2015

Local prediction and classification techniques for machine learning and data mining

Cory Lee Lanker
Iowa State University

Local prediction and classification techniques for machine learning and data mining

by

Cory L. Lanker

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Statistics

Program of Study Committee:
Stephen B. Vardeman, Co-Major Professor
Max D. Morris, Co-Major Professor
Kris De Brabanter
Dan Nettleton
Huaiqing Wu

Iowa State University

Ames, Iowa

2015

DEDICATION

To Karen

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

ABSTRACT ix

CHAPTER 1. INTRODUCTION 1

CHAPTER 2. A DATA-DERIVED MIXTURE PRIOR FOR PREDICTION BASED ON HIERARCHICAL BAYES GAUSSIAN MIXTURE MODELS 3

2.1 Introduction 4

2.2 Prediction Using Mixture Models 6

2.2.1 Population probability model 6

2.2.2 Dirichlet process model 7

2.2.3 Prior distribution 7

2.2.4 Posterior distribution 9

2.2.5 Prediction 9

2.2.6 Tuning model parameters 11

2.3 Demonstration of Our Method 11

2.3.1 Predicting the doppler function 12

2.3.2 Comprehensive simulation study 17

2.3.3 Application to machine learning data sets 27

2.4 Discussion 31

2.5 Appendix: Derivation of Posterior Conditional Distributions 32

2.5.1 Conditional posterior of θ_i 34

2.5.2 Conditional posterior on μ_m 34

2.5.3 Conditional posterior of Σ_m 36

CHAPTER 3. CLASSIFICATION WITH A DATA-DERIVED MIXTURE PRIOR BASED ON HIERARCHICAL BAYES GAUSSIAN MIXTURE MODELS 37

3.1 Introduction 38

3.2 Classification Using Mixture Models 40

3.2.1 Classification problem setup 40

3.2.2 Model prior specification 42

3.2.3 Posterior distribution 45

3.2.4 Classification 49

3.3 Demonstration of Our Method 51

3.3.1 Predicting the doppler function 51

3.3.2 Comprehensive simulation study 52

3.3.3 Application to machine learning data set 65

3.4 Discussion 66

3.5 Appendix: Derivation of Posterior Conditional Distribution for μ_m 68

CHAPTER 4. A GENERIC CLASSIFICATION ROUTINE FOR CATEGORICAL PREDICTORS USING LIKELIHOOD RATIO STATISTICS AND RANDOM FORESTS 70

4.1 Introduction 71

4.2 Classification Using Categorical Predictors 72

4.3 Generic Classification With Likelihood Ratios 75

4.4 Demonstration of Our Method 78

4.4.1 Simulation study 78

4.4.2 King-rook versus king-pawn data set 82

4.5 Conclusion 84

BIBLIOGRAPHY 85

LIST OF TABLES

Table 2.1 Unexplained Variation of the Doppler Function for 50 Random Seeds 14

Table 2.2 Unexplained Variation of the Doppler Function With Spurious Predictor for 50 Random Seeds 15

Table 2.3 Unexplained Variation of the Doppler Function With Various Scale Matrices Provided for Mixture Prior for 50 Random Seeds 16

Table 2.4 Average Computation Times, in Minutes, for the Prediction Methods 27

Table 2.5 R^2 for Prediction of Four Machine Learning Data Sets Using Mixture Prior Bayes, k-Nearest Neighbors, and Random Forests . . . 27

Table 3.1 Average Misclassification Error for Banknote Authentication Data 67

LIST OF FIGURES

Figure 2.1 Doppler Function Test Case Data. 13

Figure 2.2 Comparison of Doppler Function Prediction Values. 14

Figure 2.3 Fewer Scale Matrices Degrades Prediction Performance. 16

Figure 2.4 Component Parameters During Gibbs Sampler Initialization. . . 17

Figure 2.5 Maximum Component Overlap From MixSim-Generated Data. . 19

Figure 2.6 Optimal Number of Components for the Likelihood-Based Method. 21

Figure 2.7 Optimal Number of Neighbors for k-Nearest Neighbors. 22

Figure 2.8 Mixture Prior Bayes Prediction Error Ratios With Comparison Methods for Comprehensive Simulation Study. 23

Figure 2.9 Prediction Error Ratios For Six Predictors. 26

Figure 2.10 Iterate Squared Error Rates for Gibbs Sample for the California Housing Data Analysis. 30

Figure 3.1 Classification of Data From the Doppler Function 53

Figure 3.2 Simulation Study Error Ratios for $p = 6$ and Multivariate Normal Scenario. 57

Figure 3.3 Simulation Study Error Ratios for $p = 6$ and Spurious Predictor Scenario. 58

Figure 3.4 Simulation Study Error Ratios for $p = 6$ and Uniform X Scenario. 59

Figure 3.5 Simulation Study Error Ratios for Small-Sized Data Sets and Multivariate Normal Scenario. 60

Figure 3.6 Simulation Study Error Ratios for Medium-Sized Data Sets and Multivariate Normal Scenario. 61

Figure 3.7 Simulation Study Error Ratios for Small-Sized Data Sets and Spurious Predictor Scenario. 62

Figure 3.8 Simulation Study Error Ratios for Medium-Sized Data Sets and Spurious Predictor Scenario. 63

Figure 3.9 Simulation Study Computation Time. 66

Figure 3.10 Pairs of Predictors for Banknote Authentication Data Set. 67

Figure 4.1 Simulation Study Category Proportions for Different α Values. . 80

Figure 4.2 Ratio of Misclassification Error for Random Forests With Different Input. 81

Figure 4.3 Ratio of Misclassification Error With Sparse Data Correction Factor. 83

ABSTRACT

A variety of conditional probability models estimate the regression or class probability function for the purpose of prediction or classification. Bayesian mixture models provide flexible prediction and classification methods for modeling local linearities of the regression or class probability function. A hierarchical Bayes Gaussian mixture model is proposed that directly uses data to define a mixture prior for its Gaussian mixture component parameters. This nonparametric Bayesian mixture model uses the stick-breaking construction of a Dirichlet process model. Prediction and classification comes directly from the posterior distribution via Gibbs sampling. Comprehensive simulation studies demonstrate performance of both the regression and classification methods. Five standard machine learning data sets show prediction and classification results competitive with local methods. A generic classification algorithm is outlined given categorical predictors. If too many categories are present or if many interaction levels affect the class probability function, no current methods can reduce bias effectively. A proposed solution is a generic way to characterize the information about the class probability function available in the predictors through likelihood ratio statistics. This proposed classifier relies on random forests to reduce bias by utilizing all information in the generated log likelihood ratio features. A simulation study and an application data set demonstrate potential advantages of this classification method for categorical predictors.

CHAPTER 1. INTRODUCTION

My research is in predictive analytics—the use of data to form prediction or classification values for new observations. The goal of prediction is to form values with minimum squared error by estimating the regression function. The goal of classification is to form values with minimum misclassification error by choosing the most probable class. Most of my research extends a Bayesian model formulated in an unpublished manuscript by Ken Ryan and Stephen Vardeman in 2012. My research applies this Bayesian model to prediction and classification problems. The Ryan and Vardeman model defines a Dirichlet process clustering procedure using a mixture prior on the model's latent Gaussian mixture parameters. Such mixture prior corrects a deficiency of multivariate Gaussian clustering procedures being too sensitive to the choice of covariance prior as noted in Gelman et al. (2013). The model has known conditional distributions for all parameters, allowing efficient Gibbs sampling of the posterior distribution.

In my first research project, I create a flexible prediction method based on the Ryan and Vardeman model. I apply this model to prediction problems by calculating the conditional mean of the predictive posterior distribution. The provided Gibbs sampler C code had implementation problems, requiring significant changes. I wrote R code to allow comprehensive simulation studies. Prediction results are competitive with the frequentist alternative and other methods in a variety of tested scenarios. In general, this mixture prior prediction method improves prediction for small sample multivariate data with locally linear regression functions. Four standard machine learning data sets demonstrate improved prediction performance.

My second research project extends this functioning mixture prior prediction method to classification problems. Difficulties arise as I cannot simply input a binary response into a multivariate Gaussian mixture model. Instead, I create a latent Gaussian response with its own mixture prior. The most probable class comes from the resulting posterior distribution. I derived the form of the conditional distributions for the mean parameter and the latent response, and I rewrote the C code Gibbs sampler and R code for this context. A simulation study and standard data sets characterize method performance.

My third research project deals with classification with sparse categorical data. Methods currently use categorical data ineffectively when there are many sparse categorical variables or the response is primarily a function of interactions between sparse categorical variables. I demonstrate that using likelihood ratio statistics of the categorical counts instead of the categorical predictors improves classification performance of random forest classifiers. I offer a correction to the calculation of likelihood ratio features to prevent overfitting the classifier. This method improves classification of a standard machine learning chess classification data set.

CHAPTER 2. A DATA-DERIVED MIXTURE PRIOR FOR PREDICTION BASED ON HIERARCHICAL BAYES GAUSSIAN MIXTURE MODELS

A paper in preparation

Cory L. Lanker[1,2], Kenneth J. Ryan[3], Mark V. Culp[3], Max D. Morris[1],
Stephen B. Vardeman[1]

Abstract

A variety of conditional probability models estimate the regression function for the purpose of prediction. Bayesian mixture models provide flexible prediction methods for modeling local linearities of the regression function. We propose a hierarchical Bayes Gaussian mixture model as a flexible prediction method given continuous predictors. This paper outlines a probability model that amends a standard Bayes prediction method based on Gaussian mixture models by directly using data to define a mixture prior for its Gaussian mixture component parameters. Without this mixture prior the posterior distribution is strongly influenced by the choice of prior variance and typically yields over-smoothed estimates of the regression function. A data-derived prior circumvents this problem and gives a posterior distribution with more localized modeling of the regression function. This nonparametric Bayesian mixture model uses the stick-breaking

[1] Graduate student, Professor, and University Professor, respectively, Department of Statistics, Iowa State University.

[2] Primary researcher and author.

[3] Associate Professor, Department of Statistics, West Virginia University.

construction of a Dirichlet process model. Prediction of the regression function comes directly from the posterior distribution via Gibbs sampling. A comprehensive simulation study demonstrates method performance with regression functions composed of a mixture of eight overlapping multivariate normal densities. Four standard machine learning data sets show prediction results competitive with local regression methods.

2.1 Introduction

The subject of this paper is flexible prediction for locally linear regression functions. The problem under consideration is estimation of a multivariate regression function. This paper offers a prediction method that extends the hierarchical Bayes Gaussian mixture model of Gelman et al. (2013) and Bayesian curve fitting approach of Müller et al. (1996) by constructing a mixture prior for the model's Gaussian parameters. Estimating the regression equation typically involves a joint probability model; e.g., a linear regression model can characterize multivariate normal data.

We construct a model as if data are from a mixture of multivariate normal densities. If identifiers existed to tell which densities the observations come from, we could estimate a multivariate normal probability model for each density of the mixture. However, if density identifiers do not exist, prediction requires estimation of both the mixture component parameters and the source densities for new observations. In the case of missing identifiers, we could include indicators in the probability model to express these missing data. If the number of mixture densities was known, we could use the EM algorithm to get estimates for the mixture Gaussian parameters. However, if the number of densities was unknown, using the EM algorithm would first require estimation of the number of subpopulations and such estimation is a difficult problem as information criteria are unreliable (McLachlan and Peel, 2004). A nonparametric Bayes approach does not require estimation of the number of densities—this approach uses a Dirichlet process prior for the mixture proportion sizes. To get realizations from this prior we use the stick-breaking construction of Sethuraman (1994).

The requirement that data come from a finite number of mixture component densities motivates a hierarchical Bayes model. While diffuse, proper priors could be put on the Gaussian component parameters, these generally do not give good prediction results (Gelman et al., 2013). A diffuse normal–inverse-Wishart prior may induce flatness in the posterior distribution that itself becomes too diffuse to give prediction results competitive with other smoothing methods. The challenge of using a Bayesian mixture model for prediction is focusing the prior on the data in a principled approach that generates mixture components close to the support of the data (Gelman et al., 2013).

This paper outlines a mixture prior on the Gaussian component parameters in the mixture model. The proposed prediction method uses data to calculate parameters in both the stick-breaking and Gaussian mixture portion of the prior. Bayesian solutions to mixture problems often are computationally intensive and are not easily sampled, necessitating Gibbs sampling through derivation of conditional distributions for all parameters (Diebolt and Robert, 1994). Our model has fully defined conditional distributions for all parameters allowing efficient posterior sampling with a Gibbs sampler.

A variety of test cases shows results when using this mixture prior Bayesian mixture model for prediction. Using this method on data from the bivariate doppler function—a function with strong local linearity—demonstrates that the mixture prior Bayesian method adapts predicted values to the local structure of the regression function. We also present a comprehensive simulation study involving a regression function composed from a mixture of eight multivariate normal conditional distributions. The study has various data-generating scenarios using 6, 12, and 18 predictors, and prediction results are competitive with local regression methods. Prediction of standard regression data sets shows similar competitive results.

In this paper, Section 2.2 formally outlines the problem and details our hierarchical Bayes model and prediction algorithm. Section 2.3 shows results of prediction with our proposed method, comparing performance with other local regression techniques. Section 2.4 concludes the paper with a discussion of this method and its performance.

2.2 Prediction Using Mixture Models

We have N_{obs} observations (x_i, y_i) and wish to predict the response y^* for a new x^* that minimizes squared error loss. We use the mixture model described in Section 2.1 without a priori specification of the number of mixture components or of individual observations to mixture components. Let parameter θ_i represent the multivariate Gaussian parameters for observed (x_i, y_i), yielding the likelihood function $f(x, y|\theta) = \prod_{i=1}^{N_{obs}} N(x_i, y_i|\theta_i)$, where $N(x_i, y_i|\theta_i)$ represents the density of a multivariate normal distribution with mean and variance parameters θ_i evaluated at (x_i, y_i).

2.2.1 Population probability model

Model the joint distribution of (x, y) with a mixture of H components, for $x \in \mathbb{R}^p$, $y \in \mathbb{R}$, and unknown H. Define parameter λ as the collection of population proportions λ_h, with $\sum_{h=1}^{H} \lambda_h = 1$. The sampling distribution of an observation (x_i, y_i) from this population has the density

$$p(x_i, y_i|\lambda, \xi) = \sum_{h=1}^{H} \lambda_h \, N(x_i, y_i|\xi_h), \tag{2.1}$$

where $N(x, y|\xi_h)$ denotes the density of the joint multivariate Gaussian distribution for component h. Gaussian parameter vector ξ_h consists of a mean vector of size $p + 1$ and covariance matrix of size $(p + 1) \times (p + 1)$, and the population proportions λ_h sum to one. Such a model captures variation in the Gaussian parameters across mixtures of the population.

2.2.2 Dirichlet process model

We approximate the population mixture distribution of (2.1) with the following Dirichlet process model employing a latent multivariate Gaussian mixture. Let π_m and η_m represent the proportion and Gaussian parameters for component $m = 1, \ldots, \infty$, with the proportions π_m summing to one. For practical purposes we truncate this Dirichlet process mixture at n components. We select n large enough such the conditional distribution of y given x obtained from our model $\sum_{m=1}^{n} \pi_m \, N(x, y|\eta_m)$ closely approximates the regression function of y on x from the population model $\sum_{h=1}^{H} \lambda_h \, N(x, y|\xi_h)$.

Let parameter η_m consist of the Gaussian mean μ_m and variance Σ_m for component m. Let μ denote the collection of n mean vectors μ_m and let Σ denote the collection of n covariance matrices Σ_m. Conditional on estimates for model parameters π, μ, and Σ, prediction of y^* for a new x^* follows from the conditional Gaussian density $f(y^*|x^*) = \sum_{m=1}^{n} p_m(x^*) N(y^*|x^*, \mu_m, \Sigma_m)$ with component membership proportion

$$p_m(x^*) = \pi_m \, N(x^*|\mu_{m,x}, \Sigma_{m,xx}) / (\textstyle\sum_{m'=1}^{n} \pi_{m'} \, N(x^*|\mu_{m',x}, \Sigma_{m',xx})),$$

where $\mu_{m,x}$ and $\Sigma_{m,xx}$ are the marginal parameters for x.

2.2.3 Prior distribution

We follow a hierarchical Bayesian approach and define an appropriate prior on model parameters θ, π, and η, using the resulting posterior distribution for prediction.

Each θ_i is assigned to an η_m according to the mixture component proportion size π_m. The conditional prior distribution for parameter θ_i given parameters π and η is $\prod_{m=1}^{n} \pi_m^{\mathbb{I}[\eta_m = \theta_i]}$. Given the data (x, y), one can find optimal values for θ, π, and η that maximize the product of the likelihood function $f(x, y|\theta)$ and prior for θ. Such a model is not identifiable due to exchangeability of mixture labels.

The prior distribution for Gaussian distribution parameters η, denoted $g(\eta)$, is an extension of the normal–inverse-Wishart prior for (μ, Σ) of Müller et al. (1996). We extend this prior by composing two mixture distributions: one for the component mean μ prior using multivariate Gaussian densities, and another for the component variance Σ prior using inverse-Wishart densities. With μ_m and Σ_m independent, the prior for each η_m becomes $g(\eta_m) = g_1(\mu_m)g_2(\Sigma_m)$, with the mean and covariance mixture priors as

$$g_1(\mu_m) \propto |\Gamma|^{-1/2} \sum_{i=1}^{N_{obs}} e^{-\frac{1}{2}(\mu_m - z_i)^T \Gamma^{-1}(\mu_m - z_i)}, \tag{2.2}$$

$$g_2(\Sigma_m) \propto |\Sigma_m|^{-(\rho+p+2)/2} \sum_{l=1}^{L} w_l \, |M_l|^{\frac{\rho}{2}} \, e^{-\frac{1}{2}\operatorname{tr}\left(M_l \Sigma_m^{-1}\right)}, \tag{2.3}$$

where $z_i \equiv (x_i, y_i)$. Let $g_1(\mu_m)$ be a mixture of N_{obs} Gaussian densities with equal weights and $g_2(\Sigma_m)$ be a mixture of L inverse-Wishart densities with weights defined by w. Therefore the prior of $g(\eta_m)$ is a product mixture in $(\mu \times \Sigma)$-space composed of $N_{obs} \cdot L$ normal–inverse-Wishart densities.

The prior (2.2) on mean parameter μ_m is an equal-weighted average of Gaussian densities centered at the observations (x_i, y_i). To make the priors for each μ_m and Σ_m independent, we do not use Σ_m for the mixture variance. Instead the variance for each of the densities is the empirical multivariate bandwidth

$$\Gamma = N_{obs}^{-(p+7)/(p+5)} \sum_{i=1}^{N_{obs}} (z_i - \bar{z})(z_i - \bar{z})^T$$

derived from Simonoff (1996).

Similarly, we introduce prior (2.3) on covariance parameter Σ_m as a mixture of inverse-Wishart densities with user-provided scale matrices M_l and degrees of freedom ρ. For example, these scale matrices could be comprised of the covariance matrices obtained by a clustering algorithm performing an exhaustive search of the mixture structure of the data; such an algorithm produces L covariance matrix guesses M_l, each with selection weight w_l. We generate a large collection of guesses M_l to have an inverse-Wishart density in the mixture to closely approximate each true subpopulation covariance Σ_h in (2.1).

The priors for the other model parameters θ and π are standard for a Dirichlet process model, e.g., see Gelman et al. (2013). Each independent θ_i has discrete prior $\sum_{m=1}^{n} \pi_m \delta_{\eta_m}$, where δ_{η_m} is a unit point mass at η_m, i.e., $P(\theta_i = \eta_m) = \pi_m$. We use a truncated stick-breaking prior for π via variables $\phi_1, \ldots, \phi_{n-1}$, each following a conjugate beta distribution. Declare π_m as functions of variables ϕ created by the stick-breaking representation of the Dirichlet process through the following equations $\pi_m(\phi) \equiv \phi_m \prod_{k=1}^{m-1}(1 - \phi_k)$, with the first $n-1$ $\phi_m \sim$ iid Beta(α, β), and $\phi_n \equiv 1$. The beta distribution hyperparameters α and β can be determined from the data to conform to some optimal mixture proportions from a clustering algorithm.

2.2.4 Posterior distribution

The conditional posterior distribution for the parameters, $p(\theta, \eta, \phi | z)$, where z represents the collection of data (x, y), is proportional to

$$p(\eta, \phi, \theta | z) \propto \; p(z | \eta, \phi, \theta) p(\eta, \phi, \theta) = p(z|\theta) p(\theta | \eta, \phi) p(\eta, \phi). \tag{2.4}$$

Note in equation (2.4) that the likelihood is independent of latent η and π. Conditional distributions can be found for the individual θ_i, ϕ_m, μ_m, and Σ_m to allow Gibbs sampling of the posterior. The exact posterior distribution and the derivation details are shown in the Appendix. As a summary, the conditional distributions of θ are multinomial, ϕ are beta, μ are a Gaussian mixture, and Σ are an inverse-Wishart mixture.

2.2.5 Prediction

A common error metric for prediction of continuous y is mean squared prediction error. The function of the predictors x that minimizes the mean squared prediction error is the expected value of the response y given x, according to the joint probability model. Therefore the task of finding a prediction function that minimizes the mean squared prediction error is equivalent to estimating the regression function of y on x (Izenman, 2009).

The Gibbs sampler generates a collection of J samples of the parameter values that are taken to be random draws from the joint posterior distribution after convergence is reached (Gamerman and Lopes, 2006). The sampler saves the values for parameters π_m, μ_m, and Σ_m for J iterations, not saving the θ_i values that are unused for prediction. See a resource such as Gamerman and Lopes (2006) for a discussion of burn-in and thinning for proper convergence of Gibbs samplers.

These J saved sets of parameter values, $\big(\pi_m^{(j)}, \mu_m^{(j)}, \Sigma_m^{(j)}\big)$ for $j = 1, \ldots, J$, are used to make a prediction for y^* given a new observation x^*. To get a prediction $\hat{y}$ for y^* at x^*, compute the following for each sample $j = 1, \ldots, J$:

1. for each mixture component m, compute $\ell_m^{(j)}(x^*)$, the likelihood that x^* belongs to component m given the mixture arrangement at sample j,

   $$\ell_m^{(j)}(x^*) = \pi_m^{(j)}\, N(x^* | \mu_{m,x}^{(j)}, \Sigma_{m,xx}^{(j)}),$$

   where the Gaussian density is based only on the marginal of x (Bernardo et al., 2011),

2. for each component m, compute $\hat{y}_m^{(j)}(x^*)$, the sample predicted conditional mean at x^*, $\hat{y}_m^{(j)}(x^*) = \mu_{m,y}^{(j)} + \Sigma_{m,yx}^{(j)} \Sigma_{m,xx}^{(j)\,-1}(x^* - \mu_{m,x}^{(j)})$, where $\mu_{m,y}^{(j)}$ is the y marginal of $\mu_m^{(j)}$ and $\Sigma_{m,yx}^{(j)}$ is the x-y covariance portion of $\Sigma_m^{(j)}$,

3. form $\hat{y}^{(j)}(x^*)$, the prediction for y^* at x^* for sample j, by computing the weighted average of the predicted conditional means

   $$\hat{y}^{(j)}(x^*) = \big(\textstyle\sum_{m=1}^n \ell_m^{(j)}(x^*)\, \hat{y}_m^{(j)}(x^*)\big) / \big(\textstyle\sum_{m=1}^n \ell_m^{(j)}(x^*)\big).$$

With these J sample predictions, the calculation of $\hat{y}(x^*)$, the predicted value for y^* at x^*, is the average $\hat{y}(x^*) = \frac{1}{J}\sum_{j=1}^J \hat{y}^{(j)}(x^*)$.

2.2.6 Tuning model parameters

Implementation of this method involves tuning the parameters of the model: α, β, ρ, and n. Our implementation always assumes α is one, but tuning this parameter along with β would allow flexibility in controlling the prior for mixture proportions. We estimate β through minimization of the expected mixture proportions compared to the empirically found BIC-optimal mixture sizes, and in this way we again use the data in coming up with the hyperparameters. Parameter ρ must be $p+1$ or larger, and in our implementation defaults to $p+3$, a balance between too much and too little variation in covariance matrix draws. The number of model components n should always be larger than the number of local linear regions of the regression function, and we suspect that the Gibbs converges faster as n increases. An estimate of the number of local linear regions, choosing a conservatively large value, can come from a clustering algorithm.

2.3 Demonstration of Our Method

We demonstrate our method's performance by predicting the two-dimensional doppler function, a variety of simulated higher-dimensional test cases, and data sets from machine learning repositories. In the simulated and repository test cases, we compare method performance with that of a frequentist implementation, a Bayes implementation without a mixture prior, and some nonparametric prediction methods including k-nearest neighbors and random forests.

In the following simulations, we use R library Mclust to get the mixture prior M_l matrices and weights w_l. In general, there should be enough variety among the collection of M_l matrices to ensure one of them is a close match to the covariance structure of any local linear region. Our weights w_l have the form $1/kC$ where k is the number of clusters in the M_l-generating Mclust run and C is the maximum number of clusters tried.

In these simulations, the sampler burn-in time is about one-fourth of the total number of iterations. The Gibbs sampler is coded in C to shorten run times. The sample sizes are between 500 and 1000, with a thinning rate to balance computation time and prediction quality. Gibbs sampler convergence checking is based on iteration error rates that decrease initially and reach a minimum range; in our cases this happens very quickly, often after no more than 500 iterations. A higher thinning rate for the sampler allows more frequent switching in the assignments of θ to η. There were never singularity problems in running the C code for this paper's simulations, and the first random seed's Gibbs sampler output generates this method's resulting prediction. Default implementation is chosen for comparison methods, using k-fold cross-validation to tune model parameters, with k depending on computation demands.

2.3.1 Predicting the doppler function

A two-dimensional test case helps explain the intuition behind the prior and prediction using the resulting posterior distribution. These bivariate data are not likely to come from any physical system and are chosen only for demonstration purposes as a general prediction scenario with a locally linear regression function and complex covariance mixtures.

2.3.1.1 Doppler function prediction performance

To demonstrate that this method is able to predict values according to the different underlying covariance structures, we first consider a simulation of the doppler function $d(x) = \sqrt{x(1-x)}\sin(2.1\pi/(x+.05))$, for $x \in [0,1]$, from Wasserman (2004). The data consist of $N_{obs} = 2048$ points from $y_i|x_i \sim d(x_i) + N(0, 0.2^2)$ with $x_i = i/N_{obs}$. The task is to predict the doppler function from these data with minimum squared error loss. Performance is evaluated at the 5000 points $x_i^* = i/5000$. A plot of this function with the test case data is shown in Figure 2.1.

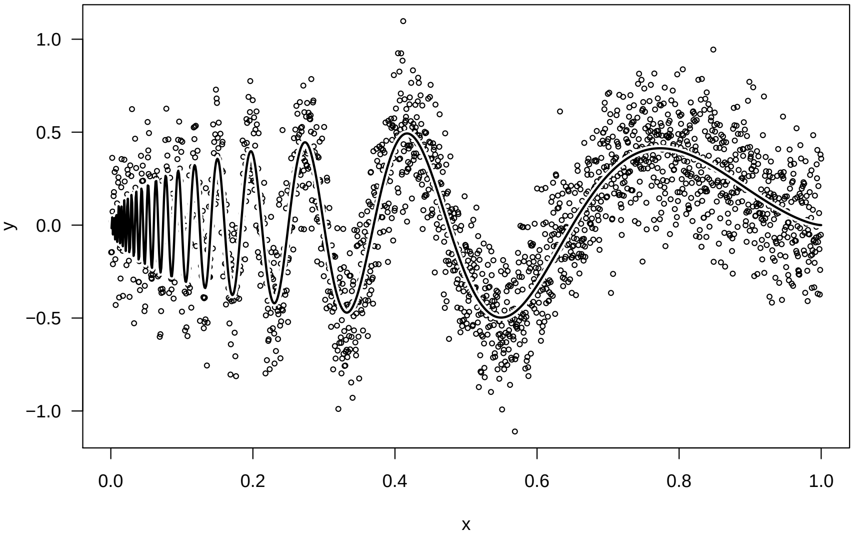

Figure 2.1 Doppler Function Test Case Data. 2048 points from the doppler function (shown as line) with added error $N(0, 0.2^2)$.

We use the mixture prior Bayesian method to get predictions for the regression function of y with $n = 30$ latent mixture components and 171 scale matrices for the prior for Σ_m. The added Gaussian noise is regenerated to make 50 different test cases. In these test cases the performance metric, sq. err., is the ratio of the sum of squared prediction errors compared with the true regression function $f(x_i)$ for observations $i = 1, \ldots, N_{test}$, $\sum_{i=1}^{N_{test}}(f(x_i) - \hat{y}_i)^2$, relative to the total sum of squares for the regression function, $\sum_{i=1}^{N_{test}}(f(x_i) - E(y_i))^2$. This error metric is proportional to the amount of unexplained variation of the true regression function at the test data observations.

Figure 2.2 shows the prediction curves for Gaussian noise generated with the first random seed. The squared error ratio for this first seed are 0.0404 for local regression and 0.0616 for the mixture prior Bayesian method. Note that the small optimal bandwidth for the local regression method is chosen due to the variability at small x values, leading to choppy predicted values for larger x. An appealing aspect of the mixture prior Bayesian method is its ability to adapt the smoothing bandwidth to the local regression function structure.

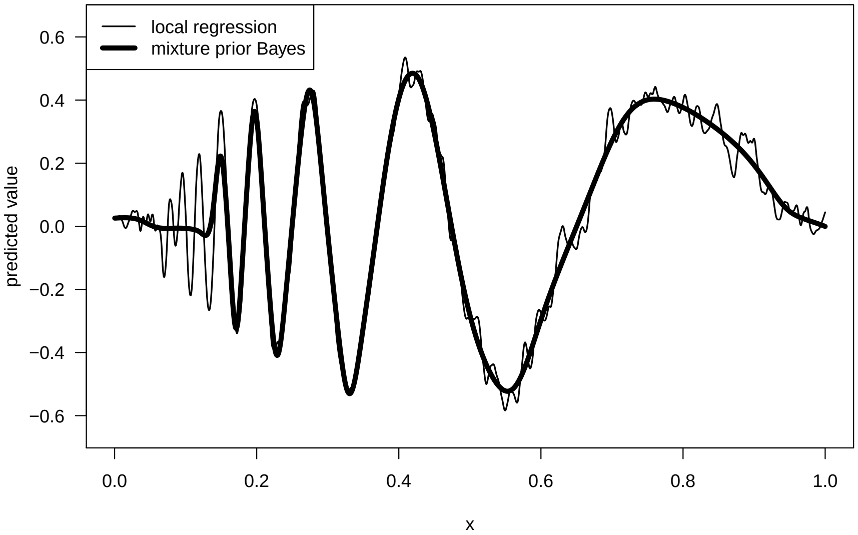

Figure 2.2 Comparison of Doppler Function Prediction Values. Prediction curves based on local regression and mixture prior Bayes. Note that the mixture prior Bayesian method provides a smoother fit of the doppler function.

Table 2.1 Unexplained Variation of the Doppler Function for 50 Random Seeds

prediction method	sq. err. mean	sq. err. s.d.
mixture prior Bayes	0.069	0.007
k-nearest neighbors ($\bar{k} = 18.6$)	0.042	0.004
local linear regression	0.038	0.003

Table 2.1 shows results of prediction of the doppler function $d(x)$ over the same 5000 test points but with 50 different sets of random errors generated for the 2048 training data points. While prediction values for the mixture prior Bayesian method have higher squared error than other local regression methods such as k-nearest neighbors using R package FNN and local regression using R package locfit, the mixture prior Bayesian performance is not poor and this method does not overfit the data.

2.3.1.2 Performance with a spurious predictor

Prediction performance of the mixture prior Bayesian method improves over other local regression methods when the system has a spurious predictor, i.e., a variable unrelated to the response. A predictor $X_2 \sim \text{Unif}(0,1)$ is added to the previous test case, and prediction values are recalculated. Table 2.2 shows results of prediction of the doppler function $d(x)$ values over the same 50 test cases as before, except now with a spurious second covariate generated from a uniform density. In this situation, performance of the mixture prior Bayesian method is substantially better than that of the local methods. It is interesting that the mixture prior Bayesian method has higher variability of unexplained error even though its average error rate is much lower than that of the other two methods. Note that the average optimal number of neighbors in k-nearest neighbors is much lower in the presence of a spurious predictor.

Table 2.2 Unexplained Variation of the Doppler Function With Spurious Predictor for 50 Random Seeds

prediction method	sq. err. mean	sq. err. s.d.
mixture prior Bayes	0.129	0.015
k-nearest neighbors ($\bar{k} = 7.7$)	0.283	0.009
local linear regression	0.248	0.009

2.3.1.3 Performance with varying mixture prior input

The performance of the mixture prior Bayesian method changes greatly with the number of input scale matrices for Σ_m. Simply reducing the number of scale matrices input into the mixture prior Bayesian method does not exactly mimic a standard Bayesian mixture model implementation. However, inputing only a single diffuse scale matrix in the mixture prior Bayesian method would result in a marginal prior for the mean and variance parameters that is approximately as diffuse as the marginal prior for a standard Bayesian mixture model. Therefore, the resulting performance degradation that occurs when using only diffuse scale matrices highlights the difficulties of the standard Bayes model mentioned in the introduction.

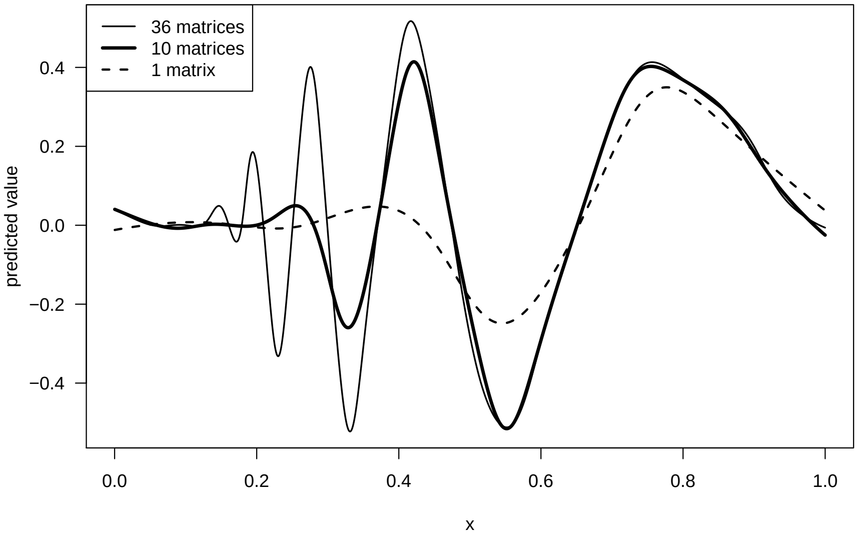

Figure 2.3 Fewer Scale Matrices Degrades Prediction Performance. The amount of empirical information provided to the Σ_m prior is reduced by decreasing the maximum number of grouping in the clustering program. A less-informative prior leads to over-smoothed prediction values.

The same 50 test cases from 2.3.1.1 are rerun with varying empirical information provided to the Σ_m prior of the Bayes model. We reduce the number of scale matrices by decreasing the maximum number of groupings found by the clustering algorithm. Table 2.3 shows prediction results with varying number of clustering algorithm cumulative groupings for the 50 test cases. The prediction curves for the first seed, shown in Figure 2.3, demonstrate what typically happens: as less empirical information is provided to the Bayes model, the resulting more diffuse prior muddles the mixture arrangement in the posterior, leading to degraded prediction performance.

Table 2.3 Unexplained Variation of the Doppler Function With Various Scale Matrices Provided for Mixture Prior for 50 Random Seeds

no. of scale matrices	1	3	10	21	36	78	171
sq. err. mean	.545	.285	.217	.137	.117	.086	.069
sq. err. s.d.	.010	.009	.032	.016	.014	.010	.007

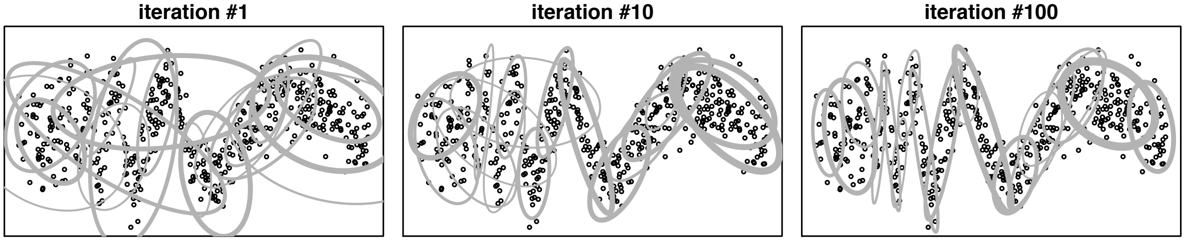

Figure 2.4 Component Parameters During Gibbs Sampler Initialization. 90% probability contours are shown for μ_m and Σ_m for any components with membership more than 10 data points. Width of contour ellipse is relative to the component mixture proportion size.

The left panel of Figure 2.4 shows 90% probability contours of the Gaussian parameters for the 24 latent mixture component parameters as the Gibbs sampler starts up. In this simulation, 171 scale matrices are provided for the prior of Σ_m. While the population mixture is not apparent in the initial Gibbs sampling of the posterior, the posterior distribution largely has the structure of the regression function and this structure is present in the sampler by the 100th iteration, shown in the right panel of Figure 2.4.

2.3.2 Comprehensive simulation study

A comprehensive simulation study of higher-dimensional simulated data is presented to show how the method performs in a variety of mixture distribution situations. This simulation study is loosely modeled after the comprehensive nonparametric regression testing in Banks et al. (2003). A regression function is created, simulated data are generated, and predictions are made using this mixture prior Bayesian method and four comparison methods. Each scenario is randomly regenerated a total of 25 times and performance of each method is assessed using a prediction error ratio.

2.3.2.1 Generation of simulated test cases

The study focuses on prediction of a continuous response whose regression function is the conditional mean function of a mixture of eight multivariate normal densities

$$E(Y|X=x) = \sum_{h=1}^{8} p_h(x)\left(\mu_{h,y} + \Sigma_{h,yx}\Sigma_{h,xx}^{-1}(x-\mu_{h,x})\right) \tag{2.5}$$

where $p_h(x)$ is the probability that an observation at x is from density h. Note that the eight probabilities $p_h(x)$ sum to one for any x, and these probabilities consider the density population proportion sizes for the eight densities. $p_h(x)$ is equal to the component h density evaluated at x times the component proportion size, and then these eight $p_h(x)$ values are rescaled so they sum to one.

Simulations use multivariate normal component parameters generated from R package MixSim (Melnykov et al., 2012). The amount of overlap can be specified when generating data with MixSim, and we choose an average overlap of $\bar{\omega} = .05$ for the simulations, which specifies low mixture component separation (Melnykov and Maitra, 2010). We also choose non-spherical densities and draw the first seven mixture proportions from Stick($\alpha = 16, \beta = 64$), with the eighth mixture assigned the remaining probability. After the Gaussian component parameters and mixture proportions are determined, data are generated with MixSim function simdataset using default choices for all other parameters.

We use the MixSim package to calculate the amount of cluster overlap in our simulated data sets. Our implementation of MixSim has a single purpose, and that purpose is to get a locally linear conditional response function calculated by equation 2.5. MixSim provides mean and covariance values μ_h and Σ_h for the eight densities of our mixture distribution. By altering the MixSim component proportions to match those from our stick-breaking proportion assignments, the average overlap $\bar{\omega}$ differs from the target of .05. Instead, the observed average overlap ranges from .052 to .062 in our test cases. The maximum overlap $\check{\omega}$ measures the highest level of overlap between the densities of the mixture. See Figure 2.5 for a plot showing calculated maximum overlap $\check{\omega}$ values for the multivariate normal data generation scenario (see 2.3.2.2 below) for the different number of predictors in the simulation study. As the dimension increases and the number of components stays fixed at eight densities, the probability that any two components exhibit high overlap decreases.

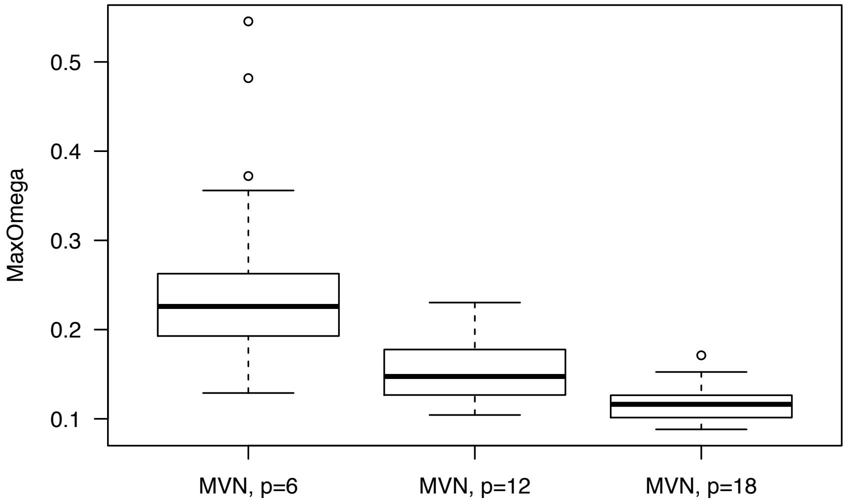

Figure 2.5 Maximum Component Overlap From MixSim-Generated Data. These boxplots show the range of the maximum overlap (MaxOmega) values $\check{\omega}$ calculated with the MixSim package function overlap. These values are calculated from all test cases of the multivariate normal data generation scenario, see 2.3.2.2. The three p values represent the number of predictors in the test cases.

2.3.2.2 Simulated test case scenarios

Testing is performed for three data dimensions (p = 6, 12, and 18 covariates), three data set sizes $N_{obs} = k(1.2)^p$ (k = 200, 500, 1250), and these three data generation scenarios:

MVN Data. Data are generated from an eight-component multivariate Gaussian mixture. This scenario is an ideal situation for the mixture prior Bayesian prediction method and we expect the mixture prior Bayesian prediction method to perform well.

Uniform X. The regression function is generated from the eight-component multivariate normal mixture, but then the predictors are regenerated from independent Uniform(0,1). The response Y equals the regression function at $X = x$ plus $\mathrm{N}(0, 0.1^2)$ noise. This scenario tests how the method performs when the multivariate normal structure in the predictors is lost and the only relationship available is the locally linear regression function. The mixture prior Bayesian method might not perform well in this case as all of the multivariate normal structure is gone except for the regression function.

Spurious. Data are generated from an eight-component multivariate Gaussian mixture as in the first scenario, except now one-third of the predictors are spurious with respect to, i.e. independent of, all other variables. This scenario tests how the methods are able to handle noisy predictors in the absence of variable selection.

2.3.2.3 Local regression methods for comparison

The method's prediction performance, as measured in squared error relative to the true regression function given the predictors, is compared with the following four methods.

Method G. The first comparison method is a likelihood-based Gaussian mixture method implemented via the MATLAB package gmdistribution using the function fitgmdist. The likelihood-based method first requires determination of the optimal number of mixture components. This optimal number is found through 5-fold cross-validation, using a certain number of restarts that scales with the square root of the dimension. Once the optimal component number is determined, then many restarts are tried in the final model. See Figure 2.6 for quartiles of the optimal number of components densities in the Gaussian mixture model. At $p = 18$ predictors, the correct number of eight subpopulations is the optimal number of mixture components for each run. The variability of optimal components greatly increases for the uniform regeneration scenario.

Method B. The second comparison method is a diffuse prior Bayesian mixture model. Instead of independence of between the mean and covariance parameters, a conditional structure in the prior of a traditional Bayesian mixture model approach may yield better

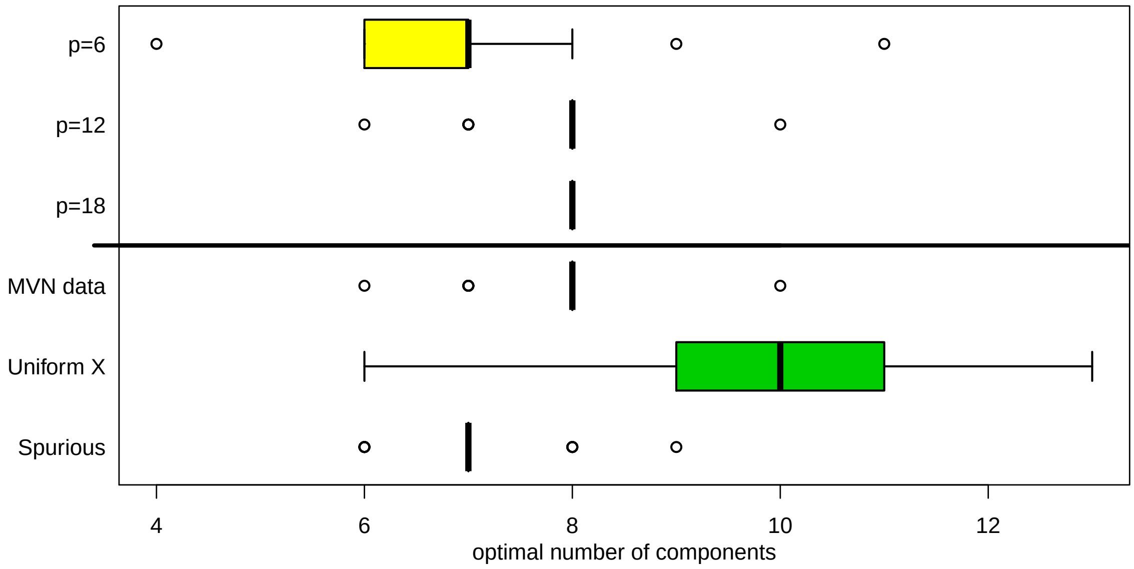

Figure 2.6 Optimal Number of Components for the Likelihood-Based Method. These are boxplots showing the determined number of components per run for the MATLAB implementation of Gaussian mixture models for prediction.

results. We implement in C a prediction method that uses the conditionally conjugate normal–inverse-Wishart prior of the classification mixture model given in Gelman et al. (2013). Good results came from using a scale matrix equal to one-half of the empirical covariance of the whole data, and that is the scale matrix used in the prior for Σ_m. All other parameters are the same as the mixture prior Bayesian implementation.

Method N. The third comparison method is k-nearest neighbors implemented in R using library FNN function knn.reg. The optimal k neighbors is determined with 20-fold cross-validation by optimizing the internal calculation for leave-one-out cross-validation of function knn.reg. See Figure 2.7 for quartiles of the optimal number of neighbors for k-nearest neighbors. The uniform regenerated cases have lower optimal numbers of neighbors than the other data-generation scenarios. The variability of optimal k does not diminish as the dimension increases.

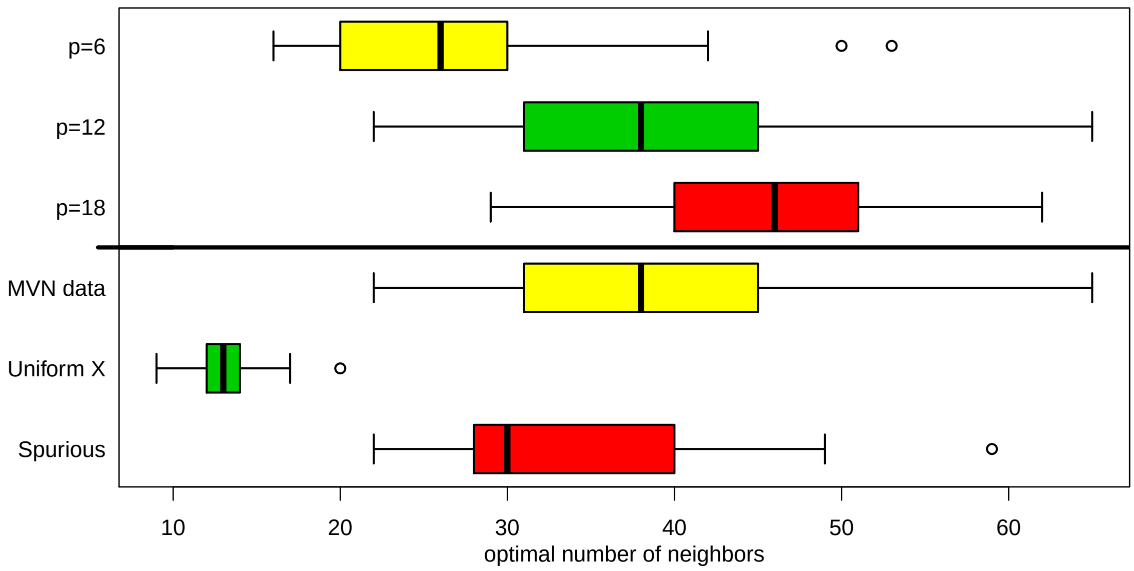

Figure 2.7 Optimal Number of Neighbors for k-Nearest Neighbors. These are boxplots for the optimal number of neighbors determined by the R implementation of k-nearest neighbors for prediction.

Method F. The fourth comparison method is random forests implemented in R using library randomForest with default parameters. As overfitting is not a significant concern for random forests, the method is allowed to run for a long time until a stopping rule decides that enough trees have been grown to minimize prediction bias. The stopping rule is satisfied when linear regression on the out-of-bag prediction error for the previous 400 trees has a positive slope, meaning that the bias reduction has largely ceased.

Others. Other methods are not reported because they do not offer better results and are not be expected to predict a linear response well. Gradient boosting models (Hastie et al., 2009) and support vector regression do not have better results than the above methods. Multivariate adaptive regression splines and locally linear regression have computational difficulties with the dimension of these data sets; besides, both do not predict well for higher-dimensional data (Banks et al., 2003).

2.3.2.4 Simulated test case results

The comprehensive test case results are summarized in two plots, Figures 2.8 and 2.9. Figure 2.8 displays results across all scenarios, data set sizes, and dimensionality. Figure 2.9 displays complete results for $p = 6$ covariates.

We determine the following findings about mixture prior Bayesian prediction performance from the simulation study results shown in Figure 2.8, grouped by comparison methods.

Likelihood-Based Methods. The mixture prior Bayesian method outperforms the likelihood-maximizing solution across all scenarios as long as the dimension is not moderately large ($p < 18$) and the data set size is not large relative to the number of predictors p. This convergence of Bayesian and frequentist solutions as sample size increases is sensible because the prior distribution is dominated by the likelihood as the amount of data increases. It is reasonable that the Bayesian solution is slightly better, even with plentiful data, as the frequentist solution uses only one arrangement of mixture components while the Bayesian solution can average several equally good, though different, arrangements via the Dirichlet process prior. For 18 predictors, the data set size ($N_{obs} = 13{,}312$) is large enough for the frequentist solution to perform similarly. This indicates that the chosen scale of 1.2 for the data size inflation formula, $N_{obs} = k(1.2)^p$, may be too large. We conjecture a constant of 1.1 might give more comparable results across dimensionality. An alternative explanation is that the reduced maximum overlap values at higher dimensions leads to both easier prediction and similarity in results.

Standard Bayes Implementation. The mixture prior Bayesian method outperforms prediction values using a standard Bayesian mixture model when the data set is smaller and when there are spurious predictors. When the predictors are regenerated from independent uniform distributions, the performance of the two priors is similar. Also, the predictions of the priors converge as the data set size increases. Again this is an indication the scale in the data size inflation formula is too large, and if this scale is

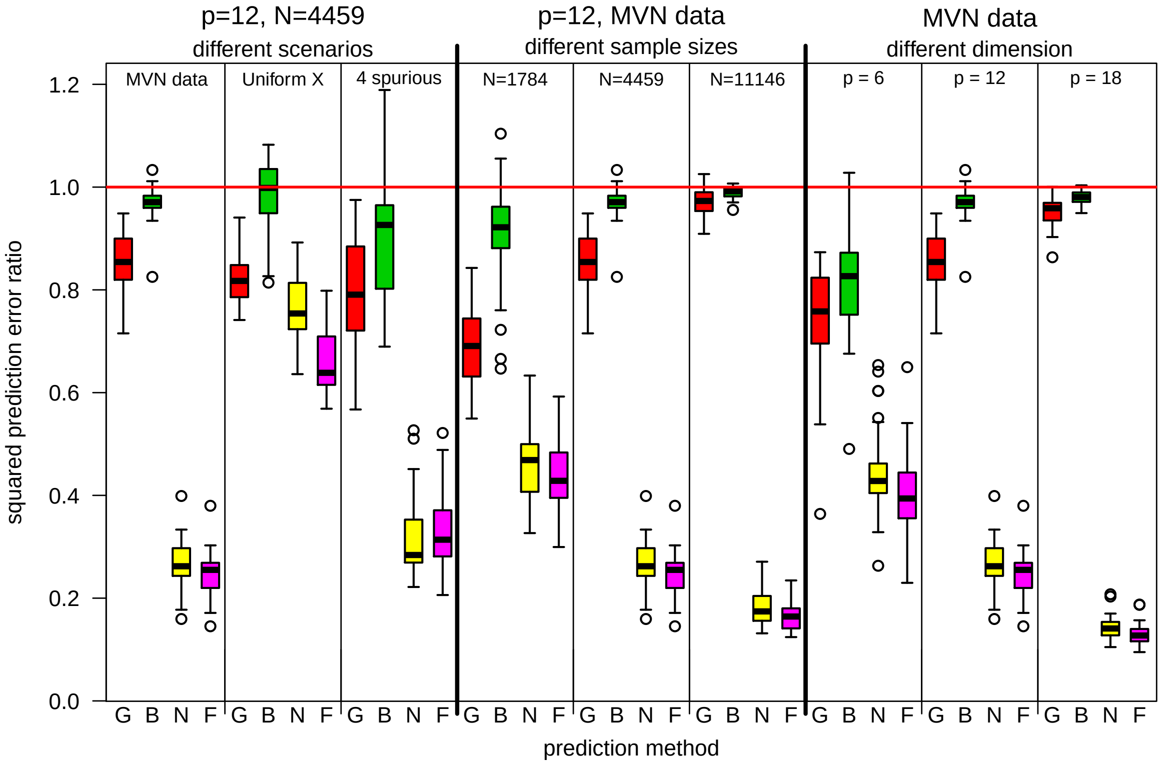

Figure 2.8 Mixture Prior Bayes Prediction Error Ratios With Comparison Methods for Comprehensive Simulation Study. This plot displays the ratio of the mixture prior Bayes squared prediction error to that of other methods, with results below one being favorable to the mixture prior Bayesian method. The first section is a comparison across data generation scenarios, the second section is a comparison across data set sizes, and the third section is a comparison across number of predictors p. The comparison methods are (G) likelihood-based Gaussian mixture models, (B) standard Bayesian implementation, (N) k-nearest neighbors, and (F) random forests. Note that the results in the left panel of the left section (comprising four boxplots) appears in the middle panel in the other two sections.

smaller, perhaps a value of 1.1 instead of 1.2, the results across dimensionality might be comparable.

Other Methods. The mixture prior Bayesian method outperforms both k-nearest neighbors and random forests consistently. This performance advantage is not surprising due to the local linearity in the regression function. These other methods perform best when the predictors are regenerated from uniform distributions. It is interesting to note that as the data size increases, the Gaussian mixture model based methods use the new data more efficiently and yield greater prediction improvement over these two methods.

Figure 2.9 shows similar findings to Figure 2.8. In addition, we note that the mixture prior Bayesian method performance versus other methods is fairly constant across data set sizes for the uniform regeneration scenario; however, the other two scenarios show improvement versus k-nearest neighbors and random forests and degradation versus the likelihood-based Gaussian mixture model and a standard Bayesian implementation. We also note that as data set size increases in the presence of spurious predictors, comparative results have more variability—contrast the left section versus the right section in Figure 2.9.

2.3.2.5 Simulated test case computing times

Computing times averaged among the 75 runs for the medium-sized test sets are shown in Table 2.4. Results are averaged among all scenarios due to high variability of job completion times on the server—numbers listed in the table are only rough estimates. These results display that for these data sets k-nearest neighbors and random forests can get poor results quickly. The Bayesian methods perform prediction tasks faster than their frequentist counterpart as Bayesian methods do not have to determine the optimal number of mixture components. Note that the Gaussian mixture model is run on a different machine with about half the memory but similar processor speeds as the server running the other methods.

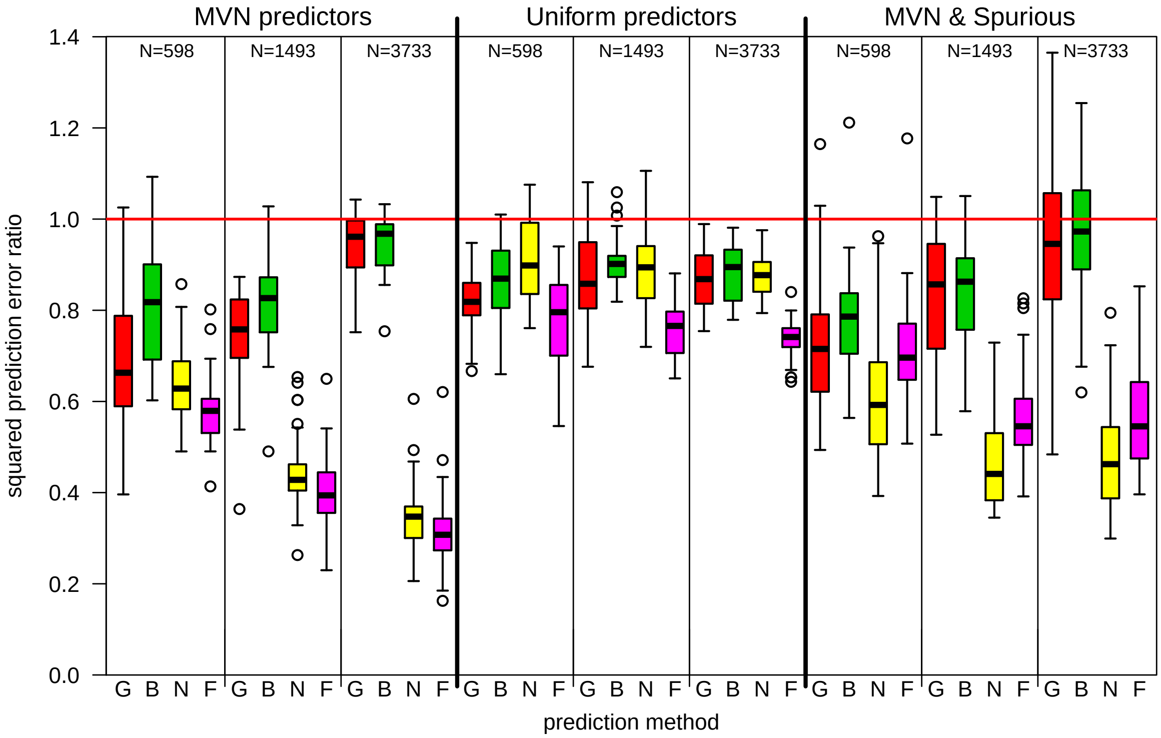

Figure 2.9 Prediction Error Ratios For Six Predictors. This plot displays the ratio of the mixture prior Bayesian squared prediction error to that of other methods, with results below one being favorable to the mixture prior Bayesian method. The data set size increases across frames in each section. The left section is a comparison among multivariate normal data, the middle section is a comparison when the predictors are regenerated from independent uniform distributions, and the right section is a comparison when a third of predictors are spurious.

Table 2.4 Average Computation Times, in Minutes, for the Prediction Methods

prediction method	$p = 6$	$p = 12$
mixture prior Bayes	6.0	55
Gaussian mixture model	24	146
standard Bayes	5.1	23
k-nearest neighbors	0.1	0.9
random forest	0.2	1.5

The mixture prior Bayesian method has computational complexity $O(N_{obs} \cdot n \cdot p^2)$, where N_{obs} is the number of observations in the data set, n is the number of mixture components, and p is the number of predictors. The method may not be computationally prohibitive as some local smoothing methods such as local linear regression and generalized additive models.

2.3.3 Application to machine learning data sets

We now study prediction performance of the mixture prior Bayesian prediction method applied to four standard regression data sets from two machine learning repositories: the University California–Irvine (UCI) Machine Learning Repository (Bache and Lichman, 2013) and StatLib, hosted by the Department of Statistics at Carnegie Mellon University. Table 2.5 summarizes the size of each data set and results of three prediction methods, and afterwards the data sets and performance is elaborated. In the table, p represents the number of predictors derived from the original data set for analysis.

2.3.3.1 Body fat data set

The body fat data set is a collection of measurements for 252 men that have had their body fat percentage calculated by underwater weighing (Penrose et al., 1985). The underwater measurement technique is more accurate than other standard techniques but hard to implement, so it is advantageous to have an accurate body fat assessment

Table 2.5 R^2 for Prediction of Four Machine Learning Data Sets Using Mixture Prior Bayes, k-Nearest Neighbors, and Random Forests

data set	N_{obs}	p	MPB	kNN	RF
body fat	143	12	.747	.626	.679
California housing	20460	8	.819	.803	.828
concrete strength	1030	4	.446	.325	.402
yacht hydrodynamics	308	2	.977	.961	.968

technique involving only body measurements. The collection of measurements include age, weight, height, and ten circumferences, such as for wrist and ankle. All of the 13 predictors and response are continuous, and the first 143 observations of the data set are reserved for training and the remaining 109 observations for testing. The predictive equations in the original research are formed using the first 143 observations.

In analysis, the variable weight is highly correlated with many other predictors, especially hip circumference, and removing weight from the analysis improves prediction for all tested methods. Another modification that improves prediction is to replace outliers with their boxplot fence values. On this modified data set of 12 predictors, the mixture prior Bayesian method has the lowest squared prediction error of all methods tried, with $R^2 = .747$. No tuning is done in obtaining prediction values. The closest method in predictive performance is LASSO that attained $R^2 = .735$, with 10 degrees of freedom, selected with Mallow's C_p to avoid overfitting.

Although the data set is practically globally linear, shown by the fact a multivariate regression fit achieves $R^2 = .729$, there are enough regions of distinct local linearity to make the mixture prior Bayesian prediction method appealing. The likelihood-based Gaussian mixture model has $R^2 = .727$, worse than multivariate regression. Likelihood-based methods have the disadvantages associated with simultaneous estimation of a large number of model parameters, limiting the number of mixtures available for such a small data set, and only a single arrangement of the limited mixture is used for prediction. The mixture prior Bayesian method is well-suited for prediction with this data set because the data set is small, the regression function has a linear structure, the predictors are all continuous, and there are too many predictors for some local regression methods such as local linear regression or multivariate adaptive regression splines.

2.3.3.2 California housing data

A test case with eight predictors, demonstrated in Hastie et al. (2009), is the California housing data set from Pace and Barry (1997), consisting of aggregate housing price data from 20,460 neighborhoods in California. The response to be predicted is the median house value in the neighborhood. The predictors capture important demographic characteristics along with median income, housing density and occupancy, geographic location, and house size.

Our analysis of this data set follows the optimal linear regression from Pace and Barry (1997), except the mixture prior Bayesian method now allows a linear analysis and include the latitude and longitude information in a way not previously possible. We randomly divide the data into a two-thirds training set and one-third test set. The mixture prior Bayesian method algorithm runs for 200 samples, with a thinning rate of 2 iterations per sample and a burn-in period of 200 iterations.

Random forests achieve an R^2 of .828 while the mixture prior Bayesian method has an R^2 of .819. A straight average of these two predictions attains an R^2 of .835, and this points out another advantage of this prediction technique. The mixture prior Bayesian structure is very different than many successful prediction methods, in this test case allowing local linear modeling of geographic coordinates. Our mixture prior Bayesian prediction method could be valuable in forming prediction ensembles. For reference k-nearest neighbors prediction has $R^2 = .803$ for an optimal $k = 8$. The data set is not globally linear as shown by the highest ridge regression R^2 being .665.

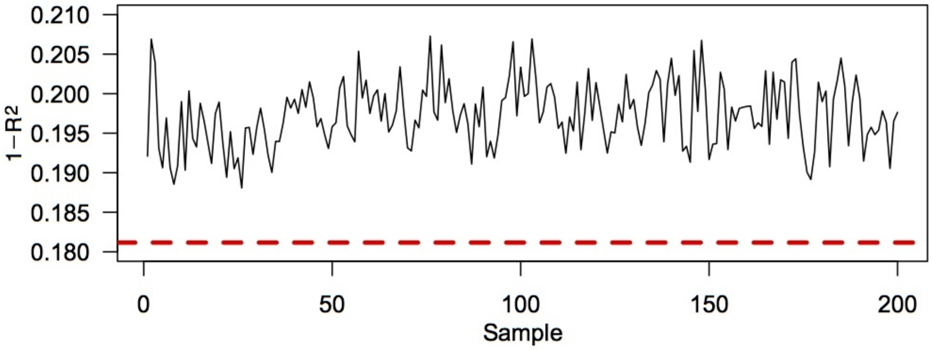

Figure 2.10 Iterate Squared Error Rates for Gibbs Sample for the California Housing Data Analysis. The dashed red line shows the overall prediction squared error rate of .181. Per iteration error rates are not improving, and a longer Gibbs sample confirms no further gains are possible. The sampler appears to be in the main part of the posterior.

See Figure 2.10 to see the per-iteration mean squared error rates for the sample of size 200. Shown are iterations 201 through 599, as every other iteration is not saved by the sampler. Note that even with eight predictors the Gibbs sampler reaches the main part of the posterior when the burn-in period ends and the Gibbs sampler starts saving iterations.

2.3.3.3 Concrete compressive strength data set

The UCI concrete data set (Yeh, 1998) consists of 1030 observations with eight predictors and a quantitative response. The compressive strength of concrete is to be predicted. For our analysis we remove four of the eight predictors to simplify the data set. Breaking up the data set into four folds and predicting each fold by training on the other three folds, we achieve $R^2 = .446$ with the mixture prior Bayesian method. This is not a globally linear data set, but is strongly locally linear—such a data set is a good candidate for this mixture prior Bayesian method. Multivariate adaptive regression splines did the best of other methods we tried, with $R^2 = .434$, demonstrating the strong locally linear structure.

2.3.3.4 Yacht hydrodynamics data set

Yacht hydrodynamics is a regression data set in the UCI repository containing residuary resistance measurements from an experiment on 22 different hull forms (Gerritsma et al., 1981). Each hull has resistance measurements associated with 14 different Froude numbers. The other five predictors are constant within each hull's 14 measurements, even though there is a response for each Froude number. Therefore the analysis includes as predictors only the Froude number and the beam-draught ratio, the latter having the most variation among the hulls. The removal of the four predictors does not degrade the prediction ability in general, due to the fact that 22 sets of Froude observations among 5 predictors are too sparse to be useful. In testing, the 22 hulls are split into 4-folds, and three folds train the prediction model to predict observations in the fourth fold. The mixture prior Bayesian method does better than other tested methods in predicting the resistance, attaining $R^2 = .977$.

2.4 Discussion

This paper has introduced a Bayesian prediction method based on Gaussian mixture models with mixture priors applied to the component parameters. Such a model offers potential improved prediction of regression functions with a locally linear structure, especially when limited data are present or the dimension is large enough to rule out computationally demanding methods. A mixture prior on the component Gaussian parameters offers a way for Bayesian mixture models to be competitive in prediction analytics. We have shown how the posterior distribution has full conditionals thereby allowing computationally-efficient Gibbs sampling.

There could be an alternative implementation of the ideas of this paper that involves less tuning. We can set both stick-breaking parameters, α and β, to one and similarly eliminate the Wishart degrees of freedom parameter by setting ρ to $p + 3$. This reduction leaves only two tuning levers: the number of model mixture components n and the user-provided scale matrices M_l. A further reduction in complexity of the method can be achieved by replacing the user-provided scale matrices with a collection of spherical covariance matrices. This collection of spherical covariance matrices should have a suitable range of variance values, which can easily be achieved given that a collection of 200 scale matrices will not considerably increase computation times. Note that to implement this alternative method, standardization of the predictors is necessary.

The simulation study of this paper shows that situations exist where the mixture prior Bayesian method is competitive versus other prediction methods. With limited data, the mixture prior Bayesian method is not restricted to a single arrangement of mixtures, and this flexibility offers advantages over other prediction methods using Gaussian mixture models. While the mixture prior Bayesian method, standard Bayesian implementation, and likelihood-based methods based on Gaussian mixture models all give prediction values that will converge as data size increases, the mixture prior Bayesian method appears to predict better than other methods when the data sets have a small number of observations or spurious information. When the predictors are regenerated from uniform distributions, the mixture prior Bayesian method maintains a small advantage over Gaussian mixture models in the simulation study. The mixture prior Bayesian method can manage a greater number of predictors than some other local regression methods.

Analysis of standard regression data sets demonstrates that the mixture prior Bayesian prediction method is competitive with other methods when the regression function has a locally linear structure. The California housing data shows that the mixture prior method allows local linear modeling of information not used in other prediction methods, such as neighborhood geographic location. Even if the mixture prior does not give the best prediction values, this method is different enough from other methods to be potentially valuable in a prediction ensemble.

2.5 Appendix: Derivation of Posterior Conditional Distributions

The conditional posterior distribution for the parameters, $p(\theta, \eta, \phi|z)$, from (2.4) becomes

$$p(\eta, \phi, \theta|z) \propto \prod_{i=1}^{N_{obs}} \left(N(z_i|\theta_i)\, \mathbb{I}\left[\theta_i \in \{\eta_1, \ldots, \eta_n\}\right] \prod_{m=1}^{n} \pi_m(\phi)^{\,\mathbb{I}[\eta_m = \theta_i]} \right) \\ \times \left(\prod_{m=1}^{n} g(\eta_m) \right) \left(\prod_{m=1}^{n-1} \mathrm{Beta}(\phi_m|\alpha, \beta) \right). \tag{2.6}$$

Conditional distributions can be found from (2.6) for the individual θ_i, ϕ_m, μ_m, and Σ_m to allow Gibbs sampling of the posterior (Gelman et al., 2013). The exact posterior distribution, in terms of $\theta_1, \ldots, \theta_{N_{obs}}$, $\phi_1, \ldots, \phi_{n-1}$, $\mu_1, \ldots, \mu_n$, $\Sigma_1, \ldots, \Sigma_n$, hyperparameters α, β, ρ, and Γ, and user-provided covariance matrices $M_1, \ldots, M_L$ and weights $w_1, \ldots, w_L$, is proportional to

$$\prod_{m=1}^{n} \left(|\Sigma_m|^{-\frac{n_m}{2}} e^{-\frac{1}{2}\operatorname{tr}(\Sigma_m^{-1} S_m)} \sum_{i=1}^{N_{obs}} e^{-\frac{1}{2}(z_i - \mu_m)^T \Gamma^{-1} (z_i - \mu_m)} \right. \\ \left. \times \sum_{l=1}^{L} w_l \, |M_l|^{\frac{\rho}{2}} \, |\Sigma_m|^{-\frac{\rho+p+2}{2}} \, e^{-\frac{1}{2}\operatorname{tr}\left(M_l \Sigma_m^{-1}\right)} \right) \\ \times \left(\prod_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i \in \{\eta_1, \ldots, \eta_n\}\right] \right) \left(\prod_{m=1}^{n-1} \phi_m^{n_m + \alpha - 1} (1 - \phi_m)^{n_m^{+} + \beta - 1} \right), \tag{2.7}$$

where $n_m = \sum_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i = \eta_m\right]$, $n_m^{+} = N_{obs} - \sum_{j=1}^{m} n_j$, and $S_m = \sum_{i:\theta_i = \eta_m} (z_i - \mu_m)(z_i - \mu_m)^T$.

We derive the conditional distributions from (2.7) by analyzing the distribution of each θ_i, μ_m, Σ_m, or ϕ_m with all other variables held constant. A Gibbs sampler designed with these conditionals provides a numerical estimate of the posterior distribution for $p(\eta, \pi, \theta|z)$ of (2.6) (Gelfand and Smith, 1990). After initialization of ϕ and η, the Gibbs sampler operates by:

1. updating each θ_i, one at a time, as a random draw from the discrete distribution $\{\eta_1, \ldots, \eta_n\}$ with probabilities proportional to $\pi_m \, N(x_i, y_i|\mu_m, \Sigma_m)$,

2. updating each ϕ_m for $m = 1, \ldots, n-1$, one at a time, from its conditional posterior distribution, Beta$(n_m + \alpha, n_m^+ + \beta)$,
3. updating each η_m, one at a time:
   (a) first draw μ_m from a multivariate normal distribution, according to its conditional parameters,
   (b) then draw Σ_m from an inverse-Wishart distribution with ρ degrees of freedom, according to its conditional parameters.

The posterior distribution for the parameters $\phi_1, \ldots, \phi_{n-1}$, $\mu_1, \ldots, \mu_n$, $\Sigma_1, \ldots, \Sigma_n$ and $\theta_1, \ldots, \theta_{N_{obs}}$ is shown in equation (2.7). The following is a derivation of the conditional distribution for each of these parameters.

2.5.1 Conditional posterior of θ_i

The N_{obs} θ parameters are conditionally independent of each other. The conditional distribution in (2.7) reduces to a constant (with respect to θ_i) times

$$\prod_{m=1}^{n} \left(\left| \Sigma_m^{-1} \right|^{\frac{1}{2}} e^{-\frac{1}{2}(z_i - \mu_m)^T \Sigma_m^{-1} (z_i - \mu_m)} \pi_m(\phi) \right)^{\mathbb{I}[\eta_m = \theta_i]} \mathbb{I}\left[\theta_i \in \{\eta_1, \ldots, \eta_m\}\right], \qquad (2.8)$$

which is a multinomial distribution among $\eta_1, \ldots, \eta_n$ according to the probabilities

$$\big(\pi_m \, N(z_i | \mu_m, \Sigma_m)\big) / \Big(\sum_{m'=1}^{n} \pi_{m'} \, N(z_i | \mu_{m'}, \Sigma_{m'}) \Big).$$

2.5.2 Conditional posterior on μ_m

Each of $\mu_1, \ldots, \mu_n$ are independent due to its factorization. For any particular μ_m, its conditional distribution in (2.7) is proportional to

$$e^{-\frac{1}{2} \sum_{i'=1}^{N_{obs}} \mathbb{I}[\theta_{i'} = \eta_m] (z_{i'} - \mu_m)' \Sigma_m^{-1} (z_{i'} - \mu_m)} \sum_{i=1}^{N_{obs}} e^{-\frac{1}{2}(z_i - \mu_m)' \Gamma^{-1} (z_i - \mu_m)}. \qquad (2.9)$$

The left term of (2.9) includes only z_i with θ_i currently assigned to component m in the exponent's summation. After removing the constant $\exp(-\frac{1}{2} \sum_{i'=1}^{N_{obs}} \mathbb{I}\left[\theta_{i'} = \eta_m\right] z_{i'}' \Sigma_m^{-1} z_{i'})$

from all terms in the sum in (2.9), the left term reduces to $\exp(n_m \bar{z}_m' \Sigma_m^{-1} \mu_m - n_m \mu_m' \Sigma_m^{-1} \mu_m / 2)$, with $n_m \bar{z}_m = \sum_{i'=1}^{N_{obs}} \mathbb{I}\left[\theta_{i'} = \eta_m\right] z_{i'}$. The right term of (2.9) expands to $\sum_{i=1}^{N_{obs}} \exp(-\frac{1}{2} z_i' \Gamma^{-1} z_i + z_i' \Gamma^{-1} \mu_m - \frac{1}{2} \mu_m' \Gamma^{-1} \mu_m)$.

Rearranging the exponential terms, the posterior conditional density of μ_m is proportional to the z_i-mixture of N_{obs} multivariate Gaussian densities

$$\sum_{i=1}^{N_{obs}} \left(e^{-\frac{1}{2}\mu_m' \left(\Gamma^{-1} + n_m \Sigma_m^{-1}\right)\mu_m}\right) \left(e^{-\frac{1}{2} z_i' \Gamma^{-1} z_i}\right) \left(e^{\left(z_i' \Gamma^{-1} + n_m \bar{z}_m' \Sigma_m^{-1}\right)\mu_m}\right). \tag{2.10}$$

For each z_i multivariate normal density kernel in this mixture, the variance is $\left(\Gamma^{-1} + n_m \Sigma_m^{-1}\right)^{-1}$ and the mean is the variance times $(\Gamma^{-1} z_i + n_m \Sigma_m^{-1} \bar{z}_m)$. To get this result, let $A = \left(\Gamma^{-1} + n_m \Sigma_m^{-1}\right)^{-1}$ and $b = \Gamma^{-1} z_i + n_m \Sigma_m^{-1} \bar{z}_m$, then each μ_m density in (2.10) is proportional to $e^{-\frac{1}{2}(\mu_m - Ab)' A^{-1} (\mu_m - Ab)}$ within each z_i density kernel.

As the distribution for μ_m is a mixture of multivariate normal density kernels, the Gibbs sampler will need to first select a kernel from this mixture. The kernel selected should be equal to the marginal density of (2.10) with respect to the z_i's of the mixture. This marginal density for each kernel is attained by integrating out the μ_m over each z_i kernel.

Multiply (2.10) by $e^{\frac{1}{2}(z_i' \Gamma^{-1} + n_m \bar{z}_m' \Sigma_m^{-1})(\Gamma^{-1} + n_m \Sigma_m^{-1})^{-1}(\Gamma^{-1} z_i' + n_m \Sigma_m^{-1} \bar{z}_m)}$, then the μ_m terms integrate to some constant with respect to the kernel variable z_i.

The marginal distribution is proportional to, with respect to z_i for each kernel

$$e^{-\frac{1}{2} z_i' \Gamma^{-1} z_i + \frac{1}{2} z_i' \Gamma^{-1} (\Gamma^{-1} + n_m \Sigma_m^{-1})^{-1} \Gamma^{-1} z_i} e^{n_m z_i' \Gamma^{-1} (\Gamma^{-1} + n_m \Sigma_m^{-1})^{-1} \Sigma_m^{-1} \bar{z}_m} \tag{2.11}$$

The left exponential term is equivalent to $e^{-\frac{1}{2} z_i' \Gamma^{-1} \left[\Gamma(\Gamma^{-1} + n_m \Sigma_m^{-1}) - \mathbb{I}\right] (\Gamma^{-1} + n_m \Sigma_m^{-1})^{-1} \Gamma^{-1} z_i}$, which reduces to $e^{-\frac{n_m}{2} z_i' \Gamma^{-1} (\Gamma^{-1} + n_m \Sigma_m^{-1})^{-1} \Sigma_m^{-1} z_i}$.

Multiply each term by $e^{-\frac{n_m}{2} \bar{z}_m' \Gamma^{-1} (\Gamma^{-1} + n_m \Sigma_m^{-1})^{-1} \Sigma_m^{-1} \bar{z}_m}$, which is constant with respect to z_i, and (2.11) reduces to

$$e^{-\frac{n_m}{2} (z_i - \bar{z}_m)' \Gamma^{-1} (\Gamma^{-1} + n_m \Sigma_m^{-1})^{-1} \Sigma_m^{-1} (z_i - \bar{z}_m)} \tag{2.12}$$

The kernel is chosen proportional to the marginal kernel probabilities shown in (2.12) for $i = 1, \ldots, N_{obs}$.

2.5.3 Conditional posterior of Σ_m

Each Σ_m in (2.7), independent of the other component covariances, has the following mixture distribution, composed of L inverse-Wishart kernel densities:

$$\sum_{l=1}^{L} w_l \, |M_l|^{\frac{\rho}{2}} \, |\Sigma_m|^{-\frac{n_m+\rho+p+2}{2}} \, e^{-\frac{1}{2}\operatorname{tr}\left((M_l+S_m)\Sigma_m^{-1}\right)} \tag{2.13}$$

Each M_l-kernel density follows an inverse-Wishart distribution for Σ_m with $n_m+\rho$ degrees of freedom and scale matrix $(M_l + S_m)^{-1}$ (Gelman et al., 2013).

To find the kernel density of this discrete distribution to select in the Gibbs sampler, we find the marginal distribution of each term in the sum of (2.13). This is found by integrating out the Σ_m in each term. Each term has the form

$$w_l \, |M_l|^{\frac{\rho}{2}} \, \frac{|M_l + S_m|^{\frac{n_m+\rho}{2}}}{|M_l + S_m|^{\frac{n_m+\rho}{2}}} \, |\Sigma_m|^{-\frac{n_m+\rho+p+2}{2}} \, e^{-\frac{1}{2}\operatorname{tr}\left((M_l+S_m)\Sigma_m^{-1}\right)} \tag{2.14}$$

Integrating Σ_m out of (2.14), using unity of the inverse-Wishart density, leaves the quantity

$$w_l \frac{|M_l|^{\frac{\rho}{2}}}{|M_l + S_m|^{\frac{n_m+\rho}{2}}} \tag{2.15}$$

and this is the proportional kernel selection probability, $l = 1, \ldots, L$. Kernel l is selected in the Gibbs sampler according to (2.15).

CHAPTER 3. CLASSIFICATION WITH A DATA-DERIVED MIXTURE PRIOR BASED ON HIERARCHICAL BAYES GAUSSIAN MIXTURE MODELS

A paper in preparation

Cory L. Lanker[1,2], Stephen B. Vardeman[1], Max D. Morris[1], Kenneth J. Ryan[3],
Mark V. Culp[3]

Abstract

This paper outlines a flexible classification method based on mixture distributions for data whose class probability function has a locally linear structure. The method extends the data-derived prior previously implemented for regression to the classification context. The underlying probability model is a Dirichlet process model that is a mixture of multivariate Gaussian distributions. We invent a latent continuous response to use in this mixture model, and classification values for new observations come from the conditional posterior density of this latent variable. Mixture priors of Gaussian and inverse Wishart distributions comprise the prior on the Dirichlet mixture components. A standard normal prior is applied to the latent response with user-provided mean values from a smoothed version of the class data. Prediction of the class probability function comes directly from the posterior distribution via Gibbs sampling. A comprehensive

[1] Graduate student, University Professor, and Professor, respectively, Department of Statistics, Iowa State University.

[2] Primary researcher and author.

[3] Associate Professor, Department of Statistics, West Virginia University.

simulation study with a variety of scenarios demonstrates the method's ability to classify new data. The method outperforms other classifiers in predicting the Banknote Authentication data set with a minimum of required tuning.

3.1 Introduction

In classification problems, a first analysis step is determination of whether the class probability function is global or local in structure. The proper approach depends on the extent of localization of the classes. The problem addressed in this paper is the classification of new data given a set of class data that exhibits strong local relationships with continuous predictors. An approach to solve such problems is mixture modeling. Mixture modeling composes a population model out of various subpopulations having different class probability relationships. Mixture models are natural to consider when a population is composed of two or more subpopulations, and these types of population distributions are common in many scientific disciplines (Kang and Ghosal, 2009). We employ Gaussian mixture distributions that can approximate many types of population densities and strike a balance between modeling smoothness and overfitting the local linearities (Wade et al., 2014).

This paper outlines use of a mixture prior estimated from the data for use in Gaussian mixture models as a flexible classification method when the response is of two classes. The proposed classification method is an extension of classification with hierarchical Bayes Gaussian mixture models of Gelman et al. (2013) by constructing a mixture prior for the model's Gaussian parameters. A mixture prior leads to prediction improvement in some contexts, and similar improvements are demonstrated in this paper for classification. The exact local structure of the population does not need to be known with this method, as the nonparametric Bayes approach does not require estimation of the number of mixture densities. This approach uses a Dirichlet process prior for the mixture proportion sizes.

To get realizations from this prior we use the stick-breaking construction of Sethuraman (1994).

This implementation is an extension of the mixture prior prediction model except for the handling of the response variable. An obstacle to using Gaussian mixture models for classification is that the multivariate normal densities cannot be used when only the class of the response is known. To solve this problem, we add a latent variable for the response to the model. This latent variable has a Gaussian prior with user-provided mean values and common variance. In this paper, these user-provided mean values for the latent response are the normal quantiles of smoothed class probabilities. The existence of conditional posterior distributions for the model variables maintains efficient Gibbs sampling used for classification of new data.

A variety of test cases demonstrate performance when applying this mixture prior Bayes mixture model to classification problems. Using this method on class data generated from the bivariate doppler function—a function with strong local linearity—demonstrates that the mixture prior adapts the estimated class probability function to the underlying local structure. We present a comprehensive simulation study of class data generated from a mixture of eight multivariate normal class probability functions. The study has various data-generating scenarios using six and twelve predictors, with low and high amounts of mixture overlapping. Results of the study show the mixture prior method is competitive with local classification methods. We also show that this classification method outperforms other classifiers in predicting the Banknote Authentication data set with a minimum of required tuning.

In this paper, Section 3.2 formally presents the problem, outlines how the mixture prior Bayes model can be applied to the classification context, and discusses the details of implementation of an algorithm that is a locally linear smoother for a class probability function. Section 3.3 explains how the method works with classification simulations based on the doppler function, a variety of simulated test cases, and a real data set, comparing performance with other local classification techniques. Section 3.4 concludes the paper with discussion of this method in the classification context.

3.2 Classification Using Mixture Models

Classification can be performed using a joint probability model for multivariate input x and output $w \in \{-1, +1\}$. Assume that w is what we observe of a latent continuous y, where $w = \mathbb{I}[y \geq 0] - \mathbb{I}[y < 0]$. This input x is a random vector representing p known continuous quantities.

3.2.1 Classification problem setup

Consider a population model of the joint distribution of (x, y) as a mixture of H components, with $x \in \mathbb{R}^p$ and $y \in \mathbb{R}$, but we only observe whether y is greater than 0 through w. For each component $h = 1, \ldots, H$, the joint Gaussian distribution $N(x, y|\xi_h)$ depends on parameter vector ξ_h, made up of a mean parameter of size $p+1$ and covariance matrix of size $(p + 1) \times (p + 1)$. Define parameter λ as the collection of population proportions λ_h, with $\sum_{h=1}^{H} \lambda_h = 1$. The density of the sampling distribution of an observation (x_i, y_i) from this population is $p(x_i, y_i|\lambda, \xi) = \sum_{h=1}^{H} \lambda_h \, N(x_i, y_i|\xi_h)$. Just as in the analogous prediction model, this classification model allows a flexible classifier that captures variation in the mean and covariance structures between different groups of the population. The population model allows an adaptive relationship between the predictors x and the latent response y.

We have N_{obs} observations (x_i, w_i) and wish to predict the class w^* for a new x^*. Model the parameter vector for each observation with the continuous latent response y and parameter θ_i. The observation (x_i, y_i) has the Gaussian distribution $x_i, y_i|\theta_i \sim N(x_i, y_i|\theta_i)$, and with the requirement that $\mathbb{I}[w_i y_i \geq 0]$ for all i, leads to the model likelihood for these data $f(x, y|\theta) \propto \prod_{i=1}^{N_{obs}} N(x_i, y_i|\theta_i)\, \mathbb{I}\,[w_i y_i \geq 0]$, where θ represents all assignments of the observations' Gaussian parameters.

Consider the following joint probability model to approximate the defined population mixture distribution. This approximating model is a multivariate Gaussian mixture of n components, and we assume that x and y are jointly Gaussian for tractability. For each component of this mixture, $m = 1, \ldots, n$, let parameter η_m denote the component Gaussian parameters. Parameter η_m consists of three parameters: first, a mean vector for x, μ_m; second, a mean for the latent y, ψ_m; and third, a covariance matrix for (x, y), denoted Σ_m. Parameter π_m represents the mixture proportion size, with $\sum_{m=1}^{n} \pi_m = 1$. Parameter θ_i is distributed discretely among the component parameters η_m according to probability π_m. The joint distribution of θ and latent variables π and η has density $\prod_{i=1}^{N_{obs}} \Big(\mathbb{I}\,[\theta_i \in \{\eta_1, \ldots, \eta_n\}] \prod_{m=1}^{n} \pi_m^{\mathbb{I}[\eta_m = \theta_i]} \Big)$.

We want our model $\sum_{m=1}^{n} \pi_m\, N(x, y|\eta_m)$ to approximate the truth $\sum_{h=1}^{H} \lambda_h\, N(x, y|\xi_h)$ as closely as possible. This model has $n-1$ free parameters for π, p parameters for each of the n mean vectors μ, n parameters for the ψ variables, and $\frac{1}{2}(p+1)(p+2)$ parameters for each of the n symmetric, positive definite covariance matrices Σ. There are a total of $\frac{1}{2}n(p+2)(p+3) - 1$ parameters in this approximating mixture model.

With estimates for model parameters π and η, however attained, prediction of the class for a new observation x^* follows from the conditional Gaussian density for the latent response $f(y^*|x^*)$. Classification at x^* depends only on mixture component parameters and does not depend on the θ assignments or latent y values. This conditional density is the mixture of the Gaussian densities with mean $\psi_m + \Sigma_{m,yx}\Sigma_{m,xx}^{-1}(x^* - \mu_m)$ and variance $\sigma_{m,y}^2 - \Sigma_{m,yx}\Sigma_{m,xx}^{-1}\Sigma_{m,xy}$, with mixture probabilities equal to the component membership probability

$$\Big(\pi_m\, N(x^*|\mu_m, \Sigma_{m,xx})\Big) / \Big(\sum_{m'=1}^{n} \pi_{m'}\, N(x^*|\mu_{m'}, \Sigma_{m',xx})\Big).$$

In these equations, $\Sigma_{m,xx}$ is the x marginal covariance for component m, $\Sigma_{m,xy}$ is the x-y covariance, and $\sigma^2_{m,y}$ is the marginal variance of y. From this conditional density for y^*, we select class w^* to minimize absolute error loss by selecting the most likely class.

3.2.2 Model prior specification

We define appropriate priors for model parameters π, θ, μ, ψ, and Σ in a similar way as done for the mixture prior model for prediction. A difference in this model is the required prior for the latent response variable y. The resulting posterior distribution for parameters π, μ, ψ, and Σ are used for classification. As an overview, our model's prior distribution has the following structure:

1. There is a Gaussian prior on the latent y parameters with a user-provided mean τ_i for each y_i and common variance ν^2. The τ values represent the user's guesses for the latent response using the observed $-1, +1$ class data.

2. The prior for Gaussian parameters θ is a discrete distribution at values η, representing the mixture component parameters, with probability mass π.

3. The component proportions π of the Dirichlet process prior use the stick-breaking construction of Sethuraman (1994).

4. The prior for each η_m is the product of mixture priors for mean μ_m and the covariance Σ_m, multiplied by a standard normal prior on ψ_m:

   (a) The prior for each mean ψ_m is standard normal.

   (b) The prior for each mean parameter μ_m is a mixture of multivariate Gaussian density kernels centered at the data locations (the x marginal) with a common covariance (equal to the empirical bandwidth for the x marginal) and equal kernel selection probability.

(c) The prior for each covariance matrix Σ_m is a mixture of inverse-Wishart density kernels with ρ degrees of freedom and user-provided scale matrices for the x marginal covariance portion, unity y marginal variance, and x-y covariance from a collection of correlation guesses. The scale matrices are derived from the data and represent guesses for the x marginal covariance and simple x-y correlations. The respective selection probabilities for the prior scale matrices are user-provided.

3.2.2.1 Prior for y_i

There is a Gaussian prior for each y_i centered at user-provided mean values τ and common variance ν^2, proportional to $\exp\big(-\frac{1}{2}(y_i - \tau_i)^2\nu^{-2}\big)$. As there is no information about the y, besides whether or not it is greater than zero, we let y be independent of x in the prior. The values τ are user-provided guesses for the values of the latent y.

One way to generate τ is using a smoother on the observed $-1, +1$ class values of w. Another way to compose τ is from normal quantiles for a function of two kernel-based priors, one for $w = -1$ and another for $w = 1$, through a mixture of Gaussian density kernels of weight $1/N_{obs}$ centered at each x_i.

3.2.2.2 Prior for θ_i

The defined priors for θ_i and π_m are similar to those priors in the prediction method. The prior for θ_i, the multivariate Gaussian parameters for observation i, is $\sum_{m=1}^{n} \pi_m \delta_{\eta_m}$, where δ_{η_m} is a unit point mass at η_m, i.e., $P(\theta_i = \eta_m) = \pi_m$.

3.2.2.3 Prior for π_m

A truncated stick-breaking prior is defined for the n mixture component proportion sizes π via variables $\phi_1, \ldots, \phi_{n-1}$ that follow a conjugate beta distribution. Declare π_m as functions of ϕ created by the stick-breaking representation of the Dirichlet process through the following equations

$$\pi_m(\phi) \equiv \phi_m \prod_{k=1}^{m-1}(1-\phi_k),$$

with the first $n-1$ $\phi_m \sim$ iid Beta(α, β), and $\phi_n \equiv 1$. Note that any function conditional on π is equivalently conditional on ϕ. The beta distribution hyperparameters α and β can be determined from the data to conform to some optimal mixture proportions from a clustering algorithm.

3.2.2.4 Prior for η_m

The prior distribution for Gaussian distribution parameters η, denoted $g(\eta)$, is similar to the extension of the conditionally conjugate normal–inverse-Wishart prior for (μ, Σ) of Müller et al. (1996) as defined in the prediction method.

The prior on μ, the Gaussian mean parameter for the x, incorporates Gaussian density kernels centered at the observations x_i, but in this case with equal kernel selection probability. To make the priors for each μ_m and Σ_m independent, these Gaussian density kernels will not use Σ_m for the covariance matrices, but instead an empirical multivariate bandwidth

$$\Gamma = N_{obs}^{-(p+6)/(p+4)} \sum_{i=1}^{N_{obs}} (x_i - \bar{x})(x_i - \bar{x})^T$$

derived from Simonoff (1996). The prior for ψ_m, for the mean of y for component m, is simply standard normal.

The inverse-Wishart densities comprising the kernels of the mixture come from x marginal covariance matrices found by a clustering algorithm performing an exhaustive search of the underlying covariance structure of the mixtures in the x marginal data. The user may wish to combine each of these x marginal covariance matrices with a collection of guesses for the x-y correlations; the product set would comprise all guesses as to the Σ structure of both the predictors and latent response within the population mixture. From such an algorithm we generate L covariance matrix guesses M_l, each with kernel selection weight w_l. The resulting prior for each Σ_m is a mixture of inverse-Wishart distributions, each with ρ degrees of freedom, with scale matrices M_l and density kernel probability mass w_l. We generate a large enough collection of guesses M_l to try to have at least one density kernel that is a close match to each true subpopulation covariance matrix Σ_h.

With μ_m, ψ_m, and Σ_m independent, the prior for each η_m becomes

$$g(\eta_m) = g_1(\mu_m) g_2(\psi_m) g_3(\Sigma_m),$$

with the mean and covariance mixture priors as

$$\begin{aligned} g_1(\mu_m) &\propto |\Gamma|^{-1/2} \sum_{i=1}^{N_{obs}} \exp\left(-\tfrac{1}{2}(\mu_m - x_i)^T \Gamma^{-1} (\mu_m - x_i)\right) \\ g_2(\psi_m) &\propto \prod_{m=1}^{n} e^{-\frac{1}{2}\psi_m^2} \\ g_3(\Sigma_m) &\propto |\Sigma_m|^{-(\rho+p+2)/2} \sum_{l=1}^{L} w_l \exp\left(-\tfrac{1}{2} \operatorname{tr}\left(M_l \Sigma_m^{-1}\right)\right), \end{aligned}$$

with $g_1(\mu_m)$ having N_{obs} kernel densities with equal kernel probabilities and $g_3(\Sigma_m)$ having L kernel densities with kernel probabilities defined by weight vector w. Conditional on the prior for ψ_m, the prior of $g(\eta_m)$ is a product mixture in $(\mu \times \Sigma)$-space composed of $N_{obs}L$ normal–inverse-Wishart distributions.

3.2.3 Posterior distribution

From the classification probability model, the conditional posterior distribution for the parameters, $p(\theta, \eta, \phi, y | x, w)$, is proportional to

$$\begin{aligned} p(\theta, \eta, \phi, y | x, w) \propto \prod_{i=1}^{N_{obs}} & \left(N(x_i, y_i | \theta_i) N(y_i | \tau_i, \nu^2)\, \mathbb{I}\left[w_i y_i \geq 0\right] \prod_{m=1}^{n} \pi_m(\phi)^{\,\mathbb{I}[\eta_m = \theta_i]} \right) \\ & \times \left(\prod_{m=1}^{n} g(\eta_m) \right) \left(\prod_{m=1}^{n-1} \operatorname{Beta}(\phi_m | \alpha, \beta) \right) \prod_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i \in \{\eta_1, \ldots, \eta_n\}\right]. \end{aligned} \tag{3.1}$$

Conditional distributions can be found from (3.1) for the individual θ_i, ϕ_m, μ_m, ψ_m, Σ_m, and y_i, allowing a Gibbs sampler of the posterior. The exact posterior distribution in terms of $\theta_1, \ldots, \theta_{N_{obs}}$, $\phi_1, \ldots, \phi_{n-1}$, $\mu_1, \ldots, \mu_n$, $\psi_1, \ldots, \psi_m$, $\Sigma_1, \ldots, \Sigma_n$ and $y_1, \ldots, y_{N_{obs}}$, hyperparameters α and ρ, and user-provided covariance matrices $M_1, \ldots, M_L$ and weights $w_1, \ldots, w_L$, is proportional to

$$\prod_{m=1}^{n} \Bigg(|\Sigma_m^{-1}|^{\frac{n_m}{2}} e^{-\frac{1}{2}S_{1,m} - S_{2,m} - \frac{1}{2}S_{3,m}} e^{-\frac{1}{2}\psi_m^2} \sum_{i=1}^{N_{obs}} e^{-\frac{1}{2}(x_i-\mu_m)^T\Gamma^{-1}(x_i-\mu_m)}$$
$$\times \sum_{l=1}^{L} w_l \left|M_l^{-1}\right|^{-\frac{\rho}{2}} \left|\Sigma_m^{-1}\right|^{\frac{\rho-p-2}{2}} e^{-\frac{1}{2}\operatorname{tr}\left(\Sigma_m^{-1}M_l\right)} \Bigg) \left(\prod_{i=1}^{N_{obs}} e^{-\frac{1}{2}(y_i-\tau_i)^2\nu^{-2}} \, \mathbb{I}\left[w_i y_i \geq 0\right] \right)$$
$$\times \left(\prod_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i \in \{\eta_1, \ldots, \eta_n\}\right] \right) \left(\prod_{m=1}^{n-1} \phi_m^{n_m}(1-\phi_m)^{n_m^+ + \alpha - 1} \right), \qquad (3.2)$$

where

$$n_m = \sum_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i = \eta_m\right] \qquad n_m^+ = \sum_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i \in \{\eta_{m+1}, \ldots, \eta_n\}\right]$$
$$S_{1,m} = \sum_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i = \eta_m\right](y_i - \psi_m)^2 V_m^{(y)}$$
$$S_{2,m} = \sum_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i = \eta_m\right](y_i - \psi_m) V_m^{(yx)}(x_i - \mu_m)$$
$$S_{3,m} = \sum_{i=1}^{N_{obs}} \mathbb{I}\left[\theta_i = \eta_m\right](x_i - \mu_m)' V_m^{(x)}(x_i - \mu_m),$$

with V_m being the inverse of the component covariance matrix, Σ_m^{-1}:

$$\Sigma_m^{-1} \equiv V_m = \begin{bmatrix} V_m^{(y)} & V_m^{(yx)} \\ V_m^{(xy)} & V_m^{(x)} \end{bmatrix} \begin{array}{l} 1 \times (p{+}1) \\ p \times (p{+}1) \end{array}$$

The conditional distributions are derived from (3.2) by analyzing the distribution of each θ_i, ϕ_m, μ_m, ψ_m, Σ_m, and y_i separately when all other variables are held constant. A Gibbs sampler designed with these conditionals draws samples from the posterior distribution for $p(\theta, \pi, \eta, y|x, w)$.

3.2.3.1 Conditional posteriors for θ_i, ϕ_m, Σ_m

These parameters have the same conditional distributions as in the prediction model. The N_{obs} conditionally independent θ parameters have a conditional distribution in (3.2) that is a multinomial distribution among $\eta_1, \ldots, \eta_n$ according to the probabilities

$$(\pi_m \, N(x_i, y_i|\eta_m)) \, / \, (\textstyle\sum_{m'=1}^{n} \pi_{m'} \, N(x_i, y_i|\eta_{m'}))\,.$$

After the N_{obs} θ_i reassignments are made, the ϕ variables are redrawn. Each $\phi_m : m = 1, \ldots, n-1$, has the conditional posterior is proportional to $\phi_m^{n_m+\alpha}(1-\phi_m)^{n_m^+ + \beta}$, where $n_m^+ = \sum_{k=m+1}^{n} n_k$. Therefore each ϕ_m follows a Beta$(n_m+\alpha, n_m^+ + \beta)$ distribution, noting that ϕ_n always equals one.

Each Σ_m in (3.2), independent of the other component covariances, has the following mixture distribution, composed of L inverse-Wishart kernel densities each with $n_m + \rho$ degrees of freedom and scale matrix $(M_l + S_m)^{-1}$. To find the kernel density the Gibbs sampler selects for this discrete distribution, we find the marginal distribution of each kernel in the density mixture by integrating out the Σ_m in each term. After integrating out Σ_m, each term has the form

$$w_l \, |M_l|^{\frac{\rho}{2}} \, / \, |M_l + S_m|^{\frac{n_m+\rho}{2}}\,, \tag{3.3}$$

and this is the proportional kernel selection probability, $l = 1, \ldots, L$. Kernel l is selected in the Gibbs sampler according to a draw from a L component multinomial distribution with probabilities according to (3.3).

3.2.3.2 Conditional posterior for ψ_m

The conditional posterior for ψ_m, the mixture component response mean, is normal with variance $(n_m V_m^{(y)} + 1)^{-1}$ and mean $(n_m V_m^{(y)} + 1)^{-1}(n_m \bar{y}_m V_m^{(y)} + n_m V_m^{(yx)}(\bar{x}_m - \mu_m))$, where $\bar{x}_m$ and $\bar{y}_m$ represent the means for observations assigned to component m. The conditional is proportional to

$$\exp\left\{-\tfrac{1}{2}\psi_m^2(n_m V_m^{(y)} + 1) + \psi_m \left(n_m \bar{y}_m V_m^{(y)} + n_m V_m^{(yx)}(\bar{x}_m - \mu_m)\right)\right\}.$$

3.2.3.3 Conditional posterior for μ_m

The conditional posterior distribution for μ_m, all independent of each other, is a mixture of multivariate normals with density proportional to

$$\sum_{i=1}^{N_{obs}} e^{-\frac{1}{2}\mu_m'(\Gamma^{-1}+n_m V_m^{(x)})\mu_m+(n_m(\bar{y}_m-\psi_m)V_m^{(yx)}+n_m\bar{x}_m' V_m^{(x)}+x_i'\Gamma^{-1})\mu_m-\frac{1}{2}x_i'\Gamma^{-1}x_i}. \tag{3.4}$$

For each x_i multivariate normal density kernel in this mixture, the variance is $(\Gamma^{-1} + n_m V_m^{(x)})^{-1}$ and mean is the variance times vector $b(x_i)$, where $b(x_i) = n_m(\bar{y}_m - \psi_m)V_m^{(xy)} + n_m V_m^{(x)}\bar{x}_m + \Gamma^{-1}x_i$. As the distribution for μ_m is a mixture of normal densities, the Gibbs sampler must first select a kernel from the mixture distribution. The kernel is selected with probability proportional to the marginal density of (3.4) with respect to the x_i, with kernel marginal selection density attained by integrating out the μ_m over each x_i kernel. The draw of μ_m is from a multivariate normal kernel density whose kernel is chosen from an N_{obs} component multinomial distribution. Details of the marginal density derivation are in the Appendix.

3.2.3.4 Conditional posterior for y_i

For each y_i, independent of one another, the conditional posterior distribution is proportional to

$$\exp\left\{-\tfrac{1}{2}y_i^2(V_0^{(y)} + \nu^{-2}) + y_i\Big(\psi_0 V_0^{(y)} + \tau_i\,\nu^{-2} - V_0^{(yx)}(x_i - \mu_0)\Big)\right\} \mathbb{I}\left[w_i y_i \geq 0\right]$$

yielding a truncated normal distribution with variance $(V_0^{(y)} + \nu^{-2})^{-1}$ and mean equal to $(V_0^{(y)} + \nu^{-2})^{-1}(\psi_0 V_0^{(y)} + \tau_i\,\nu^{-2} - V_0^{(yx)}(x_i - \mu_0))$. This posterior density for y_i is truncated at zero with positive density on the same side as w_i.

3.2.3.5 Posterior sampling

After the latent response y, stick-breaking parameters ϕ, and the Gaussian component parameters η are initialized by drawing values from their prior distributions, the Gibbs sampler operates with the following algorithm.

1. Update each θ_i, one at a time, as a random draw from the discrete distribution $\{\eta_1, \dots, \eta_n\}$ with probabilities proportional to $\pi_m N(x_i, y_i|\eta_m)$.

2. Update each ϕ_m for $m = 1, \dots, n-1$, one at a time, from its conditional posterior distribution, Beta$(n_m + \alpha, n_m^+ + \beta)$. After the ϕ update, the component proportions π are recalculated.

3. Update each η_m, one at a time:

   (a) for μ_m, draw from a multivariate normal distribution, according to its conditional parameters as shown in the Appendix,

   (b) for ψ_m, draw from a univariate normal distribution, according to its conditional parameters as shown above,

   (c) for Σ_m, draw from an inverse-Wishart distribution with $n_m + \rho$ degrees of freedom, according to its conditional parameters as shown above.

4. Update each y_i, one at a time, as a random draw from a normal density with conditional parameters as shown above.

3.2.4 Classification

The error metric we consider for classification is misclassification error. We minimize misclassification error by selecting the class -1 or $+1$ for w depending on whether the conditional of y given x has more density below or above zero, respectively, according to the joint probability model. Therefore a classification rule that minimizes the misclassification error is found by estimating the amount of the conditional density of the latent y on the predictors x that is greater than zero (Izenman, 2009).

The Gibbs sampler draws a collection of J samples from the posterior distribution assumed to be randomly generated values from the posterior. The sampler saves J iterations, and for each of these samples the values for parameters π_m, μ_m, ψ_m, and Σ_m

are kept. After an appropriate burn-in period of B iterations and thinning by saving only one of every t iterations, a total of approximately $B + t\,J$ iterations are required for the Gibbs sampler; see Gamerman and Lopes (2006) for convergence checking guidelines. These J saved sets of parameter values are used to classify the response w^* for a new observation x^*.

To get a predicted class at x^*, follow this algorithm for each iteration $j = 1, \dots, J$ saved by the sampler:

1. For each mixture component m, compute $\ell_m^{(j)}(x^*)$, the likelihood that x^* belongs to component m given the mixture arrangement at sample j,

   $$\ell_m^{(j)}(x^*) = \pi_m^{(j)}\, N(x^* | \mu_m^{(j)}, \Sigma_{m,xx}^{(j)}),$$

   where the Gaussian density is based only on the marginal of x (Bernardo et al., 2011).

2. For each component m, compute $\hat{p}_m^{(j)}(x^*)$, the sample estimated probability that the component m conditional Gaussian density of y^* at x^* is greater than zero, when the density has variance $(V_m^{(y)} + \nu^{-2})^{-1}$ and mean $(V_m^{(y)} + \nu^{-2})^{-1}(\psi_m\, V_m^{(y)} + \tau_i\, \nu^{-2} - V_m^{(yx)}(x_i - \mu_0))$.

3. Form $\hat{p}^{(j)}(x^*)$, the estimated probability that y^* at x^* is greater than zero for sample j, by computing the weighted average of the conditional probabilities

   $$\hat{p}^{(j)}(x^*) = \frac{\sum_{m=1}^{n} \ell_m^{(j)}(x^*)\, \hat{p}_m^{(j)}(x^*)}{\sum_{m=1}^{n} \ell_m^{(j)}(x^*)}.$$

From these J sample predictions, calculate $\hat{p}(x^*)$, the estimated probability that the conditional density of y^* at x^* is greater than zero, $\hat{p}(x^*) = \frac{1}{J}\sum_{j=1}^{J} \hat{p}^{(j)}(x^*)$. The classification rule is to predict class $+1$ for an observation at x^* if the estimated probability $\hat{p}(x^*)$ is greater than one-half; otherwise -1 is chosen as the predicted class at x^*.

Implementation of this method involves some tuning of the parameters of the model, but the classifier follows a similar implementation as in the regression paper regarding parameters α, β, ρ, and n. An exception is that this method requires tuning the additional parameter ν, the standard deviation of the Gaussian prior on the latent response y.

3.3 Demonstration of Our Method

First, a two-dimensional test case based on the doppler function shows functionality of this classification method. Second, we present a comprehensive simulation study with six and twelve predictors. Third, this method classifies the Banknote Authentication data set, which is a standard machine learning test set for classification techniques. In these tests we compare performance with support vector machines (SVM), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), k-nearest neighbors, and random forests.

In the following simulations, we use R library Mclust in the same way to get the mixture prior M_l matrices and weights w_l as in the regression paper simulations. A large number of groupings is preferred so there is enough variety of M_l matrices to ensure a close match to any existing covariance structure. The scale matrix weights are inversely proportional to the number of clusters. In these simulations, the classification Gibbs sampler is coded in C with sample sizes between 500 and 1000. Burn-in periods and thinning rates are determined to balance classification accuracy with computation demands. The first random seed's Gibbs sampler output generates this method's resulting predicted classes. The comparison methods run in R with standard implementations, using leave-one-out or k-fold cross-validation to tune model parameters where necessary, depending on computational complexity.

3.3.1 Predicting the doppler function

The doppler function is a simple bivariate test case that allows for visualization of how the method works, while adequately demonstrating that the mixture prior classifier extends Bayesian mixture models to allow flexible classification. This function is chosen not so much as a potential classification use for the method, but to show classification in a situation with a mixture population with complex covariance structure.

To demonstrate that this method is able to pick out subpopulations with very different covariance structures, we first consider a simulation of the doppler function shown in Wasserman (2004), $d(x) = \sqrt{x(1-x)}\sin(2.1\pi/(x+.05))$, for $x \in [0,1]$. The data consist of $N_{obs} = 2048$ points from $y|x \sim d(x) + N(0, .1^2)$ with $x = i/N_{obs}$, except we observe (x, w), where w indicates if y is greater than zero. The task is to estimate the conditional class probability function to minimize the misclassification error. Performance is evaluated at the 5000 points $i/5000$ for $i = 1, \ldots, 5000$.

We use this method to predict the conditional probability that y is greater than zero based on $n = 36$ latent mixture components, the number of η components. There are 900 input scale matrices that form the prior mixture density for the covariance parameters. These scale matrices come from the R clustering algorithm Mclust on the x portion of the data, first fitting 1 cluster, then 2 clusters, etc. until 24 clusters, yielding 300 covariance matrices. Then each scale matrix is combined with three potential x-y correlations: $-0.5, 0, 0.5$. Results are based on 800 samples, thinning rate of 10 iterations per sample, with a burn-in period of 2000 iterations, for a total of approximately 10000 iterations.

A plot of this function with the training data is shown in Figure 3.3.1. The plot shows the estimated probability that the conditional response is greater than zero at the 5000 test points, rescaled from -1 to 1 to easily show the classification decision boundary that becomes the line $y = 0$. Misclassification rates for the mixture prior classifier is .070, which is inferior to other classification techniques—for example, k-nearest neighbors achieves .062 with optimal $k = 3$. However, note that the predicted class function is smooth, similar to the performance of the mixture prior Dirichlet process method for regression.

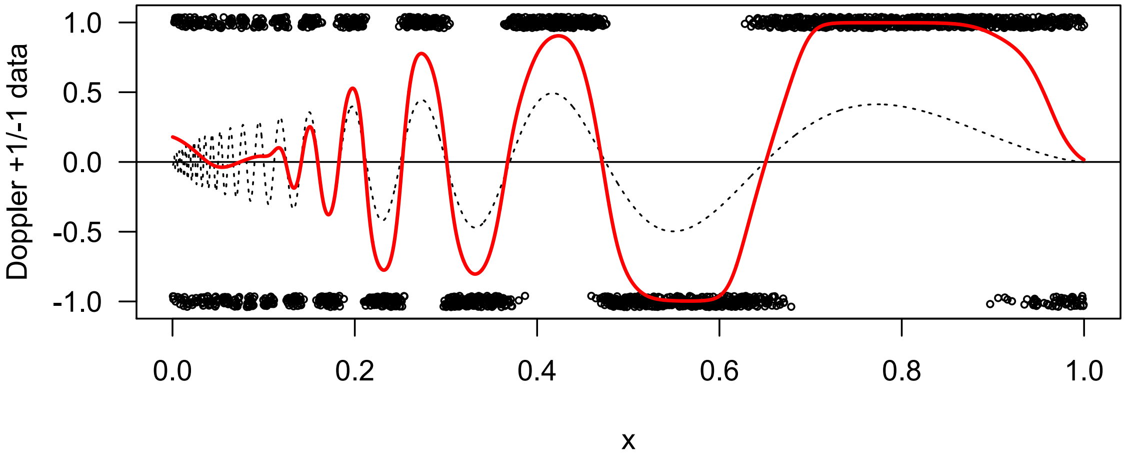

Figure 3.1 Classification of Data From the Doppler Function. A training set of size 2048 points from doppler function, with normal noise variance $(0.1)^2$. The dashed line is the doppler function, and the red line is the estimated probability that the latent response y is greater than zero based on the posterior distribution. Note that this probability is rescaled from $(0, 1)$ to $(-1, 1)$. Misclassification occurs when the red line does not cross the $y = 0$ line where the doppler function does.

3.3.2 Comprehensive simulation study

A comprehensive test case of higher-dimensional simulated data is presented to show how the method performs in a variety of mixture distribution situations. This classification simulation study is modeled after the comprehensive nonparametric regression testing in Banks et al. (2003). A class probability function is created from various scenarios, simulated data are generated, then this mixture prior method and five comparison methods classify the data with the aim to minimize classification error. Each scenario is randomly regenerated a total of 25 times.

3.3.2.1 Generation of simulated test cases

The study focuses on class prediction for data generated from a continuous class probability function, conditional on x, that is a mixture of eight multivariate normal densities

$$E(Y|X = x) = \sum_{h=1}^{8} p_h(x) \left(\psi_h + \Sigma_{h,yx}\Sigma_{h,xx}^{-1}(x - \mu_h)\right)$$

where $p_h(x)$ is the probability that an observation at x is from density h when considering both the mixture Gaussian parameters and proportion sizes. The objective is to minimize misclassification error by predicting whether or not this conditional class function is greater that zero.

Simulations use multivariate normal component parameters generated from R package MixSim (Melnykov et al., 2012). The amount of overlap can be specified when generating data with MixSim, and we choose an average overlap $\bar{\omega}$ of either .02 or .05 for the simulations, which specifies moderate or low mixture component separation, respectively (Melnykov and Maitra, 2010). We also choose non-spherical densities and draw the first seven mixture proportions from Stick($\alpha = 16, \beta = 64$), with the eighth mixture assigned the remaining probability. After the Gaussian component parameters and mixture proportions are determined, data are generated with MixSim function simdataset using default choices for all other parameters. Our manual selection of stick-breaking proportion sizes slightly increases the average overlap $\bar{\omega}$ in the test cases from the target values of .02 and .05.

3.3.2.2 Simulated test case scenarios

Testing is performed under two dimensions ($p = 6, 12$ covariates), three data set sizes $N_{obs} = k(1.2)^p$ ($k = 200, 500, 1250$), two degrees of population mixture separation ($\bar{\omega} = .02, .05$), and these three data generation scenarios:

MVN Data. Data are generated from an eight-component multivariate Gaussian mixture. This scenario is an ideal situation for the mixture prior Bayes classifier. We expect the mixture prior Bayes classifier to perform competitively with other classification techniques for this scenario.

Spurious. Data are generated from an eight-component multivariate Gaussian mixture as in the first scenario, except now one-third of the predictors are spurious with respect to, i.e. are independent of, all other variables. This scenario tests how the methods are able to handle noisy predictors in the absence of variable selection.

Uniform X. The class probability function is generated from the eight-component multivariate normal mixture, but then the predictors are regenerated from independent Uniform(0,1). The class response w is based on y that is equal to the class conditional probability function at $X = x$ plus $\mathrm{N}(0, 0.1^2)$ noise. This scenario tests how the method performs when the multivariate normal structure in the predictors is removed and the only relationship available is the localized class data.

3.3.2.3 Local classification methods for comparison

The method's classification performance, as measured in misclassification error relative to the true class probability function given the predictors, is compared with the following five methods.

SVM. The first comparison method is support vector classification implemented via the R package e1071 using the function svm with a Gaussian radial basis function kernel. This function requires tuning of the two kernel parameters cost and gamma, and do this we use a three-layer grid search using 10-fold cross validation. Tuning is performed with function tune.svm, and effectively considers cost parameters in the range 10^{-2} to 10^{3} and gamma values in the range 10^{-4} to 10. Graphical checking of the search results confirm that this tuning process yields sensible values for the classifier. The average tuning parameters for $p = 6$ simulations are 140 and 0.2 for cost and gamma, respectively.

K-Nearest Neighbors. The second comparison method is a k-nearest neighbor classifier implemented in R using function knn from library class. The optimal k neighbors is determined by two stages of tuning—first with 10-fold cross-validation, second with 50-fold cross-validation—via the function's calculation for leave-one-out cross-validation. The range of average optimal number of neighbors across the scenarios is 18 to 50, with lower values preferred by smaller data sets and the uniform scenario and higher values preferred by higher overlapping mixtures and spurious conditions. The optimal k appears to greatly increase with the number of predictors.

LDA and QDA. The third and fourth comparison methods are classification rules using linear and quadratic discriminant analysis implemented in R using library MASS functions lda and qda. These functions run very quickly and seem to adequately order the resulting class probabilities for the observations. However, these methods do not decide the class cutoff accurately, leading to higher misclassification rates while maintaining competitive receiver operating characteristics.

Random Forests. The fifth comparison method is a random forest classifier implemented in R using library randomForest. Tuning for both terminal node size and number of trees is performed via a grid search with function tune.randomForest. On average, the range of mean parameter values for the scenarios are four to seven for terminal node size and 600 to 900 for number of trees. As overfitting is not a significant concern for random forests, we run the method longer than may be necessary to ensure enough trees have been grown to minimize prediction bias.

3.3.2.4 Simulated test case results

The comprehensive test case results are summarized by scenario in three plots, Figures 3.2, 3.3, and 3.4. Figures 3.5 through 3.8 display results for different data set sizes for the multivariate normal scenarios, with and without spurious predictors, across mixture separation levels and dimensionality. Figure 3.9 displays computation time for these methods. To obtain these results, we run the mixture prior Bayes algorithm for 400 samples using a burn-in period of 1000 iterations and thinning rate of 10 iterations per sample. The algorithm uses 36 mixture components in the Dirichlet process, sets y-prior standard deviation parameter ν to 2.5, and determines values for the y_i-prior mean parameters τ_i as a ratio of optimized class densities using kernel density estimation. From the simulation study results shown in these figures, we determine the following findings about mixture prior Bayes classifier performance.

Competitive Simulation Results for Mixture Prior Classifier. From the simulation study we find the mixture prior Dirichlet process classifier to be competitive with the comparison classifiers. The best competing method is SVM with a radial basis function kernel, as this is an effective method to classify data with a highly local structure. The next best competing classifiers are k-nearest neighbors and random forests, which are both local classification techniques but without the local linearity inherent in SVM. QDA performs well in some contexts but overall performs worse than the other methods. The poorest competing classifier is LDA, which lacks adaptive covariance structures to adequately fit these data.

Classifier Performance Comparisons. Performance of the mixture prior improves versus the comparison classifiers in general as the size of the data set or dimension increases. Misclassification error ratios for the mixture prior show an advantage for the mixture prior classifier when the data set sizes increase, similar to performance of the mixture prior regression method. The mixture prior classifier also performs better than the other methods, except for QDA, when there are more predictors. However, performance degrades on average as the amount of mixture overlap increases.

The mixture prior Bayes classifier works best in the multivariate data generation scenario. The good performance is due to the fact the probability model underlying this model is also a multivariate normal mixture. The method performs less well when spurious predictors are introduced. There are some runs with spurious predictors that

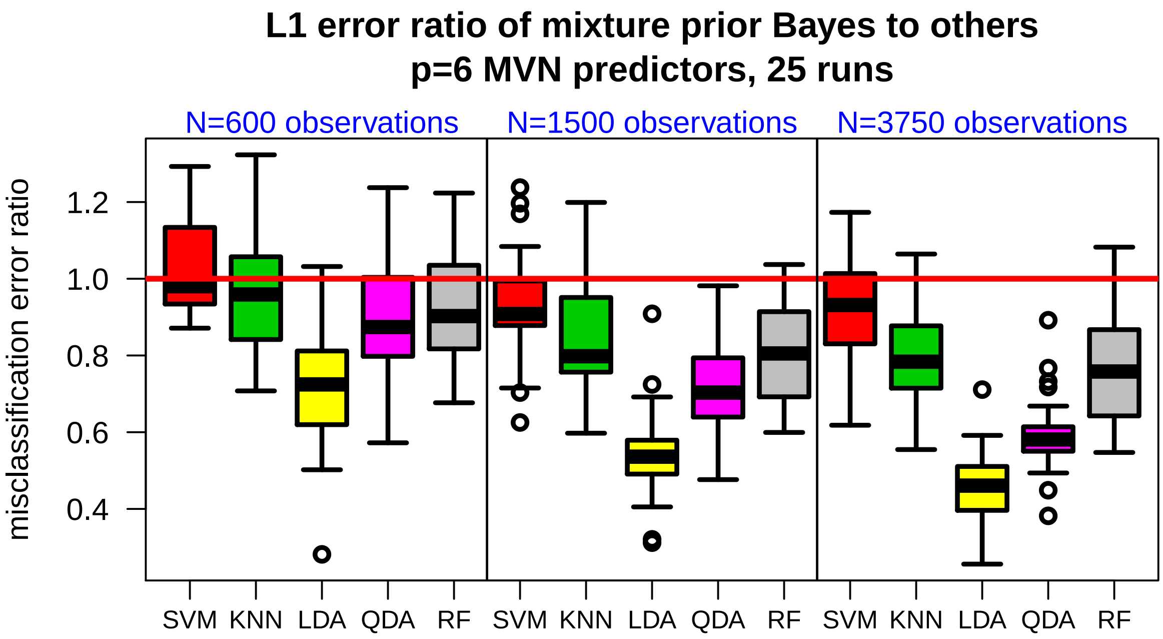

Figure 3.2 Simulation Study Error Ratios for $p = 6$ and Multivariate Normal Scenario. This plot displays mixture prior Bayes misclassification error to that of the comparison classification techniques, with results below one being favorable to the mixture prior Bayes classifier. Results shown are for the multivariate normal data generation scenario for six predictors. These simulations use low mixture separation for the eight-component population mixture distribution. This large overlap of population mixtures makes classification of these data more difficult. The three sections denote the small, medium, and large data set sizes.

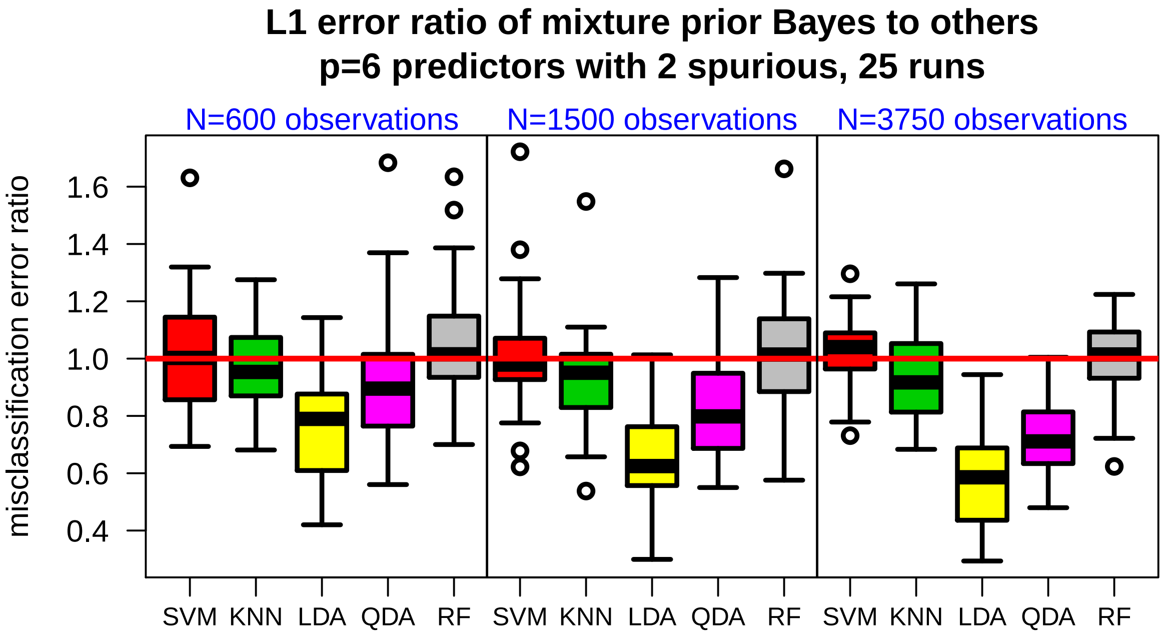

Figure 3.3 Simulation Study Error Ratios for $p = 6$ and Spurious Predictor Scenario. This plot displays mixture prior Bayes misclassification error to that of the comparison classification techniques, with results below one being favorable to the mixture prior Bayes classifier. Results shown are for the multivariate normal data generation with one-third of $p = 6$ predictors are spurious. These simulations use low mixture separation for the eight-component population mixture distribution. This large overlap of population mixtures makes classification of these data more difficult. The three sections denote the small, medium, and large data set sizes.

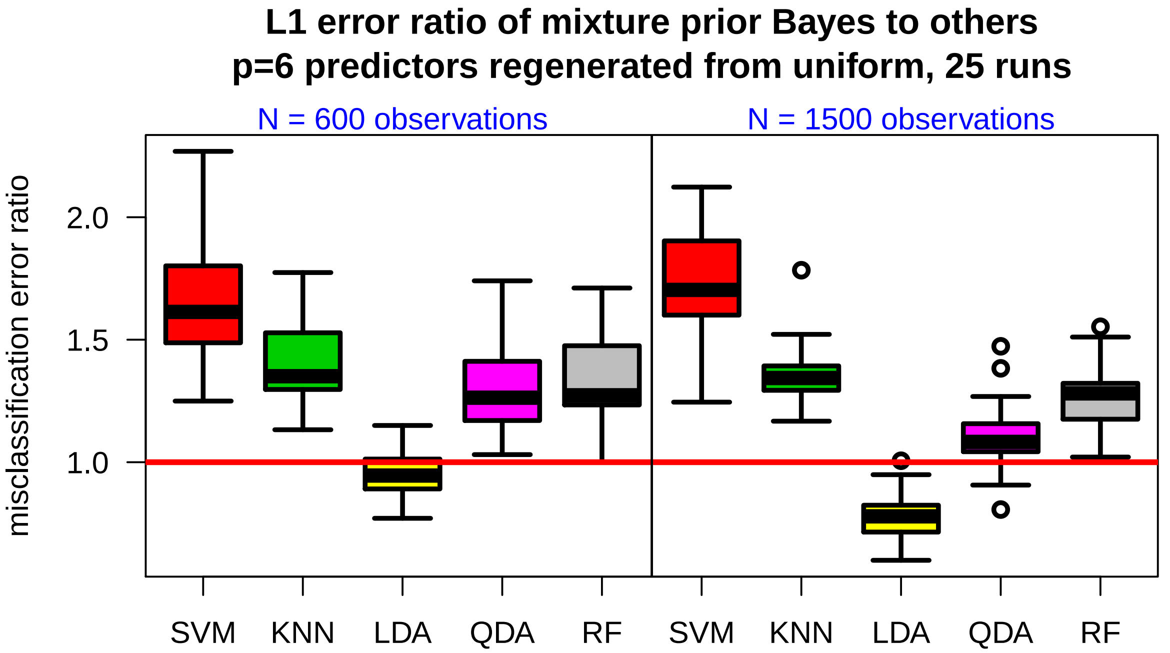

Figure 3.4 Simulation Study Error Ratios for $p = 6$ and Uniform X Scenario. This plot displays mixture prior Bayes misclassification error to that of the comparison classification techniques, with results above one (denoted by the red line) being unfavorable to the mixture prior Bayes classifier. Results shown are for $p = 6$ predictors regenerated from uniform distributions. These simulations use high mixture separation for the eight-component population mixture distribution. This scenario is inherently difficult to classify due to the regeneration of data from uniform distributions, therefore there is no need to increase the difficulty by increasing the amount of population mixture overlap. The two sections denote the small and medium set sizes.

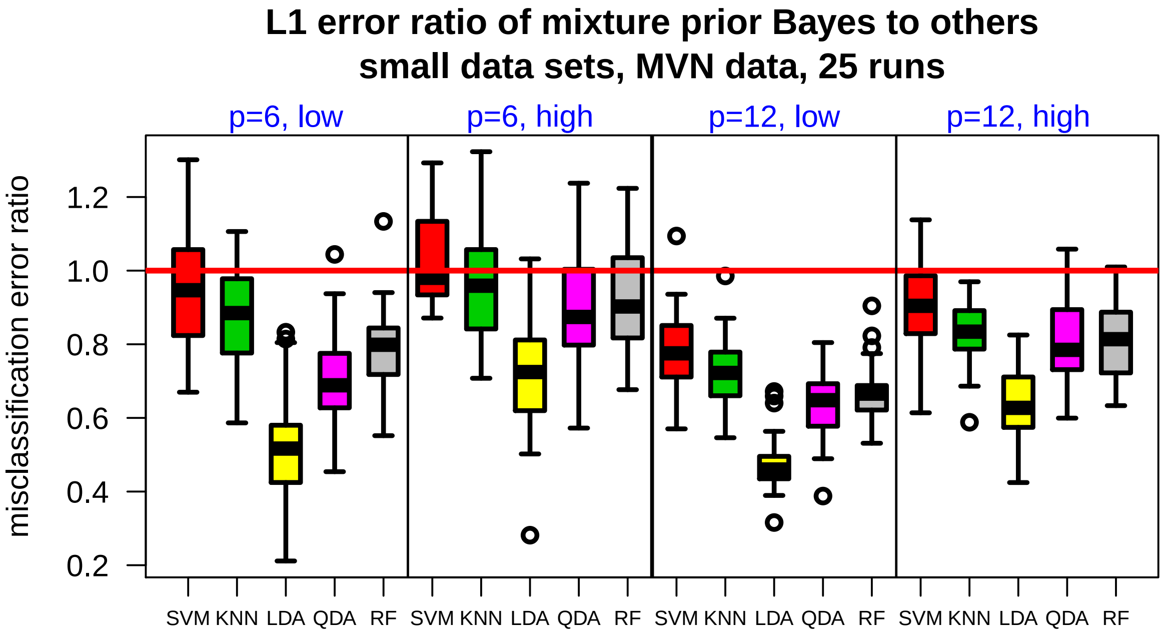

Figure 3.5 Simulation Study Error Ratios for Small-Sized Data Sets and Multivariate Normal Scenario. This plot displays mixture prior Bayes misclassification error to that of the comparison classification techniques, with results below one being favorable to the mixture prior Bayes classifier. Results shown are for small data set sizes and the multivariate normal data generation scenario. The left two panels use six predictors and 598 observations, and the right two panels use twelve predictors and 1784 observations. Within these pairs of results, the left-side panel displays low degree of mixture overlap while the right-side panel displays high degree of mixture overlap. The low degree of overlap shown on the left-side is both easier to predict and a more ideal situation for the mixture prior algorithm.

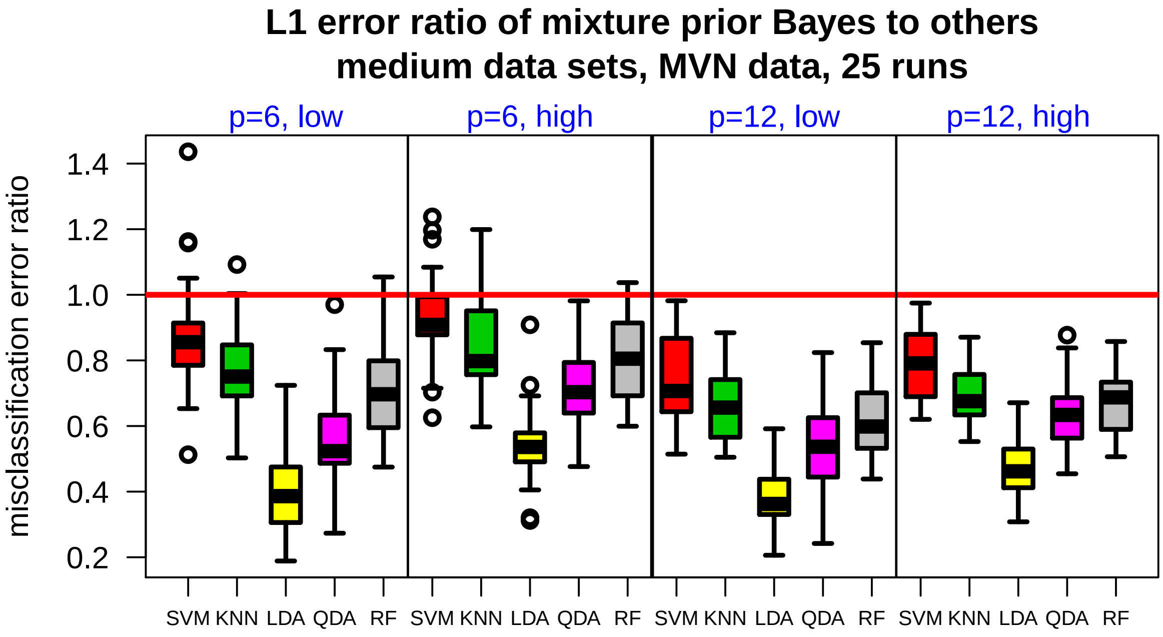

Figure 3.6 Simulation Study Error Ratios for Medium-Sized Data Sets and Multivariate Normal Scenario. This plot displays mixture prior Bayes misclassification error to that of the comparison classification techniques, with results below one being favorable to the mixture prior Bayes classifier. Results shown are for medium data set sizes and the multivariate normal data generation scenario. The left two panels use six predictors and 1493 observations, and the right two panels use twelve predictors and 4459 observations. Within these pairs of results, the left-side panel displays low degree of mixture overlap while the right-side panel displays high degree of mixture overlap. The low degree of overlap shown on the left-side is both easier to predict and a more ideal situation for the mixture prior algorithm.

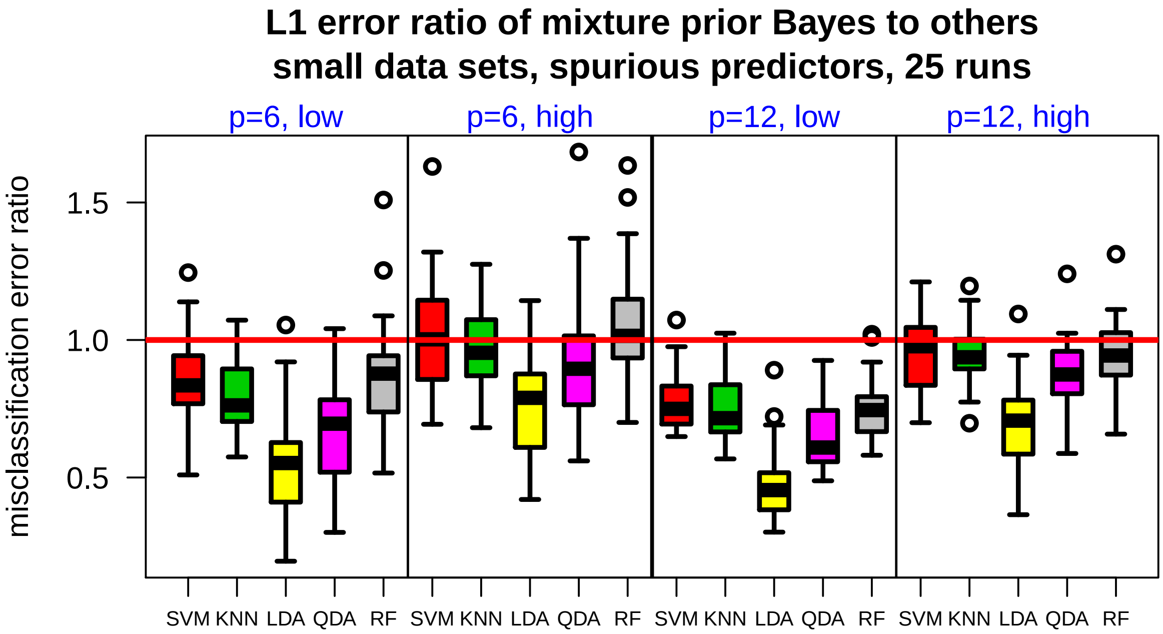

Figure 3.7 Simulation Study Error Ratios for Small-Sized Data Sets and Spurious Predictor Scenario. This plot displays mixture prior Bayes misclassification error to that of the comparison classification techniques, with results below one being favorable to the mixture prior Bayes classifier. Results shown are for small data set sizes and the spurious predictor data generation scenario, when one-third of the predictors have no relationship with the response or other predictors. The left two panels use six predictors and 598 observations, and the right two panels use twelve predictors and 1784 observations. Within these pairs of results, the left-side panel displays low degree of mixture overlap while the right-side panel displays high degree of mixture overlap. The low degree of overlap shown on the left-side is both easier to predict and a more ideal situation for the mixture prior algorithm.

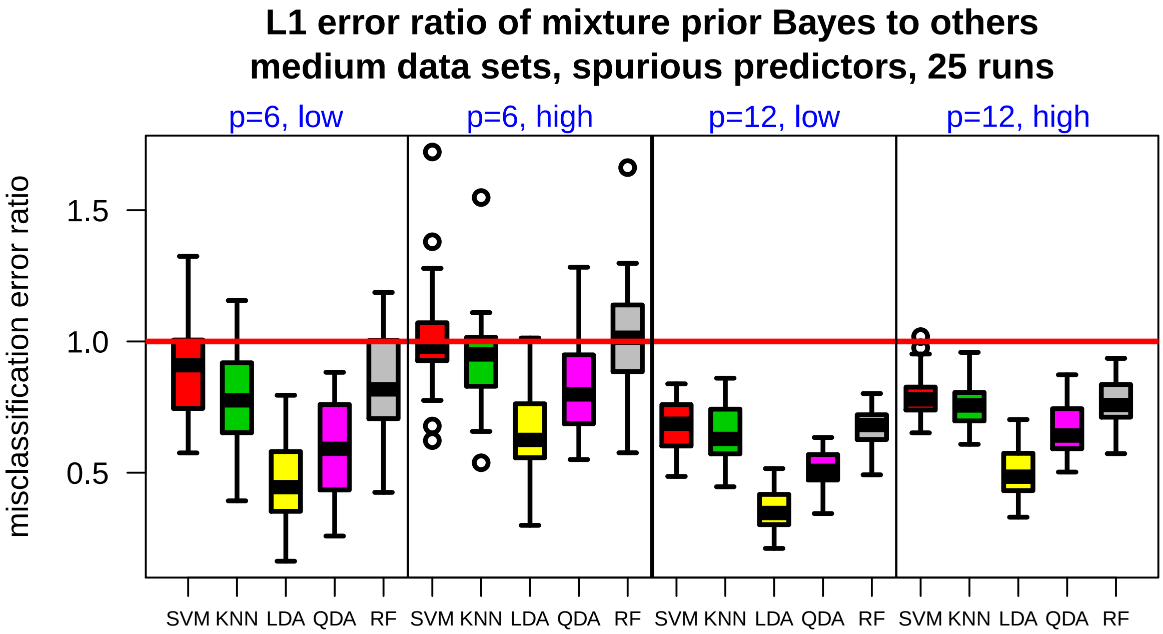

Figure 3.8 Simulation Study Error Ratios for Medium-Sized Data Sets and Spurious Predictor Scenario. This plot displays mixture prior Bayes misclassification error to that of the comparison classification techniques, with results below one being favorable to the mixture prior Bayes classifier. Results shown are for medium data set sizes and the spurious predictor data generation scenario, when one-third of the predictors have no relationship with the response or other predictors. The left two panels use six predictors and 1493 observations, and the right two panels use twelve predictors and 4459 observations. Within these pairs of results, the left-side panel displays low degree of mixture overlap while the right-side panel displays high degree of mixture overlap. The low degree of overlap shown on the left-side is both easier to predict and a more ideal situation for the mixture prior algorithm.

give poor results for the mixture prior Bayes classifier, especially for small and medium data set sizes. Performance is still good for the spurious scenario as the data are generated from a multivariate normal mixture. Classification is poor when data are regenerated from uniform distributions as there is the local class structure becomes too hard to find via the Dirichlet process method.

Potential Use in Classification Ensembles. Versus all methods outside of the uniform regeneration, there are many individual runs where the mixture prior Bayes classifier performs better than the competing classifiers, even when the overall average misclassification error for the 25 runs is the same. Perhaps this mixture prior Bayes classifier is useful in classification ensembles in that the mixture prior method considers information complementary to that of other classifiers.

3.3.2.5 Simulated test case computing times

Computing times for the 25 runs for the multivariate normal data generation scenario are shown in Figure 3.9. LDA and QDA run in a very small fraction of the time of these methods and are not shown in the plot. These results display that for these data sets k-nearest neighbors and random forests both get results quickly. The mixture prior algorithm appears to have a computational advantage over SVM as both the size of data and the dimension increases. One outlier for SVM at 150 minutes is removed from the $p = 12$ medium-sized data panel.

3.3.3 Application to machine learning data set

We now study prediction performance of the mixture prior method applied to a standard classification data set from the University California–Irvine (UCI) Machine Learning Repository (Bache and Lichman, 2013). The subject of this test data set is classification of banknote authentication. We analyze these data to demonstrate the existence of potential applications of the mixture prior Bayesian classification method.

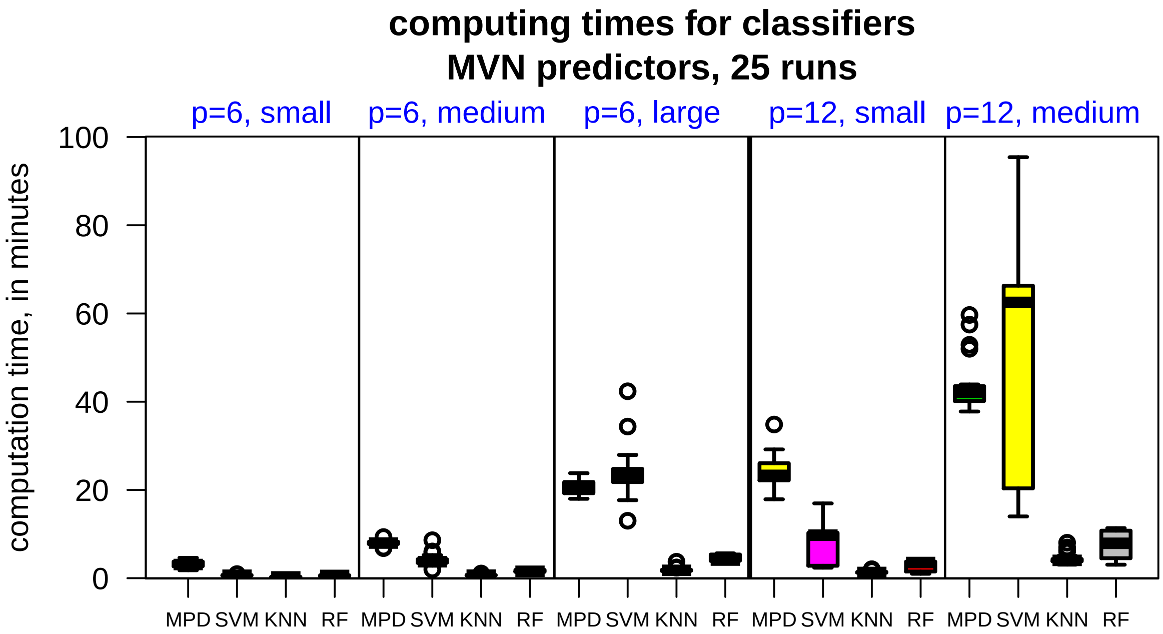

Figure 3.9 Simulation Study Computation Time. This plot displays classifier computation time, in minutes, for the 25 runs of the multivariate normal scenario. The left three panels show six predictor results, with increasing data set size, and the right two panels show twelve predictor results. One outlier for SVM at 150 minutes is removed from the $p = 12$ medium-sized data panel.

The banknote authentication data set contains four continuous predictors summarizing image analysis of 1372 banknote and banknote-like specimens that are roughly half authentic and half forgeries. The image analysis variables are gray-scale summaries by industrial cameras at a 400-by-400 pixel resolution. Three of the four predictors are the variance, skewness, and kurtosis of wavelet-transformed images of the banknotes, while the fourth predictor is the entropy of the image. There is not distinction in the data set or the literature as to which class is authentic or forgery.

We test performance of the same set of classifiers via 10-fold cross-validation. A total of 25 different foldings is considered. The mixture prior Bayes classifier is perfect in 22 of 25 runs, while k-nearest neighbor and SVM are perfect in 2 and 1 runs, respectively.

Table 3.1 Average Misclassification Error for Banknote Authentication Data

mixture prior Bayes	k-nearest neighbors	SVM	random forests	QDA	LDA
.0001	.0007	.0007	.0077	.015	.024

All times that these three classifiers are not perfect they miss on the same observation, highlighted by the triangle in Figure 3.10. The average misclassification error over the 25 runs is shown in Table 3.1.

3.4 Discussion

This paper introduces a classifier for categorical data analysis using Gaussian mixture methods. This new classification technique is based on Bayesian Gaussian mixture models using a mixture prior. Such a model offers potential improved classification of data whose class probability function exhibits a strongly localized structure. This classifier may have performance advantages in both limited data and data-rich environments and also when the dimension is large enough to rule out computationally demanding methods. We have shown how the posterior distribution has full conditionals thereby allowing computationally-efficient Gibbs sampling.

A simulation study demonstrates situations where the mixture prior Bayes method is competitive versus other classification methods. It is true that the data generation mechanism for the simulation study closely matches the underlying model of the mixture prior Bayesian classification method because the class label is determined from latent response y values. Accordingly, it is fair criticism that these data sets of the simulation study are picked to demonstrate good performance. Future research will implement an alternative simulation study data generation procedure. These alternative data will come from two separate mixture distributions, with one distribution for each class label and

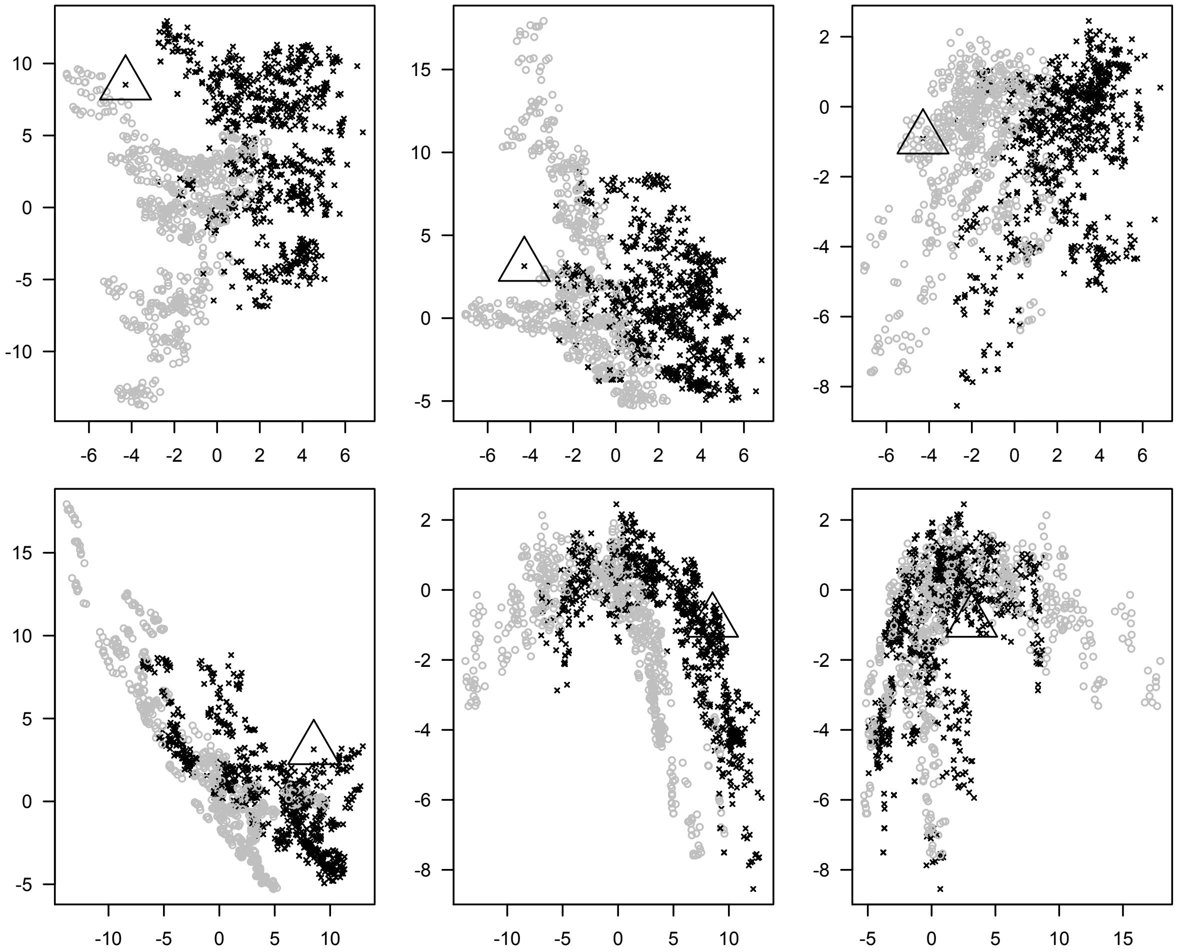

Figure 3.10 Pairs of Predictors for Banknote Authentication Data Set. This plot displays scatterplots for all combinations of the four continuous predictors with the classes shown by the grey circles and black X's. The difficult banknote to classify is highlighted by the triangle in all of the plots. The mixture prior Bayes method correctly classifies this difficult banknote in 22 of 25 foldings of the data. In the first row, the first predictor is along the *x*-axis and the second, third, and fourth predictors are along the *y*-axis, from left to right.

with specified class proportions. By generating data in this way, the latent response y is not used in determining the class labels of the data.

Analysis of the banknote authentication data set shows favorable results for the mixture prior Bayes classifier. The banknote authentication data set has a strongly localized structure that is well-suited for classification with this method. Even if the mixture prior classifier does not give the best result, this method is different enough from other methods to be potentially valuable in a classification ensemble.

3.5 Appendix: Derivation of Posterior Conditional Distribution for μ_m

As the distribution for μ_m is a mixture of normal densities, the Gibbs sampler must first select a kernel from the mixture distribution. The kernel should be selected with probability proportional to the marginal density of (3.4) with respect to the x_i. The marginal selection density for each kernel is attained by integrating out the μ_m over each x_i kernel.

Multiply each term in (3.4) by $e^{-\frac{1}{2}b(x_i)'(\Gamma^{-1}+n_mV_m^{(x)})^{-1}b(x_i)}$ and its inverse. Then each kernel density is multivariate normal with respect to μ_m and each kernel integrates to a common constant times

$$e^{-\frac{1}{2}x_i'\Gamma^{-1}x_i+\frac{1}{2}b(x_i)'(\Gamma^{-1}+n_mV_m^{(x)})^{-1}b(x_i)}. \tag{3.5}$$

Therefore selection of the kernel is proportional to $e^{-\frac{1}{2}x_i'\Gamma^{-1}x_i+\frac{1}{2}b(x_i)'(\Gamma^{-1}+n_mV_m^{(x)})^{-1}b(x_i)}$, and the quadratic term of (3.5) is

$$e^{-\frac{1}{2}x_i'(\Gamma^{-1}-\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}\Gamma^{-1})x_i} = e^{-\frac{1}{2}x_i'\Gamma^{-1}(\Gamma-(\Gamma^{-1}+n_mV_m^{(x)})^{-1})\Gamma^{-1}x_i}$$

$$= e^{-\frac{1}{2}x_i'\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}((\Gamma^{-1}+n_mV_m^{(x)})\Gamma-\mathbb{I})\Gamma^{-1}x_i} = e^{-\frac{1}{2}x_i'\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}n_mV_m^{(x)}x_i}$$

This is a multivariate normal density with respect to x_i with variance

$$\left(\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}n_mV_m^{(x)}\right)^{-1}. \tag{3.6}$$

To find the mean of this multivariate normal distribution, the x_i term in the exponential of (3.5) is

$$e^{x_i'\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}n_m((\bar{y}_m-\psi_m)V_m^{(xy)}+V_m^{(x)}\bar{x}_m)}, \tag{3.7}$$

from which we derive the mean of the kernel multivariate normal as the variance times the vector multiplying x_i' in (3.7):

$$\left(\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}n_mV_m^{(x)}\right)^{-1}\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}n_m((\bar{y}_m-\psi_m)V_m^{(xy)}+V_m^{(x)}\bar{x}_m)$$

$$= \left(\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}n_mV_m^{(x)}\right)^{-1}\left[\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}n_mV_m^{(x)}\right]$$

$$\times(V_m^{(x)})^{-1}((\bar{y}_m-\psi_m)V_m^{(xy)}+V_m^{(x)}\bar{x}_m)$$

$$= \bar{x}_m+(\bar{y}_m-\psi_m)(V_m^{(x)})^{-1}V_m^{(xy)}, \tag{3.8}$$

which is different from the mean of the selection kernels in the prediction model, where the mean is simply the component sample mean.

Therefore the selection probability is proportional to

$$e^{-\frac{n_m}{2}\left(x_i-\bar{x}_m-(\bar{y}_m-\psi_m)(V_m^{(x)})^{-1}V_m^{(xy)}\right)'\Gamma^{-1}(\Gamma^{-1}+n_mV_m^{(x)})^{-1}V_m^{(x)}\left(x_i-\bar{x}_m-(\bar{y}_m-\psi_m)(V_m^{(x)})^{-1}V_m^{(xy)}\right)},$$

the density for a multivariate normal with mean and variance as given in (3.6) and (3.8).

CHAPTER 4. A GENERIC CLASSIFICATION ROUTINE FOR CATEGORICAL PREDICTORS USING LIKELIHOOD RATIO STATISTICS AND RANDOM FORESTS

A paper in preparation

Cory L. Lanker[1,2], Wen Zhou[3], Stephen B. Vardeman[1], Max D. Morris[1]

Abstract

This paper outlines a generic classification algorithm given categorical predictors. The problem is that if too many categories are present or if many interaction levels affect the class probability function, no current methods can reduce bias effectively. The reasons for this is that the number of columns in the feature matrix is too large and requires dimension reduction or variable selection techniques that induce bias through the removal of key bias-reducing data pieces. Our solution is to have a generic way to characterize the information about the class probability function available in the predictors through likelihood ratio statistics. There is a theoretical justification for this based on sufficiency of these statistics, though an approximation is necessary to implement this data characterization method. We rely on random forests to reduce bias in our classifier by utilizing all information in the generated log likelihood ratio features. This algorithm is solely for predictive analytic purposes and is not used for estimation of predictor effects on the class probability function. A simulation study and an application data set demonstrate potential advantages of this classification method.

[1] Graduate student, University Professor, and Professor, respectively, Department of Statistics, Iowa State University.

[2] Primary researcher and author.

[3] Assistant Professor, Department of Statistics, Colorado State University.

4.1 Introduction

The problem considered in this paper is effective classification for categorical predictors that only indicate the unordered type of the observations. There are successful classification methods when there are a limited number of categories. One method used to analyze this type of data is one-hot encoding, a technique that makes a matrix of indicators for the various category levels in the data. With a feature matrix from one-hot encoding, one could use regularized logistic regression to determine the important category levels to include for classification. Alternatively, a random forest is a way to reduce classification bias without increasing variance.

These analyses allow modeling of category level effects on the class probability function. These methods can be successful in classification for a variety of data sets, but are less successful if there are a large number of categories or if two-factor or three-factor interactions are important. A large number of categories would have a one-hot encoded feature matrix of indicators with the same large number of columns. The number of columns in the feature matrix is typically far greater with interaction terms because of two reasons. First, there are more unique values for the interactions of any two factors than in the factors themselves. Second, there are many combinations of factors in order to consider all interactions. A large feature matrix suffers from a large number of predictors involved in the relationship between the predictors and the class probability function. In many of these cases, the number of features to consider gets too large to use current classification methods for categorical predictors.

This paper provides an algorithm to automate the process of feature generation using categorical predictors to improve classification. This algorithm does not aim to explain relationships between the class probability function and the predictors—just for predictive analytic purposes. The proposed algorithm targets situations when either there are a large number of categories within the predictors or there are many unknown but necessary interactions to consider in the class probability function. Our method works by using log likelihood ratio statistics, with theoretical justification for this method in Shao (2003). An approximation is necessary to implement this method on real data due to problem of zeros appearing in the logarithm's ratio. A sparse data correction factor is explored that may improve performance in some situations. This method relies on random forests as a bias reduction technique. The classifier described in this paper was successfully implemented in the feature matrix of the winning 2014 Data Mining Cup classification solution.

In this paper, Section 4.2 formally presents the problem and Section 4.3 develops our solution with log likelihood ratio statistics. Section 4.4 shows performance of our method in a simulation study of five categorical predictors and on a standard machine learning chess classification data set. Section 4.5 concludes the paper with a summary of our findings.

4.2 Classification Using Categorical Predictors

Suppose there exists a binary classification problem with predictors that consist only of categorical data, and that the information in these categorical predictors is unordered or nominal data. It could be that some of the predictors are unrelated with the class probability function. For predictors that are useful in classification, some category levels are indicative of membership in one class while other categories indicate membership in the other class. Even for informative predictors, some of the category levels may not be indicative of one class or the other. It may also be true that two predictors have levels that are independent of the response, but their two-factor interaction could be strongly predictive of class probability. Further, it may be that a group of predictors are uninformative of the class probability function when taken alone or in pairs, but have three-factor interactions that are informative.

Denote a single categorical predictor as x or a set of p categorical predictors as $x \equiv (x_1, \ldots, x_p)$, and denote w as the response of class 0 or 1. Let $f(x)$ be the class probability function representing the probability w is in class 1 when x is of category m or unique category combination $m \equiv (m_1, m_2, \ldots, m_p)$, depending on whether x represents a single predictor or multiple predictors. Suppose that there are N cases of data x containing J unique category levels or category level combinations, and w belongs to class 0 in N_0 cases and to class 1 in N_1 cases. Analysis of these data can be reduced into a contingency table of size $2 \times J$, where J is the number of unique categories. In an estimation problem, there are $J - 1$ free parameters to estimate—such estimation would be useful to understand the relationship of individual category levels on the class probability function.

In the classification context, we have a new observation x of one of the observed J categories and have to estimate the class probability function for w at x. This prediction is possible given estimates from the contingency table. Estimation of this kind becomes difficult as the number of categories becomes large relative to the size of the data or if considering multiple categorical predictors. If there are many categorical predictors each with many unique category levels, the number of categories J is likely very large and accurate estimation of the class probabilities would require an enormous amount of data. Expand the consideration to potential interaction effects and this approach becomes impossible to implement.

The goal of this paper is to develop a generic classification routine given categorical predictors, specifically when there are many predictors to consider, some or all predictors have a large number of categories, and there are potential unknown interaction effects. It could be that the class probability function is relatively independent with the predictors but depends heavily on two-factor or even three-factor combinations of the predictors. A technique in practice today is using a random forest classifier directly with categorical predictors. This bias-reducing technique is effective without increasing classification variance, meaning this method will generally minimize overfitting on a data set (Breiman, 2001). The random forest classifier is a collection of classification trees each with a unique bootstrap sample and randomly selected predictor choices for each decision split. Random forests have advantages of accuracy, minimal tuning, and induced randomization. Also, this classifier can effectively handle a large quantity of input data as by design they result in classifiers that ignore spurious predictors.

However, there are significant disadvantages of random forest classifiers given the context of classification problems considered in this paper. The number of partitions available to a decision tree explodes with the number of categories, which leads to overfitting of the data due to sparsity of data among the categories (Hastie et al., 2009). One way to circumvent the inability of random forest classifiers to handle a large number of categories is with one-hot encoding. In theory, the random forest can handle a large number of predictors, but in practice implementation slows as one-hot encoding greatly expands the number of variables. This obstacle makes one-hot encoding infeasible without variable selection, which is difficult to do while both retaining all information and avoiding overfitting. Regarding interaction effects on the class probability function, while random forests can consider interactions by splitting on a second predictor after splitting on a first predictor, this approach is indirect. Direct consideration of interaction effects via one-hot encoding also results in a problem of too many variables for the random forest to consider.

There are potential issues with using the one-hot encoding method for classification without random forests. A large number of categories for any predictor will necessitate a large number of indicator columns. Even if using a classification technique with variable selection such as regularized logistic regression, selection of the meaningful factor levels among many category levels becomes increasingly difficult. Screening predictors complicates the implementation by potentially increasing the required computation.

As the number of predictors increases, determining interaction effects becomes difficult. For p covariates that have potential two-factor interaction effects, there are $\frac{1}{2}p(p-1)$ combinations to consider. For each pair of variables there are more unique category combinations between the two predictors, and classification with one-hot encoding increases in difficulty as the number of necessary indicator columns explodes. This problem becomes worse for even higher-order interactions.

4.3 Generic Classification With Likelihood Ratios

We propose a classifier that has the above benefits of the random forest classifier while overcoming difficulties from having too many unique category levels or interaction effects. Our method is a generic way of extracting information in categorical predictors for classification using likelihood ratios. This technique utilizes the sufficiency of likelihood ratio statistics in that if we know the true likelihood ratio of class $w = 1$ to class $w = 0$ for all unique category arrangements m, we would have all the necessary information to make the best possible classification decision. The foundation for this idea is that this likelihood ratio is a minimal sufficient sufficient (Shao, 2003). We define the log likelihood ratio for predictors x having category combination m as

$$\log \frac{\hat{L}(w=1|x=m)}{\hat{L}(w=0|x=m)} = \log \frac{N_1(m)/N_1}{N_0(m)/N_0} \propto \log N_1(m) - \log N_0(m) \tag{4.1}$$

where $N_0(m)$ and $N_1(m)$ represent the number of observations of category combination m in class 0 and in class 1, respectively.

We estimate this ratio from the data, and in doing can have the problem of sparsity. If either $N_0(m)$ or $N_1(m)$ is zero in (4.1), meaning that there are no observations of category combination m in one or both classes, the logarithm will equal $+\infty$ or $-\infty$, falsely implying a strong class probability when due only to small sample sizes. To address this problem of sparse data leading to overly confident class predictions, we add $\frac{1}{2}$ to the empirical category sizes for both likelihood functions, yielding the estimated likelihood ratio

$$\log \frac{\hat{L}(w=1|x=m)}{\hat{L}(w=0|x=m)} \propto \log(N_1(m)+\tfrac{1}{2}) - \log(N_0(m)+\tfrac{1}{2}). \tag{4.2}$$

By adding a constant to the numerator and denominator in the log likelihood ratio, we approximate a Bayesian approach of adding a certain amount of prior information. Introducing the constant $\frac{1}{2}$ into the likelihood ratio effectively dampens the large log likelihood ratio values from categories with small samples while leaving the ratios relatively unaffected when there are sizable data. This ratio in (4.2) transforms a categorical predictor with a potentially large number of levels into a single quantitative feature that is an estimate of a minimal sufficient statistic for the information the predictors characterize.

This likelihood ratio statistic poses significant implementation problems when there are sparse data. Data sparsity is primarily reflected in the inability of obtaining good estimates for the likelihood ratio because there are too many category combinations J given the number of observations N. To maintain accuracy of the estimate of the likelihood ratio statistic in (4.2), the number of observations N must increase linearly with the number of unique category combinations J. Unfortunately sparse data conditions are practically guaranteed when the number of unique category combinations is large. As another approach, we analyze the predictors individually, looking at the p likelihood ratio statistics based on the individual predictors. Each predictor will have much fewer unique levels than the entire data set, so the data will be less sparse. In this way, the p categorical predictors are reduced to p quantitative features. We aim to approximate the overall likelihood ratio of class $w = 1$ to class $w = 0$, given $x = m$, with the combination of p individual likelihood ratios.

We also consider all two-factor interactions of categorical predictors. There may be enough data such that these combinations are not completely data sparse. Even if data sparsity exists with interactions, there may be some information available to further reduce the bias of the classifier based only on the p predictors. Considering all two-factor interactions requires only $\frac{1}{2}p(p-1)$ quantitative features. Higher-order interaction effects can also be added to the model, with $\frac{1}{6}p(p-1)(p-2)$ features required for all three-factor interactions.

We then input all of the generated quantitative features and the observation class labels into a random forest classifier. The random forest technique results in a classifier with bias reduction due to the relationship of class labels to the category levels when considering the predictor combinations represented in the feature matrix. Any features derived from predictor combinations that are independent with the class probability function will, on average, be ignored by the random forest. Therefore there is no disadvantage to adding many features that are unrelated with the response as the random forest classifier minimizes overfitting of the data. What is novel about this method is that the information extraction is done in an automated way, relying on random forests to sift through the large number of features. These features are generated in a way that both reduces classification bias and avoids overfitting.

We introduce a sparse data correction factor for use with this method. For a prior constant of $\frac{1}{2}$ if there is only one observation the resulting log ratio has the same value as five observations that split four-to-one. There may be data analysis situations where four-to-one is more informative than one-to-zero, and if there are a very large number of categories, increasing the prior constant may degrade performance. Define the sparse data correction factor $\gamma(m)$ as $(N(m)-1)/N(m)$ where $N(m)$ is the total number of observations of category m. The corrected log likelihood ratio estimate for category m, adjusted relative to the overall ratio of N_1 to N_0, is

$$\log\frac{\hat{L}(w=1|x=m)}{\hat{L}(w=0|x=m)} \propto \left(\log\frac{N_1(m)+\frac{1}{2}}{N_0(m)+\frac{1}{2}} - \log\frac{N_1}{N_0}\right)\gamma(m) + \log\frac{N_1}{N_0}. \tag{4.3}$$

Note that the $\log(N_1/N_0)$ term omits the constant of $\frac{1}{2}$ added to address the problem of infinite ratios. As the values of N_1 and N_0 are large, this overall ratio should be adequately estimated.

4.4 Demonstration of Our Method

The aim of our simulation study is to determine when this likelihood ratio feature generation method is competitive with categorical input into random forests considering (1) number of category levels, (2) even or uneven distribution of data among the categories, and (3) when the class probability function is primarily a function of the predictors, two-factor interactions, or three-factor interactions. We also consider when the sparse data correction factor lowers misclassification rates. We conclude with classification results of a standard machine learning chess classification data set.

4.4.1 Simulation study

The simulation study assumes that there are five independent categorical predictors, each with M unique category levels, where M is either 10 or 40 for all predictors. There are 20,000 observations with probability $p_{m,k}$ of being in category m for predictor k, with $\sum_{m=1}^{M} p_{m,k} = 1$ for each k. The simulation study is run in R using the library randomForest for the first 25 starting seeds.

There are twelve cases considered that entail the possible combinations of three factors. First, six cases each have 10 or 40 category choices for each predictor, denoted $M = 10$ and $M = 40$. Second, six cases each have a realistic long tail distribution of category proportion sizes or a more evenly spread portion size distribution, denoted $\alpha = 3$ and $\alpha = 30$. Third, four cases each have the class probability function decided by the main category levels themselves, by the two-factor interactions, or by the three-factor interactions.

Let $x_{i,k}$ denote the unique category level for observation x_i while considering predictor or interaction k. A standard normal variable is drawn for $\beta_k(x_{i,k})$, denoting the probit probability contribution of category level $x_{i,k}$ for predictor or interaction k. The class probability function for each observation is $\Phi^{-1}\Big(\beta_0 + \sum_{k=1}^{K} \beta_k(x_{i,k})\Big)$. Note that for this simulation study, β_0 always equals zero, designating equal class probabilities. Therefore the class $w = 1$ probability is the normal probability left of the resulting value when considering all category levels and interaction combinations an observation has. The data class labels are then drawn from a Bernoulli distribution with those class probabilities.

Classifier performance is compared through the ratio of misclassification error. When using categorical input the random forests are composed of 1000 classification trees. Due to computational limitation the random forests consist of only 75 trees when using likelihood ratio inputs. The cases have 10 or 40 unique category levels for each of the five predictors. We use the value of 40 to represent a larger number of categories that will still run with the R function randomForest, which has a limit of 53 categorical levels. The random forests are first run with categorical inputs and then rerun with likelihood ratio inputs, comparing the resulting misclassification error rates of each input method. For the likelihood ratio feature matrix, all main effects, two-factor interactions, and three-factor interactions are considered. Given the five predictors, there are only a total of 25 features in the likelihood ratio feature matrix despite the large number of categories.

The probabilities $p_{m,k}$ for each predictor k come from the stick-breaking process for $M + 1$ components, rescaling the first M probability draws to be a valid probability that sums to one. The stick-breaking process is described in Sethuraman (1994). There are two choices for the stick-breaking probabilities, and these are generation using Beta(α, α^2) with α equal to 3 or 30. The cumulative probabilities given α of 3 and 30 are displayed in Figure 4.1. When $\alpha = 3$, there are a few categories with a majority of observations, while for $\alpha = 30$ the category proportions are more evenly distributed. The $\alpha = 3$ case is more representative of the spread of proportions found in real data that appears to have a long tail in the distribution of observations to categories.

The random forests classification results for the log likelihood ratio inputs compared to categorical inputs are shown in Figure 4.2. This figure shows the ratio of misclassification error given a likelihood ratio feature matrix versus a categorical predictor matrix.

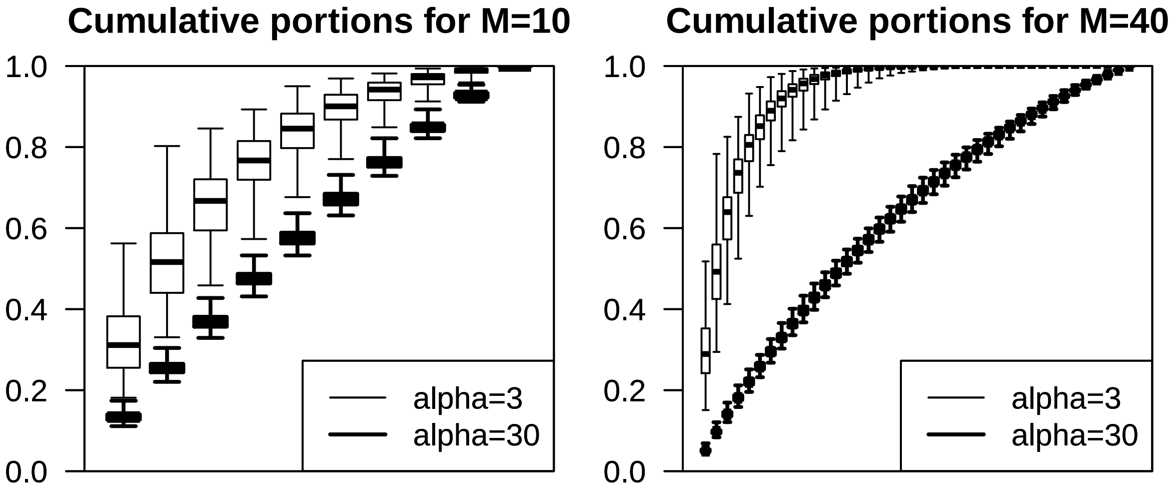

Figure 4.1 Category Proportions for Different α Values. Resulting cumulative proportions from the stick-breaking generation process. Boxplots represent the spread of category proportion sizes when reordered from largest to smallest for the 25 runs of this study. The strong curvature in the $\alpha = 3$ case indicates that much of the probability is assigned to a small portion of the categories.

There are two conclusions from this plot. For the first conclusion, consider the main effects class probability function. The likelihood ratio input performs worse compared to categorical inputs when there are only 10 category levels but performs much better when there are 40 choices per predictor. This implies that the random forest classifier performs less effectively on categorical inputs as the number of categories increases. The second conclusion is that the comparative advantage to using likelihood ratio inputs is greater when there are many categories across all class probability function scenarios. Note that the variability of the advantage is less when the class probability function depends on higher-order interactions. Further, the advantage is strongest when the distribution of category proportion sizes is not even.

Considering the three-factor interaction scenario when there are 40 categories per predictor and the categories are more evenly spread across predictors, there are not

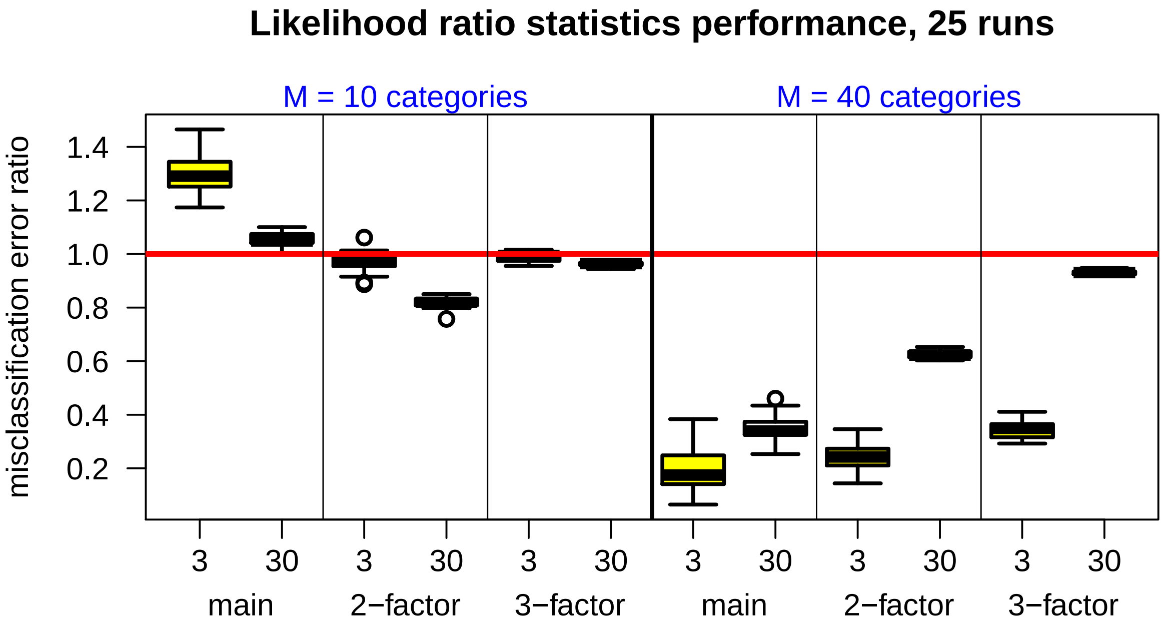

Figure 4.2 Ratio of Misclassification Error for Random Forests With Different Input. This plot shows misclassification error ratio for random forests with likelihood ratio inputs to random forests with categorical predictor input for the 25 runs of the simulation study. Values below one indicate use of likelihood ratio inputs yields lower misclassification error. The x-axis represents the two α category proportion spread values, with 30 representing a more even distribution of category proportion sizes. The left three panels have 10 category choices for each of the five predictors, while the right three panels have 40 category choices for the predictors. From left-to-right, the panels indicate that the class probability function is primarily a function of the predictor main effects without interactions, one-factor interactions without main effects, and two-factor interactions without main effects or one-factor interactions.

enough observations for each of the approximately 40^3 unique interaction levels for the likelihood ratio method to have an advantage. For the two-factor interactions scenario, there are only 40^2 unique interaction levels and are enough data for the likelihood ratio to have a significant advantage. This implies that the likelihood ratio has an advantage when there is a moderate amount of data for the categories, but not a large amount as there is in the $M = 10$ cases, especially given main effects generation.

The results for the sparse data correction factor are shown in Figure 4.3. From this plot we see that for low number of categories there is no effect from the multiplier, likely due to the fact there are enough data in such a limited number of categories, even when there are three-factor interactions. Where the correction factor demonstrates an advantage is when there are many category levels, in particular for class probability functions dominated by main effects or two-factors interactions. As these later cases are more realistic in nature and represent how sparsity appears in real categorical data sets, the use of such a correction factor could be important in proper implementation of this method.

4.4.2 King-rook versus king-pawn data set

The King-Rook Versus King-Pawn classification data set is a standard machine learning test set from the University California–Irvine (UCI) Machine Learning Repository (Bache and Lichman, 2013). This test data set is analyzed to demonstrate analysis of a data set with class probability function of the type that is a potential application of this classification method. These data consist of characterizations of 3196 chess end-game scenarios when White has a king and rook and Black has a king and pawn on A7, one square from queening. The next move belongs to White. The 3196 rows represent different possible arrangements of pieces on the chess board meeting the above requirements, and the response is whether it is possible for White to win from that point. Knowing the arrangements White can win helps a computer decide moves throughout the chess

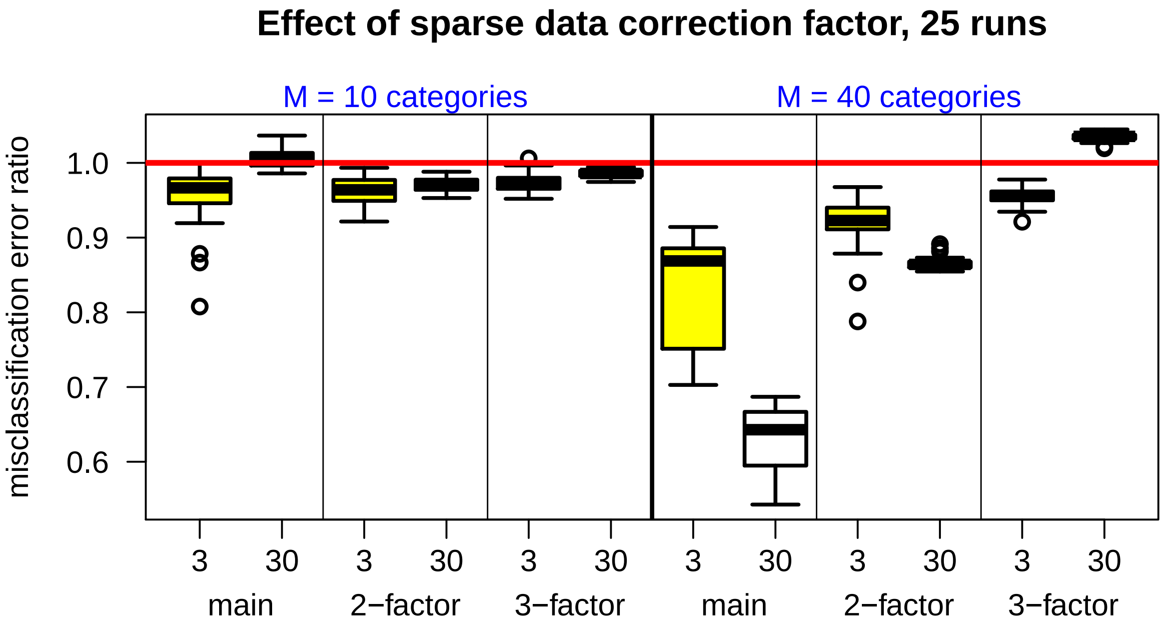

Figure 4.3 Ratio of Misclassification Error With Sparse Data Correction Factor. This plot shows misclassification error ratio for random forests with likelihood ratio inputs with the sparse data correction factor to random forests without this correction factor for the 25 runs of the simulation study. Values below one indicate use of the sparse data correction factor results in lower misclassification error. The x-axis represents the two α category proportion spread values, with 30 representing a more even distribution of category proportion sizes. The left three panels have 10 category choices for each of the five predictors, while the right three panels have 40 category choices for the predictors. From left-to-right, the panels indicate that the class probability function is primarily a function of the predictor main effects without interactions, one-factor interactions without main effects, and two-factor interactions without main effects or one-factor interactions.

match that may result in the present arrangement. In 1669 of the positions, White can win, and in the other 1527 positions, White cannot win. For each arrangement, there are 36 predictors to characterize the position of the board. For example, one predictor represents if the Black king is on the eighth row. These predictors are mostly binary, with a few having three category levels. The 28th predictor is removed as it is determined to be uninformative. The classification goal is to predict if White can win given a certain arrangement of the board.

Testing uses 10-fold cross-validation with 25 random foldings. The average misclassification error for random forests with likelihood ratio input is .0040 while with categorical input is .0159. The better performance using likelihood statistics is likely due to the fact that the categories are evenly spread and the class probabilities are defined by a mix of main effects, two-factor interactions, and three-factor interactions. When using the sparse data correction factor the average misclassification error is .0041, a difference that is not significant given these 25 runs.

4.5 Conclusion

This paper outlines a generic classification algorithm given categorical predictors. Effective bias reduction is challenging for classification if too many categories are present or if many interaction levels affect the class probability function. Our solution introduces a generic way to characterize the predictor's categorical information about the class probability function by using likelihood ratio statistics. We rely on random forests as a way to reduce bias with the many generated log likelihood ratio features while avoiding overfitting.

BIBLIOGRAPHY

Bache, K., and Lichman, M. (2013), "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, URL: http://archive.ics.uci.edu/ml.

Banks, D. L., Olszewski, R. T., and Maxion, R. A. (2003), "Comparing Methods for Multivariate Nonparametric Regression," *Communications in Statistics–Simulation and Computation*, 32, 541–571.

Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M. (2011), "Nonparametric Bayes Regression and Classification Through Mixtures of Product Kernels," *Bayesian Statistics 9*, pp. 145–164.

Breiman, L. (2001), "Random Forests," *Machine Learning*, 45, 5–32.

Clarke, B. S., Fokoué, E., and Zhang, H. H. (2009), *Principles and Theory for Data Mining and Machine Learning*, New York, NY: Springer.

Diebolt, J., and Robert, C. P. (1994), "Estimation of Finite Mixture Distributions Through Bayesian Sampling," *Journal of the Royal Statistical Society, Series B*, 56, 363–375.

Gamerman, D., and Lopes, H. F. (2006), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, Boca Raton, FL: CRC Press.

Gelfand, A. E., and Smith, A. F. M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, 398–409.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013), *Bayesian Data Analysis* (3rd ed.), Boca Raton, FL: CRC press.

Gerritsma, J., Onnink, R., and Versluis, A. (1981), "Geometry, Resistance, and Stability of the Delft Systematic Yacht Hull Series," *International Shipbuilding Progress*, 28, 276–297.

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning* (2nd ed.), New York, NY: Springer.

Izenman, A. J. (2009), *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*, New York, NY: Springer.

Kang, C., and Ghosal, S. (2009), Clusterwise Regression Using Dirichlet Mixtures,, in *Advances in Multivariate Statistical Methods*, ed. A. Sengupta, World Scientific Publishing Company, Singapore, pp. 305–325.

McLachlan, G., and Peel, D. (2004), *Finite Mixture Models*, New York, NY: John Wiley & Sons.

Melnykov, V., Chen, W. C., and Maitra, R. (2012), "MixSim: An R Package for Simulating Data to Study Performance of Clustering Algorithms," *Journal of Statistical Software*, 51, 1–25.

Melnykov, V., and Maitra, R. (2010), "Finite Mixture Models and Model-Based Clustering," *Statistics Surveys*, 4, 80–116.

Müller, P., Erkanli, A., and West, M. (1996), "Bayesian Curve Fitting Using Multivariate Normal Mixtures," *Biometrika*, 83, 67–79.

Pace, R. K., and Barry, R. (1997), "Sparse Spatial Autoregressions," *Statistics & Probability Letters*, 33, 291–297.

Penrose, K. W., Nelson, A. G., and Fisher, A. G. (1985), "Generalized Body Composition Prediction Equation for Men Using Simple Measurement Techniques," *Medicine and Science in Sports and Exercise*, 17, 189.

Sethuraman, J. (1994), "A Constructive Definition of Dirichlet Priors," *Statistica Sinica*, 4, 639–650.

Shao, J. (2003), *Mathematical Statistics* (2nd ed.), New York, NY: Springer.

Simonoff, J. S. (1996), *Smoothing Methods in Statistics*, New York, NY: Springer.

Wade, S., Walker, S. G., and Petrone, S. (2014), "A Predictive Study of Dirichlet Process Mixture Models for Curve Fitting," *Scandinavian Journal of Statistics*, 41, 580–605.

Wasserman, L. (2004), *All of Statistics: A Concise Course in Statistical Inference*, New York, NY: Springer.

Yeh, I. C. (1998), "Modeling of Strength of High-Performance Concrete Using Artificial Neural Networks," *Cement and Concrete Research*, 28, 1797–1808.