SEMIPARAMETRIC CLASSIFICATION UNDER A FOREST DENSITY ASSUMPTION

A Dissertation

by

MARY FRANCES DORN

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Clifford Spiegelman
Committee Members,	Vaughn Bryant
	Valen Johnson
	Bani Mallick
Head of Department,	Valen Johnson

May 2017

Major Subject: Statistics

ABSTRACT

This dissertation proposes a new semiparametric approach for binary classification that exploits the modeling flexibility of sparse graphical models. This approach is based on nonparametrically estimated densities, which are notoriously difficult to obtain when the number of dimensions is even moderately large. In this work, it is assumed that each class can be well-represented by a family of undirected sparse graphical models, specifically a forest-structured distribution. By making this assumption, nonparametric estimation of only one- and two-dimensional marginal densities are required to transform the data into a space where a linear classifier is optimal.

This work proves convergence results for the forest density classifier under certain conditions. Its performance is illustrated by comparing it to several state-of-the-art classifiers on simulated forest-distributed data as well as a panel of real datasets from different domains. These experiments indicate that the proposed method is competitive with popular methods across a wide range of applications.

ACKNOWLEDGMENTS

My deepest gratitude goes to my Ph.D. advisor, Dr. Cliff Spiegelman, for his encouragement and support from the very beginning stages of this research. I can't thank him enough for his guidance and his generous contribution of time, energy, and expertise.

Others have also been instrumental to this research. I want to thank Dr. Boaz Nadler for his careful and thorough reading of my drafts, and for helping me to see the bigger picture. Thanks to Amit Moscovich, who has become a good friend as well as collaborator. Thanks also to my committee members, Drs. Val Johnson, Bani Mallick, and Vaughn Bryant, for their insightful comments and suggestions on my dissertation.

This research was made possible by the generosity of Tina and Paul Gardner. Their exceptional dedication to U.S.-Israel research efforts gave me many wonderful opportunities during these past years, including several visits to the Weizmann Insitute of Science.

Many thanks go to my fellow students, mentors, and dear friends who helped me on this journey and made my time in College Station memorable. I want to thank Professor Michael Longnecker, a wonderful teacher who always has his door open, ready to share words of wisdom and encouragement. I also want to specially thank Pat and Bill Smith, for their kindness and generosity, and for including me as a part of their family.

Finally, none of this would have been possible without the unconditional love, support, and encouragement I received from my parents. I owe much to my father, for always pushing me to strive for more, and to my mother, for her endless patience and confidence in my abilities.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Professors Cliff Spiegelman, Valen Johnson, and Bani Mallick of the Department of Statistics, Professor Vaughn Bryant of the Department of Anthropology, and Professor Boaz Nadler of the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science.

Some of the theoretical proofs in this dissertation were done in collaboration with Professors Cliff Spiegelman and Boaz Nadler, and Amit Moscovich, a graduate student at the Weizmann Institute of Science. All other work for the dissertation was completed independently by the student.

Funding Sources

Graduate study was supported by Graduate Assistantships, including research assistantships funded by the National Agricultural Statistics Service (USDA), and a grant from Tina and Paul Gardner through the Texas A&M-Weizmann Institute Collaborative Program. Graduate study was also supported by Department of Statistics resources and by the exeptional generosity of Professor Michael Longnecker.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The classification problem is ubiquitous, with applications in nearly every field. In the simplest case of binary classification, the goal is to assign an observation to one of two groups, given previous cases belonging to each group. For example, microarray datasets may consist of expression profiles for thousands of genes for a number of samples (here, the patients). Given the gene expression information for a new patient, how can we accurately diagnose the patient as having a certain type of cancer? Alternatively, how can we determine whether or not a cancer patient will respond to a specific drug?

Linear discriminant analysis (LDA) is one of the oldest and most widely used methods for classification due to its simplicity and robustness, and its effectiveness in a fair number of cases. However, it is known to fail when the classes are not separable by linear boundaries, when the number of features is large relative to the number of observations, or when the distribution of the data for each class is far from Gaussian. Many improvements have extended LDA to the high-dimensional setting, such as regularization approaches to achieve sparsity (Friedman, 1989; Bickel and Levina, 2004; Guo et al., 2005; Witten and Tibshirani, 2011).

In addition to these methods, numerous approaches exist that have weaker assumptions and offer more flexibility for classifying large datasets from various domains. Examples include Classification and Regression Trees (Breiman et al., 1984), Random Forests (Breiman, 2001), Nearest Neighbor (Cover and Hart, 1967), Artificial Neural Networks (Ripley, 1994; Cheng and Titterington, 1994), and Support Vector Machines (SVM) (Cortes and Vapnik, 1995; Lee et al., 2004). We propose a new semiparametric approach to classification that utilizes a flexible assumption on the dependency structure of the data, and is motivated by the varying approaches taken by Lafferty et al. (2012), Pérez et al. (2009), Park et al. (2011), and Tan et al. (2010).

To motivate our work, consider a binary classification problem, such as diagnosing a patient to have one of two types of breast cancer. The explanatory variables $\mathbf{X}$ belongs to some d-dimensional space $\Omega = \Omega_1 \times \cdots \times \Omega_d$, and the response variable Y takes values in the set $\mathcal{Y} = \{+1, -1\}$. Let p denote the joint probability distribution of $(\mathbf{X}, Y)$. Given a set of labeled samples $\{\mathbf{x}_\ell, y_\ell\}$, the standard task is to construct a classifier $f : \Omega \to \mathcal{Y}$, with good performance (e.g. small misclassification error) on new unlabeled instances of $\mathbf{X}$. Assuming equal costs of misclassification for each class, the optimal (Bayes) classification rule assigns a sample to the class with highest posterior probability, so that in the binary case

$$f_B(\mathbf{x}) = \begin{cases} +1 & \text{if } p(Y = +1|\mathbf{x}) \geq 1/2 \\ -1 & \text{otherwise.} \end{cases} \tag{1.1}$$

Through the application of Bayes' rule, the decision boundary for the optimal classifier can be written in terms of the likelihood ratio

$$LR(\mathbf{x}) = \frac{p(\mathbf{x}|Y = +1)}{p(\mathbf{x}|Y = -1)}. \tag{1.2}$$

A new sample would then be predicted according to whether or not the likelihood ratio is larger than the ratio of class prior probabilities, $p(Y = -1)/p(Y = +1)$. In general, the conditional class densities are unknown and are difficult to estimate in a reliable manner, particularly in high-dimensional settings. The problem then is to find a classifier whose performance is close to that of the Bayes rule.

Methods based on density estimation have not gained much popularity. A main impediment is that the classification data are multivariate, often high-dimensional, and are typically not normally distributed. Accurately estimating the densities nonparametrically requires a large number of labeled samples, and as that requirement grows exponentially with the dimension, it is thus impractical for many real applications. Some methods approximate the decision boundary by simple functions, such as a linear separating hyperplane in the case of LDA and SVM. A second class of methods approximate the conditional densities from some parametric family. Due to the complex interactions among variables in the high-dimensional setting, a large number of parameters are often required, and in many cases the assumed parametric model is not a sufficiently accurate representation of the underlying relationships among the variables. Finally, a third class of methods assumes conditional dependence relationships among the random variables that can be represented by a graphical model.

We take this last approach, restricting our models to a family of distributions based on forest-structured undirected graphical models. This flexible assumption on the structure of the data allows the likelihood ratio to be constructed using densities of no more than two dimensions, based on product rules. This facilitates an approach based on nonparametrically estimated densities that circumvents the “curse of dimensionality” problem. Under certain conditions, and if the forest distribution assumption holds, the proposed method achieves the same asymptotic loss as if we knew the exact densities.

This dissertation is organized as follows. Section 2 provides a brief introduction to undirected graphical models and a review of relevant literature that apply graphical model ideas to the classification task. Section 3 describes a new semiparametric approach to classification that utilizes a flexible graphical model assumption on the dependency structure of the data. Section 4 describes the results from applying the proposed classification approach to synthetic and real datasets. A simulation study explores the performance of the algorithm on data which follow forest-structured distributions. A comparison of the forest density classifier with state-of-the-art approaches is provided on a panel of publicly available datasets. These examples help to evaluate its performance when the forest distribution assumption do not hold.Section 5 provides a summary of the work and proposes future research directions that extend this work. Appendices appear at the end of this document which contain proofs for the theoretical results and supplementary simulation results.

2 LITERATURE REVIEW

2.1 Undirected graphical models

A graphical model is a family of multivariate probability distributions defined on a graph $G = (V,E)$. The vertices $V = \{1,\dots,d\}$ represent the components of the random vector $\mathbf{x} = (x_1,\dots,x_d)$, and the edges in E encode the conditional dependencies among the random variables. We are interested in undirected graphical models, often called Markov random fields, which are based on graphs having only undirected edges. In these graphs, the absence of an edge $e_{ij} \in E$ indicates that the corresponding random variables are conditionally independent given all the other variables, i.e. $x_i \perp\!\!\!\perp x_j | \mathbf{x}_{V\setminus\{i,j\}}$. This is called the *pairwise Markov property* for undirected graphical models, and the joint distribution $p(\mathbf{x}) > 0$ having this property is said to be *Markov* with respect to the graph G. This graph specifies the factorization properties of $p(\mathbf{x})$, as we will see in the following sections. See Lauritzen (1996) and Jordan (1999) for more details on the conditional independence properties of undirected graphical models.

Figure 2.1 illustrates a simple example of a forest, an undirected graphical model which has no cycles. Forests will be of particular interest in the following for our approach. In this example, the variables x_2 and x_3 are conditionally independent given x_1, and the variables $\{x_1,x_2,x_3\}$ are independent of $\{x_4,x_5\}$.

Assuming that the dependencies among the variables for a particular dataset can be reasonably accurately described by a graph structure, there are two different tasks to con-

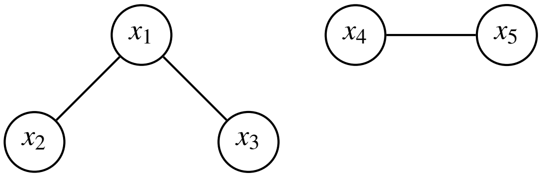

Figure 2.1: A simple forest on five vertices.

sider: the first is graph learning, namely estimating the typically unknown d-dimensional distribution by a distribution that is Markov on an undirected graph, and the second is the construction of classifiers that sensibly use this modeling assumption. We first briefly review the relevant prior work on both tasks. To avoid redundancy in notation, we let p denote either the probability density function or the probability mass function as appropriate to the context.

2.2 Graph learning

The problem of learning the graph structure from the data has a large literature. There are many real applications, such as social networks and biological networks, where the data are naturally represented by a graph. In data having complex relationships among the variables, a graph may reasonably describe useful local structure in a manner that is easy to visualize (Jordan, 1999, 2004).

The most popular methods of estimating the graph G assume that the distribution is Gaussian, as in this case the missing edges in the graph correspond to the zeros in the inverse covariance matrix. Meinshausen and Bühlmann (2006) developed an algorithm to estimate the locations of the zeros in the inverse covariance matrix when the dimensionality is large. However, an assumption that the data is Gaussian is often unreasonable. Liu et al. (2012) and Xue and Zou (2012) proposed the nonparanormal model (or semi-parametric Gaussian copula model) to relax the Gaussian assumption, while still taking advantage of the efficient computational procedures developed under the Gaussian assumption for high-dimensional data.

A second approach, which makes no distributional assumptions, restricts the graph structure to be an undirected tree, where each pair of vertices is connected by only one path. In this case, the probability distribution $p(\mathbf{x})$ factorizes according to the vertices V and edges E, providing a compact representation that involves only pairwise relationships between the variables. To estimate an unknown discrete distribution, Chow and Liu (1968) considered approximations from the following set of permissible tree-structured distributions

$$p_T(\mathbf{x}) = \prod_{i=1}^{d} p(x_{m_i} | x_{m_{j(i)}}), \;\; 0 < j(i) < i,$$

where $(m_1, \ldots, m_d)$ is an unknown permutation of $1, \ldots, d$, $m_0 = 0$, $j(i) \in \{0, \ldots, i-1\}$ is the parent of i in the dependency tree, and $p(x_i | x_0)$ is defined to be equal to $p(x_i)$. They proposed a computationally efficient algorithm to find the optimal p_T, namely the tree-based distribution with minimal Kullback-Leibler (KL) divergence from p. Their procedure finds the maximum-likelihood estimator of the dependence tree. When p itself has a tree structure, Chow and Wagner (1973) showed asymptotic consistency, namely that as sample size tends to infinity, the method recovers the exact distribution, $p_T \to p$.

Recent work has extended this approach to allow a more sparse structure. Tan et al.

(2011) derived a procedure to remove weak edges from the learned tree, resulting in the more general forest structure composed of a union of disjoint trees. A forest is simply an undirected graph having no cycles. Such a forest-structured distribution admits the factorization

$$p_F(\mathbf{x}) = \prod_{(i,j)\in E_F} \frac{p(x_i,x_j)}{p(x_i)p(x_j)} \prod_{i\in V_F} p(x_i), \tag{2.1}$$

where E_F and V_F are the edge and vertex sets of the forest. Here, a forest may include fewer edges than the Chow-Liu tree, which by construction has $d-1$ edges. Lafferty et al. (2012) extended the Chow and Liu approach to the case of multivariate continuous data by using kernel density estimates of the univariate and bivariate densities in (2.1).

We build on these graph-learning approaches in the following section, where the primary goal is accurate classification. In particular, we look at how the properties of the forest distribution can be used to build a classifier with good prediction performance.

2.3 Classification

We return to the binary classification problem. Suppose that the data from each class has a forest-structured distribution of the form Eq. (2.1). In a slight abuse of notation, we denote the density for class y by $p_y(\mathbf{x}) = p(\mathbf{x}|Y=y)$, and the class-conditional univariate and bivariate densities by

$$p_y^i(x_i) = \Pr(x_i|Y=y)$$

$$p_y^{ij}(x_i,x_j) = \Pr(x_i,x_j|Y=y) \quad \text{for } y \in \{+1,-1\}.$$

We will often use the shorthand $p_y(x_i,x_j)$ for $p_y^{ij}(x_i,x_j)$.

Given a labeled training set $\{\mathbf{x}_\ell, y_\ell\}_{\ell=1}^n$, any of the referenced algorithms for model building can be easily adapted to the purpose of classifying new observations by taking a *generative* approach. Once the graphical model representing the conditional dependence structure has been learned for each class, the corresponding estimated densities $p_y(\mathbf{x})$ can then be used to construct the likelihood ratio of Eq. (1.2). The naïve Bayes classifier (e.g. John and Langley, 1995), which assumes the variables to be independent conditional on the class, is the simplest example of this approach. Friedman et al. (1997) introduced the generative likelihood ratio classifier based on the Chow and Liu approach as the tree-augmented naïve Bayes (TAN) classifier. Pérez et al. (2009) formulated the continuous data adaptations of both naïve Bayes and TAN using kernel density estimation as part of their Kernel Based Bayesian Network paradigm, which includes classifiers that assume more complex graph structures than the forest. However, these generative approaches may result in poor classification performance when the classes have nearly identical distributions, as learning the two distributions separately does not emphasize the important differences that may aid in classification.

To allow for the learning of *discriminative* forests in the discrete case, Tan et al. (2010) modified the Chow and Liu algorithm by constructing a graphical model for each class using an objective function that simultaneously minimizes the distance from the distribution for that class and maximizes the distance from the distribution for the other class. Once the forest structure is learned and the corresponding marginal probabilities are estimated, classification proceeds by taking the ratio of the learned forest distributions.

Park et al. (2011) proposed a very different approach for continuous data. Assuming that there is some unknown Markov chain ordering according to which both classes are distributed, they observed that the log-likelihood ratio is linear in the new variables

$$s_{ij} = \log\left(p_{+1}(x_i,x_j)/p_{-1}(x_i,x_j)\right) - \log\left(p_{+1}(x_j)/p_{-1}(x_j)\right),$$

which are simply the log-ratios of univariate and bivariate densities. Their proposed algorithm proceeds by nonparametrically estimating all d univariate densities and all $\binom{d}{2}$ bivariate densities. The log-transformed density ratios, which are pieces of the log-likelihood ratio, are used to obtain a linear classifier. That is, the s_{ij} are treated as variables in a Fisher linear discriminant approach.

In a recent work, Fan et al. (2016) developed a related algorithm, motivated by generalizing naïve Bayes, called feature augmentation via nonparametrics and selection (FANS). Under the naïve Bayes assumption that each feature is independent given the class labels, the log-likelihood ratio can be written very simply as a linear combination of the log-ratios of univariate densities,

$$t_i = \log\left(p_{+1}(x_i)/p_{-1}(x_i)\right).$$

After nonparametrically estimating the univariate densities from each class, their two-step procedure proceeds by running a penalize logistic regression in the space of the etimated transformed variables t_i.

3 CLASSIFICATION UNDER A FOREST DENSITY ASSUMPTION

This section describes the forest density classification method, which addresses the binary classification problem for multivariate continuous (and mixed) data with only weak assumptions about the data structure. This method combines aspects of the approach taken by Lafferty et al. for estimating an arbitrary density with those of Park et al. and Fan et al. for classification.

We begin by generalizing the latter approaches in two key aspects: we allow a general forest dependency structure instead of restricting it to be a simple Markov chain or assuming independence, and we allow the two classes to have potentially different dependency structures. Let us now consider the structure of the optimal classifier when the two classes have forest-structure distributions of the form given in Eq. (2.1), albeit with possibly different forests. The proof of the following lemma is straightforward and left to Appendix A.

Lemma 1. *If the class conditional densities have forest-structure distributions, then the optimal classifier is linear in the variables* $\log p_y(x_i)$ *and* $\log p_y(x_i, x_j)$ *for* $y \in \{+1, -1\}$.

There are many ways to perform linear classification, including popular methods such as LDA and SVM. For the remainder of this work, we will use linear SVM in the transformed (log density) space.

3.1 Forest density classification algorithm

Let $D = \{(\mathbf{x}_\ell, y_\ell)\}_{\ell=1}^{n}$ be a set of i.i.d. labeled training samples from an unknown probability distribution p. We split the n training samples into two disjoint sets D_0 and D_1 of sizes n_0 and n_1, respectively ($n = n_0 + n_1$). In the first step, the n_0 labeled samples in D_0 are used to construct kernel density estimates $\widehat{p}_y(x_i)$ and $\widehat{p}_y(x_i, x_j)$ of the univariate and bivariate densities for each class $y \in \{-1, +1\}$. For simplicity, we use the bivariate notation with $i = j$ to denote the univariate density $p_y(x_i)$.

In the second step, we define a transformation $\widehat{T}_{n_0} : \mathbb{R}^d \to \mathbb{R}^{d(d+1)}$ that takes a feature vector $\mathbf{x}$ and returns all of its estimated univariate and bivariate log densities $\log \widehat{p}_y(x_i, x_j)$ for both classes. We then apply this transformation to the feature vectors in D_1, and construct a linear classifier using the set $S = \{(\widehat{T}_{n_0}(\mathbf{x}_\ell), y_\ell)\}_{\ell=n_0+1}^{n}$ of these transformed vectors and their corresponding labels.

Estimating the log densities with an independent sample helps to simplify the theoretical results in the following section. In practice, the same training sample may be used for both estimating the transformations and constructing the classifiers, particularly when the total number of samples may be relatively small.

3.2 Theoretical properties

In this section, we present our theoretical results on the consistency of our classifier. We begin by studying the consistency of the estimated transformation. All technical proofs are provided in Appendix A.

3.2.1 Estimating the nonlinear transformation

Define the oracle transformation $T:\mathbb{R}^d \to \mathbb{R}^{d(d+1)}$ that takes a feature vector $\mathbf{x}$ and returns all of its exact univariate and bivariate densities. In the following, we provide the rate of convergence of the estimated transformation $\widehat{T}_{n_0}$ to T in the sup norm.

As detailed in Section 3.1, our procedure estimates the univariate and bivariate densities using the n_0 samples in D_0. We denote by n_y the number of samples belonging to class y that are used to estimate the corresponding densities. The univariate kernel density estimator of $p_y(x_i)$ based on n_y observations $(\mathbf{x}_\ell)_i = (x_{\ell,i})$ is defined as

$$\widehat{p}_y(x_i) = \frac{1}{n_y h_1}\sum_{\ell=1}^{n_y} K\left(\frac{(\mathbf{x})_i - (\mathbf{x}_\ell)_i}{h_1}\right), \qquad (\mathbf{x})_i \in \Omega_i,$$

where $K(\cdot)$ is a univariate kernel function and $h_1 > 0$ is the bandwidth which depends on n_y. The bivariate density $p_y(x_i,x_j)$ is estimated using a product kernel as

$$\widehat{p}_y(x_i,x_j) = \frac{1}{n_y h_2^2}\sum_{\ell=1}^{n_y} K\left(\frac{(\mathbf{x})_i - (\mathbf{x}_\ell)_i}{h_2}\right) K\left(\frac{(\mathbf{x})_j - (\mathbf{x}_\ell)_j}{h_2}\right), \qquad (\mathbf{x})_i \in \Omega_i, (\mathbf{x})_j \in \Omega_j, \tag{3.1}$$

where $h_2 > 0$ is the bivariate bandwidth. Let $\nu = 1$ in the univariate case where $i = j$, and let $\nu = 2$ in the bivariate case.

Some smoothness assumptions on the true densities are needed to make the estimation tractable, and for this we use the Hölder class. Our assumptions follow those of Liu et al. (2011). Let $\Omega_{ij} \subset \mathbb{R}^\nu$ be a compact space, and fix constants $\beta, L > 0$. Given any vectors $s = (s_1,...,s_\nu) \in \mathbb{N}^\nu$ and $\mathbf{x} = (x_1,...,x_\nu) \in \Omega_{ij}$, define $|s| = s_1 + \cdots + s_\nu$, $s! = s_1!\cdots s_\nu!$

and $\mathbf{x}^s = x_1^{s_1} \cdots x_\nu^{s_\nu}$. Let D^s denote the differential operator

$$D^s = \frac{\partial^{|s|}}{\partial x_1^{s_1} \cdots \partial x_\nu^{s_\nu}}.$$

For any real-valued function g on Ω_{ij} that is $\lfloor \beta \rfloor$–times continuously differentiable at $\mathbf{x}_0 \in \Omega_{ij}$, let $g_{\mathbf{x}_0}^{(\beta)}(\mathbf{x})$ be its Taylor polynomial of degree $\lfloor \beta \rfloor$ at point $\mathbf{x}_0$:

$$g_{\mathbf{x}_0}^{(\beta)}(\mathbf{x}) = \sum_{|s| \leq \lfloor \beta \rfloor} \frac{(\mathbf{x} - \mathbf{x}_0)^s}{s!} D^s g(\mathbf{x}_0).$$

Denote by $\Sigma(\beta, L_\nu, r, \mathbf{x}_0)$ the set of functions $g : \Omega_{ij} \to \mathbb{R}$ that are $\lfloor \beta \rfloor$–times continuously differentiable at $\mathbf{x}_0$ and satisfy

$$\left| g(\mathbf{x}) - g_{\mathbf{x}_0}^{(\beta)}(\mathbf{x}) \right| \leq L_\nu \|\mathbf{x} - \mathbf{x}_0\|_2^\beta, \quad \forall \mathcal{B}(\mathbf{x}_0, r),$$

where $\mathcal{B}(\mathbf{x}_0, r) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_0\|_2 \leq r\}$. The set $\Sigma(\beta, L_\nu, r, \mathbf{x}_0)$ is called the $(\beta, L_\nu, r, \mathbf{x}_0)$-locally Hölder class of functions. We assume the following about the true univariate and bivariate densities:

(D1) $p_y(x_i, x_j)$ is bounded from above and away from zero and has the same compact support $\Omega_{ij} \subset \mathbb{R}^\nu$ for each class y. Furthermore, there exists $L_\nu > 0$ such that $p_y(x_i, x_j) \in \Sigma(\beta, L_\nu, h_\nu, \mathbf{x}_0)$ $\forall \mathbf{x}_0 \in \Omega_{ij}$.

Here, we require that $p_y(x_i, x_j)$ be bounded because our classifier is constructed in the space of transformed variables which are log-densities.

Our result follows from the application of a finite sample bound for the kernel density estimate due to Giné and Guillou (2002). Their conditions on the kernel are as follows:

(K1) the kernel K is a bounded, square integrable function satisfying $\int K(t)dt = 1$; and

(K2) K is in the linear span of functions $\xi \geq 0$ such that the subgraph of ξ, $\{(s, u) \in$

$\mathbb{R}^{\nu} \times \mathbb{R} : \xi(s) \geq u\}$, can be represented as a finite number of Boolean operations among sets of the form $\{(s,u) \in \mathbb{R}^{\nu} \times \mathbb{R} : q(s,u) \geq \psi(u)\}$, where q is a polynomial on $\mathbb{R}^{\nu} \times \mathbb{R}$ and ψ is an arbitrary real-valued function.

As discussed following the statement of Corollary 2.2 in Giné and Guillou (2002), under these conditions and for any $h_1, h_2 > 0$, the classes of functions

$$\mathcal{F}_{h_1} = \left\{ K\left(\frac{t-\cdot}{h_1}\right), t \in \mathbb{R} \right\}$$

$$\mathcal{F}_{h_2} = \left\{ K\left(\frac{t-\cdot}{h_2}\right) K\left(\frac{u-\cdot}{h_2}\right), t,u \in \mathbb{R} \right\}$$

are bounded measurable VC classes of functions. In other words, for $\nu = 1,2$, the class of functions $\mathcal{F}_{h_\nu}$ is separable and for every probability measure P on $\mathbb{R}^{\nu}$ and any $0 < \varepsilon < 1$ satisfies

$$N\left(\mathcal{F}_{h_\nu}, L_2(P), \varepsilon \|F_{h_\nu}\|_{L_2(P)}\right) \leq \left(\frac{A}{\varepsilon}\right)^{V},$$

where $N(\mathcal{F}_{h_\nu}, L_2(P), \varepsilon)$ denotes the $\varepsilon-$covering number of the metric space $(\mathcal{F}_{h_\nu}, L_2(P))$, and F_{h_ν} is the envelope function of $\mathcal{F}_{h_\nu}$. The constants A and V are called the VC characteristics of the class $\mathcal{F}_{h_\nu}$. Additionally, we assume that

(K3) K has compact support, and for any integer $\ell \geq 1$ and $1 \leq m \leq \lfloor \beta \rfloor - 1$

$$\int |K(t)|^{\ell} dt < \infty, \int |t|^{\beta} |K(t)| dt < \infty, \text{ and } \int t^m K(t) dt = 0.$$

This condition specifies that the kernel be β–valid (Tsybakov, 2009; Rigollet and Vert, 2009). These kernel assumptions allow, for example, both the box and triangular kernels.

Finally, we require that the sequence of kernel bandwidths $\{h_\nu\}$ satisfies

$$h_\nu \to 0, \quad \frac{n_y (h_\nu)^{\nu}}{|\log h_\nu|} \to \infty \quad \text{as } n_y \to \infty. \tag{3.2}$$

It is well known that non-boundary kernel density estimators are not consistent near the boundary of the support of the density being estimated. Numerous approaches have been suggested to correct for the boundary effect. To simplify our analysis, we instead consider the uniform convergence in the following result only for points in the interior of the support, $\Omega^{\mathrm{o}} = \{\mathbf{x} \in \Omega : d(\mathbf{x}, \partial\Omega) > \varepsilon\}$, for some fixed $\varepsilon > 0$. In subsequent analyses using these transformations, we will restrict our estimation to these points which are ε away from the boundary.

Lemma 2. *Assume that (D1) and (K1)-(K3) hold. Then,*

$$\sup_{\mathbf{x}\in\Omega^{\mathrm{o}}} \left\| \widehat{T}_{n_0}(\mathbf{x}) - T(\mathbf{x}) \right\|_{\infty} = O_P\left(\left(\frac{\log n_0}{n_0} \right)^{\frac{\beta}{2+2\beta}} \right). \tag{3.3}$$

Now that we have provided asymptotically bivariate rates of convergence for estimating $\widehat{T}_{n_0}$, we turn to the performance of the linear SVM classifier constructed using these transformed variables.

3.2.2 Asymptotic convergence to oracle SVM

Any linear classifier can be written as $sign(g(\mathbf{x}))$, where $g(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ and $\mathbf{w}$ is its vector of weights. Our procedure constructs a linear classifier in the transformed variables $\widehat{T}_{n_0}(\mathbf{x})$. We use the formulation of support vector machines that falls under the framework of empirical risk minimization. Consider a convex loss function ϕ that is L_ϕ-Lipschitz, i.e. for some $L_\phi \geq 0$, $|\phi(z_1) - \phi(z_2)| \leq L_\phi |z_1 - z_2|$ for all z_1, z_2 in the function domain. In the standard SVM formulation, this is the hinge loss defined as

$$\phi(z) = \max(0, 1 - zy),$$

which is convex in z with Lipschitz constant $L_\phi = 1$. The following definitions are are made for points in the interior of the support Ω^{o}. Define the empirical risk of our classifier in terms of the loss ϕ as

$$\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) = \frac{1}{n_1}\sum_{\ell=1}^{n_1} \phi(\mathbf{w}^{\mathrm{T}}\widehat{T}_{n_0}(\mathbf{x}_\ell), y_\ell). \tag{3.4}$$

The vector of weights is estimated by

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) + \frac{\lambda}{2}\|\mathbf{w}\|^2. \tag{3.5}$$

For any fixed value of $\lambda > 0$, this is equivalent to the following constrained optimization problem,

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}), \tag{3.6}$$

where $\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\| \leq A\}$ for some constant $A > 0$ (see Oneto et al., 2015).

Recall the oracle transformation $T : \mathbb{R}^d \to \mathbb{R}^{d(d+1)}$, which takes a feature vector $\mathbf{x}$ and returns all of its exact univariate and bivariate densities. The expected risk is

$$R(\mathbf{w}, T) = \mathbb{E}_{(\mathbf{X},Y)}[\phi(\mathbf{w}^{\mathrm{T}}T(\mathbf{X}), Y)]. \tag{3.7}$$

We denote by $\mathbf{w}^*$ the population minimizer,

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} R(\mathbf{w}, T). \tag{3.8}$$

In this part, we will show that our procedure is risk consistent with the optimal linear SVM classifier, and that as the number of training samples increases, $\widehat{\mathbf{w}}$ converges in probability to the minimizer of the population risk, $\mathbf{w}^*$. In order to prove this formally, we require the uniform convergence of the empirical risk to the expected risk for all $\mathbf{w} \in \mathcal{W}$. We first aim to provide some intuition.

We have in the previous section shown the uniform convergence of $\widehat{T}_{n_0}(\mathbf{x}) \to T(\mathbf{x})$. Due to the continuity of ϕ, for a fixed $\mathbf{w} \in \mathcal{W}$ and at a particular instance $(\mathbf{x}, y)$,

$$\lim_{n_0 \to \infty} \phi(\mathbf{w}^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}), y) = \phi(\mathbf{w}^{\mathrm{T}} T(\mathbf{x}), y).$$

Since we assumed that the estimate $\widehat{T}_{n_0}$ was constructed using a separate sample D_0, the random variables $\left\{\phi(\mathbf{w}^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}_\ell), y_\ell)\right\}_{\ell=1}^{n_1}$ of the set D_1 are independent and identically distributed. Hence, for any fixed $\mathbf{w}$

$$\lim_{n_1, n_0 \to \infty} \widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) = \lim_{n_1 \to \infty} \widehat{R}_{n_1}(\mathbf{w}, T) = R(\mathbf{w}, T),$$

where the second equality is given by the law of large numbers. We now show that the above convergence is uniform for all $\mathbf{w} \in \mathcal{W}$. The proof for this as well as the following results will be provided in Appendix A.

Recall that $T(\mathbf{x})$ is a vector of log transformed univariate and bivariate densities which are bounded. By Lemma 2, $\|\widehat{T}_{n_0}(\mathbf{x}) - T(\mathbf{x})\|_\infty$ is bounded with high probability for any $\mathbf{x} \in \Omega^{\mathrm{o}}$. In the following results, we will further assume that $\|\widehat{T}_{n_0}(\mathbf{x})\|_\infty < B$ for some $B > 0$. In practice, any transformed variables that are estimated to be infinity are thresholded at a large value. This boundedness of the estimated transformations ensures that the loss is also bounded.

Theorem 3. *Let $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0})$ and $R(\mathbf{w}, T)$ be as defined in Eqs.* (3.4) *and* (3.7) *for a convex loss function ϕ that is L_ϕ–Lipschitz. Under the assumptions of Lemma 2,*

$$\sup_{\mathbf{w} \in \mathcal{W}} \left|\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, T)\right| = O_P\left(\left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}} + n_1^{-1/2}\right). \qquad (3.9)$$

We use this uniform rate to prove the following result, which states that the risk of our classification procedure is close to that of the optimal SVM classifier.

Corollary 4. *Let* $\widehat{\mathbf{w}}$ *and* $\mathbf{w}^*$ *be as defined in Eqs.* (3.6) *and* (3.8) *for a fixed set* $\mathcal{W}$ *and convex loss function* ϕ *that is* L_ϕ*–Lipschitz. Under the assumptions of Lemma 2, the risk of our estimated classifier converges to its minimum in the set* $\mathbf{w} \in \mathcal{W}$ *at a rate,*

$$R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(\mathbf{w}^*, T) = O_P\left(\left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}} + n_1^{-1/2}\right). \tag{3.10}$$

We now turn to the convergence $\widehat{\mathbf{w}} \to \mathbf{w}^*$. If the loss function ϕ is strictly convex, then $\widehat{\mathbf{w}}$ is the minimizer of a strictly convex objective function, and is thus unique. In the case of the hinge loss function, which is not strictly convex, although any solution is a global minimizer, there may be more than one solution. Burges and Crisp (2000) studied the cases where the linear SVM solution is not unique. In particular, they showed that all solutions share the same vector $\mathbf{w}$ of weights. In these degenerate cases, the non-uniqueness is in the intercept term in the classifier.

Theorem 5. *Let* $\widehat{\mathbf{w}}$ *and* $\mathbf{w}^*$ *be as defined in Eqs.* (3.6) *and* (3.8) *for a fixed set* $\mathcal{W}$ *and convex loss function* ϕ *that is* L_ϕ*–Lipschitz. Under the assumptions of Lemma 2,*

$$\widehat{\mathbf{w}} \xrightarrow{P} \mathbf{w}^*.$$

3.2.3 Remarks on Bayes risk consistency

We have thus far compared the forest density classifier with the optimal linear SVM classifier. Specifically, we considered the effect of using a finite number of samples to construct the classifier. Here, we briefly dicuss the performance of a new classifier relative to the optimal Bayes classifier, as defined in Eq. (1.1), under the forest distribution assumption.

The expected misclassification error for some measurable function $g : \Omega \rightarrow \mathbb{R}$ is given by $R_{mis}(g) = \mathbb{E}\{\theta(g(\mathbf{X})Y)\}$, where θ is the $0-1$ loss function defined as

$$\theta(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ \\ 0 & \text{otherwise.} \end{cases}$$

The Bayes classifier can then be written as $f_B(\mathbf{x}) = sgn(g_B(\mathbf{x}))$, where

$$g_B(\mathbf{x}) = p(Y = +1|\mathbf{x}) - \frac{1}{2}$$

is the minimizer of $R_{mis}(g)$. It has been shown (Lin, 2001; Zhang, 2004) that the minimizer of the expected risk with respect to the hinge loss among all measurable functions is

$$g^*(\mathbf{x}) = sgn\left\{p(Y = +1|\mathbf{x}) - \frac{1}{2}\right\}.$$

The corresponding decision rule, $f^*(\mathbf{x}) = sgn(g^*(\mathbf{x}))$, is therefore equivalent to the Bayes decision rule and hence achieves the optimal Bayes risk.

The performance of our method is evaluated by comparing its expected misclassification error to that of the Bayes classifier. This excess risk under the $0-1$ loss function is bounded above by the excess risk with respect to the hinge loss ϕ (e.g., Bartlett, Jordan, and McAuliffe, 2006). That is,

$$R_{mis}(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R_{mis}(g_B) \leq R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(g^*).$$

Hence, in order to show Bayes consistency, it is sufficient to show that the excess risk in terms of hinge loss goes to zero.

This expression can be decomposed into the estimation error and the approximation error,

$$R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(g^*) = \big(R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(\mathbf{w}^*, T)\big) + \big(R(\mathbf{w}^*, T) - R(g^*)\big),$$

where $\mathbf{w}^*$ is the optimal linear SVM defined in Eq.(3.8). This decomposition reflects the variance-bias tradeoff that we see in all nonparametric estimation. The convergence of the estimation error, or the effect of using a finite number of samples to construct the classifier, was shown in Corollary 4.

The approximation error measures the error of the best linear SVM classifier constructed using infinite samples. Ensuring that the approximation error is small requires that the class of functions considered for the classifier be sufficiently complex, and bounding this error generally requires making further assumptions that are specific to the distribution (e.g. Steinwart and Scovel (2007) in the case where the classification is done by SVM using radial basis functions). We do not attempt to do so, but instead emphasize once again that if indeed each class follows a forest structured distribution, the optimal Bayes classifier is linear in the transformed variables $T(\mathbf{x})$ and so the optimal classifier is contained in the class of functions considered by linear SVM.

4 RESULTS

In this section we use a simulation study and several real data analyses to illustrate the performance of our proposed forest density classifier. In particular, we compare its predictive accuracy with popular methods to better understand the utility and flexibility of the forest structure assumption for classification.

Our proposed method (denoted as `forestdens`) trains a linear support vector machine classifier in the transformed space of univariate and bivariate densities. The univariate and bivariate densities are estimated using Matlab's kernel smoothing function `ksdensity` with a Gaussian kernel and bandwidths which are optimal for Gaussian data. For the bivariate densities, this function uses a Gaussian product kernel.

Our method, like many others, may be tuned to improve performance on a specific domain by considering such things as distinct costs of misclassification and prior probabilities that are different from the relative class frequencies in the training data. These parameters can be easily specified when implementing SVM, given further knowledge about each dataset. However, we refrain from optimizing each classifier on every domain, and instead aim to demonstrate the validity of our approach and its competitiveness across a broad range of problems.

4.1 Simulation study in $d = 20$ dimensions

Consider the binary classification problem in $d = 20$ dimensions. We study the predictive performance of our forest density classification method on simulated datasets that satisfy the assumption that each class has a forest-structured distribution.

The following experiment was designed to study the classifier's predictive accuracy under a variety of forest-distributed models. We consider the following factors in a 4×2^4 factorial design:

- Sample size of training data, $n =$100, 200, 400, 800.
- Class priors. The problem is either balanced (Bal), with half of the training samples belonging to each class, or unbalanced (Unb), with 25% of the training samples belonging to the minority class.
- Sparsity of the forest structure, measured as the number of edges in the two forests. The structures are either fully connected trees with $d-1$ edges or sparser forests having roughly two-thirds as many edges.
- Complexity of the marginal distributions. In the simple case, the joint distribution for each forest is multivariate normal. In the more complicated setting, the marginals $p(x_i)$ and $p(x_i|x_j)$ are either t-distributed with 3 degrees of freedom, or come from a mixture of two normal distributions having different means and variances.
- Common structure between the two classes. In the first case, the two forests are constructed independently of each other without constraining the forest structures to be similar. In the second case, roughly two-thirds of the structural features (edges

and isolated nodes) are identical for the two classes. Figure 4.1 displays an example of a model in which two forest distributions were generated with similar dependency structures.

The levels of these factors were chosen to represent simpler and more challenging classification scenarios, respectively. Although the difficulty of the classification problem cannot be fully characterized by these factors, they are common challenges that help to illustrate the behavior of the forest density classifier.

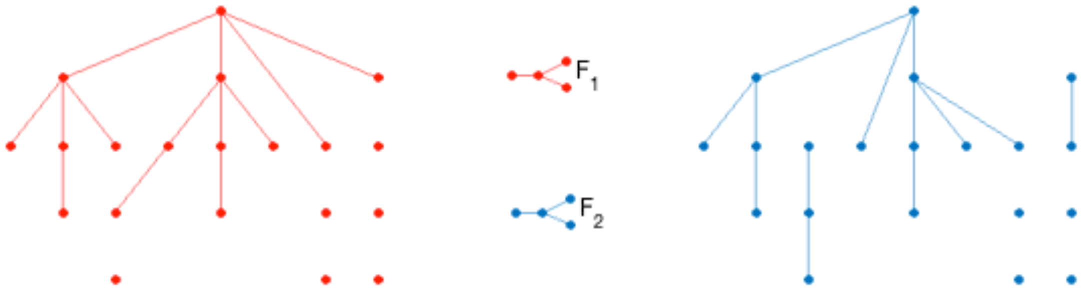

Figure 4.1: Example of two forests with similar dependency structures.

In the following experiments, we compare the forest density classifier with a diverse set of popular learning methods. We consider Linear Discriminant Analysis (LDA), the 5-nearest-neighbor classifier (5NN), naïve Bayes (NB), kernel support vector machines (SVM), and a random forest (RF) classifier. These methods were implemented using the built-in functions in Matlab, with parameters generally set to their default specifications.

Naïve Bayes was implemented using kernel smoothing density estimates with a Gaussian kernel. For each predictor and class, the bandwidth is automatically chosen to be a value that is optimal for a Gaussian distribution. Kernel SVM was performed using radial basis functions since they are frequently chosen for the kernel function in the absence of prior knowledge of what would be a good separator between the classes. The scale value for the kernel function is automatically chosen using a heuristic procedure as implemented by the Matlab function `fitcsvm`. Random forest was implemented using Matlab's bagged decision tree function `TreeBagger`, with the number of trees set to 50. The number of variables randomly selected for each split is set to the default value of the square root of the total number of variables.

For each combination of factor levels, $s = 100$ forest-distributed models were generated. A training dataset drawn from each model was used to construct the different classifiers, and the predictive accuracy of these classifiers was evaluated on an independent test set of 1000 samples generated from the same forest-distributed models. In these test sets, half of the observations were generated from each class. Average misclassification error rates from the s replicates are illustrated in Figure 4.2. The full results, including standard deviations, are displayed in Tables 4.1– 4.4. The error rates are listed as percentages for easier reading, and for each experiment the classifiers with best performance according to a Wilcoxon signed-rank test at the $\alpha = 0.0001$ level are printed in boldface. LDA performed no better than a random guess because of the highly nonlinear separating boundary in each model, and has therefore been omitted from these tables.

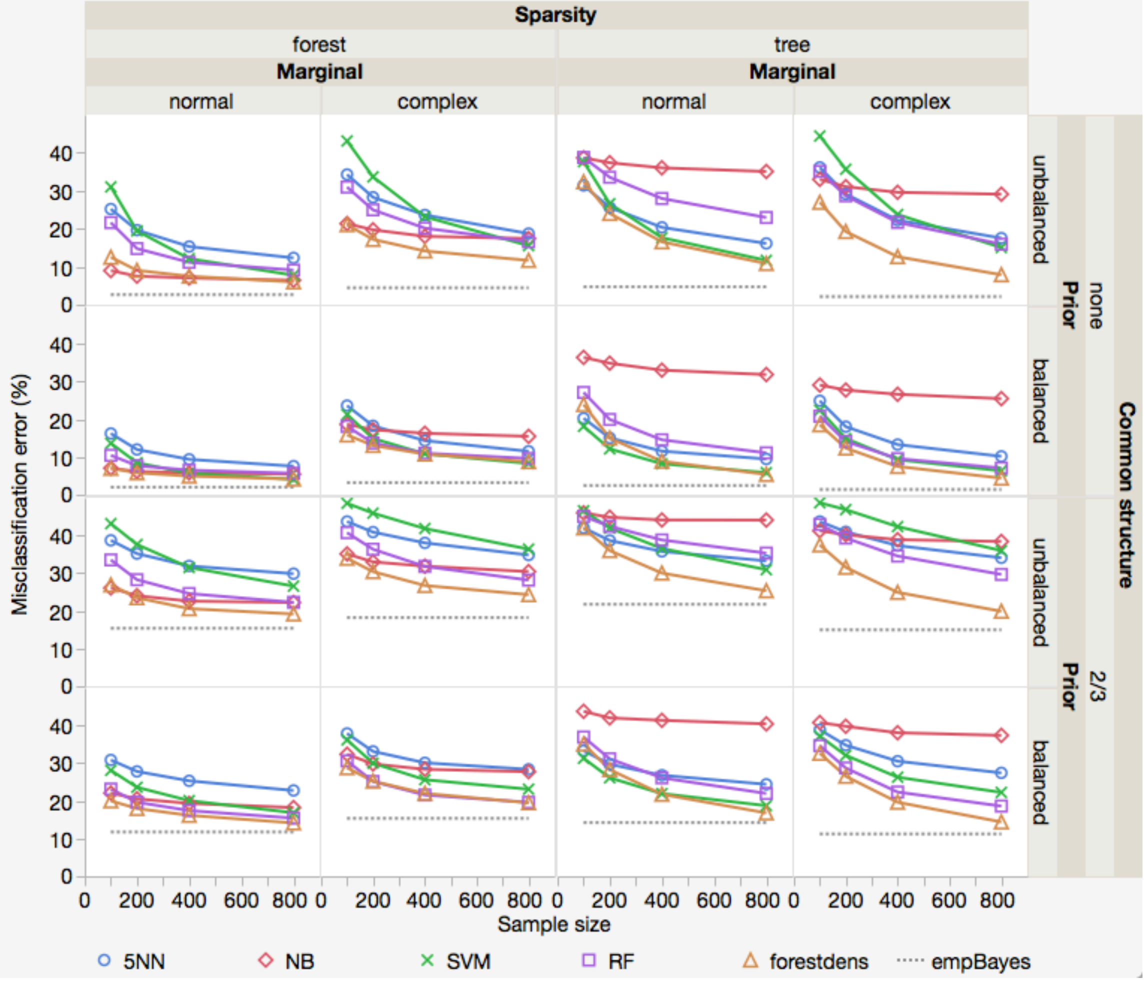

Figure 4.2: Summary of simulation study results. Misclassification error rates are plotted for 5NN, NB, SVM, RF, and the forest density classifier averaged across the 100 replicate datasets generated at each factor level combination. The misclassification error rate of the likelihood ratio classifier given knowledge of the true forest structures (empBayes) is plotted for reference.

Common structure	Priors	n	5-NN	NB	SVM(rbf)	RF	forestdens
None	Unb	100	25.2 ± 6.3	**9.0 ± 4.3**	31.0 ± 6.5	21.6 ± 6.2	12.5 ± 6.0
		200	19.6 ± 5.9	**7.5 ± 3.3**	19.4 ± 5.4	14.8 ± 4.5	9.1 ± 4.1
		400	15.3 ± 5.5	**7.0 ± 3.2**	12.1 ± 3.7	11.2 ± 3.9	**7.5 ± 3.2**
		800	12.3 ± 4.8	**6.4 ± 3.1**	7.8 ± 3.0	9.1 ± 2.9	**6.0 ± 2.7**
	Bal	100	16.2 ± 4.3	**7.1 ± 3.6**	13.6 ± 3.2	10.5 ± 3.5	**7.0 ± 2.9**
		200	12.0 ± 3.5	**6.2 ± 3.1**	8.5 ± 2.3	7.7 ± 2.7	**5.8 ± 2.5**
		400	9.4 ± 2.8	5.8 ± 2.8	5.7 ± 1.7	6.5 ± 2.2	**5.0 ± 1.9**
		800	7.6 ± 2.6	5.5 ± 2.5	**4.1 ± 1.4**	5.7 ± 1.9	**4.2 ± 1.6**
2/3	Unb	100	38.7 ± 6.4	**26.1 ± 9.9**	43.1 ± 6.3	33.5 ± 8.8	**26.9 ± 9.3**
		200	35.2 ± 6.7	**24.0 ± 10.2**	37.6 ± 7.8	28.3 ± 9.4	**23.5 ± 9.6**
		400	31.9 ± 7.9	22.6 ± 10.3	31.5 ± 9.3	24.6 ± 9.7	**20.6 ± 9.4**
		800	29.9 ± 8.2	22.2 ± 10.2	26.6 ± 9.4	22.2 ± 9.2	**19.2 ± 8.8**
	Bal	100	30.8 ± 6.4	22.1 ± 8.5	28.1 ± 7.2	23.1 ± 7.5	**20.0 ± 7.8**
		200	27.8 ± 7.1	20.5 ± 8.6	23.6 ± 7.4	19.7 ± 7.6	**18.0 ± 7.8**
		400	25.3 ± 7.4	19.3 ± 8.4	20.1 ± 7.2	17.4 ± 6.8	**16.2 ± 7.1**
		800	22.8 ± 7.0	18.2 ± 7.6	16.8 ± 6.7	15.4 ± 6.3	**14.1 ± 6.3**

Table 4.1: Simulation results for *sparse* forest distributions with *normal* marginals.

Common structure	Priors	*n*	5-NN	NB	SVM(rbf)	RF	forestdens
None	Unb	100	**31.4 ± 3.6**	38.6 ± 4.1	37.64 ± 4.1	38.75 ± 3.6	**32.32 ± 4.4**
		200	25.6 ± 3.6	37.4 ± 3.9	26.58 ± 3.7	33.54 ± 3.2	**24.01 ± 3.2**
		400	20.4 ± 2.6	36.0 ± 4.1	17.64 ± 2.3	27.95 ± 2.7	**16.55 ± 2.4**
		800	16.1 ± 2.4	35.1 ± 3.7	11.67 ± 2.0	22.95 ± 2.3	**10.83 ± 1.8**
	Bal	100	20.3 ± 3.0	36.4 ± 5.2	**18.14 ± 3.0**	27.13 ± 3.3	23.93 ± 4.2
		200	15.1 ± 2.4	34.9 ± 4.6	**12.17 ± 2.2**	19.97 ± 2.7	15.02 ± 2.7
		400	11.6 ± 2.0	33.0 ± 4.5	**8.22 ± 1.7**	14.57 ± 2.1	8.86 ± 2.0
		800	9.5 ± 1.8	31.9 ± 4.2	5.85 ± 1.3	11.10 ± 1.8	**5.39 ± 1.2**
2/3	Unb	100	**42.0 ± 4.6**	46.0 ± 3.4	46.5 ± 3.5	45.1 ± 3.4	**42.0 ± 4.4**
		200	38.7 ± 5.4	44.8 ± 4.1	41.9 ± 5.9	42.4 ± 4.4	**36.0 ± 5.6**
		400	35.8 ± 6.1	44.1 ± 4.1	36.6 ± 7.4	38.8 ± 5.1	**30.0 ± 6.3**
		800	33.3 ± 6.7	44.1 ± 4.2	30.9 ± 8.4	35.3 ± 5.9	**25.3 ± 7.3**
	Bal	100	33.3 ± 5.4	43.7 ± 4.3	**31.3 ± 5.7**	36.9 ± 4.6	35.1 ± 5.2
		200	29.7 ± 5.6	42.0 ± 4.8	**26.1 ± 6.0**	31.2 ± 5.4	28.3 ± 5.8
		400	26.8 ± 6.1	41.4 ± 4.3	**21.9 ± 6.2**	26.1 ± 5.7	**21.7 ± 6.0**
		800	24.4 ± 6.3	40.4 ± 4.8	18.7 ± 6.5	22.0 ± 5.8	**16.8 ± 6.3**

Table 4.2: Simulation results for *tree* distributions with *normal* marginals.

Common structure	Priors	n	5-NN	NB	SVM(rbf)	RF	forestdens
None	Unb	100	34.3 ± 7.0	**21.3 ± 4.6**	43.1 ± 5.0	31.0 ± 4.9	**21.1 ± 5.4**
		200	28.3 ± 7.0	19.7 ± 4.1	33.6 ± 6.3	25.0 ± 4.7	**17.2 ± 4.2**
		400	23.6 ± 6.6	18.0 ± 3.7	23.2 ± 5.3	20.2 ± 3.5	**14.1 ± 3.2**
		800	18.7 ± 6.0	17.4 ± 4.1	15.5 ± 4.2	16.6 ± 3.2	**11.7 ± 2.7**
	Bal	100	23.6 ± 5.5	18.7 ± 4.6	21.3 ± 4.9	18.1 ± 4.5	**15.9 ± 4.2**
		200	18.3 ± 4.3	17.2 ± 4.2	14.9 ± 3.4	**13.7 ± 3.1**	**13.1 ± 3.4**
		400	14.4 ± 4.1	16.3 ± 3.9	**10.9 ± 2.9**	**11.0 ± 2.5**	**10.8 ± 2.6**
		800	11.5 ± 3.4	15.5 ± 3.7	**8.3 ± 2.5**	9.6 ± 2.2	**8.8 ± 2.2**
2/3	Unb	100	43.7 ± 3.9	35.1 ± 5.5	48.5 ± 2.3	40.7 ± 5.0	**33.9 ± 6.2**
		200	40.9 ± 5.1	33.0 ± 6.5	45.9 ± 3.8	36.3 ± 6.0	**30.4 ± 7.0**
		400	38.0 ± 5.9	31.8 ± 6.7	41.8 ± 6.1	31.8 ± 7.0	**26.7 ± 7.0**
		800	34.8 ± 6.4	30.4 ± 6.6	36.4 ± 7.3	28.2 ± 6.9	**24.3 ± 7.0**
	Bal	100	37.9 ± 6.1	32.4 ± 7.3	36.2 ± 6.4	30.6 ± 7.1	**28.8 ± 7.3**
		200	33.1 ± 6.5	29.8 ± 7.2	30.0 ± 7.1	**25.1 ± 6.9**	**25.2 ± 7.1**
		400	30.1 ± 7.2	28.4 ± 7.2	25.6 ± 7.5	**21.6 ± 6.2**	**22.0 ± 6.8**
		800	28.4 ± 7.7	27.7 ± 7.4	23.1 ± 7.9	**19.6 ± 6.3**	**19.5 ± 6.7**

Table 4.3: Simulation results for *sparse* forest distributions with *complex* marginals.

Common structure	Priors	n	5-NN	NB	SVM(rbf)	RF	forestdens
None	Unb	100	36.1 ± 6.9	33.0 ± 5.0	44.4 ± 4.0	35.1 ± 4.1	**26.9 ± 5.8**
		200	29.1 ± 7.1	31.1 ± 4.7	35.7 ± 5.9	28.6 ± 4.8	**19.3 ± 4.3**
		400	22.2 ± 6.7	29.6 ± 4.1	23.8 ± 6.5	21.7 ± 3.7	**12.6 ± 3.1**
		800	17.5 ± 6.2	29.1 ± 4.5	15.0 ± 4.9	15.9 ± 3.3	**7.9 ± 2.0**
	Bal	100	24.9 ± 4.9	29.1 ± 4.6	22.4 ± 4.7	20.8 ± 3.8	**18.6 ± 3.6**
		200	18.1 ± 4.0	27.8 ± 5.1	14.9 ± 3.8	14.1 ± 3.2	**12.4 ± 2.7**
		400	13.3 ± 4.3	26.6 ± 4.9	9.3 ± 3.2	9.6 ± 2.3	**7.5 ± 1.9**
		800	10.2 ± 3.1	25.5 ± 4.1	6.4 ± 2.1	7.1 ± 1.9	**4.4 ± 1.3**
2/3	Unb	100	43.7 ± 3.7	41.2 ± 4.6	48.7 ± 1.8	42.9 ± 4.0	**37.5 ± 6.1**
		200	41.0 ± 4.7	40.2 ± 4.7	46.8 ± 3.0	39.3 ± 4.9	**31.5 ± 5.7**
		400	37.3 ± 5.7	38.9 ± 5.3	42.3 ± 5.4	34.5 ± 5.7	**24.9 ± 5.9**
		800	34.1 ± 6.3	38.4 ± 5.3	36.0 ± 7.5	29.7 ± 6.2	**19.9 ± 5.7**
	Bal	100	38.8 ± 5.4	40.8 ± 5.8	37.2 ± 6.1	34.6 ± 6.1	**32.6 ± 6.4**
		200	34.8 ± 6.4	39.8 ± 5.5	32.1 ± 7.5	28.7 ± 6.7	**26.5 ± 6.7**
		400	30.5 ± 6.8	38.1 ± 6.1	26.3 ± 7.5	22.3 ± 6.8	**19.6 ± 6.0**
		800	27.4 ± 6.9	37.4 ± 6.2	22.2 ± 7.6	18.6 ± 6.1	**14.4 ± 5.6**

Table 4.4: Simulation results for *tree* distributions with *complex* marginals.

The models generated in this study represent a wide variety of forest-structured distributions. The independent generation of new forest structures in each simulation run allowed for more variation in the datasets that were tested. The results as a whole demonstrate that our proposed classifier performs very well when the forest density assumption holds. We now call particular attention to a few of the interesting features in the comparisons between the forest density classifier and its competitors.

As expected, naïve Bayes performed particularly well when the forest structure was sparse with marginal Gaussian densities (Table 4.1), since the sparse forest structure is conceptually close to its assumption of independence. In the first of these models, where the forest distributions for the different classes were independently constructed, the performance of the forest density classifier was second only to naïve Bayes and performed just as well when the sample size increased. However, in the models where a portion of the two forest graphs were constrained to be identical, and hence the corresponding marginals did not aid in discriminating between the two classes, the forest density classifier outperformed the generative naïve Bayes classifier even for small sample sizes. This outcome reflects a key advantage of the forest density classifier compared to generative methods such as naïve Bayes. In these models with overlapping class structures, generative methods that separately estimate the two distributions may blur the important differences between the two classes that may aid in classification.

Of particular interest is the performance of the forest density classifier as compared to that of kernel SVM, since both methods perform nonlinear classification by learning a linear classifier in a high-dimensional feature space. As seen with the models in Table 4.2 having tree distributions with normal marginals, kernel SVM performed well when the classes were balanced, but the forest density classifier's performance was just as accurate for sample sizes of 400 and 800. Furthermore, the forest density classifier performed notably better in the presence of class imbalance.

Tables 4.3 and 4.4 show that the forest density classifier generally outperformed the other approaches, including random forest, in the cases where the data were generated from tree and forest distributions having marginals that were mixtures of Gaussians and *t*-distributions. Even at small sample sizes, the forest density classifier was competitive with the best performing classifier.

A 5-way analysis of variance for the forest density classifier was performed to study the behavior of the forest density classifier under different factor combinations. The ANOVA table and results of the lack of fit test are displayed in Table 4.5, and the effects tests for a full factorial analysis are listed in Table 4.6.

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Model	31	16.2396	0.5239	171.2057	$< .0001*$
Error	1888	5.7769	0.0031		
Total	1919	22.0166			
Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Lack Of Fit	32	0.2927	0.0091	3.0955	$< .0001*$
Pure Error	1856	5.4842	0.0030		
Total Error	1888	5.7769			

Table 4.5: ANOVA table and lack of fit test for the forest density classifier on the simulated data.

Source	SS	F ratio	Prob > F	
Prior	1.6420366	536.6442	< 0.0001	*
Sparsity	0.9071059	296.4569	< 0.0001	*
Prior*Sparsity	0.1297412	42.4015	< 0.0001	*
Common structure	7.2332893	2363.956	< 0.0001	*
Prior*Common structure	0.0128185	4.1893	0.0408	*
Sparsity*Common structure	0.0001894	0.0619	0.8036	
Prior*Sparsity*Common structure	0.0007563	0.2472	0.6191	
Marginal	0.1541371	50.3745	< 0.0001	*
Prior*Marginal	0.0009227	0.3015	0.5830	
Sparsity*Marginal	1.2947499	423.1453	< 0.0001	*
Prior*Sparsity*Marginal	0.0405261	13.2446	0.0003	*
Common structure*Marginal	0.0036383	1.1890	0.2757	
Prior*Common structure*Marginal	0.0016558	0.5411	0.4621	
Sparsity*Common structure*Marginal	0.0008387	0.2741	0.6006	
Prior*Sparsity*Common structure*Marginal	0.0078773	2.5744	0.1088	
$\sqrt{n}$	3.9773975	1299.878	< 0.0001	*
Prior*$\sqrt{n}$	0.0138639	4.5310	0.0334	*
Sparsity*$\sqrt{n}$	0.6833792	223.3394	< 0.0001	*
Prior*Sparsity*$\sqrt{n}$	0.0001158	0.0378	0.8458	
Common structure*$\sqrt{n}$	0.0059073	1.9306	0.1649	
Prior*Common structure*$\sqrt{n}$	0.0249595	8.1572	0.0043	*
Sparsity*Common structure*$\sqrt{n}$	0.0065234	2.1320	0.1444	
Prior*Sparsity*Common structure*$\sqrt{n}$	0.0093229	3.0469	0.0811	
Marginal*$\sqrt{n}$	0.0100945	3.2991	0.0695	
Prior*Marginal*$\sqrt{n}$	0.0012831	0.4193	0.5173	
Sparsity*Marginal*$\sqrt{n}$	0.0497693	16.2654	< 0.0001	*
Prior*Sparsity*Marginal*$\sqrt{n}$	0.0036848	1.2042	0.2726	
Common structure*Marginal*$\sqrt{n}$	0.0078359	2.5609	0.1097	
Prior*Common structure*Marginal*$\sqrt{n}$	0.0000066	0.0022	0.9630	
Sparsity*Common structure*Marginal*$\sqrt{n}$	0.0152068	4.9698	0.0259	*
Prior*Sparsity*Common structure*Marginal*$\sqrt{n}$	0.0000010	0.0003	0.9855	

Table 4.6: ANOVA effects tests for a full factorial analysis of the forest density classifier on the simulated data. The columns give the source of variation, corresponding sum of squares, F statistic and p-value for testing the significance of the effect.

Closer inspection of interaction plots corresponding to significant interaction effects in Table 4.6 showed that the performance of the forest density classifier was better in the balanced case than the unbalanced case across the levels of priors, sparsity, and marginal distributions. Following our intuition, the classifier also performed better when the two class distributions were generated separately than when their structures were similar, across different levels of the other factors.

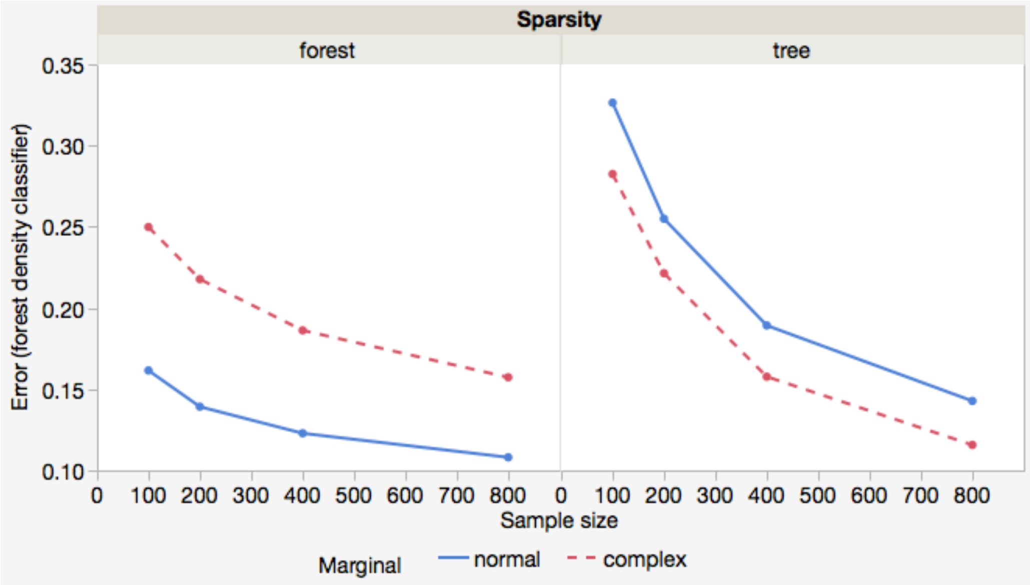

Figure 4.3: 3-way interaction plot for the simulation study. Average misclassification error rates for the forest density classifier, plotted to show the interaction between the levels of sparsity, marginal distributions, and sample size.

The effect of sparsity on the error rate was discovered to be less straightforward. The interaction plots in Figure 4.3 show that the average error rate in the tree models, unlike the sparser forest models, is higher for the normally distributed cases than for the cases with bivariate marginals which are mixtures of normal distributions having different variances. This suggests that although the more complex distributions may be harder to estimate, their distinct features may have provided more useful information in discriminating between the two classes for the forest density classifier.

Pairwise comparisons were made separately for each of the 64 factor level combinations to compare the performance of the forest density classifier with each of the four competitor classifier. Note that the forest density classifier and the competitor classifiers are trained and evaluated on the same training and test datasets for each simulation run, and hence their respective error rates are paired observations. For each comparison, a one-sided Wilcoxon signed-rank test was used to test the null hypothesis that the median difference of the misclassification error rates for the forest density classifier and the competitor classifier is greater than zero against the alternative that it is less than zero. Significance was tested at the $\alpha = 0.0002$ level, using the Bonferroni method to correct for the problem of multiplicity. Figure 4.4 displays, separately for each sample size, the proportion of the models for which the test rejected the null hypothesis, i.e. there was enough evidence to conclude that the misclassification error for the forest density classifier was smaller than that of its competitor classifier. The detailed results of these tests, including the test statistics and p-values for each model, are included in Appendix B.

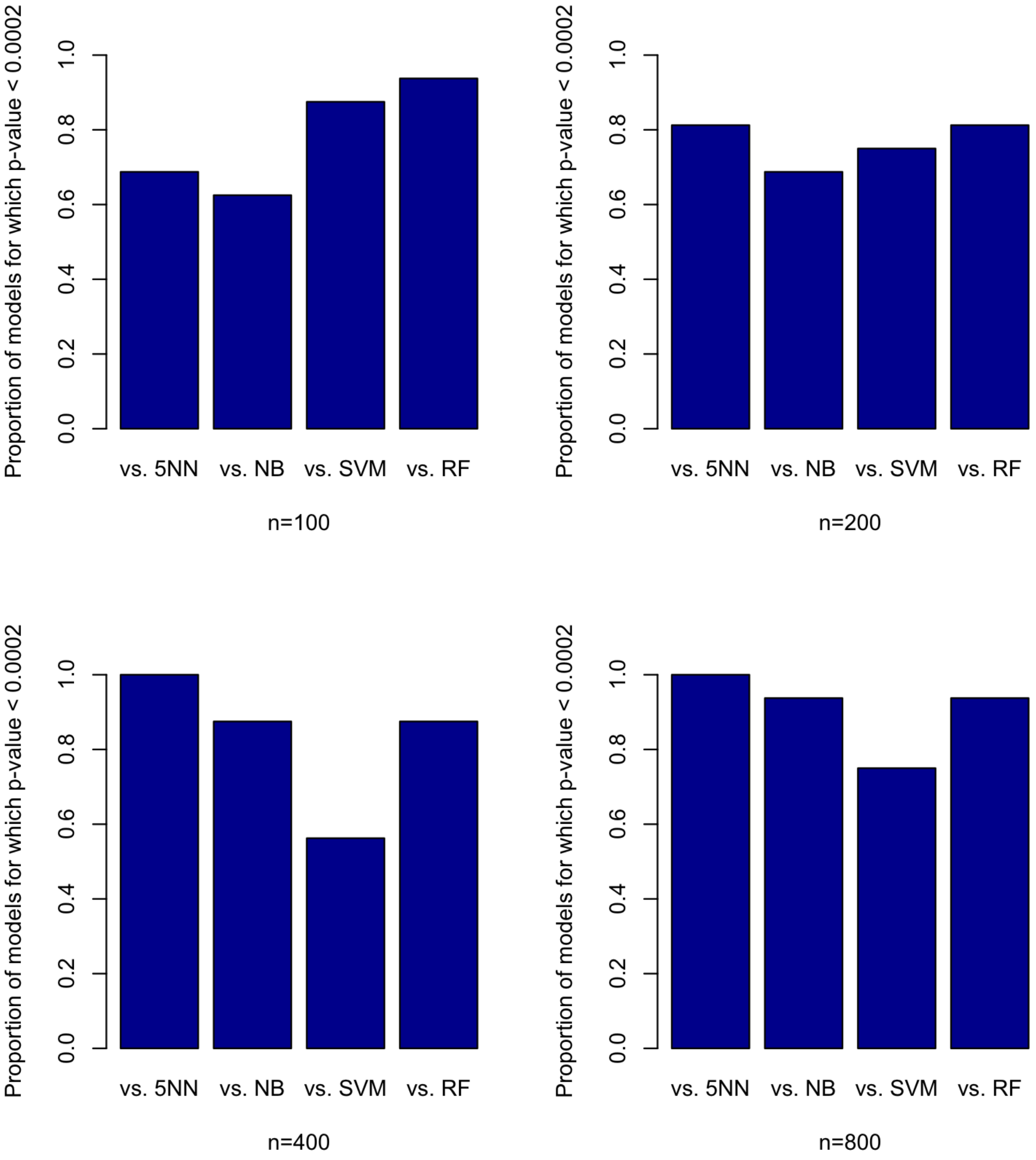

Figure 4.4: Summary of pairwise comparison tests for the simulation study. The proportion of the simulation models for which a Wilcoxon signed-rank test found the median performance of the forest density classifier to be significantly better than that of 5NN, NB, SVM, and RF, is plotted separately for each sample size.

4.2 Benchmark datasets

We applied our proposed method to publicly available binary classification datasets from different domains in order to demonstrate its potential when the forest assumption may not hold. Table 4.7 summarizes the datasets used in this analysis, all of which are available from the UCI Machine Learning Database (Lichman, 2003). The datasets selected have mostly continuous predictor variables, but range in sample size and dimensionality. Since we do not deal with missing data in this paper, we removed instances with missing values from the datasets. In particular, 35 samples were removed from the Pima diabetes dataset that were miscoded in one variable. In the Wisconsin breast (prognostic) dataset, which was used for the problem of predict cancer recurrence within 24 months, 4 samples with missing values were removed.

In the Ionosphere dataset, two ordinal variables are treated as categorical when constructing the NB classifier, so that the corresponding marginals are estimated as a multinomial distribution instead of a Gaussian distribution. Where features are not commensurate, such as the Pima diabetes dataset in which one feature represents diastolic blood pressure and another is body mass index, they were standardized prior to applying kNN and SVM, both of which are sensitive to the scale of the data. Our proposed approach does not require such preprocessing.

The misclassification error was estimated by 10-fold cross validation, with the folds sampled in a stratified manner so that they have approximately the same proportions of class labels as the full dataset. The cross validation was repeated 10 times to account for

Dataset	d	n	(n_1, n_2)
Ionosphere	34	351	(225, 126)
Liver disorder	6	345	(145, 200)
Ozone	72	1,847	(128, 1,719)
Parkinsons	22	195	(147, 48)
Pima diabetes	8	733	(252, 481)
Sonar	60	208	(111, 97)
SPECT heart	44	267	(55, 212)
Vertebral column	6	310	(210, 100)
WI breast (prognostic)	32	194	(28, 166)
WI breast (diagnostic)	30	569	(212, 357)

Table 4.7: Summary of benchmark datasets. The datasets are from the UCI ML Repository. d is the number of predictor variables, n is the total number of available training samples, and n_1 and n_2 are the number of samples in the 'positive' and 'negative' classes, respectively.

the variance in the error estimates due to taking a different random partition of the data. For each dataset, the various classifiers were learned on the same training sets and their performance evaluated on the same test sets. In particular, the cross-validation folds were the same for all the experiments on each dataset.

For datasets having some degree of class imbalance, simple accuracy (or the misclassification error rate) gives an incomplete picture of classifier performance. For example, SVM often produces models which are biased towards the majority class. This can be corrected in a number of ways, including by tuning the costs of making mistakes in each class or by oversampling the minority class to create a balanced training dataset. As our interest is not focused on solving the issues raised by class imbalance, we instead measured performance using the balanced error rate, defined as

$$BER = 1 - \frac{1}{2}(Sensitivity + Specificity).$$

Sensitivity is the true positive rate, or the proportion of predicted positives that are actually positive. Specificity is the true negative rate, or the proportion of predicted negatives that are actually negative. Note that the misclassification error rates reported for the simulated datasets in the previous subsection were estimated on a test set that had an equal number of samples belonging to each class, and hence are equivalent to measuring the BER.

The empirical mean and standard deviation of the BER taken across the ten runs and cross validation folds are given in Table 4.8. For each dataset, the classifier having the smallest balanced error rate is underline, and classifiers which are not significantly different from it according to a two-sided paired t-test at the Bonferroni-corrected $\alpha = 0.001$ level are reported in bold text. In half of the cases, more than four classifiers were found not to be significantly different from each other and none were bolded. The tests were performed with means and variances estimated using the 100 individual accuracies (from the 10 repeated 10-fold cross validations) and with 10 degrees of freedom.

The performance of the forest density classifier was better than SVM in six of the ten domains, including the two most highly imbalanced datasets Ozone and SPECT, and outperformed naïve Bayes and Random Forest in seven of the ten domains. Only for the Parkinsons dataset did the forest density classifier perform worse than all of the other classifiers tested here, and even then its performance was not significantly worse than the other

Dataset	*d*	*n*	LDA	5NN	NB	SVM	RF	forestdens
Ionosphere	34	351	18.3 ± 6.8	21.2 ± 7.1	25.2 ± 6.9	**5.5 ± 4.4**	**7.7 ± 5.2**	**7.1 ± 4.8**
Liver disorder	6	345	**37.2 ± 7.3**	40.0 ± 7.4	41.0 ± 8.1	**37.3 ± 7.2**	**30.0 ± 7.5**	**32.2 ± 7.5**
Ozone	72	1,847	41.7 ± 5.6	40.2 ± 5.1	**25.0 ± 5.0**	43.1 ± 4.7	41.2 ± 5.1	38.8 ± 5.3
Parkinsons	22	195	19.5 ± 10.8	12.1 ± 10.1	23.2 ± 9.3	21.3 ± 10.6	15.4 ± 9.7	26.7 ± 10.7
Pima diabetes	8	733	28.1 ± 5.6	30.6 ± 4.8	28.6 ± 5.5	31.5 ± 4.6	27.9 ± 5.0	27.7 ± 5.0
Sonar	60	208	25.2 ± 9.4	**18.5 ± 8.1**	31.3 ± 9.4	**15.3 ± 7.3**	**17.5 ± 8.3**	**17.2 ± 8.3**
SPECT heart	44	267	44.5 ± 8.6	44.4 ± 10.7	**23.6 ± 7.6**	43.3 ± 8.8	**38.8 ± 9.7**	**37.6 ± 9.8**
Vertebral column	6	310	19.3 ± 6.7	21.5 ± 7.5	19.8 ± 6.7	19.5 ± 7.8	19.2 ± 7.8	18.8 ± 7.1
WI breast (diag)	32	198	5.5 ± 3.4	4.0 ± 2.5	7.3 ± 3.3	2.9 ± 2.3	4.5 ± 2.9	6.4 ± 3.1
WI breast (prog)	30	569	40.7 ± 13.8	39.7 ± 12.9	33.3 ± 14.7	50.0 ± 0.3	45.1 ± 8.4	44.1 ± 11.1

Table 4.8: Results comparing six classifiers on benchmark datasets. The average balanced error rates and standard errors are displayed for each classifier and dataset.

methods. Although there are clear winners in a couple of instances, such as naïve Bayes on the two most highly imbalanced datasets Ozone and SPECT, these results are highly dependent on both the choice of performance measure and its method of estimation. This experiment demonstrates that the forest density classifier has the flexibility to handle datasets which do not satisfy the forest density assumption, and has the potential to excel if the forest density assumption is reasonable. While there is not one method that is the best in every case, the forest density classifier appears to be competitive with state-of-the-art methods such as kernel SVM and Random Forest on various domains.

5 CONCLUSIONS

In this work, we proposed a semiparametric classification procedure that makes a flexible assumption about the dependency structure of the data. In particular, we assumed that the distribution of data in each class $p(\mathbf{x}|y)$ follows a forest-structured distribution. This enabled us to overcome the main issue with nonparametric density estimation of having enough samples, since our approach estimates only univariate and bivariate densities, which can then be used as inputs into any standard classifier.

Empirical results on simulated and real datasets demonstrated that the forest density classifier performs as wells as popular nonparametric classifiers on a broad set of examples. Experiments on forest-structured data validated the potential of the proposed method even when sample sizes are small, distributions are highly non-Gaussian, classes are unbalanced, and many features are irrelevant to the discrimination task. Real data examples showed that the forest density assumption is flexible enough to be useful for medical, image, chemical, and signal processing applications.

5.1 Future work

The proposed procedure offers the opportunity to extend the basic algorithm in several ways. We conclude with remarks on potential directions for future work, particularly in high dimensional settings. Variable selection is an important consideration for this method when d is very large and the number of samples is relatively limited. Classification in high dimensions is intrinsically difficult due to noise accumulation and spurious collinearity among the predictors. The existence of many features that do not help to reduce the classification error is a challenge for the sparse forest structured distributions, where many of the bivariate probabilities in the transformed space will be zero. Furthermore, the proposed procedure incurs a heavy computational cost when working in high dimensions due to the estimation of an increasingly large number of low-dimensional densities. There is a vast literature on variable selection methods that alleviates the computational burden and improves classification performance in these challenging high-dimensional problems.

The forest density classifier allows variable selection to be incorporated in several ways. It can be performed on the original variables and in the transformed space variables. One approach is to test for independence among the predictor variables. For example, hypothesis tests may be performed to determine which edges should be included. The intuitive graphical representation underlying our new classification approach also offers the option to use any available data or domain specific knowledge about the dependency structure to guide the selection of log-transformed densities that are used in the classifier.

A second category of variable selection tests for the relevance of predictors to the response. This includes Sure Independence Screening (Fan and Lv, 2010), which ranks the marginal correlation of each variable with the response, assuming independence among the variables. The particular form of the transformed variables can also be used to determine which log densities are included in the classifier. For example, Tang et al. (2014) proposed an algorithm ranks variables according to the correlation coefficient between a univariate or bivariate density with its pair in the second class. We are currently investigating improvements in performance by incorporating variable selection into the transformed space of univariate and bivariate densities.

Another direction is to combine the variable selection and classification steps by using sparse classifiers (e.g. Clemmensen et al., 2011) and penalty-based methods such as the Lasso that may be tailored to the forest assumptions. Finally, there are Bayesian approaches that may be relevant to the forest density classification approach, such as variable selection using Markov random field priors (Stingo and Vannucci, 2011).

REFERENCES

Bartlett, P. L., Bousquet, O., and Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482.

Bickel, P. J. and Levina, E. (2004). Some theory for Fisher's linear discriminant function, 'naive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6):989–1010.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees: An Introduction*. Wadsworth, Monterey, CA.

Burges, C. and Crisp, D. (2000). Uniqueness of the SVM solution. *NIPS*, 12:223–229.

Cheng, B. and Titterington, D. M. (1994). Neural networks: A review from a statistical perspective. *Statistical Science*, 9(1):2–54.

Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467.

Chow, C. and Wagner, T. J. (1973). Consistency of an estimate of tree-dependent probability distributions. *IEEE Transactions on Information Theory*, 19:369–371.

Clemmensen, L., Hastie, T., Witten, D., and Ersbøll, B. (2011). Sparse discriminant analysis. *Technometrics*, 53(4):406–413.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *Proc. IEEE Transactions on Information Theory*, pages 21–27.

Fan, J., Feng, Y., Jiang, J., and Tong, X. (2016). Feature augmentation via nonparametrics and selection (fans) in high-dimensional classification. *Journal of the American Statistical Association*, 111(513):275–287.

Fan, J. and Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101–148.

Friedman, J. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifier. *Machine Learning*, 29:131–163.

Giné, E. and Guillou, A. (2002). Rates of strong uniform consistency for multivariate kernel density estimators. *Ann. I. H. Poincaré*, 38(6):907–921.

Guo, Y., Hastie, T., and Tibshirani, R. (2005). Regularized discriminant analysis and its application in microarrays. *Biostatistics*, 1(1):1–18.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proc. Eleventh Conf. on Uncertainty in Artificial Intelligence*, pages 338–345.

Jordan, M. I., editor (1999). *Learning in Graphical Models*. MIT Press, Cambridge, MA.

Jordan, M. I. (2004). Graphical models. *Statistical Science*, 19(1):140–155.

Lafferty, J., Liu, H., and Wasserman, L. (2012). Sparse nonparametric graphical models. *Statistical Science*, 27(4):519–537.

Lauritzen, S. L. (1996). *Graphical Models*, volume 17. Clarendon Press, Oxford.

Lee, Y., Lin, Y., and Wahaba, G. (2004). Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99:67–81.

Lichman, M. (2003). UCI machine learning repository. [http://archive.ics.uci.edu/ml] University of California, Irvine, School of Information and Computer Sciences.

Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L. (2012). High dimensional semiparametric Gaussian copula graphical models. *Annals of Statistics*, 40(4):2293–2326.

Liu, H., Xu, M., Gu, H., Gupta, A., Lafferty, J., and Wasserman, L. (2011). Forest density estimation. *Journal of Machine Learning Research*, 12:907–951.

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462.

Newey, W. K. (1991). Uniform convergence in probability and stochastic equicontinuity. *Econometrica*, 59(4):1161–1167.

Oneto, L., Ridella, S., and Anguita, D. (2015). Tikhonov, Ivanov and Morozov regularization for support vector machine learning. *Machine Learning*, 103(1):103–136.

Park, E. S., Spiegelman, C., and Ahn, J. (2011). A nonparametric approach based on a Markov like property for classification. *Chemometrics and Intelligent Laboratory Systems*, 108(2):87 – 92.

Pérez, A., Larrañaga, P., and Inza, I. (2009). Bayesian classifiers based on kernel density estimation: Flexible classifiers. *International Journal of Approximate Reasoning*, 50(2):341–362.

Rigollet, P. and Vert, R. (2009). Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15(4):1154–1178.

Ripley, B. D. (1994). Neural networks and related methods for classification. *Journal of the Royal Statistical Society Series B*, 56(3):409–456.

Steinwart, I. and Scovel, C. (2007). Fast rates for support vector machines using Gaussian kernels. *Annals of Statistics*, 35(2):575–607.

Stingo, F. and Vannucci, M. (2011). Variable selection for discriminant analysis with Markov random field priors for the analysis of microarray data. *Bioinformatics*, 27:495–501.

Tan, V., Anandkumar, A., and Willsky, A. (2011). Learning high-dimensional Markov forest distributions: Analysis of error rates. *Journal of Machine Learning Research*, 12:1617–1653.

Tan, V., Sanghavi, S., Fisher, J., and Willsky, A. (2010). Learning graphical models for hypothesis testing and classification. *IEEE Transactions on Signal Processing*, 58(11):5481–5495.

Tang, S., Chen, L., Tsui, K.-W., and Doksum, K. (2014). Nonparametric variable selection and classification: The CATCH algorithm. *Computational Statistics and Data Analysis*, 72:158–175.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer, New York.

Witten, D. and Tibshirani, R. (2011). Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society Series B*, 73(5):753–772.

Xue, L. and Zou, H. (2012). Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *Annals of Statistics*, 40(5):2541–2571.

APPENDIX A

PROOFS

A.1 Lemma 1

Proof. Let E_y and V_y denote the edge and vertex sets of the graphical model for class y. Then the corresponding forest density is

$$p_y(\mathbf{x}) = \prod_{(i,j)\in E_y} \frac{p_y(x_i,x_j)}{p_y(x_i)p_y(x_j)} \prod_{i\in V_y} p_y(x_i)$$

Then, the log-likelihood ratio for a 2-class problem is

$$\begin{aligned}
\log\frac{p_{+1}(\mathbf{x})}{p_{-1}(\mathbf{x})} =& \log\left\{\prod_{(i,j)\in E_1} \frac{p_{+1}(x_i,x_j)}{p_{+1}(x_i)p_{+1}(x_j)} \prod_{i\in V_1} p_{+1}(x_i)\right\} \\
& -\log\left\{\prod_{(i,j)\in E_2} \frac{p_{-1}(x_i,x_j)}{p_{-1}(x_i)p_{-1}(x_j)} \prod_{i\in V_2} p_{-1}(x_i)\right\} \\
=& \sum_{(i,j)\in E_1} \log p_{+1}(x_i,x_j) - \left\{\sum_{(i,j)\in E_1} \log p_{+1}(x_i)p_{+1}(x_j) - \sum_{i\in V_1} \log p_{+1}(x_i)\right\} \\
& - \sum_{(i,j)\in E_2} \log p_{-1}(x_i,x_j) + \left\{\sum_{(i,j)\in E_2} \log p_{-1}(x_i)p_{-1}(x_j) - \sum_{i\in V_2} \log p_{-1}(x_i)\right\}
\end{aligned}$$

The log-LR can thus be written as

$$\begin{aligned}
\log\frac{p_{+1}(\mathbf{x})}{p_{-1}(\mathbf{x})} =& \sum_{(i,j)\in E_1} \log p_{+1}(x_i,x_j) + \sum_{i\in V_1} C_{i,1}\cdot \log p_{+1}(x_i) \\
& - \sum_{(i,j)\in E_2} \log p_{-1}(x_i,x_j) - \sum_{i\in V_2} C_{i,2}\cdot \log p_{-1}(x_i),
\end{aligned}$$

where $C_{i,y} = deg_y(i) - 1$ is one less than the degree of node i in the forest of class y. That is, the likelihood ratio is linear in the new variables $\log p_y(x_i)$ and $\log p_y(x_i,x_j)$. □

A.2 Lemma 2

To prove Lemma 2, we need the following result, which provides the convergence rate for the kernel density estimates in the bivariate case where $\nu = 2$. The univariate rates can be found in the same manner. density estimates converge faster than the bivariate density estimates.

We denote the interior region of the support for the true density by $\Omega_{ij}^{o} = \{\mathbf{x} \in \Omega_{ij} : d(\mathbf{x}, \partial\Omega_{ij}) > \varepsilon\}$. The following consistency result holds uniformly for $\mathbf{x}$ in Ω_{ij}^{o}.

Lemma A.2.1. *Suppose that assumptions (D1), (K1)-(K3) hold. Then, with probability* $\geq 1 - \delta$*,*

$$\sup_{(x_i,x_j)\in\Omega_{ij}^{o}} \left|\widehat{p}_y(x_i,x_j) - p_y(x_i,x_j)\right| < C_\delta \left(\frac{\log n_y}{n_y}\right)^{\frac{\beta}{2+2\beta}}. \tag{A.1}$$

where $\widehat{p}_y$ *is the bivariate kernel density estimator defined in Eq.* (3.1).

Proof of Lemma A.2.1. By the triangle inequality,

$$|\widehat{p}_y(x_i,x_j) - p_y(x_i,x_j)| \leq |\widehat{p}_y(x_i,x_j) - \mathbb{E}\widehat{p}_y(x_i,x_j)| + |\mathbb{E}\widehat{p}_y(x_i,x_j) - p_y(x_i,x_j)|. \tag{A.2}$$

We begin by bounding the first summand. Corollary 2.2 in Giné and Guillou (2002) provided the following finite sample bound for the uniform consistency of the density estimate, given the kernel assumptions in (K2) and that

$$\sup_{\mathbf{t}\in\mathbb{R}^2} \sup_{h_{n_y}>0} \int K^2(\mathbf{t}-\mathbf{x}) p_y(\mathbf{x}) d\mathbf{x} \leq D < \infty.$$

This condition is satisfied under our kernel assumptions and the requirement that $p_y(x_i,x_j)$ is bounded. Then, there exist constants $c_1, c_2 > 0$ such that for sufficiently large n_y,

$$\Pr\left[\sup_{(x_i,x_j)\in\Omega_{ij}} |\widehat{p}_y(x_i,x_j) - \mathbb{E}\widehat{p}_y(x_i,x_j)| > \frac{\varepsilon}{2}\right] \leq c_1 \exp\left(-c_2 n_y h_{n_y}^2 \varepsilon^2\right) \tag{A.3}$$

for all ε satisfying

$$\varepsilon \geq c_3\sqrt{\frac{\log r_{n_y}}{n_y h_{n_y}^2}},$$

where $r_{n_y} \geq c_3 h_{n_y}^{-1}$ for some constant c_3. The constants c_1, c_2 depend only on the VC characteristics of the kernel, D, and $\|K\|_\infty$.

Next, we bound the second summand in Eq. (A.2). Under the assumptions (D1) and (K3), the bias of the estimated density is well known (see Tsybakov, 2009, Proposition 1.2) to satisfy

$$|\mathbb{E}\widehat{p}_y(x_i,x_j) - p_y(x_i,x_j)| \leq c_4 h_{n_y}^{\beta} \tag{A.4}$$

for all interior points $(x_i,x_j) \in \Omega_{ij}^{\mathrm{o}}$, where $c_4 = \frac{L}{\ell!}\int \|t\|^{\beta}|K(t)|dt$.

Then, combining (A.2), (A.3) and (A.4), we have that

$$\Pr\left[\sup_{(x_i,x_j)\in\Omega_{ij}^{\mathrm{o}}} \left|\widehat{p}(x_i,x_j) - p(x_i,x_j)\right| > \varepsilon\right] \leq c_1 \exp\left(-c_2 n_y h_{n_y}^2 \varepsilon^2\right)$$

for any ε satisfying

$$\varepsilon \geq \max\left\{c_3\sqrt{\frac{\log n_y}{n_y h_{n_y}^2}}, c_4 h_{n_y}^{\beta}\right\}.$$

The lemma follows by setting the bivariate bandwidth to

$$h_{n_y} = c_5\left(\frac{\log n_y}{n}\right)^{1/(2+2\beta)}.$$

This satisfies the conditions in Eq. (3.2) and minimizes the convergence rates of the two summands in Eq. (A.2). □

Proof of Lemma 2. For any (x_i,x_j), it follows from the Taylor expansion of $\log\widehat{p}_y(x_i,x_j)$

about $p_y(x_i, x_j)$ that

$$\log \widehat{p}_y(x_i, x_j) - \log p_y(x_i, x_j) \leq \frac{\widehat{p}_y(x_i, x_j) - p_y(x_i, x_j)}{p_y(x_i, x_j)}. \tag{A.5}$$

From Lemma A.2.1, for sufficiently large n_y,

$$\Pr\left[\sup_{(x_i, x_j) \in \Omega^o_{ij}} |\widehat{p}_y(x_i, x_j) - p_y(x_i, x_j)| < C_\delta \left(\frac{\log n_y}{n_y}\right)^{\frac{\beta}{2+2\beta}}\right] \geq 1 - \delta,$$

It follows that with high probability,

$$\left|\log \widehat{p}_y(x_i, x_j) - \log p_y(x_i, x_j)\right| \leq \frac{C_\delta}{p^{min}_{y,i,j}} \left(\frac{\log n_y}{n_y}\right)^{\frac{\beta}{2+2\beta}},$$

where $p^{min}_{y,i,j} > 0$ is the lower bound on the density $p_y(x_i, x_j)$. Now, since

$$\sup_{\mathbf{x} \in \Omega^o} \|\widehat{T}(\mathbf{x}) - T(\mathbf{x})\|_\infty = \max_{y \in \{-1,+1\}} \max_{i,j \in \{1,\ldots,d\}} \left\{\sup_{(x_i, x_j) \in \Omega^o_{ij}} |\log \widehat{p}_y(x_i, x_j) - \log p_y(x_i, x_j)|\right\},$$

we can take a union bound over the d univariate and $\binom{d}{2}$ bivariate densities for each class that make up the vectors $\widehat{T}$ and T. It follows that

$$\begin{aligned}
&\Pr\left[\sup_{\mathbf{x} \in \Omega^o} \|\widehat{T}(\mathbf{x}) - T(\mathbf{x})\|_\infty > \max_{i,j,y}\left\{\frac{C_\delta}{p^{min}_{y,i,j}} \left(\frac{\log n_y}{n_y}\right)^{\frac{\beta}{2+2\beta}}\right\}\right] \\
&\leq 2\sum_{i=1}^{d}\sum_{j=i}^{d} \Pr\left[\sup_{(x_i, x_j) \in \Omega^o_{ij}} |\log \widehat{p}_y(x_i, x_j) - \log p_y(x_i, x_j)| > \frac{C_\delta}{p^{min}_{y,i,j}} \left(\frac{\log n_y}{n_y}\right)^{\frac{\beta}{2+2\beta}}\right] \\
&\leq d(d+1) \cdot \delta
\end{aligned}$$

Therefore, for some constant C'_δ, with probability $\geq 1 - \delta$,

$$\sup_{\mathbf{x} \in \Omega^o} \|\widehat{T}_{n_0}(\mathbf{x}) - T(\mathbf{x})\|_\infty < C'_\delta \left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}}. \tag{A.6}$$

□

A.3 Theorem 3

In order to prove the theorem, we will use the following bound on the convergence of the empirical risk which holds simultaneously for all $\mathbf{w} \in \mathcal{W}$.

Lemma A.3.1. *Given an independent sample* $\{\widehat{T}_{n_0}(\mathbf{x}_\ell), y_\ell\}_{\ell=1}^{n_1}$*, let* $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0})$ *and* $R(\mathbf{w}, T)$ *be as defined in Eqs.* (3.4) *and* (3.7) *for a convex loss function* ϕ *that is* L_ϕ*–Lipschitz. Recall that* $\|\widehat{T}_{n_0}(\mathbf{x})\|_\infty < B$ *for some* $B > 0$*. With probability* $\geq 1 - \delta$*, the following holds uniformly for all* $\mathbf{w} \in \mathcal{W}$*:*

$$\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, \widehat{T}_{n_0}) \leq \frac{c_1}{\sqrt{n_1}} + c_2\sqrt{\frac{\log(1/\delta)}{2n_1}}$$

Proof. In order to prove bound the empirical risk uniformly over a class of functions, we use the Rademacher average as a measure of the complexity of the class. Recall the definition of the Rademacher complexity of a function class $\mathcal{F}$,

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i f(\mathbf{x}_i)\right]$$

where ε_i are independent and uniformly distributed on $\{\pm 1\}$ and $(\mathbf{x}_1, ..., \mathbf{x}_n)$ are iid. In our case, we are interested in the class of linear prediction functions $\mathcal{F}_\mathcal{W} = \{\widehat{T}_{n_0}(\mathbf{x}) \rightarrow \mathbf{w}^\mathrm{T}\widehat{T}_{n_0}(\mathbf{x}) : \mathbf{w} \in \mathcal{W}\}$. Theorem 7 of Bartlett et al. (2002) states that for $L_\phi-$Lipschitz loss ϕ bounded by c, with probability $\geq 1 - \delta$, every function in $\mathcal{F}_\mathcal{W}$ satisfies

$$\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, \widehat{T}_{n_0}) \leq 2L_\phi \mathcal{R}_{n_1}(\mathcal{F}_\mathcal{W}) + c\sqrt{\frac{\log(1/\delta)}{2n_1}}. \tag{A.7}$$

Note that the hinge loss, given our assumptions on the boundedness of $\widehat{T}_{n_0}(\mathbf{x})$, is bounded by $1 + d(d+1)AB$.

We can bound the Rademacher complexity for our class of functions, using the

Cauchy–Schwarz inequality, Jensen's inequality, and properties of norms,

$$\begin{aligned}
\mathcal{R}_n(\mathcal{F}_{\mathcal{W}}) &= \frac{1}{n_1}\mathbb{E}\left[\sup_{\mathbf{w}\in\mathcal{W}} \sum_{\ell=1}^{n_1} \varepsilon_\ell \mathbf{w}^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}_\ell)\right] \\
&= \frac{1}{n_1}\mathbb{E}\left[\sup_{\mathbf{w}\in\mathcal{W}} \mathbf{w}^{\mathrm{T}} \sum_{\ell=1}^{n_1} \varepsilon_\ell \widehat{T}_{n_0}(\mathbf{x}_\ell)\right] \\
&= \frac{A}{n_1}\mathbb{E}\left[\left\| \sum_{\ell=1}^{n_1} \varepsilon_\ell \widehat{T}_{n_0}(\mathbf{x}_\ell)\right\|_2\right] \\
&\leq \frac{A}{n_1}\sqrt{\mathbb{E}\left[\left\| \sum_{\ell=1}^{n_1} \varepsilon_\ell \widehat{T}_{n_0}(\mathbf{x}_\ell)\right\|_2^2\right]} \\
&\leq \frac{A}{n_1}\sqrt{\mathbb{E}\left[\sum_{\ell=1}^{n_1} \left\| \varepsilon_\ell \widehat{T}_{n_0}(\mathbf{x}_\ell)\right\|_2^2\right]} \\
&\leq \frac{\sqrt{d(d+1)}AB}{\sqrt{n_1}}
\end{aligned}$$

Combining this with Eq. (A.7) gives the result. □

Proof of Theorem 3. For any given $\mathbf{w} \in \mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq A\}$, we write the difference between the empirical risk using estimated transformations and the expected risk given in Eq. (3.7) as follows:

$$\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, T) = \left(\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, \widehat{T}_{n_0})\right) + \left(R(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, T)\right). \quad \text{(A.8)}$$

The previous lemma provided a uniform bound on the convergence $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) \to R(\mathbf{w}, \widehat{T}_{n_0})$. To bound the second term, note that since the loss function ϕ is L_ϕ–Lipschitz,

$$\begin{aligned}
\left|\phi(\mathbf{w}^{\mathrm{T}}\widehat{T}_{n_0}(\mathbf{x})) - \phi(\mathbf{w}^{\mathrm{T}}T(\mathbf{x}))\right| &\leq L_\phi \left|\mathbf{w}^{\mathrm{T}}\widehat{T}_{n_0}(\mathbf{x}) - \mathbf{w}^{\mathrm{T}}T(\mathbf{x})\right| \\
&\leq L_\phi \left|\left\langle \mathbf{w}, \widehat{T}_{n_0}(\mathbf{x}) - T(\mathbf{x})\right\rangle\right| \\
&\leq L_\phi \|\mathbf{w}\|_1 \left\|\widehat{T}_{n_0}(\mathbf{x}) - T(\mathbf{x})\right\|_\infty,
\end{aligned}$$

where the final line follows from Hölder's inequality. By an application of the Cauchy-Schwarz inequality, $\|\mathbf{w}\|_1 \leq \sqrt{d(d+1)}\|\mathbf{w}\|_2$, whereas from Lemma 2,

$$\left\|\widehat{T}_{n_0}(\mathbf{x}) - T(\mathbf{x})\right\|_\infty = O_P\left(\left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}}\right).$$

Since $\|\mathbf{w}\|_2 \leq A$, this second term is uniformly bounded.

Hence,

$$\sup_{\mathbf{w}\in\mathcal{W}}\left|\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, T)\right| = O_P(n_1^{-1/2}) + O_P\left(\left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}}\right)$$

□

A.4 Corollary 4

Proof. Consider the following decomposition of the excess risk of our estimated classifier relative to the oracle SVM classifier,

$$R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(\mathbf{w}^*, T) = \Big(R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(\widehat{\mathbf{w}}, T)\Big) + \Big(R(\widehat{\mathbf{w}}, T) - R(\mathbf{w}^*, T)\Big).$$

Using the Lipschitz property of ϕ, we can easily show (as we did in the proof of Theorem 3) that the first term converges at the uniform rate of consistency of $\widehat{T}_{n_0}$ given in Lemma 2,

$$R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(\widehat{\mathbf{w}}, T) = O_P\left(\left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}}\right). \tag{A.9}$$

Now consider the second term. By the definition of $\mathbf{w}^*$ as the minimizer of $R(\mathbf{w}, T)$ for all $\mathbf{w} \in \mathcal{W}$, we know this term to be nonnegative. We can further decompose this second term by adding and subtracting terms as follows,

$$\begin{aligned}0 \le R(\widehat{\mathbf{w}}, T) - R(\mathbf{w}^*, T) &= \Big(R(\widehat{\mathbf{w}}, T) - \widehat{R}_{n_1}(\widehat{\mathbf{w}}, \widehat{T}_{n_0})\Big) + \Big(\widehat{R}_{n_1}(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - \widehat{R}_{n_1}(\mathbf{w}^*, \widehat{T}_{n_0})\Big) \\ &\quad + \Big(\widehat{R}_{n_1}(\mathbf{w}^*, \widehat{T}_{n_0}) - R(\mathbf{w}^*, T)\Big) \\ &\le \Big(R(\widehat{\mathbf{w}}, T) - \widehat{R}_{n_1}(\widehat{\mathbf{w}}, \widehat{T}_{n_0})\Big) + \Big(\widehat{R}_{n_1}(\mathbf{w}^*, \widehat{T}_{n_0}) - R(\mathbf{w}^*, T)\Big),\end{aligned}$$

where the inequality follows from the fact that by the definition of $\widehat{\mathbf{w}}$, $\widehat{R}_{n_1}(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) \le \widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0})$ for any $\mathbf{w} \in \mathcal{W}$. By applying the convergence result in Theorem 3 to each of the terms in this upper bound, we have that

$$R(\widehat{\mathbf{w}}, T) - R(\mathbf{w}^*, T) = O_P(n_1^{-1/2}) + O_P\left(\left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}}\right). \tag{A.10}$$

Therefore, combining this with Eq. (A.9),

$$R(\widehat{\mathbf{w}}, \widehat{T}_{n_0}) - R(\mathbf{w}^*, T) = O_P(n_1^{-1/2}) + O_P\left(\left(\frac{\log n_0}{n_0}\right)^{\frac{\beta}{2+2\beta}}\right).$$

□

A.5 Theorem 5

Proof. Recall from Eqs. (3.6) and (3.8) that $\widehat{\mathbf{w}}$ is the minimizer of the empirical risk with estimated $\widehat{T}_{n_0}$ and $\mathbf{w}^*$ is the minimizer of the expected risk with oracle transformations T. To prove that $\widehat{\mathbf{w}} \to \mathbf{w}^*$ in probability it suffices to show that $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) \xrightarrow{n_0, n_1 \to \infty} R(\mathbf{w}, T)$ *uniformly* for all $\mathbf{w} \in \mathcal{W}$. That is, as $n_0, n_1 \to \infty$

$$\sup_{\mathbf{w} \in \mathcal{W}} \left|\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0}) - R(\mathbf{w}, T)\right| = o_P(1).$$

This uniform convergence is satisfied if and only the following four conditions hold (Newey, 1991):

1. $\mathcal{W}$ is compact,
2. $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0})$ converges pointwise to $R(\mathbf{w}, T)$
3. $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0})$ is stochastic equicontinuous, and
4. $R(\mathbf{w}, T)$ is equicontinuous.

It remains for us to show that the third and fourth conditions hold. To prove stochastic equicontinuity, it is sufficient that $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0})$ satisfies a global Lipschitz property: for all $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{W}$, there exists a random variable $K_{n_1} = O_P(1)$ such that

$$\left|\widehat{R}_{n_1}(\mathbf{w}_1, \widehat{T}_{n_0}) - \widehat{R}_{n_1}(\mathbf{w}_2, \widehat{T}_{n_0})\right| \leq K_{n_1} ||\mathbf{w}_1 - \mathbf{w}_2||_2.$$

Note that for any $\mathbf{w}_1, \mathbf{w}_2 \in \mathcal{W}$,

$$\left|\widehat{R}_{n_1}(\mathbf{w}_1, \widehat{T}_{n_0}) - \widehat{R}_{n_1}(\mathbf{w}_2, \widehat{T}_{n_0})\right| \leq \frac{1}{n_1} \sum_{\ell=1}^{n_1} \left|\phi(\mathbf{w}_1^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}_\ell), y_\ell) - \phi(\mathbf{w}_2^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}_\ell), y_\ell)\right|.$$

Hence, we only need to show that $\phi(\mathbf{w}^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}), y)$ is Lipschitz. Since ϕ is L_ϕ–Lipschitz,

$$\begin{aligned} \left|\phi(\mathbf{w}_1^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x})) - \phi(\mathbf{w}_2^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}))\right| &\leq L_\phi \left|\mathbf{w}_1^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}) - \mathbf{w}_2^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x})\right| \\ &= L_\phi \left|\left\langle \mathbf{w}_1 - \mathbf{w}_2, \widehat{T}_{n_0}(\mathbf{x}) \right\rangle\right| \\ &\leq L_\phi \left\|\widehat{T}_{n_0}(\mathbf{x})\right\|_2 \|\mathbf{w}_1 - \mathbf{w}_2\|_2. \end{aligned}$$

Recall that $\|\widehat{T}_{n_0}(\mathbf{x})\|_\infty \leq B$. It follows that $\phi(\mathbf{w}^{\mathrm{T}} \widehat{T}_{n_0}(\mathbf{x}))$, and hence also $\widehat{R}_{n_1}(\mathbf{w}, \widehat{T}_{n_0})$, is $L_\phi B \sqrt{d(d+1)}$-Lipschitz.

Similarly, the equicontinuity of the limiting function $R(\mathbf{w}, T)$ follows from the fact that $\phi(\mathbf{w}^{\mathrm{T}} T(\mathbf{x}))$ is $L_\phi B \sqrt{d(d+1)}$-Lipschitz.

□

APPENDIX B

EXTENDED SIMULATION RESULTS

B.1 Wilcoxon signed-rank test to compare classifier performance

The performance of the forest density classifier was compared with that of four alternative classifiers on each model in the simulation study, as described in Section 4. Detailed results from applying a one-sided Wilcoxon signed-rank test for each simulation model are displayed in Tables B.1–B.4. The test statistic W is defined as the sum of ranks of positive differences between the misclassification error rates for the forest density classifier and an alternative classifier. An asterisk marks the comparisons for which the p-value $< 0.05/256$. This significance level reflects a conservative Bonferroni adjustment for the 256 tests performed. In these cases, we conclude that the misclassification error for the forest density classifier was significantly smaller than that of its competitor classifier.

Common			forestdens vs 5NN			forestdens vs NB			forestdens vs SVM			forestdens vs RF		
structure	Priors	n	W	p-value		W	p-value		W	p-value		W	p-value	
None	Unb	100	4	< 0.0001	*	378.5	0.9991		0	< 0.0001	*	4	< 0.0001	*
		200	1.5	< 0.0001	*	304.5	0.9306		0	< 0.0001	*	1	< 0.0001	*
		400	6	< 0.0001	*	273.5	0.7992		11	< 0.0001	*	15.5	< 0.0001	*
		800	15	< 0.0001	*	79.5	0.0019		31.5	< 0.0001	*	0	< 0.0001	*
	Bal	100	0	< 0.0001	*	222	0.5403		0	< 0.0001	*	2	< 0.0001	*
		200	7	< 0.0001	*	199	0.2497		26	< 0.0001	*	35.5	< 0.0001	*
		400	18	< 0.0001	*	67	0.0003		118	0.0151		24	< 0.0001	*
		800	0	< 0.0001	*	11	< 0.0001	*	258	0.8942		10.5	< 0.0001	*
$2/3$	Unb	100	0	< 0.0001	*	163.5	0.2750		0	< 0.0001	*	0	< 0.0001	*
		200	5	< 0.0001	*	173	0.1132		0	< 0.0001	*	8.5	< 0.0001	*
		400	17.5	< 0.0001	*	52.5	< 0.0001	*	0	< 0.0001	*	3	< 0.0001	*
		800	30	< 0.0001	*	13	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*
	Bal	100	1	< 0.0001	*	66.5	0.0002	*	11.5	< 0.0001	*	23	< 0.0001	*
		200	2	< 0.0001	*	4.5	< 0.0001	*	36	< 0.0001	*	48	0.0001	*
		400	17	< 0.0001	*	21.5	< 0.0001	*	27	< 0.0001	*	60.5	0.0001	*
		800	26.5	< 0.0001	*	0	< 0.0001	*	31	< 0.0001	*	35	< 0.0001	*

Table B.1: Summary of Wilcoxon signed-rank tests comparing the forest density classifier to four competitor classifiers on simulated datasets having sparse forest distributions and normal marginals.

Common			forestdens vs 5NN			forestdens vs NB			forestdens vs SVM			forestdens vs RF		
structure	Priors	n	W	p-value		W	p-value		W	p-value		W	p-value	
None	Unb	100	4	0.7203		16.5	< 0.0001	*	10	< 0.0001	*	0	< 0.0001	*
		200	1.5	0.0016		0	< 0.0001	*	52	< 0.0001	*	0	< 0.0001	*
		400	3	< 0.0001	*	0	< 0.0001	*	104	0.0035		0	< 0.0001	*
		800	0	< 0.0001	*	0	< 0.0001	*	102.5	0.0103		0	< 0.0001	*
	Bal	100	1	1.0000		0	< 0.0001	*	460	1.0000		24	< 0.0001	*
		200	0	0.4766		0	< 0.0001	*	458	1.0000		0	< 0.0001	*
		400	22	< 0.0001	*	0	< 0.0001	*	308	0.9399		0	< 0.0001	*
		800	0	< 0.0001	*	0	< 0.0001	*	70.5	0.0009		0	< 0.0001	*
$2/3$	Unb	100	3	0.2125		6	< 0.0001	*	14	< 0.0001	*	16.5	< 0.0001	*
		200	0	< 0.0001	*	0	< 0.0001	*	1	< 0.0001	*	3	< 0.0001	*
		400	23.5	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*
		800	5.5	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*	3	< 0.0001	*
	balanced	100	0	0.9514		0	< 0.0001	*	419	1.0000		47	< 0.0001	*
		200	0	0.0014		0	< 0.0001	*	423.5	1.0000		38	< 0.0001	*
		400	5	< 0.0001	*	0	< 0.0001	*	197	0.2369		0	< 0.0001	*
		800	1	< 0.0001	*	0	< 0.0001	*	4	< 0.0001	*	0	< 0.0001	*

Table B.2: Summary of Wilcoxon signed-rank tests comparing the forest density classifier to four competitor classifiers on simulated datasets having tree distributions and normal marginals.

Common			forestdens vs 5NN			forestdens vs NB			forestdens vs SVM			forestdens vs RF		
structure	Priors	n	W	p-value		W	p-value		W	p-value		W	p-value	
none	Unb	100	261	< 0.0001	*	222.5	0.4217		0	< 0.0001	*	0	< 0.0001	*
		200	193	< 0.0001	*	85.5	0.0009		0	< 0.0001	*	0	< 0.0001	*
		400	73	< 0.0001	*	10	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*
		800	47.5	< 0.0001	*	0	< 0.0001	*	20	< 0.0001	*	0	< 0.0001	*
	Bal	100	93.5	< 0.0001	*	9	< 0.0001	*	31	< 0.0001	*	42.5	< 0.0001	*
		200	48	< 0.0001	*	2	< 0.0001	*	98.5	0.0024		161.5	0.1158	
		400	41	< 0.0001	*	0	< 0.0001	*	190.5	0.2841		98.5	0.0044	
		800	13.5	< 0.0001	*	0	< 0.0001	*	232.5	0.5020		68	0.0002	*
$2/3$	Unb	100	420.5	< 0.0001	*	195	0.2245		0	< 0.0001	*	0	< 0.0001	*
		200	313	< 0.0001	*	95.5	0.0019		0	< 0.0001	*	0	< 0.0001	*
		400	5	< 0.0001	*	5	< 0.0001	*	0	< 0.0001	*	2.5	< 0.0001	*
		800	28	< 0.0001	*	4.5	< 0.0001	*	0	< 0.0001	*	28.5	< 0.0001	*
	Bal	100	214.5	< 0.0001	*	0	< 0.0001	*	44.5	< 0.0001	*	121	0.0103	
		200	91.5	< 0.0001	*	0	< 0.0001	*	74	0.0003		211.5	0.3368	
		400	0	< 0.0001	*	0	< 0.0001	*	83.5	0.0014		185.5	0.2490	
		800	0	< 0.0001	*	0	< 0.0001	*	53	< 0.0001	*	169	0.0980	

Table B.3: Summary of Wilcoxon signed-rank tests comparing the forest density classifier to four competitor classifiers on simulated datasets having sparse forest distributions and complex marginals.

Common			forestdens vs 5NN			forestdens vs NB			forestdens vs SVM			forestdens vs RF		
structure	Priors	n	W	p-value		W	p-value		W	p-value		W	p-value	
None	Unb	100	21	0.0003		72	0.0003		0	< 0.0001	*	1	< 0.0001	*
		200	2	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*	1	< 0.0001	*
		400	11.5	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*
		800	3	< 0.0001	*	0	< 0.0001	*	6	< 0.0001	*	0	< 0.0001	*
	Bal	100	0	< 0.0001	*	0	< 0.0001	*	38	< 0.0001	*	39.5	< 0.0001	*
		200	0	< 0.0001	*	0	< 0.0001	*	64.5	0.0001	*	51	0.0001	*
		400	7	< 0.0001	*	0	< 0.0001	*	79	0.0005		14	< 0.0001	*
		800	0	< 0.0001	*	0	< 0.0001	*	19	< 0.0001	*	0	< 0.0001	*
2/3	Unb	100	2	< 0.0001	*	64	0.0001	*	0	< 0.0001	*	0	< 0.0001	*
		200	1	< 0.0001	*	5.5	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*
		400	0	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*
		800	0	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*
	Bal	100	0	< 0.0001	*	0	< 0.0001	*	77.5	0.0004		59.5	0.0001	*
		200	0	< 0.0001	*	0	< 0.0001	*	3.5	< 0.0001	*	74.5	0.0003	
		400	0	< 0.0001	*	0	< 0.0001	*	5	< 0.0001	*	34	< 0.0001	*
		800	0	< 0.0001	*	0	< 0.0001	*	0	< 0.0001	*	9	< 0.0001	*

Table B.4: Summary of Wilcoxon signed-rank tests comparing the forest density classifier to four competitor classifiers on simulated datasets having tree distributions and complex marginals.