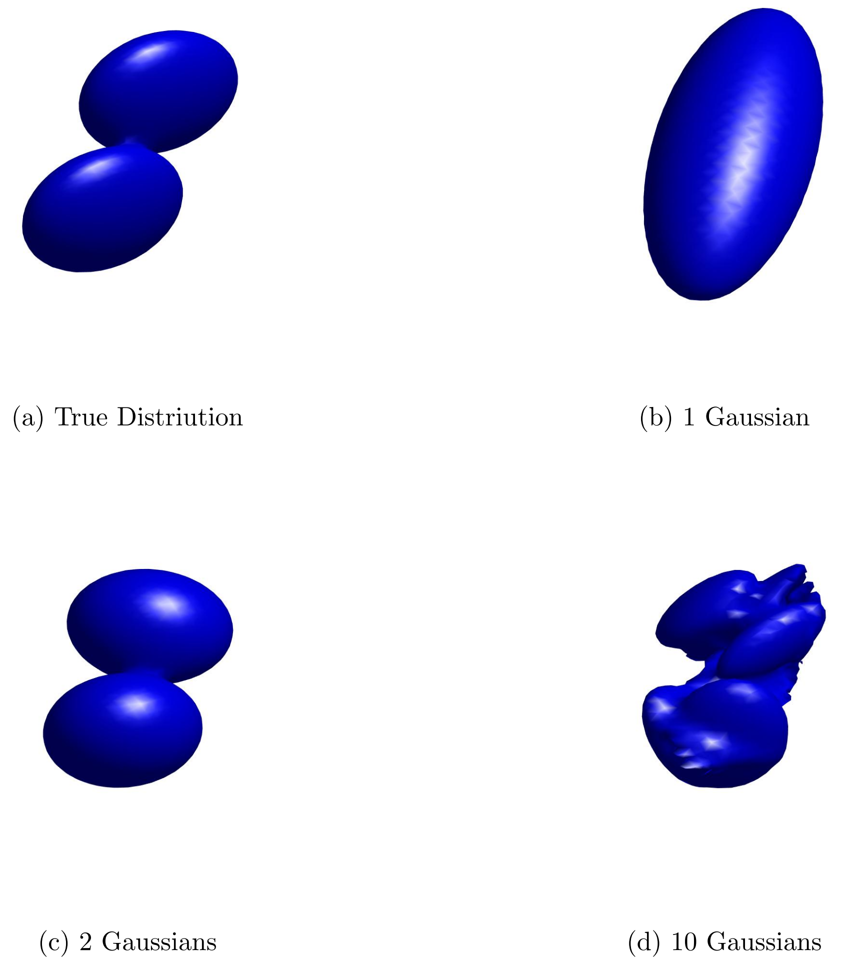
In this Section, we provide a general overview of the three different approaches of estimation in order to provide important context for the remainder of the dissertation. Figure 1.1 provides an overview of the three methods. Plug-in estimators function by first estimating the underlying density functions, then *plugging* them into the appropriate formula to estimate the desired quantity. Parametric plug-in methods, assume that the data fits a parametric model (such as a Gaussian), then estimate the parameters of the chosen model to characterize the distribution. Alternatively, non-parametric plug-in methods estimate the density without using a parametric model e.g., kernel density estimators. Direct methods estimate the quantity of interest directly from the data without performing density estimation.

In comparing these three estimation methods, it is important to note up front, that in the scenario where an accurate parametric model of the data is known, there is little motivation to employ non-parametric estimators. In these scenarios, their $1/\sqrt{N}$ RMS convergence properties [77] will outperform those of non-parametric estimators, and assuming that an analytical expression for the desired information theoretic quantity can be derived the challenges of high-dimensional integration can be avoided as well.

Where this estimation problem gets interesting is the scenario where no accurate parametric model is available, and it is worth accepting the limitations of non-parametric [10, 56, 57, 77, 88] estimation methods in order to avoid the bias of an inaccurate parametric model. In general, this dissertation will focus on scenarios where no accurate parametric model for the data is available. In this Section, we will look at a few academic examples illustrating the perils of plug-in estimators when a flawed parametric model is used and insufficient data exists for a robust non-parametric estimate of the density function.

As an illustrative example, let us consider a sample of data $(\mathbf{x}_i, y_i)$ for $i \in [1, ..., n]$, representing $n = 200$ samples drawn from one of two different class distributions ($f_0(\mathbf{x})$ and $f_1(\mathbf{x})$) in $\mathbb{R}^3$, where $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ are each bimodal Gaussian distributions with unique parameters. Suppose that in this example, we wish to estimate bounds on the Bayes risk,the theoretically optimal performance for a classifier constructed on this dataset, but we have no prior knowledge of the underlying distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$.

If in order to estimate the bounds, we choose to utilize a plug-in estimator, the first step is to form density estimates $\hat{f}(\mathbf{x})$ and $\hat{g}(\mathbf{x})$ for each of the underlying distributions. Considering first the task of estimating $f_0(\mathbf{x})$, a graphical representation of



(a) True Distriution

(b) 1 Gaussian

(c) 2 Gaussians

(d) 10 Gaussians

Figure 1.2: Parametric density estimation using Gaussian mixture models.

this distribution can be found in Figure 1.2a. In this scenario the optimal parametric model is a mixture of two Gaussians, and as a result attempting to model this distribution using a GMM with two mixtures will achieve a relatively accurate fit (see Figure 1.2c). If however we attempt to model the distribution using only a single Gaussian, as seen in Figure 1.2b, we end up with a heavily biased solution that places the center of the distribution as the empty space between the two modes of the true density function. If, on the other hand, we attempt to model the distribution using

(a) True Distriution

(b) Kernel density estimate

Figure 1.3: True distribution and KDE using Gaussian kernel.

an excessive number of Gaussians, the GMM will overfit the data and end up with a heavily distorted solution.

Now suppose that perform non-parametric estimation of the density function using Kernel density estimation. Using implement non-parametric density estimation using Gaussian kernels [67]. Kernel density estimation resemble a mixture density that allows for one mixture for every data point in the data set, as a result they suffer from the same overfitting problems yielded shown in the previous example, and typically yield a high-variance estimate. The density estimate yielded by this approach is shown in Figure 1.3b. In this case the selected bandwidth appears to be too small, and the resulting density estimate assigns near zero probabilities to anywhere outside immediate vicinity of points in the sample data.

As you can see, the limited number of samples provided combined with the lack of prior knowledge regarding the underlying distributions leads to major challenges in density estimation. And, unfortunately, without an accurate estimate of the underlying densities, our ability to accurately information divergence functions is severely

Table 1.1: Parameters of $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$.

| | $f_0(\mathbf{x})$ | | $f_1(\mathbf{x})$ | |
|---|---|---|---|---|
| | Gaussian 1 | Gaussian 2 | Gaussian 1 | Gaussian 2 |
| $\mu$ | $[-2.25\ -2.25\ -2.25]$ | $[.75\ .75\ .75]$ | $[-.75\ -.75\ -.75]$ | $[2.25\ 2.25\ 2.25]$ |
| $\mathbf{\Sigma}$ | $\mathbf{I_3}$ | $\mathbf{I_3}$ | $\mathbf{I_3}$ | $\mathbf{I_3}$ |

inhibited. In these scenarios, graph-based (or direct) estimation becomes a very appealing alternative to the traditional plug-in estimators. To illustrate this, we consider the challenge of estimating the $D_p$-divergence [16], a directly estimable divergence function which we will discuss further in Chapter 3. Consider two 3-dimensional distributions $f_0(\mathbf{x})$ and $f_2(\mathbf{x})$, each composed of 2 Gaussians with properties described in Table 1.1. Figure 1.4 illustrates how each of the previously described methods functions in estimating the $D_p$-divergence for sample sizes varying from $N = 50$ to $N = 500$ averaged across 100 Monte-Carlo trials. As expected, when the parametric does not fit the data, it's prediction is highly biased, and doesn't converge to the correct value. The non-parametric method converges to the ground truth, but contains a large amount of finite sample bias, and converges rather slowly with $N$. The direct estimate however, contains little finite sample bias, and converges rapidly to the correct solution. Comparison between plug-in and direct estimation methods will be explored in greater detail in Chapter 2.

## 1.2 Problem Statement

This dissertation is motivated by the research task of non-invasive monitoring of neurological health through speech data, particularly for individuals with a set of motor-speech disorders known as Dysarthria. Data collected in this domain is high dimensional, non-Gaussian and plagued with complex interdependencies amongst seem-

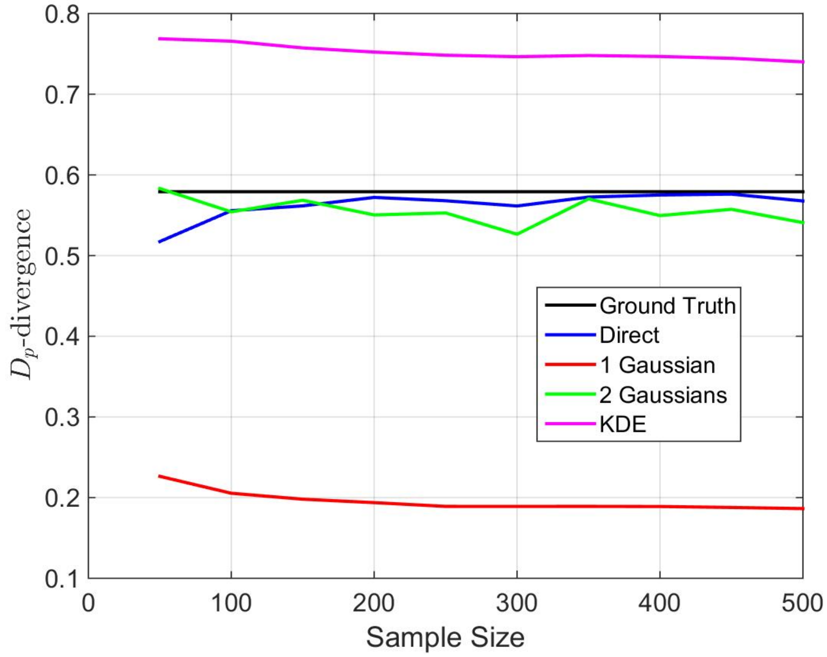


Figure 1.4: Parametric, non-parametric, and direct estimates of the $D_p$-divergence plotted as a function of sample size ($N$).

ingly unrelated features. These features make density estimation highly challenging, and thus limit the accuracy of divergence estimates which rely on them. While there currently exist some methods of directly estimating information divergence functions, the set of divergence functions that can be estimated is profoundly limited. This dissertation aims to expand this methodology to a larger family of divergences, and to create a framework which will enable non-expert users to construct new divergences.

## 1.3 Contributions Of This Dissertation

The contributions of this dissertation include:

1. A new divergence measure that is directly estimable using minimum spanning trees (MSTs) is introduced [16, 118] (Chapter 3).

2. Using this divergence measure we introduce bounds on the Bayes risk for binary classification, and show these bounds to be provably tighter than the commonly used Bhattacharyya bounds [16, 118] (Chapter 3.2).

3. Extend proposed binary classification bounds to

   - Domain adaptation problems [16, 118] (Chapter 3.3).
   - Multi-class problems [120] (Chapter 4.1 ).
   - Regression problems [119] (Chapter 4.2).

4. We introduce a set of fitting rules that can be used to directly estimate *any* unknown information-theoretic quantity (Chapter 5).

## 1.4 Outline Of Dissertation

Chapter 2 surveys the relevant literature in information theory and machine learning. Chapter 3 introduces a directly estimable divergence measure from which we develop bounds on the Bayes error in single-domain and cross-domain learning problems. Chapter 4 extends the bounds from the previous Chapter to multi-class and regression problems. Chapter 5 introduces a procedure for constructing new directly estimable quantities. Finally Chapter 6 uses the bounds from the previous Chapters to develop preprocessing algorithms for machine learning problems and evaluates their utility on some health analytics problems in the speech domain.

# Chapter 2

# LITERATURE SURVEY

In this chapter, we will review the relevant literature on a number of information divergence measures, the methods by which they can be estimated, and their applications in the field of machine learning.

## 2.1 Measures Of Entropy And Divergence

Information theoretic measures, such as entropy and divergence, play a key role in a number of problems throughout the fields of signal processing and machine learning. They have been extensively used in many signal processing applications involving classification [85], segmentation [48], source separation [59], indexing [56], clustering [6], and other domains. Information divergences can be simply thought of as a measure of how different two distributions are. While there exist a number of different divergence measures of varying properties, they all share the characteristic of increasing as the two measure distributions "move apart" [2].

A common class of divergence functions are known as $f$-divergences, which were introduced independently by Csiszar [27] and Ali and Silvey[2]. Any $f$-divergence can be represented in the form

$$D_\phi(f_0||f_1) = \int f_1(\mathbf{x})\phi\Big(\frac{f_0(\mathbf{x})}{f_1(\mathbf{x})}\Big)d\mathbf{x}, \tag{2.1}$$

for measured distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$, where $\phi(t)$ is a generating function of the likelihood ratio $t$ unique to a each $f$-divergence. Additionally, all $f$-divergences have the following properties (see [3, 24])

- **Non-Negativity**: $D(f_0||f_1) \geq \phi(1) = 0$

Table 2.1: Table of common $f$-divergence measures.

| $f$-divergence | $\phi(t)$ | $D(f_0\|\|f_1)$ |
|---|---|---|
| KL-Divergence | $t\log(t)$ | $\int f_0(\mathbf{x})\log\left(\frac{f_0(\mathbf{x})}{f_1(\mathbf{x})}\right)d\mathbf{x}$ |
| Hellinger Distance | $(\sqrt{t}-1)^2$ | $\frac{1}{2}\int(\sqrt{f_0(\mathbf{x})}-\sqrt{f_1(\mathbf{x})})^2d\mathbf{x}$ |
| $\alpha$-Divergence | $\frac{4}{1-\alpha^2}(1-t^{\frac{1+\alpha}{2}})$ | $\frac{4}{1-\alpha^2}\left(1-\int f_0^{\frac{1-\alpha}{2}}(\mathbf{x})f_1^{1+\alpha}(\mathbf{x})d\mathbf{x}\right)$ |
| TV-Divergence | $\frac{1}{2}\|t-1\|$ | $\frac{1}{2}\int\|f_0(\mathbf{x})-f_1(\mathbf{x})\|d\mathbf{x}$ |

- **Convexity**: $\phi(t)$ is convex on $t > 0$

- **Monotonicity**: For an arbitrary transform $\mathbf{x}' = T(\mathbf{x})$,

$$D_\phi(f_0||f_1) \geq D(f_0T^{-1}||f_1T^{-1})$$

- **Scaling**:For any positive constant $c$

$$cD_\phi(f_0||f_1) = D_{c\phi}(f_0||f_1)$$

This family of $f$-divergences have become increasingly popular in the field of machine learning where they are frequently used in the development of surrogate loss functions for the probability of error. Since it is computationally intractable to minimize the probability of error directly, we often employ surrogate loss functions. $f$-divergences have become a common choice for this task in large part due to Blackwells theorem [19], which states that if procedure $A$ has a greater $f$-divergence than procedure $B$, then there exists a set of prior probabilities such that $A$ will have lower error probability than $B$. Table 2.1 introduces the functional representations for a few divergence measures that are found to be most relevant in the machine learning literature. In the following sections, we will discuss these divergences and the problem of estimating them in high dimensional spaces.

### *2.1.1 Rényi Entropy And Divergence*

The Rényi entropy is a generalized measure of entropy which can be defined as

$$H_\alpha = \frac{1}{1-\alpha}\log\int f^\alpha(\mathbf{x})d\mathbf{x}. \tag{2.2}$$

for distribution $f(\mathbf{x})$ and parameter $\alpha$. The Rényi entropy represents a generalization of the well known Shannon entropy ($\alpha = 1$). Among other interesting special cases of the Rényi entropy are the Hartley entropy ($\alpha = 0$), collision entropy ($\alpha = 2$), and the min entropy ($\alpha = \infty$)[26].

The Rényi $\alpha$-divergence was proposed by Alfred Rényi [97] as a general dissimilarity measure between distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$:

$$D_\alpha(f_0||f_1) = \frac{1}{1-\alpha}\log\int f_1(\mathbf{x})\left(\frac{f_0(\mathbf{x})}{f_1(\mathbf{x})}\right)^\alpha d\mathbf{x}. \tag{2.3}$$

where $\alpha \in (0,1)$. Although the $\alpha$-divergence can specialized to numerous forms depending on the choice of $\alpha$, the two most popular cases are the Hellinger dissimilarity, which is attained when $\alpha = \frac{1}{2}$, and can be related to the Hellinger distance by

$$\begin{aligned} D_{\text{Hellinger}}(f_0||f_1) &= \int\left(\sqrt{f_0(\mathbf{x})} - \sqrt{f_1(\mathbf{x})}\right)^2 d\mathbf{x} \\ &= 2\left(1 - exp\left(\frac{1}{2}D_{\frac{1}{2}}(f_0||f_1)\right)\right). \end{aligned} \tag{2.4}$$

The Bhattacharyya coefficient, introduced by Bhattacharyya [17], represents a measure of similarity between distributions defined by

$$BC(f_0,f_1) = \int\sqrt{f_0(\mathbf{x})f_1(\mathbf{x})}d\mathbf{x}. \tag{2.5}$$

Both the Hellinger distance [51], also known as the Matusita measure [82],

$$D_{Hellinger}(f_0||f_1) = 1 - BC(f_0,f_1) \tag{2.6}$$

and the Bhattacharyya distance

$$D_B(f_0,f_1) = -\log(BC(f_0,f_1)) \tag{2.7}$$

can be represented in terms of the Bhattacharyya coefficient. Whereas the Hellinger distance represents a special case of the $\alpha$-divergence, the Bhattacharyya distance represents a special case of the Chernoff distance [7].

Similarly, we can represent the KL-divergence as the limit of the $\alpha$-divergence as $\alpha \rightarrow \infty$

$$D_{KL}(f_0||f_1) = \lim_{\alpha \to 1} \log \int f_1(\mathbf{x}) \left( \frac{f_0(\mathbf{x})}{f_1(\mathbf{x})} \right)^{\alpha} d\mathbf{x} = \int f_1(\mathbf{x}) \log \left( \frac{f_1(\mathbf{x})}{f_0(\mathbf{x})} \right). \tag{2.8}$$

The Kullback-Leibler (KL) divergence was originally proposed in [70], and has become a widely used criteria in variable selection and decision tree construction. Part of the reason for the popularity of the KL-divergence is that it is the only case of the $\alpha$-divergence in which the chain rule of conditional probability holds exactly. The KL-divergence is an $f$-divergence with the function $f(t) = t \log(t)$. The KL divergence can be viewed in terms of the information entropy as

$$D_{KL}(f_0 \| f_1) = H(f_0, f_1) - H(f_0), \tag{2.9}$$

where $H(f_0)$ entropy in $f_0(\mathbf{x})$ and $H(f_0, f_1)$ represents the cross entropy between $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$, defined as

$$H(f_0) = -\int f_0(\mathbf{x}) \ln[f_0(\mathbf{x})] d\mathbf{x} \tag{2.10}$$

and

$$H(f_0, f_1) = -\int f_0(\mathbf{x}) \ln[f_1(\mathbf{x})] d\mathbf{x}. \tag{2.11}$$

The information gain criteria was originally proposed for feature selection in [60]. In addition to its widespread use in feature selection, it is also used as the primary criteria for building decision trees in the extremely popular Iterative Dichotomiser 3 (ID3) algorithm introduced in [93].

### *2.1.2 The $\alpha$-Jensen Difference*

The Jensen-difference is a method of deriving distance measures from entropy functionals, and can be applied to the $\alpha$-entropy to form the $\alpha$-Jensen dissimilarity measure. While the $\alpha$-Jensen dissimilarity does not meet the properties of $f$-divergences, it can be represented in terms of the entropy of the underlying distributions and is therefore directly estimable using minimal graphs.

Rao introduced the Jensen difference

$$J(p_0, f_0, f_1) = H(f_0, f_1) - p_0 H(f_0) - (1 - p_0) H(f_1)) \tag{2.12}$$

as a general measure of the entropic difference between two functions $f_0$ and $f_1$, with prior probabilities $p_0$ and $p_1 = (1 - p_0)$ respectively [94]. When the Rényi entropy is used, this becomes the $\alpha$-Jensen dissimilarity measure

$$\begin{aligned} J_\alpha(p_0, f_0, f_1) &= H_\alpha(f_0, f_1) - p_0 H_\alpha(f_0) - (1 - p_0) H_\alpha(f_1)) \\ &= \frac{1}{1-\alpha}\left[\log \int [p_0 f_0(\mathbf{x}) + (1 - p_0) f_1(\mathbf{x})]^\alpha d\mathbf{x} - p_0 \int f_0^\alpha(\mathbf{x}) d\mathbf{x} - (1 - p_0) \int f_1^\alpha(\mathbf{x}) d\mathbf{x} - \right]. \end{aligned} \tag{2.13}$$

The $\alpha$-Jensen dissimilarity measure has been used for image registration problems [54, 77] and was originally proposed for assessing the complexity of time-frequency distribution images [83].

## 2.2 Estimation Of Information Divergence Functions

There are three general approaches for estimating information divergence functions: parametric estimation, non-parametric estimation, and direct estimation. The first two approaches [56] are often referred to as *plug-in estimators*, since they involve first estimating the density function for each random variable, then plugging them into the divergence formula (2.1), and only vary in the manner in which the density function is estimated. Parametric estimation assumes that the underlying distribution fits a particular parametric model and estimates the parameters of this model. Though parametric methods have a number of desirable properties when an accurate model can be chosen, their solution can be heavily biased if the model doesn't accurately fit the true distribution. Non-parametric methods employ non-parametric density estimation. While these methods avoid the pitfalls of model-dependent estimates, non-parametric density estimators are generally high variance, subject to outliers, and perform poorly without stringent smoothness conditions. Additionally, without a parametric model for the underlying distributions, (2.1) must be calculated numerically, a task which can be difficult in the high-dimensional spaces found in many practical applications. The third method which is of growing interest due to its ability to combat the limitations of plug-in estimators is direct estimation. Direct estimation, or graph-based estimation, employs the asymptotic properties of minimal graphs to *directly* estimate information-theoretic quantities without the need to ever estimate the underlying densities.

### *2.2.1 Plug-In Estimation : Parametric*

Parametric methods are the simplest and most popular way of estimating information divergence functions. Parametric estimation assumes that the underlying distribution fits a predefined parametric model. Using this approach, we are able to model the underlying density simply by estimating a few necessary parameters. The ease of computation and high convergence rate [77], make parametric estimation appealing for a variety of practical applications. In addition, when there is an analytical solution for the desired parametric model, we can calculate the desired quantity without the challenges of integration, which can be infeasible in high-dimensional spaces. Table 2.2 displays analytical solutions for some of the common divergence measures

Table 2.2: Analytical solutions for divergence functions of multivariate normal distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ with mean and covariance matrices $\mu_0, \mu_1$ and $\Sigma_0, \Sigma_1$ respectively where $\Delta = \mu_1 - \mu_0$.

| $f$-divergence | Analytical Solution |
|---|---|
| KL-Divergence | $\frac{1}{2}\Big[\log\big(\frac{det(\Sigma_0)}{det(\Sigma_1)} - d + tr(\Sigma_1^{-1}\Sigma_0) + \Delta^T\Sigma_1^{-1}\Delta\big)\Big]$ |
| Hellinger Distance | $1 - \sqrt{\frac{\lvert\frac{1}{2}(\Sigma_0+\Sigma_1)\rvert}{\sqrt{\lvert\Sigma_0\rvert\lvert\Sigma_1\rvert}}exp\{\frac{1}{8}\Delta^T\left(\frac{\Sigma_0+\Sigma_1}{2}\right)^{-1}\Delta\}}$ |
| $\alpha$-Divergence | $-\frac{1}{2}\log\left(\frac{\lvert\Sigma_0\rvert^{\alpha}\lvert\Sigma_1\rvert^{1-\alpha}}{\lvert\alpha\Sigma_0+(1-\alpha)\Sigma_1\rvert}\right) + \frac{\alpha(1-\alpha)}{2}\Delta^T(\alpha\Sigma_0 + (1-\alpha)\Sigma_1)^{-1}\Delta$ |

discussed in this Dissertation.

The fundamental drawback of parametric estimates is that they depend on the data fitting correct parametric form in order to be asymptotically unbiased. In general these approaches perform very well when there basic assumptions hold, however when these assumptions are violated the performance can significantly degrade. This will be observed in several simulations throughout this dissertations where parametric methods are used as a baseline for comparison. Because we will rarely find data that exactly fits any common parametric model this drawback presents a major concern in applying parametric estimation to real world problems.

### *2.2.2 Plug-In Estimation : Non-Parametric*

Non-parametric estimation offers an appealing alternative to parametric methods in scenarios where no accurate parametric model is available. Parametric estimates will generally converge faster than non-parametric estimates, however if the parametric model does not fit the data the solution that it converges to could be heavily biased [102]. In general, relative to parametric methods, these methods offer us universal consistency in exchange for slower convergence rates and the computational challenges that come with non-parametric density estimation.

The performance of non-parametric plug-in estimators is dependent on the methods used for density estimation and integral estimation. In density estimation, we face the problem of estimating an underlying distribution $f(\mathbf{x})$ given a set of data points $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ that are drawn from it. Histograms, kernel density estimators, k-NN density estimators and projection pursuit density estimators are all popular choices for this problem. Histograms partition the range of data set into a predefined number of bins, then predict the density in each bin to equal the number of points in that range divided by the total number of points in the dataset. Histograms are the simplest solution however they are highly sensitive to the choice of origin and bin width parameters. For example a histogram estimate will assign zero probability to a point if there exist no samples from $\mathbf{X}$ in the bin it falls in. This is true even if there exist several samples nearby that fall into adjacent bins, thus the assigned density estimate is highly sensitive to the size and location of each of the bins.

Alternatively, kernel density estimation provides a smooth density estimation, without endpoints. Kernel density estimators function by placing some kernel, such as the Gaussian kernel, on each of the data points $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ in $\mathbf{X}$. This can be calculated by

$$\hat{f}_K(\mathbf{x}) = \frac{1}{nh} \sum_{i=1}^{N} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \tag{2.14}$$

where K represents the kernel function and $h$ represents the kernel bandwidth. Kernel density estimators offers a smoothed density estimate free of the discontinuities found in histogram estimates, but are still dependent on the choice of kernel and kernel bandwidth [110]. Utilizing KDE in higher dimensions typically relies on the product kernel and assumes each of the dimensions to be uncorrelated [102].

The method of k-NN density estimation attempts to estimate the density at any point in space $\mathbf{x}$ by assessing the distance to the $k$th nearest neighbor from that point

$D_k(\mathbf{x})$. Because we know that therer are $k$ points points within the hypersphere of radius $D_k(\mathbf{x})$, the density $\hat{f}(\mathbf{x})$ can be estimated by

$$\hat{f}(\mathbf{x}) = \frac{k}{Nvol_k(\mathbf{x})} \tag{2.15}$$

where $vol_k(\mathbf{x})$ represents the volume of said hypersphere. This approach can be a powerful tool for estimating the density at a particular point, it is not particularly successful for estimating the entire density function [62].

For the task of estimating entropy in a non-parametric fashion Berilant suggests there are four general classes of non-parametric integral estimators 1) Integral estimates 2) resubstitution estimates 3) splitting data estimates and 4) cross-validation estimates [10]. While all of these methods follow the basic procedure of first estimating the density, then utilizing the density estimate to calculate the entropy, the way in which the second step is performed varies across the four different approaches.

Integral estimates utilize a given set of data $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$ to form a non-parametric estimate $\hat{f}(\mathbf{x})$ of the underlying distribution $f(\mathbf{x})$ and estimate the entropy using

$$\hat{H}(f) = -\int_A \hat{f}(\mathbf{x}) \log[\hat{f}(\mathbf{x})] d\mathbf{x}, \tag{2.16}$$

where $A$ represents the region of integration which typically excludes small or tail values of $\hat{f}(\mathbf{x})$. This process was originally proposed in [31], and is perhaps the most straight forward approach to non-parametric entropy estimation. The fundamental limitation to this approach lies in the calculations of (2.16), which presents a major challenge in higher dimensions. While both the KDE and k-NN methods are, in general, more easily extended to higher dimensions than histograms, their convergence in higher dimensions is often glacially slow and on the order of $O(N^{-\frac{\gamma}{d}})$, where $\gamma > 0$ is a rate parameter [104]. As a result, some methods such as projection pursuit density estimation, which attempts to estimate the distributions structure based on "interesting" low-dimensional projections [37]. Ensemble methods have been proposed to achieve the dimension invariant rate of $O(N^-1)$

Resubstitution estimates once again form an estimate $\hat{f}(\mathbf{x})$ of the underlying distribution $f(\mathbf{x})$ using the dataset $\mathbf{X}$, however instead of integrating across an entire region of space, it is evaluated at all of the points in the data set using

$$\hat{H}(f) = -\frac{1}{N}\sum_{i=1}^{N}\log[\hat{f}(\mathbf{x}_i)]. \tag{2.17}$$

These methods have shown mean square consistency ($\underset{N\to\infty}{lim} E[(\hat{H}(f) - H(f))^2] = 0$) and avoid much of the computational challenges of integral estimates [1].

Data splitting estimates begin by partitioning the data into two groups $\mathbf{X}_a = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l$ and $\mathbf{X}_b = \mathbf{x}_{l+1}, \mathbf{x}_{l+2}, ..., \mathbf{x}_N$. First $\mathbf{X}_a$ is used to estimate $f(\mathbf{x})$, then the entropy of this estimate is evaluated across all of the points in $\mathbf{X}_b$ via

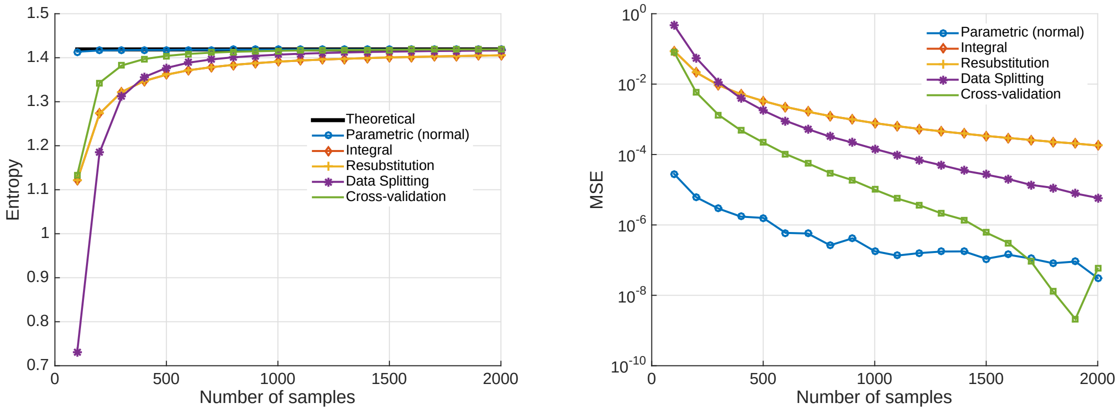
$$\hat{H}(f) = -\frac{1}{N-l}\sum_{i=l+1}^{N} I_{[\mathbf{x}_i \in A]} \log \hat{f}(\mathbf{x}_i). \tag{2.18}$$

Finally, cross-validation estimates function similarly to the data splitting estimates, however rather than splitting the data single time, this approach forms $N$ different estimates $\hat{f}^{(i)}(\mathbf{x})$ of the underlying distribution $f(\mathbf{x})$ each calculated on the set $\mathbf{X} \setminus \mathbf{x}_i$. The entropy is then estimated using

$$\hat{H}(f) = -\frac{1}{N}\sum_{i=1}^{N} I_{[\mathbf{x}_i \in A]} \log[\hat{f}^{(i)}(\mathbf{x}_i)], \tag{2.19}$$

thus evaluating each density estimate $\hat{f}^{(i)}(\mathbf{x})$ on the sample that was excluded in its estimation procedure. Ivanov and Rozhkova originally proposed this approach for kernel density estimators and showed it to be strongly consistent [61].

As a simple illustration of how each of these methods each of these methods work, we consider a simple univariate standard normal distribution, and evaluate the performance of each of the previously described integral estimation methods



(a) Entropy plot

(b) MSE of entropy estimators

Figure 2.1: Histogram estimates of the entropy of the standard normal distribution for various integral estimators.

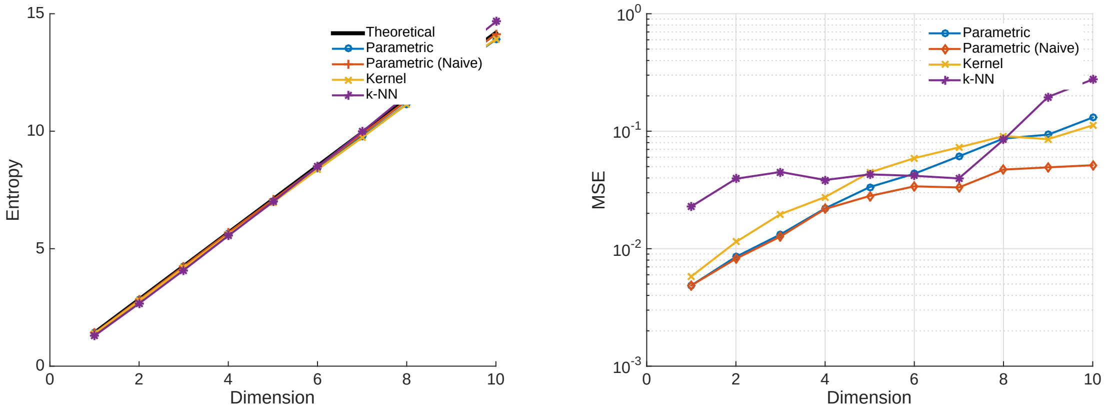
based on a histogram density estimation. The sample size is then varied across $N = 100, 200, ..., 2000$ and the simulation is repeated across a 100000 iteration Monte Carlo simulation. The large number of iterations chosen for this particular simulation is necessary in order to detect the small differences in the error of each method and possible due to the ease of computation in the 1-dimensional case. The performance of each of estimator is displayed in terms of its mean squared error (MSE) in Figure 2.1. In this scenario, the cross-validation estimate consistently outperforms the other approaches, and even approaches the performance of the baseline parametric estimate at higher samples. This comes at the cost of significantly greater computational costs than resubstitution and data splitting estimates. The integral and resubstitution estimates, which are equivalent for histograms, outperform the data splitting estimate at lower samples, but inferior at higher samples.

Now to better understand how the performance of the different density estimators varies with dimension, we will evaluate several different non-parametric density estimation methods when applied to two different multivariate normal distributions $N(0_d, \mathbf{I}_d)$ and $N(0_d, \mathbf{\Sigma}_d)$ where

$$\mathbf{\Sigma}_d = \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,d} \\ \sigma_{2,1} & \sigma_{2,2} & \cdots & \sigma_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d,1} & \sigma_{d,2} & \cdots & \sigma_{d,d} \end{bmatrix} \tag{2.20}$$

for $\sigma_{i,j} = 0.9^{|i-j|}$ and $\mathbf{I}_d$ is the $d$-dimensional identity matrix. We evaluate both kernel and k-NN density estimation methods where the integral is approximated by resubstitution and vary dimension from $1 - 10$. In order to compute the kernel density estimate in higher dimensions, it is necessary to assume that all dimensions are independent and that the the multivariate density at any point can be calculated by the product of the univariate density estimates for each dimension. We also compute two parametric estimates, one estimate based on a general multivariate Gaussian model along with a naive parametric estimator that similarly to the kernel estimate assumes each of the dimensions to be independent. Estimates are made for $N = 100$ samples, and results are averaged across a 100 iteration Monte-Carlo simulation. Results for distributions $N(0_d, \mathbf{I}_d)$ and $N(0_d, \mathbf{\Sigma}_d)$ are displayed in Figures 2.2 and 2.3 respectively.

In the first case, we find that the naive parametric estimates yields the best performance for all dimensions. The general parametric model performs similarly to the naive model in lower dimensions (identical for $d = 1$), however it's performance deteriorates in higher dimensions. The k-NN density estimate performs the worst in general, however from dimensions $5 - 8$ it performs relatively strong as the finite sample bias switches from negative to positive. The kernel estimate performs slightly worse than the parametric estimates for lower dimensions, however it eventually outperforms the general parametric model in higher dimensions where the difficulty of



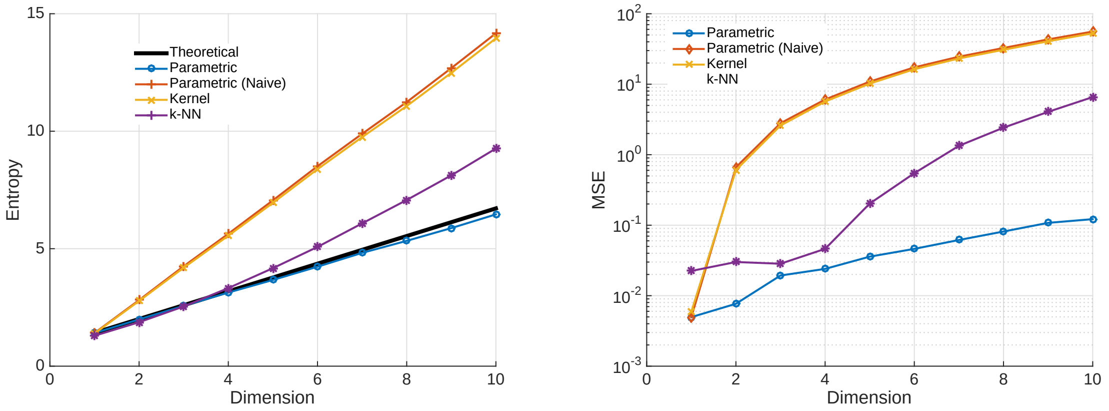
(a) Entropy plot

(b) MSE of entropy estimators

Figure 2.2: Substitution estimates for entropy of distribution $N(0_d, \mathbf{I}_d)$ plotted across dimension.

parameterizing a $d \times d$ covariance matrix exceeds that of non-parametric density estimation.

In the second case when there is some degree of dependency across the various dimensions, the performance of both the kernel estimate and naive parametric estimate deteriorate rapidly with dimension, since their failed assumption leads to an increasingly large asymptotic bias. While the k-NN estimate has a rather large bias in the higher dimensions due to the small sample size, its finite sample performance still greatly exceeds the two previous estimates in this scenario. The general parametric estimate expectedly yields the best overall performance. While these results are based on one specific scenario, and we cannot generalize these findings to any distribution, they illustrate how rapidly the violation of any of the assumptions used in parametric and even non-parametric estimators can deteriorate a models performance. While non-parametric estimates may not converge as fast as parametric models, they will often far outperform a parametric model when fundamental assumptions are violated.



(a) Entropy plot

(b) MSE of entropy estimators

Figure 2.3: Substitution estimates for entropy of distribution $N(0_d, \mathbf{\Sigma}_d)$ plotted across dimension.

### *2.2.3 Review Of Graph Theory*

In order to understand some of the concepts related to minimal graphs which will be used in the following section, and throughout the rest of this Dissertation, we provide a brief review of some relevant concepts in graph theory. For a more thorough review of concepts related to graph theory see [115]. To begin let us introduce some relevant vocabulary:

- **Graph**:A graph $G$ is a triple containing
  - a vertex set $V(G)$
  - an edge set $E(G)$
  - a relation that associates each edge with two vertices $v_i, v_j \in V(G)$
- **Tree**: A connected acyclic simple graph.

- **Subgraph**: A subgraph $H$, of a graph $G$, is a graph whose vertices $V(H)$ are a subset of $V(G)$ and whose edges $E(H)$ are a subset of $E(G)$.
- **Spanning Subgraph**: A subgraph $H$, that contains the same vertex set as $G$, $V(H) = V(G)$.
- **Spanning Tree**: A Spanning subgraph $H$ that is acyclic and connected.
- **Minimum Spanning Tree**: Spanning subgraph $H$ that is acyclic and connected, that minimizes the total weight of the edge set $W(E(H))$.

In the next Section, we will discuss minimum spanning trees in more depth and their use in direct estimation.

### *2.2.4 Direct Estimation*

The convergence properties of minimal graphs have been studied extensively in past years [9, 96, 106, 125]. In some cases, we can exploit knowledge of these convergence properties to directly estimate known quantities without the need for density estimation.

Consider some data set $\mathbf{X}$ which defines $N$ points within a $d$-dimensional space. We can define some graph $G = (V, E)$, that minimizes the length function

$$L_\gamma(G) = \sum_{e \in E} \|e_{i,j}\|^\gamma \tag{2.21}$$

where $\|e_{i,j}\|$ represents the euclidean distance between instances $i$ and $j$ in $\mathbf{X}$, and $\gamma$ is a power weighting constant $0 < \gamma < d$, and meets dome desired properties (e.g. spanning, connected).

Assume that the points in $\mathbf{X}$ are i.i.d. samples drawn from a Lebesgue multivariate density $f(\mathbf{x})$ and that $L_\gamma$ is continuous quasi additive, then $L_\gamma$ will converge in the following manner

$$\lim_{n\to\infty} L_\gamma(G)/n^{\frac{(d-\gamma)}{d}} = \beta_{d,\gamma}\int f^{\frac{(d-\gamma)}{d}}(\mathbf{x})d\mathbf{x}, \tag{2.22}$$

where $\alpha = (d-\gamma)/d$ and $\beta_{d,\gamma}$ is a constant independent of $f$. Now suppose that we want to estimate the Rényi entropy:

$$H_\alpha = \frac{1}{1-\alpha} log \int f^\alpha(\mathbf{x})d\mathbf{x}. \tag{2.23}$$

We can rearrange (2.22) to attain

$$\hat{H}_\alpha = \frac{1}{1-\alpha}\Big[\log\Big(\frac{L_\gamma(G)}{n^\alpha}\Big) - \log(\beta_{d,\gamma})\Big] \tag{2.24}$$

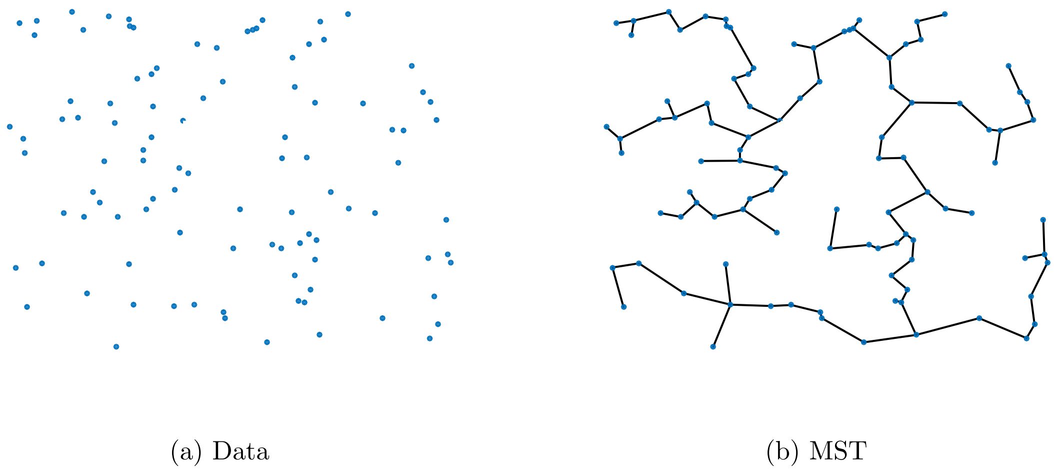
where $\hat{H}_\alpha$ is a direct estimate of Rényi entropy that will be asymptotically consistent. Because $\beta_{d,\gamma}$ doesn't depend on the underlying distribution $f$, it can be estimated offline. A simple method of doing so is to estimate the limit of $L_\gamma(G)/n^\alpha$ as $n \to \infty$ by using a Monte Carlo simulation for a large number of samples drawn from a $d$-dimensional unit cube.

$\hat{H}_\alpha$ can be calculated for any value of $\alpha \in [0,1]$, however $\gamma$ must be adjusted accordingly. While this requires a recalculation of the bias term and the length functional, reconstruction of the minimal graph is unnecessary.

#### 2.2.4.1 Minimum Spanning Trees Estimators

Minimum spanning trees, are a useful tool in direct estimation. Due in large part to the study of the asymptotic graph length [53, 106].

Note that in this case (and all future cases discussed in this dissertation) $T$ is a subgraph of the complete graph on $\mathbf{X}$ and the length function we are trying to minimize is the euclidean distance across all edges $e \in E(T)$. Figure 2.4 illustrates what a minimum spanning tree on a set of uniformly distributed points in $\mathbb{R}^2$ would look like.



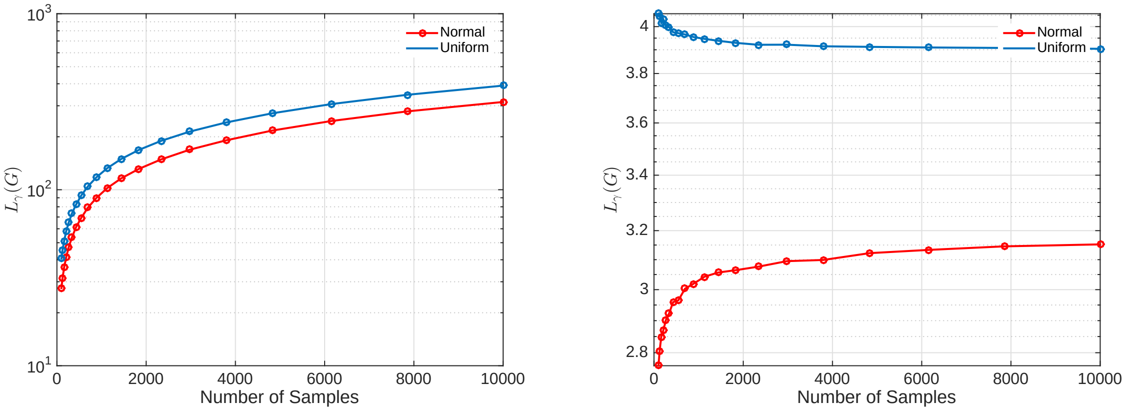
(a) Data

(b) MST

Figure 2.4: Scatter plot of 100 uniformly distributed data points in $\mathbb{R}^2$ and the corresponding euclidean MST.

One simple application of MSTs that we discussed in Section 2.2.4, is the estimation of the Rényi entropy. This idea, which is based on the works of Steele [106] and Redmond et al. [96] and has been extensively investigated by [56, 57, 77, 88], relies on the fact that the length function of some minimal graphs, such as MSTs, will converge to a known function of the underlying distribution (2.22) that can be related to entropy.

We illustrate the convergence of the graph length function and the normalized graph length functions of both uniformly and normally distributed distributions in Figure 2.5. One important observation that can be made from this figure is that while the normalized graph lengths for both distributions asymptotically converge to a constant value, their rates of convergence are noticeably different. In particular, we note that the length function converges faster in the uniformly distributed points. It is also worth noting that while the uniform distribution exhibits a positive finite sample bias for the mean length function, the bias for the normal distribution is negative.



(a) Unnormalized

(b) Normalized

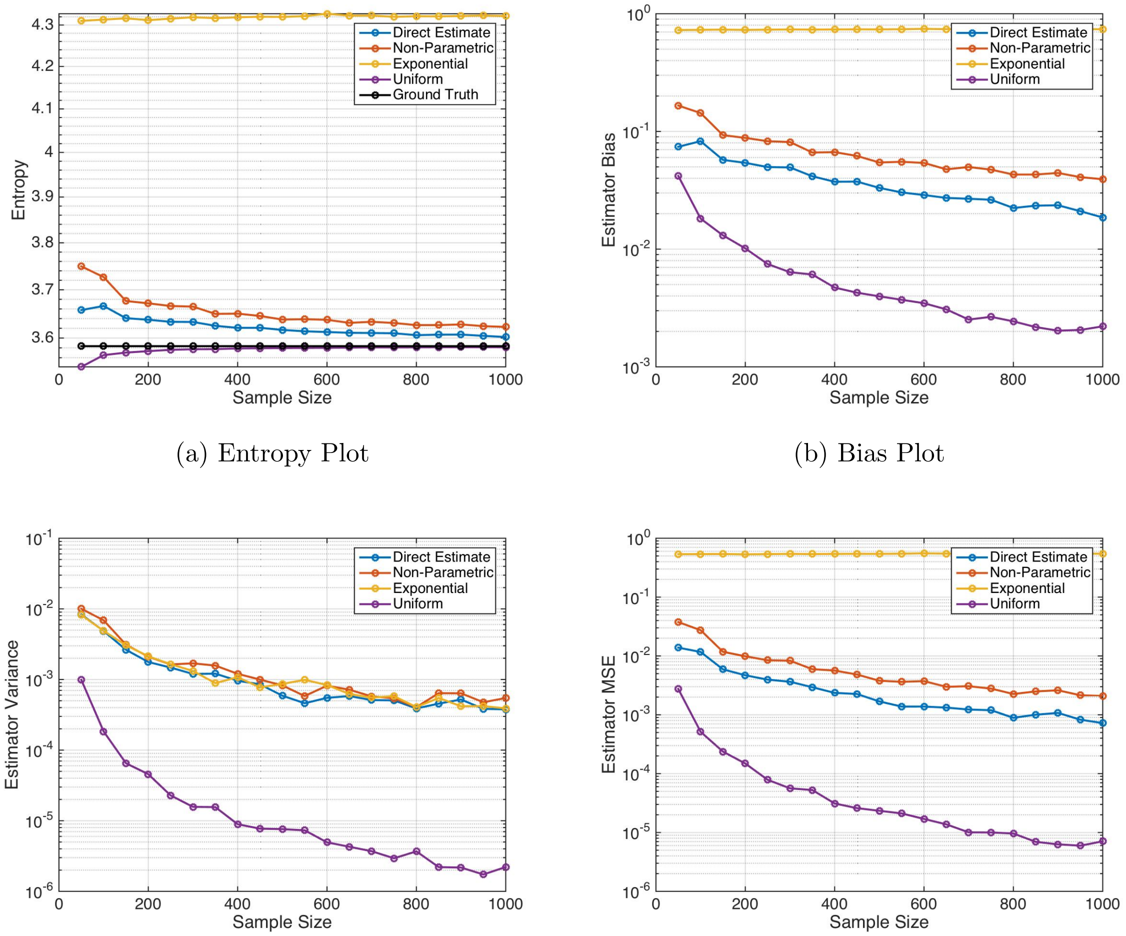
Figure 2.5: Mean length $L_\gamma(T)$ of minimum spanning trees constructed on $\mathbf{X}$.

Because of this, any bias resulting from error in estimating the bias constant $\beta_{d,\gamma}$, will compound the bias when applied to normally distributed data, but will reduce the bias when applied to uniformly distributed data. This illustrates that despite our desire to view these estimates as independent of the underlying distribution, their finite sample properties still depend to some degree on the underlying distributions, and it is only in the asymptotic regime can we create estimates that are truly independent of the parameters of the underlying distribution.

To better understand the effectiveness of this approach relative to some other parametric and non-parametric estimators, we have plotted the entropy, bias, variance, and MSE as a function of sample size for several different estimators. In addition to the previously described direct estimation, we test two parametric plug-in estimators, one which correctly assumes that the data is uniformly distributed and another that incorrectly assumes that the data belongs to the exponential family of distributions. The parametric model operates on the assumption that the data is uniformly distributed across a $[a, b]^d$ hypercube and identifies $a$ and $b$ as the minimum and maximum respectively of the points found in the data, then calculates the entropy as $d \log(\hat{b} - \hat{a})$. The exponential model forms a maximum likelihood estimate of the exponential model and plugs the model into the analytical solution in [90].

To run this experiment, we vary the sample size incrementally across $N = 50$, 100, 150,..., 1000. For each sample size, we run a 100 iteration Monte-Carlo simulation and compare the estimates to the theoretical ground truth to assess each estimate in terms of bias, variance, and mean squared error (MSE). This experiment is run in 2 and 10 dimensions, and the results are displayed in Figures 2.6 and 2.7 respectively. In both simulations we find, not surprisingly, that the parametric model which correctly assumes that the data is distributed uniformly across the d-dimensional hypercube performs the best in terms of bias, variance, and MSE. In this experiment the direct estimate outperforms the exponential parametric and non-parametric plug-in estimators. While the non-parametric estimator vastly outperforms the exponential model in the 2-dimensional case, it is outperformed in the 10-dimensional simulation for low sample sizes. Thus if our data is sparse enough, an obviously flawed parametric model can outperform a non-parametric model. Since the finite-sample bias of the flawed parametric model works against its asymptotic bias its performance declines with additional samples, while the other three approaches all appear to asymptote towards the ground truth though their rates of convergence vary.

Prior to this point we have focused on estimating entropy. Now, let us consider that we have two sets of data $\mathbf{X}_0 = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_f})$ and $\mathbf{X}_1 = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_g})$, sampled from distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ respectively. Along with the Reńyi entropy of the individual data sets, MSTs can be used to directly estimate the both the $\alpha$-divergence and the $\alpha$-Jensen divergence between the two underlying distributions. The approach to estimate the $\alpha$-divergence relies on transforming the coordinates of the feature vectors in order to "flatten" the reference density $f_0(\mathbf{x})$. Since $f_0(\mathbf{x})$ is uniform in the



(a) Entropy Plot

(b) Bias Plot

(c) Variance Plot

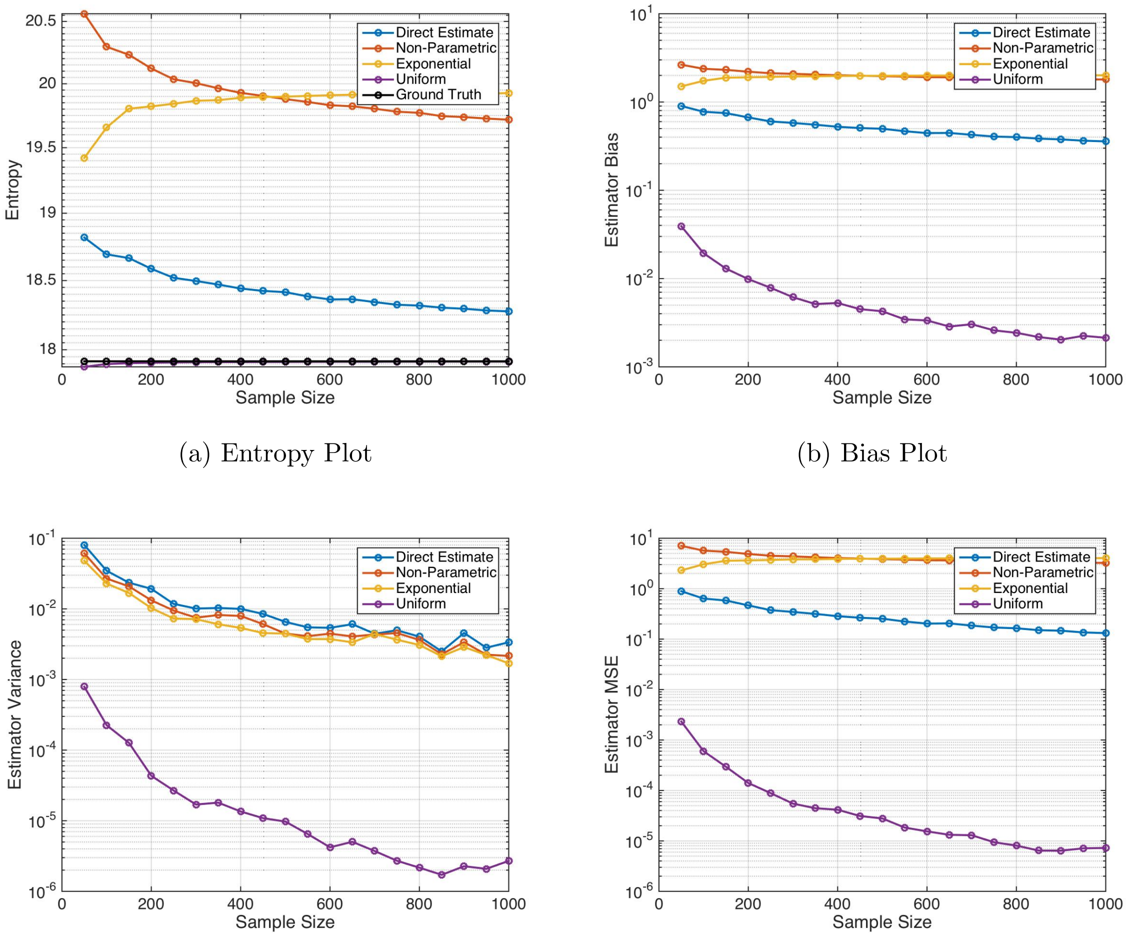
(d) MSE Plot

Figure 2.6: Plots illustrating the convergence properties of direct and plug-in estimates of the entropy for points distributed uniformly within square $[0, 3]^2$.

new coordinate space, the $\alpha$-divergence between $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$, can be estimated by measuring the $\alpha$-entropy of $f_1(\mathbf{x})$ in the transformed coordinate space [55]. The $\alpha$-Jensen divergence can be estimated by applying the following formula

$$\hat{J}(p_0, f_0, f_1) = \hat{H}_\alpha(\mathbf{X}_0 \bigcup \mathbf{X}_1) - p_0 \hat{H}_\alpha(\mathbf{X}_0) - (1 - p_0)\hat{H}_\alpha(\mathbf{X}_1) \tag{2.25}$$

where $p_0 = n_f/(n_f + n_g)$ and $\hat{H}_\alpha(\mathbf{X}_0 \bigcup \mathbf{X}_1)$ is an estimate of the joint $\alpha$-entropy



(a) Entropy Plot

(b) Bias Plot

(c) Variance Plot

(d) MSE Plot

Figure 2.7: Plots illustrating the convergence properties of direct and plug-in estimates of the entropy for points distributed uniformly within a d-dimensional hypercube $[0, 3]^{10}$.

between $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$.

### 2.2.4.2 Nearest Neighbor Graphs

Nearest neighbor graphs, much like the MST's discussed in the previous section, can be used to directly estimate a number of information-theoretic quantities. As minimal graphs, k-nearest neighbor (k-NN) graphs have many of the same properties as MSTs. For example, the length function of k-NN graphs can be used to directly estimate the Rényi entropy in the same manner as was previously described for MSTs [18, 88]. However they also offer a number of unique properties that can be exploited to develop more robust and diverse estimators.

The use of nearest neighbor graphs as a non-parametric classification algorithm was first proposed by Fix and Hodges[35], and later popularized by the work of Cover and Hart [25]. The its asymptotic properties have been very well defined. In particular, we know that the asymptotic error rate of any k-NN classifier will never exceed two times the Bayes risk regardless of the underlying distributions. To illustrate this, consider a single point $\mathbf{x}_i \in \mathbf{X}$ and its nearest neighbor $\mathbf{x}_j$. In this scenario, there are two possible errors that the classifier can make: A) the classifier assigns $\hat{y}_i$ a value of 0, when $y_i = 1$ and B) the classifier assigns $\hat{y}_i$ a value of 1, when $y_i = 0$. For a 1-NN classifier it is easy to see that the probability of the first error is equal to $P[y_i = 1 \cap y_j = 0]$ and the probability of the second error is equal to $P[y_i = 0 \cap y_j = 1]$. Since

$$P[y = 0] = \eta_0(\mathbf{x}) = \frac{pf_0(\mathbf{x})}{pf_0(\mathbf{x}) + (1-p)f_1(\mathbf{x})} \tag{2.26}$$

and

$$P[y = 1] = \eta_1(\mathbf{x}) = \frac{(1-p)f_1(\mathbf{x})}{pf_0(\mathbf{x}) + (1-p)f_1(\mathbf{x})} \tag{2.27}$$

where $\eta_0(\mathbf{x})$ and $\eta_1(\mathbf{x})$ are the posterior distributions for each class, the probability of misclassifying $\mathbf{x}_i$ can be simplified to

$$\epsilon(\mathbf{x}_i) = \eta_1(\mathbf{x}_i)\eta_0(\mathbf{x}_j) + \eta_0(\mathbf{x}_i)\eta_1(\mathbf{x}_j). \tag{2.28}$$

Now, since $\mathbf{x}_j$ is the nearest neighbor to $\mathbf{x}_i$, we can represent it as $\mathbf{x}_j = \mathbf{x}_i + \delta$. We can also observe that $\delta \to 0$ as $N \to \infty$. Substituting this into (2.28) yields

$$\lim_{N\to\infty} \epsilon(\mathbf{x}_i) = \lim_{\delta\to 0} \eta_1(\mathbf{x}_i)\eta_0(\mathbf{x}_i+\delta) + \eta_0(\mathbf{x}_i)\eta_1(\mathbf{x}_i+\delta) = 2\eta_0(\mathbf{x}_i)\eta_1(\mathbf{x}_i). \tag{2.29}$$

Thus by integrating over all possible $\mathbf{x}_i$, the asymptotic error rate for the 1-NN classifier is found to be

$$\epsilon_1^{NN} = E[2\eta_0(\mathbf{x})\eta_1(\mathbf{x})] = \int \epsilon(\mathbf{x})Pr(\mathbf{x}) = \int 2\eta_0(\mathbf{x})\eta_1(\mathbf{x})(pf_0(\mathbf{x}) + (1-p)f_1(\mathbf{x}))d\mathbf{x}. \tag{2.30}$$

For comparison, the Bayes risk ($R^*$) conditioned on a single point $\mathbf{x}$ can be defined as

$$r^* = \min[\eta_0, \eta_1] = \min[\eta_0, 1-\eta_0]. \tag{2.31}$$

By exploiting the symmetry of $r^*$ w.r.t. $\eta_0$ we find that

$$r^*(\mathbf{x})(1-r^*(\mathbf{x})) = \eta_0(\mathbf{x})(1-\eta_0(\mathbf{x})). \tag{2.32}$$

Using this we can rewrite (2.30) as

$$\begin{aligned} \epsilon_1^{NN} &= E[2r^*(\mathbf{x})(1-r^*(\mathbf{x}))] \\ &= 2(E[r^*] - E[(r^*)^2]) \end{aligned} \tag{2.33}$$

and since $E[r^*] = R^*$, this can be simplified to

$$\begin{aligned} &= 2R^* - (var(r) + R^{*2}) \\ &= 2R^*(1-R^*) - var(r^*). \end{aligned} \tag{2.34}$$

Therefore the asymptotic error rate (AER) for the 1-NN classifier can be bounded by

$$R^* \leq \epsilon_1^{NN} \leq 2R^*(1-R^*) \leq 2R^* \tag{2.35}$$

These guarantees on the asymptotic performance of the 1-NN classifier that Cover and Hart introduced [25] were instrumental in making the k-NN algorithm one of the most widely used classification algorithms in existence. Now if we consider the more general case, when $k \neq 1$, a similar approach to that employed for the $k = 1$ case can be used to show that the AER for a traditional k-NN classifier can be expressed as

$$\begin{aligned}\epsilon_k^{NN} &= E\Bigg[\sum_{i=\lceil \frac{k}{2}\rceil}^{k} \binom{k}{i}\Big(\eta^i(\mathbf{x})(1-\eta(\mathbf{x}))^{k-i+1} + \eta^{k-i+1}(\mathbf{x})(1-\eta(\mathbf{x}))^i\Big)\Bigg] \\ &= \int \sum_{i=\lceil \frac{k}{2}\rceil}^{k} \binom{k}{i}\Big(\eta^i(\mathbf{x})(1-\eta(\mathbf{x}))^{k-i+1} + \eta^{k-i+1}(\mathbf{x})(1-\eta(\mathbf{x}))^i\Big) \\ &\qquad (pf_0(\mathbf{x}) + (1-p)f_1(\mathbf{x}))d\mathbf{x}.\end{aligned} \tag{2.36}$$

Perhaps the most important attribute of the k-NN classifier, is it was the first classifier proven to be universally consistent with the Bayes risk, in other words

$$\lim_{k,N\to\infty,\frac{k}{N}\to 0} \epsilon_k^{NN} = R^*. \tag{2.37}$$

Stone showed that when $n$ and $k$ approach infinity as $k/n \to 0$, the expected risk of a k-NN classifier approaches the Bayes risk [107]. While this is a powerful property as it guarantees the optimality of the $k$-nearest neighbor classifier given enough data, it tells us little about what performance can be expected in the finite sample regime. In fact, Devroye showed [30] that for any integer $N$ and classification rule $g_N$, there exists a distribution of $(\mathbf{X}, y)$ with Bayes risk $R^* = 0$ in which

$$E[\epsilon\Big(g_N(\mathbf{X})\Big)] \geq \frac{1}{2} - \delta \tag{2.38}$$

where $\delta > 0$ represents an arbitrarily small constant. In other words, even though a classifier may be proven to perform optimally in the asymptotic regime, it's convergence for a particular distribution may be arbitrarily slow and we cannot provide any distribution-free guarantees on performance in the finite sample regime. The strong asymptotic performance guarantees of the nearest neighbor classifier combined with the observably poor performance in many finite sample problems has led to a large amount of interest in identifying distance metrics [103, 113, 114], decision rules [29, 33, 52, 84, 108], ensemble methods [8, 32] and editing techniques [30, 34, 46, 58, 66, 109, 117] that can achieve superior finite-sample performance. In some cases, particularly within the editing techniques and decision rules [52, 117], these variations will affect the asymptotic performance of the classifier, however in general this work is targeted at improving the rate at which the k-NN classifier convergences to it's asymptotic performance. In addition to its primary benefit, improving the finite-sample performance of these classifiers can also improve their viability as non-parametric estimators of unknown information-theoretic quantities, such as in the scenario in which the nearest neighbor error rate is used to bound the Bayes risk [25, 40, 41, 42]. The importance of this will become clearer in Section 5.1, when we introduce the concept of directly estimable basis functions.

## 2.3 Information Theory In Machine Learning

Machine learning is important to a number of disciplines ranging from health and wellness to finance. At its core machine learning is about learning from data. Information theory plays a key role in a number of machine learning problems including feature selection [123], classification [85, 93], and clustering [55]. For the purposes of this dissertation, we will be primarily talking about supervised learning, which reflects the scenario in which we have access to some sample data for the variable that we are trying to predict (outcome variable). Figure 2.8 depicts the basic components of a traditional supervised learning system. Given some raw data, we first attempt to extract features that we believe contain important information about the variable of interest. We then use machine learning to attempt and find some hypothesized mapping $h(\mathbf{x})$ between the feature data $\mathbf{x}$ and the outcome variable $y$. This mapping $h(\mathbf{x})$ is called a classifier, and we consider any occurrence in which $h(\mathbf{x}) \neq y$ a classi-

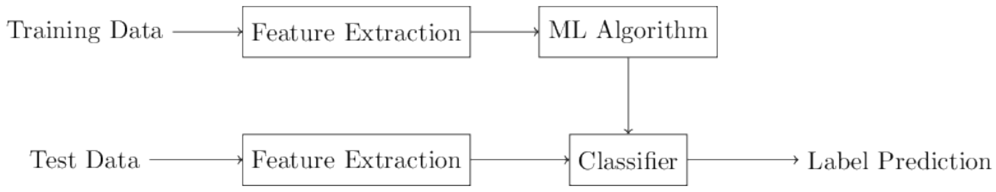


Figure 2.8: Supervised learning classifier block diagram.

fication error. We thus evaluate the viability of a classifier in terms of the frequency of error

$$\epsilon[h] = P[h(\mathbf{x}) \neq y]. \tag{2.39}$$

Since no approach in machine learning can be universally optimal [121], our ability to construct successful classification models in a particular domain is dependent on our ability to accurately evaluate those models within that domain. Let us consider two classifiers $h_1$ and $h_2$. It is easy then to see that if $\epsilon[h_1(\mathbf{x})] < \epsilon[h_2(\mathbf{x})]$ for some dataset $(x, y)$, then $h_1$ is the superior classifier to $h_2$ in this domain, however this tells us little about the performance of $h_1$ relative to the unlimited set of potential classifiers that could be applied to this problem. As a results $h_1$ may look good against the finite set of alternatives that are considered, while remaining a highly suboptimal choice for this particular problem. This illustrates the need for accurate measures of optimal performance such as the Bayes risk, which will be discussed in the following Section.

### *2.3.1 Bayes Error Rate*

Define the conditional density functions as $f_0(\mathbf{x}) = f_x(\mathbf{x}|y = 0)$ and $f_1(\mathbf{x}) = f_x(\mathbf{x}|y = 1)$ and the prior probabilities are given by $p_0 = P[y = 0]$ and $p_1 = P[y = 1]$, respectively. Given complete knowledge of the underlying distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ the optimal classifier will assign class $\hat{y}(\mathbf{x}_i) = 0$ only when the posterior distribution for class 0 $\eta_0 = \frac{p_0 f_0(\mathbf{x})}{p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})} \geq 0.5$. This classifier is referred to as the

Bayes classifier, and the error rate of this classifier is the Bayes risk and can be described by:

$$R^* = \int_{pf_0(\mathbf{x}) \leq qf_1(\mathbf{x})} pf_0(\mathbf{x})d\mathbf{x} \;\; + \int_{qf_1(\mathbf{x}) \leq pf_0(\mathbf{x})} qf_1(\mathbf{x})d\mathbf{x}. \tag{2.40}$$

The Bayes risk is useful not only for assessing the performance of specific classifiers, but also as a criteria in data modification algorithms [22] and in the development of the classification algorithms themselves [45].

While the Bayes risk is a useful benchmark, we are never given complete knowledge of the underlying distributions in practical scenarios, and the Bayes risk cannot be calculated, only estimated. This is a variation of the problem described in Section 2.2 and a detailed overview of approaches to Bayes risk estimation can be found in [49].

While estimation of the Bayes risk is useful in its own right, the Bayes risk is a non-convex function of the posterior distributions and as a result it is difficult to build algorithms that minimize the risk directly. As a result, there has been significant research interest in forming convex bounds on the Bayes risk that can be represented using estimable measures of distance between probability functions [5, 23, 50, 63].

The total variation (TV) distance is closely related to the Bayes error rate [63], as illustrated in Figure 2.9. A number of bounds exist in the literature relating the KL divergence and the TV distance. The well-known Pinsker inequality provides a bound on the total variation distance in terms of the KL divergence [27]. Sharpened inequalities that bound the KL divergence in terms of a polynomial function of the TV distance were derived in [69]. One drawback of the Pinsker-type inequalities is that they become uninformative for completely separable distributions where the KL divergence goes to $\infty$ (since the TV distance is upper bounded). Vajda's refinement to these bounds addresses this issue [112].

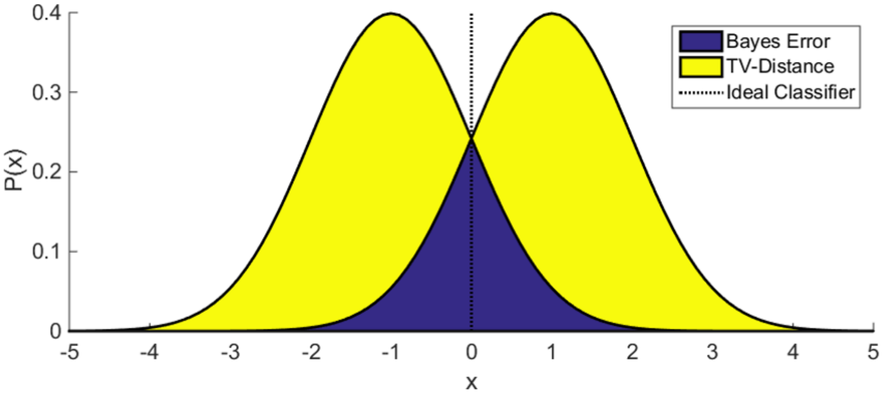


Figure 2.9: Relationship between BER & TV-distance.

For classification problems, the well-known upper bound on the probability of error based on the Chernoff $\alpha$-divergence has been used in a number of statistical learning applications [23]. The tightest bound is determined by finding the value of $\alpha$ that minimizes the upper bound. The Bhattacharya (BC) divergence, a special case of the Chernoff $\alpha$-divergence for $\alpha = \frac{1}{2}$, upper and lower bounds the BER [17, 63]. The BC bounds are often used as motivation for algorithms in the statistical learning literature because these bounds have closed form expressions for many commonly used distributions. In addition, for small differences between the two classes, it has been shown that, for the class of Chernoff $\alpha$-divergence measures, $\alpha = \frac{1}{2}$ (the BC divergence) results in the tightest upper bound on the probability of error [56].

Beyond the bounds on the BER based on the divergence measures, a number of other bounds exist based on different functionals of the distributions. In [50], the authors derive a new functional based on a Gaussian-Weighted sinusoid that yields tighter bounds on the BER than other popular approaches. Avi-Itzhak proposes arbitrarily tight bounds on the BER in [5].

### *2.3.2 Domain Adaptation*

Traditional machine learning problems such as those described previously in this Section rely on the assumption that the data used to train the algorithm and the data it will be tested on are generated by the same set of underlying density functions. The field of transfer learning is based upon the idea of generalizing knowledge found in one domain to make predictions in a slightly adjacent domain. Pan defines domain adaptation as a subproblem in this field which occurs when no labeled data exists in the domain of interest.

In domain adaptation problems, we are attempting to train our system for a domain adjacent to the one it is being designed for. As a result, we expect that the density functions in the training set or source domain $f_{S,0}(\mathbf{x})$ and $f_{S,1}(\mathbf{x})$ will be slightly different from those in the test set or target domain $f_{T,0}(\mathbf{x})$ and $f_{T,1}(\mathbf{x})$. This approach particularly useful for problems in which labeled data in the domain of interest is either unavailable or expensive to acquire.

In addition to work on bounding the Bayes error rate, recently there have been a number of attempts to bound the error rate in cross-domain learning challenges (see Section 2.3.2. In [11, 12], Ben-David *et al.* relate the expected error on the test data to the expected error on the training data, for a given classifier $h$ as

$$\epsilon_T(h) \leq \epsilon_S(h) + D_{TV}(f_S, f_T) + min\Big[E_{f_S}[|\theta_S(\mathbf{x}) - \theta_T(\mathbf{x})|], E_{f_T}[|\theta_S(\mathbf{x}) - \theta_T(\mathbf{x})|]\Big] \quad (2.41)$$

where $\theta_S(\mathbf{x})$ and $\theta_T(\mathbf{x})$ represent oracle labeling functions for the source and target domain respectively. Since our knowledge of $\theta_T(\mathbf{x})$ is limited, particularly in the scenario where no labeled data is available in the target domain, it is not uncommon in domain adaptation problems to make the assumption that $\theta_S(\mathbf{x}) = \theta_T(\mathbf{x})$. This is known as the covariate shift assumption, and we can use it to simplify (2.41)

$$\epsilon_T(h) \leq \epsilon_S(h) + D_{TV}(f_S, f_T). \quad (2.42)$$

The ability to bound the target domain error by the sum of the source domain error and a measure of the divergence between the underlying distributions of the two domains is a profoundly useful tool which we will study in greater detail in Chapter 3.3 In [11, 12], Ben-David *et al.* showed that the error in the target domain for a given hypothesis can be bounded by its error in the source domain the expected error on the test data to the expected error on the training data, for the case when no labeled test data is available. In [20], the authors derive new bounds for the case where a small subset of labeled data from the test distribution is available. In [79], Mansour *et al.* generalize these bounds to the regression problem. In [80], the authors present a new theoretical analysis of the multi-source domain adaptation problem based on the $\alpha$-divergence.

# Chapter 3

# DIRECTLY ESTIMABLE DIVERGENCE MEASURE

Information-theoretic quantities have played a key role in the development of bounds and algorithms in machine learning. However much of this work is reliant on parametric assumptions, and not robust to the varying types of data that may be encountered. In this section, we will introduce a divergence measure that can be directly estimated from data without prior knowledge of the underlying probability density functions. We investigate some of the properties of this divergence measure and show how it can be used to form more reliable bounds on the BER for binary classification problems.

In Section 3.1 we introduce a non-parametric divergence measure which can be directly estimated from data using minimum spanning trees. Section 3.2 shows how this divergence measure can be used to bound the Bayes risk for binary classification problems. In Section 3.3 we use this divergence measure to form an upper bound on the Bayes risk for domain adaptation problems.

## 3.1 A Nonparametric Divergence Measure

For probability density functions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ with prior probabilities $p_0 \in (0,1)$ and $p_1 = 1-p_0$ respectively in domain $\mathbb{R}^d$, we can define the following divergence function:

$$D_{p_0}(f_0, f_1) = \frac{1}{4p_0p_1}\left[\int \frac{(p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x}))^2}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})}d\mathbf{x} - (p_0 - p_1)^2\right]. \tag{3.1}$$

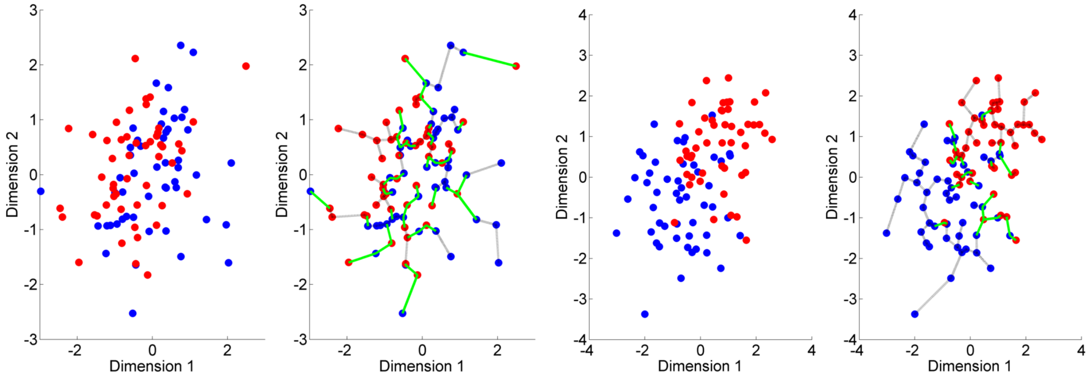
The divergence in (3.1), first introduced in [14], has the remarkable property that it can be estimated directly without estimation or plug-in of the densities $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ based on an extension of the Friedman-Rafsky (FR) multi-variate two sample test statistic [38]. Let us consider sample realizations from $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$, denoted by $\mathbf{X}_0 \in \mathbb{R}^{N_0 \times d}$, $\mathbf{X}_1 \in \mathbb{R}^{N_1 \times d}$. The FR test statistic, $\mathcal{C}(\mathbf{X}_0, \mathbf{X}_1)$, is constructed by first generating a Euclidean minimal spanning tree (MST) on the concatenated data set, $\mathbf{X}_0 \cup \mathbf{X}_1$, and then counting the number of edges connecting a data point from $f_0(\mathbf{x})$ to a data point from $f_1(\mathbf{x})$. The test assumes a unique MST for $\mathbf{X}_0 \cup \mathbf{X}_1$ - therefore all inter-point distances between data points must be distinct. However, this assumption is not restrictive since the MST is unique with probability one when $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ are Lebesgue continuous densities. In Theorem 1, we present an estimator that relies on the FR test statistic and asymptotically converges to $D_{p_0}(f_0, f_1)$. Note that this theorem combines the results of Theorem 1 and equations (3) and (4) in [14]. The proof of this theorem can be found in Appendix A.

**Theorem 1** *As $N_0 \to \infty$ and $N_1 \to \infty$ in a linked manner such that $\frac{N_0}{N_0+N_1} \to p_0$ and $\frac{N_1}{N_0+N_1} \to p_1$,*

$$1 - \mathcal{C}(\mathbf{X}_0, \mathbf{X}_1)\frac{N_0 + N_1}{2N_0N_1} \to D_{p_0}(f_0, f_1).$$

*almost surely.*

In Figure 3.1a and 3.1b we show two numerical examples in order to visualize the results of Theorem 1 - we plot samples from two distributions, $\mathbf{X}_0 \sim f_0(\mathbf{x})$ and $\mathbf{X}_1 \sim f_1(\mathbf{x})$, and evaluate the value of $\mathcal{C}(\mathbf{X}_0, \mathbf{X}_1)$. In Figure 3.1a, both data sets are drawn from the same distribution, $f_0(\mathbf{x}) = f_1(\mathbf{x}) = \mathcal{N}([0,0]^{\mathrm{T}}, \mathbf{I})$. In Figure 3.1b, we plot data drawn from $f_0(\mathbf{x}) = \mathcal{N}([-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}]^{\mathrm{T}}, \mathbf{I})$ and $f_1(\mathbf{x}) = \mathcal{N}([\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]^{\mathrm{T}}, \mathbf{I})$. $\mathbf{I}$ is the identity matrix. For both data sets, an equal number of points are drawn, therefore $N_0 = N_1 = N$ and $p = p_0 = p_1 = \frac{1}{2}$. The dotted line in each figure represents the Euclidean MST associated with $\mathbf{X}_0 \cup \mathbf{X}_1$. The green lines represent the edges of the MST connecting points from $f_0(\mathbf{x})$ to points from $f_1(\mathbf{x})$, $\mathcal{C}(\mathbf{X}_0, \mathbf{X}_1)$. We can use this to estimate $D_p(f_0, f_1)$ using the results of Theorem 1. It is clear from



(a) $f_0(\mathbf{x}) = f_1(\mathbf{x})$

(b) $f_0(\mathbf{x}) \neq f_1(\mathbf{x})$

Figure 3.1: Estimation of the $D_p$-divergence for the case when (a) $f = g$ and (b) $f \neq g$.

the figures that this value is much smaller for overlapping distributions (Figure 3.1a) than for separable distributions (Figure 3.1b). Indeed, as Theorem 1 suggests, in the limit, this statistic converges to the integral used in the divergence measure in (3.1).

In the ensuing sections we outline some important properties of this divergence measure and develop new bounds for classification using this distance function between distributions.

### *3.1.1 Properties Of The $D_p$-Divergence*

The divergence measure in (3.1) exhibits several properties that make it useful for statistical analysis. It is relatively straightforward to show that the following three properties are satisfied.

1. $0 \leq D_{p_0}(f_0, f_1) \leq 1$
2. $D_{p_0}(f_0, f_1) = 0 \iff f_0(\mathbf{x}) = f_1(\mathbf{x})$.
3. $D_{p_0}(f_0, f_1) = D_q(g, f)$.

The lower bound in the first property follows from the fact that when $f_0 = f_1$ and $p_0 = p_1$, the minimum value of $D_{p_0}$ is 0. To show that the divergence measure is upper bounded by 1, we first note that

$$\int \frac{(p_0 f_0(\mathbf{x}) - p_1 f_1(\mathbf{x}))^2}{p_0 f_0(\mathbf{x}) + q g(\mathbf{x})} d\mathbf{x} = 1 - 4 p_0 p_1 A_{p_0}(f_0, f_1), \tag{3.2}$$

where

$$A_p(f_0, f_1) = \int \frac{f_0(\mathbf{x}) f_1(\mathbf{x})}{p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})} d\mathbf{x}.$$

The function $A_p(f_0, f_1)$ attains its minimum value of 0, when $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ have no overlapping support (since $f_0(\mathbf{x}) > 0$ and $f_1(\mathbf{x}) > 0$ for all $\mathbf{x}$); therefore $D_{p_0} = \frac{1}{4p_0p_1}[1 - (p_0 - p_1)^2] = 1$. The second property is closely related to the first: the minimum value $D_{p_0} = 0$ only when $f_0 = f_1$ and $p_0 = p_1$. The third property follows from commutativity.

The divergence measure in (3.1) belongs to the class of $f$-divergences. Every $f$-divergence can be expressed as an average of the ratio of two distributions, weighted by some function $\phi(t)$: $D_\phi(f_0, f_1) = \int \phi(\frac{f(\mathbf{x})}{g(\mathbf{x})}) g(\mathbf{x}) d\mathbf{x}$. For $D_{p_0}(f_0, f_1)$, the corresponding function $\phi(t)$ is,

$$\phi(t) = \frac{1}{4p_0p_1} \left[ \frac{(p_0 t - p_1)^2}{p_0 t + p_1} - (2p_0 - 1)^2 \right]. \tag{3.3}$$

Furthermore, $\phi(t)$ is defined for all $t > 0$, is convex - $\phi''(t) = \frac{2p_0p_1}{(p_0t+p_1)^3} > 0$, and $\phi(1) = 0$. This is consistent with the requirements of the definition of an $f$-divergence [28]. Indeed, for the special case of $\alpha = \frac{1}{2}$, the divergence in (3.1) becomes the symmetric $\chi^2$ $f$-divergence in [21] and is similar to the Rukhin $f$-divergence in [99].

## 3.2 Bounds On Bayes Classification Error

In this section, we show how $D_p$ in (3.1) can be used to bound the Bayes error rate (BER) for binary classification. Further, we show that, under certain conditions, this bound is tighter than the well-known Bhattacharya bound commonly used in the machine learning literature and can be empirically estimated from data.

Before deriving the error bounds, for notation convenience, we introduce a slightly modified version of the divergence measure in (3.1),

$$\begin{aligned} u_{p_0}(f_0, f_1) &= 1 - 4p_0p_1 \int \frac{f_0(\mathbf{x})f_1(\mathbf{x})}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})} d\mathbf{x} \\ &= \int \frac{(pf(\mathbf{x}) - p_1f_1(\mathbf{x}))^2}{pf(\mathbf{x}) + p_1f_1(\mathbf{x})} d\mathbf{x}. \end{aligned} \tag{3.4}$$

It is easy to see that $D_{p_0} = \frac{u_{p_0}}{4p_0p_1} - \frac{(p_0-p_1)^2}{4p_0p_1}$ and when $p = q = 0.5$, $D_{p_0} = u_{p_0}$. While this function no longer satisfies $u_{p_0}(f_0, f_1) = 0$, for $f = g$, and therefore is no longer a valid divergence measure, it greatly simplifies the notation of the ensuing error bounds. As with $D_p$, w can estimate this quantity using the FR test statistic since, under the same conditions as those in Theorem 1,

$$1 - 2\frac{\mathcal{C}(\mathbf{X}_0, \mathbf{X}_1)}{N_0 + N_1} \to u_{p_0}(f_0, f_1). \tag{3.5}$$

Given a binary classification problem with binary labels $y \in \{0, 1\}$ and $\mathbf{x}$ drawn from $f_S(\mathbf{x})$, we denote the conditional distributions for both classes as $f_0(\mathbf{x}) = f_S(\mathbf{x}|y = 0)$ and $f_1(\mathbf{x}) = f_S(\mathbf{x}|y = 1)$. We draw samples from these distributions with probability $p_0$ and $p_1 = 1 - p_0$, respectively, and formulate two data matrices denoted by $\mathbf{X}_0 \in \mathbb{R}^{N_0 \times d}$ and $\mathbf{X}_1 \in \mathbb{R}^{N_1 \times d}$. The Bayes error rate associated with this problem is given in (2.40). In Theorem 2 below, we show that we can bound this error from above and below using the divergence measure introduced in the previous section. The proof of this theorem can be found in Appendix B.

**Theorem 2** *For two distributions, $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$, with prior probabilities $p$ and $q$ respectively, the Bayes error rate, $\epsilon^{\text{Bayes}}$, is bounded above and below as follows:*

$$\frac{1}{2} - \frac{1}{2}\sqrt{u_p(f_0, f_1)} \leq \epsilon^{\text{Bayes}} \leq \frac{1}{2} - \frac{1}{2}u_p(f_0, f_1).$$

Combining the results from Theorem 1 with the results of Theorem 2, we see that we can approximate the upper and lower bounds on the BER from the data matrices $\mathbf{X}_0$ and $\mathbf{X}_1$ as

$$\frac{1}{2} - \frac{1}{2}\sqrt{u_p(f_0, f_1)} \approx \frac{1}{2} - \frac{1}{2}\sqrt{1 - 2\frac{\mathcal{C}(\mathbf{X}_0, \mathbf{X}_1)}{N_0 + N_1}},$$

and

$$\frac{1}{2} - \frac{1}{2}u_p(f_0, f_1) \approx \frac{\mathcal{C}(\mathbf{X}_0, \mathbf{X}_1)}{N_0 + N_1}.$$

The derived bound is tight for the case $p_0 = p_1 = \frac{1}{2}$. For $f_0(\mathbf{x}) = f_1(\mathbf{x})$, the BER is 0.5. Under these conditions, $u_{p_0}(\mathbf{x}) = 0$, and both the upper and lower bound in Theorem 2 go to 0.5. For the case where $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ are completely separable, the BER is 0, $u_p(\mathbf{x}) = 1$, and both the upper and lower bound go to 0.

### *3.2.1 Relationship To The Chernoff Information Bound*

Here we compare the tightness of the bounds on the Bayes error rate based on $D_p$ to the bounds based on the Chernoff information function (CIF) [56], defined as

$$I_\alpha(f_0, f_1) = \int p^\alpha f_0^\alpha(\mathbf{x}) q^{1-\alpha} f_1^{1-\alpha}(\mathbf{x}) d\mathbf{x}.$$

In Theorem 3, we derive an important relationship between the affinity measure, $A_p(f_0, f_1)$, and a scaled version of the CIF. The proof of this theorem can be found in Appendix C.

**Theorem 3** *The affinity measure, $A_p(f_0, f_1)$, is a lower bound for a scaled version of the Chernoff information function:*

$$A_p(f_0, f_1) \leq \int f_0^q(\mathbf{x}) f_1^p(\mathbf{x}) d\mathbf{x}.$$

It is important to note that the second term in Theorem 3 is exactly equal to the CIF for $\alpha = p_0 = p_1 = 1/2$. For this special case, the Chernoff bound reduces to the Bhattacharyya (BC) bound, a widely-used bound on the Bayes error in machine learning that has been used to motivate and develop new algorithms [63, 100, 122]. The popularity of the BC bound is mainly due to the the fact that closed form expressions for the bound exist for many of the commonly used distributions. Let us define the Bhattacharya coefficient as:

$$BC(f_0, f_1) = 2\int \sqrt{pq f_0(\mathbf{x}) f_1(\mathbf{x})} d\mathbf{x}. \tag{3.6}$$

The well-known Bhattacharya bound on the BER is given by

$$\frac{1}{2} - \frac{1}{2}\sqrt{1 - BC^2(f, g)} \leq \epsilon^{\text{Bayes}} \leq \frac{1}{2} BC(f, g). \tag{3.7}$$

In Theorem 4 below, we show that, for equiprobable classes, the $D_p$ bound provides tighter upper and lower bounds on the BER when compared to the bound based on the BC coefficient under all separability conditions. The proof of this theorem can be found in Appendix D.

**Theorem 4** *For $p_0 = p_1 = \frac{1}{2}$, the $D_p$ upper and lower bounds on the Bayes error rate are tighter than the Bhattacharyya bounds:*

$$\begin{aligned}\frac{1}{2} - \frac{1}{2}\sqrt{1 - BC^2(f_0, f_1)} \leq & \frac{1}{2} - \frac{1}{2}\sqrt{u_{\frac{1}{2}}(f_0, f_1)} \\ & \leq \epsilon^{\text{Bayes}} \leq \frac{1}{2} - \frac{1}{2} u_{\frac{1}{2}}(f_0, f_1) \leq \frac{1}{2} BC(f_0, f_1).\end{aligned}$$

Using asymptotic analysis of the Chernoff exponent, for small differences between the two classes, it was shown that $\alpha = \frac{1}{2}$ results in the tightest bound on the probability of error - this corresponds to the bound in (3.7) [56]. Using a variant of this analysis, we derive a local representation of the CIF and relate it to the divergence measure proposed here. In particular, if we let

$$\begin{aligned} pf_0(\mathbf{x}) &= \frac{1}{2}(pf_0(\mathbf{x}) + p_1 f_1(\mathbf{x})) + \frac{1}{2}(pf_0(\mathbf{x}) - p_1 f_1(\mathbf{x})) \\ &= f_{\frac{1}{2}}(\mathbf{x})(1 + \frac{1}{2}\Delta_{\mathbf{x}}), \end{aligned}$$

where $f_{\frac{1}{2}}(\mathbf{x}) = \frac{1}{2}(pf_0(\mathbf{x}) + p_1 f_1(\mathbf{x}))$ and $\Delta_{\mathbf{x}} = (pf_0(\mathbf{x}) - p_1 f_1(\mathbf{x}))/f_{\frac{1}{2}}(\mathbf{x})$. Similarly,

$$p_1 f_1(\mathbf{x}) = f_{\frac{1}{2}}(\mathbf{x})(1 - \frac{1}{2}\Delta_{\mathbf{x}}).$$

As in [56], after a Taylor series expansion around $p^\alpha f_0^\alpha(\mathbf{x})$ and $q^{1-\alpha} f_1^{1-\alpha}(\mathbf{x})$, the Chernoff information function can be expressed as (see proof of Proposition 5 in [56]):

$$\begin{aligned} I_\alpha(f_0, f_1) &= \int f_{\frac{1}{2}}(\mathbf{x}) \Bigg[ 1 - (2\alpha - 1)\frac{\Delta_{\mathbf{x}}}{2} \\ &\qquad - \alpha(1-\alpha)\left(\frac{\Delta_{\mathbf{x}}}{2}\right)^2 + o(\Delta_{\mathbf{x}}^3) \Bigg] d\mathbf{x} \\ &= \int f_{\frac{1}{2}}(\mathbf{x}) d\mathbf{x} - (2\alpha - 1)\int f_{\frac{1}{2}}(\mathbf{x})\frac{\Delta_{\mathbf{x}}}{2} d\mathbf{x} \\ &\qquad - \alpha(1-\alpha)\int f_{\frac{1}{2}}(\mathbf{x})\left(\frac{\Delta_{\mathbf{x}}}{2}\right)^2 + o(\Delta^2) \\ &= \frac{1}{2} - (2\alpha - 1)(2p - 1)/2 \\ &\qquad - \frac{\alpha(1-\alpha)}{2}\int \frac{(pf_0(\mathbf{x}) - p_1 f_1(\mathbf{x}))^2}{pf_0(\mathbf{x}) + p_1 f_1(\mathbf{x})} d\mathbf{x} + o(\Delta^2) \\ &= (p + \alpha) - 2\alpha p - \frac{\alpha(1-\alpha)}{2} u_p(f_0, f_1) + o(\Delta^2). \end{aligned}$$

The local equivalence of $D_p$ and $I_\alpha$ is not surprising since all $f$-divergences are locally equivalent (they induce the same Riemann-Fisher metric on the manifold of densities) [28]. This useful property allows us to estimate the CIF for small differences between $f_0$ and $f_1$ using the MST procedure in Section 3.1. Further, we can express the BER in terms of the CIF:

$$\epsilon^{\text{Bayes}} \le I_\alpha \approx (p + \alpha) - 2\alpha p - \frac{\alpha(1-\alpha)}{2} u_p(f_0, f_1).$$

For $p_0 = p_1 = \frac{1}{2}$, this bound reduces to $\epsilon^{\text{Bayes}} \leq \frac{1}{2} - \frac{\alpha(1-\alpha)}{2} u_{\frac{1}{2}}(f_0, f_1)$. This is very similar to the upper bound in Theorem 2, differing only in the scale of the second term. Further, it is easy to see from this that the bound in Theorem 2 is tighter than the Chernoff bound since $\frac{\alpha(1-\alpha)}{2} < \frac{1}{2}$ for all $\alpha$. This is not surprising since, locally, $\alpha = 0.5$ yields the tightest bounds on the BER [56]. This corresponds to the BC bound in (3.7) and we have already shown that new bound is tighter than the BC bound in Theorem 4. This analysis further confirms that result.

In addition to providing tighter bounds on the BER, we can estimate the new $D_p$ bound without ever explicitly computing density estimates. We provide a numerical example for comparison. We consider two data samples from two classes, each of which comes from a normally distributed bivariate distribution with varying mean and spherical unit variance. The separation in means between the two class distributions is increased incrementally across 150 trials. The two distributions completely overlap initially, and are almost entirely separated by the final trial. In each trial we calculate the BER analytically using (2.40), as well as the upper and lower bounds introduced in Theorem 2. We calculate the bounds both analytically (through numerical integration) and empirically (using the results from Theorem 1). In order to demonstrate the tightness of this bound we also plot it against the upper and lower Bhattacharyya error bounds for Gaussian data (the closed form expression of the bound for Gaussian data is known) [63]. Figure 3.2 displays the true BER along with both error bounds as a function of the Euclidean separation between the means of two bivariate normal distributions of unit variance. We see in this plot that the proposed error bounds are noticeably tighter than the Bhattacharyya error bounds and are well correlated with the true BER. Although the analytically calculated $D_p$ bound never crosses the BC bound, the empirically estimated $D_p$ bound crosses the BC bound for small values of the mean separation. This is due to the variance of the estimator.

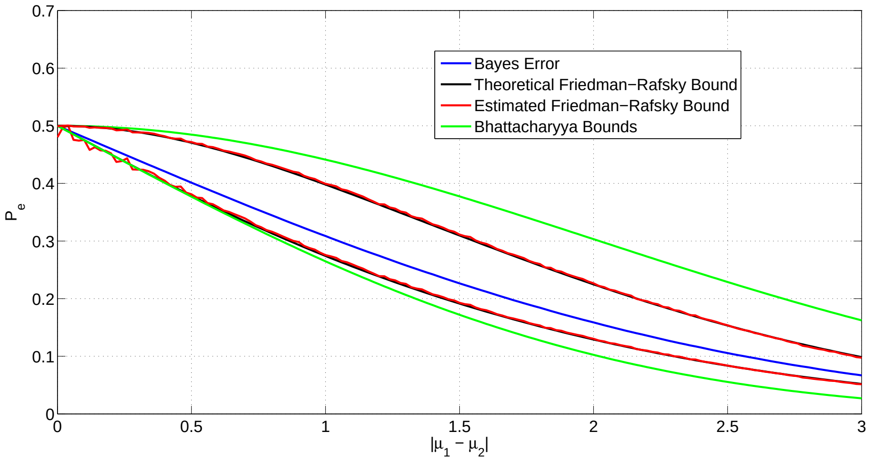


Figure 3.2: The $D_p$ and BC bounds on the Bayes error rate for a bivariate Gaussian example.

It is important to note that the estimator used here *asymptotically* converges to the $D_p$ divergence; however this result doesn't necessarily extend to finite data. In fact, for any fixed estimator, there exists a distribution for $\mathbf{X}$ and $y$ such that the error converges arbitrarily slowly [4].

## 3.3 Bounds On The Domain Adaptation Error

In this section, we consider a cross-domain binary classification problem and show how the $D_p$ distance can be used to bound the error rate in this setting also. Let us define data from two domains, the source (training) and the target (testing) domain and the corresponding labeling functions for each domain $y_{\mathrm{S}}(\mathbf{x}), y_{\mathrm{T}}(\mathbf{x}) \in \{0, 1\}$ that yields the true class label of a given data point $\mathbf{x}$. The source domain, denoted by the pair $(\mathbf{X}_{\mathrm{S}}, y_{\mathrm{S}})$, represents the data used to train the machine learning algorithm and the data $(\mathbf{X}_{\mathrm{T}}, y_{\mathrm{T}})$ represents the data the algorithm will encounter once deployed. Let us further define the conditional distributions $f_{\mathrm{S},0}(\mathbf{x}) = f_{\mathrm{S}}(\mathbf{x}|y_s(\mathbf{x}) = 0)$ and $f_{\mathrm{S},1}(\mathbf{x}) = f_{\mathrm{S}}(\mathbf{x}|y_s(\mathbf{x}) = 1)$. The rows of the source and target data are drawn from

$f_{\mathrm{S}}(\mathbf{x})$ and $f_{\mathrm{T}}(\mathbf{x})$. The risk, or the probability that the decision, $h$, disagrees with the true label is defined as

$$\epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) = \mathbf{E}_{f_{\mathrm{S}}(\mathbf{x})}[|h(\mathbf{x}) - y_{\mathrm{S}}|], \tag{3.8}$$

for the source data. It is similarly defined for the target data. In Theorem 5, we identify a relationship between the error rates on the source and target data. The proof of this theorem can be found in Appendix E.

**Theorem 5** *Given a hypothesis, $h$, the target error, $\epsilon_{\mathrm{T}}(h, y_{\mathrm{T}})$, can be bounded by the error on the source data, $\epsilon_{\mathrm{S}}(h, y_{\mathrm{S}})$, the difference between labels, and a distance measure between source and target distributions as follows:*

$$\begin{aligned}\epsilon_{\mathrm{T}}(h, y_{\mathrm{T}}) \leq & \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) + \mathbf{E}_{f_{\mathrm{S}}(\mathbf{x})}[|y_{\mathrm{S}} - y_{\mathrm{T}}|] \\ & +2\sqrt{u_{\frac{1}{2}}(f_{\mathrm{S}}, f_{\mathrm{T}})},\end{aligned} \tag{3.9}$$

*where $u_{\frac{1}{2}}(f_{\mathrm{S}}, f_{\mathrm{T}})$ assumes equiprobable data from the source and target distributions.*

The bound in Theorem 5 depends on three terms: the error on the source data, the expected difference in the labeling functions across the two domains, and a measure of the distance between source and target distributions ($D_p$ distance). We expect that the selected training algorithm will seek to minimize the first term; the second term characterizes the difference between labeling functions in the source and target domains; the third term is of particular interest to us - it provides a means of bounding the error on the *target* data as a function of the distance between source and target distributions.

In the *covariate shift* scenario, we assume that there exists no difference between labeling functions (e.g. $y_{\mathrm{S}}(\mathbf{x}) = y_{\mathrm{T}}(\mathbf{x})$) and only the distributions between the source and target data change [11]. Under this assumption, the bound in Theorem 5 reduces to

$$\epsilon_{\mathrm{T}}(h, y_{\mathrm{T}}) \leq \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) + 2\sqrt{u_{\frac{1}{2}}(f_{\mathrm{S}}, f_{\mathrm{T}})}. \tag{3.10}$$

Furthermore, if we assume that the decision rule $h$ attains the Bayes error rate, $\epsilon^{\mathrm{Bayes}}$, on the source domain, we can use the results from Theorem 2 to rewrite the bound in Theorem 5 using only the $D_p$ distance:

$$\epsilon_{\mathrm{T}} \leq \frac{1}{2} - \frac{1}{2}u_p(f_{\mathrm{S},0}, f_{\mathrm{S},1}) + 2\sqrt{u_{\frac{1}{2}}(f_{\mathrm{S}}, f_{\mathrm{T}})}. \tag{3.11}$$

If we denote the training data matrices by $\mathbf{X}_{\mathrm{S},0} \sim f_{\mathrm{S},0}$ and $\mathbf{X}_{\mathrm{S},1} \sim f_{\mathrm{S},1}$, then we can estimate this upper bound using the FR test statistic by

$$\frac{\mathcal{C}(\mathbf{X}_{\mathrm{S},0}, \mathbf{X}_{\mathrm{S},1})}{N_{\mathrm{S},0} + N_{\mathrm{S},1}} + 2\sqrt{1 - 2\frac{\mathcal{C}(\mathbf{X}_{\mathrm{S}}, \mathbf{X}_{\mathrm{T}})}{N_{\mathrm{S}} + N_{\mathrm{T}}}}. \tag{3.12}$$

The result shown in (3.12) represents an upper bound on the target domain error that can be computed without access to any labels in this domain. This bound provides interesting insight on the importance of invariant representations for classification. The target error is bounded by the sum of the affinity between class distributions in the source domain and the square root of the $D_p$-distance between domains. Because of the square root and the multiplicative factor, it is clear that the second term in (3.12) is weighted much more heavily. This stresses the importance of invariant representations in classification. In other words, the bound provides a means of quantifying the relative importance of selecting features that are invariant across domains versus features that provide good separation separation between classes in the source domain.

Chapter 4

# EXTENSIONS TO MULTI-CLASS AND REGRESSION PROBLEMS

The most stringent limitation of the error bounds presented in the previous section is its limitation to binary classification problems. In this chapter we show how these bounds can be extended to both multi-class classification problems and regression problems. These extensions will enable the application of the proposed methodology to a much greater range of problems. The extension to multi-class problems is described in Section 4.1, while the extension to regression problems is covered in Section 4.2.

## 4.1 Extending Bounds To Multi-Class Problems

In this section we extend the bounds introduced in Section 3.2 to multi-class problems. Section 4.1.1 uses a closed form extensions originally introduced in [74], while Section 4.1.2 utilizes a recursive extension introduced in [44]. In Section 4.1.3, we provide a brief comparison of the two methods.

### *4.1.1 Closed-Form Extension*

Consider an $M$-class problem with prior probabilities $p_1, ..., p_M$ and conditional class distributions $f_1(\mathbf{x}), ..., f_M(\mathbf{x})$ in hypothesis space $\mathbf{x}$. We first consider extending the bounds using the approach described in [74]. In this paper, the authors show that the BER in multi-class ($\mathcal{R}^M$) problems can be bounded by

$$\frac{2}{M}\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}(p_i+p_j)P_{eij} \leq \mathcal{R}^M \leq \sum_{i=1}^{M-1}\sum_{j=i+1}^{M}(p_i+p_j)P_{eij} \tag{4.1}$$

where $P_{eij}$ represents the pairwise Bayes risk of the 2-class subproblem of classifying between classes $i$ and $j$. Substituting in the upper and lower bounds on the Bayes

Risk defined in Section 3 yields

$$\begin{aligned}&\frac{2}{M}\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}(p_i+p_j)\left[\frac{1}{2}-\frac{1}{2}\sqrt{u_{\tilde{p}_i^{i,j}}(f_i(\mathbf{x}),f_j(\mathbf{x}))}\right]\\ &\leq \mathcal{R}^M \leq \sum_{i=1}^{M-1}\sum_{j=i+1}^{M}(p_i+p_j)\left[\frac{1}{2}-\frac{1}{2}u_{\tilde{p}_i^{i,j}}(f_i(\mathbf{x}),f_j(\mathbf{x}))\right]\end{aligned} \tag{4.2}$$

where $\tilde{p}_i^{i,j}$ represents the normalized prior probability for class $i$ defined by

$$\tilde{p}_i^{i,j} = \frac{p_i}{p_i+p_j}. \tag{4.3}$$

One limitation of this approach is that the upper bound becomes very loose when the overlap between class distributions is large. In fact, for completely overlapping distributions, the upper bound will converge to $(M-1)/2$ while the true BER converges to $(M-1)/M$. Section 4.1.2 will introduce an alternative that remedies this shortcoming.

### *4.1.2 Recursive Extension*

Next we consider an expression introduced by Garber and Djouadi that represent bounds on the Bayes risk in terms of the Bayes risk of the $M$ $(M-1)$-class subproblems created by removing different classes as

$$\begin{aligned}&\frac{M-1}{(M-2)M}\sum_{i=1}^{M}(1-p_i)\mathcal{R}_i^{M-1} \leq \mathcal{R}^M \leq\\ &\min_{\alpha\in\{0,1\}}\frac{1}{M-2\alpha}\sum_{i=1}^{M}(1-p_i)\mathcal{R}_i^{M-1}+\frac{1-\alpha}{M-2\alpha}.\end{aligned} \tag{4.4}$$

Here $\mathcal{R}_i^{M-1}$ represents the Bayes risk for the $(M-1)$-class subproblem created by removing class $i$ and $\alpha$ is an optimization parameter used to attain the tightest possible upper bound. By using these upper and lower bounds in a recursive manner we can attain upper and lower bounds for the multi-class BER in terms of the pairwise Bayes risks between conditional class distributions. As in the first extension, we can bound each pairwise BER in terms of the $D_p$-divergence using Theorem 2. For example let us consider the 3-class case, we can compute the upper bound as

$$\mathcal{R}^3 \leq \min_{\alpha \in \{0,1\}} \frac{1}{3-2\alpha} \sum_{i=1}^{3} (1-p_i)\mathcal{R}_i^2 + \frac{1-\alpha}{3-2\alpha}. \tag{4.5}$$

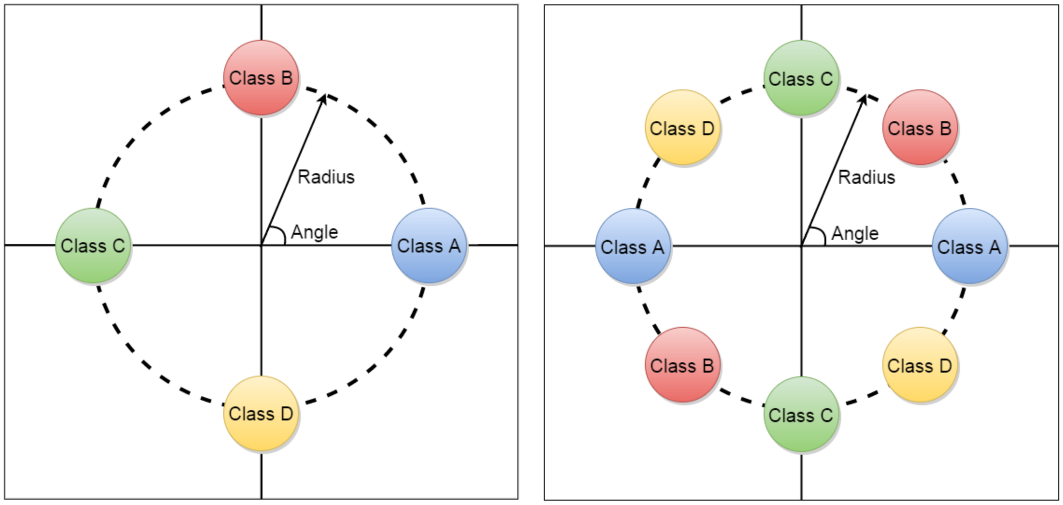
Substituting in the bounds expressed in Theorem 2 yields

$$\begin{aligned} \mathcal{R}^3 \leq \min_{\alpha \in \{0,1\}} \frac{1}{3-2\alpha} \Bigg\{ &(p_1+p_2)\left[\frac{1}{2} - \frac{1}{2} u_{\tilde{p}_1^{1,2}}(f_1,f_2)\right] \\ &+ (p_1+p_3)\left[\frac{1}{2} - \frac{1}{2} u_{\tilde{p}_1^{1,3}}(f_1,f_3)\right] \\ &+ (p_2+p_3)\left[\frac{1}{2} - \frac{1}{2} u_{\tilde{p}_2^{2,3}}(f_2,f_3)\right] \Bigg\} + \frac{1-\alpha}{3-2\alpha}. \end{aligned} \tag{4.6}$$

To better understand the role that $\alpha$ plays in this calculation, let us consider the two extreme cases in which the three class distributions are either completely overlapping or completely separable, and all class distributions have equal priors $p_1 = p_2 = p_3 = \frac{1}{3}$. In the first case, $\mathcal{R}_1^2 = \mathcal{R}_2^2 = \mathcal{R}_3^2 = \frac{1}{2}$ and $\alpha = 0$ yields the tightest bound of $\mathcal{R}^3 \leq \frac{2}{3}$ while $\alpha = 1$ yields the loosest bound of $\mathcal{R}^3 \leq 1$. In the second case $\mathcal{R}_1^2 = \mathcal{R}_2^2 = \mathcal{R}_3^2 = 0$, $\alpha = 0$ yields the loosest bound of $\mathcal{R}^3 \leq \frac{1}{3}$ while $\alpha = 1$ yields the tightest bound of $\mathcal{R}^3 \leq 0$. In general the value of $\alpha$ will depend on the total of the summation in (4.4). When this summation is greater than $(M-2)/2$ then $\alpha = 0$, otherwise $\alpha = 1$.

### *4.1.3 Comparison Of Bounds*

Because the two bounds are equivalent when $\alpha = 1$, Garber was able to show that for problems with equal priors the recursive extension is guaranteed to be at least as tight as the closed-form extension [44]. Extended proofs in Section A shows both the upper and lower recursive bounds will be at least as tight as the closed-form bounds regardless of priors. The price for this superiority comes in the increased computational burden. The computational burden of the closed-form bound can be approximated by $M(M-1)\gamma(n_c)/2$, where $\gamma(n_c)$ represents the number of computations required for a single pairwise risk function between classes containing $n_c$ samples.



(a) Unimodal Scenario

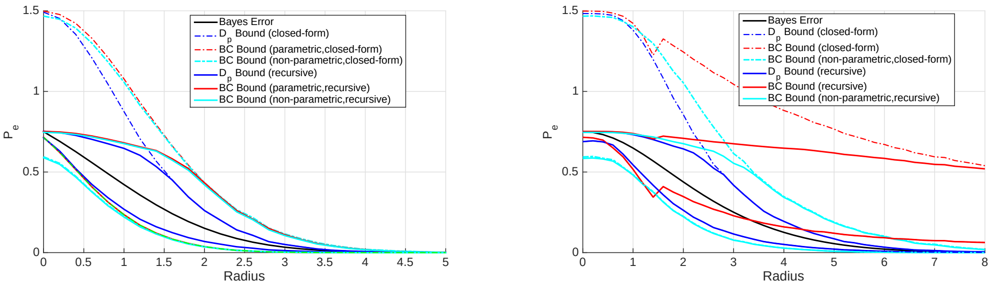
(b) Bimodal Scenario

Figure 4.1: Illustration of distribution placement for generating the synthetic data.

In addition to these computations, the recursive bound requires calculation of (4.4) for all $\sum_{i=3}^{M-1}\binom{M}{i}$ unique subproblems of 3 or more classes. While these additional computations are inconsequential for small $M$, their rapid growth w.r.t. $M$ makes this method infeasible for problems containing a large number of classes ($M > 30$).

To test the accuracy of the proposed bounds we consider the scenario in which four bivariate class distributions are equally spaced in a radial formation around the origin. We consider two scenarios. In the first scenario, each class distribution is represented by a single Gaussian distribution. In the second scenario, the class distributions from the first scenario are augmented by a second Gaussian distribution at a 180° rotation from the first. This second scenario is used to illustrate the behavior of the Bhattacharyya bound when the parametric assumption that each class can be modeled by a single Gaussian does not fit the actual data. The distribution placements used in each scenario are presented in Figure 4.1.

Throughout this experiment, each Gaussian is isotropic with unit covariance, and



(a) Unimodal Scenario

(b) Bimodal Scenario

Figure 4.2: True BER and error bounds for varying radii generated by each scenario of the synthetic data simulation.

mean determined by the angle and radius. The angle used to place each distribution is held constant (see Figure 4.1) while the radius is varied from zero, where the distributions in each scenario are completely overlapping, to eight where the distributions in each scenario contain almost no overlap with the neighboring distributions. The radius is varied in increments of 0.2, and each class distribution is represented by 1000 samples of data generated according to the parameters of the distribution. At each radius, we generate bounds on the Bayes error using both the recursive and closed-form extensions described in the previous Sections for the $D_p$ and BC bounds. In these calculations, the $D_p$-divergences are calculated using the approach described in Section 3. The Bhattacharyya distances are estimated in a parametric fashion by empirically estimating the mean and covariance matrices, then plugging the results into the explicit formula for multivariate normal distributions defined in [63], and in a non-parametric fashion by using a 2-dimensional histogram to estimate each underlying distribution and solving for the BC by integration. We obtain a ground truth value of the BER by integrating across the true underlying class distributions. To reduce the variance of the estimator we average our results across 25 Monte Carlo iterations. The resulting bounds are shown in Figure 4.2.

In Figure 4.2a, we see little difference between the parametric and non-parametric estimates of the Bhattacharyya bound, other than a slight negative bias that is most pronounced for tightly overlapping distributions. Figure 4.2b shows that while the non-parametric $D_p$ and $BC$ bounds remain largely unaffected by the addition of the second Gaussian for each class, the parametric bounds do not hold for radii exceeding  1.5 when the separation between modes is sufficient to violate the parametric assumption. In both scenarios, the $D_p$ bound provides a tighter bound on the BER. It should be noted that the benefits of the $D_p$ bound will only become more pronounced in high-dimensional spaces where accurate non-parametric density estimation is often infeasible [56].

## 4.2 Bounds On Regression Error

In the previous two Section we introduced bounds on performance in classification problems, where our data belongs to a discrete number of classes we would like to predict. In this Section, we will extend these ideas to regression problems where the response values we are trying to model are continuous. Consider the set of data $(\mathbf{x}_i, y_i)$ for $i \in [1 \ldots n]$, where each instance $\mathbf{x}_i \in \mathbf{R}^d$ is sampled from the underlying distribution $f(\mathbf{x})$ and each response $y_i \in \mathbf{R}^1$ is defined as:

$$y_i = \theta(\mathbf{x}_i) + \beta_i \tag{4.7}$$

where $\theta(\mathbf{x}_i)$ reflects an oracle function that provides the ground truth response for each exemplar and $\beta_i$ is zero mean white noise with variance $\sigma_i^2$. We assume that both the exemplars and the corresponding responses are independent identically distributed random variables. In this section, we will introduce a graph-theoretic measure that upper bounds the mean squared error (MSE) of the optimal estimator of the true response $\theta(\mathbf{x})$. Additionally, we will show that as the number of samples approaches infinity, the proposed function converges to the MSE. To begin, we define a spanning tree on $\mathbf{X}$ as a connected graph, $G = (\mathbf{X}, E)$, with vertex set $\mathbf{X}$, edges $E$, edge weights given by Euclidean distances between vertices, and no cycles. The length $\mathcal{L}(G)$ of the spanning tree is the sum of its edge lengths. The minimal spanning tree (MST), is defined as the spanning tree with the minimum length. Throughout the rest of this Section we will use $G$ to refer to the MST of $\mathbf{X}$.

Recall that the Friedman-Rafsky test statistic $\mathcal{C}(\mathbf{X_0}, \mathbf{X_1})$ represents a count of the number of edges in the minimum spanning tree on $\mathbf{X_0} \cup \mathbf{X_1}$ that connect points in $\mathbf{X_0}$ to points in $\mathbf{X_1}$. If we reformulate this problem such that $\mathbf{X} = \mathbf{X_0} \cup \mathbf{X_1} = [\mathbf{x_1}, ..., \mathbf{x_n}]$ and $\mathbf{y} = [y_i, ..., y_n]$ where

$$y_i = \begin{cases} 1 & : \mathbf{x}_i \in \mathbf{X_1} \\ 0 & : \mathbf{x}_i \in \mathbf{X_0} \end{cases} \tag{4.8}$$

then we can redefine $\mathcal{C}(\mathbf{X_0}, \mathbf{X_1})$ as

$$\mathcal{C}(\mathbf{X}, \mathbf{y}) = \sum_{e_{ij} \in E} |y_i - y_j|^2. \tag{4.9}$$

For every edge in the Euclidean MST of $\mathbf{X}$, this statistic calculates the squared difference in $y$ values between neighboring nodes in $G$. Using this notation, we can generalize this statistic beyond the classification problem and use it to measure the intrinsic difficulty of regression problems. To do so, we show that the expected value of this statistic asymptotically converges to the variance of $\beta_i$. Substitute (4.7) into

(4.9) we get:

$$\begin{aligned}
E\big[\mathcal{C}(\mathbf{X},\mathbf{y})\big] &= E\left[\sum_{e_{ij}\in E} |\theta(\mathbf{x}_i) + \beta_i - \theta(\mathbf{x}_j) - \beta_j|^2\right] \\
&= \sum_{e_{ij}\in E} E\big[(\theta(\mathbf{x}_i) - \theta(\mathbf{x}_j))^2\big] + \sum_{e_{ij}\in E} E\big[(\beta_i - \beta_j)^2\big] \\
&+ \sum_{e_{ij}\in E} 2E\big[(\beta_i - \beta_j)(\theta(\mathbf{x}_i) - \theta(\mathbf{x}_j))\big] \\
&= \sum_{e_{ij}\in E} E\big[(\theta(\mathbf{x}_i) - \theta(\mathbf{x}_j))^2\big] + \sum_{e_{ij}\in E} (\sigma_i^2 + \sigma_j^2).
\end{aligned} \tag{4.10}$$

The second term in (4.10) is the sum of the variances for each set of points connected by an edge in the MST. Without loss of generality, we can simplify this term by defining the degree of each vertex in the MST as $\rho_i$ for $i \in [1 \dots n]$. It is easy to see that this variance term in (4.10) can be rewritten as

$$\sum_{e_{ij}\in E} (\sigma_i^2 + \sigma_j^2) = \sum_{i=1}^{n} E[\rho_i]\sigma_i^2. \tag{4.11}$$

This corresponds to weighted sum of the variance of each label. A number of theoretical and empirical studies suggest that the degree distribution in random graphs follows a power law, $f_\rho(\rho) \sim (\frac{1}{\rho})^\gamma$ [78]. This is known to be a relatively low-entropy distribution, where the variable takes one of only a few values with high probability. As a result, the weights in the sum would not exhibit great variability, especially for large values of $\gamma$. If we now assume that the noise is identically distributed ($\sigma_i^2 = \sigma^2$ for $i \in [1 \dots n]$), this weighted sum simplifies to

$$\sum_{i=1}^{n} E[\rho_i]\sigma_i^2 = \sigma^2 \sum_{i=1}^{n} E[\rho_i] \tag{4.12}$$

It is known that the sum of the degrees across all nodes in a spanning tree is equal to two times the number of edges ($2||G||$), and that the number of edges is one less than the number of nodes, therefore

$$\sigma^2 \sum_{i=1}^{n} E[\rho_i] = \sigma^2(2||G||) = 2(n-1)\sigma^2. \tag{4.13}$$

Now let us define

$$\mathit{\Phi}(\mathbf{X},\mathbf{y}) = \frac{\mathcal{C}(\mathbf{X},\mathbf{y})}{2(n-1)}, \tag{4.14}$$

with mean

$$\begin{aligned} E[\mathit{\Phi}(\mathbf{X},\mathbf{y})] =& \frac{1}{2(n-1)} \sum_{e_{ij}\in E} E[(\theta(\mathbf{x}_i) - \theta(\mathbf{x}_j))^2] \\ &+ \frac{1}{2(n-1)} \sum_{i=1}^{n} E[\rho_i]\sigma_i^2. \end{aligned} \tag{4.15}$$

Under this assumption, it is easy to see that the first term in (4.15) will be greater than or equal to zero, therefore $\mathit{\Phi}$ upper bounds a weighted version of the MSE of the ideal estimator. Furthermore, if $\theta(\mathbf{x})$ is a Lipschitz continuous function with Lipschitz constant $K$, then $\|\theta(\mathbf{x}_i) - \theta(\mathbf{x}_j)\|^2 \leq K\|\mathbf{x}_i - \mathbf{x}_j\|^2$. Combining this relationship with the assumption that the noise is identically distributed in (4.12) and (4.13), we can form the following inequality:

$$E[\mathit{\Phi}(\mathbf{X},\mathbf{y})] \leq \frac{K}{2(n-1)} \sum_{e_{ij}\in E} \|\mathbf{x}_i - \mathbf{x}_j\|^2 + \sigma^2. \tag{4.16}$$

This not only bounds the MSE of the ideal estimator, we will go on to show that when $\mathbf{x}$ is drawn from a compactly supported distribution $f(\mathbf{x})$, it asymptotically converges to the true noise variance, $\sigma^2$, since the first term in the inequality goes to 0. If we define the average euclidean distance of the edges in $G$ as

$$\Gamma(G) = \frac{1}{n-1} \sum_{e_{ij}\in E} \|\mathbf{x}_i - \mathbf{x}_j\|^2, \tag{4.17}$$

then the first term in (4.16) is equal to $\frac{K}{2}\Gamma(G)$. This term manages the tightness of the MSE bound. This quantity has been studied in the literature. In fact, in [106] Steele showed that, with probability 1,

$$\lim_{n\to\infty} \Gamma(G) = \lim_{n\to\infty} c(d)(n-1)^{\frac{-2}{d}} \int_{\Re} f(\mathbf{x})^{\frac{d-2}{d}} d\mathbf{x}, \tag{4.18}$$

where $f(\mathbf{x})$ represents the compactly supported probability density function of the underlying distribution from which $\mathbf{X}$ is sampled, $c(d)$ denotes a strictly positive constant, and $d$ represents the data dimension. Note that this expression is only valid when $d > 2$.

We see from (4.18) that the bias of $\varPhi$ is affected by 1) the Lipschitz constant; 2) the number of samples; 3) the dimension; 4) the density of the input distribution. From this relationship it is easy to see that the bound becomes arbitrarily tight as $n \to \infty$. Furthermore, the rate of convergence for smaller $d$, smaller $K$, and more compact distributions, $f(\mathbf{x})$, increases.

Chapter 5

# DESIGNING FITTING RULES FOR CONSTRUCTING DIVERGENCE MEASURES

In Section 2.2.4 we presented some of the advantages of graph-based estimators over plug-in methods, however a very restrictive limitation of graph-based estimation is that there is no general approach their usage. This section will introduce a procedure for approximating unknown information-theoretic quantities via a linear combination of directly estimable quantities. Unlike plug-in estimators, graph-based estimators require different graphs and statistics depending on the quantity we wish to estimate. Additionally there are a number of information theoretic quantities for which there currently exists no established approach for graph-based estimation. To resolve these issues, we propose a set of directly estimable basis functions that we can use in the estimation of any unknown quantity. We refer to this proposed method as a Fitting Routine for Optimizing Your Own divergence (FROYO).

The remainder of this chapter is organized as follows. In Section 5.1, we will elaborate on the idea of directly estimable basis functions. Section 5.2 we present the framework for fitting weights which can be used with the chosen basis set in order to approximate an arbitrary information theoretic quantity. Section 5.3 will illustrate the efficacy of the proposed methodology for forming tighter bounds on the Bayes risk.

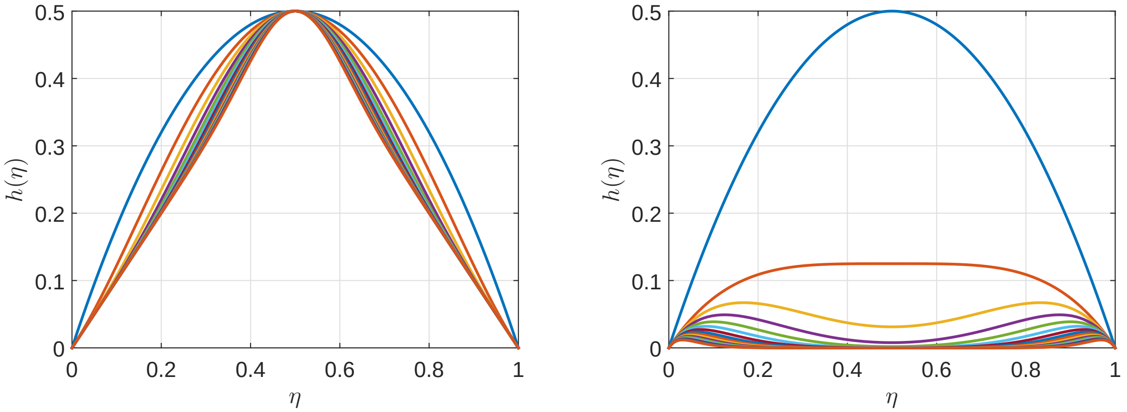
## 5.1 Directly-Estimable Basis Functions

In the previous chapters we have gone over a number of directly estimable information theoretic quantities including the Rényi entropy, the $\alpha$-Jensen divergence, and the $D_p$-divergence. We also discussed the well established asymptotic error rates of various k-NN classifiers. While knowledge of the asymptotic properties of the error rates for k-NN classifiers is useful in its own right for providing performance guarantees on the classifier, we also propose using these quantities as directly estimable basis functions that can be used to estimate information theoretic quantities. This property has been exploited in previous work to estimate and bound the Bayes risk [25, 40, 41, 42], however there has been little investigation into the use of these error rates for other information-theoretic quantities. We would like to exploit the variation in the asymptotic properties of the k-NN classifier w.r.t. $k$ in order to generate data-driven basis functions that can be used to form estimates of information theoretic quantities that would otherwise require the use of plug-in estimators.

Recall the previously defined posterior class distributions $\eta = \eta_0$ and $\eta_1$, along with the fact that any quantity that can be expressed as a function of these two quantities can expressed by only one by exploiting their symmetry ($\eta_0 = 1 - \eta_1$). Now consider the family of functions that can be expressed in the following form:

$$D(\eta) = \int g(\eta)(p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x}))d\mathbf{x}. \tag{5.1}$$

All $f$-divergence functions are part of this family, as are the Asymptotic Error Rate (AER) functions for the k-NN classifier and the $D_p$-divergence function. Figure 5.1 illustrates how both the traditional and Hellman k-NN rules can be represented as a function of $\eta$. We see as the value of k increases in the traditional k-NN classifier, $h(\eta)$ more tightly bounds $\min(\eta, 1 - \eta)$, which is consistent with Stone's Lemma, and the regardless of k the AER function peaks at $h(0.5)$. This means that, the risk is maximized in regions where the posterior likelihoods are equal for both classes. This is not the case for Hellman's rule, which (for $k \geq 5$) has two peaks that approach zero and one with increasing k. This is due to the fact that Hellman's rule will



(a) Traditional Rule

(b) Hellman Rule

Figure 5.1: Basis functions $(h_k(\eta))$ for the traditional and Hellman nearest neighbor rules for odd $k \leq 31$.

reject almost all instances in regions of equal posterior likelihoods for higher values of k. The two rules exhibit unique asymptotic properties, the usefulness of which will depend on the quantity we wish to estimate. While it is quite possible that none of these AER functions may closely approximate $g(\eta)$, we hypothesize that there exists some linear combination of them that will. In this manner, we would like to think of these AER functions as data-driven basis functions that can be used to estimate new information theoretic quantities given 1) such weights exist and 2) we are able to estimate them. In the next section we will propose a method of estimating the weights, and illustrate the approach on an academic example.

## 5.2 Fitting Routine

First consider the series of discrete points $\tilde{\eta}_1, \tilde{\eta}_2, ..., \tilde{\eta}_{\tilde{N}}$, where $0 \leq \tilde{\eta}_1 < \tilde{\eta}_2 < ... < \eta_{\tilde{N}} \leq 1$. Suppose that we have some set of functions $h_1(\eta), h_2(\eta), ..., h_K(\eta)$ that are directly estimable in an asymptotically consistent manner. While none of these individual functions may closely approximate $g(\eta)$, we hypothesize that there exists some linear combination of them that will. To identify weights which best approximate $g(\eta)$, we form the following optimization problem

$$w_1, ..., w_K = \operatorname*{argmin}_{w_i,...,w_K} \sum_{i=1}^{\tilde{N}} \left| g(\tilde{\eta}_i) - \sum_{k=1}^{K} w_k h_k(\tilde{\eta}_i) \right|^2. \tag{5.2}$$

Using these weights we have identified the directly estimable approximate for $g(\eta)$

$$\hat{g}^*(\eta) = \sum_{k=1}^{K} w_k h_k(\eta). \tag{5.3}$$

As a simple example, let us suppose that for our set of directly estimable functions $h_1(\eta), h_2(\eta), ..., h_K(\eta)$ we have the AER functions for the traditional k-NN classifiers for all odd $k \leq 21$, and we wish to estimate is the Hellinger distance. In this scenario, we can define

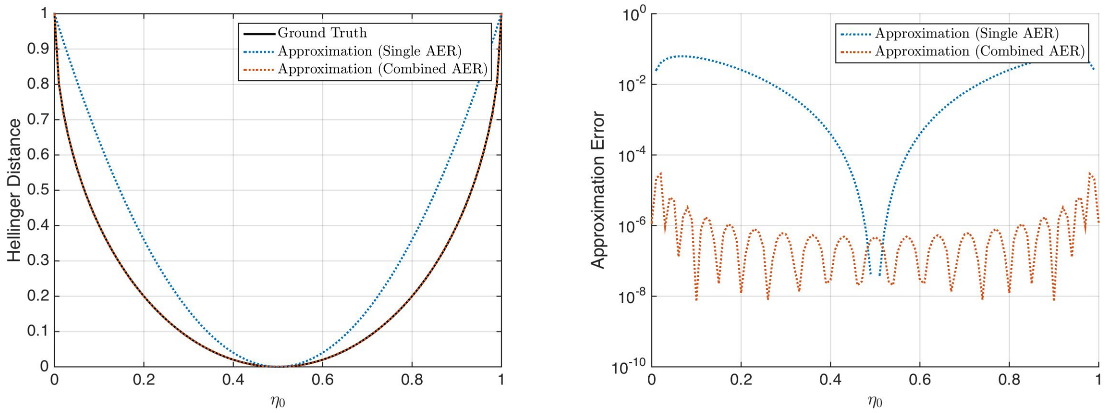
$$g(\eta) = (\sqrt{\eta} - \sqrt{1-\eta})^2. \tag{5.4}$$

If we tried to form a linear combination from a single AER function, the best we could do is the blue line in Figure 5.2a. However, by forming a linear combination we are able to form a near perfect approximation of the Hellinger distance. This example problem will be examined in greater detail in the following Section.

### *5.2.1 Approximation Vs. Estimation Error*

To better understand the performance of the proposed approach it is important to understand the two primary sources of error, which we term the approximation and estimation error. The approximation error ($e_A$) represents the difference between the combination of AER functions $\hat{g}^*(\eta)$ and the desired function $g(\eta)$. The estimation error $e_{est}$ represents the difference between the finite sample estimate

$$\hat{g}(\eta) = \sum_{k=1}^{K} w_k \hat{h}_k(\eta). \tag{5.5}$$



(a) Approximation Function

(b) Approximation Error

Figure 5.2: Single AER vs. linear combination of 11 AERs for approximating the Hellinger distance with directly estimable functions.

and the asymptotic approximation $\hat{g}^*(\eta)$. Using these error types, we can express $g(\eta)$ as

$$\begin{aligned} g(\eta) &= \hat{g}(\eta) + e_A + e_{est} \\ &= \hat{g}(\eta) + (g(\eta) - \hat{g}^*(\eta)) + (\hat{g}^*(\eta) - \hat{g}(\eta)) \end{aligned} \tag{5.6}$$

where

$$e_A = g(\eta) - \hat{g}^*(\eta) \tag{5.7}$$

and

$$e_{est} = \hat{g}^*(\eta) - \hat{g}(\eta). \tag{5.8}$$

Looking back to the previous example, we see that while the approximation formed by solving (5.2) is extremely small, the weights required to reach this solution are on the order of $10^3$, and thus the variance scales similarly. Thus without implementing some type of constraint on the magnitude of these weights, we are unlikely to reach a practical solution. To resolve this issue, we propose the use of a regularization term that penalizes the convex solver for selecting large weights. We propose the following regularized equation

$$\begin{aligned} w_1, ..., w_K &= \underset{w_1,...,w_K}{\operatorname{argmin}} \sum_{i=1}^{\tilde{N}} \left[ g(\tilde{\eta}_i) - \sum_{k=1}^{K} w_k h_k(\tilde{\eta}_i) \right]^2 + \lambda \sum_{k=1}^{K} \left| w_k E_{\tilde{\eta}}[h_k(\tilde{\eta})] \right| \\ &= \underset{w_1,...,w_K}{\operatorname{argmin}} \sum_{i=1}^{\tilde{N}} \left[ g(\tilde{\eta}_i) - \sum_{k=1}^{K} w_k h_k(\tilde{\eta}_i) \right]^2 + \lambda \sum_{k=1}^{K} \left| \frac{w_k}{\tilde{N}} \sum_{j=1}^{\tilde{N}} h_k(\tilde{\eta}_j) \right| \end{aligned} \tag{5.9}$$

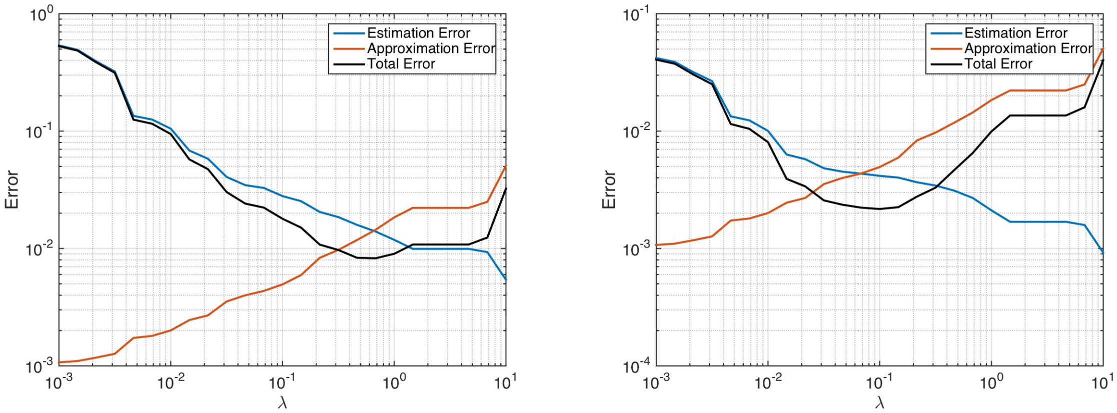
where $\lambda$ represents a tuning constant that balances between the approximation error and estimation error of the final solution. Selecting a small $\lambda$ will yield a solution which very closely approximates the desired measure in the asymptotic regime, however the large weights required to yield such a solution will result in high variance. Larger values of $\lambda$ will yield a solution, that while easier to estimate could contain a large amount of finite sample bias. To illustrate this, lets look at a sample dataset from [39] which considers two multivariate normal distributions $f_0(\mathbf{x})$ $N(\mu_0, \mathbf{\Sigma}_0)$ and $f_1(\mathbf{x})$ $N(\mu_1, \mathbf{\Sigma}_1)$ with parameters

$$\mu_0 = \begin{bmatrix} 2.56 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T ; \mathbf{\Sigma}_0 = \mathbf{I}_8 \tag{5.10}$$

and

$$\mu_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T ; \mathbf{\Sigma}_1 = \mathbf{I}_8 \tag{5.11}$$

respectively. Based on sample sizes of 100 and 1000 drawn from each of these distributions, the proposed methodology is used to estimate the Hellinger distance for several $\lambda$ values varying from $10^{-3}$ to 10. This experiment is repeated across a 100 iteration Monte Carlo simulation, and the results are presented in Figure 5.3. These plots clearly illustrate the trade off between approximation error and estimation error that occurs with the selection of $\lambda$. Furthermore in comparing Figure 5.3a to Figure 5.3b, we see that as sample size increases, the optimal weighting for the regulariza-



(a) $N = 100$

(b) $N = 1000$

Figure 5.3: Plots illustrating the trade-off between estimation error and approximation error for varying values of $\lambda$.

tion term decreases. This is due to the fact that the estimation error decreases with sample size, while the approximation error is unaffected by changes in sample size.

## 5.3 Tighter Bounds On The Bayes Risk

In addition to estimating information-theoretic quantities, we can also use the proposed approach to formulate tighter non-parametric bounds on the Bayes risk. We previously illustrated how the proposed methodology can be used to approximate any general function. Since we can represent the Bayes risk in the form

$$R^* = \int min(\eta, 1-\eta)\Big(p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})\Big), \tag{5.12}$$

the Bayes risk can be approximated by setting

$$g(\eta) = \min(\eta, 1-\eta). \tag{5.13}$$

This approximation can be converted to an upper bound simply by adding the constraint

$$g(\tilde{\eta}) \leq \hat{g}(\tilde{\eta}) \qquad \forall \tilde{\eta} \tag{5.14}$$

making our final optimization problem with L2 regularization:

$$\begin{aligned} \underset{w_i}{\text{argmin}} \qquad & \sum_{i=1}^{\tilde{N}} \left[ \min(\tilde{\eta}_i, 1-\tilde{\eta}_i) - \sum_{k=1}^{K} w_k h_k(\tilde{\eta}_i) \right]^2 + \lambda \sum_{k=1}^{K} \Big| w_k E_{\tilde{\eta}}[h_k(\tilde{\eta})] \Big|^2 \\ \text{Subject to} \quad & \min(\tilde{\eta}_i, 1-\tilde{\eta}_i) \leq \sum_{k=1}^{K} w_k h_k(\tilde{\eta}_i) \\ & \qquad \qquad \cdots \end{aligned} \tag{5.15}$$

Optimizing according to (5.15) we can construct bounds that are, in the asymptotic regime, much tighter than the Bhattacharyya and $D_p$ bounds that have been previously described. Figure 5.4 illustrates $\hat{g}(\eta)$ for each of these bounds relative to the BER, along with the looseness of each bound as a function of $\eta$.

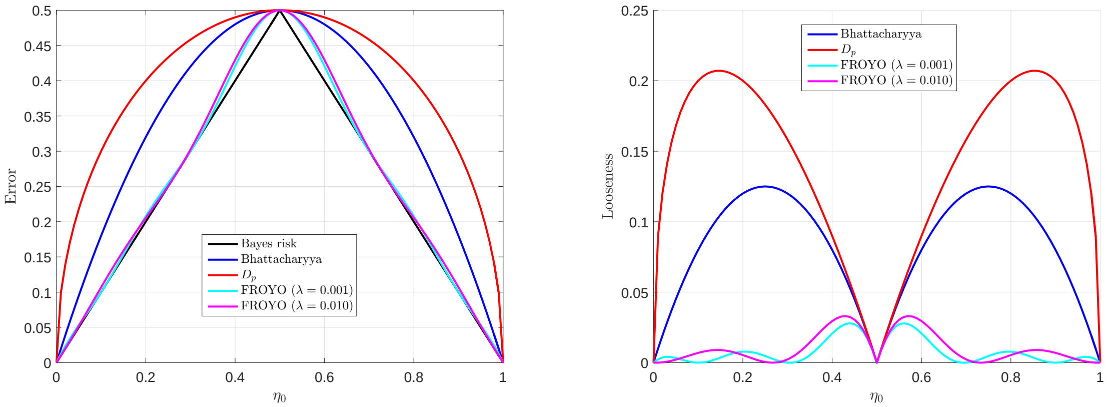
The next step is to empirically examine the efficacy of these bounds in the finite sample regime. To accomplish this we consider two 2-dimensional Gaussian distributions $f_0(\mathbf{x})$ $N(\mu_0, \mathbf{\Sigma}_0)$ and $f_1(\mathbf{x})$ $N(\mu_1, \mathbf{\Sigma}_1)$ with parameters

$$\mu_0 = -\begin{bmatrix} \text{c} \\ \text{c} \end{bmatrix}; \Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.16}$$

and

$$\mu_1 = \begin{bmatrix} \text{c} \\ \text{c} \end{bmatrix}; \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.17}$$

respectively. The mean parameter $c$ is varied from 0 to 2 in increments of 0.1. We then estimate the $D_p$ directly and Bhattacharyya bounds parametrically (assumes Gaussianity), along with the optimized bounds for $\lambda = 0.01$ and $\lambda = 0.1$. We then



(a) Bounds

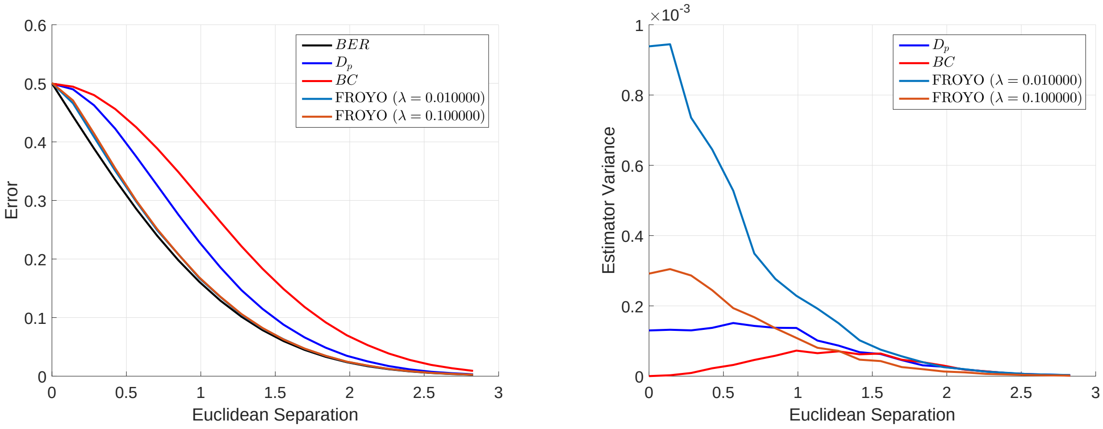
(b) Looseness

Figure 5.4: Plots illustrating the asymptotic bounds as a function of the normalized posterior distribution.

repeat this experiment across a 500 iteration Monte-Carlo simulation. The average value of these bounds in comparison to the true BER and the previously described Bhattacharyya and $D_p$-divergence bounds is displayed as a function of the separation between distributions in Figure 5.5. The results of this simulation are presented in Figure 5.5, in terms of the actual bounds, their average looseness, estimator variance, and MSE w.r.t. the true BER. These results indicate that the proposed method allows much tighter bounds on the BER and that the estimator variance remains relatively small.

In an effort to better understand the finite sample characteristics, we apply the proposed methodology a sample dataset from the Fukunaga book on Pattern Recognitio [39]. This set contains two 8-dimensional Gaussian distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ with parameters

$$\mu_0 = \begin{bmatrix} 2.56 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T ; \mathbf{\Sigma}_0 = \mathbf{I}_8 \tag{5.18}$$



(a) Bounds

(b) Variance

Figure 5.5: Estimation characteristics of the proposed bounds along with the $D_p$ and Bhattacharyya bounds.

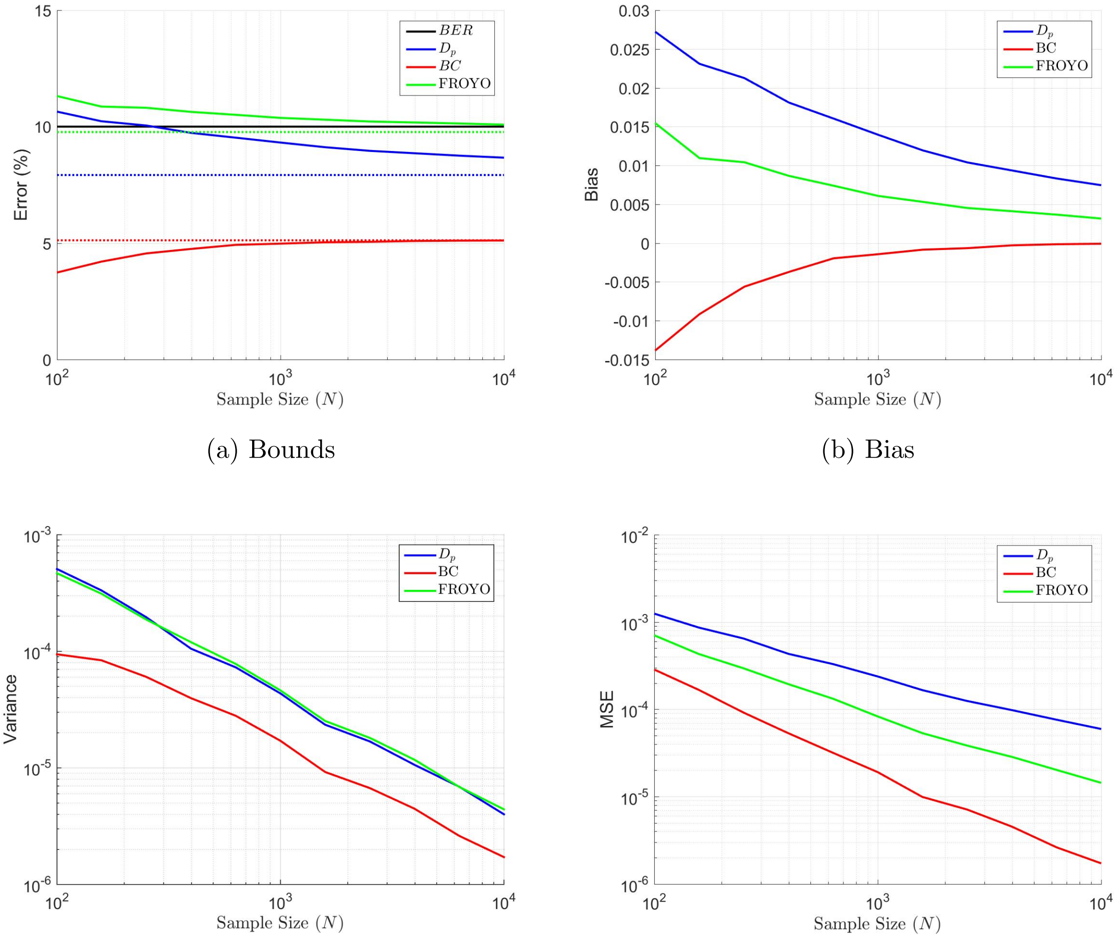
and

$$\mu_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T ; \mathbf{\Sigma}_1 = \mathbf{I}_8 \tag{5.19}$$

respectively. Using these distributions, we compare the performance of the directly estimable $D_p$-divergence bound, the Bhattacharyya bound which assumes Gaussian class distributions, and optimized bounds as a function of sample size, which we vary logarithmically between 100 and 10000. This simulation is then repeated across a 500 iteration Monte Carlo simulation and the results are presented in Figure 5.6 for the lower bounds and 5.7 for the upper bounds. These rsults

In Figures 5.6a and 5.7a the finite samples estimates for each bound are depicted using solid lines while the asymptotic bounds are are depicted using dotted lines. While the Bhattacharyya bounds generally have the best finite sample properties, the accuracy of this estimate is subject to large finite sample and asymptotic bias should the data not fit the assumed parametric model. Furthermore these bounds are looser than the other two methods by a wide margin. We see that in both simulations the proposed methodology yields by far the tightest bounds in the asymptotic regime. Additionally the finite sample properties are superior to the $D_p$ bound in both scenarios and similar to the Bhattacharyya in the lower bound.

The variance of all 3 methods, converges at a rate of $\approx \frac{1}{N}$, however the bias of the Bhattacharyya bound converges to zero much faster than that of the other two methods, and as a result, it converges much faster in the MSE sense as well. Despite the proposed method having a lower initial bias ($N = 100$) in the lower bound, it is quickly surpassed by the faster convergence of the Bhattacharyya bound. One concern this simulation illustrates, particularly for the lower bound, is that as the bounds become tighter, the amount of estimation error necessary to invalidate the bounds lessens. Because of this the $D_p$ lower bound falls above the true BER at smaller sample sizes $N < 300$, and the lower FROYO bound exceeds the BER for all tested sample sizes. As a result an important direction of future work is in improving the convergence rate of the proposed method, however if we are concerned about this problem, a simple solution is to artificially loosen the bound by adding or subtracting some constant.



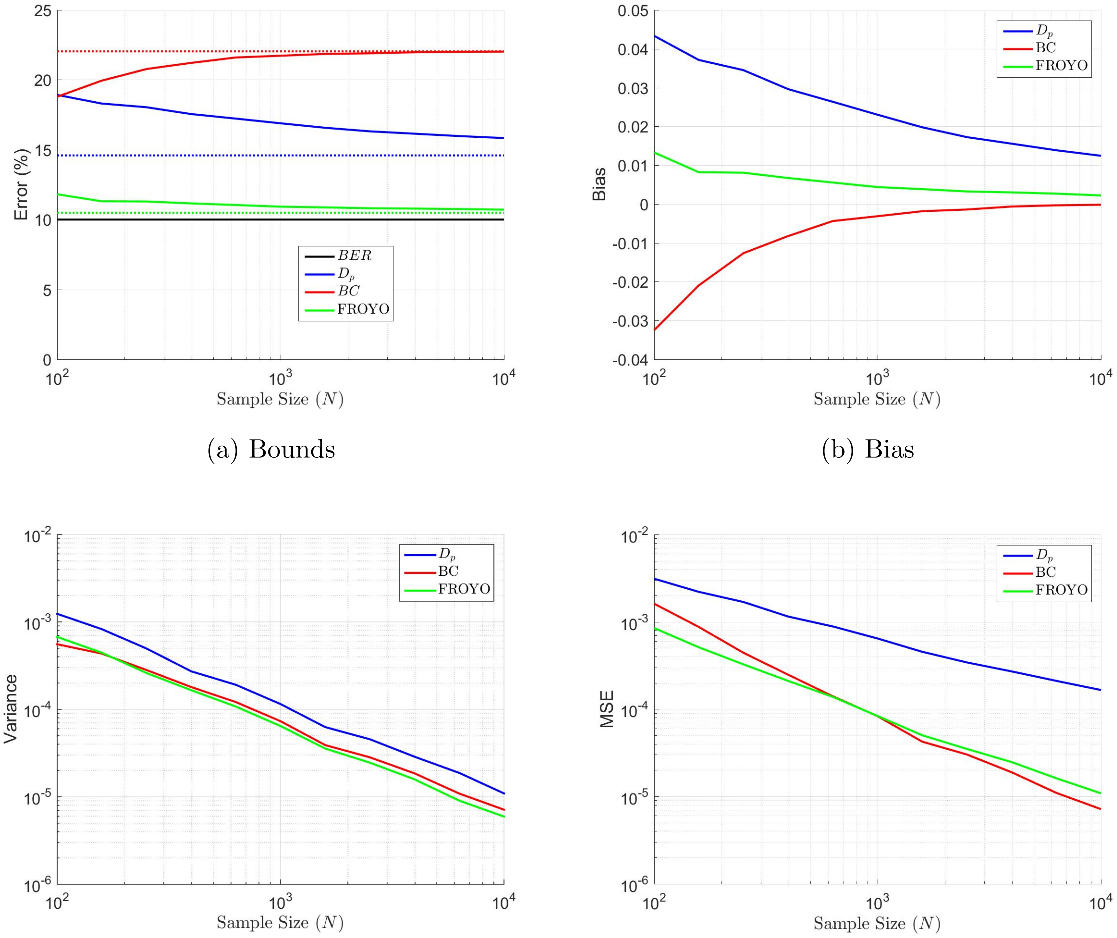
(a) Bounds

(b) Bias

(c) Variance

(d) MSE

Figure 5.6: Lower bounds based on the $D_p$-divergence, Bhattacharyya distance, and proposed methodology along with their estimator properties in terms of bias, variance and MSE.



(a) Bounds

(b) Bias

(c) Variance

(d) MSE

Figure 5.7: Upper bounds based on the $D_p$-divergence, Bhattacharyya distance, and proposed methodology along with their estimator properties in terms of bias, variance and MSE.

Chapter 6

# ALGORITHMS AND RESULTS

In this Section we utilize the error bounds developed in Section 3 to develop robust algorithms for preprocessing data prior to modeling classification and regression problems. In Section 6.1 we use the previously established error bounds to develop feature selection algorithms for both single-domain and cross-domain binary classification problems. We then show how these algorithms can be used in automatic speech-based diagnosis/screening of Dysarthrias. In particular we show how the cross-domain learning algorithm can be used to identify features that will generalize to new speech disorders outside of the training data. Section 6.2 introduces a data removal algorithm that minimizes the bound on the regression error introduced in Section 4.2. We then use this algorithm in order to identify and remove instances that are corrupted by high levels of bias from individual raters.

## 6.1 Feature Selection Using Direct Estimation

In machine learning, feature selection algorithms are often used to reduce model complexity and prevent over-fitting [75]. In many scenarios, feature selection will improve model performance since the reduced dimensionality leads to a much more densely populated hypothesis space and helps prevent overfitting. One way in which we can use the performance bounds proposed in this Dissertation is to identify features that will minimize the estimated error bounds for a given sample of data. Consider that we have some set of training data $\mathbf{X}_{\mathrm{S}}$ with corresponding labels $\mathbf{y}_{\mathrm{S}}$, and an unlabeled set of test data $\mathbf{X}_{\mathrm{T}}$. Algorithm 1 describes a forward selection algorithm that iteratively selects features to minimize a given bound $\mathcal{I}$. Within this general framework we propose to develop feature selection algorithms that minimize any of the error bounds proposed in this dissertation, simply by varying the function defined by $\mathcal{I}$.

We consider four different classification experiments to test the efficacy of different bounds proposed throughout this dissertation. The first two experiments, described in Sections 6.1.1 and 6.1.2 respectively, investigate the binary classification task of discriminating between healthy individuals and those with Dysarthria using sentence-level speech data. The third experiment, found in Section 6.1.3 is a variation on this binary classification task, where the training data contains only individuals with a single subtype of Dysarthria, and we would like to identify features that will generalize well to the other subtypes found in the test data. The fourth experiment, found in Section 6.1.4, is a multi-class classification problem in which we are trying to identify features that discriminate well between the different subtypes of Dysarthria.

### *6.1.1 Experiment 1: Binary Classification Bounds In Dysarthric Speech Classification*

We showed in Theorem 2 that the BER in binary classification problems can be bounded by

$$\frac{1}{2} - \frac{1}{2}\sqrt{u_p(f_0, f_1)} \leq \epsilon^{\text{Bayes}} \leq \frac{1}{2} - \frac{1}{2}u_p(f_0, f_1). \tag{6.1}$$

This experiment will explore the efficacy of selecting features in order to minimize

$$\mathcal{I}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{X}_{\mathrm{T}}(\Omega), \mathbf{y}_{\mathrm{S}}) = \frac{\mathcal{C}(\mathbf{X}_1(\Omega), \mathbf{X}_2(\Omega))}{N_1 + N_2}, \tag{6.2}$$

which reflects the direct estimate of the upper bound in (6.1) for a given set of features $\Omega$ and $\mathbf{X}_1$ and $\mathbf{X}_2$ represent subsets of $\mathbf{X}_{\mathrm{S}}$ when $\mathbf{y}_{\mathrm{S}}$ equals zero and one respectively.

We empirically evaluate the feature selection algorithm on a pathological speech database recorded from patients with neurogenic disorders. In particular, we consider

**Algorithm 1** Forward selection algorithm to minimize performance bound $\mathcal{I}$.

```
Input: Feature data from two different classes in the source
       domain and unlabelled data from the target
       domain:$\mathbf{X}_{\mathrm{S}}$, $\mathbf{y}_{\mathrm{S}}$, $\mathbf{X}_{\mathrm{T}}$, $\mathcal{I}$
Output: Top $k$ features that minimize $\mathit{\Phi}$ :
        $\Omega$
Define: $\Omega = \emptyset$
        $F = 1 \dots M$
for $j \in 1 \dots k$ do
    $\mathit{\Phi} = \emptyset$
    for $F_i \in F \setminus \Omega$ do
        $\mathit{\Phi}(F_i) = \mathcal{I}(\mathbf{X}_{\mathrm{S}}(\Omega \cup F_i), \mathbf{X}_{\mathrm{T}}(\Omega \cup F_i), \mathbf{y}_{\mathrm{S}})$
    end for
    $\Omega = \Omega \cup \{\mathbf{argmin}_{F_i}\ \mathit{\Phi}(F_i)\}$
end for
```

the problem of classifying between healthy and dysarthric speech. Dysarthria is a motor speech disorder resulting from an underlying neurological injury. We make use of data collected in the Motor Speech Disorders Laboratory at Arizona State University, consisting of 34 dysarthric speakers and 13 healthy speakers (H). The dysarthria speakers included: 12 speakers with ataxic dysarthria, secondary to cerebellar degeneration (A), 10 mixed flaccid-spastic dysarthria, secondary to amyotrophic lateral sclerosis (ALS), 8 speakers with hypokinetic dysarthria secondary to Parkinson's Disease (PD), and 4 speakers with hyperkinetic dysarthria secondary to Huntington's disease (HD). Each patient provided speech samples, including a reading passage, phrases, and sentences. The speech database consists of approximately 10 minutes of recorded material per speaker. These speech samples were taken from the larger pathological speech database described in [72].

The recordings from each speaker were split into individual sentences by hand and features were extracted at the sentence level. Three different feature sets were used: envelope modulation spectrum (EMS) features, long-term average spectrum (LTAS) features, and ITU-T P.563 features. EMS is a representation of the slow amplitude modulations in a signal and captures aspects of the speech signal related to rhythm. The LTAS features capture atypical average spectral information in the signal. The P.563 features measure atypical and unnatural voice and articulatory quality. For a more detailed discussion of these features, we refer the readers to [15].

For this experiment we form both the training and test sets by randomly drawing 300 dysarthric speech samples and 300 healthy speech samples for each set, ensuring that there is no overlap between training and test data. Using the FS algorithm in Alg. 1, we use the training data to find the top 20 features that minimize the respecitive BER bound. We compare this feature selection algorithm against one that uses a parametric estimate of the Bhattacharyya bound for multivariate normal distributions. For every feature set ranging in size from 1-20, we build support vector machine (SVM) classifiers on the training data and evaluate their accuracy on the test data. This experiment is repeated ten times using different randomly generated training and test sets, and the average accuracy is displayed in Figure 6.1.

The results of this experiment indicate that the initial features selected by the $D_p$-distance criteria provide faster convergence to the maximum classification rate when compared to those selected by the BC criteria; however, as expected, as additional features are selected, both algorithms eventual converge to roughly the same level of performance. We purposefully restrict ourselves here to a very limited training set (300 samples per class) in order to evaluate the $D_p$-distance in a small $N$ setting. Next, we consider the same problem but with a variable number of training samples

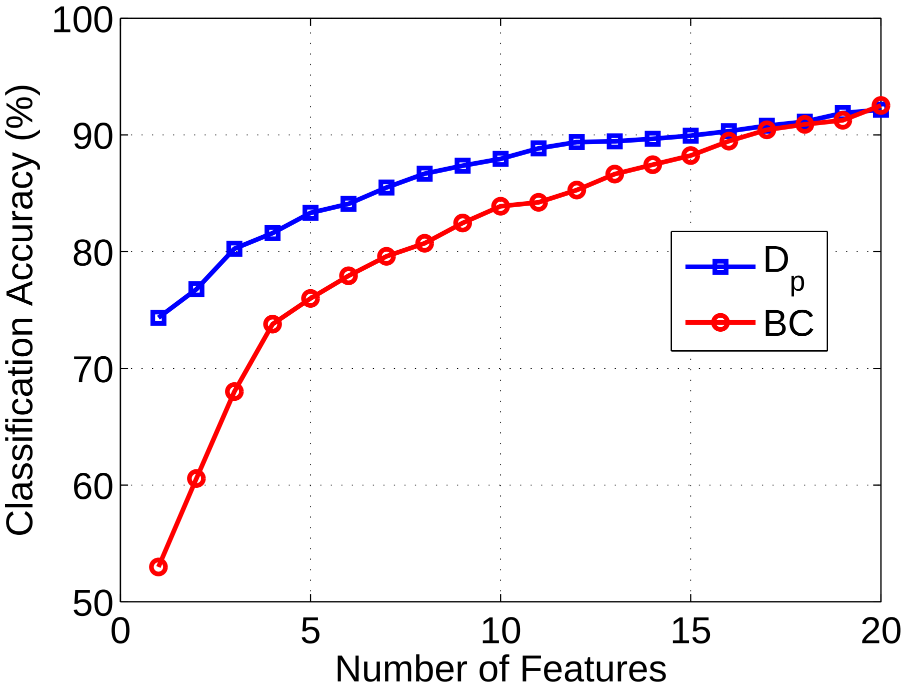


Figure 6.1: Average classification accuracy using reduced feature sets.

per class. The results of this experiment are presented in Table 6.1. As the number of training instances increases, the classifier success rate increases for the $D_p$-based method, however it stays relatively flat for the BC-based method. For very small values of $N$, the bias/variance associated with the $D_p$-distance estimator seems to results in features that provide poorer separability when compared to the BC method. Given that the results of this estimator are asymptotic, this is expected. As the number of features increase, both the $D_p$ and BC algorithms converge to approximately the same value.

### *6.1.2 Experiment 2: Fitting Routine Bounds For Dysarthric Speech Classification*

This experiment shares the same goal as the previous experiment (detection of Dysarthria based on speech data) however it will be used to evaluate the use of

Table 6.1: Average classification accuracies (in percent) of top 10 features selected $D_p$-divergence and Bhattacharyya distance criteria

| **Number of Features** | **Algorithm** | **Number of Training Instances** | | | | |
|---|---|---|---|---|---|---|
| | | *100* | *200* | *300* | *400* | *500* |
| 10 | BC | **86.88** | 86.93 | 87.61 | 87.98 | 87.22 |
| | $D_p$ | 86.36 | **88.67** | **89.59** | **89.20** | **90.03** |
| 15 | BC | **90.84** | 90.46 | 90.51 | 91.69 | 90.88 |
| | $D_p$ | 88.08 | **90.66** | **92.00** | **92.12** | **92.72** |
| 20 | BC | **91.10** | **93.02** | **93.35** | **93.98** | 93.72 |
| | $D_p$ | 89.28 | 92.15 | 93.20 | 93.41 | **94.21** |

the fitting routine introduced in Chapter 5 for feature selection. One interesting question which arose from the previous experiment, and that we hope to answer in this experiment, is how much of the performance benefits $D_p$ bound are due to the non-parametric method of its estimation rather than overall tightness of the bound. Because the proposed fitting routine allows us a method of directly estimating any $f$-divergence, we can identify a set of weights that will yield a directly estimable approximation of the Bhattacharyya distance. For this experiment, we set the optimization criteria to

$$\mathcal{I}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{X}_{\mathrm{T}}(\Omega), \mathbf{y}_{\mathrm{S}}) = \sum_{k=1}^{8} w_k \epsilon_{2k-1}^{NN}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{y}_{\mathrm{S}}) + w_0, \tag{6.3}$$

where $\epsilon_k^{NN}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{y}_{\mathrm{S}})$ represents the error rate of the k-NN classifier on dataset $(\mathbf{X}_{\mathrm{S}}, \mathbf{y}_{\mathrm{S}})$ using feature set $\Omega$ and the weights $w_0, w_1, ..., w_8$ are identified by solving (5.9) for $\lambda = 0.1$. Based on the results shown in Figure 5.3, this value of $\lambda$ appears to be close to optimal for the sample sizes used in this experiment (300 samples per class). By comparing features identified by this direct estimate of the Bhattacharyya distance with features selected by the parametric estimate of the Bhattacharyya distance and the direct estimate of the $D_p$-divergence, we hope to gain some insight into the importance of direct estimation in selecting optimal features for binary classification tasks.

The initial data set used in this experiment is drawn from an updated version of the previously described data set. Our starting data for this experiment contains samples from 78 dysarthric speakers and 13 healthy speakers (H). The dysarthric speakers included: 17 speakers with ataxic dysarthria, secondary to cerebellar degeneration (A), 15 mixed flaccid-spastic dysarthria, secondary to amyotrophic lateral sclerosis (ALS), 39 speakers with hypokinetic dysarthria secondary to Parkinson's Disease (PD), and 7 speakers with hyperkinetic dysarthria secondary to Huntington's disease (HD). From this data set, we extract same LTAS and EMS features discussed previously, however we have replaced the P.563 features with a set of 234 features based on Mel-Frequency Cepstral Coefficients. From this data, disjoint sets of four healthy speakers and 30 Dysarthric speakers are randomly selected and assigned to the training and test data. We randomly select 75 instances from each healthy speaker and 10 instances from each dysarthric speaker so that both the training and test data are composed of 600 total samples.

From this data the top 20 features are selected to minimize the parametric and direct estimates of the Bhattacharyya bound along with the direct estimate of the $D_p$ bound. For every feature set ranging in size from 1-20, we build a SVM and evaluate the subset based on the error of the SVM on the reduced feature set. This experiment is repeated 100 times and the results are averaged and displayed in Figure 6.2.

Despite the slight variations the experimental design the performance of feature selection algorithms based on the parametric estimate of the Bhattacharyya bound and the direct estimate of the $D_p$ bound is largely consistent with the findings of ex-

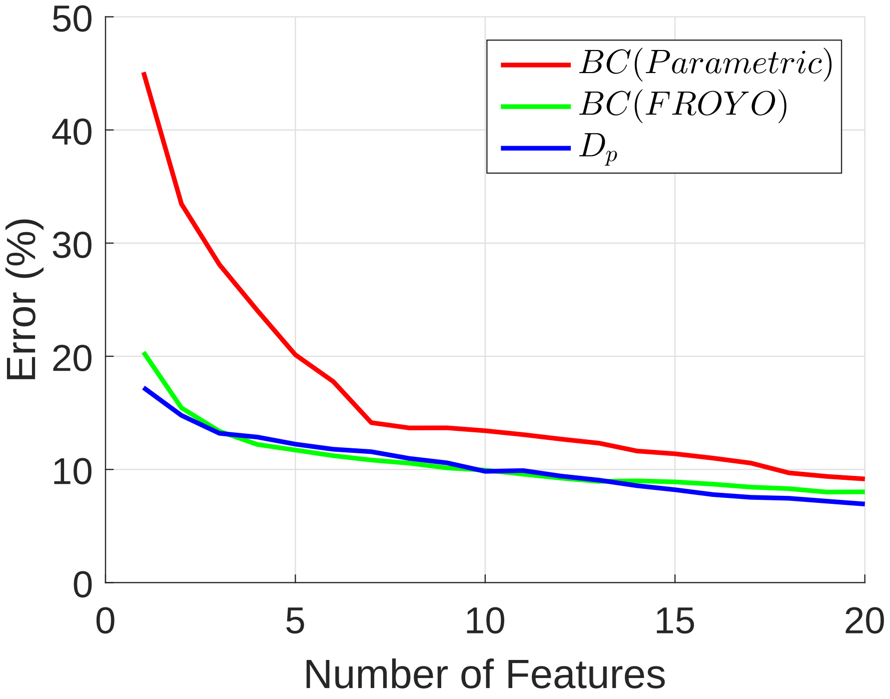


Figure 6.2: Average classification accuracy using reduced feature sets.

periment 1. The direct estimate of the Bhattacharyya bound yields performance that closely resembles the performance of the $D_p$ criteria, indicating that non-parametric estimation method likely factors more into the performance on the resulting feature set than the tightness of the bound.

### *6.1.3 Experiment 3: Domain Adaptation Bounds In Dysarthric Speech Classification*

Like in the previous two experiments, Experiment 3 focuses on the task of distinguishing between healthy individuals and individuals with Dysarthria using sentence-level speech data. However unlike in the previous two experiments, we restrict our data so that the training data contains only a single subtype of Dysarthria that is different from the subtypes contained in the test set. Many of the ways in which

Dysarthria affects speech varies across subtypes, as a result traditional machine learning strategies may not generalize well to the test data. To counter this we will select features that minimize the proposed upper bound on the error in the target domain

$$\epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) = \mathbf{E}_{f_{\mathrm{S}}(\mathbf{x})}[|h(\mathbf{x}) - y_{\mathrm{S}}|], \tag{6.4}$$

which can be accomplished by assigning

$$\begin{aligned}\mathcal{I}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{X}_{\mathrm{T}}(\Omega), \mathbf{y}_{\mathrm{S}}) = &\frac{\mathcal{C}(\mathbf{X}_{\mathrm{S},0}(\Omega), \mathbf{X}_{\mathrm{S},1}(\Omega))}{N_{\mathrm{S},0} + N_{\mathrm{S},1}} \\ &+ 2\sqrt{1 - 2\frac{\mathcal{C}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{X}_{\mathrm{T}}(\Omega))}{N_{\mathrm{S}} + N_{\mathrm{T}}}}.\end{aligned} \tag{6.5}$$

To generate the training and test groups used in this experiment, we start by selecting 300 healthy instances for the training set and 300 (different) healthy instances for the test set. The rest of the training and test data is made up of 300 randomly selected samples from one of the four Dysarthria subtypes: Ataxic, ALS, Huntington's and Parkinson's. Each model is then evaluated on the test sets for each subtype not contained in the training set.

Using each training set-test set combination, we generate feature subsets using the proposed selection algorithm, along with three competing algorithms that are used for comparison. The first algorithm we use for comparison is a standard forward selection algorithm based on the BC distance. This algorithm is used as a baseline for comparison, however because it assumes the training and test data come from the same distribution [47], we expect it to perform poorly relative to the other algorithms. Next we use the same Bhattacharyya FS algorithm, however we account for the separation in domains by using feature normalization, as described in [64], prior to feature selection. We refer to this method as BC with feature normalization (BCFN).

The final domain-invariant feature learning algorithm we compare against is based on Conditional Probability Models (CPM), as described in [101]. This approach attempts to select a sparse mapping that maximizes an objective function that trades off between prediction algorithm performance and the distance between target and source distributions (controlled by a Lagrangian parameter $\lambda$). For classification, the logistic regression function is used and a penalization term is added to ensure that the mapping contains minimal contribution from features containing large differences between source and target data. For the specifics of the implementation, we refer the reader to [101]. The same parameter settings are used here. Because this approach utilizes an optimization criteria involving a trade-off between the source-domain separation and the train-test separation, it resembles the proposed FS algorithm more closely than any other method proposed in the literature.

Table 6.2: Classification accuracies of SVM classifier using the top 20 features returned by each feature selection method for each combination of training and test data.

| Trial | Source | Target | BC | BCFN | CPM | $D_p$ |
|---|---|---|---|---|---|---|
| 1 | Ataxic | ALS | 56.50 | 73.28 | 75.82 | **76.22** |
| 2 | Ataxic | Huntington's | 56.83 | 72.52 | 70.12 | **75.12** |
| 3 | Ataxic | Parkinson's | 49.27 | 60.75 | 58.53 | **64.43** |
| 4 | ALS | Ataxic | 52.95 | 66.35 | 54.68 | **67.15** |
| 5 | ALS | Huntington's | 64.25 | **73.67** | 65.50 | 72.23 |
| 6 | ALS | Parkinson's | 54.32 | 65.97 | 69.48 | **73.60** |
| 7 | Huntington's | Ataxic | 49.95 | **53.63** | 43.00 | 49.30 |
| 8 | Huntington's | ALS | 63.40 | 64.12 | 63.17 | **73.00** |
| 9 | Huntington's | Parkinson's | 59.48 | 62.22 | 69.73 | **76.03** |
| 10 | Parkinson's | Ataxic | 41.13 | **55.65** | 42.15 | 48.23 |
| 11 | Parkinson's | ALS | 62.10 | 66.30 | 61.25 | **67.35** |
| 12 | Parkinson's | Huntington's | **73.67** | 71.12 | 64.47 | 68.98 |

We present the average classification accuracies yielded by the top 20 features from each FS algorithm for each train-test combination in Table 6.2. The DA algorithm introduced in Section 6.1 achieved the highest classification accuracy in 8 of the 12 trials, while the BC algorithm scored the lowest 8 of 12 trials. The results clearly illustrate the importance of utilizing domain adaptation in this type of scenario; even an approach as simple as feature normalization yields roughly 8.5 % higher classification accuracy on average. To observe the value of the lower-dimensional subsets generated by each algorithm, we average the accuracy across all twelve trials and display the accuracy as a function of the number of features in Figure 6.3. We can see in this figure that the performance of the proposed algorithm consistently improves as additional features are added. Because the optimization criteria we have selected minimizes the upper bound on the error, the algorithm has a tendency to pick "safe" features; e.g. using this algorithm invariant features are preferred, even if they are less informative in the source domain.

To better understand how DA helps us build robust models, we look at the top two features returned general and DA FS criterions introduced in Section 6.1. Figure 6.4a displays the training and test data plotted across the top two features returned by the general FS criteria. We see that these two features represent a strong separation between the two classes in the training set, however this separation is not similarly represented in the test data, and as a result these features will not be beneficially for the target application. Figure 6.4b displays the data plotted against the top two features returned by the DA FS criteria. Even though the separation between classes in the training data isn't as noticable as in the features returned by the general criteria, both Dysarthria subtypes manifest themselves very similarly within this feature space, and as a result models built on them will generalize well between these two subtypes.

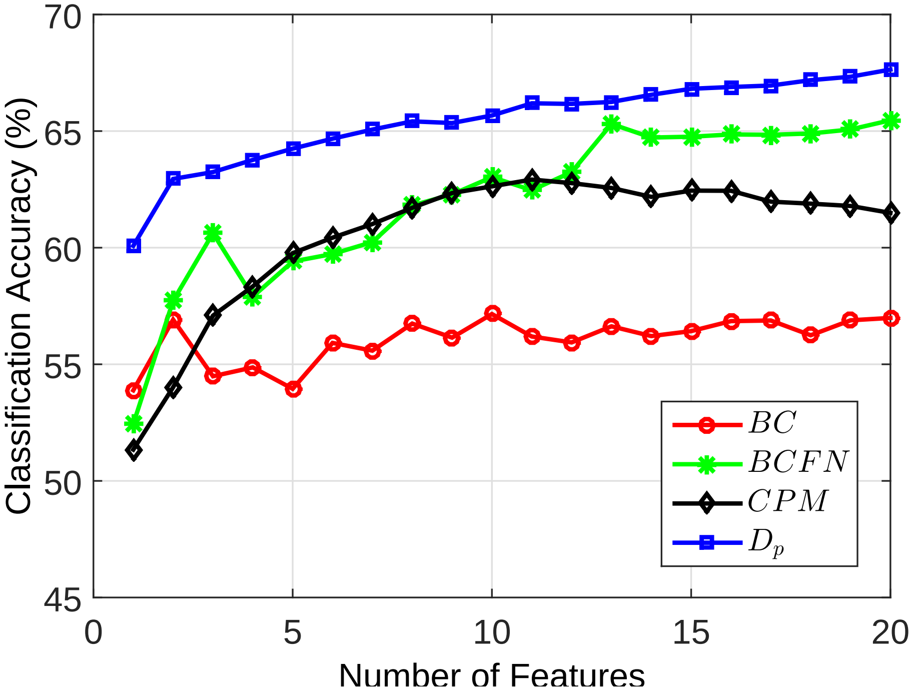
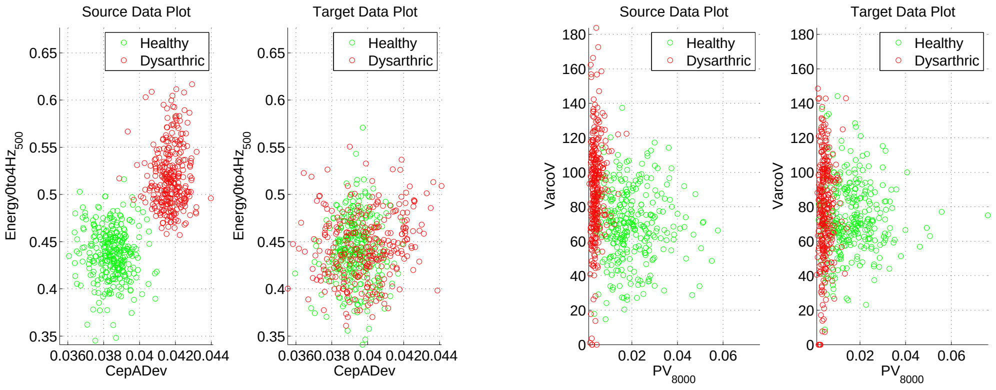


Figure 6.3: Average classification accuracy on foreign subtypes using reduced feature sets.



(a) Source and target data using top domain-specific features

(b) Source and target data using top domain-invariant features

Figure 6.4: Low dimensional representation of datasets (Source Domain:ALS, Target Domain:Parkinson's).

### *6.1.4 Experiment 4: Multi-Class Bounds In Dysarthric Speech Classification*

Unlike the previous three experiments, experiment 4 considers the task of distinguishing between different subtypes of dysarthria, rather than distinguishing between healthy and dysarthric individuals. In Section 4.1 we introduced closed-form and recursive extensions of the $D_p$ bound to multi-class problems. For a general $M$-class classification problem, we can minimize the closed-form extension of the $D_p$ bound by assigning

$$\mathcal{I}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{X}_{\mathrm{T}}(\Omega), \mathbf{y}_{\mathrm{S}}) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \frac{\mathcal{C}(\mathbf{X}_i(\Omega), \mathbf{X}_j(\Omega))}{N}, \tag{6.6}$$

where $\mathbf{X}_i$ represents the set of instances in $\mathbf{X}_{\mathrm{S}}$ with label $i$. Minimization of the recursive bound can be minimized by assigning

$$\mathcal{I}(\mathbf{X}_{\mathrm{S}}(\Omega), \mathbf{X}_{\mathrm{T}}(\Omega), \mathbf{y}_{\mathrm{S}}) = U(\mathbf{X}_1(\Omega), ..., \mathbf{X}_M(\Omega)). \tag{6.7}$$

This can be estimated recursively solving

$$U(\mathbf{X}_1, ..., \mathbf{X}_M) = \min_{\alpha \in \{0,1\}} \frac{1}{M - 2\alpha} \sum_{i=1}^{M} (1 - \frac{N_i}{N}) U([\mathbf{X}_1, ..., \mathbf{X}_M] \setminus \mathbf{X}_i) + \frac{1-\alpha}{M-2\alpha}, \tag{6.8}$$

where $N_i$ represents the number of instances in $\mathbf{X}_i$, $N$ represents the total number of instances in $\mathbf{X}_{\mathrm{S}}$, and $U$ represents the direct estimate of the recursive $M$-class upper bound on the Bayes risk. This is achievable since the 2-class Bayes risk can be calculated using (6.1).

Beginning with the dataset described in Experiment 2, minus the set of Huntington speakers which are removed due to the limited amount of data that is available within that subtype we extract a total of 1201 features including 99 long-term average spectrum (LTAS) features [98], 60 Envelope Modulation Spectrum (EMS) features [73], 234 mel frequency cepstral coefficients (MFCC) features, and 783 additional spatio-temporal features [116]. We then partition the database into training and test sets, by randomly selecting 10 speakers from each subtype and 20 sentences from each speaker to be placed in the training set. From this data we iteratively select features using a forward selection feature selection algorithm that attempts to minimize the criteria in (6.6) and (6.7) along with the closed-form and recursive extensions of the Bhattacharyya bound estimated parametrically. We also include a wrapper feature selection method that iteratively selects the features that maximize the performance of the classifier on a held-out validation set. Wrappers will typically identify the optimal subset of features for the selected classifier, but are computationally very burdensome [65]. Each FS algorithm is used to identify feature subsets of sizes 1-10. For each subset a classification tree is trained on the training data, and evaluated on the test data. This entire procedure is repeated over a 20-iteration Monte Carlo simulation and the average performance achieved by the subsets from each FS algorithm is displayed in Figure 6.5.

Figure 6.5 shows that the $D_p$-based FS algorithm achieved superior performance to BC-based algorithm throughout the experiment, although the gap narrows as additional features are added. While the $D_p$ algorithm achieves slightly higher performance in the smaller subsets, the wrapper yields the highest overall performance. We were not able to observe any significant difference in the closed-form and recursive bounds in this experiment, and other than some of the later features chosen by the $D_p$ algorithm the two methods generally returned the same set of features. This indicates that we are operating in the regime in Figure 4.2 after the two methods have converged and the bounds become virtually identical.

## 6.2 Exemplar Removal Using Bound On Regression Error

A starting point for the algorithm is the theory outlined in Section 4.2. Because $\Phi(\mathbf{X})$ asymptotically converges to the label variance, we propose an algorithm which

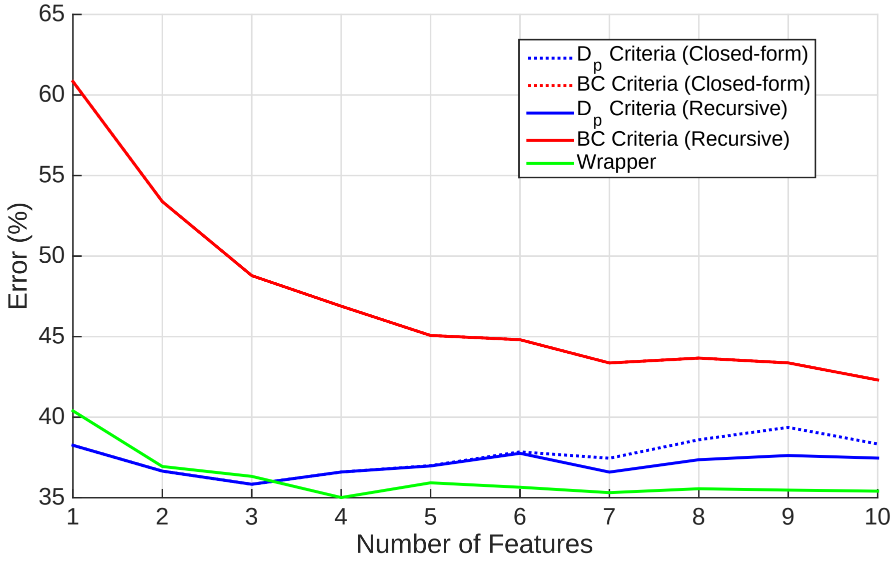


Figure 6.5: Error rates in distinguishing between different speech disorders as a function of the number of features selected.

seeks to identify the subset of instances Ω that minimizes the following metric w.r.t. Ω:

$$\Phi(\mathbf{X}(\Omega), \mathbf{y}) = \frac{1}{(2|\Omega| - 1)} \sum_{e_{ij} \in E} |y_i - y_j|^2 \tag{6.9}$$

There are two problems with seeking to minimize this criterion. First, evaluating this criterion for every potential subset of exemplars would require construction of $2^n - 1$ minimum spanning trees, which will be computationally infeasible even for relatively small data sets. Second, minimizing this criterion without penalizing or restricting the number of exemplars removed would yield a subset containing the two instances with the closest $y$ values. We remedy both of these problems by using a sequential backward selection (SBS) algorithm which will iteratively remove the ”worst” instances until it reaches the desired subset size $k$. By using the SBS algorithm we reduce the required number of MSTs to $\sum_{n=k+1}^{N} n = \frac{1}{2}(N - k)(N + k + 1)$. For large values of $N$, this approach may still be computationally prohibitive in many applications. To further

**Algorithm 2** Iterative exemplar removal using minimum spanning trees.

```
Input: Data Matrix $\mathbf{X}$, Stopping Criteria $k$
Output: Top $N-k$ exemplars that minimize $\mathit{\Phi}$ :
        $\Omega$
Define:  $\Omega = 1 \dots N$
for $j \in 1 \dots k$ do
    $\mathcal{G}(E, \mathbf{X}(\Omega)) = MST(\mathbf{X}(\Omega))$
    for $i \in \mathcal{G}(E, \mathbf{X}(\Omega))$ do
        $\mathit{\Psi}_i(\mathbf{X}(\Omega), \mathbf{y}) = \frac{1}{\rho_i} \sum_{e_{ij} \in E} |y_i - y_j|^2$
    end for
    $\Omega = \Omega \setminus \mathbf{argmax}_i \, \mathit{\Psi}_i$
end for
```

reduce the computational burden we introduce an alternate criterion that represents average squared difference in labels across branches connected to point $\mathbf{x}_i$

$$\mathit{\Psi}_i(\mathbf{X}, \mathbf{y}) = \mathit{\Phi}(\mathbf{X}, \mathbf{y}|i) = \frac{1}{\rho_i} \sum_{e_{ij} \in E} |y_i - y_j|^2 \tag{6.10}$$

In short, this function estimates the average difference between $y$ values for all points connected to $\mathbf{x}_i$ in the MST. As such, it estimates the contribution of instance $\mathbf{x}_i$ to $\mathit{\Phi}(\mathbf{X}(\Omega))$. While this is a heuristic simplification of the optimal criterion in (6.9), empirical simulations found that the new, simplified criterion performs well on a number of experiments (see Section 4). By using the $\mathit{\Psi}$ criterion, the algorithm now requires the construction of $k$ MSTs.

### *6.2.1 Academic Example*

To test the effectiveness of the proposed algorithm we examine the scenario where $y$ can be expressed as a linear combination of $\mathbf{x}$ plus additive gaussian white noise

(AGWN) $\beta \sim N(0, \sigma^2)$.

$$y_i = \sum_{j=1}^{M} \alpha(j)\mathbf{x}_i(j) + \beta_i = \tilde{y}_i + \beta_i \tag{6.11}$$

where $M$ represents the number of features and $\mathbf{x}_i(j)$ represents the value of feature $j$ for instance $i$. For our simulation, we make $\mathbf{X}$ a 25-dimensional matrix containing 2000 exemplars that are normally distributed with zero mean and unit variance. We set all $\alpha$ coefficients equal to one while $\beta$ may take on one of 5 different noise levels with equal probability.

Once the dataset is generated, we use Algorithm 2 to iteratively remove exemplars from the dataset. After each exemplar is removed, we use the reduced dataset to train a linear model. We evaluate the performance of each model based on its mean squared error (MSE) in predicting the true labels $\tilde{y}_i$. The resulting MSE values are displayed in Figure 6.7. It is clear from this figure that removing the high-noise values results in a smaller MSE. In fact, there is an initial dramatic drop in the the MSE (resulting from removing the highest-noise exemplars), followed by a more gradual decline. To verify that the algorithm is correctly identifying the exemplars generated from the higher $\sigma^2$ values we also plot the histogram displaying how many of the first 400 exemplars removed are drawn from each noise level in Figure 6.6. This Figure shows that while the algorithm does select instances for rejection from every noise level, the majority of exemplars that are removed are drawn from the higher noise levels.

### *6.2.2 Dysarthric Speech Example*

A graphical depiction of the experimental setup is provided in Figure 6.8. Beginning with a 4180x123 matrix containing several feature sets including long-term average spectrum (LTAS) [98], Envelope Modulation Spectrum [73], and P.563 [95], we partition the data for cross-validation by removing a single speaker from the train-

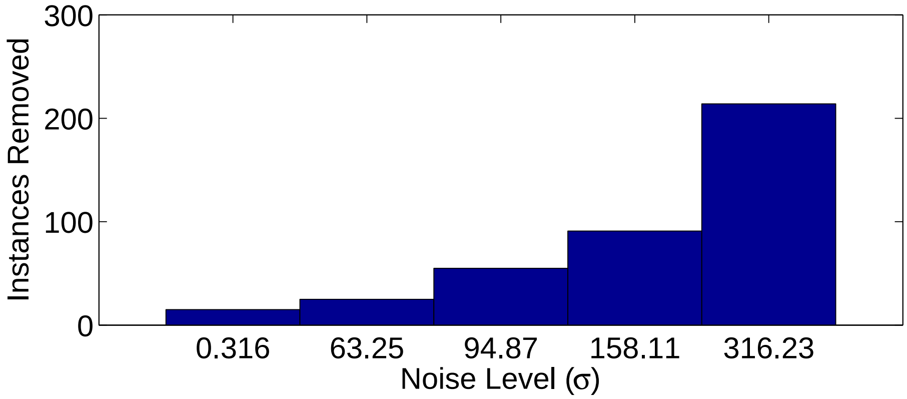


Figure 6.6: Number of exemplars removed corresponding to each noise level among the total 400 samples rejected by the proposed algorithm.

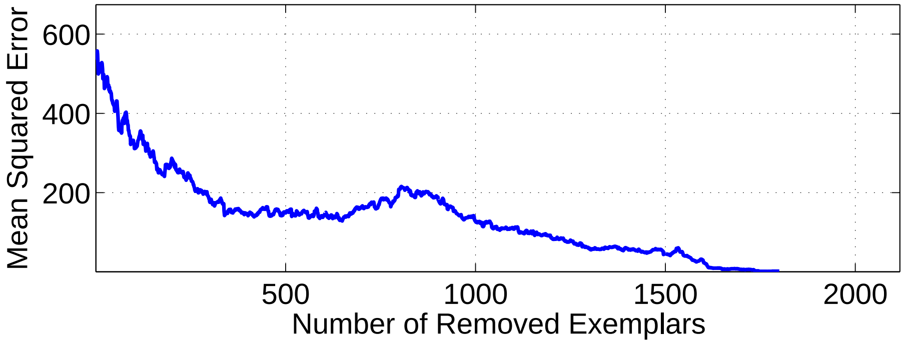


Figure 6.7: MSE as a function of the number of exemplars removed.

ing set and placing them in the test set. We then employ Principal Feature Analysis algorithm [76] with the correlation matrix to identify the top 50 features with minimal redundancy. We then apply Principal Components Analysis to further reduce the dimensionality. The instance rejection procedure described in Algorithm 2 is applied to the resulting dataset to iteratively identify and remove noisy instances. We set $K = 1000$ and select 5 points per MST for removal, and monitor the performance of the algorithm as a function of the number of exemplars removed. After removing the noisy instances we use PCA to form a low-dimensional representation. Finally we use

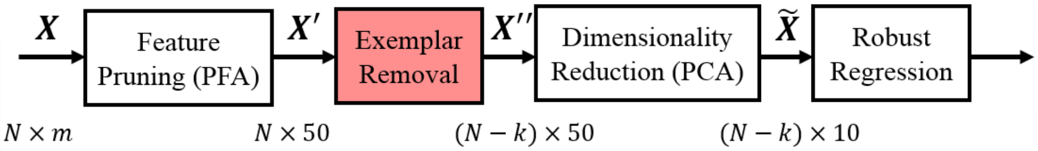


Figure 6.8: Block Diagram of Experimental Design.

the modified dataset, which now contains $N - k$ exemplars expressed in 10 dimensions, to train a linear model using robust least squares with Huber loss. The severity ratings of a single SLP are used as the response variable. The resulting model is then used to predict responses for each of the instances that were partitioned into the test dataset. This yields a set of responses pertaining to a single speaker that we then average to achieve an estimate of the severity rating for that speaker. This process is then repeated until we have generated severity ratings for all 33 speakers and 6 SLPs in the data set. In addition to the exemplar selection algorithm introduced in Section 6.2, we also tested 3 alternate selection methods for selecting examples: (a) k-means clustering, (b) support vector regression, and (c) speaker based exemplar removal. In k-means exemplar selection we identify 250 clusters using the k-means algorithms, then iteratively remove points with the largest distance from their respective cluster mean. Next we a construct a support vector machine (SVM), and remove all instances not chosen as support vectors. We control the number of support vectors by varying the $C$ parameter from 0.025 to 10 on a logarithmic scale while $\epsilon$ is held constant at 0.25. Finally we employ a speaker based pruning method where we look at the data from each individual speaker and remove iteratively remove the instances with the largest standardized euclidean distance from the mean.

Because severity is inherently subjective and no ground truth scores exist, we use the average severity rating of the 5 SLPs not used in training to approximate the true severity rating. We evaluate the performance of each model by correlating the

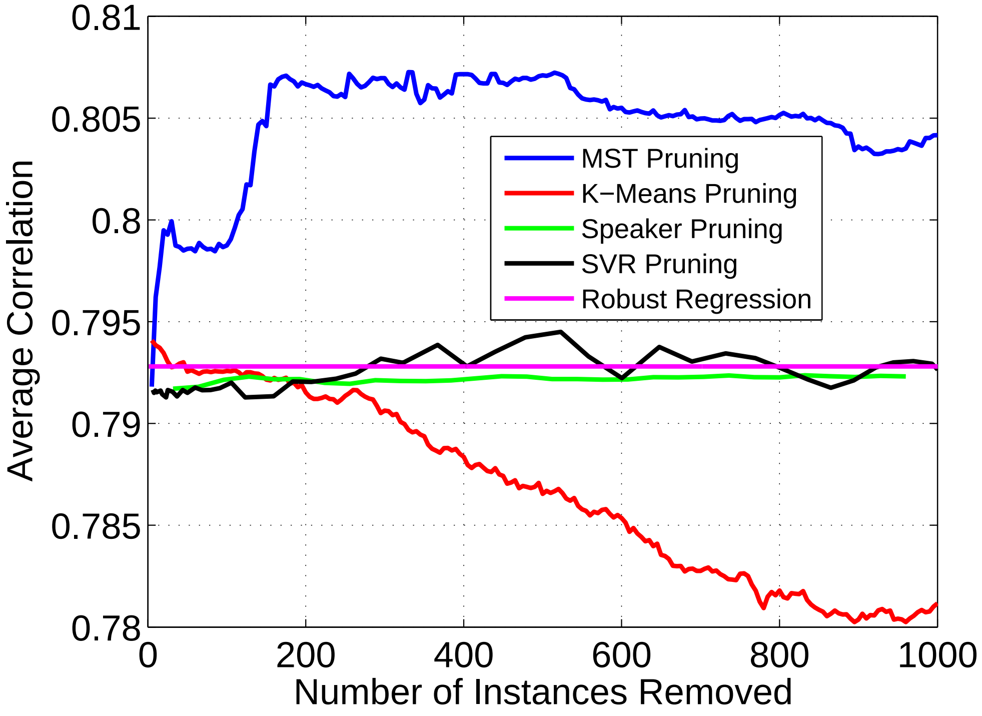


Figure 6.9: Average correlation between objective model and the average subjective rating as a function of the number of exemplars removed, the mean correlation of the individual SLPs against the average score is 0.9332.

predicted ratings with the average rating of the 5 SLPs not used to train the model. These correlation values are then averaged across the 6 different SLP CV stages and the resulting average correlation scores are displayed as a function of the number of exemplars removed in Figure 6.9.

Figure 6.9 shows that of the four exemplar pruning methods that were tested, the proposed MST-based pruning approach is the only one that achieved a noticeable improvement in correlation when averaged across the six SLPs. The proposed method yields an improvement of $\sim 0.015$ in the average correlation score which plateaus at around 150 instances and begins to decline after $\sim 500$ exemplars have been removed. This decline marks the point at which the benefit of reducing the noise in the dataset no longer outweighs the information lost in reducing the overall size of the dataset.

## 6.3 Computational Considerations

In this Section, we will discuss the computational complexity of the algorithms introduced in this dissertation. These algorithms are based on graph-theoretic estimators, and as a result their computational burden is primarily based on the efficiency of the graph construction process. We will first review recent work to develop computationally efficient methods of constructing minimum spanning trees and nearest neighbor graphs, then we will outline the specific computational challenges for each of the major algorithms developed in this document.

### *6.3.1 Euclidean Minimum Spanning Tree Construction*

The computational complexity of minimum spanning tree algorithms is typically described as a function of the number of vertices $n$ and the number of edges $m$ in the original graph. The traditional MST algorithms such as Kruskal's and Prim's work in $O(m \log n)$ and $O(m+n \log n)$ time respectively [68, 92]. Recently, more advanced algorithms have been able to reduce this burden to $O(m \log^* n)$ [36], $O(m \log \log^* n)$[43], $O(m \log \log n)$ [124], and $O(m\alpha(n))$ [91] where $\alpha(n)$ represents the inverse Ackerman. These algorithms function quite well in applications in which are starting edge set is small ($m \ggg n$). However, in the case of euclidean MST computation every pair of vertices represents a candidate edge $m = \frac{n(n-1)}{2}$ and all of these algorithms act in $\emptyset(n^2)$ time.

When working in $\mathbb{R}^2$, we can generate an exact MST in $O(n \log n)$ by first computing the Delaunay triangulation, then running and exact MST algorithm on the reduced edge set. While this provides an optimal solution in $\mathbb{R}^2$, it won't necessarily extend well to higher dimensions where the Delaunay triangulation might be equivalent to the complete graph. Bentley and Friedman proposed an algorithm utilizing $kd$-trees for nearest neighbor searches [13], which they suggest (but do not rigorously prove) operates in $O(n \log n)$ time. March proposed the dual-tree Boruvka algorithm, which modifies the framework of Boruvkas original solution, by forming $kd$-trees across various components as a fast method of identifying neighboring components within a spanning forest [81]. The authors suggest show that this algorithm achieves an asymptotic run time of $O(\alpha(n)n \log n)$, however the complexity of this approach increases exponentially with dimensions, as a results the actual time requirements might be much greater in practice.

An approach employed by a number of algorithms is to first extract a sparse graph from the complete graph, drastically reducing the number of edges, then compute an exact solution on the sparse graph. The accuracy and speed of these methods vary based on the method used to extract the sparse graph. One solution, proposed by Pravin Vaidya [111], is to first subdivide the space into equally sized cubes. Any pair of vertices lying in the same cube will remain connected, while any two vertices lying in different cubes will only remain connected if their respective edge length falls below a given threshold. This algorithm runs in $O(\epsilon^{-d} n \log n)$ time and yields a graph with weight$\leq (1+\epsilon)W_{MST}$, where $W_{MST}$ represents the weight of the true minimum spanning tree. Later Callahan and Kosaraj employed well separated pair decomposition in order to attain a graph with the same maximum weight in $O(n \log n + (\epsilon^{d/2} \log \frac{1}{\epsilon})n)$ time. Zhong et al. proposes a fast minimum spanning tree algorithm, that uses K-means clustering to form $\sqrt{n}$ different subdivisions, then forms uses exact MST algorithms to compute the MSTs within and between the clusters [126]. This algorithm operates in $O(n^{1.5})$ time and is experimentally shown to generate a more accurate solution when compared with the algorithm proposed in [71] the uses the Hilbert curve for clustering. A number of algorithms have been proposed which utilize a well separated pair decomposition (WSPD) to form the graph subdivisions. Agarwal used this approach to develop an algorithm that is bound to $O(F_d(n,n)log(n))$ time, where $F_d(n,n)$ is the time required to solve the bichromatic closest pair problem for $n$ points in $d$ dimensions. Narsimhan et al. propose two variants of this approach in the GeoMST and GeoMST2 algorithms. While the authors expect the latter of these algorithms to operate in $O(n \log n)$ time, this figure ignores how the algorithms complexity varies with dimension [86]. Neemuchwala proposed the modified projection-decomposition algorithm, which attempts to accelerate Kruskal's algorithm from $O(m \log(n) \approx n^2 \log(n))$ to $O(n \log(n))$ by restricting the set of candidate edges to only those likely to be in the minimum spanning tree by removing any edges exceeding $\epsilon$ in length [89]. When $\epsilon$ is chosen such that the number of edges is significantly reduced without removing any edges belonging to the minimum spanning tree, this approach will yield the optimal tree in a highly efficient manner. However if $\epsilon$ is too high the edge set will be inadequately pruned and the performance will be similar to Kruskal's algorithm. If on the other hand $\epsilon$ is too low, then the desired minimum spanning tree will not be a subset of the reduced edge set, and will thus be unattainable. While there are approaches for estimating the appropriate value of $\epsilon$ [89], it is possible in certain data sets that there exists no value of $\epsilon$ that will reach the optimal solution in $O(n \log(n))$ time.

### *6.3.2 Feature Selection Algorithm*

For a given euclidean minimum spanning tree algorithm, suppose constructing said tree requires $\mathcal{T}(n,d)$ steps, where $n$ is the number of samples and $d$ is the dimensionality of the sample space.

For the single-domain feature selection algorithm we begin with a data set composed of $N$ instances measured by $M$ features. Suppose we wish to select the top $L$

features using the feature selection algorithm described in 1, then the total number of steps required for the feature selection algorithm equals

$$T(N, M, L) = \sum_{d=1}^{L} (M - d + 1)\mathcal{T}(N, k). \tag{6.12}$$

Now for the cross-domain learning algorithm, the optimization criterion is calculated the same number of times but now requires the construction of two separate EMSTs in it's calculation. Considering a data set with $N_S$ training instances and $N_T$ test instances, and once again selecting the top $L$ of $M$ total features, the algorithm now requires

$$T(N_S, N_T, M, L) = \sum_{d=1}^{L} (M - d + 1)(\mathcal{T}(N_S, d) + \mathcal{T}(N_S + N_T, d)) \tag{6.13}$$

total steps for computation.

### *6.3.3 Multi-Class Feature Selection*

For the general bound, calculation of the multi-class error rate for a $K$-class problem using $d$ variables requires

$$T_{bound}(N, K, d) = \sum_{i=1}^{K} \sum_{j=i+1}^{K} \mathcal{T}(n_i + n_j, d) \tag{6.14}$$

computations, where $n_i$ represents the number of instances in class $i$. For the recursive bound, the recursive calculation of (4.4) requires approximately an additional $\sum_{i=1}^{d-1} \sum_{i=3}^{M-1} \binom{M}{i} i$ computations. When the calculation of these bounds is integrated into a forward selection algorithm the number of calculation becomes

$$T_{MC}(N, M, L) = \sum_{d=1}^{L} (M - k + 1) T_{bound}(N, K, d). \tag{6.15}$$

By substituting the computations in (6.14) into (6.15) the total calculations becomes

$$T_{MC}^{general}(N, M, L) = \sum_{d=1}^{L} (M - d + 1) \sum_{i=1}^{K} \sum_{j=i+1}^{K} \mathcal{T}(n_i + n_j, d) \tag{6.16}$$

or for the recursive bound

$$T_{MC}^{recursive}(N, M, L) = \sum_{d=1}^{L}(M - d + 1)\left[\sum_{i=1}^{K}\sum_{j=i+1}^{K}\mathcal{T}(n_i + n_j, d) + \sum_{i=3}^{K-1}\binom{M}{i}i\right]. \quad (6.17)$$

If we assume the number of instances-per-class to be a constant value $n = N/K$, then (6.16) can be simplified to

$$T_{MC}^{general}(N, M, L) = \sum_{d=1}^{L}(M - d + 1)\frac{K(K+1)}{2}\mathcal{T}(2n, d) \quad (6.18)$$

### *6.3.4 Data Removal Algorithm*

Consider a data set of $N$ instances measured across a $M$-dimensional feature space. Each time a group of instances is removed this algorithm requires $\mathcal{T}(N_r, M)$ steps, where $N_r$ is the number of remaining instances at that stage. Supposing we wish to remove $L$ samples of data $K$ samples at a time, this algorithm requires

$$T(N, M, L, K) = \sum_{i=1}^{\lceil \frac{L}{K} \rceil}(\mathcal{T}(N - (i-1) * K, M)) \quad (6.19)$$

total steps. While the computational benefits of increasing $K$ are significant, higher values of $k$ are more likely to lead to the removal valuable instances that are corrupted by their proximity to noisy data.

# Chapter 7

# CONCLUSION AND FUTURE WORK

This dissertation has introduced error bounds for binary classification (single and multi-domain), multi-class classification, and regression problems that can be estimated using minimum spanning trees. The bounds introduced for binary classification are tighter than the popular Bhattacharyya bounds and can be estimated without assuming a parametric model. These desirable properties are maintained when these bounds are extended to multi-class problems. The cross-domain learning bounds introduced in this paper similarly do not require parametric assumptions on the class distributions in either domain, nor any labeled data in the target domain. We go on to show each of these bounds can be a powerful tool when used as optimization criteria in a feature selection algorithm. When generalized to regression problems, we are able to upper bound the MSE for an optimal model. Additionally an approach for approximating information-theoretic quantity of the form

$$D(f_0, f_1) = \int f_1(\mathbf{x})\phi\left(\frac{f_0(\mathbf{x})}{f_1(\mathbf{x})}\right)d\mathbf{x} \tag{7.1}$$

for underlying distributions $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ using convex optimization is introduced. While previous graph-based estimates have been limited to particular divergence functions, this approach is applicable for any $f$-divergence and any dispersion function $\phi(t)$.

This work is motivated by a desire to measure the intrinsic difficulty of machine learning problems directly from the dataset. For such a measure to be generally accepted it should be applicable to a wide variety of datasets and have the following properties:

1. Applicable to a variety of problems (binary classification, regression, multi-class classification, cross-domain learning)

2. Estimable for a variety of attributes (discrete, continuous, categorical)

3. Independent of the underlying distributions

4. Estimable and computationally feasible even in large high-dimensional datasets.

These properties are to some degree attained by the work proposed in this dissertation, though not completely. A fundamental limitation of non-parametric statistics in general is that consistency can only be attained asymptotically, and convergence may be arbitrarily slow for particular distributions. As a result, while we can work to develop more robust and practical solutions, we are never truly invariant to the underlying nature of the data being studied. Additionally, due to their basis in minimal graphs, none of the work proposed in this paper is applicable to data with categorical attributes.

There are several interesting directions for future work related to the topics discussed in this dissertation. Acquiring a better understanding the asymptotic properties of minimal graphs is essential for the development of new graph-based estimators. Similarly acquiring a better understanding of their finite sample characteristics is essential for understanding the utility of these measures for different problems. While there exists a direct relationship between asymptotic value of quantities such as the Henze-Penrose 2-sample test statistic and the 1-NN error rate, the difference between the properties of their finite sample estimates remains largely unexplored.

With regards to the fitting routine proposed in Chapter 5, there remain several topics to explore. Improving the finites sample characteristics of estimations based on this approach is critical. This could be accomplished simply by improving the estimator properties of the individual basis functions by using ensemble methods such as those introduced in [105]. However, it is also possible, given a good understanding of the convergence rate of each basis function, to incorporate the bias reduction within the general optimization procedure for the fitting routine. This is potentially a far more computationally efficient approach than applying ensembles to each individual basis function. Additionally, while we have only considered error rates for the traditional k-NN rule as our basis set, we could potentially include any directly estimable quantity in our basis set, and the role that the chosen basis set plays in the methods performance requires further investigation. Obviously, if we wished to estimate a quantity that is asymmetric w.r.t. $\eta_0$, we would need utilize basis functions that are also asymmetric. A final consideration that remains to be explored is the sampling of $\eta_0$ in the fitting routine. We have proposed a uniform sampling of $\eta_0$, with an equal weighting applied to each sample in the optimization criteria, however may not accurately reflect the expected posterior values for the data being studied. It remains to be seen whether a non-uniform sampling of $\eta_0$ could improve the performance of the proposed approach.

Future work in this area should focus not only on improving methods for graph-based estimators but on development of practical tools around the approaches that currently exist. We introduced algorithms that exploit graph-based estimators for performing feature selection and data removal. Previous research has used these estimators for image registration [87] and robust clustering [55], however a number of applications in machine learning remain largely unexplored.

# REFERENCES

[1] Ahmed, N. A. and D. Gokhale, "Entropy expressions and their estimators for multivariate distributions", *IEEE Transactions on Information Theory* **35**, 3, 688–692 (1989).

[2] Ali, S. and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another", *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 131–142 (1966).

[3] Amari, S.-I., "-divergence is unique, belonging to both-divergence and bregman divergence classes", *Information Theory, IEEE Transactions on* **55**, 11, 4925–4931 (2009).

[4] Antos, A., L. Devroye and L. Gyorfi, "Lower bounds for Bayes error estimation", *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **21**, 7, 643–645 (1999).

[5] Avi-Itzhak, H. and T. Diep, "Arbitrarily tight upper and lower bounds on the Bayesian probability of error", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**, 1, 89–91 (1996).

[6] Banerjee, A., S. Merugu, I. S. Dhillon and J. Ghosh, "Clustering with Bregman divergences", *The Journal of Machine Learning Research* **6**, 1705–1749 (2005).

[7] Basseville, M., "Distance measures for signal processing and pattern recognition", *Signal processing* **18**, 4, 349–369 (1989).

[8] Bay, S. D., "Nearest neighbor classification from multiple feature subsets", *Intelligent data analysis* **3**, 3, 191–209 (1999).

[9] Beardwood, J., J. H. Halton and J. M. Hammersley, "The shortest path through many points", in "Mathematical Proceedings of the Cambridge Philosophical Society", vol. 55, pp. 299–327 (Cambridge Univ Press, 1959).

[10] Beirlant, J., E. J. Dudewicz, L. Györfi and E. C. Van der Meulen, "Nonparametric entropy estimation: An overview", *International Journal of Mathematical and Statistical Sciences* **6**, 1, 17–39 (1997).

[11] Ben-David, S., J. Blitzer, K. Crammer, A. Kulesza, F. Pereira and J. W. Vaughan, "A theory of learning from different domains", *Machine learning* **79**, 1-2, 151–175 (2010).

[12] Ben-David, S., J. Blitzer, K. Crammer, F. Pereira *et al.*, "Analysis of representations for domain adaptation", *Advances in neural information processing systems* **19**, 137 (2007).

[13] Bentley, J. L. and J. H. Friedman, "Fast algorithms for constructing minimal spanning trees in coordinate spaces", *Computers, IEEE Transactions on* **100**, 2, 97–105 (1978).

[14] Berisha, V. and A. Hero, “Empirical non-parametric estimation of the fisher information”, *Signal Processing Letters, IEEE* **22**, 7, 988–992 (2015).

[15] Berisha, V., J. Liss, S. Sandoval, R. Utianski and A. Spanias, “Modeling pathological speech perception from data with similarity labels”, in “Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on”, pp. 915–919 (IEEE, 2014).

[16] Berisha, V., A. Wisler, A. O. Hero III and A. Spanias, “Empirically estimable classification bounds based on a nonparametric divergence measure”, *Signal Processing, IEEE Transactions on* **64**, 3, 580–591 (2016).

[17] Bhattacharyya, A., “On a measure of divergence between two multinomial populations”, *Sankhyā: The Indian Journal of Statistics* pp. 401–406 (1946).

[18] Bickel, P. J. and L. Breiman, “Sums of functions of nearest neighbor distances, moment bounds, limit theorems and a goodness of fit test”, *The Annals of Probability* pp. 185–214 (1983).

[19] Blackwell, D. *et al.*, “Comparison of experiments”, in “Proceedings of the second Berkeley symposium on mathematical statistics and probability”, vol. 1, pp. 93–102 (1951).

[20] Blitzer, J., K. Crammer, A. Kulesza, F. Pereira and J. Wortman, “Learning bounds for domain adaptation”, in “Advances in neural information processing systems”, pp. 129–136 (2008).

[21] Cha, S.-H., “Comprehensive survey on distance/similarity measures between probability density functions”, *City* **1**, 2, 1 (2007).

[22] Chen, C., “On information and distance measures, error bounds, and feature selection”, *Information Sciences* **10**, 2, 159–173 (1976).

[23] Chernoff, H., “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations”, *The Annals of Mathematical Statistics* pp. 493–507 (1952).

[24] Cichocki, A. and S.-i. Amari, “Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities”, *Entropy* **12**, 6, 1532–1568 (2010).

[25] Cover, T. M. and P. E. Hart, “Nearest neighbor pattern classification”, *Information Theory, IEEE Transactions on* **13**, 1, 21–27 (1967).

[26] Crooks, G. E., “On measures of entropy and information”, (2015).

[27] Csisz, I. *et al.*, “Information-type measures of difference of probability distributions and indirect observations”, *Studia Sci. Math. Hungar.* **2**, 299–318 (1967).

[28] Csiszár, I. and P. C. Shields, “Information theory and statistics: A tutorial”, *Communications and Information Theory* **1**, 4, 417–528 (2004).

[29] Denoeux, T., “A k-nearest neighbor classification rule based on dempster-shafer theory”, *Systems, Man and Cybernetics, IEEE Transactions on* **25**, 5, 804–813 (1995).

[30] Devroye, L., “Necessary and sufficient conditions for the pointwise convergence of nearest neighbor regression function estimates”, *Probability Theory and Related Fields* **61**, 4, 467–481 (1982).

[31] Dmitriev, Y. G. and F. Tarasenko, “On the estimation of functionals of the probability density and its derivatives”, *Theory of Probability & Its Applications* **18**, 3, 628–633 (1974).

[32] Domeniconi, C. and B. Yan, “Nearest neighbor ensemble”, in “Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on”, vol. 1, pp. 228–231 (IEEE, 2004).

[33] Dudani, S. A., “The distance-weighted k-nearest-neighbor rule”, *Systems, Man and Cybernetics, IEEE Transactions on* , 4, 325–327 (1976).

[34] Ferri, F. J., J. V. Albert and E. Vidal, “Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules”, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **29**, 5, 667–672 (1999).

[35] Fix, E. and J. L. Hodges Jr, “Discriminatory analysis-nonparametric discrimination: Small sample performance”, Tech. rep., DTIC Document (1952).

[36] Fredman, M. L. and R. E. Tarjan, “Fibonacci heaps and their uses in improved network optimization algorithms”, *Journal of the ACM (JACM)* **34**, 3, 596–615 (1987).

[37] Friedman, J. H., “Exploratory projection pursuit”, *Journal of the American statistical association* **82**, 397, 249–266 (1987).

[38] Friedman, J. H. and L. C. Rafsky, “Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests”, *The Annals of Statistics* pp. 697–717 (1979).

[39] Fukunaga, K., *Introduction to statistical pattern recognition* (Academic press, 1990).

[40] Fukunaga, K. and L. D. Hostetler, “K-nearest-neighbor bayes-risk estimation”, *Information Theory, IEEE Transactions on* **21**, 3, 285–293 (1975).

[41] Fukunaga, K. and D. M. Hummels, “Leave-one-out procedures for nonparametric error estimates”, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **11**, 4, 421–423 (1989).

[42] Fukunaga, K. and D. L. Kessell, “Nonparametric bayes error estimation using unclassified samples”, *Information Theory, IEEE Transactions on* **19**, 4, 434–440 (1973).

[43] Gabow, H. N., Z. Galil, T. Spencer and R. E. Tarjan, "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs", *Combinatorica* **6**, 2, 109–122 (1986).

[44] Garber, F. and A. Djouadi, "Bounds on the Bayes classification error based on pairwise risk functions", *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **10**, 2, 281–288 (1988).

[45] Goel, V. and W. J. Byrne, "Minimum bayes-risk automatic speech recognition", *Computer Speech & Language* **14**, 2, 115–135 (2000).

[46] Gowda, K. C. and G. Krishna, "The condensed nearest neighbor rule using the concept of mutual nearest neighborhood", *IEEE Transactions on Information Theory* **25**, 4, 488–490 (1979).

[47] Guorong, X., C. Peiqi and W. Minhui, "Bhattacharyya distance feature selection", in "Pattern Recognition, 1996., Proceedings of the 13th International Conference on", vol. 2, pp. 195–199 (IEEE, 1996).

[48] Hamza, A. B. and H. Krim, "Image registration and segmentation by maximizing the Jensen-Rényi divergence", in "Energy Minimization Methods in Computer Vision and Pattern Recognition", pp. 147–163 (Springer, 2003).

[49] Hand, D. J., "Recent advances in error rate estimation", *Pattern Recognition Letters* **4**, 5, 335–346 (1986).

[50] Hashlamoun, W. A., P. K. Varshney and V. Samarasooriya, "A tight upper bound on the bayesian probability of error", *IEEE Transactions on pattern analysis and machine intelligence* **16**, 2, 220–224 (1994).

[51] Hellinger, E., "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen.", *Journal für die reine und angewandte Mathematik* **136**, 210–271 (1909).

[52] Hellman, M. E., "The nearest neighbor classification rule with a reject option", *Systems Science and Cybernetics, IEEE Transactions on* **6**, 3, 179–185 (1970).

[53] Henze, N., M. D. Penrose *et al.*, "On the multivariate runs test", *The Annals of Statistics* **27**, 1, 290–298 (1999).

[54] Hero, A., B. Ma and O. Michel, "Imaging applications of stochastic minimal graphs", in "Image Processing, 2001. Proceedings. 2001 International Conference on", vol. 2, pp. 573–576 (IEEE, 2001).

[55] Hero, A. and O. J. Michel, "Estimation of rényi information divergence via pruned minimal spanning trees", in "Higher-Order Statistics, 1999. Proceedings of the IEEE Signal Processing Workshop on", pp. 264–268 (IEEE, 1999).

[56] Hero, A. O., B. Ma, O. Michel and J. Gorman, "Alpha-divergence for classification, indexing and retrieval", *Communication and Signal Processing Laboratory, Technical Report CSPL-328, U. Mich* (2001).

[57] Hero III, A. O., B. Ma, O. J. Michel and J. Gorman, "Applications of entropic spanning graphs", *Signal Processing Magazine, IEEE* **19**, 5, 85–95 (2002).

[58] HILBORN, C., "Dg lainiotis", *IEEE transactions on information theory* (1968).

[59] Hild, K. E., D. Erdogmus and J. C. Principe, "Blind source separation using Renyi's mutual information", *Signal Processing Letters, IEEE* **8**, 6, 174–176 (2001).

[60] Hunt, E. B., J. Marin and P. J. Stone, "Experiments in induction.", (1966).

[61] Ivanov, A. V. and M. Rozhkova, "On properties of the statistical estimate of the entropy of a random vector with a probability density", *Problemy Peredachi Informatsii* **17**, 3, 34–43 (1981).

[62] Izenman, A. J., "Review papers: recent developments in nonparametric density estimation", *Journal of the American Statistical Association* **86**, 413, 205–224 (1991).

[63] Kailath, T., "The divergence and bhattacharyya distance measures in signal selection", *Communication Technology, IEEE Transactions on* **15**, 1, 52–60 (1967).

[64] Kinnunen, T. and H. Li, "An overview of text-independent speaker recognition: from features to supervectors", *Speech communication* **52**, 1, 12–40 (2010).

[65] Kohavi, R. and G. H. John, "Wrappers for feature subset selection", *Artificial intelligence* **97**, 1, 273–324 (1997).

[66] Koplowitz, J. and T. A. Brown, "On the relation of performance to editing in nearest neighbor rules", *Pattern Recognition* **13**, 3, 251–255 (1981).

[67] Kristan, M., A. Leonardis and D. Skočaj, "Multivariate online kernel density estimation with gaussian kernels", *Pattern Recognition* **44**, 10, 2630–2642 (2011).

[68] Kruskal, J. B., "On the shortest spanning subtree of a graph and the traveling salesman problem", *Proceedings of the American Mathematical society* **7**, 1, 48–50 (1956).

[69] Kullback, S., "A lower bound for discrimination information in terms of variation (corresp.)", *Information Theory, IEEE Transactions on* **13**, 1, 126–127 (1967).

[70] Kullback, S. and R. A. Leibler, "On information and sufficiency", *The Annals of Mathematical Statistics* pp. 79–86 (1951).

[71] Lai, C., T. Rafa and D. Nelson, "Approximate minimum spanning tree clustering in high-dimensional space", *Intelligent Data Analysis* **13**, 4, 575–597, URL `http://www.scopus.com/inward/record.url?eid=2-s2.0-69749122146&partnerID=40&md5=8dbaa335548c6246eab3c23eb2c9050c`, cited By 4 (2009).

[72] Lansford, K. L. and J. M. Liss, "Vowel acoustics in dysarthria: Speech disorder diagnosis and classification", *Journal of Speech, Language, and Hearing Research* **57**, 1, 57–67 (2014).

[73] Liss, J. M., S. LeGendre and A. J. Lotto, "Discriminating dysarthria type from envelope modulation spectra", *Journal of Speech, Language, and Hearing Research* **53**, 5, 1246–1255 (2010).

[74] Lissack, T. and K.-S. Fu, "Error estimation in pattern recognition via $L_\alpha$-distance between posterior density functions", *Information Theory, IEEE Transactions on* **22**, 1, 34–45 (1976).

[75] Liu, H. and H. Motoda, *Computational methods of feature selection* (CRC Press, 2007).

[76] Lu, Y., I. Cohen, X. S. Zhou and Q. Tian, "Feature selection using principal feature analysis", in "Proceedings of the 15th international conference on Multimedia", pp. 301–304 (ACM, 2007).

[77] Ma, B., *Parametric and nonparametric approaches for multisensor data fusion*, Ph.D. thesis, The University of Michigan (2001).

[78] Macdonald, P., E. Almaas and A.-L. Barabási, "Minimum spanning trees of weighted scale-free networks", *EPL (Europhysics Letters)* **72**, 2, 308 (2005).

[79] Mansour, Y., M. Mohri and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms", *arXiv preprint arXiv:0902.3430* (2009).

[80] Mansour, Y., M. Mohri and A. Rostamizadeh, "Multiple source adaptation and the Rényi divergence", in "Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence", pp. 367–374 (AUAI Press, 2009).

[81] March, W. B., P. Ram and A. G. Gray, "Fast euclidean minimum spanning tree: algorithm, analysis, and applications", in "Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 603–612 (ACM, 2010).

[82] Matusita, K., "Decision rules, based on the distance, for problems of fit, two samples, and estimation", *The Annals of Mathematical Statistics* pp. 631–640 (1955).

[83] Michel, O., R. G. Baraniuk and P. Flandrin, "Time-frequency based distance and divergence measures", in "Time-Frequency and Time-Scale Analysis, 1994., Proceedings of the IEEE-SP International Symposium on", pp. 64–67 (IEEE, 1994).

[84] Mitchell, H. and P. Schaefer, "A soft k-nearest neighbor voting scheme", *International journal of intelligent systems* **16**, 4, 459–468 (2001).

[85] Moreno, P. J., P. P. Ho and N. Vasconcelos, “A kullback-leibler divergence based kernel for svm classification in multimedia applications”, in “Advances in neural information processing systems”, (2003).

[86] Narasimhan, G., J. Zhu and M. Zachariasen, “Experiments with computing geometric minimum spanning trees”, in “Proceedings of ALENEX00”, pp. 183–196 (Citeseer, 2000).

[87] Neemuchwala, H., A. Hero and P. Carson, “Feature coincidence trees for registration of ultrasound breast images”, in “Image Processing, 2001. Proceedings. 2001 International Conference on”, vol. 3, pp. 10–13 (IEEE, 2001).

[88] Neemuchwala, H., A. Hero and P. Carson, “Image matching using alpha-entropy measures and entropic graphs”, *Signal processing* **85**, 2, 277–296 (2005).

[89] Neemuchwala, H. F., *Entropic graphs for image registration*, Ph.D. thesis, The University of Michigan (2005).

[90] Nielsen, F. and R. Nock, “Entropies and cross-entropies of exponential families”, in “Image Processing (ICIP), 2010 17th IEEE International Conference on”, pp. 3621–3624 (IEEE, 2010).

[91] Pettie, S. and V. Ramachandran, “An optimal minimum spanning tree algorithm”, *Journal of the ACM (JACM)* **49**, 1, 16–34 (2002).

[92] Prim, R. C., “Shortest connection networks and some generalizations”, *Bell system technical journal* **36**, 6, 1389–1401 (1957).

[93] Quinlan, J. R., “Induction of decision trees”, *Machine learning* **1**, 1, 81–106 (1986).

[94] Rao, C. R., “Diversity and dissimilarity coefficients: a unified approach”, *Theoretical population biology* **21**, 1, 24–43 (1982).

[95] Rec, I., “P. 563: Single-ended method for objective speech quality assessment in narrow-band telephony applications”, *International Telecommunication Union, Geneva* (2004).

[96] Redmond, C. and J. Yukich, “Asymptotics for euclidean functionals with power-weighted edges”, *Stochastic processes and their applications* **61**, 2, 289–304 (1996).

[97] Rényi, A., “On measures of entropy and information”, in “Fourth Berkeley symposium on mathematical statistics and probability”, vol. 1, pp. 547–561 (1961).

[98] Rose, P., *Forensic speaker identification* (CRC Press, 2003).

[99] Rukhin, A., “Optimal estimator for the mixture parameter by the method of moments and information affinity”, in “Trans. 12th Prague Conference on Information Theory”, pp. 214–219 (1994).

[100] Saon, G. and M. Padmanabhan, "Minimum Bayes error feature selection for continuous speech recognition.", in "Advances in Neural Information Processing Systems", pp. 75–78 (MIT Press, 2001).

[101] Satpal, S. and S. Sarawagi, "Domain adaptation of conditional probability models via feature subsetting", in "Knowledge Discovery in Databases: PKDD 2007", pp. 224–235 (Springer, 2007).

[102] Scott, D. W., "Multivariate density estimation and visualization", in "Handbook of Computational Statistics", pp. 549–569 (Springer, 2012).

[103] Short, R. D. and K. Fukunaga, "The optimal distance measure for nearest neighbor classification", *Information Theory, IEEE Transactions on* **27**, 5, 622–627 (1981).

[104] Sricharan, K., R. Raich and A. O. Hero, "Estimation of nonlinear functionals of densities with confidence", *Information Theory, IEEE Transactions on* **58**, 7, 4135–4159 (2012).

[105] Sricharan, K., D. Wei and A. O. Hero, "Ensemble estimators for multivariate entropy estimation", *Information Theory, IEEE Transactions on* **59**, 7, 4374–4388 (2013).

[106] Steele, J. M., "Growth rates of euclidean minimal spanning trees with power weighted edges", *The Annals of Probability* pp. 1767–1787 (1988).

[107] Stone, C. J., "Consistent nonparametric regression", *The annals of statistics* pp. 595–620 (1977).

[108] Tomašev, N. and D. Mladenić, "Nearest neighbor voting in high-dimensional data: Learning from past occurrences", in "Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on", pp. 1215–1218 (IEEE, 2011).

[109] Tomek, I., "An experiment with the edited nearest-neighbor rule", *IEEE Transactions on Systems, Man, and Cybernetics* , 6, 448–452 (1976).

[110] Turlach, B. A. *et al.*, *Bandwidth selection in kernel density estimation: A review* (Université catholique de Louvain, 1993).

[111] Vaidya, P. M., "Minimum spanning trees in k-dimensional space", *SIAM Journal on Computing* **17**, 3, 572–582 (1988).

[112] Vajda, I., "Note on discrimination information and variation (corresp.)", *Information Theory, IEEE Transactions on* **16**, 6, 771–773 (1970).

[113] Wang, J., P. Neskovic and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure", *Pattern Recognition Letters* **28**, 2, 207–213 (2007).

[114] Weinberger, K. Q., J. Blitzer and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification", in "Advances in neural information processing systems", pp. 1473–1480 (2005).

[115] West, D. B. *et al.*, *Introduction to graph theory*, vol. 2 (Prentice hall Upper Saddle River, 2001).

[116] Williamson, J. R., T. F. Quatieri, B. S. Helfer, R. Horwitz, B. Yu and D. D. Mehta, “Vocal biomarkers of depression based on motor incoordination”, in “Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge”, pp. 41–48 (ACM, 2013).

[117] Wilson, D. L., “Asymptotic properties of nearest neighbor rules using edited data”, *Systems, Man and Cybernetics, IEEE Transactions on* , 3, 408–421 (1972).

[118] Wisler, A., V. Berisha, J. Liss and A. Spanias, “Domain invariant speech features using a new divergence measure”, in “Spoken Language Technology Workshop (SLT), 2014 IEEE”, pp. 77–82 (IEEE, 2014).

[119] Wisler, A., V. Berisha, K. Ramamurthy, A. Spanias and J. Liss, “Removing data with noisy responses in regression analysis”, in “Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on”, pp. 2066–2070 (IEEE, 2015).

[120] Wisler, A., V. Berisha, K. Ramamurthy, D. Wei and A. Spanias, “Emperically-estimable multi-class performance bounds”, in “Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on”, (IEEE, 2016).

[121] Wolpert, D. H., “The supervised learning no-free-lunch theorems”, in “Soft Computing and Industry”, pp. 25–42 (Springer, 2002).

[122] Xuan, G., X. Zhu, P. Chai, Z. Zhang, Y. Q. Shi and D. Fu, “Feature selection based on the Bhattacharyya distance”, in “Pattern Recognition, 2006. ICPR 2006. 18th International Conference on”, vol. 3, pp. 1232–1235 (IEEE, 2006).

[123] Yang, J. and V. Honavar, “Feature subset selection using a genetic algorithm”, in “Feature extraction, construction and selection”, pp. 117–136 (Springer, 1998).

[124] Yao, A. C.-C., “An 0 (— e— loglog— v—) algorithm for finding minimum spanning trees”, *Information Processing Letters* **4**, 1, 21–23 (1975).

[125] Yukich, J. E., *Probability theory of classical Euclidean optimization problems* (Springer, 1998).

[126] Zhong, C., M. Malinen, D. Miao and P. Fränti, “A fast minimum spanning tree algorithm based on k-means”, *Information Sciences* **295**, 1–17 (2015).

# APPENDIX A

# PROOFS

## A.1 Proof Of Theorem 1

By combining Eq. (3.1) and (3.2) from the text we can rewrite

$$D_{p_0} = \frac{1}{4p_0p_1}[1 - 4p_0p_1A_{p_0}(f_0, f_1) - (p_0 - p_1)^2] \tag{A.1}$$

$$= \frac{1 - (p_0 - p_1)^2}{4p_0p_1} - A_{p_0}(f_0, f_1) \tag{A.2}$$

$$= 1 - A_{p_0}(f_0, f_1), \tag{A.3}$$

where

$$A_{p_0} = \int \frac{f_0(\mathbf{x})f_1(\mathbf{x})}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})}d\mathbf{x}. \tag{A.4}$$

From Theorem 2 in [53], we know that as $N_0 \to \infty$ and $N_1 \to \infty$ in a linked manner such that $\frac{N_0}{N_0+N_1} \to p_0$ and $\frac{N_1}{N_0+N_1} \to p_1$,

$$\frac{\mathcal{C}(f_0, f_1)}{N_0 + N_1} \to 2p_0p_1A_{p_0}(f_0, f_1), \tag{A.5}$$

almost surely.

Combining the asymptotic relationship in Eq. (A.5) with the results from Eq. (A.3), we see that

$$1 - \mathcal{C}(f_0, f_1)\frac{N_0 + N_1}{2N_0N_1} \to D_{p_0}(f_0, f_1), \tag{A.6}$$

almost surely as $N_0 \to \infty$ and $N_1 \to \infty$ in a linked manner such that $\frac{N_0}{N_0+N_1} \to p_0$ and $\frac{N_1}{N_0+N_1} \to p_1$.

## A.2 Proof Of Theorem 2

We begin with the realization that the Bayes error rate can be expressed in terms of the total variation (TV) distance between distributions [63]:

$$R^* = \frac{1}{2} - \frac{1}{2}\int |p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x})|d\mathbf{x}. \tag{A.7}$$

Next, we show that we can bound the TV distance from above and below using $\tilde{D}_{p_0}$:

$$
\begin{aligned}
\tilde{D}_{p_0} &= 1 - 4p_0p_1A_{p_0}(f_0, f_1) && \text{(A.8a)}\\
&= 1 - 4p_0p_1 \int \frac{f_0(\mathbf{x})f_1(\mathbf{x})}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})} d\mathbf{x} && \text{(A.8b)}\\
&= \int \left[p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})\right] d\mathbf{x} \\
&\quad - 4p_0p_1 \int \frac{f_0(\mathbf{x})f_1(\mathbf{x})}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})} d\mathbf{x} && \text{(A.8c)}\\
&= \int \frac{[p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})]^2 - 4pqf_0(\mathbf{x})f_1(\mathbf{x})}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})} d\mathbf{x} && \text{(A.8d)}\\
&= \int \frac{p_0f_0(\mathbf{x})^2 + p_1f_1(\mathbf{x})^2 - 2pqf_0(\mathbf{x})f_1(\mathbf{x})}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})} d\mathbf{x} && \text{(A.8e)}\\
&= \int \frac{[p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x})]^2}{p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x})} d\mathbf{x} && \text{(A.8f)}\\
&= \int |p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x})| \frac{|p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x})|}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})} d\mathbf{x}. && \text{(A.8g)}
\end{aligned}
$$

Since

$$
\frac{|p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x})|}{p_0f_0(\mathbf{x}) + p_1f_1(\mathbf{x})} \leq 1 \quad \text{for all } \mathbf{x}, \qquad \text{(A.9)}
$$

we can simplify (A.8g) to

$$
1 - 4p_0p_1A_{p_0}(f_0, f_1) \leq \int |p_0f_0(\mathbf{x}) - p_1f_1(\mathbf{x})| \, d\mathbf{x}. \qquad \text{(A.10)}
$$

This provides a lower bound on the TV distance based on $\tilde{D}_{p_0}$. In order to derive the upper bound we begin with

$$
\begin{aligned}
D_{\mathrm{TV}}(f_0, f_1) &= \int |p_0 f_0(\mathbf{x}) - p_1 f_1(\mathbf{x})| \, d\mathbf{x} && \text{(A.11a)}\\
&= \int |p_0 f_0(\mathbf{x}) - p_1 f_1(\mathbf{x})| \frac{\sqrt{p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})}}{\sqrt{p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})}} d\mathbf{x} && \text{(A.11b)}\\
&\leq \sqrt{\int \left( \frac{p_0 f_0(\mathbf{x}) - p_1 f_1(\mathbf{x})}{\sqrt{p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})}} \right)^2 d\mathbf{x}} && \text{(A.11c)}\\
&\times \cancelto{1}{\sqrt{\int \left( \sqrt{p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})} \right)^2 d\mathbf{x}}}\\
&\leq \sqrt{\tilde{D}_{p_0}(f_0, f_1)}. && \text{(A.11d)}
\end{aligned}
$$

By combining the inequalities in (A.10) and (A.11d) with the relationship in (A.7), we see that we can bound the BER by

$$\frac{1}{2} - \frac{1}{2}\sqrt{\tilde{D}_{p_0}(f_0, f_1)} \leq R^* \leq \frac{1}{2} - \frac{1}{2}\tilde{D}_{p_0}(f_0, f_1). \tag{A.12}$$

## A.3 Proof Of Theorem 3

By the geometric vs harmonic mean inequality,

$$f_0(\mathbf{x})^{p_1} f_1(\mathbf{x})^{p_0} \geq \frac{f_0(\mathbf{x}) f_1(\mathbf{x})}{p_0 f_0(\mathbf{x}) + p_1 f_1(\mathbf{x})}. \tag{A.13}$$

It immediately follows that $A_{p_0}(f_0, f_1) \leq \int f_0(\mathbf{x})^{p_1} f_1(\mathbf{x})^{p_0}$, a scaled Chernoff information function. Thus,

$$A_{p_0}(f_0, f_1) \leq \int f_0(\mathbf{x})^{p_1} f_1(\mathbf{x})^{p_0}. \tag{A.14}$$

## A.4 Proof Of Theorem 4

For equiprobable classes ($p = q = \frac{1}{2}$) The upper and lower bounds on the Bayes error rate based on the Bhattacharyya distance are defined by [63]

$$\frac{1 - \sqrt{1 - BC^2(f_0, f_1)}}{2} \leq R^* \leq \frac{BC(f_0, f_1)}{2}, \tag{A.15}$$

where

$$BC(f_0, f_1) = \int \sqrt{f_0(\mathbf{x}) f_1(\mathbf{x})} d\mathbf{x}. \tag{A.16}$$

To show that the $\tilde{D}_{\frac{1}{2}}$ bound upper bound is tighter than the Bhatacharyya bound we must show that $A_{\frac{1}{2}}(f_0, f_1) \leq BC(f_0, f_1)$. It is clear that this is the case from

Theorem 3. For the $\tilde{D}_{\frac{1}{2}}$ lower bound to be tighter, $BC^2(f_0, f_1)$ must be less than equal to $A_{\frac{1}{2}}(f_0, f_1)$. We show this to be true using the Cauchy-Schwartz inequality:

$$
\begin{aligned}
BC^2(f_0, f_1) &= \left[\int \sqrt{f_0(\mathbf{x})f_1(\mathbf{x})}\right]^2 && \text{(A.17a)}\\
&= \left[\int \frac{\sqrt{f_0(\mathbf{x})f_1(\mathbf{x})}}{\sqrt{\frac{1}{2}(f_0(\mathbf{x}) + f_1(\mathbf{x}))}} \sqrt{\frac{1}{2}(f_0(\mathbf{x}) + f_1(\mathbf{x}))d\mathbf{x}}\right]^2 && \text{(A.17b)}\\
&\le \int \frac{f_0(\mathbf{x})f_1(\mathbf{x})}{\frac{1}{2}(f_0(\mathbf{x}) + f_1(\mathbf{x}))}d\mathbf{x} \cancelto{1}{\int \frac{1}{2}(f_0(\mathbf{x}) + f_1(\mathbf{x}))d\mathbf{x}} && \text{(A.17c)}\\
&= A_{\frac{1}{2}}(f_0, f_1). && \text{(A.17d)}
\end{aligned}
$$

Combining both bounds, we see that

$$
\begin{aligned}
\frac{1}{2} - \frac{1}{2}\sqrt{1 - BC^2(f_0, f_1)} \le & \frac{1}{2} - \frac{1}{2}\sqrt{\tilde{D}_{\frac{1}{2}}(f_0, f_1)}\\
& \le R^* \le \frac{1}{2} - \frac{1}{2}\tilde{D}_{\frac{1}{2}}(f_0, f_1) \le \frac{1}{2}BC(f_0, f_1).
\end{aligned}
$$

## A.5 Proof Of Theorem 5

The proof begins in the same fashion as the result in [11] and then diverges.

$$
\begin{aligned}
\epsilon_{\mathrm{T}}(h, y_{\mathrm{T}}) =& \epsilon_{\mathrm{T}}(h, y_{\mathrm{T}}) + \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) - \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) && \text{(A.18a)}\\
& + \epsilon_{\mathrm{S}}(h, y_{\mathrm{T}}) - \epsilon_{\mathrm{S}}(h, y_{\mathrm{T}})\\
\le& \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) + |\epsilon_{\mathrm{S}}(h, y_{\mathrm{T}}) - \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}})| && \text{(A.18b)}\\
& + |\epsilon_{\mathrm{T}}(h, y_{\mathrm{T}}) - \epsilon_{\mathrm{S}}(h, y_{\mathrm{T}})|\\
\le& \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) + \mathbf{E}_{f_{\mathrm{S}}(\mathbf{x})}[|y_S - y_T|] && \text{(A.18c)}\\
& + \left|\int f_{\mathrm{T}}(\mathbf{x})|h(\mathbf{x}) - y_T|d\mathbf{x}\right.\\
& \qquad \left. - \int f_{\mathrm{S}}(\mathbf{x})|h(\mathbf{x}) - y_T|d\mathbf{x}\right|\\
\le& \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) + \mathbf{E}_{f_{\mathrm{S}}(\mathbf{x})}[|y_S - y_T|] && \text{(A.18d)}\\
& + \int |f_{\mathrm{T}}(\mathbf{x}) - f_{\mathrm{S}}(\mathbf{x})||h(\mathbf{x}) - y_T|d\mathbf{x}\\
\le& \epsilon_{\mathrm{S}}(h, y_{\mathrm{S}}) + \mathbf{E}_{f_{\mathrm{S}}(\mathbf{x})}[|y_S - y_T|] && \text{(A.18e)}\\
& + \int |f_{\mathrm{T}}(\mathbf{x}) - f_{\mathrm{S}}(\mathbf{x})|d\mathbf{x}
\end{aligned}
$$

In (A.18e), we identify an upper bound on the target error expressed using the TV distance between source and target distributions. Using (A.11d) this can be expressed in terms of $\tilde{D}_{\frac{1}{2}}$:

$$\begin{aligned} \epsilon_T(h, y_T) \leq \epsilon_S(h, y_S) + E\{|y_S - y_T|\} \\ + 2\sqrt{\tilde{D}_{\frac{1}{2}}(f_T, f_S)} \end{aligned} \tag{A.19}$$