# BAYESIAN TESTS FOR CHECKING THE EQUALITY OF DISTRIBUTIONS, WITH APPLICATION TO SCREENING VARIABLES FOR CLASSIFICATION

A Dissertation

by

NAVEED NABEEL MERCHANT

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTORATE OF PHILOSOPHY

| | |
|---|---|
| Chair of Committee, | Jeffrey D. Hart |
| Co-Chair of Committee, | Debdeep Pati |
| Committee Members, | Ximing Wu |
| | Lan Zhou |
| Head of Department, | Brani Vidakovic |

May 2022

Major Subject: Statistics

ABSTRACT

A new framework for nonparametrically testing of equality of two densities is proposed. From this framework, two different tests are constructed. The two tests themselves are then investigated and compared to other tests on simulated and real data. After establishing their legitimacy, the tests are applied to choose variables for classification problems. We study the benefit of these tests, when they are useful and what classification techniques work best in conjunction with them. The method is then applied to simulated data sets and real data sets where the number of variables is large.

# DEDICATION

To my mother, my father, and my brother.

CONTRIBUTORS AND FUNDING SOURCES

## Contributors

This work was supported by a dissertation committee consisting of Professor Hart as lead chair, Professor Pati and Professor Zhou of the Department of Statistics, and Professor Wu of the Department of Agricultural Economics.

The data sets analyzed in all chapters can be found from the UCI Machine Learning Repository.

Authors of "Use of Cross-Validation Bayes Factors to test Equality of Two Densities": are Naveed Merchant, Jeffrey Hart, and Taeryon Choi. The paper has been submitted to the Bayesian Analysis journal, but is still undergoing revisions and is not yet accepted or published.

Authors of "A Bayesian Motivated Two-sample Test Based on Kernel Density Estimates" and "Screening Methods for Classification Based on Non-parametric Bayesian tests" are Naveed Merchant and Jeffrey Hart.

All other work conducted for the dissertation was completed by the student independently.

## Funding Sources

There are no funding sources and we declare no conflict of interest.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

# 1. INTRODUCTION AND ROAD-MAP

## 1.1 Introduction

This is a sandwich-style dissertation, consisting of three papers, and an introduction and a conclusion. The goal of this section is to provide an idea as to how the dissertation will flow.

Chapter 2 provides an overview of feature selection methods and other Bayesian tests that can check for the equality of distributions. Chapter 3 proposes a new Bayesian approach for testing equality of two distributions, the Cross-Validation Bayes Factor, and compares it to its Bayesian competitor, the Pólya tree two sample test. Performance is evaluated on simulated and real datasets, and a proof for consistency of the test is discussed. Chapter 4 eliminates a tuning parameter in the Cross-validation Bayes Factor, and adapts the Bayesian test to an exact frequentist style approach, somewhat improving its power and increasing its speed. Consistency of this test is discussed, and the methodology is applied for checking if data from different classes in univariate and bivariate problems share the same distribution. Chapter 5 applies the test in Chapter 4 to the variable selection problem. Cases where this method is suitable for that problem and can do better than its competitors are discussed, by comparing how the variable selection improves other models and observing in simulations which of the variable selection methods correctly retain important variables.

# 2. LITERATURE REVIEW

## 2.1 Introduction

A common problem in many statistical methods is choosing a subset of variables to include in a model. The purpose of this is two-fold. One, for extremely large data sets, fitting the model itself can be time consuming, and spending time fitting models and checking how good each one is can simply take too much time. Fitting a model with fewer predictors can decrease the amount of time it takes to fit a model by a huge amount, and in some cases is the only way to proceed. The other purpose is to get rid of variables that are irrelevant for prediction. This is often the case for data where the amount of samples, $n$, is significantly smaller than the number of predictors $p$. This is a commonly occuring problem in genomic data, where only a small subset of genes is likely responsible for causing changes in the response.

There has been significant research in the field of variable selection. Even in the field of variable selection, however, statisticians typically need to preprocess their data sets. Often, it is too time consuming to apply variable selection methods to a very large data set with a large number of variables, and instead variables must be screened from a huge set before variable selection begins. After some important variables are screened, then further variable selection can be done, to pick the important ones from those that are screened. Proceeding directly to variable selection can make algorithms incredibly time consuming.

## 2.2 A review of variable selection in the machine learning literature

A wide variety of feature selection methods have been researched. Trying to list all of them is essentially impossible, but we can give a review of the general types of methods that are typically used.

### 2.2.1 Filtering methods

A filtering method chooses a subset of the variables such that each variable alone provides useful information in a model. These methods are also known as screening methods. These methods

examine how each variable alone provides information, and as a result tend to be very fast, but can miss variables that work jointly to predict the response. In addition, it is common for these methods to pick multiple variables that lie in similar directions, which can result in multicollinearity. For ultra high dimensional data sets, it is common to run a screening method before applying methods in general. The advantage of these methods lies in their speed, moreso than in their effectiveness at choosing the best subset of variables to use.

### 2.2.2 Wrapper methods

The idea behind wrapper methods is to create several models, and then to evaluate how well each model is fit. Common methods used for evaluation include: cross-validation error rates, BIC, and AIC. The problem can then be reduced to searching for a subset where the chosen criterion is optimized. Forward selection was once a common method in this field, where the best variable set would be chosen in a greedy fashion. While a greedy method lacks gurantees for finding the optimal subset in a general setting, some sort of heuristic is required in searching for the best subset. If there are $p$ predictors, then $2^p$ models must be considered if all choices of predictors are considered. Since $p$ is typically large, a strategy not using a heuristic is computationally prohibitive. The heuristic's intention is to decrease the number of models fit and quickly find a good model. There are search methods outside of greedy strategies though, particle swarm optimization being one (3). As these methods create the models and evaluate their effectiveness, they typically do much better than filtering type methods at selecting important variables. However, unlike filtering methods, because they must create many models, and than in turn evaluate their effectiveness, they are significantly slower. It is impossible to apply these methods to a full data set and expect a fast result.

### 2.2.3 Embedded methods

Embedded methods create a single model that has some sparsity inherently embedded into it. They fit the model and perform variable selection at the same time, and as a result tend to be much faster than wrapper methods. They are still slower than filter methods, but some can avoid

running into problems of multicollinearity. Furthermore, they are capable of picking variables that alone are not good predictors, but offer predictive power when included with other variables. LASSO (4), Elastic net (5), Bayesian inference using spike and slab priors (6), and sparse BART type procedures (DART, soft BART, and spike-and-tree BART are examples) (7) are examples of embedded methods that are commonly used. However, the computational complexity of these variable selection methods is still quite high. BART type methods in general are well known for their accuracy as well as their large run time, while LASSO type methods are known to have computational complexity scaling cubically with the number of predictors.

There is an argument that viewing LASSO's time complexity as cubically scaling is pessimistic, especially considering how fast coordinate descent is at solving it, but there still remains a matter of selecting a tuning parameter. While there are some defaults (picking by cross-validation), the choice of tuning parameter can still dramatically change results. In addition, LASSO is applied as a type of regression. For example, classification can be done with logistic regression with LASSO by imposing an $L_1$ norm penalty on the coefficients. Some coefficients will be chosen, but changing the model might result in different variables being selected. Using a probit regression with a $L_1$ norm penalty on the coefficients is an alternative way to use LASSO, but will not necessarily pick the same coefficients.

While BART was quite fast in the simulations presented later, it should be noted as well that mixing time of MCMC methods in general can vary in how they scale with the number of predictors. Some conditions have been given that ensure that the mixing time of some MCMC methods scale linearly with the number of predictors (8), but if they are not met, it is possible that MCMC times can scale polynomially, or even exponentially with the number of predictors. While the MCMC methods done on BART are fast, mixing time has been recorded as an issue for high dimensions (9). BART type methods typically perform variable selection by choosing predictors that are used most often in fitting trees. However, it is not obvious how often a variable needs to be picked to be determined to be a good predictor. This ends up being an important tuning parameter that needs to be carefully selected. Typical methods select one by repeatedly permuting the

response, and examining how often variables are picked in permutations. The probability distribution of the number of times a variable is included can provide instead guidance on picking a threshold (10), but there are two problems with this. Firstly, repeatedly computing BART models is time consuming. Even if we are willing to ignore this, BART methods can be improved upon by screening before application. As a result, a question should be asked as to whether new trees should be constructed using the variables selected. If they are, then the computation time of this has grown even more, and a question on when to stop making BART models also arises. A partial answer to this question will be given later.

The goal of this proposal will be to introduce a new filtering method built for the classification setting. We can resort to using embedded methods after using a filter method to assess how good the variables chosen by the filter method were. First we review other filtering methods that have been seen recently.

### 2.2.4 A review of recent filtering methods

Sure independence screening (SIS) was built for regression, but can be applied to classification as well (11). It is a popular method to perform preliminary screening as it only requires centering and scaling the data and then computing $X^TY$, where $X$ is the design matrix and $Y$ is a vector of responses. As a result, the total complexity of the algorithm is only $O(np)$. In the continuous case, this method corresponds to picking the variables that are most correlated with the response, or alternatively, picking variables for which $X_j^TY$ is largest, where $X_j$ corresponds to the $j$th predictor. As matrix multiplication can be parallelized, this computation can be sped up further if $n$ and $p$ are prohibitively large. Suppose there is a true underlying model (for example $P(Y = 1|X_s) = \text{logit}(X_s^T\beta)$, where $X_s$ is a subset of columns of $X$), and a variable $X_k$ is deemed important if $X_k \in X_s$. The method has been shown to possess a sure screening property, which guarantees that all important variables are retained with probability tending to 1. For the 2 class classification setting, (11) suggested that their test devolves into comparing 2-sample $t$-statistics, with the samples corresponding to the different classes being compared. Their suggestion then, is to simply pick variables that correspond to the largest test statistics computed. An iterative proce-

dure is proposed that uses SIS with LASSO to pick variables that jointly provide information, as well as screen out variables that can cause issues with multicollinearity.

Model Free Feature Screening (or SIRS) is intended to be used regardless of model as well, but can easily be applied for the classification setting (12). The goal instead is to pick variables for which $E[x_k F(y|x_k)]^2$ is large in magnitude, where $y$ is a response, and $x_k$ is a predictor. This is of greater time complexity than SIS, as a cdf must be computed every time a variable is picked. To estimate this for predictor $k$ from the sample, the quantity $\tilde{w_k} = \frac{1}{n}\sum_{j=1}^{n}[\frac{1}{n}\sum_{i=1}^{n} X_{ik}1[Y_i < Y_j]]^2$ must be computed, which implies the method is $O(n^2 p)$. Regardless, this computation can be parallelized across predictors. Variables are picked if the corresponding $E[x_k F(y|x_k)]^2$ is higher than some prespecifed amount $\gamma$, an important tuning parameter. The recommended way to choose this tuning parameter is to generate normal, independent noise, and see how large $\gamma$ need be before it can filter out the induced noise. Conditions are provided that give it the property that it will keep all important variables with probability tending to 1. Similar to SIS, an iterative procedure is recommended that tries to pick variables that jointly provide information. Whatever the case, the need to choose a tuning parameter is the biggest criticism of this method, as it is difficult to directly interpret it.

Kolmogorov Distance Screening shares the sure screening property under weaker conditions than SIS (13), and is about as fast as SIS. This test is not as general as SIS or SIRS, and instead restricts its scope solely to binary classification. The idea is to compute Kolmogorov-Smirnov test statistics and use variables with large test statistics as the important variables.

Model-free feature screening for ultra high dimensional discriminant analysis (or MV SIRS) uses a version of SIRS that does better for classification (14). Variables are picked instead if $\sum_{r=1}^{2} p_r \int [F_r(x_k) - F(x_k)]^2 \mathrm{d}F(x)$ is large, where $F_1(x) = P(X \leq x|Y = 1)$, $F_2(x) = P(X \leq x|Y = 2)$, and $F(x) = P(X \leq x)$. Alternatively, this is the same as picking $x_k$ where $E_{x_k}[\mathrm{Var}_Y(F(x_k|Y)]$ is large. This is again of greater time complexity than SIS, and should be of the same time complexity as SIRS. It can intuitively be thought of as picking variables whose cdf conditional on the class is different in the $L_1$ norm from the unconditional cdf.

Robust Model Free Feature Screening (15) is the only method with weaker conditions than Model Free Feature Screening. This method standardizes all variables, then transforms them into Gaussian variables using the paranormal transform. On these new variables, Henze-Zirkler's test for multivariate normality is applied. This can also be done in parallel and is of complexity $O(n^2p)$.

If we want to perform variable selection for classification, then SIRS should devolve into checking how the expected value of the cdf changes with each different class. A more powerful way to check which variables are useful could heuristically involve checking if the distributions of the different classes are the same for each predictor. There is the problem of choosing a tuning parameter for Zhu's method; however we would avoid running into this problem if we use tests based on Bayes factors. The interpretability of Bayes factors in 2-sample tests for checking equality of distributions provides a natural cutoff to pick. Of course, a procedure to determine a cutoff based on what Zhu et al. attempted in their method would probably also work. A test that checks if distributions differ can capture a larger class of variables than those that just check the mean, so this test might inherit the sure screening property under the same conditions as for SIS. It is likely that the conditions required for this test to pick all important variables with probability tending to 1 would be weaker than those required by SIS.

Before we delve more into the efficacy of this sort of test, we first review some two-sample Bayesian tests.

### 2.3 A review of two-sample Bayesian testing

Suppose $X_1, X_2, ..., X_m$ are i.i.d. as $f$ and independently $Y_1, Y_2, ..., Y_n$ are i.i.d. as $g$. A classic problem is to test the hypothesis that $f$ and $g$ are identical.

#### 2.3.1 Pólya tree-based methods

A nonparametric Bayesian solution to this problem is based on a Pólya Tree prior (1). The Pólya tree prior itself can be described as a prior on bins of a distribution. We treat $f$ and $g$ as histograms, and the prior itself specifies a distribution over all potential histograms. To do this, the Polya process produces histograms in an iterative fashion to draw observations from the underlying

distribution. Figure 1 illustrates this for an iterative scheme with 3 levels.

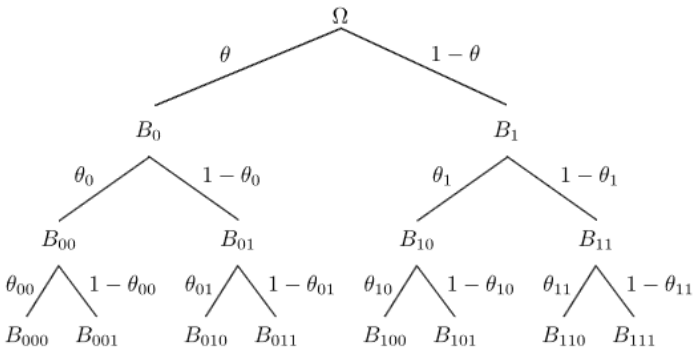



Figure 2.1: *An iterative scheme for Polya Tree as proposed by Holmes (1).* The sample space, $\Omega$, has been partitioned in to tiny bins and there are probabilities for finding an observation in each bin.

In the figure the sample space, $\Omega$, has been partitioned. At the first level, a drawn observation can lie in either bin $B_0$ or $B_1$. The probability it lies in bin $B_0$ is $\theta$ and the probability it lies in bin $B_1$ is $1-\theta$. We then partition each bin into two more bins, and may do so indefinitely. $B_0$ is partitioned into $B_{00}$ and $B_{01}$ and $B_1$ is partitioned into $B_{10}$ and $B_{11}$. We denote these finer partitions as the second level. The probability of an observation lying in $B_{00}$ is given as $\theta\theta_0$ and we can similarly find the probability that an observation lies in any other bin at the second level. At the third level, we have 8 bins, with the probabilities of the observations in the $8$ bins being as follows: $(\theta)(\theta_0)(\theta_{00})$, $(\theta)(\theta_0)(1-\theta_{00})$, $\theta(1-\theta_0)\theta_{01}$, $\theta(1-\theta_0)(1-\theta_{01})$, $(1-\theta)(\theta_1)\theta_{10}$, $(1-\theta)\theta_1(1-\theta_{10})$, $(1-\theta)(1-\theta_1)\theta_{11}$, and $(1-\theta)(1-\theta_1)(1-\theta_{11})$. We do not know the bin probabilities in practice, so a prior is placed on these probabilities. In the figure, the prior for the $8$ probabilities is applied by placing priors on the 7 parameters: $\theta$, $\theta_0$, $\theta_1$, $\theta_{00}$, $\theta_{01}$, $\theta_{10}$, and $\theta_{11}$. For each of these parameters, we use a beta prior. Once we are given draws from a distribution, we can compute the number that occur in a particular bin, which can be interpreted as a multinomial likelihood. We placed a beta prior on the probabilities to induce a conjugate form for the posterior, and hence that prior is used so that marginal likelihoods can be computed easily. Holmes suggests that if the two distributions are

the same, then the marginal likelihood of the full data set fit under one Pólya tree will be larger than the product of the marginal likelihoods for the individual datasets. Bayes consistency is proven, the only problem being how to create bins and how to choose the parameters for the beta prior. Those parameters do not seem to be too troublesome based on our examination. However choosing the bins does have a major impact. Holmes et al. (2015) recommend centering and scaling the joint data set to have 0 mean and standard deviation 1. They then recommend constructing the bins by using quantiles of the normal distribution. While we can verify that this does work well in some situations, it tends to work poorly if the data set has heavy tails. If the dataset is Cauchy, then the mean and variance estimates will be unstable, and results can vary in this case. Holmes recommends continuing to partition bins until the Bayes factor seems to stabilize upon adding new bins. The computational complexity of this method is difficult to say, but the rough scaling of it is $O(np2^j)$, where $j$ is the number of times each bin is more further partitioned. The scaling of $j$ should vary with distribution, but it should scale at worst at a log rate with $n$.

The optional Pólya tree can be viewed as a generalization of the idea in Holmes et al. (2015) (16). A rule is given as to when to stop constructing bins. We bisected each bin that was created in Holmes's method, whereas in the Ma and Wong (2011) method we draw from a categorical distribution to determine how finely we divide a bin. Finally, we note that this method is more computationally intensive than Holmes et al. (2015), as it requires repeated computation of when to stop, and how many bins to divide a bin into, which is something that Holmes's method simply assumes. Finding the computational complexity of this is even more tricky. Due to its computationally difficult nature, any screening tests based on the Pólya tree should rely on using the version of Holmes et al. (2015).

# 3. USE OF CROSS-VALIDATION BAYES FACTORS TO TEST EQUALITY OF TWO DENSITIES

## 3.1 Abstract

We propose a non-parametric, two-sample Bayesian test for checking whether or not two data sets share a common distribution. The test makes use of data splitting ideas and does not require priors for high-dimensional parameter vectors as do other nonparametric Bayesian procedures. We provide evidence that the new procedure provides more stable Bayes factors than do methods based on Pólya trees. Somewhat surprisingly, the behavior of the proposed Bayes factors when the two distributions are the same is usually superior to that of Pólya tree Bayes factors. We showcase the effectiveness of the test by proving its consistency, conducting a simulation study and applying the test to Higgs boson data.

## 3.2 Introduction

In frequentist hypothesis testing, there is no universal statistic whose values are interpretable across different problems. In contrast, Bayes factors *do* have a universal interpretation. When the prior probabilities of two hypotheses are the same, the Bayes factor is the ratio of posterior probabilities of the two hypotheses. This is a compelling motivation for developing objective Bayesian procedures that depend only minimally on prior distributions. (17) proposed the use of *cross-validation* Bayes factors (CVBFs) to compare the fit of parametric and nonparametric models. The CVBF is an objective Bayesian procedure in which the nonparametric model is a kernel density estimate, the simplest version of which cannot typically be used in a Bayesian analysis since it only becomes a model once it is computed from data. This problem is sidestepped by computing a kernel estimate from a subset of the data, and then using the estimate as a model for the remainder of the data. As detailed by (18), the notion of a CVBF is also useful in a purely parametric context, wherein data splitting allows one to compare two parametric models via a legitimate Bayes factor that does not require a prior distribution for either model.

The purpose of the current paper is to explore CVBFs in the problem of comparing densities corresponding to two different populations. Given independent random samples $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_n$ from densities $f$ and $g$, respectively, we wish to test the null hypothesis that $f$ and $g$ are identical against the alternative that they are different, *without specifying a parametric model for either density.* This is accomplished by means of a Bayes factor that makes use of data splitting and kernel density estimates. Unlike the setting of either (17) or (18), both hypotheses in the current setting are nonparametric, which necessitates different techniques to show that CVBFs behave desirably. In particular, it is of interest to prove that a CVBF is Bayes consistent when either $f \equiv g$ or the two densities are different. Although the current investigation is restricted to comparison of two densities, we will lay the groundwork for justifying the use of CVBFs in other settings where both hypotheses are nonparametric.

A classic Bayesian approach for checking the equality of two densities involves the construction of priors on the elements of a wide class of distributions. For testing goodness of fit and obtaining posterior predictive distributions, (19) proposes methodology based on a Pólya tree prior constructed from a centering distribution. Methods that use a similar strategy for checking equality of two densities have been suggested by (20), (21) and (1). (22) propose the use of restricted dependent Dirichlet process priors when testing the equality of distributions against ordered alternatives. Both (1) and (21) use their Bayes factors in frequentist fashion, i.e., they choose rejection regions to produce desired type I error probabilities. In our opinion, such an approach is not truly Bayesian. If one uses a traditional level of significance such as 0.05, this practice yields a test with the unsettling property that in some cases the hypothesis of equal densities is rejected when the Bayes factor *favors* equal densities. We prefer an approach that chooses the hypothesis of unequal densities only when the odds in favor of unequal densities has increased in light of the data.

In Section 3.5 it will be seen that any two-sample procedure based on Bayes factors ultimately depends on the difference between entropy estimates. (23) have suggested using either a kernel density estimate or histogram to estimate entropy and provide conditions under which these types of estimators are consistent. In addition, (23) use entropy estimates to check equality of distribu-

tions. Entropy estimates are also seen in the information gain filter in machine learning methods for picking important features; see (24). The current paper makes use of results in (25), who proves consistency of entropy estimates that rely on data-driven smoothing parameters.

An important issue is that of Bayes factor consistency, which we address in Section 3.5. Suppose the Bayes factor is defined so that values smaller than 1 favor the hypothesis of equal densities. Then consistency means that the Bayes factor converges in probability to $0$ when the densities are equal and diverges to $\infty$ when the densities are unequal. (1) contains a proof showing that their Bayes factor is consistent. We argue that our cross-validation Bayes factor is consistent as well. Moreover, when the densities are equal, we argue that a cross-validation Bayes factor converges to $0$ at a much faster rate than do Bayes factors based on traditional Bayesian methods.

A main motivation for our proposed methodology is its conceptual simplicity. The models used are kernel density estimates from training data and each one depends on but a single parameter, a bandwidth. In contrast, the approaches of (19) and (1) depend on choice of base distribution and $2^k$ parameters, where $k$ is typically at least 10. One also needs to choose a prior for all these parameters, although (1) propose one that requires specification of just one parameter. In simulations in Section 3.7 we will compare our method with that of (1), and show that the odds ratios produced by the latter test can be highly sensitive to the choice of base distribution.

The rest of the paper may be outlined as follows. In Section 3.3 we describe in detail our methodology for the two-sample problem. Section 3.4.1 considers the choice of kernel and also the prior used for the bandwidth parameter, and Section 3.4.2 investigates the use of a Laplace approximation for marginal likelihoods. In Section 3.5 we provide theoretical evidence that our Bayes factor is consistent, and in Section 3.6 we discuss methods for choosing the training set sizes. Finally, Sections 6 and 7 are devoted to a simulation study and real-data analysis, respectively, and concluding remarks are given in Section 8.

### 3.3 Methodology

Suppose that we observe independent random samples $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_n$ from cumulative distribution functions $F$ and $G$, respectively. We assume that $F$ and $G$ have respective

densities $f$ and $g$, and the goal is to test the following hypotheses by means of a Bayesian approach:

$$H_0 : f \equiv g \quad \text{vs.} \quad H_a : f \not\equiv g.$$

We wish to use kernel density estimates to do the testing, and in order to do so we will use the CVBF idea. In contrast to the setting of (17), both the null and alternative hypotheses are non-parametric, and hence training data will be used to formulate the alternative *and* null models. The Bayes factor will then be computed from validation data.

We first introduce some notation. For an arbitrary collection of (scalar) observations $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$, define the kernel density estimate (KDE) $\hat{f}(\cdot\,|h, \boldsymbol{Z})$ by

$$\hat{f}(x|h, \boldsymbol{Z}) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - Z_i}{h}\right),$$

where the kernel $K$ is a probability density and $h > 0$ is the bandwidth. For the moment, all we ask of $K$ is that it be symmetric about 0, unimodal and have finite variance.

Now, partition $X_1, \ldots, X_m$ into $\boldsymbol{X}_T = (X_1, \ldots, X_r)$ and $\boldsymbol{X}_V = (X_{r+1}, \ldots, X_m)$, and likewise $Y_1, \ldots, Y_n$ into $\boldsymbol{Y}_T = (Y_1, \ldots, Y_s)$ and $\boldsymbol{Y}_V = (Y_{s+1}, \ldots, Y_n)$. Under $H_0$ there is a common density, call it $f$. The model for $f$ will be $M_0 = \{\hat{f}(\cdot\,|h, \boldsymbol{X}_T, \boldsymbol{Y}_T) : h > 0\}$. In other words, we pool the two training sets together and use these data to estimate the common density $f$. Under the alternative we have separate models for $f$ and $g$, which are $M_X = \{\hat{f}(\cdot\,|\alpha, \boldsymbol{X}_T) : \alpha > 0\}$ and $M_Y = \{\hat{f}(\cdot\,|\beta, \boldsymbol{Y}_T) : \beta > 0\}$.

Let $\pi$, $\pi_X$ and $\pi_Y$ be priors for $h$, $\alpha$ and $\beta$, respectively. The likelihood under $H_0$ is

$$L_0(h) = \prod_{i=r+1}^{m} \hat{f}(X_i|h, \boldsymbol{X}_T, \boldsymbol{Y}_T) \prod_{j=s+1}^{n} \hat{f}(Y_j|h, \boldsymbol{X}_T, \boldsymbol{Y}_T).$$

The likelihood under $H_a$ is

$$L_a(\alpha, \beta) = \prod_{i=r+1}^{m} \hat{f}(X_i|\alpha, \boldsymbol{X}_T) \prod_{j=s+1}^{n} \hat{f}(Y_j|\beta, \boldsymbol{Y}_T) = L_X(\alpha) L_Y(\beta),$$

and the cross-validation Bayes factor (CVBF) is

$$\begin{aligned} CVBF &= \frac{\int_0^\infty \int_0^\infty \pi_X(\alpha)\pi_Y(\beta)L_a(\alpha,\beta)\,d\alpha d\beta}{\int_0^\infty \pi(h)L_0(h)\,dh} \\ &= \frac{\int_0^\infty \pi_X(\alpha)L_X(\alpha)\,d\alpha \cdot \int_0^\infty \pi_Y(\beta)L_Y(\beta)\,d\beta}{\int_0^\infty \pi(h)L_0(h)\,dh}. \end{aligned} \tag{3.1}$$

Interestingly, each of $M_0$, $M_X$ and $M_Y$ is a parametric model, inasmuch as each depends on just a single parameter, a bandwidth. It should be acknowledged that we know with certainty that, for example, $M_X$ *does not* contain the true density $f$. However, there is a key difference between $M_X$ and a traditional one-parameter model. Since KDEs are consistent estimators, we have reason to believe that some members of $M_X$ will be quite close to $f$, especially if the training set size $r$ is large. In contrast, members of a traditional one-parameter model, such as all $N(\mu, 1)$ densities, would be close to the truth only under very special circumstances. So, in spite of being formally "wrong," $M_X$ can be expected to be a good model, which echoes the sentiment of George Box in his famous quote about statistical models.

Even if one objects to our models not formally containing the truth, the same criticism can arguably be leveled against the Pólya tree approach of (1). Each element of the parameter space in that approach is of histogram type, and since one usually envisions a certain degree of smoothness in the underlying density, the true density does not necessarily lie in the parameter space employed by Pólya trees.

We close this section with some remarks about our methodology.

- The quantity (A.8) is referred to as a cross-validation Bayes factor (17) since each data set is split into two parts. For example, the data $X_1, \ldots, X_m$ are split into a training set, $\boldsymbol{X}_T$, and a validation set, $\boldsymbol{X}_V$.

- In spite of the fact that the models being compared in $CVBF$ are formulated from data, it is important to appreciate that $CVBF$ *is* a legitimate Bayes factor. This is because the models are defined from data that are independent of the validation sets $\boldsymbol{X}_V$ and $\boldsymbol{Y}_V$. The Bayesian

paradigm does not specify *where* posited models must come from, so long as they are not defined from the data used to evaluate those models.

- By assuming that the bandwidths $\alpha$ and $\beta$ are a priori independent, the computation of $CVBF$ reduces to calculating three separate marginals, each of which has the form dealt with in (17).

- (26) show that the $L_1$ norm difference between a kernel density estimate and the true density tends to 0 for any kernel integrating to 1 as long as the sample size $n$ tends to $\infty$, the bandwidth $h$ tends to 0, and $nh \to \infty$. Because of results like this, the conventional wisdom in kernel density estimation is that the choice of kernel $K$ is not overly important. This is not at all the case in the current context. In Section 3.4.1 we will point out the importance of using relatively *heavy-tailed* kernels, a specific version of which is proposed.

- Ideally, the value of $CVBF$ should not depend on the particular data split chosen. To deal with this problem, we advise that one compute an average of log-$CVBF$ values obtained from multiple splits of the data. Doing so also has the benefit of making the Bayes factor more stable. Geometric means of Bayes factors have also been employed by (27) in the context of intrinsic Bayes factors.

- (28) discusses the notion of "updating consistency," according to which the conclusions of a Bayesian procedure should not depend on whether all available data are considered at once, or Bayesian analyses are applied sequentially to partitions of the data set. The only way CVBF can be even provisionally update consistent is if only one data split is considered. However, if one averages log-$CVBF$ values from different data splits, as advised in the last bullet point, then CVBF is *not* update consistent. We therefore acknowledge that CVBF fails to be Bayesian in the full sense of the term.

### 3.4 Implementation issues

Some practical issues must be addressed in order to make use of CVBFs. A kernel has to be chosen for each of the KDEs, and priors for the bandwidths of the KDEs are needed. Furthermore, the integrals defining the three marginals cannot (in general) be computed analytically, and hence approximations of the integrals are necessary. We first address the choice of kernel and priors.

#### 3.4.1 Choice of kernel and priors

For densities $f_1$ and $f_2$, the Kullback-Leibler divergence between $f_1$ and $f_2$ is defined to be

$$KL(f_1, f_2) = \int_{-\infty}^{\infty} f_1(x) \log\left(\frac{f_1(x)}{f_2(x)}\right) dx.$$

As will be discussed in Section 3.5, consistency of our proposed Bayes factor depends crucially on the behavior of $KL(f, \hat{f}(\cdot\,|\alpha, \boldsymbol{X}_T))$ and $KL(g, \hat{f}(\cdot\,|\beta, \boldsymbol{Y}_T))$. (25) shows that, when the bandwidths $\alpha$ and $\beta$ are chosen by likelihood cross-validation, the right sort of kernel needs to be used to ensure that these divergences are well-behaved. A number of practical and technical difficulties arising from tail behavior of the underlying density are eliminated if one uses a relatively heavy-tailed kernel. A kernel that suffices in this regard is the following that was proposed by (25):

$$K_0(z) = \frac{1}{\sqrt{8\pi e}\,\Phi(1)} \exp\left[-\frac{1}{2}(\log(1+|z|))^2\right], \tag{3.2}$$

where $\Phi$ is the standard normal distribution function.

The perils of using light-tailed kernels in conjunction with leave-one-out likelihood cross-validation are well-established in the literature (29). The case of finite support kernels illustrates the problem. If the support of the kernel is $(-1, 1)$, then any bandwidth smaller than the difference between the two largest order statistics will produce a likelihood of 0. This has disastrous effects for sufficiently long-tailed distributions (such as the Cauchy) since it entails that the bandwidth maximizing the likelihood cross-validation function will diverge to $\infty$. (25) advises that when using likelihood cross-validation to choose a bandwidth, one should use a kernel with tails that are,

roughly speaking, at least as thick as those of the true density. Hence, likelihood cross-validation can fail even when certain infinite support kernels, including the Gaussian, are used. In contrast, when $K_0$ is used and the data are Cauchy, then (25) shows that the likelihood cross-validation bandwidth is asymptotic to the minimizer of expected Kullback-Leibler loss. Our theory shows that these results for the Cauchy distribution are also true for the version of cross-validation used in the current paper.

If one is confident that the tails of the underlying density are no heavier than those of a Gaussian density, then it would be appropriate to use a Gaussian kernel in our procedure. Simulations we have done suggest that the Gaussian kernel produces somewhat more stable Bayes factors than does $K_0$ in the case of light-tailed densities. In practice, though, we often do not know how thick the density's tails are, and so it is better to use $K_0$, which works works well for a wider variety of densities than do Gaussian and other lighter-tailed kernels. We will thus use $K_0$ for all simulations and data analyses in this paper.

The prior we propose for each bandwidth is as follows:

$$\pi(h|\gamma) = \frac{2\gamma}{\sqrt{\pi}h^2} \exp\left(-\frac{\gamma^2}{h^2}\right) I_{(0,\infty)}(h). \tag{3.3}$$

Prior (A.5) is the same type as used successfully by (17). An aspect of (A.5) that we find appealing is that it tends to $0$ as $h$ tends to 0. This is in concert with the fact that, due to the data-driven nature of our kernel density estimation models, we are (essentially) a priori certain that the very smallest bandwidths produce untenable densities.

Despite depending on just the one parameter $\gamma$, which serves as both location and scale, our experience suggests that (A.5) works very well as long as $\gamma$, the mode of (A.5), is chosen appropriately. We propose that for each marginal, $\gamma$ be chosen to equal the maximizer of the corresponding likelihood. For example, for the marginal $\int_0^\infty \pi_X(\alpha)L_X(\alpha)\,d\alpha$, we take $\pi_X \equiv \pi(\,\cdot\,|\hat{\gamma})$, where $\hat{\gamma}$ is the maximizer of $L_X$. The scale of $\pi(\,\cdot\,|\gamma)$ is proportional to $\gamma$, which entails that the prior $\pi(\,\cdot\,|\hat{\gamma})$ has low information relative to the likelihood. This is because the standard deviation of the

cross-validation bandwidth $\hat{\gamma}$ is $o(\hat{\gamma})$, a fact that is ensured by using the kernel $K_0$. Centering a low information prior at the maximizer of the likelihood is akin to using a unit reference prior (30) centered at the data, which by now is a fairly common practice.

Choosing $\gamma$ to be a constant independent of sample size is not necessarily a good practice. Some analysis shows that doing so can produce bandwidths that are asymptotically too large unless the training set size is chosen small enough. For this reason we suggest using (A.5) with $\gamma = \hat{\gamma}$, at least until more research is done on the question of choosing priors and their parameters.

### 3.4.2 Laplace approximation

Interestingly, there exists a closed-form expression for each marginal *if* one uses a Gaussian kernel in conjunction with a prior of the form (A.5). This results from the fact that, for example, $\pi(\alpha|\gamma)L_X(\alpha)$ is a linear combination of functions each of which is proportional to a function of the form $\alpha^k \exp(-A/\alpha^2)$, whose integral over $(0, \infty)$ may be expressed in terms of the gamma function. The practical usefulness of this closed-form solution is limited for two reasons. First, as was noted in Section 3.4.1, the Gaussian kernel is not a good all-purpose kernel, and secondly it turns out that more computations are required for the closed form solution than for standard methods of approximating integrals. The solution requires $r^{m-r}$ sums to be computed, where $r$ is the size of the training set and $m - r$ the size of the validation set. For these reasons we will not pursue the closed form solution further.

In general, the integrations required to calculate a $CVBF$ cannot be done analytically. (17) used numerical integration to approximate marginal likelihoods, either by simple or adaptive quadrature. Other methods that could be used are importance sampling, bridge sampling or a Laplace approximation. A Laplace approximation has the advantage of being less computationally intensive. Let $\hat{h}$ be the maximizer of $L_0$ and define

$$\hat{H} = -\frac{\partial^2}{\partial h^2} \log L_0(h)\Big|_{h=\hat{h}}.$$

Then the Laplace approximation of $\int \pi(h)L_0(h)\,dh$ is

$$\int \pi(h)L_0(h)\,dh \approx \sqrt{\frac{2\pi}{\hat{H}}} \cdot \pi(\hat{h})L_0(\hat{h}).$$

The quantity $\hat{H}$ can be expressed as a functional of kernel estimates based on the kernel $K$ and two other related kernels. An expression for $\hat{H}$ may be found in the Supplementary material.

To investigate how well the Laplace approximation works in our context we generate samples from standard normal and standard Cauchy distributions and compare the Laplace approximation of the marginal likelihood with an approximation using the R function `integrate` (which uses adaptive quadrature). Samples of sizes 200, 500 and 1000 were generated from each of the two distributions. The kernel used was $K_0$, and the prior was (A.5) with $\gamma$ taken to be the maximizer of the likelihood. The training set size was always 1/4 of the sample size, and 500 replications for each $n$ and distribution were considered. Table 1 summarizes the results.

**Normal data**

| $n$ | Median | Interquartile range |
|---|---|---|
| 200 | $6.99 \cdot 10^{-4}$ | $3.46 \cdot 10^{-4}$ |
| 500 | $2.66 \cdot 10^{-4}$ | $1.80 \cdot 10^{-4}$ |
| 1000 | $1.33 \cdot 10^{-4}$ | $2.77 \cdot 10^{-4}$ |

**Cauchy data**

| $n$ | Median | Interquartile range |
|---|---|---|
| 200 | $2.10 \cdot 10^{-5}$ | $3.21 \cdot 10^{-6}$ |
| 500 | $3.32 \cdot 10^{-6}$ | $3.99 \cdot 10^{-7}$ |
| 1000 | $8.09 \cdot 10^{-7}$ | $1.07 \cdot 10^{-7}$ |

Table 3.1: *Relative error of Laplace approximation of marginal likelihood. Each median and interquartile range is based on 500 replications. The measure of error is* $|(log\hat{M} - \log M)/\log M|$, *where* $M$ *and* $\hat{M}$ *are quadrature and Laplace approximations, respectively, of the marginal.*

The Laplace approximations were excellent. The median relative error was no larger than 0.000699 for any $n$ or distribution, and for each distribution the error became smaller as the sample size increased. A plot of the results for $n = 500$ is given in Figure 3.1. We also found that the computations for our Laplace approximation are 7 to 8 times faster at $n = 1000$ than those for the quadrature approximation when running on an 8 core Intel Skylake 6132 CPU running at 2.6GHz

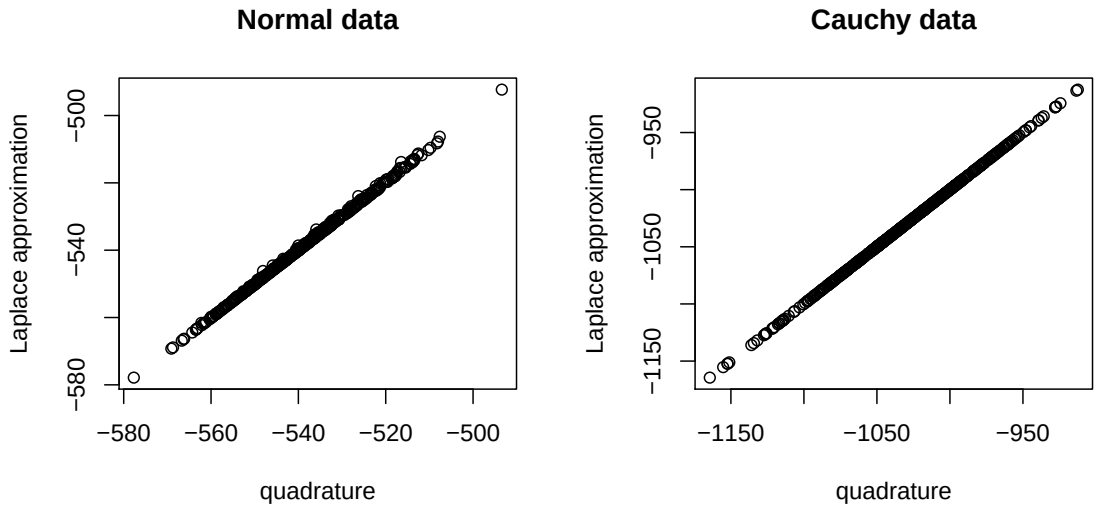



Figure 3.1: *Laplace and quadrature approximations to log-marginal likelihoods.* These results are for the case where the training set and validation sizes were $125$ and $375$, respectively.

with 32GB of 2666MHz DDR4 memory. For these reasons we will use the Laplace approximation in all subsequent simulations and examples.

We note that the parameter of our prior is chosen so that the prior mode is equal to the maximizer of the likelihood. This has two computational benefits. First of all, our algorithm starts by determining the maximizer of the log-likelihood, which is necessary to avoid underflow problems. But once this maximizer has been determined it is not necessary to find the posterior mode since the two quantities are one and the same. Secondly, choosing the prior parameter in this way renders null the distinction between the two versions of the Laplace approximation, one using the maximizer of $L_0$ and the other the maximizer of $\pi L_0$.

### 3.5 Bayes consistency

Here we address Bayes consistency of a $CVBF$ in the two-sample problem.

#### 3.5.1 Large sample behavior of $CVBF$

We begin with a list of assumptions, and then state a theorem.

A1. The Laplace approximation of each of the three marginals is asymptotically correct in that the log of the marginal likelihood is equal to the log of the Laplace approximation plus a term that is negligible in probability relative to the approximation.

A2. The densities $f$ and $g$ are bounded away from 0 and $\infty$ on $(-\lambda, \lambda)$ for each $\lambda > 0$, with $f(x) \sim c_1 x^{-a_1}$ and $f(-x) \sim c_2 x^{-a_2}$ as $x \to \infty$, where both $c_1$ and $c_2$ are positive and $a_1$ and $a_2$ larger than 1. Density $g$ satisfies the same properties as $f$, albeit with possibly different constants.

A3. The second derivatives $f''$ and $g''$ exist and are bounded and almost everywhere continuous on $(-\infty, \infty)$. In addition, for a constant $C_2 < \infty$,

$$|f''(x)| \leq C_2 x^{-a_1-2} \quad \text{and} \quad |f''(-x)| \leq C_2 x^{-a_2-2} \quad \text{for } x > 1,$$

where $a_1$ and $a_2$ are the same as in A2. The function $g''$ satisfies the same properties as $f''$ with possibly different constants.

A4. The kernel used is $K_0$, as defined in (A.4).

A5. The prior is (3) and its parameter is chosen as described in Section 3.1.

Condition A1 simplifies the proof by allowing us to approximate the ratio of marginal likelihoods by a likelihood ratio. Conditions A2 and A3 are those of (25) and are needed to ensure that the maximizer of the likelihood cross-validation criterion is optimal in a Kullback-Leibler sense. (25) provides another set of conditions that could be used in place of A2 and A3. These conditions deal with compactly supported densities, but for the sake of brevity we do not repeat these. Assumption A4 is needed to guard against cases where one or both of the underlying densities are long-tailed. It is not needed, for example, if the data are Gaussian. In that case it would suffice to use a Gaussian kernel. However, since one does always know what type of tail behavior to expect, it is better to use a kernel that works well in all cases, and $K_0$ is one such kernel.

**Theorem 1.** *Suppose that assumptions A1-A4 hold, and that $r$ and $s$ each tend to $\infty$ in such a way that $r = o(m)$, $r = o(n)$, $s = o(m)$ and $s = o(n)$ as $m$ and $n$ tend to $\infty$. Then if $f \equiv g$ and $m$ and $n$ tend to $\infty$*

$$\begin{aligned} \log(CVBF) &= C_f \left\{ (m-r) \left[ \frac{1}{(r+s)^a} - \frac{1}{r^a} \right] + (n-s) \left[ \frac{1}{(r+s)^a} - \frac{1}{s^a} \right] \right\} \\ &\quad + o_p \left( \frac{(m-r)}{r^a} + \frac{(n-s)}{s^a} \right), \end{aligned} \tag{3.4}$$

*where $C_f$ is a positive constant and $0 < a < 4/5$ is a constant determined by $f$.*

*If instead $\int |f - g| > 0$, $r/(r+s) \sim m/(m+n)$, $m/(m+n) \to q$ as $m, n \to \infty$ and $0 < q < 1$, then as $m, n \to \infty$*

$$\begin{aligned} \log(CVBF) &= (m-r) KL(f, qf + (1-q)g) \\ &\quad + (n-s) KL(g, qf + (1-q)g) + o_p(m+n). \end{aligned} \tag{3.5}$$

### 3.5.2 Some heuristics for the proof

Theorem 1 is proven in our Supplementary material, but here we will provide some heuristics for the proof. Until further notice we assume that $m$ and $n$ are balanced in the sense that $m/n$ converges to a positive constant as $m$ and $n$ tend to infinity. The reader is reminded that $L_0(h)$ and $L_X(\alpha)L_Y(\beta)$ are the likelihoods corresponding to the null and alternative models, respectively. To a good approximation $\log(CVBF)$ is a log-likelihood ratio:

$$\log(CVBF) \approx \log(LR) = \log\left(L_X(\hat{\alpha})\right) + \log\left(L_Y(\hat{\beta})\right) - \log\left(L_0(\hat{h})\right),$$

where $\hat{\alpha}$, $\hat{\beta}$ and $\hat{h}$ are the maximizers of $L_X$, $L_Y$ and $L_0$, respectively. So, roughly speaking, we can expect our cross-validation Bayes factor to perform as desired when the maximized likelihood corresponding to the true model is larger than the other maximized likelihood.

To simplify notation, let $\hat{f}$, $\hat{g}$ and $\tilde{f}$ denote the KDEs calculated from the $X$-training data, the

$Y$-training data and all training data, respectively. When $f \equiv g$, we have

$$\log(LR) \sim m[K(\tilde{f}, f) - K(\hat{f}, f)] + n[K(\tilde{f}, f) - K(\hat{g}, f)]. \tag{3.6}$$

Each of the four Kullback-Leibler divergences in (3.6) converges in probability to 0, owing to the consistency of the three kernel density estimates. Importantly, however, each of $[K(\tilde{f}, f) - K(\hat{f}, f)]$ and $[K(\tilde{f}, f) - K(\hat{g}, f)]$ is negative with probability tending to 1 as $r$ and $s$ tend to $\infty$, which follows from the results of (25) and the fact that $\tilde{f}$ is based on ever more data than either $\hat{f}$ or $\hat{g}$. But since (a) each of the Kullback-Leibler divergences tends to 0 no faster than $(r+s)^{-1}$ and (b) $r$ and $s$ are of smaller order than $m$ and $n$, it follows that $\log(LR)$ diverges to $-\infty$ with probability tending to 1 as $m$ and $n$ tend to infinity.

A key to appreciating what transpires when $\int |f - g| > 0$ is to understand that $\hat{f}$, $\hat{g}$ and $\tilde{f}$ consistently estimate $f$, $g$ and $qf + (1-q)g$, respectively, where $q$ is the limit of the ratio $r/(r+s)$. In this case the analog of (3.6) is

$$\log(LR) \sim mK(f, qf + (1-q)g) + nK(g, qf + (1-q)g).$$

The last expression diverges to infinity when $0 < q < 1$ since both of the K-L divergences are positive in that case.

### 3.5.3 Implications of Theorem 1

We now consider with more rigor implications of Theorem 1. Without loss of generality we assume that $m \leq n$, which entails that $q$, the limit of $m/(m+n)$, is no more than $1/2$. If $f \equiv g$, Theorem 1 implies that

$$\log(CVBF) \sim C_f n r^{-a} A_q, \tag{3.7}$$

where

$$A_q = \left(\frac{q}{1-q}\right)(q^a - 1) + \left(\frac{q}{1-q}\right)^a [(1-q)^a - 1].$$

Due to the facts $0 < a < 4/5$, $r = o(n)$ and $A_q < 0$, it follows that $CVBF$ diverges to $-\infty$ as $r$ and $n$ tend to $\infty$ and hence Bayes consistency follows in the case $f \equiv g$.

When $f \not\equiv g$, i.e., $\int |f - g| > 0$, Theorem 1 implies that

$$\log(CVBF) \sim (n+m)\left[qK(f, f_q) + (1-q)K(g, f_q)\right], \tag{3.8}$$

where $f_q$ is the mixture density $qf + (1-q)g$. We have $\int |f - f_q| = (1-q)\int |f - g| > 0$, and by Pinsker's inequality it follows that $K(f, f_q) > 0$. Similarly $K(g, f_q) > 0$, with the consequence that Bayes consistency holds in the alternative case as well.

Concerning the sizes of $r$ and $s$, Theorem 1 suggests that these quantities be allowed to grow with $m$ and $n$ at a very slow rate. For example, if $m = n$, $r = s$ and $r \sim \log m$, then $\log CVBF$ will tend to $-\infty$ at the rate $n/(\log n)^a$ when $f \equiv g$ and to $\infty$ at rate $n$ when $\int |f - g| > 0$. On the other hand, the quality of the asymptotics underlying Theorem 1 relies on the sizes of $r$ and $s$, and this is a motivation to avoid extremely small values of $r$ and $s$. For example, the estimate of $[q/(1-q)]KL(f, f_q) + KL(g, f_q)$ resulting from use of small training set sizes may be inadequate.

A remarkable aspect of Theorem 1 is its implication that, under the null, $CVBF$ tends to 0 at a much faster rate than is typical for traditional Bayesian tests. Suppose, for example, that $m = n$, $r = s$ and $r \sim n^\gamma$ for $0 < \gamma < 1$. Then $\log(CVBF) \sim -Cn^{1-a\gamma}$ for a positive constant $C$. Since $0 < a < 4/5$, this entails that $\log(CVBF)$ tends to $-\infty$ at a rate that can be arbitarily close to $n$. In contrast, when one uses a nonparametric Bayesian procedure in which the null model is nested within the alternative, the log-Bayes factor typically diverges to $-\infty$ at a rate that is only logarithmic in the sample size; see, for example (31).

It is also important to point out that Theorem 1 remains true for an *average* of finitely many $\log(CVBF)$ values (corresponding to different training samples). The benefit of an average would show up in the $o_p$ terms of (A.6) and (A.7), although it is beyond the scope of the current paper

to quantify the benefit. The reader is referred to (18) for a theoretical result concerning average $\log(CVBF)$ in the context of parametric models.

### 3.5.4 Ratio of training set sizes differs from ratio of sample sizes

Theorem 1 assumes that $r/(r+s)$ and $m/(m+n)$ have the same limit, $q$. This is done to make the presentation of results more concise and to allow us to use results from (25) to prove the theorem. It turns out that this assumption is not necessary for consistency under either hypothesis. Suppose that $r/(r+s) \to \rho$ and $m/(m+n) \to q$, where $\rho \neq q$ (and $q > 0$). First of all, if $f \equiv g$, the only difference from the result in Theorem 1 is that the constant $A_q$ in (3.7) is replaced by a different negative constant. The same proof as before applies because of the fact that $\hat{f}_X$, $\hat{f}_Y$ and $\hat{f}_{X,Y}$ all estimate the same density.

For each $0 \leq p \leq 1$, let $f_p(x) = pf(x) + (1-p)g(x)$. To appreciate how $\rho \neq q$ affects the case $f \not\equiv g$, it is important to recognize that in this setting there are *two* mixture densities at play: $f_\rho$ and $f_q$, which represent the limiting distributions of the combined training data and the combined validation data, respectively. It turns out that this fact has a profound impact on the behavior of $\hat{h}$, the maximizer of $L_0$. Unlike classical bandwidth theory, $\hat{h}$ need not tend to 0 when $\rho \neq q$ and $f$ differs from $g$. To describe what happens, define, for each $h > 0$, the density

$$f_{\rho,h}(x) = \int \frac{1}{h} K\left(\frac{x-y}{h}\right) f_\rho(y)\, dy.$$

Importantly, if $h > 0$ is fixed as $r, s$ tend to $\infty$, then $\hat{f}(\cdot|h, \boldsymbol{X}_T, \boldsymbol{Y}_T)$ is consistent for $f_{\rho,h}$.

Now, the quantity $(n+m-r-s)^{-1} \log L_0(h)$ is an unbiased estimator of the risk function

$$J(h) = \int f_{\tilde{q}}(x) \log f_{\tilde{q}}(x)\, dx - E\left[K(f_{\tilde{q}}, \hat{f}(\cdot|h, \boldsymbol{X}_T, \boldsymbol{Y}_T))\right],$$

where $\tilde{q} = (m-r)/(m+n-r-s)$. This suggests that the maximizer of $L_0$ converges to the minimizer, $h_0$, of the K-L discrepancy $K(f_q, f_{\rho,h})$. (It has been verified in multiple special cases

that, indeed, $h_0$ need not be 0.) The consequence of these results for CVBF is that

$$\log(CVBF) \sim (n+m)\left[qK(f, f_q) + (1-q)K(g, f_q) + K(f_q, f_{\rho,h_0})\right]. \tag{3.9}$$

Note that (3.9) agrees with (3.8) when $\rho = q$ since in that case $h_0 = 0$ and hence $f_{\rho,h_0} \equiv f_q$. The most important thing about (3.9) is its implication that $CVBF$ is still Bayes consistent when $\rho \neq q$. Indeed, the constant in (3.9) is at least as big as that in (3.8).

### 3.5.5 Unbalanced sample sizes: $m/(m+n) \to 0$

Inspection of (3.8) indicates a problem when the sample sizes are unbalanced, i.e., $m/(m+n) \to 0$. In that case the terms $qK(f, f_q)$ and $(1-q)K(g, f_q)$ are each $o(1)$, which is true in the latter case since $K(g, f_q)$ tends to 0 when $q \to 0$. This implies that $\log(CVBF)$ is $o(n)$, bringing into question whether or not $CVBF$ is even consistent. The problem is that when $m/(m+n)$ is very small, the log-likelihood is dominated by the $Y$-sample, and if $r/(r+s)$ is also very small, the KDE computed from the combined training data will look very similar to the KDE computed from the $Y$-training data. This problem can be rectified by selecting $r$ and $s$ in such a way that $r/(r+s)$ does not tend to 0. We recommend that the two training set sizes be chosen so that $r \sim s$ as $m$ and $n$ tend to $\infty$. Then, when $f \not\equiv g$, expression (3.9) becomes

$$\log(CVBF) \sim nK(g, f_{1/2,h_0}).$$

We cannot absolutely rule out the possibility that $K(g, f_{1/2,h_0})$ could be 0, but it seems as if this would happen only in pathological cases.

## 3.6 Choice of training set size

We now discuss the choice of training set sizes $r$ and $s$ for given sets of data. Two competing ideas are at play when choosing these quantities. First of all, it is desired that the KDEs from the training data be good representations of $f$ and $g$, a desire that calls for large $r$ and $s$. On the other hand, we would like as much data as possible for computing the Bayes factor, which asks that

$m - r$ and $n - s$ be large.

### 3.6.1 General considerations

The sizes of $r$ and $s$ as well as $r/s$ have an impact on the behavior of CVBF. We recommend, as discussed in the previous section, that $r/s$ be asymptotic to 1/2. Doing so will, in general, produce consistent Bayes factors whether the sample sizes are balanced or not. This leaves us with the question of how to choose $r$ (since once this is done $s = r$.) Our theory suggests that $r$ should be large but very small relative to $m$. As noted previously, however, there is a price to pay for taking $r$ to be too small. Simply put, when $r$ and $s$ are too small, the resulting density estimates may be poor enough that the larger of the two maximized likelihoods does not coincide with the truth. Ultimately, though, we view our methodology as being the most appropriate when $m$ and $n$ are quite large. In such cases we have found that choice of $r$ is not problematic, in that any of a large number of values for $r$ will result in high quality density estimates while leaving plenty of data for validation. In many cases choice of $r$ becomes moot. Let $CVBF(r)$ denote the value of $CVBF$ corresponding to a training set size of $r$, and suppose that one regards a Bayes factor larger than $T$ to be convincing evidence against $f \equiv g$, and a Bayes factor less than $1/T$ to be convincing evidence in favor of $f \equiv g$. Then if either $CVBF(r) > T$ or $CVBF(r) < 1/T$ for all $r$ over a reasonable range of $r$ values, then the conclusion is clear without the necessity of choosing $r$.

When $m$ and $n$ are large, but only in the hundreds, say, it is more important that $r$ be a substantial fraction of $m$, since otherwise the size of the training sets may be too small. In such cases we feel that reasonable choices for $r$ and $s$ are $r = [m/2]$ and $s = [n/2]$, where $[x]$ denotes the integer closest to but not larger than $x$. Aside from the issue of statistical efficiency, $r = [m/2]$ and $s = [n/2]$ are not suggested for very large sample sizes since they *maximize* the length of time needed to compute the Bayes factor. To explain why, consider the marginal based on just $X_1, \ldots, X_m$. This marginal requires calculation of $m - r$ kernel estimates, each of which involves $r$ additions. So, a total of $r(m - r)$ operations are required for each likelihood evaluation, and this number is maximized when $r = m/2$. This result is of particular interest for extremely large data sets. In this case it is unlikely that half of the full data set is required for computing a good training

density, and hence significant reductions in computing time can be gained by choosing a training set size that is much smaller than the validation set size.

Dependence of Bayes procedures on tuning parameters or hyperparameters, especially in non-parametric settings, is not at all unusual. For example, the Pólya tree method of (1) relies upon specifying the prior precision parameter $c$. The authors of that article state that values of $c$ between $1$ and $10$ work well in practice, but they also recommend checking the sensitivity of their Bayes factor to choice of $c$. When employing our CVBF methodology the training set sizes may be regarded as tuning parameters, and as with any Bayes procedure it is recommended that one investigate sensitivity of $CVBF$ to different choices for $(r, s)$. As mentioned previously, if all the Bayes factors are in basic agreement, then choice of $(r, s)$ becomes moot.

### 3.6.2 A permutation-based method

Here we propose a more objective method of choosing $r$. This method is analogous to choosing the rejection region of a frequentist test to achieve a particular size. The idea is to choose $r$ in such a way that desirable behavior of $CVBF$ is more or less guaranteed when it is assumed that $f \equiv g$. This is achieved by using the notion of a permutation test.

Let $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})$ be the set of all $m + n$ observations, and suppose that a random sample, $\boldsymbol{X}^*$, of size $m$ is chosen from $\boldsymbol{Z}$ without replacement. Let $\boldsymbol{Y}^*$ be the remaining $n$ observations in $\boldsymbol{Z}$ that were not chosen. Given a set $R$ of values of $r$, we may compute $CVBF^*(r)$ from $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ for each $r \in R$. (We recommend that the largest value in $R$ not exceed $m/2$.) If $N$ splits of the original data $(\boldsymbol{X}, \boldsymbol{Y})$ are to be used, then one should also consider $N$ splits of $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$, leading to an average log-Bayes factor at each $r \in R$. This whole procedure may be repeated $M$ times and at each $r$ the $M$ (average) log-Bayes factors plotted. A good choice for $r$ would be one for which all $M$ log-Bayes factors lie below a threshold that is considered strong evidence in favor of $f \equiv g$. There may be a number of $r$-values that produce the desired behavior, but we have found that the largest such $r$ is usually a good choice, since this makes it more likely that the training sets will produce good enough kernel estimates to conclude that $f$ and $g$ are different, if in fact that is the case.

### 3.7 Simulations

Results in Section 3.5 suggest that the way in which bandwidths are selected in our methodology is important. We use a version of cross-validation in which the data are split into two parts, KDEs are calculated from one part, and then the bandwidth of the KDE is chosen by maximizing a likelihood calculated from the second part of the data. This version of likelihood cross-validation has been studied by (32). (25) studies the leave-one-out version of cross-validation. In simulation results described in our Supplementary Material we provide evidence that, when using the kernel $K_0$, our cross-validation method of selecting a bandwidth tends to be somewhat more efficient than the leave-one-out version. It is also demonstrated how poorly likelihood cross-validation behaves when the data are long-tailed and a Gaussian kernel is used.

We turn now to simulations investigating various aspects of our CVBF methodology. Part of this investigation addresses how our test fares in comparison to the Kolmogorov-Smirnov (KS) test and to the Pólya tree test of (1). For each case where the Pólya tree test is run, the precision parameter $c$ is taken to be 1. We do this for two reasons. Primarily, this choice proved to be successful in the study of (1). Secondly, choosing $c$ to be closer to 0 seems to have the effect of making the test depend less on the centering distribution utilized and more on the empirical cdf, which is what would be desired in the non-parametric setting (19).

For the null case, we generate data from a standard normal distribution, taking $m = n$ for sample sizes 200, 400 and 800. For the CVBF training set sizes we took $r = s$, with $r = 50$, $75$ and $112$ at sample sizes $200$, $400$, and $800$, respectively. So, the training set size increases by fifty percent when the sample size doubles. The value of CVBF for a given replication was the geometric mean of CVBFs corresponding to 30 pairs of randomly selected training sets, and 1500 replications were performed at each $n$. The kernel $K_0$ was used for all our simulations, and the prior for each bandwidth was (A.5) with $\gamma$ equal to the maximizer of the corresponding likelihood.

The results are shown in Figure 3.2. All but two of the 4500 values of $\log(CVBF)$ computed were smaller than 0. At $n = 400$, all 1500 replications produced a value of $\log(CVBF)$ that was smaller than 0, and just two values larger than $-\log(20)$, a value considered to be the threshold for

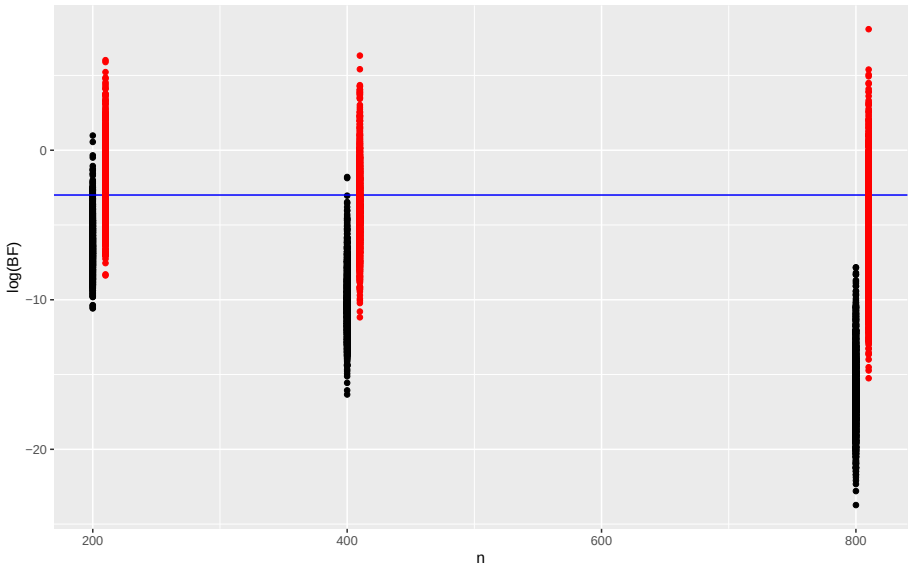



Figure 3.2: *Values of* $\log(BF)$ *for the Pólya tree and* $\log(CVBF)$ *when the null hypothesis is true.* The red and black values correspond to Pólya tree and CVBF, respectively. Each point corresponds to $X$ and $Y$ samples each of size $n$ from a $N(0,1)$ distribution. The standard deviations of the Pólya tree log-Bayes factors are $2.05$, $2.52$ and $3.31$ for $n = 200$, $400$, and $800$, respectively. The standard deviations of the $\log(CVBF)$ values are $1.60$, $1.95$ and $2.41$ for $n = 200$, $400$, and $800$, respectively.

"strong" evidence in favor of the null hypothesis (33). At $n = 800$ all 1500 values of $\log(CVBF)$ were smaller than $-\log(20)$, and at $n = 200$ all but 46 values of $\log(CVBF)$ were below this threshold. The near linear decrease in the estimates of $E(\log(CVBF))$ is evidence for the exponential rate of convergence of $CVBF$ that was discussed in Remark R6. The Pólya tree Bayes factors do not behave as well as the CVBFs. For example, at $n = 400$, the median $\log(CVBF)$ is $-10.26$, while the median Pólya tree $\log(BF)$ is but $-4.06$. At $n = 800$, 5% of the Pólya tree $\log(BF)$ values are actually larger than 0, while the *largest* $\log(CVBF)$ is $-7.83$. When a particular model is true, we desire that a Bayes factor provide the strongest possible evidence in favor of that model, and on this score CVBF has outperformed the Pólya tree method in this example.

Under the alternative hypothesis, we use a version of BayesSim, as proposed by (34), for data generation. Here, the $X$ sample is drawn from a density $f$. To obtain the $Y$ sample, we first draw $p$ from beta$(1/2, 1/2)$, a beta distribution with both parameters equal to 1/2, and then the $Y$ sample is

drawn from a mixture of the form $(1-p)f(x)+pg(x)$, where $g$ is different from $f$. This approach allows one to infer the behavior of CVBF for mixing proportions $p$ ranging from 0 to 1, where the discrepancy between the $X$ and $Y$ densities increases with $p$. Sample sizes of $m = n = 280$ were considered, the training set sizes were selected to be 120, and 500 values of $p$ were selected for each choice of $(f, g)$. For a given replication, thirty random splits for each of $X$ and $Y$ were used. For each pair of data sets the data were centered and scaled before applying the Pólya tree test. The sample median of the combination of the two data sets was subtracted from every value and then this difference was divided by IQR$/1.35$, where IQR is the interquartile range of the combined data.

In performing the Pólya tree test, specification of a precision parameter and a base distribution are required. (19) recommends centering and scaling the data and using a standard normal distribution as the base distribution. However, doing so turns out not to be efficient when the underlying distribution has sufficiently heavy tails. In the location shift case, for example, the Cauchy density turns out to be far better suited for the base distribution than the normal density since the original density is itself Cauchy. (1) provide a data-driven procedure for choosing a base distribution, but do not show that the resulting Bayes factor is consistent under the alternative. For this reason, as well as the fact that the conditional procedure is more computationally intensive, we computed Pólya tree Bayes factors for both normal and Cauchy base distributions in each simulation setting.

The described simulations were conducted in four different settings in each of which $f$ and $g$ differ in a particular way:

*Scale change:* The densities $f$ and $g$ are $\phi$ (standard normal) and $\phi(x/2)/2$, respectively, and hence differ with respect to scale.

*Location shift:* The densities $f$ and $g$ are standard Cauchy, $f_C$, and $f_C(x+1)$, respectively, and so differ with respect to location.

*Distributions with different tail behavior:* Here $f$ and $g$ are $f_C$ and $0.6745\phi(0.6745x)$, respectively. Given $p$, the mixture density in this case has the same median and interquartile

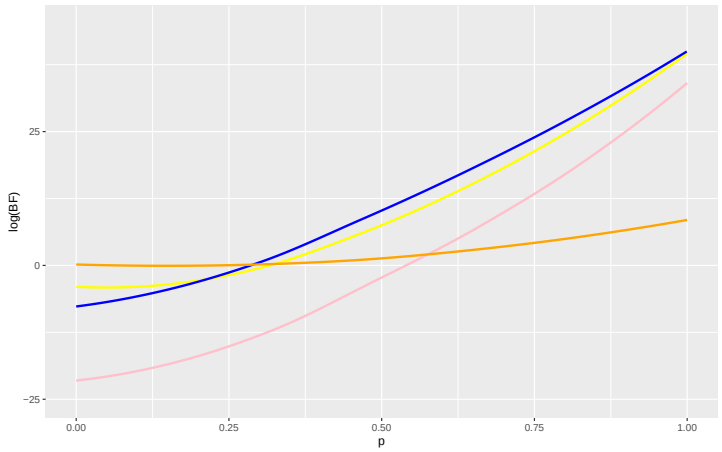



Figure 3.3: *Smoothed curves that show the values of* $\log(BF)$ *for the Pólya tree, Cross validation Bayes factors, and KS test B-values for the scale shift case.*The pink and yellow curves correspond to Pólya tree Bayes factors when standard Cauchy and standard normal are used for quantiles, respectively. The blue and orange curves correspond to CVBF and the K-S test, respectively. In the case of the K-S test, we used the calibration idea of (2). Define the quantity $B$ by $B^{-1} = -eP\log(P)$ for $P < 1/e$ and $B^{-1} = 1$ for $P \geq 1/e$, where $P$ is the $P$-value of the K-S test. (2) show that $B$ is an upper bound for a Bayes factor when the distribution of $P$ under the alternative hypothesis is a certain class of beta distributions. The orange curve is a loess smooth of all the $\log B$ values.

range as the standard Cauchy, and so the densities of the $X$ and $Y$ samples are different but have the same location and scale.

*Different distributions with same finite support:* The densities $f$ and $g$ are $U(0,1)$ (uniform on the interval $(0,1)$) and beta$(1/2,1/2)$, respectively.

For each method, we provide a loess smooth of log-Bayes factors as a function of $p$ (Figures 4-7). Plots including the data points themselves can be found in our Supplementary Material.

The following remarks are in order concerning the simulations under alternatives. To facilitate the discussion, we refer to the Pólya tree methodology based on normal and Cauchy base distributions as PN and PC, respectively.

- In general, the average Pólya tree log-Bayes factor tends to increase as the mixing parameter increases, but, depending on the base distribution, it does not rise above 0 until the mixing

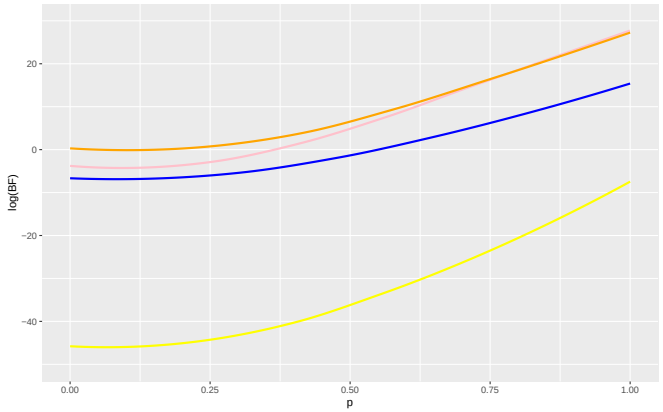



Figure 3.4: *Smoothed curves that show the values of* $\log(BF)$ *for the Pólya tree, Cross validation Bayes factors, and KS test B-values for the location shift case.* See Figure 3.3 for a legend.

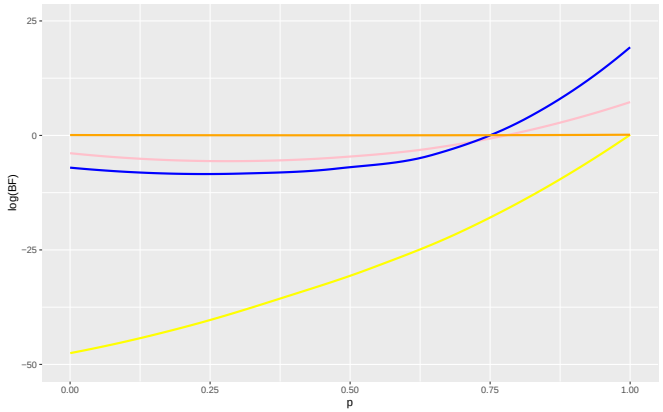



Figure 3.5: *Smoothed curves that show the values of* $\log(BF)$ *for the Pólya tree, Cross validation Bayes factors, and KS test B-values for the tail difference case.* See Figure 3.3 for a legend.

| Setting | Pólya tree: Normal base distribution | Pólya tree: Cauchy base distribution | CVBF |
|---|---|---|---|
| Scale change | 3.87 | 4.25 | 3.39 |
| Location shift | 5.49 | 4.56 | 4.21 |
| Different tail behavior | 4.31 | 2.89 | 3.56 |
| Finite support | 5.78 | 6.48 | 4.45 |

Table 3.2: *Estimated standard deviations of log-Bayes factors.*

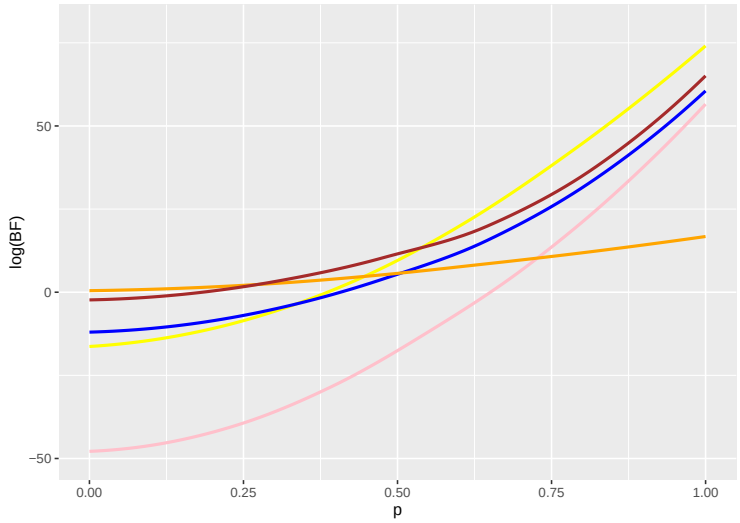



Figure 3.6: *Smoothed curves that show the values of* $\log(BF)$ *for the Pólya tree, Cross validation Bayes factors, and KS test B-values for the finite support case.* The red line gives the smoothed values for data-reflected versions of cross-validation Bayes factors [see text for more detail]. See Figure 3.3 for a legend of the rest of the curves.

parameter is relatively large. This entails that a frequentist strategy would sometimes be needed to ensure good power for a test based on the Pólya tree methodology.

- The behavior of the Pólya tree Bayes factors definitely depends on the base distribution used. Worse yet, the PN Bayes factors performed very poorly when at least one of $f$ and $g$ was Cauchy. In contrast, the performance of CVBFs based on the kernel $K_0$ was always comparable to or better than that of both PN and PC.

- The PC Bayes factors performed reasonably well in all four settings, suggesting that the Cauchy might be a good default choice of base distribution. However, the performance of PC in the finite support setting was not nearly as good as that of PN and CVBF. Also, PC Bayes factors were usually more variable than the PN Bayes factors, suggesting that PC may be less powerful in a frequentist sense than PN.

- Taken together, the last two remarks suggest that $K_0$ is at least a very good candidate for default kernel choice in the CVBF methodology, whereas identifying a good default base

distribution in the Pólya tree methodology is more of an open question.

- In the cases where $f$ and $g$ differ with respect to scale and tail behavior (Figures 3.3 and 3.5), the performance of CVBF and at least one of PC and PN was clearly better than that of the KS test.

- Table 3.2 provides estimated standard deviations for the log-Bayes factors (assuming homoscedasticity over the mixing parameter $p$). Each standard deviation is the square root of the following nonparametric variance estimate: $\sum_{i=2}^{500}(b_i - b_{i-1})^2/1000$, where $b_i$ is the log-Bayes factor at $p_{(i)}$, $i = 1, \ldots, 500$, and $p_{(1)} < p_{(2)} < \cdots < p_{(500)}$ denote the ordered values of the randomly selected mixing parameters. In most cases, the variability of the log-CVBF values is smaller than that of the Pólya tree log-Bayes factors. Importantly, the smaller variability of CVBF is understated as only thirty splits of each data set were used. Recall that in the null case the standard deviations of log-CVBF were about 3/4 of the standard deviations of the Pólya tree log-Bayes factors. So, in addition to often providing more evidence in favor of the correct hypothesis, CVBF appears, in most cases, to be more stable than the Pólya tree method.

- In the finite support case (Figure 3.6), there are two sets of CVBF results. One set is obtained as in the other three cases, and the other set uses methodology that adjusts kernel estimates for boundary effects. Kernel estimates are known to have large bias near a boundary when the density is positive at the boundary. To deal with the boundary bias we used a data reflection technique. First, we applied the $-\log$ transformation to each of the training data values (which yields exponential data when the underlying distribution is $U(0, 1)$). We then reflected these transformed data across the $y$-axis, and constructed a kernel density estimate from a combination of the original and reflected data. Doing so improved the behavior of the log-Bayes factors remarkably. In general this illustrates another point: methods of improving the kernel density estimate may be applied, and doing so can positively impact the log Bayes factors. While this also seems to be the case when choosing which base distribution should

be used for the Pólya tree test, we assert that there is more literature on modification of kernel density estimates than for fine-tuning Pólya tree base distributions. See, for example, (35), (36), and (37).

### 3.8 Data analysis

We now apply our method to the Higgs boson data set that is available from the UCL Machine Learning repository. The original data set is quite large. It has 29 columns and 11 million rows. The first column is a 0-1 variable indicating whether the data are noise or signal, and the rest of the columns are variables used for distinguishing between noise and signal. The 2nd to 22nd columns consist of predictors, while the 23rd to 29th columns are functions of columns 2 to 22 that are typically used for classification. We will illustrate our methodology by applying it to the data in columns 23 and 29.

Figure 3.7 provides KDEs for the signal and noise data in column 29 and column 23. These estimates use all 11,000,000 rows of the data set. Since the two estimates are quite different one would hope that application of our methodology to even "moderate" sized samples from the two groups would support the hypothesis of unequal densities. To investigate this question, we randomly selected 20,000 rows of the column 29 and column 23 data to perform our tests. This resulted in $m = 9543$ and $n = 10,457$ noise and signal observations, respectively. Training set sizes $r = s = 1000, 2000, 3000, 4000, 5000$ were considered, and $CVBF$ was computed for 20 different random data splits at each $r$. The resulting values are also provided in Figure 3.8. Regardless of the training set size, the evidence in favor of a difference between signal and noise for distributions in column 29 is overwhelming. Interestingly, the results are in agreement with expression (A.7), which suggests that when the alternative is true, the weight of evidence in favor of the alternative tends to decrease with an increase in training set sizes. These results are consistent with those from the Pólya tree and KS tests. The log-Bayes factor from the Pólya tree test for column 29 was 234.7242, while the $P$-value from the KS test was essentially 0. We verified that the training sizes were reasonable by utilizing the permutation method described in Section 5.2. Discussion of these results for both data columns is found in Section 6 of the Supplementary

Material.

For column 23, the difference between the two estimates is extremely small, and so it would not be surprising if Bayes factors based on a small subset of the data support the hypothesis of equal densities. The log-Bayes factors computed in this case were smaller than $-15$. This figure shows that the average of log-$CVBF$ increases as the training set size increases. Based on expression (A.6), this agrees with what we expect under the null hypothesis of equal distributions. We also considered the use of the Pólya tree and K-S tests on the column 23 data. These methods reach the same conclusion as our procedure. The log-Bayes factor of the Pólya tree test was -263.6514, which is in strong favor of the null hypothesis. The $P$-value and $B$ for the K-S test are $0.1115$ and $1.504$, respectively, neither of which is reason to reject the null hypothesis. (See Section 6 for the definition of $B$.)

Although our focus has been on testing, it is of some interest to see how our cross-validatory methodology compares with Pólya trees in *estimating* the underlying densities. To this end we compute posterior predictive densities for the column 23 noise data using both our methodology and Pólya trees. Denote the first 9543 values of the column 23 noise data by $x_1, \ldots, x_{9543}$. In regard to our cross-validation method, the posterior distribution of the bandwidth is

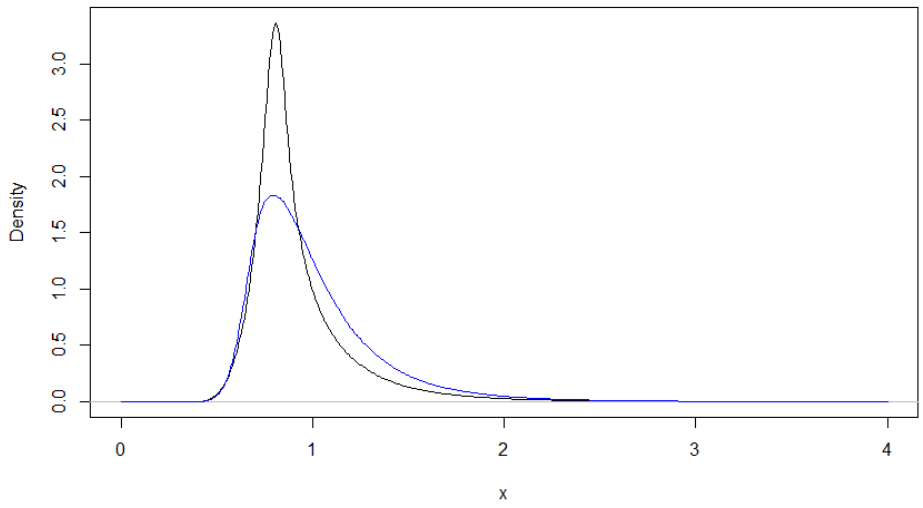
$$\pi(h|\boldsymbol{x}_V) \propto \pi(h) \prod_{i=5001}^{9543} \hat{f}(x_i|h, \boldsymbol{x}_T),$$

where $\boldsymbol{x}_T = (x_1, \ldots, x_{5000})$ are the training data and $\boldsymbol{x}_V = (x_{5001}, \ldots, x_{9543})$ the validation data. We drew 250 values, $h_1, \ldots, h_{250}$, from $\pi(\cdot|\boldsymbol{x}_V)$ using an independence-sampler version of Metropolis-Hastings with a normal proposal distribution that was a close match to the posterior. Our approximation $p_{\text{pred}}$ of the posterior predictive density was

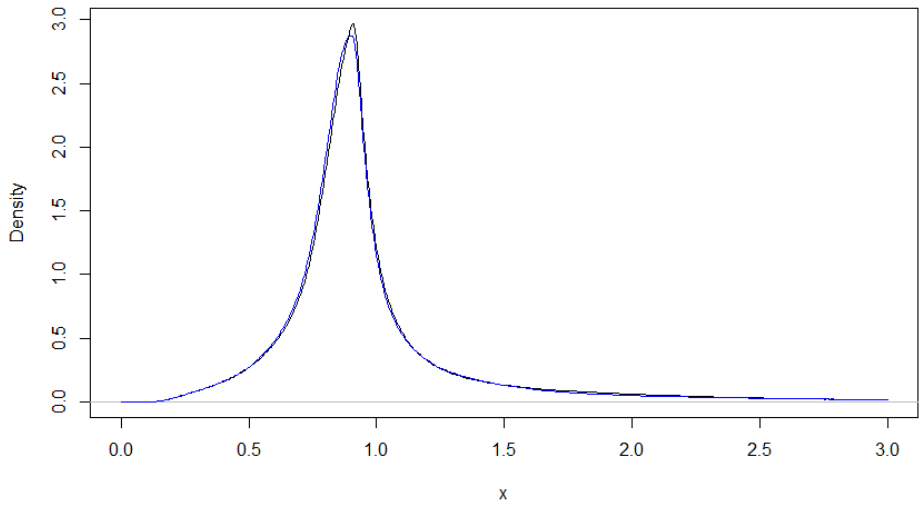
$$p_{\text{pred}}(x) = \frac{1}{250} \sum_{i=1}^{250} \hat{f}(x|h_i, \boldsymbol{x}_T),$$

which is plotted in Figure 3.9 along with the Pólya tree posterior predictive density.

The Pólya tree method is well known for producing spurious modes in the posterior predictive

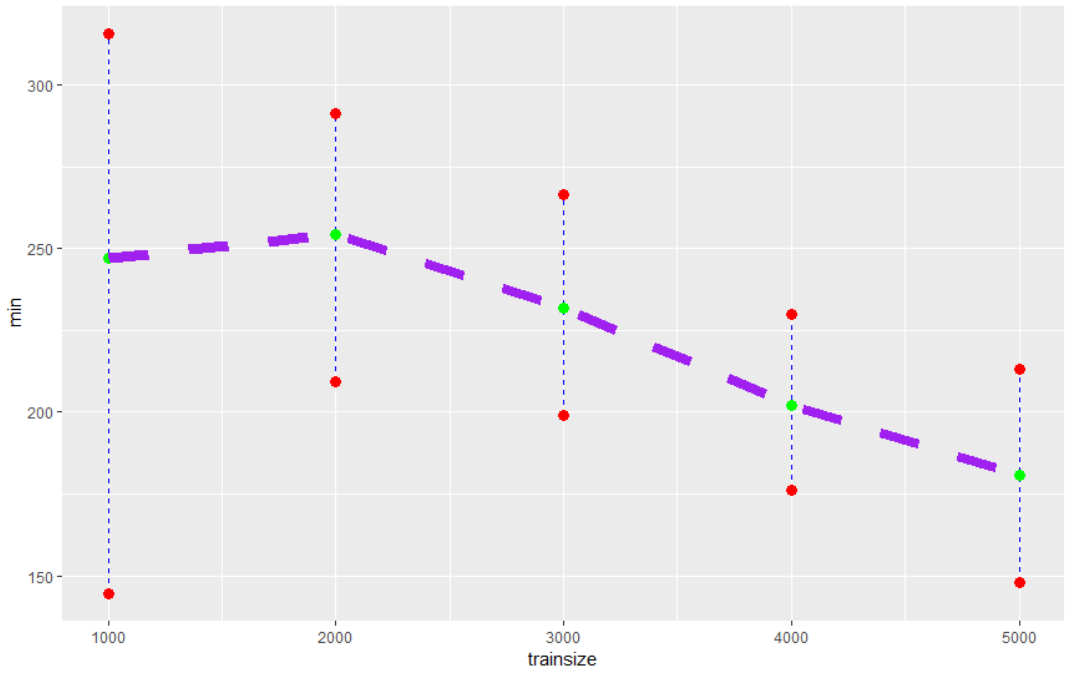



(a) Kernel density estimates for column 29 of the Higgs boson data. The blue curve is for the noise data and the black for signal.

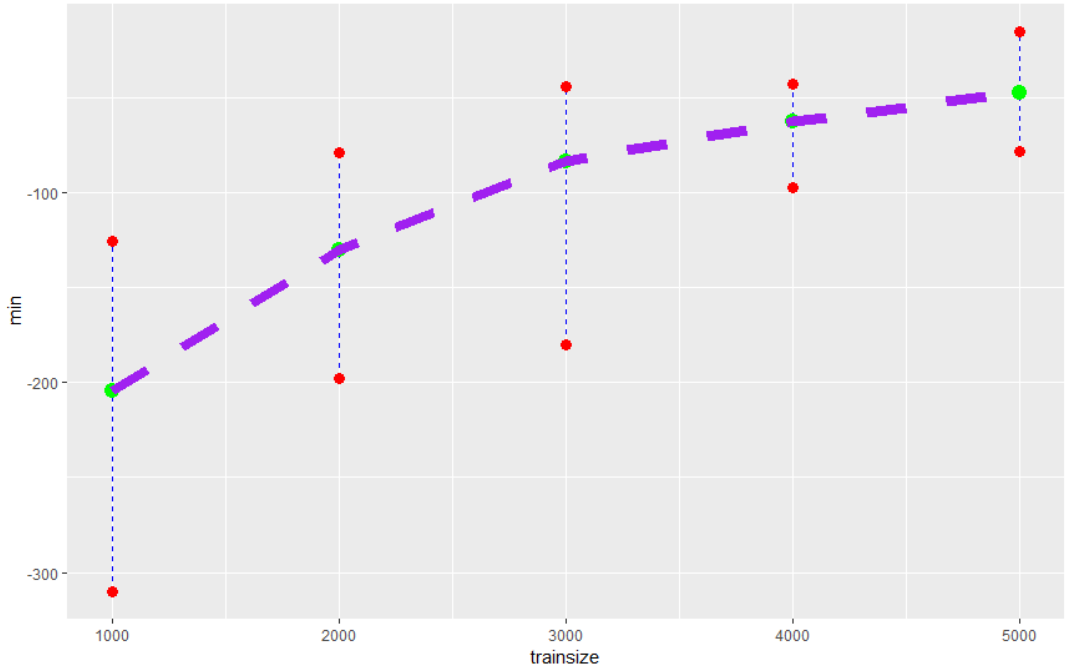



(b) Kernel density estimates for column 23 of the Higgs boson data. The blue curve is for the noise data and the black for signal.

Figure 3.7: Kernel density estimates of different columns of the Higgs Boson data




(a) Values of log-CVBF computed from column 29 of the Higgs boson data. The lines connect the averages of log-CVBF at different training set sizes.




(b) Values of log-CVBF computed from column 23 of the Higgs boson data. The lines connect the averages of log-CVBF at different training set sizes.

Figure 3.8: Performance of CVBF on different columns of the Higgs Boson data

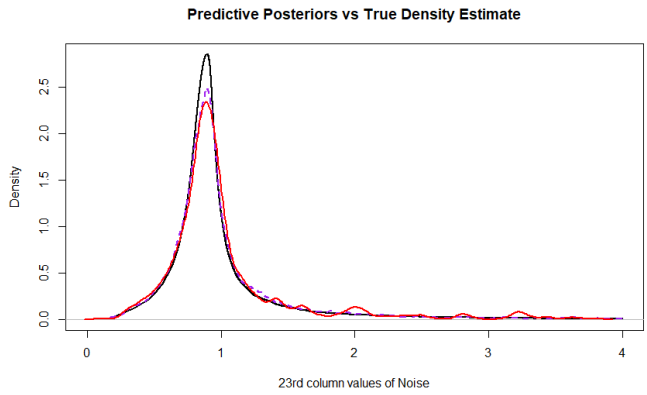



Figure 3.9: *Plots of posterior predictive densities for the column 23 Higgs boson noise data.* The black line is a KDE based on all 5,170,877 noise data. Because of the size of the data set, we regard this KDE as the truth. The red curve is the posterior predictive density corresponding to the Pólya tree method, and in purple is a cross-validation posterior predictive density.

density (19). This is evident in Figure 3.9 in the right tail of the Pólya tree density where the data are relatively sparse. The cross-validation density does not seem to be subject to this problem, at least not to the same degree. In any event, the cross-validatory method has produced at least as good an estimate of the underlying density without the necessity of a complex prior distribution. This illustrates a basic tenet of this paper: one may devise a good Bayesian nonparametric procedure that does not depend on a large number of parameters and the attendant prior specification.

## 3.9 Discussion

We have proposed a non-parametric, Bayesian two-sample test for checking equality of distributions. The methodology uses CVBFs, defining kernel density estimate models from training data and then calculating a Bayes factor from validation data. We advocate that a CVBF be used in genuine Bayesian fashion and interpreted as the relative odds of the two hypotheses. This is in contrast to the proposal of (1), who evaluate a Pólya tree Bayes factor in frequentist fashion using a permutation test. We argue the CVBF is Bayes consistent under both hypotheses, and under the null hypothesis it converges in probability to 0 at an exceptionally fast rate. We provide a supplementary R package that calculates CVBFs and the Pólya tree Bayes factor of (1), assuming that a

base distribution is supplied for the latter method.

The success of CVBFs stems in large part from how well the kernel density estimates estimate the underlying densities. Our simulation results show that ways of improving KDE performance, such as adjusting for boundary effects, can improve the behavior of CVBF. On the other hand, evidence associated with poor KDEs may not accurately reflect the relative evidence of $f \equiv g$ and $f \not\equiv g$. While one will not always be certain of when the KDEs are inadequate. A useful diagnostic is to examine $CVBF$ for a number of data splits. When the log-Bayes factors are highly volatile and yield differing conclusions, the procedure may be regarded as inconclusive. Our experience suggests this occurs due to poor kernel estimates resulting from poor bandwidth choices and using a heavy-tailed kernel, such as $K_0$, fixes the problem.

In particular, CVBF computations do not scale as well with the size of the data set as do those of the Pólya tree procedure, and so $CVBF$ can be slower than calculating the Pólya tree Bayes factor. This is because maximizing (with respect to bandwidth) the likelihood of kernel density estimates can be time consuming. A future research problem involves speeding up the test by utilizing techniques that speed bandwidth selection.

Finally, the idea of CVBF can be generalized in a fairly straightforward fashion to deal with other inference problems, including comparison of multivariate densities, comparison of more than two densities and comparison of regression functions.

# 4. A BAYESIAN MOTIVATED TWO-SAMPLE TEST BASED ON KERNEL DENSITY ESTIMATES

## 4.1 Abstract

A new nonparametric test of equality of two densities is investigated. The test statistic is an average of log-Bayes factors, each of which is constructed from a kernel density estimate. Critical values are determined by a permutation distribution, conditional on the data. An attractive property of the methodology is that a critical value of 0 leads to a test with type I error probability tending to 0 as sample sizes tend to $\infty$. The test is proven to be consistent in a frequentist sense, and its characteristics are studied via simulation. Extensions to multivariate data are straightforward, as illustrated by an application to bivariate connectionist data.

## 4.2 Introduction

(38) proposed the use of cross-validation Bayes factors in the classic two-sample problem of comparing two distributions. Their basic idea is to randomly divide the data into two distinct parts, call them $A$ and $B$, and to define two models based on kernel density estimates from part $A$. One model assumes that the two distributions are the same and the other allows them to be different. A Bayes factor comparing the two part $A$ models is then defined from the part $B$ data. In order to stabilize the Bayes factor, (38) suggest that a number of different random data splits be used, and the resulting log-Bayes factors averaged.

In the current paper we consider a special case of this approach in which the part $A$ data consists of all the available observations save one. If the sample sizes of the two data sets are $m$ and $n$, this entails that a total of $m + n$ log-Bayes factors may be calculated. The average of these $m + n$ quantities becomes the test statistic here considered, and is termed $ALB$.

Although $ALB$ is an average of log-Bayes factors, it does not lead to a consistent Bayes test because each of the log-Bayes factors is based on just a single observation. (38) suppose that the validation set size grows to $\infty$, while in our case it remains of size 1. This results in the

$ALB$ converging to the Kullback-Leibler divergence of the two densities, not $\infty$ like in the case of (38). We therefore use frequentist ideas to construct our test. The exact null distribution of $ALB$ conditional on order statistics is obtained using permutations of the data. Doing so leads to a consistent frequentist test whose size is controlled exactly. The problem of bandwidth selection is dealt with by using leave-one-out likelihood cross-validation applied to the combination of the two data sets. This method is computationally efficient in that the resulting bandwidth is invariant to permutations of the combined data, and therefore has to be computed just once. Our methodology is easily extended to bivariate data, and we do so in a real data example.

(39) also use a permutation test based on kernel estimates for the two-sample problem, their statistic being based on an $L_2$ distance. (40) shows how other distances and divergences compare when applying them to the general $k$-sample problem, restricting his comparisons to the one-dimensional case. Our method mainly differs from these procedures by virtue of its Bayesian motivation. Existing methodology that most closely resembles ours is that of (41), who use a kernel-based marginal likelihood ratio to test goodness of fit of parametric models for a distribution. Their marginal likelihood employs a prior for a bandwidth, as does ours.

### 4.3 Methodology

We assume that $\mathbf{X} = (X_1, \ldots, X_m)$ are independent and identically distributed (i.i.d.) from density $f$, and independently $\mathbf{Y} = (Y_1, \ldots, Y_n)$ are i.i.d. from density $g$. We are interested in the problem of testing the null hypothesis that $f$ and $g$ are identical on the basis of the data $\mathbf{X}$ and $\mathbf{Y}$. Let $\mathbf{U} = (U_1, \ldots, U_k)$ be an arbitrary set of $k$ scalar observations, and define a kernel density estimate by

$$\hat{f}_K(u|h, \mathbf{U}) = \frac{1}{kh} \sum_{i=1}^{k} K\left(\frac{u - U_i}{h}\right), \quad -\infty < u < \infty,$$

where $K$ is the kernel and $h > 0$ the bandwidth.

#### 4.3.1 The test statistic

Let $Z_i = X_i$, $i = 1, \ldots, m$, $Z_i = Y_{i-m}$, $i = m+1, \ldots, m+n$, $\mathbf{Z} = (Z_1, \ldots, Z_{m+n})$ and $\mathbf{Z}_i$ be the vector $\mathbf{Z}$ with all its components except $Z_i$, $i = 1, \ldots, m+n$. Also, let $\mathbf{X}_i$ be all the components

of $\mathbf{X}$ except $X_i$, $i = 1, \ldots, m$, and $\mathbf{Y}_j$ all the components of $\mathbf{Y}$ except $Y_j$, $j = 1, \ldots, n$. If we assume that $f$ is identical to $g$, then potential models for $f$ are $M_{0i} = \{\hat{f}_K(\cdot\,|h, \mathbf{Z}_i) : h > 0\}$, $i = 1, \ldots, m+n$. Suppose that $1 \leq i \leq m$. If we allow that $f$ and $g$ are different, then a model for the datum $Z_i$ is $M_{1i} = \{\hat{f}_K(\cdot\,|a, \mathbf{X}_i) : a > 0\}$. In this case a legitimate Bayes factor for comparing $M_{0i}$ and $M_{1i}$ on the basis of the datum $Z_i$ has the form

$$B_i = \frac{\int_0^\infty \pi(a)\hat{f}_K(Z_i|a, \mathbf{X}_i)\, da}{\int_0^\infty \pi(h)\hat{f}_K(Z_i|h, \mathbf{Z}_i)dh}, \quad i = 1, \ldots, m,$$

where, mainly for convenience, we have assumed that the bandwidth priors are the same in all cases. Likewise, if $i = m+1, \ldots, m+n$, then $M_{1i} = \{\hat{f}_K(\cdot\,|b, \mathbf{Y}_{i-m}) : b > 0\}$ is a model for the datum $Z_i$, and a Bayes factor for comparing $M_{0i}$ and $M_{1i}$ is

$$B_i = \frac{\int_0^\infty \pi(a)\hat{f}_K(Z_i|a, \mathbf{Y}_{i-m})\, da}{\int_0^\infty \pi(h)\hat{f}_K(Z_i|h, \mathbf{Z}_i)dh}, \quad i = m+1, \ldots, m+n.$$

When $m$ and $n$ are large, it is expected that each of $M_{1i}$, $i = 1, \ldots, m+n$, will be a good model for either $f$ or $g$. Likewise, each of $M_{0i}$ will be a good model for the common density on the assumption that $f$ and $g$ are identical. However, none of $B_1, \ldots, B_{m+n}$ will be Bayes factors that can provide convincing evidence for either hypothesis simply because each one uses likelihoods based on a single datum. At first blush one might think that a solution to this problem is to take the average of the $m+n$ log-Bayes factors:

$$ALB = \frac{1}{(m+n)} \sum_{i=1}^{m+n} \log B_i. \tag{4.1}$$

However, this results in a statistic that will consistently estimate 0 or a positive constant in the respective cases $f \equiv g$ or $f \not\equiv g$. In neither case does the statistic have the property of Bayes consistency, i.e., the property that the Bayes factor tends to 0 and $\infty$ when $f \equiv g$ and $f \not\equiv g$, respectively.

The discussion immediately above points out a fundamental fact that seems not to have been

widely discussed: combining a large number of inconsistent Bayes factors does not necessarily lead to a consistent Bayes factor. A guiding principle in (38) was that of averaging log-Bayes factors from different random splits of the data with the aim of producing a more stable log-Bayes factor. However, in order for this practice to yield a *consistent* Bayes factor, it is important that each of the log-Bayes factors being averaged is consistent. And to ensure this consistency, it is necessary that the sizes of both the training and validation sets tend to $\infty$ with the samples sizes $m$ and $n$. Obviously this is not the case when the size of each validation set is just 1, as in the current paper.

An advantage of the approach proposed herein is that the practitioner does not have to choose the size of the training sets. The cost is that the resulting statistic does not have the property of Bayes consistency. We thus propose that the statistic be used in frequentist fashion. An appealing way of doing so is to use a permutation test, which (save for certain practical issues to be discussed) leads to a test with exact type I error probability for all $m > 1$ and $n > 1$. Let $Z_{(1)} < Z_{(2)} < \cdots < Z_{(m+n)}$ be the order statistics for the combined sample. Let $\mathbf{j} = (j_1, \ldots, j_{m+n})$ be a random permutation of $1, \ldots, m+n$, and define $T(\mathbf{j})$ to be the statistic (4.1) when the $X$-sample is taken to be $Z_{j_1}, \ldots, Z_{j_m}$ and the $Y$-sample to be $Z_{j_{m+1}}, \ldots, Z_{j_{m+n}}$. It follows that, conditional on the order statistics $Z_{(1)}, \ldots, Z_{(m+n)}$, the $(m+n)!$ values taken on by $T(\cdot)$ are equally likely. Therefore, if $t_{m,n}$ is a $1-\alpha$ quantile of the empirical distribution of $T(\cdot)$, then the test that rejects $f \equiv g$ when $T \geq t_{m,n}$ will have an (unconditional) type I error probability of $\alpha$. As will be shown in the Appendix, $ALB$ is negative with probability tending to 1 as $m, n \to \infty$, implying that for any $\alpha > 0$ $t_{m,n}$ will be negative for $m$ and $n$ large enough. From an evidentiary standpoint, it is nonsense to reject $H_0$ for a negative value of $ALB$. We therefore suggest using the critical value $\max(0, t_{m,n})$, which ensures that the test is sensible and has level $\alpha$.

### 4.3.2 The effect of using scale family priors

Let $\pi_0$ be an arbitrary density with support $(0, \infty)$. A possible family of priors is one that contains all rescaled versions of $\pi_0$. For $b > 0$, using the prior $\pi(h) = \pi_0(h/b)/b$ and making the

change of variable $h/b = u$ in the denominator of $B_i$, we have

$$\int_0^\infty b^{-1}\pi_0(h/b)\hat{f}_K(Z_i|h,\mathbf{Z}_i)dh = \hat{f}_L(Z_i|b,\mathbf{Z}_i),$$

where the kernel $L$ is

$$L(z) = \int_0^\infty u^{-1}\pi_0(u)K(z/u)\,du, \quad \text{for all } z. \tag{4.2}$$

So, by using this type of prior, each marginal likelihood comprising $ALB$ becomes a kernel density estimate with bandwidth equal to the scale parameter of the prior. In one sense this is disappointing since it means that averaging kernel estimates with respect to a bandwidth prior does not actually sidestep the issue of choosing a smoothing parameter. One has simply traded bandwidth choice for choice of the prior's scale. However, it turns out that there is a quantifiable advantage to using a prior for the bandwidth of $K$. As we will subsequently detail, likelihood cross-validation is often more efficient when applied to $\hat{f}_L$ rather than to $\hat{f}_K$.

When using a scale family of priors, the result immediately above implies that

$$\begin{aligned}(m+n)ALB &= \sum_{i=1}^{m}\log(\hat{f}_L(X_i|b,\mathbf{X}^i)) + \sum_{j=1}^{n}\log(\hat{f}_L(Y_j|b,\mathbf{Y}^j)) \\ &\quad - \sum_{i=1}^{m+n}\log(\hat{f}_L(Z_i|b,\mathbf{Z}^i)),\end{aligned} \tag{4.3}$$

and so the proposed statistic is proportional to the log of a likelihood ratio. The two likelihoods are cross-validation likelihoods, and the numerator and denominator of the ratio correspond to the hypotheses of different and equal densities, respectively.

In practice one will be faced with choosing $b$, the bandwidth of the kdes based on kernel $L$. The denominator of $\exp((m+n)ALB)$ as a function of $b$ is the likelihood cross-validation criterion, as studied by (25), based on the combined sample. We propose using $b = \hat{b}$, the maximizer of this denominator. This bandwidth has the desirable property that it is invariant to the ordering of

the data in the combined sample. Let $ALB^*$ be the value of test statistic (4.1) for a permuted data set. One should use the principle that $ALB^*$ is the same function of the permuted data as $ALB$ is of the original data. So, in principle the bandwidth should be selected for every permuted data set, but because of the invariance of $\hat{b}$ to the ordering of the combined sample, this data-driven bandwidth equals $\hat{b}$ for every permuted data set. This results in a large computational savings relative to a procedure that selects the bandwidth differently for the $X$- and $Y$-samples. Using the same bandwidth under both null and alternative hypotheses also fits with the principle espoused by (42).

### 4.3.3 Choice of kernel

By far the most popular choice of kernel in practice is the Gaussian kernel, $K(x) = \phi(x)$, $-\infty < x < \infty$, where $\phi$ is the standard normal density. For $\nu > 0$, define

$$\pi_0(u) = \frac{2(\nu/2)^{\nu/2}}{\Gamma(\nu/2)} u^{-(\nu+1)} \exp\left(-\frac{\nu}{2u^2}\right), \quad u > 0. \tag{4.4}$$

If one takes $K$ to be the the standard normal kernel and uses prior (4.4), then the corresponding kernel $L$ is a $t$-density with $\nu$ degrees of freedom. An interesting aspect of these kernels is that they have heavier tails than those of the Gaussian kernel. This is especially true for the more diffuse, or noninformative priors, i.e., those for which $\nu$ is small.[1]

The fact that the kernel $L$ is more heavy-tailed than $K$ in the previous example is not an isolated phenomenon, as indicated by the following proposition (which is straightforward to prove): If $\pi_0$ has support $(0, C)$ with $1 < C \leq \infty$ and the tails of $K$ decay exponentially, then the tails of $L$ are heavier than those of $K$ in that $K(u)/L(u) \to 0$ as $u \to \infty$.

(25) established results that imply that the previous result can be quite beneficial. He showed that kernels must be relatively heavy-tailed in order for them to perform well with respect to likelihood cross-validation. In particular, he shows that likelihood cross-validation fails miserably as a method

[1]The mean and variance of (4.4) exist for $\nu > 2$. At $\nu = 3$, the two are 1.382 and 1.090, respectively, and as $\nu \to \infty$ they converge to 1 and 0.

for choosing the bandwidth of a kde based on a Gaussian kernel. The tails of the kernel must be considerably heavier than those of a Gaussian density in order for likelihood cross-validation to be effective. So, Proposition 4.3.3 shows that the Bayesian notion of averaging commonly used kernel estimates with respect to a prior brings the resulting estimate more in line with the conditions of (25). This can have a substantial benefit for our statistic inasmuch as we are using a likelihood cross-validation bandwidth in its construction.

In principle, many different choices of $\pi_0$ and $K$ could produce the same kernel $L$. Or, one might ask "given kernel $K$, what prior $\pi_0$ would produce a specified $L$?" When $K$ is Gaussian, the latter question is answered by solving an integral equation. Unfortunately, doing so, at least in a general sense, exceeds our mathematical abilities. In the case where $K$ is uniform, though, an elegant solution exists, as seen in the next section.

### 4.3.4 When $K$ is uniform

In the special case where $K$ is uniform on the interval $(-1/2, 1/2)$, it is easy to check that, for all $u$,

$$L(u) = \int_{2|u|}^{\infty} \alpha^{-1} \pi_0(\alpha)\, d\alpha. \tag{4.5}$$

If $\pi_0$ has support $(0, \infty)$, then $L$ has support $(-\infty, \infty)$, and hence we see again that averaging kernels with respect to a prior leads to a more heavy-tailed kernel.

Since our statistic ends up being a log-likelihood ratio based on kernel $L$, an interesting question is "what prior $\pi_0$ gives rise to a specified kernel $L$?" Taking $u \geq 0$, (4.5) implies that

$$\pi_0(2u) = -uL'(u). \tag{4.6}$$

When $L$ is decreasing on $[0, \infty)$ it follows that $\pi_0$ is a density. (Under mild tail conditions on $L$ and assuming that $L'(0+)$ exists finite, it is easy to show using integration by parts that (4.6) integrates to 1 on $(0, \infty)$.)

Consider the following kernel proposed by (25):

$$L_0(u) = \frac{1}{\sqrt{8\pi e}\,\Phi(1)} \exp\left[-\frac{1}{2}\left(\log(1+|u|)\right)^2\right].$$

Suppose that a kde is defined using kernel $L_0$ and its bandwidth is chosen by likelihood cross-validation. (25) shows that, in general, this cross-validation bandwidth will be asymptotically optimal in a Kullback-Leibler sense. *In contrast, using cross-validation to choose the bandwidth of a uniform kernel kde will produce a bandwidth that diverges to $\infty$ as the sample size tends to $\infty$.*

Using (4.6) the prior, shown in Figure 4.1, that produces $L_0$ is

$$\pi_0(2u) = L_0(u)\frac{u\log(1+u)}{1+u}.$$

This shape for the bandwidth prior could be considered canonical inasmuch as $L'$ will be similarly shaped for kernels that are decreasing on $(0,\infty)$.

### 4.3.5 Further properties of $ALB$

In the Appendix we will show that the $ALB$ test is consistent in the frequentist sense. In other words, for any alternative the power of an $ALB$ test of fixed level tends to 1 as $m$ and $n$ tend to $\infty$.

Interestingly, $ALB$ has the property of being sharply bounded above. It can be rewritten as follows:

$$\sum_{i=1}^{m} \log(\hat{f}_L(X_i|b,\mathbf{X}^i)/\hat{f}_L(X_i|b,\mathbf{Z}^i)) + \sum_{j=1}^{n} \log(\hat{f}_L(Y_j|b,\mathbf{Y}^j)/\hat{f}_L(Y_j|b,\mathbf{Z}^{m+j})).$$

Defining $p_{m,n} = (m-1)/(m+n-1)$,

$$\hat{f}_L(X_i|b,\mathbf{Z}^i) = p_{m,n}\hat{f}_L(X_i|b,\mathbf{X}^i) + (1-p_{m,n})\,\hat{f}_L(X_i|b,\mathbf{Y}), \quad i=1,\ldots,m,$$

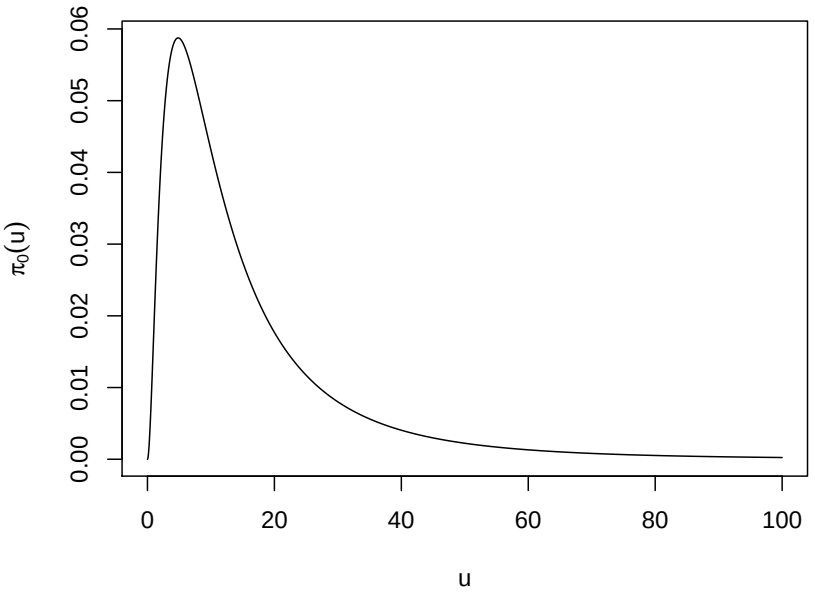



Figure 4.1: *The prior that produces the Hall kernel when $K$ is uniform.*

and therefore

$$\frac{\hat{f}_L(X_i|b,\mathbf{X}^i)}{\hat{f}_L(X_i|b,\mathbf{Z}^i)} = \frac{1}{p_{m,n}} \cdot \frac{p_{m,n}\hat{f}_L(X_i|b,\mathbf{X}^i)}{p_{m,n}\hat{f}_L(X_i|b,\mathbf{X}^i) + (1-p_{m,n})\,\hat{f}_L(X_i|b,\mathbf{Y})} \leq \frac{1}{p_{m,n}}.$$

A similar bound applies for the other component of $ALB$, implying that

$$ALB \leq -\left[\left(\frac{m}{m+n}\right)\log(p_{m,n}) + \left(1-\frac{m}{m+n}\right)\log\left(\frac{n-1}{m+n-1}\right)\right]. \tag{4.7}$$

Using the fact that $-[x\log(x)+(1-x)\log(1-x)]$ has its maximum at $x = 1/2$ when $0 \leq x \leq 1$, bound (4.7) implies that

$$ALB \leq \log(2)\cdot\max\left(\frac{m}{(m-1)}, \frac{n}{(n-1)}\right).$$

Unless one of $m$ and $n$ is very small, the effective bound on $ALB$ is $\log(2)$. This reinforces the fact

that $ALB$ does not have the property of Bayes consistency. While it is true that $ALB$ is an average of Bayes factors, none of these Bayes factors can ever provide compelling evidence in favor of the alternative. To reiterate, this problem is overcome by employing $ALB$ in frequentist fashion.

While $ALB$ can take on positive values when the null hypothesis is true, our proof of frequentist consistency shows that, under $H_0$, $P(ALB < 0) \to 1$ as $m, n \to \infty$. This implies that if 0 is used as a critical value, then the resulting test level tends to 0 as $m, n \to \infty$. So, even though $|ALB|$ does not tend to $\infty$, the *sign* of $ALB$ provides compelling evidence for the hypotheses of interest when the sample sizes are large.

The exact conditional distribution of $ALB$ is known under the null hypothesis, as we use a permutation test. Nonetheless, it is of some interest to have an impression of the *unconditional* distribution of $ALB$. To this end, we randomly select two normal mixture densities that differ. The number of components $M$ in the first mixture is between 2 and 20 and chosen from a distribution such that the probability of $m$ is proportional to $m^{-1}$, $m = 2, \ldots, 20$. Given $M = m$, mixture weights are drawn from a Dirichlet distribution with all $m$ parameters equal to $1/2$. Given $M = m$ and mixture weights, variances $\sigma_1^2, \ldots, \sigma_m^2$ of the normal components are a random sample from an inverse gamma distribution with both parameters equal to $1/2$. Finally, means $\mu_1, \ldots, \mu_m$ of the normal components are such that $\mu_1, \ldots, \mu_m$ given $\sigma_1, \ldots, \sigma_m$ are independent with $\mu_j|\sigma_j \sim N(0, \sigma_j^2)$, $j = 1, \ldots, m$. The second normal mixture is independently selected using exactly the same mechanism.

We draw a sample of size 100 from each of the two randomly generated densities (so that $m = n = 100$), and then compute $ALB$. This procedure is replicated on the same two densities 100 times. After this, we repeat the whole procedure for nine more pairs of randomly selected densities. The results are seen in Figure 4.2. Save for case 3, the proportion of positive $ALB$s is nearly 1 in all cases.

We repeated a similar procedure for the null hypothesis setting. The simulation was exactly the same except that in each of the ten cases, only one density was generated, and a pair of independent samples (of size 100 each) was selected from this same density. The resulting $ALB$ distributions

can be seen in Figure 4.3. The proportion of the cases where $ALB < 0$ for the 10 densities were, respectively, 0.89, 0.83, 0.83, 0.84, 0.85, 0.87, 0.91, 0.84, 0.84, and 0.76. These results are consistent with the fact that $P(ALB < 0)$ tends to 1 with sample size.

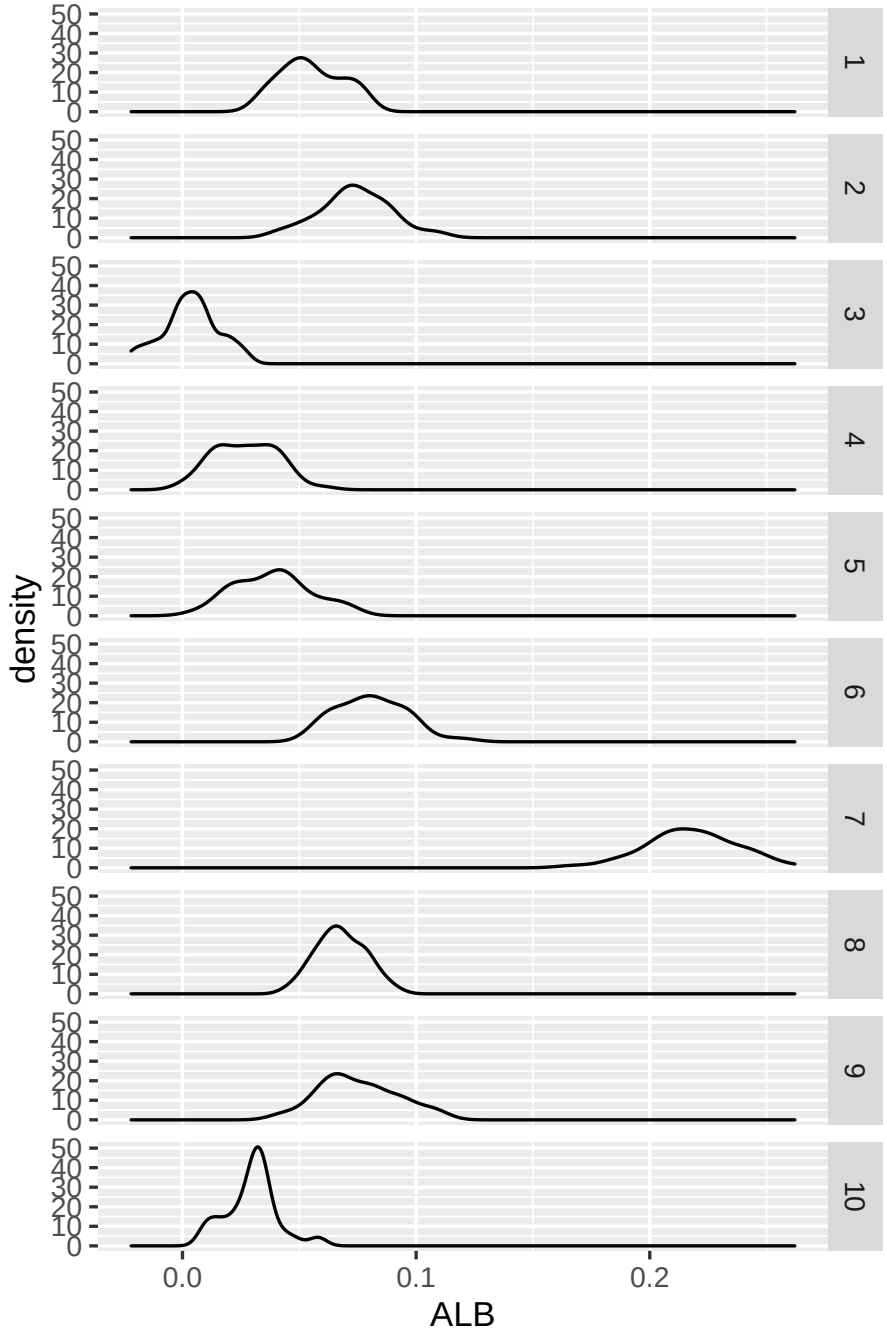



Figure 4.2: *Distribution of $ALB$ under various alternative hypotheses.*

We feel that $ALB$ has potential for screening variables in a binary classification problem. Since $ALB$ is negative with high probability under $H_0$, we feel that 0 is a nicely interpretable cutoff for variable inclusion. However, we leave this topic for future research.

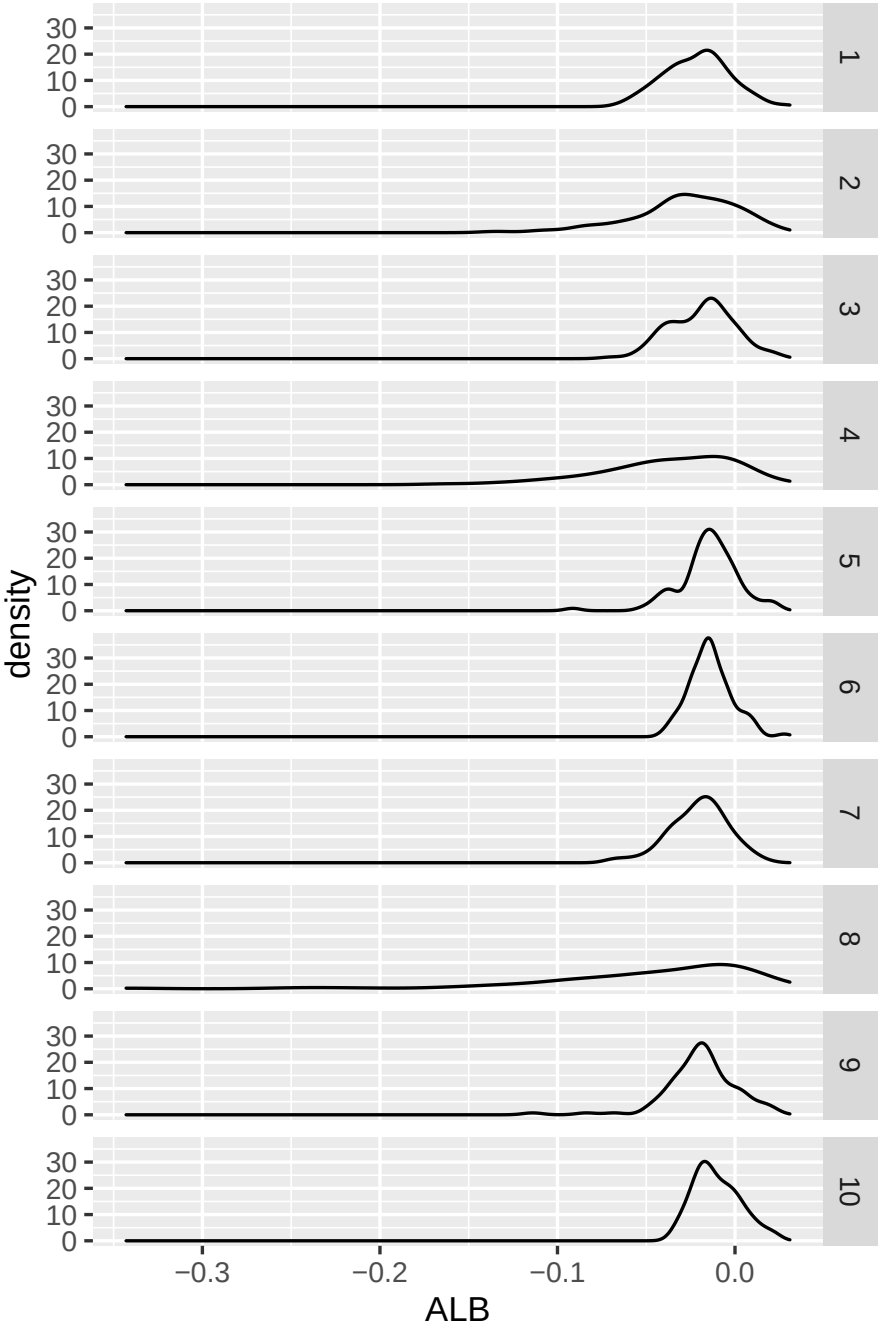



Figure 4.3: *Distribution of $ALB$ under various null hypotheses.*

## 4.4 Simulations

We perform a small simulation study to investigate the size and power of our test. The kernel $L$ is taken to be the Hall kernel, $L_0$, as defined in Section 2.4. To explore the effect of the number of permutations, we generate 500 pairs of data sets, with one data set being a random sample of size $m = 50$ from a standard normal distribution, and the other a random sample of size $n = 50$ from a normal distribution with mean 0 and standard deviation 2. For each of the 500 pairs of data sets, the 95th percentile of $ALB$s is approximated using a range of different numbers ($N$) of permutations starting at 100 and increasing by a factor of 1.5 up to 3845. Results are indicated by the boxplots in Figure 4.4. The percentiles are centered at approximately the same value for all $N$. Not surprisingly, the variability of the percentiles becomes smaller as $N$ increases. This implies a certain amount of mismatch between percentiles at $N = 3845$ and those at smaller $N$.

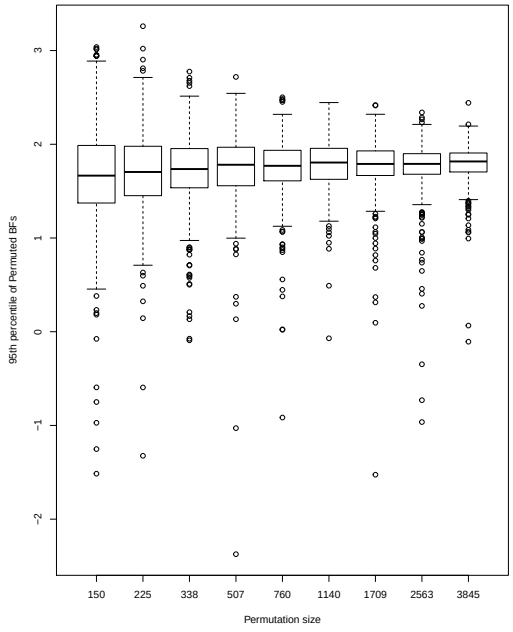



Figure 4.4: *Effect of number of permutations on the 95th percentile of permutation distributions.*

The consequence of the mismatch just alluded to can be investigated by determining the true

conditional and unconditional levels of tests based on small $N$. For the null case, two data sets, each of size 50, are generated from a common normal distribution. Since the distribution of $ALB$ is invariant to location and scale in the null case, we use a standard normal without loss of generality. For each pair of data sets, the data are randomly permuted 338 times, which leads to 338 values of $ALB$. A second set of 3845 permutations is then performed, leading to 3845 more values of $ALB$. The proportion of $ALB$s from the second set that exceed the 95th percentile of the $ALB$s formed from the first set is then determined. This proportion is approximately equal to the conditional level of the test based on 338 permutations. This same procedure is used for each of 500 data sets, and the resulting distribution of approximate levels is shown in Figure 4.5. The histogram

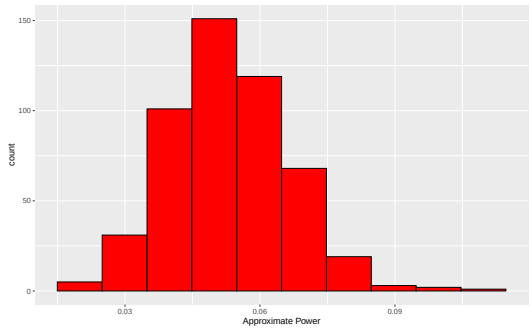



Figure 4.5: *Distribution of approximate conditional levels of permutation tests under the null hypothesis.* Each conditional level is the proportion of 3845 $ALB$s from permuted data sets that exceed the 95th percentile of $ALB$s formed from 338 permuted data sets. Results are based on 500 replications in each of which both distributions are standard normal.

is centered near 0.05, and 87% of the conditional levels are between 0.03 and 0.07. Furthermore, an approximation to the unconditional level is $\sum_{i=1}^{500} \hat{\alpha}_i/500 = 0.053$, where $\hat{\alpha}_i$ is the approximate conditional level for the $i$th data set, $i = 1, \ldots, 500$. Based on these results, use of only 338 permutations is arguably adequate.

The same experiment is repeated except now the two data sets are drawn from different distributions, a standard normal and a normal with mean 0 and standard deviation 2. Results from this experiment are given in Figure 4.6. As in the null case, the conditional levels based on the use of

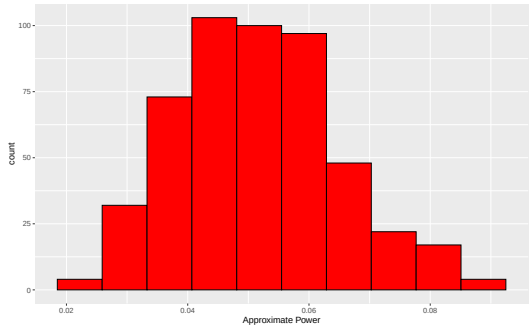



Figure 4.6: *Distribution of approximate conditional levels of permutation tests under an alternative hypothesis.* Each conditional level is the proportion of 3845 $ALB$s from permuted data sets that exceed the 95th percentile of $ALB$s formed from 338 permuted data sets. Results are based on 500 replications in each of which one distribution is standard normal and the other is normal with mean 0 and standard deviation 2.

338 permutations are quite good. Eighty-eight percent of the levels are between 0.03 and 0.07, and the approximate unconditional level is .051.

The proportion of $ALB$s from permuted data sets that are larger than the $ALB$ computed from the original data provides a $P$-value. The $P$-values obtained with our method (based on 3845 permutations) are compared to the $P$-values obtained with the Kolmogorov-Smirnov test and Bowman's two-sample test. Results are summarized in Figures 4.7 and 4.8. In 98% of the replications the K-S $P$-value was larger than the $ALB$ $P$-value, and in 57% of the cases the Bowman $P$-value was equal to or larger than the $ALB$ $P$-value. These results suggest that in this case our test has much better power than that of the Kolmogorov-Smirnov test and power at least comparable to that of Bowman's test.

## 4.5 A bivariate extension of the two-sample test and application to connectionist bench data

Our method can be extended to the bivariate case by using a bivariate kernel density estimate. Assume now that $\mathbf{X} = (X_1, ..., X_m)$ are independent and identically distributed from density $f$ and $\mathbf{Y} = (Y_1, ..., Y_m)$ are independent and identically distributed from $g$, where $X_i$ and $Y_j$ are each bivariate observations, $i = 1, \ldots, m$, $j = 1, \ldots, n$.

A product kernel $K$ will be used, i.e., the bivariate kernel $K$ is the product of two univariate

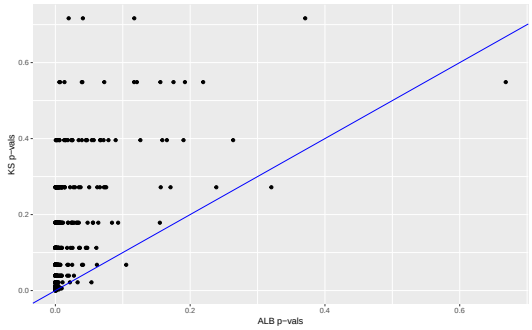



Figure 4.7: *Kolmogorov-Smirnov $P$-values versus $ALB$ $P$-values.* Results are based on 500 data sets in each of which one distribution is standard normal and the other is normal with mean 0 and standard deviation 2. The $ALB$ $P$-value is less than the KS-test $P$-value in 98% of cases. There are only 183 $P$-values from the KS-test that are less than $0.05$.

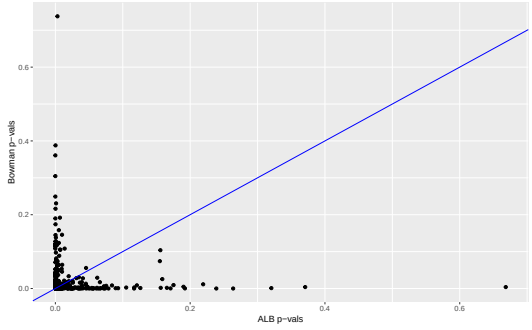



Figure 4.8: *Bowman $P$-values versus $ALB$ $P$-values.* Results are based on 500 data sets in each of which one distribution is standard normal and the other is normal with mean 0 and standard deviation 2. The number of $P$-values less than 0.05 for Bowman's test and the $ALB$ test are 454 and 458, respectively. The ALB $P$-value is less than, more than and equal to the Bowman $P$-value in 49%, 43% and 8% of cases, respectively.

kernels. For $k$ arbitrary bivariate observations $\mathbf{U} = (U_1, \ldots, U_k)$, $U_i = (U_{i1}, U_{i2})$, $i = 1, \ldots, k$, and $u = (u_1, u_2)$, the kernel estimate is defined by

$$\hat{f}_K(u|h, \mathbf{U}) = \frac{1}{kh_1h_2} \sum_{i=1}^{k} K\left(\frac{u_1 - U_{i1}}{h_1}\right) K\left(\frac{u_2 - U_{i2}}{h_2}\right),$$

where $-\infty < u_1 < \infty$, $-\infty < u_1 < \infty$ and $h = (h_1, h_2)$ is a two-vector of (positive) bandwidths.

We will use the same sort of notation as before, i.e., $Z_i = X_i$, $i = 1, \ldots, m$, $Z_i = Y_{i-m}$, $i = m+1, \ldots, m+n$, $\mathbf{Z} = (Z_1, \ldots, Z_{m+n})$ and $\mathbf{Z}_i$ is the object $\mathbf{Z}$ with all its components except $Z_i$, $i = 1, \ldots, m+n$. In this case the $i$th Bayes factor is defined as

$$B_i = \frac{\int_0^\infty \int_0^\infty \int_0^\infty \pi(h_1, h_2) \hat{f}_K(Z_i|h, \mathbf{X}_i)\, dh_1 dh_2}{\int_0^\infty \int_0^\infty \pi(h_1, h_2) \hat{f}_K(Z_i|h, \mathbf{Z}_i) dh_1 dh_2}, \quad i = 1, \ldots, m,$$

and similarly for $i = m+1, \ldots, m+n$. As before the test statistic is $ALB = \sum_{i=1}^{m+n} \log B_i/(m+n)$.

This form may seem daunting, but reduces to a more familiar form if we take $\pi(h_1, h_2) = \pi_0(h_1/b_1)\pi_0(h_2/b_2)/(b_1b_2)$. In this case, proceeding exactly as in Section 2, $B_i$ has the form

$$B_i = \frac{\hat{f}_L(Z_i|b, \mathbf{X}_i)}{\hat{f}_L(Z_i|b, \mathbf{Z}_i)}, \quad i = 1, \ldots, m,$$

and similarly for $i = m+1, \ldots, m+n$, where $b = (b_1, b_2)$ and $L$ is defined by (4.2).

We will analyze the connectionist bench data, which consist of measurements obtained after bouncing sonar waves off of either rocks or metal cylinders. The data may be found at the UCI Machine Learning repository. There are 60 variables in the data set, with $m = 111$ and $n = 97$ measurements of each variable for the metal cylinders and rocks, respectively. We will apply our test to see if the first two variables have a different distribution for rocks than they do for metal cylinders. In our analysis $K$ is taken to be $\phi$, the standard normal density, and $\pi_0$ to be of the form (4.4). In this event $L$ is a $t$-density with $\nu$ degrees of freedom. We will use $\nu = 3$, leading to a fairly heavy-tailed kernel, which is desirable for reasons discussed previously.

The data for each variable are inherently between 0 and 1, and bivariate kernel estimates display

boundary effects along the lines $x = 0$ and $y = 0$, with the largest bias near the origin. We therefore use a reflection technique to reduce bias along these two lines. Suppose one has $k$ observations $(x_1, y_1), \ldots, (x_k, y_k)$ on the unit square. Each observation $(x_i, y_i)$ is reflected to create three new observations: $(x_i, -y_i)$, $(-x_i, -y_i)$ and $(-x_i, y_i)$, $i = 1, \ldots, k$. One then simply computes, at points in the unit square, a standard kernel density estimate from the data set of size $4k$, and multiplies it by 4 to ensure integration to 1. The value of $ALB$ is computed as described previously except that each leave-out estimate leaves out four values: the observation at which the estimate is evaluated plus its three reflected versions. In this way the kde is constructed from data that are independent of the value at which the kde is evaluated.

Kernel density estimates for variables 1 and 2 in the form of heat maps are shown in Figures 4.9 and 4.10, and contours of the estimates are given in Figure 4.11. The latter figure suggests that the distributions for metal cylinders and rock are different. The value of $ALB$ turned out to be $0.013$, and an approximate $P$-value based on 10,000 permuted data sets was 0.0076. So, there is strong evidence of a difference between the rock and metal bivariate distributions. Interestingly, the percentage of negative $ALB$s among the 10,000 permutations was $0.9785$. A kernel density estimate based on the 10,000 values of $ALB^*$ is shown in Figure 4.12.

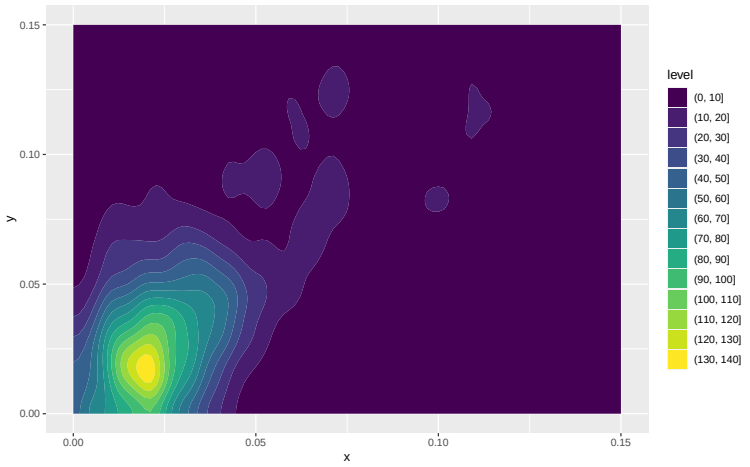



Figure 4.9: A heat map of the first two variables of the signals bounced off the metal cylinder.

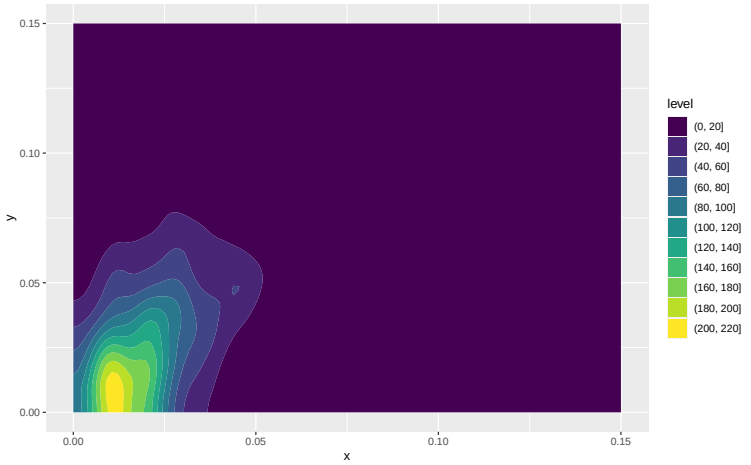



Figure 4.10: A heat map of the first two variables measured of the signals bounced off the rock object.

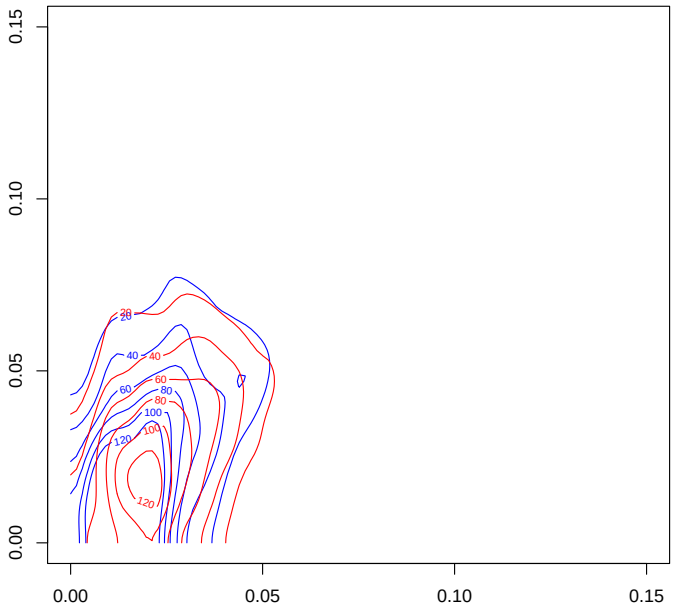



Figure 4.11: Contour plots of the first two variables of both rock and cylinder objects. The blue contour corresponds to the measurements of rocks and red contours correspond to the measurements of the cylinder.

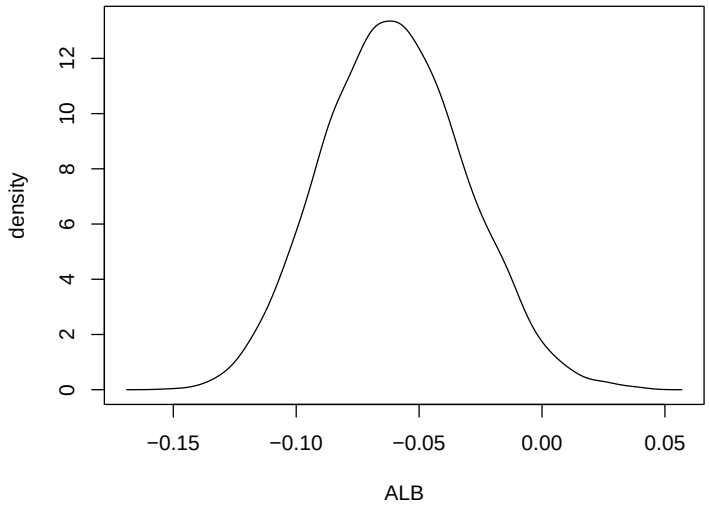



Figure 4.12: A kernel density estimate computed using 10,000 values of $ALB$ from permuted data sets. The value of $ALB$ for the original data set was 0.013.

## 4.6 Conclusion and future work

We have proposed a new nonparametric test of the null hypothesis that two densities are equal. An attractive property of the test is that its critical values are defined by a permutation distribution, allaying essentially any concern about test validity. The fact that the statistic is an average of log-Bayes factors leads to another attractive property: a critical value of 0 leads to a test with type I error probability tending to 0 with sample size. A simulation study showed the new test to have much better power than the Kolmogorov-Smirnov test in a case where the two densities differed with respect to scale. An application to connectionist data illustrated the usefulness of our methodology for bivariate data.

Future work includes efforts to increase the speed of computing the test statistic and its permutation distribution, especially for large data sets. We are also interested in applying the new test to the problem of screening variables prior to performing binary classification. A common method of doing so is to compute a two-sample test statistic for each variable, and to then select variables whose statistics exceed some threshold. An inherent problem in this approach is objectively

choosing a threshold. Results of the current paper suggest that 0 would be a natural and effective threshold for variable screening.

## 4.7 Appendix

### 4.7.1 Consistency

Here we prove

R1. frequentist consistency of our test, and

R2. $P(ALB < 0) \to 1$ as $m, n \to \infty$.

Our proof uses the following assumptions.

A1. Under the null and alternative hypotheses the following integrals exist finite:

$$I_X = \int_{-\infty}^{\infty} f(x) \log f(x)\, dx \quad \text{and} \quad I_Y = \int_{-\infty}^{\infty} g(y) \log g(y)\, dy.$$

When the alternative hypothesis is true, $f$ and $g$ are assumed to be different in the sense that the total variation distance, $\delta(f, g)$, is positive.

A2. The kernel $L$ in $ALB$ (expression (4.3)) is the Hall kernel, $L_0$.

A3. The combined data likelihood cross-validation is maximized over an interval of the form $[(m+n)^{-1+\epsilon}, (m+n)^{-\epsilon}]$, where $\epsilon$ is an arbitrarily small positive constant. The maximizer of this cross-validation is denoted $\hat{b}_{m+n}$.

A4. The ratio $m/(m+n)$ tends to $\rho$, $0 < \rho < 1$, as $m, n$ tend to $\infty$.

A5. The densities $f$, $g$ and $\rho f(x) + (1-\rho)g(x)$ satisfy the conditions of (25) that are needed for the asymptotic optimality of a likelihood cross-validation bandwidth.

A6. Under the null hypothesis, let $\ell_k(b)$ be the Kullback-Leibler risk of a kernel density estimate

based on sample size $k$, kernel $L_0$ and bandwidth $b$. Then $\ell_k$ satisfies

$$\ell_k(b) = C_V(nb)^{-1+a} + C_B b^4 + o\left((nb)^{-1+a} + b^4\right)$$

for positive constants $a$, $C_V$ and $C_B$ with $0 < a < 1$.

Before proceeding to the proof, remarks about assumption A6 are in order. This condition is needed only in proving R2, and represents a subset of the cases studied by (25). It has been assumed merely to allow a more concise proof of R2, which remains true under more general conditions on $\ell_k$.

The critical values of a test with fixed size $\alpha > 0$ will tend to 0 as $m, n$ tend to $\infty$ so long as $ALB$ tends to 0 in probability under the null hypothesis. Therefore, the power of the test will tend to 1 if we can show that $ALB$ tends to a positive constant under the alternative. Our proof of consistency thus boils down to showing that, as $m, n$ tend to $\infty$, $ALB$ converges in probability to 0 and a positive number under the null and alternative hypotheses, respectively.

For data $\mathbf{U} = (U_1, \ldots, U_k)$, define

$$CV(b|\mathbf{U}) = \frac{1}{k}\sum_{i=1}^{k} \log(\hat{f}_L(U_i|b, \mathbf{U}^i)), \quad b > 0.$$

The statistic $ALB$ may then be written

$$ALB = \left(\frac{m}{m+n}\right) CV(\hat{b}|\mathbf{X}) + \left(\frac{n}{m+n}\right) CV(\hat{b}|\mathbf{Y}) - CV(\hat{b}|\mathbf{Z}),$$

where $\hat{b}$ maximizes $CV(b|\mathbf{Z})$ for $b \in [(m+n)^{-1+\epsilon}, (m+n)^{-\epsilon}]$.

Now suppose that $\mathbf{U}$ is a random sample from density $d$, $\ell_k(b)$ is the expectation of the Kullback-Leibler loss of $\hat{f}_L(\cdot|b, \mathbf{U})$ and define

$$Q(k) = \frac{1}{k}\sum_{i=1}^{k} \log d(X_i) - \int d(x) \log d(x)\, dx,$$

where $\int d(x) \log d(x)\, dx$ exists finite. Then if $d$ satisfies the conditions of (25) and $k \to \infty$,

$$CV(b|\mathbf{U}) = \int d(x) \log d(x)\, dx - \ell_k(b) + Q(k) + o_p(\ell_k(b)) \tag{4.8}$$

uniformly in $b \in [k^{-1+\epsilon}, k^{-\epsilon}]$, where $\epsilon$ is arbitrarily small. By the strong law of large numbers $Q(k)$ converges to 0 in probability. Furthermore, $\max_{b\in[k^{-1+\epsilon},k^{-\epsilon}]} \ell_k(b)$ tends to 0 as $k \to \infty$. If the maximizer $\tilde{b}$ of $CV(b|\mathbf{U})$ is in $[k^{-1+\epsilon}, k^{-\epsilon}]$ it therefore follows that $CV(\tilde{b}|\mathbf{U})$ converges in probability to $\int d(x) \log d(x)\, dx$ as $k \to \infty$.

In the null case, (4.8) implies that

$$\left(\frac{m}{m+n}\right) CV(b|\mathbf{X}) + \left(\frac{n}{m+n}\right) CV(b|\mathbf{Y}) - CV(b|\mathbf{Z}) =$$

$$-\left[\left(\frac{m}{m+n}\right) \ell_m(b) + \left(\frac{n}{m+n}\right) \ell_n(b)\right] + \ell_{m+n}(b) + o_p(\ell_m(b)), \tag{4.9}$$

uniformly in $b \in [(m+n)^{-1+\epsilon}, (m+n)^{-\epsilon}]$, where we have used all of A1-A5. Since $\hat{b}_{m+n} \in [(m+n)^{-1+\epsilon}, (m+n)^{-\epsilon}]$, (4.9) implies that $ALB$ converges to 0 in probability as $m, n \to \infty$, which proves one part of R1.

To prove R2, we first observe that the bias component of $\ell_k(b)$ is free of sample size, and hence the first order term of (4.9) is free of bias components. Along with A3 and A6, this implies that

$$\left(\frac{m}{m+n}\right) CV(b|\mathbf{X}) + \left(\frac{n}{m+n}\right) CV(b|\mathbf{Y}) - CV(b|\mathbf{Z}) =$$

$$-C_V((m+n)b)^{-1+a} \left(\rho^a + (1-\rho)^a - 1\right) + o_p\left(((m+n)b)^{-1+a} + b^4\right), \tag{4.10}$$

uniformly in $b \in [(m+n)^{-1+\epsilon}, (m+n)^{-\epsilon}]$. By A5, $\hat{b}_{m+n}$ is asymptotic in probability to $b_{m+n}$, the

minimizer of the Kullback-Leibler risk $\ell_{m+n}$. Along with (4.10), this implies that

$$\begin{aligned} ALB &= -C_V((m+n)b_{m+n})^{-1+a}\left(\rho^a + (1-\rho)^a - 1\right) \\ &\quad + o_p\left(((m+n)b_{m+n})^{-1+a} + b_{m+n}^4\right). \end{aligned}$$

By A6, we have

$$b_{m+n} \sim C_0(m+n)^{-(1-a)/(5-a)},$$

where

$$C_0 = \left[\frac{C_V(1-a)}{4C_B}\right]^{1/(5-a)}.$$

Combining the previous results yields

$$\begin{aligned} ALB &= -\left(\frac{C_V}{C_0^{1-a}}\right)\left(\rho^a + (1-\rho)^a - 1\right)(m+n)^{-4(1-a)/(5-a)} \\ &\quad + o_p((m+n)^{-4(1-a)/(5-a)}). \end{aligned}$$

Using the fact that $(\rho^a + (1-\rho)^a - 1) > 0$ it now follows that $P(ALB < 0) \to 1$ as $m, n \to \infty$.

Turning to the alternative case, we apply (4.8) to conclude that $CV(\hat{b}|\mathbf{X})$, $CV(\hat{b}|\mathbf{Y})$ and $CV(\hat{b}|\mathbf{Z})$ are consistent for $\int f(x)\log f(x)\,dx$, $\int g(x)\log g(x)\,dx$, and $\int f_\rho(x)\log f_\rho(x)\,dx$, respectively, where

$$f_\rho(x) = \rho f(x) + (1-\rho)g(x).$$

It follows that $ALB$ is consistent for $\Delta = \rho KL(f, f_\rho) + (1-\rho)KL(g, f_\rho)$, where $KL(f_1, f_2)$ denotes the Kullback-Leibler divergence between $f_1$ and $f_2$. By the Csiszár-Kemperman-Kullback-Pinsker inequality,

$$\begin{aligned} \Delta &\geq \frac{\log e}{2}\cdot\left[\rho\delta(f, f_\rho)^2 + (1-\rho)\delta(g, f_\rho)^2\right] \\ &= \frac{\log e}{2}\cdot\left[\rho(1-\rho)^2\delta(f,g)^2 + (1-\rho)\rho^2\delta(f,g)^2\right] \\ &= \left(\frac{\log e}{2}\right)\rho(1-\rho)\delta(f,g)^2 > 0, \end{aligned}$$

with the last inequality following by assumption. This completes the proof of R1.

# 5. SCREENING METHODS FOR CLASSIFICATION BASED ON NON-PARAMETRIC BAYESIAN TESTS

## 5.1 Abstract

Feature or variable selection is a problem inherent to large data sets. While many methods have been proposed to deal with this problem, some can scale poorly with the number of predictors in a data set. Screening methods scale linearly with the number of predictors by checking each predictor one-at-a-time, and are a tool used to decrease the number of variables to consider before further analysis or variable selection. For classification, there is a variety of techniques. There are parametric based screening tests, such as $t$-test or SIS based screening, and non-parametric based screening tests, such as Kolmogorov distance based screening (13), and MV-SIS (14). We propose a method for variable screening that uses Bayesian-motivated tests, compare it to SIS based screening, and provide example applications of the method on simulated and real data. It is shown that our screening method can lead to improvements in classification rate. This is so even when our method is used in conjunction with a classifier, such as DART, that is designed to select a sparse subset of variables. Finally, we propose a classifier based on kernel density estimates that in some cases can produce dramatic improvements in classification rates relative to DART.

## 5.2 Introduction

Classification involves predicting a class label for an observation, given a set of predictor variables. The techniques for doing this now are of a wide range, including support vector machines (43), Tree based methods (44), Bayesian trees (7) and gradient boosting trees (45).

It is common to encounter a data set with many features, but rarely are all of them important. Picking a subset of these features quickly is a task that is desired, but can be tricky for very large data sets. Removing unimportant variables can result in dramatic improvement for some of the previously mentioned classification methods. Feature selection is not a new field, and can be divided into three categories: screening or filter based methods, wrapper methods, and embedded

methods (46). Screening methods examine each variable one at a time to see if it provides useful information, and as a result scale linearly with the number of predictors. However, examining each variable individually can cause information on joint behavior of variables to be lost, or can cause collinear variables to be selected (46). Wrapper methods fit different models, and then evaluate each one according to some criterion. The model that does best according to this criterion is selected. While this tends to select good variable subsets, fitting every model can be extremely time-consuming, especially if the data set is very large (46). Embedding methods produce a model that has some sparsity built into it, producing a set of useful variables and a model built with them, simultaneously. The speed of different embedding methods varies with the strategy used to obtain sparsity, but these are typically slower than screening methods (46).

Our focus in this paper will be on screening methods. It is a common strategy to employ a screening method and then employ an embedding method (such as LASSO) afterwards. Fan (11) employs this strategy and improves both the time it takes to run LASSO and the accuracy of the model as a whole. Since then several filter methods have popped up. For classification in particular, maximum marginal likelihood screening (47), MV-SIS (14), and Kolmogorov distance screening (13) are some of the screening tests that have been published. Most of these methods are applied to linear discriminant analysis and show improvement in applying these methods to a data set after the screening has been performed. This paper proposes a new screening method when the number of classes is known to be two, and show that it results in improved classification accuracy in settings where the simple model underlying linear discriminant analysis does not hold. Our screening method identifies informative features by using a two-sample Bayesian test that checks whether two data sets share the same distribution(48). One of our goals is to show that classification methods, including BART (7), DART (7) and SVM (43), can be improved when preceded by our screening procedure.

### 5.3 Methodology

Our screening method is based on computing a statistic for each individual feature. We will use kernel density estimates of the two distributions corresponding to the two classes. The idea

is similar to that of Kolmogorov distance screening: if it seems likely that the two classes have different distributions for a feature, then we will keep the feature. We define a test statistic that can make this determination.

Suppose we observe the data $\mathbf{X}$, an $(m+n) \times p$ matrix whose $i$th row, $(X_{i1}, \ldots, X_{ip})$, contains the values of the $p$ variables for one subject. The $i$th element, $Y_i$, of vector $\mathbf{Y}$ is 0 or 1 and indicates the class to which the $i$th subject belongs, $i = 1, \ldots, m+n$. For now, suppose that $Y_1, \ldots, Y_n$ are 0 and $Y_{n+1}, \ldots, Y_{n+m}$ are 1. We have $n+m$ data vectors and $p$ features. Consider the data $\mathbf{X}_j = (X_{1j}, \ldots, X_{(m+n)j})$ for feature $j$ and define $\hat{h}_i(\cdot|\mathbf{X}_j, b)$ to be a kernel density estimate that has bandwidth $b$ and uses all the data in $\mathbf{X}_j$ except that of the $i$th subject:

$$\hat{h}_i(x|\mathbf{X}_j, b) = \frac{1}{nb} \sum_{r \neq i, 1 \leq r \leq n+m} K\left(\frac{x - X_{rj}}{b}\right).$$

We also compute kernel estimates from the data sets consisting of observations where $Y = 0$ and $Y = 1$. These are

$$\hat{f}_i(\cdot|\mathbf{X}_j, b) = \frac{1}{nb} \sum_{r \neq i, 1 \leq r \leq n} K\left(\frac{x - X_{rj}}{b}\right)$$

and

$$\hat{g}_i(\cdot|\mathbf{X}_j, b) = \frac{1}{nb} \sum_{r \neq i, n < r \leq n+m} K\left(\frac{x - X_{rj}}{b}\right).$$

Then we define the test statistics $ALB_j$, $j = 1, \ldots, p$:

$$\begin{aligned}(m+n)ALB_j \;\; = \;\; & \sum_{i=n+1}^{m} \log(\hat{g}_i(X_{ij}|\mathbf{X}_j, b)) + \sum_{i=1}^{n} \log(\hat{f}_i(X_{ij}|\mathbf{X}_j, b)) - \\ & \sum_{i=1}^{n+m} \log(\hat{h}_i(X_{ij}|\mathbf{X}_j, b)).\end{aligned}$$

Define $f_{j,\text{mix}} = (nf_j + mg_j)/(n+m)$, where $f_j$ and $g_j$ are the densities of feature $j$ for classes 0 and 1, respectively. The statistic $ALB_j$ is an approximately unbiased estimator of the following quantity:

$$\left(\frac{n}{m+n}\right) KL\left(f_j, f_{j,\text{mix}}\right) + \left(\frac{m}{m+n}\right) KL\left(g_j, f_{j,\text{mix}}\right).$$

In the case $f \equiv g$, $ALB$ converges to 0 in probability. The null distribution of $ALB$ can be assessed using a permutation-based procedure to determine if two sets of observations arise from a common distribution. We refer to (48) for a thorough exploration of this procedure.

In (48), it is encouraged to use a bandwidth that is selected by leave-one-out cross validation. While this strategy has potential, we believe it to be too computationally expensive to pick a bandwidth for every variable using this procedure. We only have to do this $p$ times, suggesting linear scaling with the variable length, but this introduces quadratic scaling with $n$, which is prohibitive for large data sets. Instead, we opt to use a normal plug-in bandwidth in conjunction with the heavy tailed Hall kernel, which is:

$$K_0(z) = \frac{1}{\sqrt{8\pi e}\,\Phi(1)} \exp\left[-\frac{1}{2}(\log(1+|z|))^2\right].$$

Simulation results show that the constant for the plug-in is $0.162$ to 3 decimal places, resulting in the following plug-in rule:

$$b_{\text{plug-in}}(X_1, X_2, ..., X_n) = 0.162 n^{-1/5} s,$$

where $s$ is an estimate of the underlying standard deviation. One possibility is to take $s$ to be the sample standard deviation, but we prefer the more robust choice $s_R = IQR(X_1, X_2, ..., X_n)/1.35$.

Suppose we have computed $ALB_1, ALB_2, ALB_3, ..., ALB_p$. The matter still remains in choosing a cutoff for the $ALB$s such that we select all variables with $ALB$ larger than the cutoff. Below are some possible ways of doing so.

A1. Choose the cutoff to be some percentile of $ALB_1, ALB_2, ALB_3, ..., ALB_p$. This is in line with what some of the authors in SIS and SIRS propose. Suppose we expect $d$ features to be important in the data set, then we can set our cutoff to be the $100(1 - d/p)$th percentile. Clearly, the largest $d$ test statistics are the most likely to be significant. This does have some flaws, as a good choice of $d$ may not be obvious. The authors in SIS (11) and SIRS (12) argue that a conservative choice for $d$ is $n$ or $n\log(n)$, but this can be argued to be arbitrary.

A2. Simulate $l$ data sets, each of size $m+n$, that are all random samples from a standard normal distribution. The first $m$ and last $n$ values of each sample become two data sets, from which $ALB$ is calculated. We pick the cutoff to be the largest of the $l$ ALB values. This is also an idea proposed by SIRS (12), but it is unclear what $l$ should be. An important fact that makes this procedure sensible for our purposes is that the null distribution of $ALB$ is invariant to location and scale.

A3. Permute the response vector $Y$ with a randomly chosen permutation matrix $P$. Rather than use the permutation matrix on a single predictor, we use it on all the predictors to form $\mathbf{X}_1^* = P\mathbf{X}_1, \mathbf{X}_2^* = P\mathbf{X}_2, ..., \mathbf{X}_p^* = P\mathbf{X}_p$. We then compute $ALB_1^*, \ldots, ALB_p^*$ from $\mathbf{X}_1^*, \ldots, \mathbf{X}_p^*$, respectively. This procedure is repeated $B-1$ times using $B-1$ more randomly chosen permutation matrices, with the result being a total of $Bp$ values of $ALB^*$. The cutoff is selected to be a percentile of these $Bp$ values. This procedure approximates the null distribution of $ALB$ for a randomly selected feature conditional on the observed data. Our experience says that $B$ need not be extremely large in order for the approximation to be good. Another possibility is to use different permutations on different features, as proposed in A4.

A4. Randomly select a covariate, say $\mathbf{X}_j$. For this covariate, permute the labels, and compute the test statistic, call it $ALB_1^*$. Using the same feature $\mathbf{X}_j$, repeat this procedure $d$ times, resulting in $ALB_1^*, \ldots, ALB_d^*$. We randomly select another covariate without replacement, and repeat the previous steps $B$ times, resulting in a total of $Bd$ values of $ALB$. Once this is done, we choose the cutoff to be a percentile of these $Bd$ values. This method also approximates the null conditional distribution of $ALB$ for a randomly selected feature, but potentially has the advantage of requiring fewer statistics to be computed than in A3.

A5. Choose the cutoff based on an interpretive approach. A feature $\mathbf{X}_j$ can be considered "useful" if either $f_j(x)/g_j(x) > T$ or $f_j(x)/g_j(x) < 1/T$, where $T > 1$. Defining

$p = n/(m+n)$, these inequalities are true if and only if

$$\frac{f_j(x)}{f_{j,\text{mix}}(x)} > \frac{T}{pT + (1-p)} \quad \text{or} \quad \frac{g_j(x)}{f_{j,\text{mix}}(x)} > \frac{T}{(1-p)T + p},$$

where $0 < p < 1$. As noted previously $ALB_j$ estimates

$$p \int \log\left(\frac{f_j(x)}{f_{j,\text{mix}}(x)}\right) f_j(x)\, dx + (1-p) \int \log\left(\frac{g_j(x)}{f_{j,\text{mix}}(x)}\right) g_j(x)\, dx$$

which suggests that we define a variable as useful if

$$ALB_j \geq p \log\left(\frac{T}{pT + (1-p)}\right) + (1-p) \log\left(\frac{T}{(1-p)T + p}\right).$$

This requires a somewhat subjective choice of $T$, but is a strategy with fairly low computational resources required. The analogy of density ratios and Bayes factors suggest that we use Jeffreys' scale to choose $T$. Jeffreys' cutoff for substantial evidence is $T = \sqrt{10}$, but in our experience this cutoff is too conservative. We recommend a cutoff of $T = 2$, which results in an ALB threshold of 0.288 when $m = n$. This can be made higher or lower based on the field to which the method is applied.

A6. An empirical but computationally daunting way to approach this problem is to proceed with two training sets and a classification method. We can choose the cutoff that minimizes the error rate of the classification method when it is trained on one of the training sets and then applied to the other. To keep the computational scaling of the procedure linear with $p$, we recommend restricting the number of candidate cutoff values to be fairly small, say no more than ten.

A7. The Bayes factor interpretation of $ALB_j$ entails that variable $j$ should never survive screening when $ALB_j < 0$. Picking 0 as a cutoff corresponds to using $T = 1$ in A5. While this choice may seem liberal, (48) show that an $ALB$ cutoff of 0 produces a test whose type I

error probability tends to 0 as $m + n$ tends to $\infty$.

We note the following in regards to each procedure. Simulating i.i.d. random normal variables and performing a permutation based procedure both result in ALBs that offer insight on how ALBs behave in the case where the data from classes do not differ in distribution. Simulating normal random variables is a smaller computational burden than repeatedly permuting the labels, but the distributions sampled from permutation will often be closer to those of the observed features, and hence more relevant.

It should be noted that a particular screening method will work differently with different classifiers, although this is perhaps to be expected. We encourage the use of a classifier that can take advantage of differences in distribution other than location differences. A popular classifier is linear discriminant analysis (LDA), by which we mean the version that assumes equal covariance matrices for the two classes. Features identified as important due to a scale difference between classes will usually be of no use to LDA.

The least computationally expensive procedures involve (a) choosing a cutoff based on the interpretability of ALB, and (b) choosing the cutoff to be one of the top percentiles of the $ALB$s. As previously discussed, two interpretable ALB cutoffs are 0 and one resulting from choosing $T = 2$. Using either of these cutoffs produces a test with power tending to 1 as $m + n$ tends to $\infty$, since $ALB_j$ converges to 0 in probability when $f_j \equiv g_j$ (48). Choosing a cutoff as in (b) is recommended if there is a strong idea as to how many variables are expected to be relevant or if there is a critical number of variables that are needed for another classifier to work well.

Lastly we wish to note that each $ALB$ has a finite upper bound, as it is easily shown that

$$ALB_j \leq \log(2) \cdot \max\left(\frac{m}{(m-1)}, \frac{n}{(n-1)}\right),$$

which implies that $ALB_j$ is essentially bounded by $\log(2)$ so long as $m$ and $n$ are not too small. An alternative strategy for choosing a cutoff is to specify $q$ in $\log(2q)$, where $.5 \leq q \leq 1$. We recommend $q = .6$ in this approach. Choosing $q = .5$ results in an ALB cutoff of 0, and choosing

$q = 1$ results in a cutoff of $\log(2)$. The larger the $q$ value, the harsher the screening method and the less variables survive the cutoff.

### 5.4 Consistency results

We begin with the assumption that every variable satisfies conditions A1-A5 in (48). We will also assume that the numbers, $m$ and $n$, of samples for the two classes tend to infinity, and the number of variables, $p$, is fixed. Suppose that a variable belongs to class $D$ if the variable marginally offers information, which means that the variable has a different distribution for one class than it does for the other. Finally, we assume that the variables are independent.

**Theorem 2.** *If the above assumptions hold, then*

$$\lim_{n,m\to\infty} P(\max_{i\in D^c} ALB_i < \min_{j\in D} ALB_j) \to 1. \tag{5.1}$$

*If we choose a cutoff as in A7 (in our Section 3) and call it $Z$, then we also have the following result:*

$$\lim_{n,m\to\infty} P(\min_{j\in D} ALB_j > Z \cap \max_{j\in D^c} ALB_j < Z) \to 1. \tag{5.2}$$

*Proof.* We consider the case where $Z = 0$, although the actual value of $Z$ is immaterial. Result (5.2) implies (5.1), so we only prove the former. Using the fact that $P(A\cap B) \geq 1-P(A^c)-P(B^c)$, it is enough to show that both $P(\min_{j\in D} ALB_j > 0)$ and $P(\max_{j\in D^c} ALB_j < 0)$ tend to 1.

We only consider $P(\max_{j\in D^c} ALB_j < 0)$ as the proof for the other probability is similar. We have

$$\begin{aligned} P(\max_{j\in D^c} ALB_j < 0) &= P\left(\bigcap_{j\in D^c} \{ALB_j < 0\}\right) \\ &= \prod_{j\in D^c} P(ALB_j < 0) \\ &\geq \left(\min_{j\in D^c} P(ALB_j < 0)\right)^N, \end{aligned} \tag{5.3}$$

where $N$ is the number of elements in $D^c$. For each $j \in D^c$, (48) show that $P(ALB_j < 0) \to 1$ as $m$ and $n$ tend to $\infty$. Since $N$ is finite, this implies that the quantity on the right-hand side of (5.3) tends to 1, from which the result follows.

It is clear that if $p$ tends to $\infty$ at a sufficiently slow rate, then Theorem 1 remains true. However, determining the precise rate at which $p$ can increase relative to $m$ and $n$ requires stronger results than provided by (48), and we will not pursue this direction further.

We now show simulation results for various values of $m = n$, $p$ and $r$, where $r$ represents the proportion of important variables, i.e., variables for which the class distributions are different. We generate 500 variables for a binary classification problem in the following fashion. If it is important, the variable is drawn for one class from a $t$-distribution with 4 degrees of freedom, and drawn from the other class from a mixture of two normal distributions, where the mixing parameter is 1/2, the standard deviation of both normal distributions is 1, and the means are -2.5 and 2.5. If instead the variable is unimportant, suppose that it is drawn from a standard normal. We will name the method of generating variables in this setting “a shape difference". Finally, each variable is determined to be important or not by performing a binomial trial with success probability $r = 1/2$. Figures 5.1, 5.2, and 5.3 show how the cdfs of the $ALB$s change depending on $m$ and $n$.

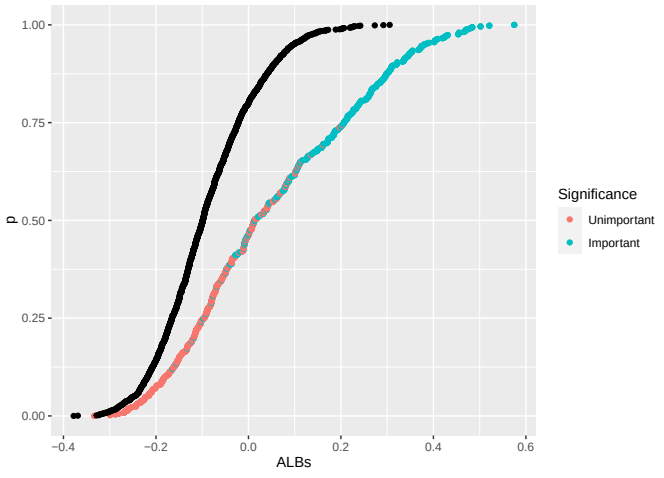



Figure 5.1: *Comparison of ALB CDFs in when the training set sizes are equal to 10 and variables are generated according to “a shape difference".*
The black curve denotes the CDF of $ALB$s generated from data where the classes are permuted.

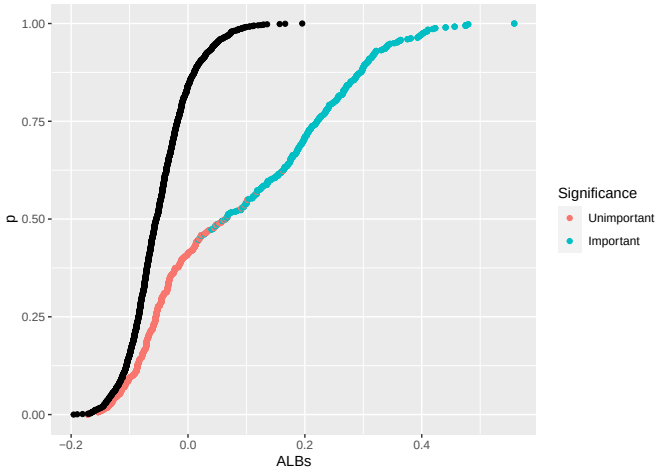



Figure 5.2: *Comparison of ALB CDFs in when the training set sizes are equal to 20 and variables are generated according to "a shape difference".*
The black curve denotes the CDF of $ALB$s generated from data where the classes are permuted.

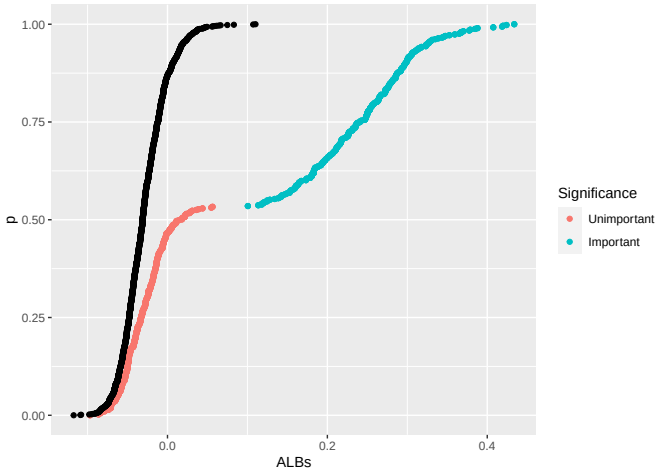



Figure 5.3: *Comparison of ALB CDFs in when the training set sizes are equal to 40 and variables are generated according to "a shape difference".*
The black curve denotes the CDF of $ALB$s generated from data where the classes are permuted.

As the sample sizes $m$ and $n$ increase, the $ALB$s for important variables gradually increase. Even when the total number of observations is only eight percent of the total number of variables, we achieve the property that the largest $ALB$ of the unimportant variables is smaller than the

smallest $ALB$ of the important variables. The black curve shows the cdf of $ALB$s computed by permuting the labels for each variable three times and computing the $ALB$ each time. A cutoff of 0 is not larger than the largest unimportant variable for any $n$, but is still useful for discarding a large portion of the unimportant variables. On the other hand, using a large percentile of the permuted variables can result in discarding almost all of the unimportant variables, and at the largest sample size, choosing the cutoff to be the maximum of the permuted ALBs does indeed almost perfectly separate the important and unimportant variables.

### 5.5 Discussion of classification methods

Ideally, one should choose the screening method and classifier that work best together. Good examples of this principle are provided by the relationship that classification methods such as logistic regression, support vector machines, and linear discriminant analysis have with $t$-test based screening. Discriminant analysis, support vector machines without a kernel trick, and logistic regression are designed to take advantage of location differences between classes. It is therefore natural to precede them with $t$-test screening, which, of course, is designed to detect differences between means. On the other hand, support vector machines that use a kernel trick create a hyperplane that best separates the two classes essentially after a transformation is performed, and can therefore deal effectively with many types of differences between distributions. To take advantage of this ability, it is thus best to use a screening method that can detect non-location differences. In summary, $t$-test screening is a natural method to use when linear discriminant analysis or logistic regression are deemed to be appropriate classifiers, but is not necessarily a good method when a support vector machine with a kernel trick is required.

In Figures 5.4 and 5.5 two (important) variables are generated according to a “shape difference". The bimodality of one of the two class distributions makes the classes hard to separate with a plane. Figure 5.4 shows the performance of a support vector machine when only two relevant variables are used for classification, while Figure 5.5 shows the improvement in the same situation when the kernel trick is applied to detect non-location based differences.

Our method seeks to outperform $t$-test screening by considering differences other than ones

of location type. Of course, this performance is not free. It comes with the cost that we lose some power in detecting differences of means. For our method to work better with a classifier, the classifier must have the ability to distinguish classes that display non-location differences. For example, a set of variables whose classes differ only with respect to scale would not be useful to support vector machines without the kernel trick and LDA, as there would be no hyperplane that nicely separates the classes. A kernel trick or increasing the number of variables by considering interactions and squared terms can sidestep this issue. But adding variables is not ideal, as a goal of our methodology is to decrease computational complexity.

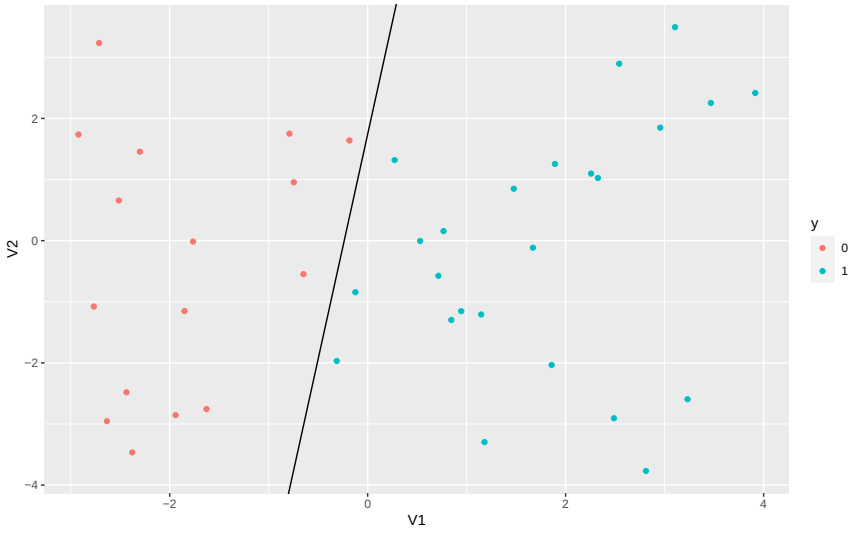



Figure 5.4: *Prediction accuracy of two important variables in the setting of "$\kappa$" using a "linear" svm.* The colors represent the predictions that the SVM produces. The line represents the discriminator that a linear SVM produces to discriminate the classes. The triangle and circle represent which class the observation arises from.

Finally, we would like to add that even though SVM with the kernel trick is a fine classification method that uncovers many different types of differences between variables, its performance can be degraded harshly by the presence of noisy variables. To illustrate this, we use data in the same setting as the "shape difference". We constructed a training and testing set such that both consist of 10 observations from each class. Five hundred variables were used, with only 10% on

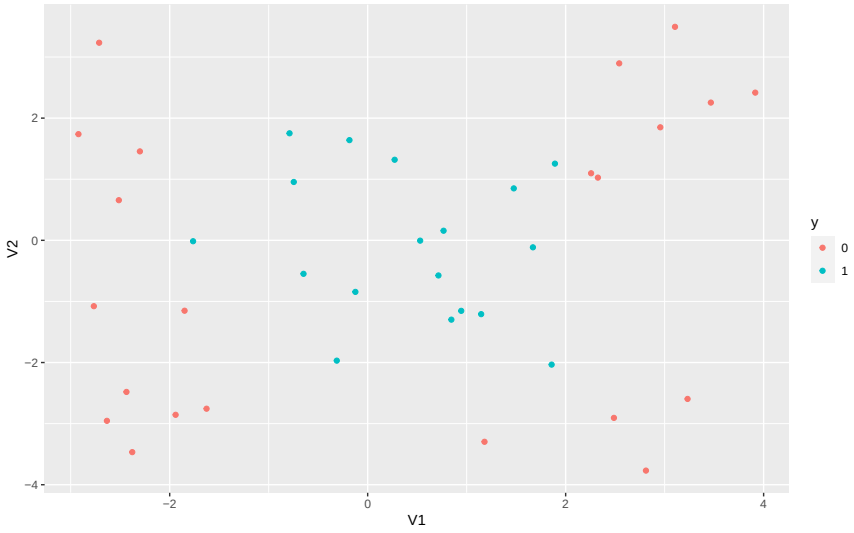
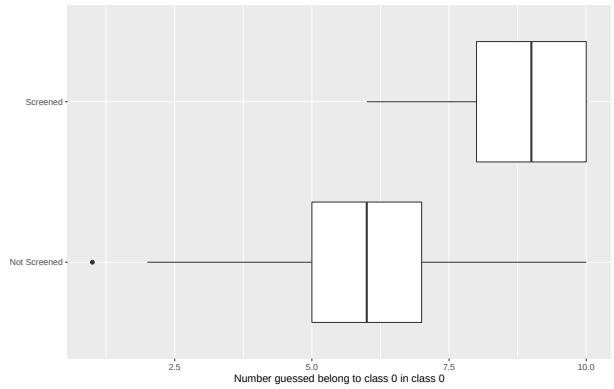


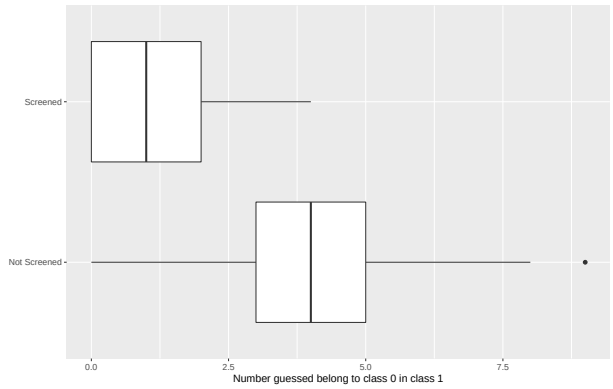

Figure 5.5: *Prediction accuracy of two important variables in the setting of "$\kappa$" with an SVM that uses a kernel trick.* The colors represent the predictions that the SVM produces when a kernel trick is applied. The triangle and circle represent which class the observation arises from. Classification is much better in this case because the trick enables the classification method to become capable of capturing differences outside of location shifts.

average being important. All data for unimportant variables have a standard normal distribution. If a variable is important, then its distribution in one class is a bimodal mixture of two normals and in the other class a $t$-distribution with 4 degrees of freedom. The normal distributions in the mixture both have standard deviation 1 and means of -2.5 and 2.5. We trained an SVM with the radial basis kernel on all of the observations, and trained another SVM with the radial basis kernel but used only variables whose ALB value was larger than the interpretative cutoff of $\log(1.2)$. Choosing this cutoff is an example of A5, and tends to be more "conservative" than a permutation based procedure with significance level of 0.01 or more. We repeat this procedure 100 times, and report on its accuracy in Figure 5.6. In general, classification accuracy is greatly improved, false negatives rarely happen after screening and the number of false positives is reduced.

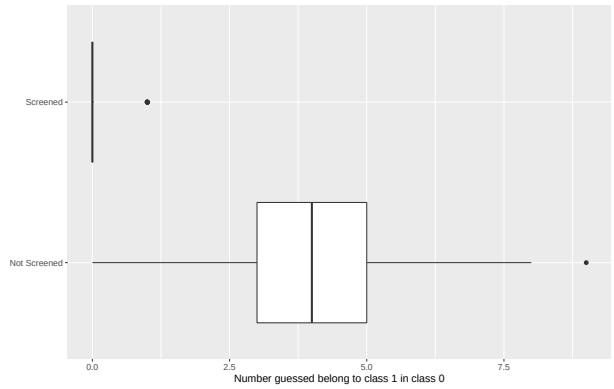
The ALB screening method need not be the final say as to which variables to include. Two variables that are individually important but highly correlated might be selected, although this may not be ideal for some classifiers. Screening can simply be a precursor that simplifies the job of a classifier, which does further variable selection. Even methods that can perform variable



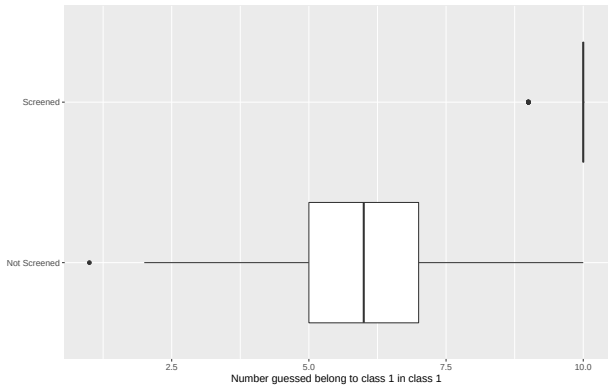

(a) The boxplots of "True Positives"




(b) The boxplots of "False Positives"




(c) The boxplots of "False Negatives"




(d) The boxplots of "True Negatives"

Figure 5.6: *Boxplots displaying the accuracy of SVM models when the data are generated from model where* $10\%$ *of the variables are important and differ according to a "shape difference".* To illustrate the accuracy of the methods, we do the following. First, suppose a positive case corresponds to an observation being in class 1 and a negative case corresponds to an observation being in the other class. Then to show the accuracy of the SVMs, we show boxplots on the number of "True Positives", "True Negatives", "False Positives", and "False Negative" occurrences.

selection and modeling simultaneously can benefit from having the number of variables reduced dramatically by screening. This is observed in SIRS (12), SIS (11), and is also true in our case.

### 5.6 Interaction with BART and a tailored classification method

We have recommended using classification methods that can take advantage of features for which the classes have non-location differences. Methods that do further variable selection or that can handle sparse data sets can also fare quite well with our screening methods. BART and DART are methods having few parametric assumptions and that are able to capture a large variety of features from the data. BART has issues as the number of predictors grow, and DART has been proposed as a solution for this issue (7). While DART can handle the case where many predictors are irrelevant, there is a cost. Mixing times of the chains for DART are increased compared to BART, and a prior that encourages sparsity may cause DART to get trapped in a posterior mode when the MCMC procedure to estimate it is run (9). While we cannot directly abate these problems, decreasing the number of variables helps speed up the MCMC procedure. Our screening method can decrease the number of variables at a faster rate than DART can. DART is resilient against correlated nuisance variables and can therefore eliminate variables that survive ALB screening but are irrelevant due to collinearity. We provide simulations showing that use of our screening method before BART or DART can result in improved misclassification rates and computing speeds.

We generate data in the same context as 5.2, but instead roughly 10% of the variables are relevant. If a variable is irrelevant, the distribution of the variable for both classes is standard normal. To assess the performance of a classifier, we computed the Rand index, or the percentage of correct decisions the classifier has made. We compute the Rand index for the BART and DART procedures applied to all variables, and the Rand index of the same procedures applied to variables that survive ALB screening. We consider different training set sizes that vary from 5 to 20. The testing set size for each simulation is the same as the training set size. We repeat this 100 times for each sample size.

Figures 5.7 and 5.9 show how accurate BART and DART alone are in these settings and Figures 5.8 and 5.10 show how the methods do when variables are screened for importance beforehand.

There is a notable gain in the Rand index as the training set size gradually increases for both methods. Of greater note is that the time it takes to run both procedures is decreased. Figure 5.11 shows the amount of time it takes the BART method to run before screening and Figure 5.12 shows how long the method takes after screening. Screening on average shaves off at least 10 seconds of computation time while increasing the average accuracy. This is an interesting result, as the methods themselves, DART especially, tend to be robust to irrelevant variables. However, the figures suggest that a larger sample size is required to achieve that robustness.

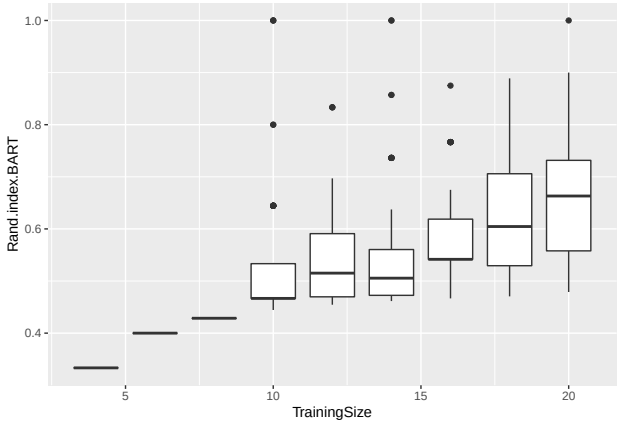



Figure 5.7: *A box plot of the rand index of BART in the setting of "a shape difference"*. We vary $m$ and $n$ but have that $m = n$, and the training size in the plot denotes $m + n$. We repeat each simulation 100 times for each sample size. Roughly 10% of the variables are relevant.

### 5.6.1 A simple Bayesian classifier

Suppose our goal is to simply leverage the differences between variables, regardless of the type of difference, and that we assume independence between variables. We can construct a simple Bayesian method for classification in the following fashion. For each variable, we compute two kernel density estimates, one for each class. For each variable $i$, let $\hat{f}_i$ and $\hat{g}_i$ be kernel density estimates using all variable $i$ data from classes 1 and 2, respectively. The prior probability that a variable arises from a class is assumed to be proportional to the number of observations for that class. Let $\boldsymbol{x} = (x_1, ..., x_p)$ be an observation to be classified. If the underlying densities are known,

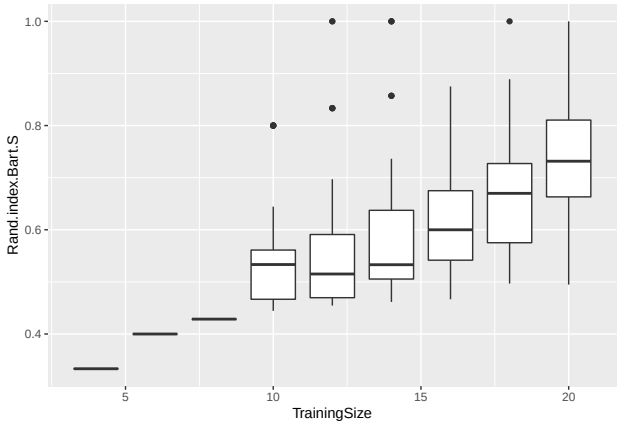



Figure 5.8: *A box plot of the rand index of BART when BART is improved by screening.* This is in the same setting as 5.7. The difference between the two plots is that we screened the variables with the ALB procedure before applying BART. We choose variables such that all generated ALBs are larger than the interpretable cutoff of $\log(.6) + \log(2)$. The power of the classification method grows large when enough data is accrued. The interpretive cutoff is likely to give variables that are quite conservative, and power of the approach is likely to be even larger if a permutation based cutoff is utilized instead.

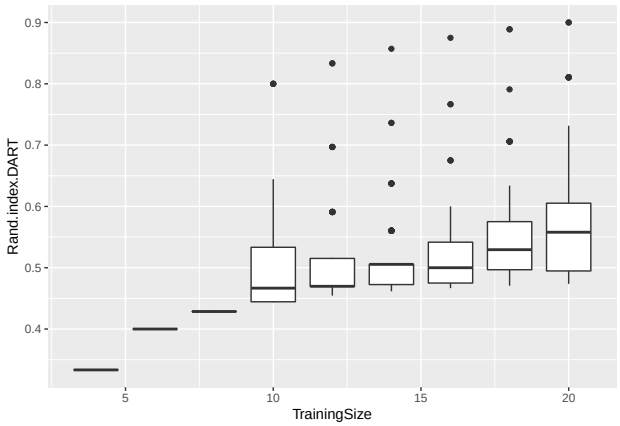



Figure 5.9: *A box plot of the rand index of DART*. This is of the same setting as 5.7, the difference is we use DART instead of BART as it is capable of automatically performing variable selection.

then the conditional probability that $\boldsymbol{x}$ came from class 1 is

$$p(\boldsymbol{x}) = P(Y = 1|\boldsymbol{x}) = \frac{\frac{n}{m+n}\prod_{i\in D} f_i(x_i)}{\frac{n}{m+n}\prod_{i\in D} f_i(x_i) + \frac{m}{m+n}\prod_{i\in D} g_i(x_i)}, \tag{5.4}$$

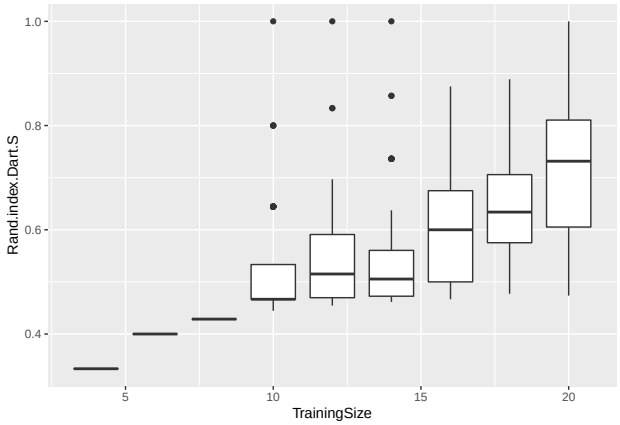



Figure 5.10: *A box plot of the rand index of DART improved by screening.* This is in the same setting as 5.9. The difference here is we screen the variables with the ALB procedure before applying DART. We choose variables such that all generated ALBs are larger than the intepretable cutoff of $\log(.6) + \log(2)$. The power of the classification method has improved after screening for variables, despite DART being fully capable of automatically performing variable selection automatically.

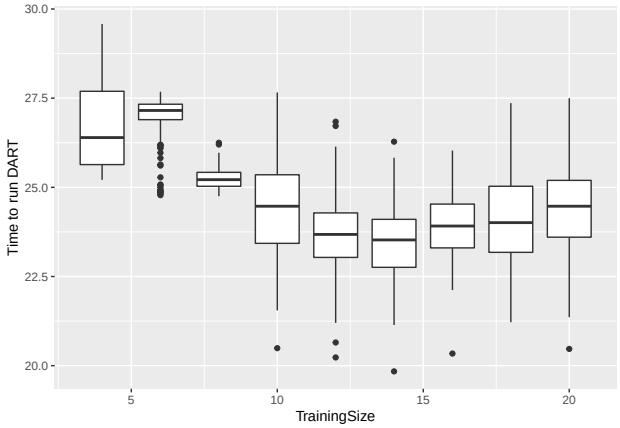



Figure 5.11: *A box plot of the time it took to run DART.*
The simulation is of the same setting as 5.7.

where $D$ is the set of indices $i$ such that $f_i \not\equiv g_i$. Of course, the densities and $D$ are unknown, but $p(\boldsymbol{x})$ can be estimated using kernel density estimates, and $D$ can be replaced by $\widehat{D}$, the set of

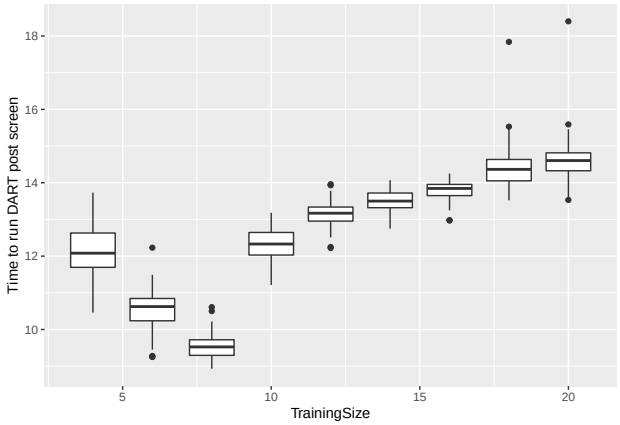



Figure 5.12: *A box plot of the time it took to run DART post screening.* This is in the same setting as 5.11. The difference here is we screen the variables with the ALB procedure before applying DART. The screening procedure itself takes much less than half a second, and as a result of removing a large number of irrelevant variables, greatly improves the amount of time it takes for DART to run.

indices such that the corresponding variables survive screening:

$$\hat{p}(\boldsymbol{x}) = \frac{\frac{n}{m+n}\prod_{i\in\widehat{D}}\hat{f}_i(x_i)}{\frac{n}{m+n}\prod_{i\in\widehat{D}}\hat{f}_i(x_i) + \frac{m}{m+n}\prod_{i\in\widehat{D}}\hat{g}_i(x_i)}. \tag{5.5}$$

A classifier based on (5.5) can be a powerful tool for capturing marginal differences in distributions, but is incapable of leveraging differences that may lie in the dependence structure of the variables. To use (5.5), we say that an observation $\boldsymbol{x}$ belongs to class 1 if $\hat{p}(\boldsymbol{x}) > n/(m+n)$ and to class 2 otherwise. We have found this classifier to have strong accuracy when dealing with independent variables, or with settings where the difference in multivariate distributions is dominated by marginal differences.

To illustrate the effectiveness of the classifier based on (5.5), we perform a simulation similar to the one referenced in Figure 5.1 but under a variety of different sample sizes. We repeat this procedure 5 times. For each sample size, we generate a balanced testing set of the same size, and compute a Rand index. A plot of the Rand indices against the sample size is found in Figure 5.13. The power of the classification method grows to be quite large at even small $m+n$ values. Since the interpretive cutoff tends to be conservative, power of the approach is likely to be even larger if

a permutation based cutoff is utilized instead. At a training size of just 9 samples in each group, the classifier makes perfect predictions over 75% of the time.

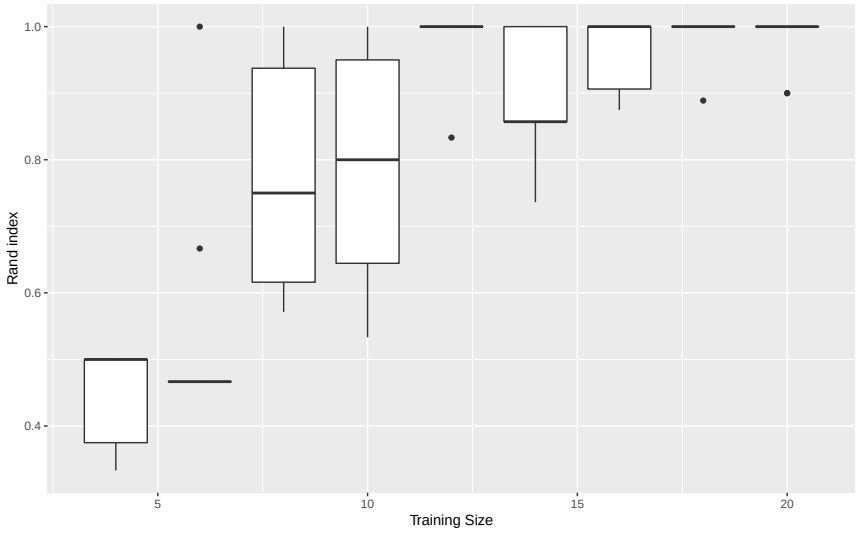



Figure 5.13: *A box plot of the Rand index of the Bayesian classifier.* The classifier can be seen by examining 5.5. We generate data in the same context as 5.2 but vary $m$ and $n$ so that $m = n$, and the training size denotes $m + n$. We choose variables such that all generated ALBs are larger than the intepretable cutoff of $\log(1.2)$.

## 5.7 Application on simulated data sets

We want to compare our method to $t$-test screening and also compare performance of the different choices of ALB cutoff. There are at least two ways to go about this. One is to compare the percentage of variables that survive screening from both procedures, and the other is to apply a classification method after screening and see which method has a better classification rate. These procedures are carried out in three cases:

Case 1 – *Location differences.* There are 600 variables, and those that are important arise from a case where there is a mean difference between classes. If the variable is important, one class has a standard normal distribution and the other a normal distribution with mean 1 and standard deviation 1. We let roughly $5\%$ of the variables be important by generating 600

independent Bernoulli variables, each with success probability $0.05$. This way of determining important variables is used in Cases 2 and 3 as well. Also, in this case and the following two the unimportant variables have standard normal distributions. The classifier we will use to compare performance in this case is SVM without the kernel trick.

Case 2 – *Scale differences.* There are 600 variables, and those that are important arise from a case where there is a variance difference between classes. If the variable is important, one class has a standard normal distribution and the other a normal distribution with mean 0 and standard deviation 3. We let roughly $20\%$ of the variables be important, and the classifier used is the support vector machine with a kernel trick.

Case 3 – *Shape differences.* There are 600 variables, and those that are important arise from a case where the class distributions have different shapes. If the variable is important, one class has a standard $t$-distribution with 4 degrees of freedom and the other a bimodal mixture of two normal distributions with means -2.5 and 2.5 and the same standard deviation of 1. Roughly $10\%$ of the variables are important, and the classifier used is the support vector machine with a kernel trick.

For all three cases, screening was done and the classifier built from a training set of $m + n$ observations on each variable, where $m = n$. The classifier so built was applied to predict $m + n$ observations, and the resulting Rand index was calculated. In $t$-test screening, variables were selected when their $P$-values were smaller than $0.005$. We repeated this procedure 100 times for each sample size. Figures 5.14-5.16 show, respectively, how well the methods performed for three ways of choosing an ALB cutoff: a "fixed type I error rate" approach, the largest $n + m$ values of ALB, and a cutoff equal to the interpretable value of $\log(1.2)$.

The results of these simulations suggest that ALB screening is effective at detecting location differences, as in Case 1, but not to the same degree as $t$-test screening. In Case 1, the performance of the SVM with ALB screening is better than with no screening, but worse than with $t$-test screening. The proportion of variables that survive ALB screening steadily increases as sample size

increases, but at a slower rate than with $t$-test screening. In Cases 2 and 3, $t$-test screening does no better than no screening in terms of classification accuracy. Regarding preservation of important variables, $t$-test screening does not improve as the sample size increases, but ALB screening does.

### 5.8 Application to the GISETTE data

The GISETTE data are obtained from $m = 3000$ and $n = 3000$ handwritten images of the digits 4 and 9, respectively. For each of the 6000 images, $p = 5000$ variables are measured, some of which are irrelevant probes, and the others pixel intensities. We will perform classification on these data, using different screening methods to choose different subsets of the variables. We will rely on DART to be the primary classification method and will explore how it performs when aided by different screening methods.

The data set was randomly split into two halves. The first half was treated as the training data. We trained our classifier and computed $t$-statistics and ALB statistics on these data. The second half was treated as the validation set, and we computed the Rand index from these data.

Five screening methods were compared. We implemented A4, setting $B = 1000$ and $d = 1$ and choosing variables such that $ALB$ was larger than the $99.5$th percentile of $ALB^*$ values. To compare this with $t$-test screening, we picked variables whose $t$-test $P$-values were less than 0.005. We tried screening method A5, with the interpretable cutoffs of $\log(1.2)$ and $0$. We compare the Rand indices of these methods with that when no screening of variables is used. Before proceeding with any of the methods, we removed each variable for which all 6000 data values were the same. As a result there were only 4835 variables in the full data set rather than 5000. A summary of Rand indices is given in Table 5.1.

For these data, $t$-test screening does as well as ALB screening based on A4 and the 99.5th percentile. The difference between the two methods is that ALB retains the same accuracy while picking roughly 200 fewer variables. The interpretable cutoff rule does the worst, but 75% accuracy using only four “pixel" measurements is a very interesting result. We believe this is a setting where most variables, or "pixels," that are marginally important are ones that are colored in for one of the two numbers (4 or 9) but not the other. This can be interpreted as a location difference, since

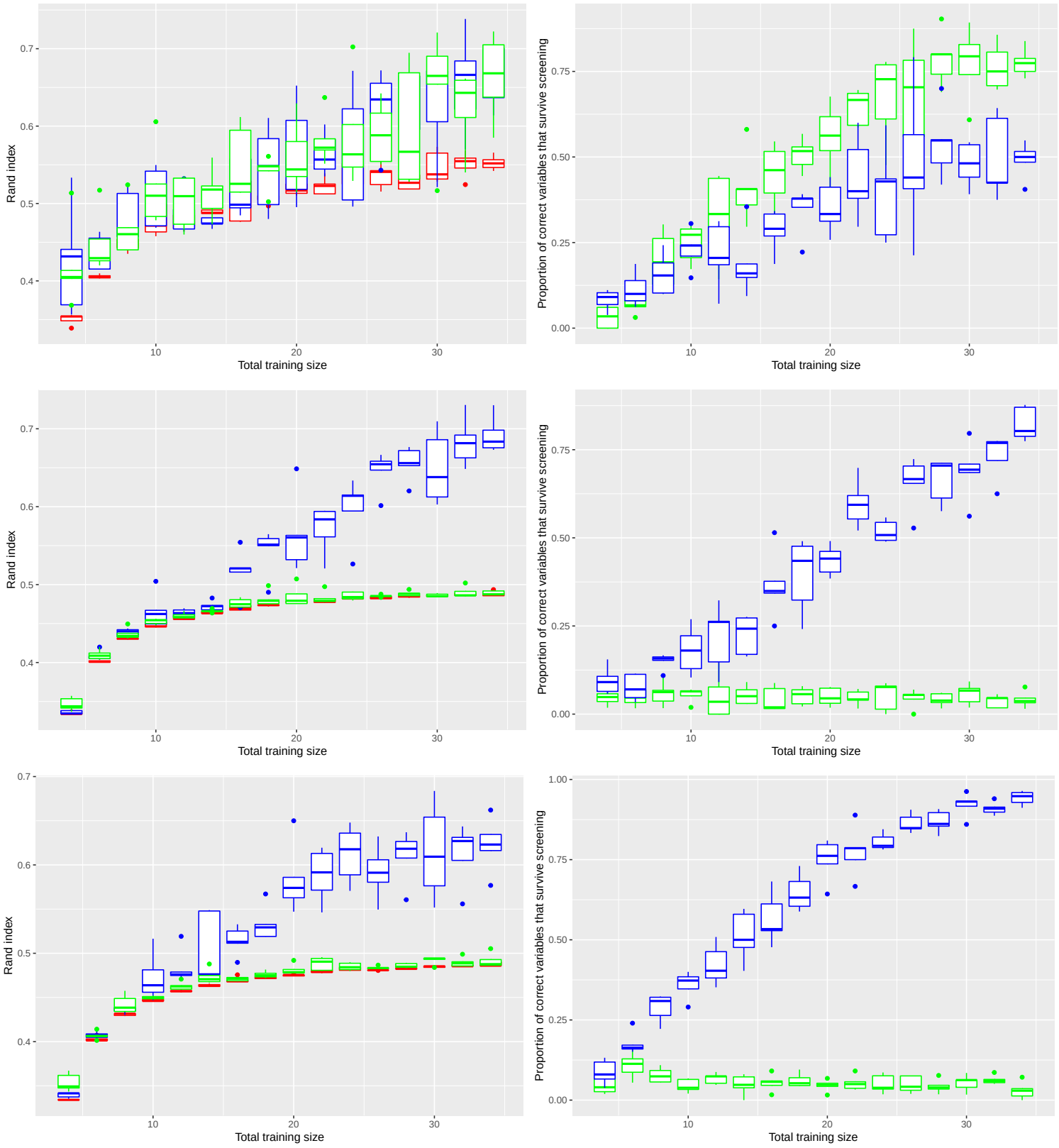



Figure 5.14: *Simulation results for $t$-test screening and ALB screening that uses A4.* Both the t-test and ALB screening are performed so that the type I error rate of each test is .05. $D$ is set to 2 and $B$ is set to 1000. Red, green and blue box plots are for no screening, $t$-test screening, and ALB screening, respectively. The first, second and third row of plots correspond to cases 1, 2 and 3, respectively.

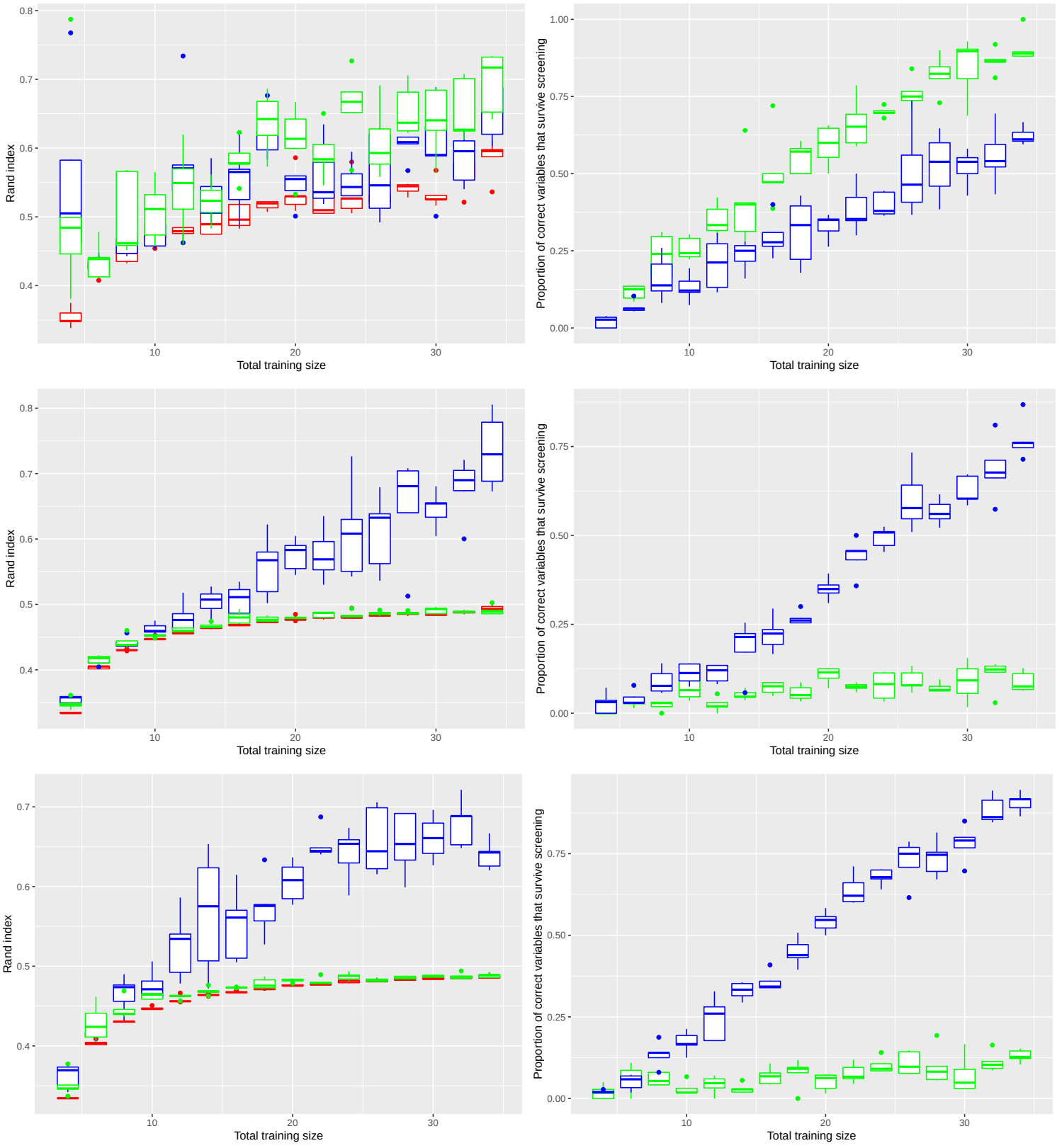



Figure 5.15: *Simulation results for $t$-test screening and ALB screening that uses variables with $n + m$ largest values of ALB.* The colors of the box plots have the same meaning as they do in Figure 5.14.

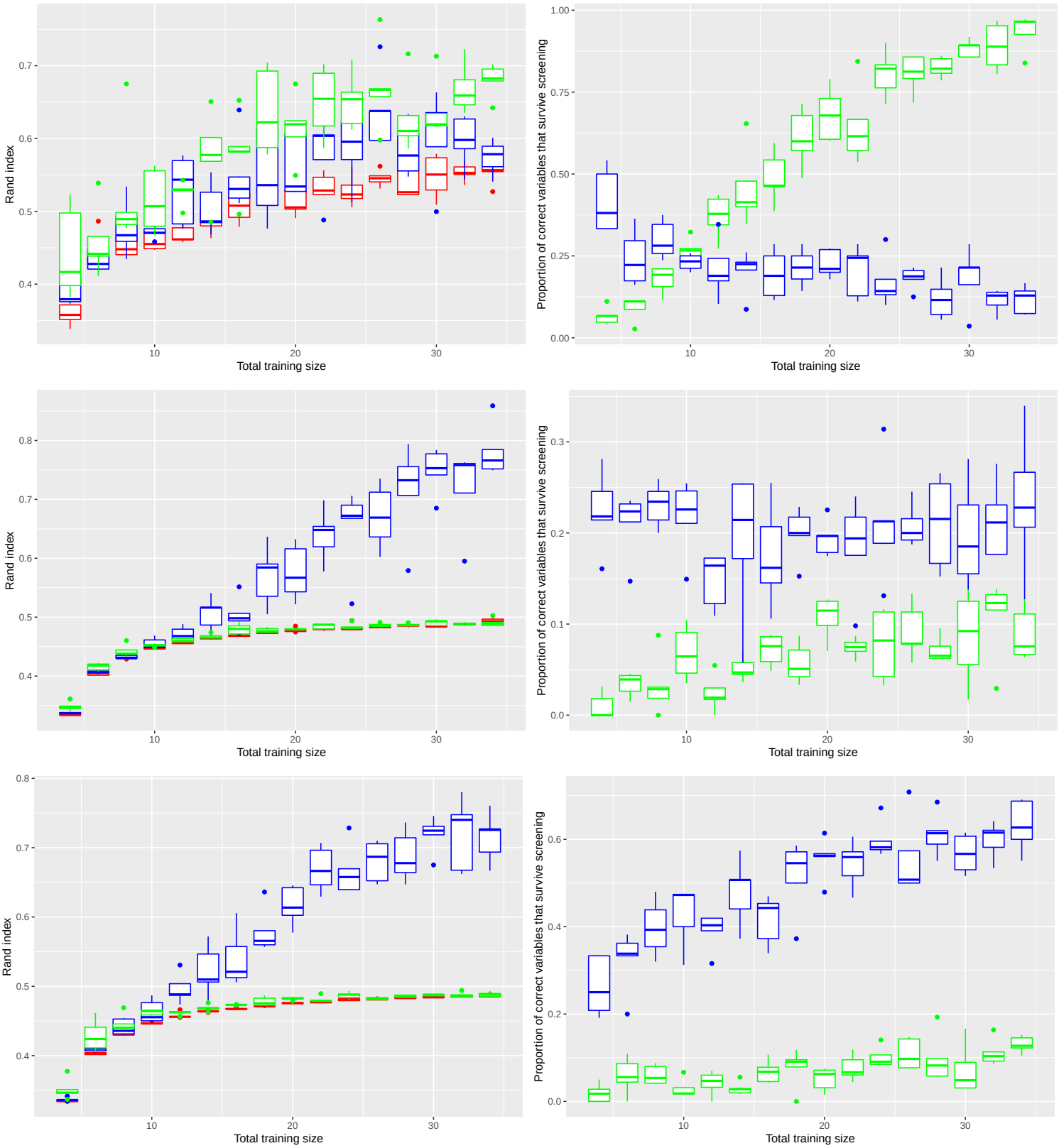



Figure 5.16: *Simulation results for t-test screening and ALB screening with cutoff* $\log(1.2)$. The colors of the box plots have the same meaning as they do in Figure 5.14.

| Rand Index | Screening Method | Number of variables chosen |
|---|---|---|
| 0.947 | $ALB > T^*_{0.005}$ | 1321 |
| 0.942 | $ALB > 0$ | 1946 |
| 0.750 | $ALB > \log(1.2)$ | 4 |
| 0.947 | $P_t < 0.005$ | 1540 |
| 0.935 | No screening | 4835 |

Table 5.1: *C*lassification and screening results for GISETTE data. All methods used a balanced training and testing set that both consisted of 3000 observations. The quantities $P_t$ and $T^*_{0.005}$ are, respectively, the $P$-value of a $t$-test and the 99.5th percentile of permuted $ALB$s.

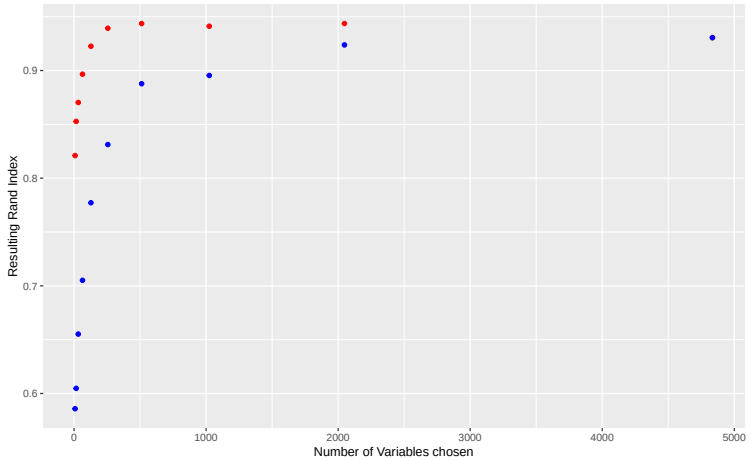



Figure 5.17: *Rand index for the GISETTE data as a function of number of variables used.* The Rand index is computed for a DART classifier using only those variables having the largest $2^j$ $t$-statistics or the largest $2^j$ values of $ALB$, where $j = 3, \ldots, 11$. The red points correspond to ALB screening and blue points to $t$-statistic screening.

the intensity of a colored-in pixel is larger than the intensity of a pixel that is rarely touched. We also believe this is the reason why $t$-test screening finds more important variables than does ALB screening based on the same type I error rate. Despite being a setting where mostly location differences exist, ALB ends up doing as well as $t$-test screening in terms of Rand index, at least when using a type I error rate of $0.005$. In general we believe that choosing a cutoff based on a type I error rate or using 0 as a cutoff is a good strategy for data sets where $n$ and $p$ are both large. A large amount of data allows us to choose fairly small significance thresholds while still maintaining

good power. Choosing variables that have the largest $n+m$ values of ALB in this case results in no variables being screened, and so we elect not to explore that avenue. A cross-validation procedure for selecting a cutoff is expensive to perform due to the large values of $p$ and $n+m$.

To explore the impact of choosing a cutoff based on quantiles, we explored the Rand indices when the cutoff corresponded to using variables with the largest $k$ statistics. We considered values of $k$ that increased geometrically: $k = 8, 16, \ldots, 2048$. The results can be seen in Figure 5.17. At each number of variables used, ALB-based screening has a larger Rand index than does $t$-test screening. Clearly, ALB and $t$-test screening are not choosing the same variables, and the ones chosen by ALB are more effective.

## 5.9 Application to Leukemia data

The Leukemia data set contains observations on 72 patients, 47 of which have one type of leukemia and the remainder another type. We have observations of 7129 variables on each patient to build a classifier that will help decide which type of leukemia a future patient has. This is a case where $p$ is much larger than $n$ and $m$. We will apply various screening methods to this problem in conjunction with DART and compare the accuracy of the methods via Rand indices.

To do this in a fair fashion, we split the data set randomly into two halves. The first half are validation data, and the second half are training data. We then split the training data in half again to make two smaller training sets. We do this for two reasons. First, we wish to assess the effect of training set size on accuracy of the methods. One of the two smaller training sets will be used to build classifiers, each one corresponding to a different screening method, and then all the training data will be used to build another set of classifiers. Both sets of classifiers will be used to predict the data in the validation set. The second reason for dividing the training set in half is that it makes possible a cross-validation approach for selecting a cutoff. We can train the model on one of the smaller training data sets, and choose a cutoff that gives the best classification accuracy on the other training data set. We can then train this best model on the full training set and apply it to the validation set. This is applying strategy A6 in the methodology, which is feasible because of the small sizes of $m$ and $n$.

| Rand Index | Screening Method Used | Number of variables |
|---|---|---|
| 0.599 | $n+m$ largest $ALB$s | 19 |
| 0.529 | $ALB > T^*_{0.05}$ | 617 |
| 0.549 | $ALB > \log(1.2)$ | 233 |
| 0.599 | $n+m$ largest $t$-statistics | 19 |
| 0.529 | $P_t < 0.05$ | 847 |
| 0.501 | No screening | 7129 |

Table 5.2: *Classification and screening results for leukemia data when training set is a quarter of full set.* All results are based on a validation set size that was roughly half that of the full data set. The quantities $P_t$ and $T^*_{0.05}$ are, respectively, the $P$-value of a $t$-test and the 95th percentile of permuted $ALB$s. The classifier used was DART.

To carry out our analysis we did the following. We performed four ALB based screening methods and two $t$-test based screening methods. The four ALB methods were A1 with the $n+m$ largest $ALB$s being selected, A4 with a type I error rate of $0.05$, $d = 3$ and $B = 7129$, and A5 with the interpretable cutoffs of $0$ and $\log(1.2)$.

Two methods of $t$-test screening were used, one using the variables with the $m+n$ largest $t$-statistics, and the other using variables whose $t$-test $P$-values were smaller than $0.05$. The latter version of $t$-test screening makes it comparable to choosing a variable using method A4 with significance level $0.05$. Once we determined relevant variables via screening, DART based on those variables was used to compute a Rand index from the validation set. Tables 5.2 and 5.3 summarize the results.

All screening methods performed similarly when the training set size was a quarter of $m+n$. However, $ALB$ screening that chose a cutoff as in $A1$ or $A4$ improved remarkably when the sample sizes were doubled, faring much better than the $t$-test based screening methods. While DART has been shown to be an effective classifier, our experience is that it may fail to recover the structure of the classification problem when the sample size is small. We thus tried the Bayesian classifier based on (5.5) with the CV-based method to choose a cutoff, and this classifier was able to achieve decent classification accuracy, surpassing DART if a different cutoff is chosen. We believe this is the case due to its simplicity, and there should be enough data to construct reasonable kernel

| Rand Index | Screening Method | Number of variables |
|---|---|---|
| 0.742 | $ALB > T_{\text{CV}}$ | 12 |
| 0.834 | $ALB > \log(1.2)$ | 150 |
| 0.834 | $ALB > T_{\text{CV}}$ | 12 |
| 0.786 | $n + m$ largest $ALB$s | 38 |
| 0.572 | $ALB > T^*_{0.05}$ | 1302 |
| 0.742 | $ALB > \log(1.2)$ | 150 |
| 0.598 | $n + m$ largest $t$-statistics | 38 |
| 0.572 | $P_t < .05$ | 1694 |
| 0.549 | No screening | 7129 |

Table 5.3: *Classification and screening results for leukemia data when training set is half of full set.* All results are based on a validation set size that was roughly half that of the full data set. The first two rows of the table correspond to use of the classifier based on (5.5), and subsequent rows to use of DART. The quantity $T_{CV}$ is the best cutoff as chosen by cross-validation, and $P_t$ and $T^*_{0.05}$ are as in Table 2. See Section 5.9 for an explanation of how cross-validation was implemented.

density estimates of the underlying distributions.

Figure 5.18 shows the cross-validated Rand indices of the Bayesian classifier as a function of cutoff. It turns out that the largest cutoff maximizing the Rand index was .288. (The largest cutoff was chosen since this corresponds to the smallest number of variables maximizing the Rand index.) Figure 5.19 shows Rand indices of the Bayesian classifier that was trained on the full training set (i.e., the training set using half of the full data set). This figure shows how well the CV cutoff fared when it was used to screen variables in the full training set. Figure 5.18 shows how the number of variables chosen is related to cutoff. Finally, Figure 5.19 also shows how sensitive the Rand index is to the selected cutoff.

We believe the Bayesian classifier has potential in other settings where there may not be sufficient data to train ensemble methods and there may exist differences between classes that are not of location type. A potential problem of this method is its sensitivity to variables that are not useful, but in our experience this can be a problem even amongst classification methods that are capable of eventually tuning out irrelevant predictors. It is encouraging that the interpretable cutoff of $\log(1.2)$ resulted in close to the best Rand index for the Bayesian classifier, but somewhat

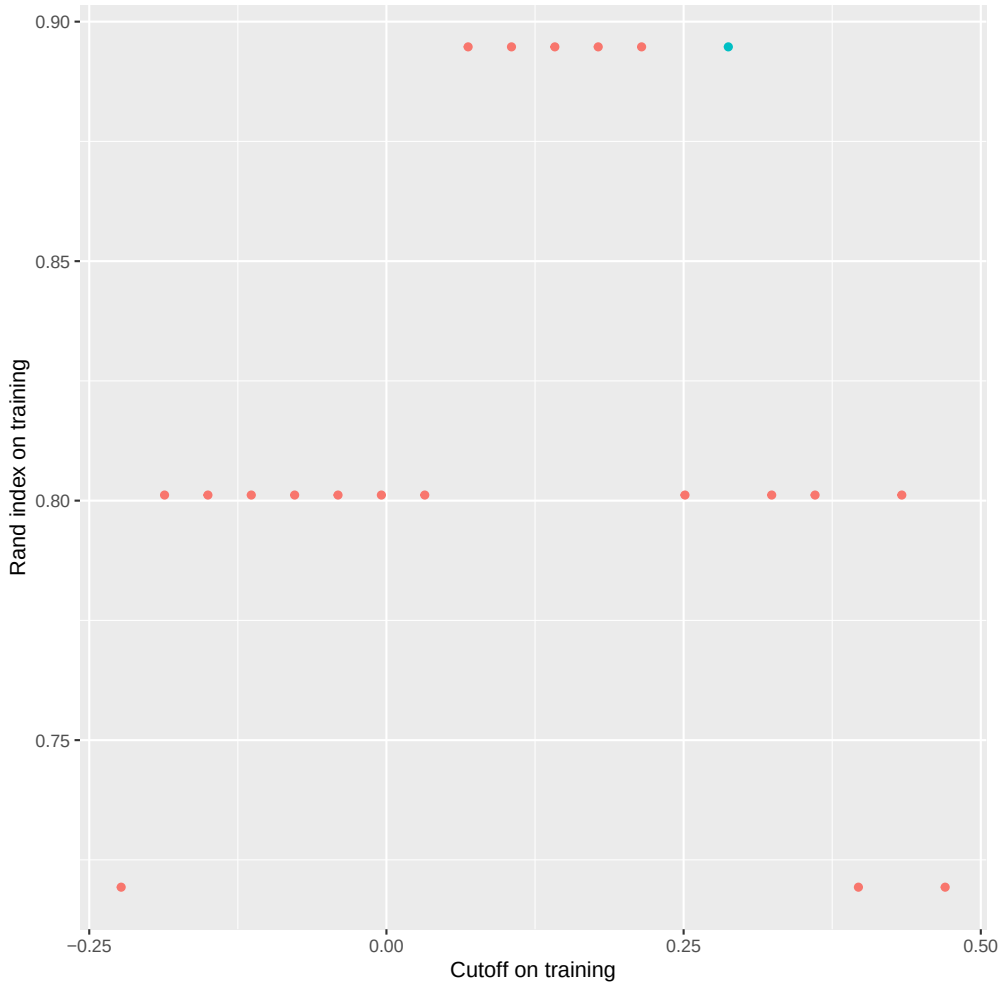



Figure 5.18: *Cross-validation performance of the Bayesian classifier on the training set of the leukemia data.* A plot of the rand index of the method referred to in 5.5 against the number of variables chosen by the method. The rand indices are the performance of the classifier on one of the training sets, applied to the other training sets. We chose the cutoff that works best by picking the cutoff which preserves the fewest variables in a sequence that maximized the rand index. That cutoff is in blue.

discouraging that the cross-validation approach could not pick a cutoff that resulted in the best performance for that classifier. The problem here is that the cross-validation approach chose an optimal cutoff when the classifier was constructed from one quarter of all the data, whereas we actually needed to know the optimal cutoff when half of all the data were used. Future research can focus on how an optimal cutoff depends on training set size. If the dependence is simple enough, it may be possible to estimate the optimal cutoff for a given training set size from cross-validation

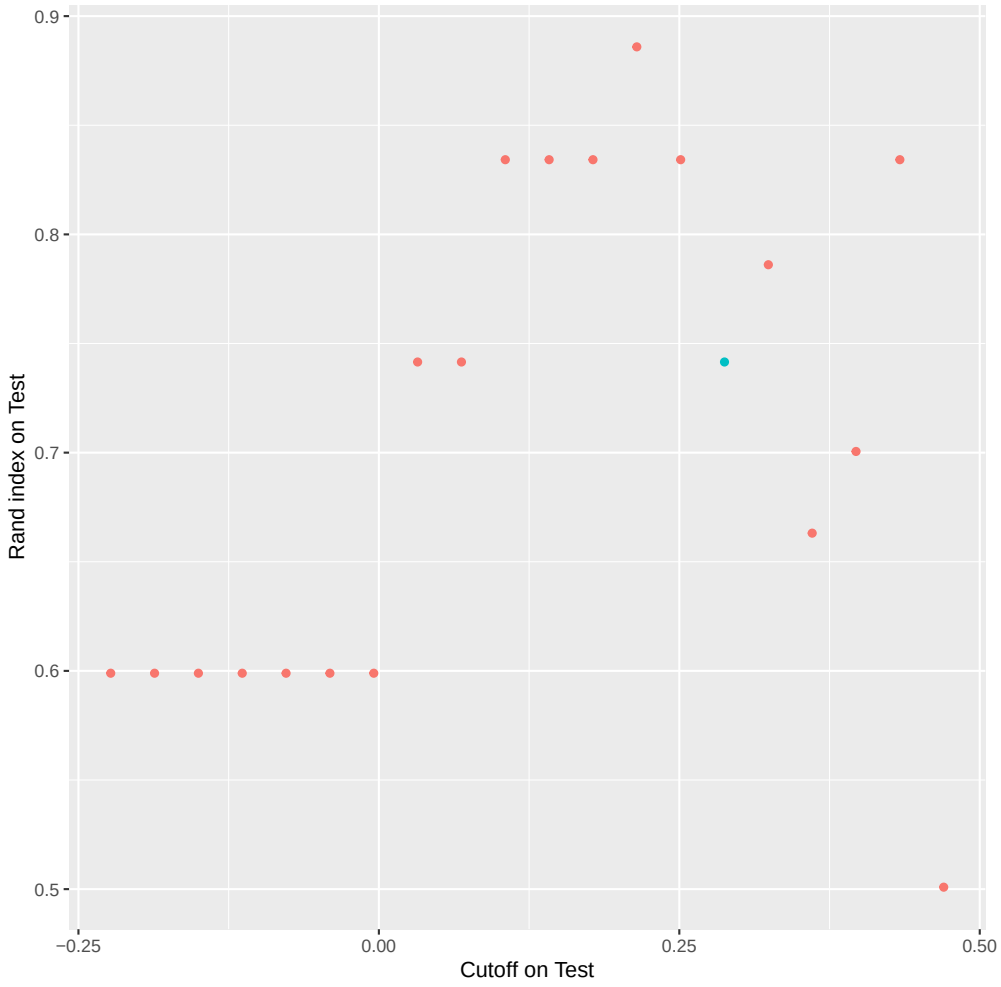



Figure 5.19: *Cross-validation performance of the Bayesian classifier on the testing set of the leukemia data set.* The rand indices are on the validation set. The cutoff we chose for cross validation is in blue and was selected by examining Figure 5.18.

results based on a smaller training set size.

ALB screening tends to do better when using the $k$ variables having the $k$ largest $ALB$s. Figure 5.21 shows the Rand index resulting from applying DART after using this method of screening. When using fewer than 500 variables, ALB screening does a better job than the analogous way of performing $t$-test screening. After the number of variables included is large enough, $t$-test screening does better than ALB screening, but at this point the number of variables included is large enough that the Rand index becomes suboptimal for both types of screening.

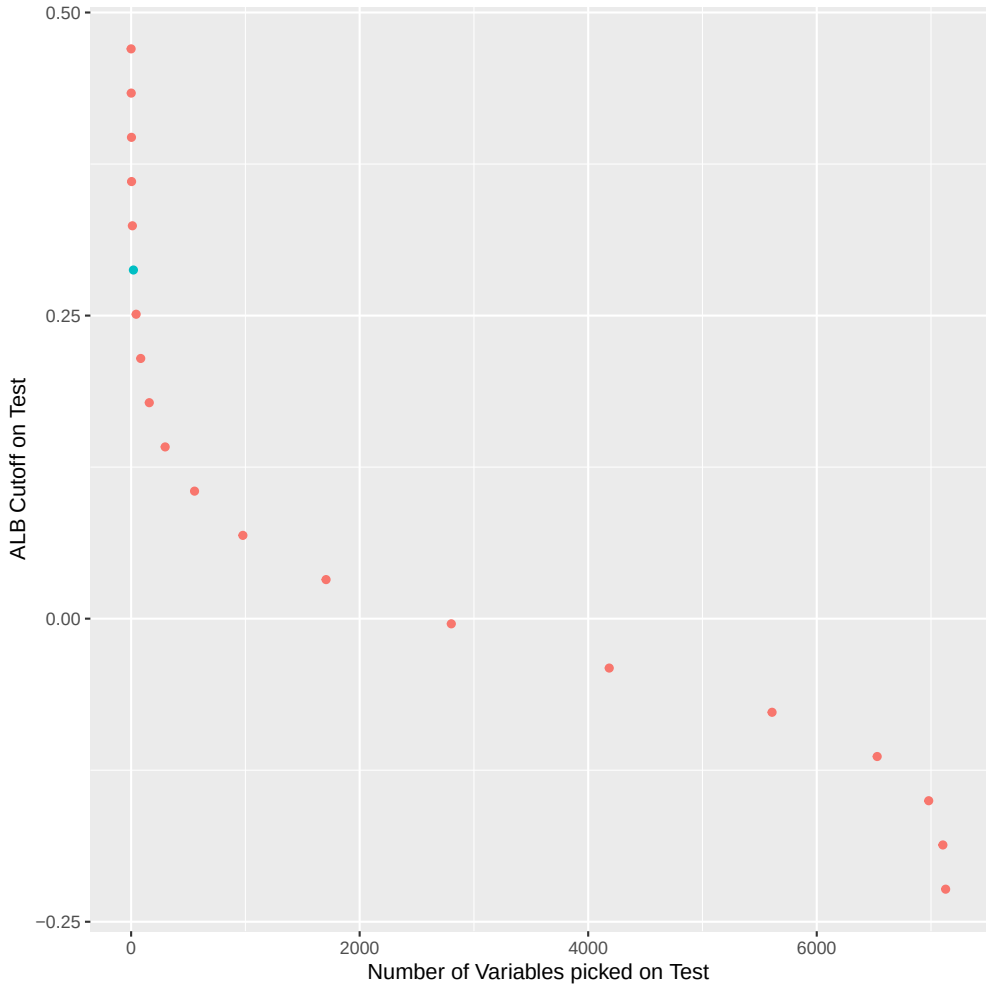



Figure 5.20: *Number of variables picked against the cutoff chosen using the ALB screening method on the Leukemia test data set.*

## 5.10 Conclusion and future work

We have proposed a new screening method that searches for differences other than those of location type. For this method to be more effective than $t$-test screening it needs to be paired with classification methods that can leverage these differences. In simulations, we pair ALB screening with BART, DART and a Bayesian classifier and show that it performs better than $t$-test screening in situations where class differences are not of location type. The Bayesian classier outperformed DART when applied to a leukemia data set. Even if the data contain primarily location differences, ALB screening performs well, although, as expected, not as well as $t$-test screening.

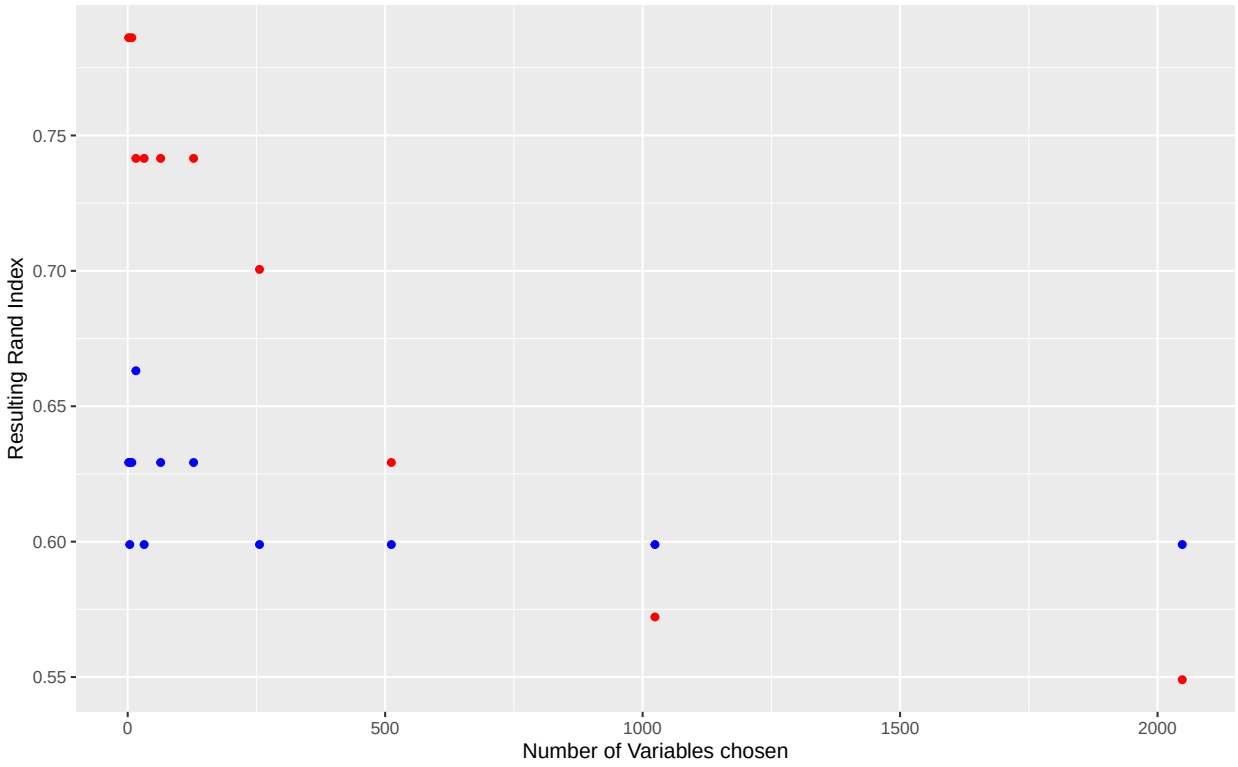



Figure 5.21: *DART rand indices against number of variables chosen for the Leukemia data set.* We plot the rand index of the DART methods where we choose relevant variables corresponding to the top number of ALBs or t-test statistics. We vary the number of variables we choose. The red points denote the rand index of DART models using the top number of ALBs, while the blue points denote the rand index of DART models using the top number of t-statistics.

Future work includes efforts to increase the speed of computing ALB statistics and their permutation distributions, especially for large data sets. An iterative approach to the screening method is available for SIS, and future research could involve investigating an ALB procedure that could capture differences in joint distributions. The simulated data in this paper all leveraged independent data, and how sensitive the method is to independence is also a property to explore. The interaction between ALB screening and random projection or sketching methods of dealing with settings where $n$ and/or $p$ are very large is also a promising direction for future research.

# 6. CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

A new framework for testing the problem if two samples share the same distribution using cross-validation Bayes factors is shown in Chapter 3. Chapter 4 extends this methodology by using ideas in permutation tests, creating a faster and more powerful test at the price of loosing some interpretability of the Bayes factor. Chapter 5 uses the test for screening and compares it to t-test screening, showing the method is as competitive for the problem of detecting if digits are 4s or 9s on the GISETTE dataset, a case where most variables probably differ by differences due to location. Chapter 5 also proposes a simple classifier that gives the probability an observation belongs to a class, and it does as well as BART and DART on the Leukemia data set.

## 6.2 Future Work

There are a few immediate direction of future research. The first would be to investigate performance of the new classifier in Chapter 5 in general, and how it performs on other data sets. The form of this classifier can be seen in equation 6.1. A large weakness of the method is its inability to leverage how variables may jointly differ, so patching up this weakness is also a research problem. This problem can be addressed by using a multivariate kernel density estimate rather than a product of univariate kernel density estimates, but it is not clear if this is an appealing idea immediately.

$$P(y=1|x) = \frac{\frac{n}{m+n}\prod_{i=1}^{p}\hat{f}^i(x_i|X,h)}{\frac{n}{m+n}\prod_{i=1}^{p}\hat{f}^i(x_i|X,h) + \frac{m}{m+n}\prod_{i=1}^{p}\hat{g^i}(x_i|X,h)} \tag{6.1}$$

$$P(y=1|x) = \frac{\frac{n}{m+n}\hat{f}(x|X,h)}{\frac{n}{m+n}\hat{f}(x|X,h) + \frac{m}{m+n}\hat{g}(x|X,h)} \tag{6.2}$$

Substituting a multivariate kernel density estimate with a product of univariate kernel density estimate allows the density to detect joint differences, but choosing the bandwidth for this new kernel density estimate is not trivial. In addition, kernel density estimates degrade fairly quickly

when dimension increases, and this sort of strategy may only be ideal when the dimension size is small. The form of this classifier appears in equation 6.2.

A compromise to this strategy is to partition the variables into sets where variables in the sets jointly vary with each other and are independent otherwise. We can then fit multivariate kernel density estimates onto each of the variables on the set and take the product of those instead. This alleviates the problem of kernel density estimates degrading significantly when the dimension increases, and potentially extends it to high dimensions. Of course, finding out which variables are important, which are important jointly, and which variables vary jointly with each other, is not a trivial task and suggests that this sort of procedure would best be paired with a more complicated variable selection method to proceed.

Another direction of future research lies in constructing an iterative method to extent the ALB screening procedure in Chapter 5 to capture joint differences. The simplest way to implement this sort of procedure would be pair the method with a classification procedure. We first screen for important variables, apply the classification method, and then see if classification errors occur with different distributions under some of the variables that were not originally picked. If a variable does, then that variable jointly offers information for classification with some other variable. We can repeat this procedure until the misclassification rate is small enough. Research involves concocting a suitable classification method, how the classification method affects this procedures, specifically when to stop the iterative procedure, and how effective this procedure is in detecting different types of joint variability among covariates.

ALB screening is a method for screening when the response variable is categorical and the other variables are numeric. The situation is well suited for two-sample testing, but an extension of this to the case where the response variable is categorical is also of interest. T-test screening turns into correlation based screening, so it is of interest to adapt the current procedure to one that fits the numeric setting instead. Correlation based screening, like t-test based screening, is sensitive to detecting location based joint differences, so we would like to develop another procedure that can detect differences outside of location based joint differences.

Finally, the ALB screening method itself is quite slow compared to t-test screening. Chapter 5 uses a plug-in bandwidth which results in a significant speed up compared to performing likelihood cross validation to select a bandwidth. Suppose $n$ is the number of observations we have, then the matter of computing the ALB itself, still requires computing roughly $n^2$ kernel computations, which is much slower than something like t-test screening which requires a linear number of operations with respect to $n$. While this is not a problem for the sake of screening, where $p$ is of greater concern than $n$, we still imagine this to become an issue if the data set itself is of too large a size. The final area of research lies in speedily or approximately computing the ALB.

REFERENCES


[1] C. C. Holmes, F. Caron, J. E. Griffin, D. A. Stephens, *et al.*, "Two-sample Bayesian Nonparametric Hypothesis Testing," *Bayesian Analysis*, vol. 10, no. 2, pp. 297–320, 2015.

[2] T. Sellke, M. Bayarri, and J. O. Berger, "Calibration of $\rho$ values for testing precise null hypotheses," *The American Statistician*, vol. 55, no. 1, pp. 62–71, 2001.

[3] J. Kennedy and R. Eberhart, "Particle Swarm Optimization (pso)," in *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, pp. 1942–1948, 1995.

[4] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[5] H. Zou and T. Hastie, "Regularization and Variable Selection via the Elastic Net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[6] E. I. George and R. E. McCulloch, "Approaches for Bayesian Variable Selection," *Statistica sinica*, pp. 339–373, 1997.

[7] A. R. Linero and Y. Yang, "Bayesian regression tree ensembles that adapt to smoothness and sparsity," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 80, no. 5, pp. 1087–1110, 2018.

[8] Y. Yang, M. J. Wainwright, M. I. Jordan, *et al.*, "On the Computational Complexity of High-Dimensional Bayesian Variable Selection," *The Annals of Statistics*, vol. 44, no. 6, pp. 2497–2532, 2016.

[9] J. Hill, A. Linero, and J. Murray, "Bayesian Additive Regression Trees: A Review and Look Forward," *Annual Review of Statistics and Its Application*, vol. 7, 2020.

[10] A. Kapelner and J. Bleich, "bartmachine: Machine Learning with Bayesian Additive Regression Trees," *arXiv preprint arXiv:1312.2171*, 2013.

[11] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5,

pp. 849–911, 2008.

[12] L.-P. Zhu, L. Li, R. Li, and L.-X. Zhu, "Model-free feature screening for ultrahigh-dimensional data," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1464–1475, 2011.

[13] Q. Mai and H. Zou, "The kolmogorov filter for variable screening in high-dimensional binary classification," *Biometrika*, vol. 100, no. 1, pp. 229–234, 2012.

[14] H. Cui, R. Li, and W. Zhong, "Model-Free Feature Screening for Ultrahigh Dimensional Discriminant Analysis," *Journal of the American Statistical Association*, vol. 110, no. 510, pp. 630–641, 2015.

[15] J. Xue and F. Liang, "A Robust Model-Free Feature Screening Method for Ultrahigh-Dimensional Data," *Journal of Computational and Graphical Statistics*, vol. 26, no. 4, pp. 803–813, 2017.

[16] L. Ma and W. H. Wong, "Coupling Optional Pólya Trees and the Two Sample Problem," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1553–1565, 2011.

[17] J. D. Hart and T. Choi, "Nonparametric goodness of fit via cross-validation Bayes factors," *Bayesian Analysis*, vol. 12, pp. 653–677, 2017.

[18] J. D. Hart and M. Malloure, "Prior-free Bayes factors based on data splitting," *International Statistical Review*, vol. 87, pp. 419–442, 2019.

[19] T. E. Hanson, "Inference for mixtures of finite Pólya tree models," *Journal of the American Statistical Association*, vol. 101, pp. 1548–1565, 2006.

[20] W. H. Wong, L. Ma, *et al.*, "Optional Pólya tree and Bayesian inference," *The Annals of Statistics*, vol. 38, pp. 1433–1459, 2010.

[21] Y. Chen and T. E. Hanson, "Bayesian nonparametric $k$-sample tests for censored and uncensored data," *Computational Statistics & Data Analysis*, vol. 71, pp. 335–346, 2014.

[22] D. B. Dunson and S. D. Peddada, "Bayesian nonparametric inference on stochastic ordering," *Biometrika*, vol. 95, pp. 859–874, 2008.

[23] J. Beirlant, E. J. Dudewicz, L. Györfi, and I. Dénes, "Nonparametric entropy estimation: An

overview," *International Journal of Mathematical and Statistical Sciences*, vol. 6, pp. 17–39, 1997.

[24] S. D. Sarkar and S. Goswami, "Empirical study on filter based feature selection methods for text classification," *International Journal of Computer Applications*, vol. 81, 2013.

[25] P. Hall, "On Kullback-Leibler loss and density estimation," *Annals of Statistics*, vol. 15, pp. 1491–1519, 1987.

[26] L. Györfi, L. Devroye, and L. Gyorfi, *Nonparametric density estimation: the L1 view*. New York; Chichester: John Wiley & Sons, 1985.

[27] J. Berger and L. Pericchi, "The intrinsic Bayes factor for model selection and prediction," *Annals of Statistics*, vol. 91, no. 2, pp. 109–122, 1996.

[28] B. Rüger, *Test-und Schätztheorie: Band I: Grundlagen*. Oldenbourg: De Gruyter, 1998.

[29] E. Schuster and G. Gregory, "On the nonconsistency of maximum likelihood nonparametric density estimators," *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface, ed. Eddy, W.*, pp. 295–298, 1981.

[30] G. Consonni, D. Fouskakis, B. Liseo, I. Ntzoufras, *et al.*, "Prior distributions for objective Bayesian analysis," *Bayesian Analysis*, vol. 13, pp. 627–679, 2018.

[31] R. McVinish, J. Rousseau, and K. Mengersen, "Bayesian goodness of fit testing with mixtures of triangular distributions," *Scandinavian Journal of Statistics*, vol. 36, pp. 337–354, 2009.

[32] M. van der Laan, S. Dudoit, and S. Keleş, "Asymptotic optimality of likelihood-based cross-validation," *Statistical Applications in Genetics and Molecular Biology*, vol. 3, p. online publication, 2004.

[33] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, pp. 773–795, 1995.

[34] J. D. Hart, "Use of BayesSim and smoothing to enhance simulation studies," *Open Journal of Statistics*, vol. 7, pp. 153–172, 2017.

[35] C. Kraft, Y. Lepage, and C. Van Eeden, "Estimation of a symmetric density function," *Communications in Statistics–Theory and Methods*, vol. 14, pp. 273–288, 1985.

[36] A. Cowling and P. Hall, "On pseudodata methods for removing boundary effects in kernel density estimation," *Journal of the Royal Statistical Society B*, vol. 58, pp. 551–563, 1996.

[37] Z. Bai, C. Rao, and L. Zhao, "Kernel estimators of density function of directional data," *Journal of Multivariate Analysis*, vol. 27, pp. 24–39, 1988.

[38] N. Merchant, J. Hart, and T. Choi, "Use of cross-validation Bayes factors to test equality of two densities," *Bayesian Analysis,*, vol. submitted, 2020.

[39] A. W. Bowman and A. Azzalini, *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, vol. 18. OUP Oxford, 1997.

[40] R. Baranzano, "Non-parametric kernel density estimation-based permutation test: Implementation and comparisons.," 2011.

[41] J. D. Hart, T. Choi, and S. Yi, "Frequentist nonparametric goodness-of-fit tests via marginal likelihood ratios," *Computational Statistics & Data Analysis*, vol. 96, pp. 120–132, 2016.

[42] S. G. Young and A. W. Bowman, "Non-parametric analysis of covariance," *Biometrics*, pp. 920–931, 1995.

[43] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

[44] L. Brieman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," *Wadsworth Inc*, vol. 67, 1984.

[45] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[46] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data classification: Algorithms and applications*, p. 37, 2014.

[47] J. Fan, R. Song, *et al.*, "Sure independence screening in generalized linear models with np-dimensionality," *The Annals of Statistics*, vol. 38, no. 6, pp. 3567–3604, 2010.

[48] N. Merchant and J. Hart, "A bayesian motivated two-sample test based on kernel density estimates,"

APPENDIX A

# USE OF CROSS-VALIDATION BAYES FACTORS TO TEST EQUALITY OF TWO DENSITIES APPENDIX

Here we derive the Hessian used in our Laplace approximation, provide a link to an R package that can compute cross-validation Bayes factors as well as Pólya tree Bayes factors, provide additional simulation results, give a heuristic justification for our Laplace approximation and prove Bayes consistency of a $CVBF$.

## A.1 Hessian derivation

Here we derive $\widehat{H}$, as defined in Section 3.2. Let $\hat{f}_h$ be a KDE based on data $Z_1, \ldots, Z_k$ and kernel $K$, and for arbitrary scalar quantities $u_1, \ldots, u_\ell$ define $L_1$ as follows:

$$L_1(h) = \prod_{j=1}^{\ell} \hat{f}_h(u_j).$$

Then $L_1$ has the same structure as $L_0$ in Section 3.2, and it suffices to consider

$$\frac{\partial^2}{\partial h^2} \log L_1(h) = \sum_{j=1}^{\ell} \left[ \hat{f}_h(u_j) \frac{\partial^2}{\partial h^2} \hat{f}_h(u_j) - \left( \frac{\partial}{\partial h} \hat{f}_h(u_j) \right)^2 \right] / \hat{f}_h^2(u_j). \tag{A.1}$$

We have

$$\frac{\partial}{\partial h} \hat{f}_h(u_j) = -\frac{1}{h} \left[ \hat{f}_h(u_j) - \hat{e}_h(u_j) \right], \tag{A.2}$$

where $\hat{e}_h$ is a kernel estimator based on data $Z_1, \ldots, Z_k$ and kernel $J(u) = -uK'(u)$. Note that $\hat{e}_h$ is a "legitimate" kernel estimator in that

$$\int_{-\infty}^{\infty} J(u)\, du = 1 \qquad \text{and} \qquad \int_{-\infty}^{\infty} uJ(u)\, du = 0.$$

Now,

$$\frac{\partial^2}{\partial h^2}\hat{f}_h(u_j) = -\frac{1}{h}\left[2\frac{\partial}{\partial h}\hat{f}_h(u_j) - \frac{\partial}{\partial h}\hat{e}_h(u_j)\right], \tag{A.3}$$

and

$$\frac{\partial}{\partial h}\hat{e}_h(u_j) = -\frac{1}{h}\left[\hat{e}_h(u_j) - \hat{g}_h(u_j)\right],$$

where $\hat{g}_h$ is a kernel estimator based on data $Z_1, \ldots, Z_k$ and kernel $L(u) = -uJ'(u)$. As before, $\hat{g}_h$ is a legitimate, i.e., consistent, density estimator. Substitution of (A.2) and (A.3) into (A.1) leads to a readily computable expression for $\widehat{H}$.

### A.2 R package

An R package that implements cross-validation Bayes factors and also Holmes's Pólya tree Bayes factors is available on github.

### A.3 More detailed simulation results

The figures in the paper provide smoothed curves of test statistics in each of four cases. Here we plot the smooth curves and also the log-Bayes factors. For reference we explored the following four cases:

*Scale change:* The densities $f$ and $g$ are $\phi$ (standard normal) and $\phi(x/2)/2$, respectively, and hence differ with respect to scale.

*Location shift:* The densities $f$ and $g$ are standard Cauchy, $f_C$, and $f_C(x+1)$, respectively, and so differ with respect to location.

*Distributions with different tail behavior:* Here $f$ and $g$ are $f_C$ and $0.6745\phi(0.6745x)$, respectively. Given $p$, the mixture density in this case has the same median and interquartile range as the standard Cauchy, and so the densities of the $X$ and $Y$ samples are different but have the same location and scale.

*Different distributions with same finite support:* The densities $f$ and $g$ are $U(0,1)$ (uniform on the interval $(0,1)$) and beta$(1/2, 1/2)$, respectively.

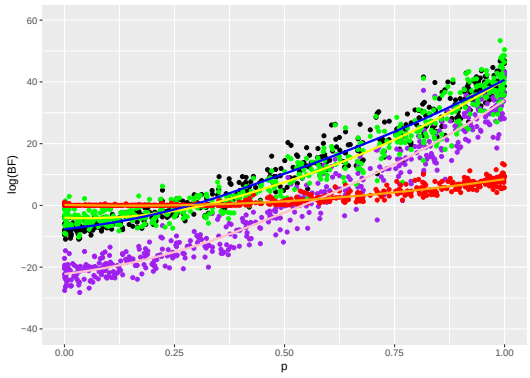



Figure A.1: Log-Bayes factors and their smooths for the scale shift case. The purple dots represent Pólya tree log-Bayes factors where a Cauchy is used for quantiles, and their smooth is pink. The green dots represent Pólya tree log-Bayes factors where a normal distribution is used for quantiles, and their smooth is yellow. The black points represent the averages of log-cross-validation Bayes factors across thirty splits, and their smooth is blue. Values of $\log B$ for the K-S test are in red, with their smooth being orange.

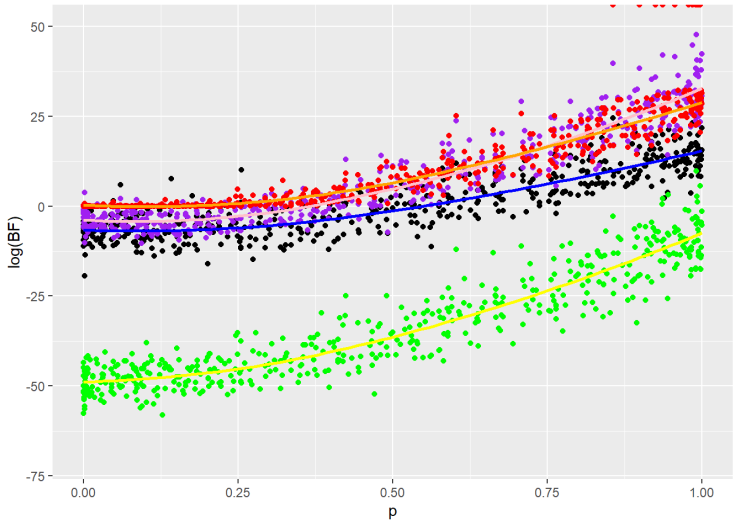



Figure A.2: Log-Bayes factors and their smooths for the location shift case. See Figure 1 for the color legend.

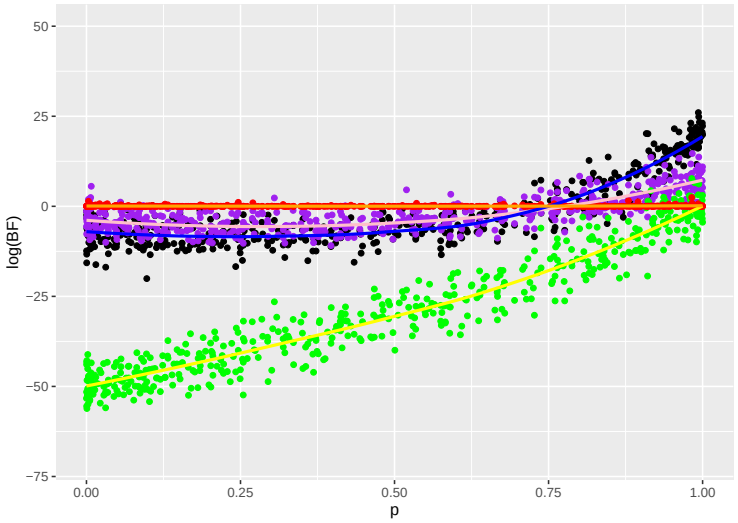



Figure A.3: Log-Bayes factors and their smooths for the tail difference case. See Figure 1 for the color legend.

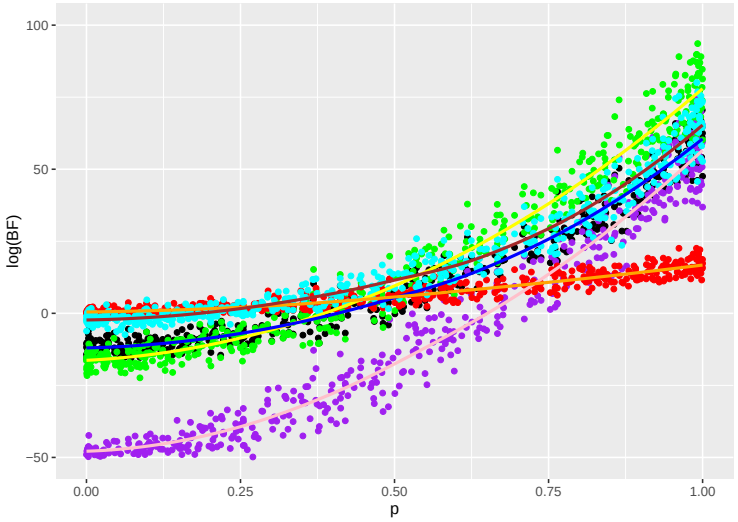



Figure A.4: Log-Bayes factors and their smooths for the finite support case. The cyan dots are log-cross-validation Bayes factors based on data-reflected KDEs, and their smooth is brown. See Figure 1 for the rest of the color legend.

## A.4 Heuristics for Laplace approximation

One of the assumptions for our theorem is that the Laplace approximation of a marginal likelihood is asymptotic to that marginal likelihood. We cannot give explicit conditions under which this assumption holds, but we can provide some heuristics of a proof. The classic Laplace approximation in a likelihood setting assumes that the maximizer of the likelihood does not occur on the boundary of the parameter space. A main technical difficulty of justifying the Laplace approximation in our setting is that the asymptotic maximizer of the likelihood becomes ever closer to a boundary, namely 0. However, by making an appropriate change of variable in the integral defining the marginal likelihood, we can effectively sidestep this difficulty.

We assume that the priors are defined as in Section 3.1, in which case the maximizer of, for example, $\pi_X(\alpha)L_X(\alpha)$ with respect to $\alpha$ is $\hat{\gamma}$, the maximizer of $L_X(\alpha)$. Using results from Hall (1987) and under very general conditions, $\hat{\gamma}$ is asymptotic (in probability) to a quantity of the form $Cn^{-b}$, where $C$ is a positive constant and $0 < b < 1$. Making the change of variable $n^b\alpha = u$ in the $X$-sample marginal likelihood, we have

$$\int_0^\infty \pi_X(\alpha)L_X(\alpha)\,d\alpha = n^{-b}\int_0^\infty \pi_X(n^{-b}u)L_X(n^{-b}u)\,du \stackrel{\text{def}}{=} n^{-b}I_n.$$

It is now plausible that a Laplace approximation, $\mathcal{L}_n$, of $I_n$ is asymptotic to $I_n$. This is because the maximizer, with respect to $u$, of the integrand of $I_n$ is $n^b\hat{\gamma}$, which converges in probability to $C$ as $n \to \infty$ and $C$ is *not* a boundary point. Our heuristic proof is complete by then noting that $n^{-b}\mathcal{L}_n$ is exactly equal to our Laplace approximation of the marginal likelihood.

## A.5 Consistency proof

For reference we list the assumptions needed for our test to be Bayes consistent.

The Hall kernel and our bandwidth prior will be used frequently, and so they are defined here for convenience:

$$K_0(z) = \frac{1}{\sqrt{8\pi e}\,\Phi(1)}\exp\left[-\frac{1}{2}(\log(1+|z|))^2\right], \tag{A.4}$$

$$\pi(h|\gamma) = \frac{2\gamma}{\sqrt{\pi}h^2} \exp\left(-\frac{\gamma^2}{h^2}\right) I_{(0,\infty)}(h). \tag{A.5}$$

A1. The Laplace approximation of each of the three marginals is asymptotically correct in that the log of the marginal likelihood is equal to the log of the Laplace approximation plus a term that is negligible in probability relative to the approximation.

A2. The densities $f$ and $g$ are bounded away from 0 and $\infty$ on $(-\lambda, \lambda)$ for each $\lambda > 0$, with $f(x) \sim c_1 x^{-a_1}$ and $f(-x) \sim c_2 x^{-a_2}$ as $x \to \infty$, where both $c_1$ and $c_2$ are positive and $a_1$ and $a_2$ larger than 1. Density $g$ satisfies the same properties as $f$, albeit with possibly different constants.

A3. The second derivatives $f''$ and $g''$ exist and are bounded and almost everywhere continuous on $(-\infty, \infty)$. In addition, for a constant $C_2 < \infty$,

$$|f''(x)| \leq C_2 x^{-a_1-2} \quad \text{and} \quad |f''(-x)| \leq C_2 x^{-a_2-2} \quad \text{for } x > 1,$$

where $a_1$ and $a_2$ are the same as in A2. The function $g''$ satisfies the same properties as $f''$ with possibly different constants.

A4. The kernel used is $K_0$, as defined in (A.4).

A5. The prior is (A.5) and its parameter is chosen as described in Section 3.1 (of the paper).

**Theorem 3.** *Suppose that assumptions A1-A4 hold, and that $r$ and $s$ each tend to $\infty$ in such a way that $r = o(m)$, $r = o(n)$, $s = o(m)$ and $s = o(n)$ as $m$ and $n$ tend to $\infty$. Then if $f \equiv g$ and $m$ and $n$ tend to $\infty$*

$$\begin{aligned} \log(CVBF) &= C_f \left\{ (m-r)\left[\frac{1}{(r+s)^a} - \frac{1}{r^a}\right] + (n-s)\left[\frac{1}{(r+s)^a} - \frac{1}{s^a}\right] \right\} \\ &\quad + o_p\left(\frac{(m-r)}{r^a} + \frac{(n-s)}{s^a}\right), \end{aligned} \tag{A.6}$$

*where $C_f$ is a positive constant and $0 < a < 4/5$ is a constant determined by $f$. If instead $\int |f - g| > 0$, $r/(r+s) \sim m/(m+n)$, $m/(m+n) \to q$ as $m, n \to \infty$ and $0 < q < 1$, then as $m, n \to \infty$*

$$\begin{aligned} \log(CVBF) &= (m-r)KL(f, qf + (1-q)g) + (n-s)KL(g, qf + (1-q)g) \\ &\quad + o_p(m+n). \end{aligned} \tag{A.7}$$

**Proof:** Throughout the proof $C_0, C_1, \ldots$ will denote a sequence of positive constants, and until further notice it is assumed that $f \equiv g$. Recall that $CVBF$ is

$$\begin{aligned} CVBF &= \frac{\int_0^\infty \int_0^\infty \pi_X(\alpha)\pi_Y(\beta)L_a(\alpha,\beta)\, d\alpha d\beta}{\int_0^\infty \pi(h)L_0(h)\, dh} \\ &= \frac{\int_0^\infty \pi_X(\alpha)L_X(\alpha)\, d\alpha \cdot \int_0^\infty \pi_Y(\beta)L_Y(\beta)\, d\beta}{\int_0^\infty \pi(h)L_0(h)\, dh}. \end{aligned} \tag{A.8}$$

Define

$$\widehat{BF} = \frac{\pi_X(\hat{\alpha})L_X(\hat{\alpha})\pi_Y(\hat{\beta})L_Y(\hat{\beta})}{\pi(\hat{h})L_0(\hat{h})},$$

where $\hat{\alpha}$, $\hat{\beta}$ and $\hat{h}$ are the maximizers of $L_X$, $L_Y$ and $L_0$, respectively. By A1,

$$CVBF \sim \sqrt{\frac{2\pi\widehat{H}_0}{\widehat{H}_X\widehat{H}_Y}} \cdot \widehat{BF} \tag{A.9}$$

in probability, where $\widehat{H}_X$, $\widehat{H}_Y$ and $\widehat{H}_0$ are the Hessians for the $X$, $Y$ and combined samples, respectively.

Suppose that $\widehat{H}$ is the Hessian calculated when a KDE is computed from $\ell$ i.i.d. observations and an independent validation sample of size $k$, where $\ell = o(k)$. Then a consequence of results to be proven subsequently is that

$$\widehat{H} = kC_0\ell^{-a/2} + o_p(k\ell^{-a/2}), \tag{A.10}$$

where $0 < a < 4/5$ is a constant determined by the underlying density $f$. It follows that $\log\left(2\pi\widehat{H}_0/(\widehat{H}_X\widehat{H}_Y)\right)$ is, as will be seen subsequently, negligible (in probability) relative to $\log(\widehat{BF})$, and so going forward it suffices to analyze $\log\widehat{BF}$.

Now let $\boldsymbol{X}_T^i$ denote all the data in $\boldsymbol{X}_T$ except for $X_i$, and define

$$\ell(\alpha) = \prod_{i=1}^{r} \hat{f}(X_i|\alpha, \boldsymbol{X}_T^i). \tag{A.11}$$

For reference recall that

$$L_X(\alpha) = \prod_{i=r+1}^{m} \hat{f}(X_i|\alpha, \boldsymbol{X}_T). \tag{A.12}$$

For each $\alpha$, $\widehat{R}_r(\alpha) = -(m-r)^{-1}\log L_X(\alpha)$ is an unbiased estimator of the risk function

$$R_r(\alpha) = E[KL(f, \hat{f}(\,\cdot\,|\alpha, \boldsymbol{X}_T))] - \int f(x)\log f(x)\,dx.$$

On the other hand, $\widetilde{R}_r(\alpha) = -r^{-1}\log\ell(\alpha)$ is only asymptotcally unbiased for $R_r(\alpha)$. Nonetheless, (25) proves that, as $r \to \infty$, the maximizer of $\ell(\alpha)$ is asymptotic in probability to $\alpha_r$, the minimizer of $R_r$. An examination of the proof of (25) reveals that $\hat{\alpha}$, the maximizer of $L_X$, is also asymptotic to $\alpha_r$. Indeed, $\widehat{R}_r$ is a more efficient estimator of $R_r$ than is $\widetilde{R}_r$, which is intuitively plausible owing to the facts (a) the KDE $\hat{f}(\,\cdot\,|\alpha, \boldsymbol{X}_T)$ is completely independent of the validation data, and (b) the estimator $\widehat{R}_r(\alpha)$ is an average of $m-r$ rather than $r$ random variables (and $r = o(m-r)$).

We may write

$$\log\widehat{BF} = \log LR + \log(\pi_X(\hat{\alpha})) + \log(\pi_Y(\hat{\beta})) - \log(\pi(\hat{h})),$$

where $LR = L_X(\hat{\alpha})L_Y(\hat{\beta})/L_0(\hat{h})$. Using A5,

$$\log\pi_X(\hat{\alpha}) = \log(2/\sqrt{\pi}) - \log(\hat{\alpha}) - 1 = O_p(\log m),$$

with the last equality due to the facts that $\hat{\alpha} \sim \alpha_r$ in probability and the optimal bandwidth $\alpha_r$ is of

order $r^{-a/4}$ (with $a$ the same as in expression (A.10)). As will be seen subsequently, this implies that the impact of $\log \pi_X(\hat{\alpha})$ on $\log \widehat{BF}$ is negliglible. Similarly the impact of the other two prior terms is negligible, and it suffices to investigate $\log LR$.

To simplify notation, we define the following quantities:

$$\hat{f}_X \equiv \hat{f}(\cdot|\hat{\alpha}, \boldsymbol{X}_T), \quad \hat{f}_Y \equiv \hat{f}(\cdot|\hat{\beta}, \boldsymbol{Y}_T), \quad \text{and} \quad \hat{f}_{X,Y} \equiv \hat{f}(\cdot|\hat{h}, \boldsymbol{X}_T, \boldsymbol{Y}_T).$$

We may then write

$$\begin{aligned} \log(LR) &= (m-r)\int \log\left(\frac{\hat{f}_X(x)}{\hat{f}_{X,Y}(x)}\right) dF_{m-r}(x) \\ &\quad +(n-s)\int \log\left(\frac{\hat{f}_Y(y)}{\hat{f}_{X,Y}(y)}\right) dG_{n-s}(y), \end{aligned} \tag{A.13}$$

where $F_{m-r}$ and $G_{n-s}$ are the empirical cdfs of $\boldsymbol{X}_V$ and $\boldsymbol{Y}_V$, respectively. The term $\log(LR)$ is essentially composed of entropy estimates. We can rewrite it as:

$$\begin{aligned} \log(LR) &= (m-r)\left[KL(f,\hat{f}_{X,Y}) - KL(f,\hat{f}_X)\right] \\ &\quad +(n-s)\left[KL(f,\hat{f}_{X,Y}) - KL(f,\hat{f}_Y)\right] + \delta_1 + \delta_2, \end{aligned} \tag{A.14}$$

where

$$\delta_1 = (m-r)\left\{\int \log\left(\frac{\hat{f}_X(x)}{\hat{f}_{X,Y}(x)}\right)[dF_{m-r}(x) - dF(x)]\right\}$$

and

$$\delta_2 = (n-s)\left\{\int \log\left(\frac{\hat{f}_Y(y)}{\hat{f}_{X,Y}(y)}\right)[dG_{n-s}(y) - dF(x)]\right\}.$$

As shown by (25), $\delta_1$ and $\delta_2$ are negligible in comparison to the other terms in (A.14). Again from (25),

$$KL(f,\hat{f}_X) = C_f r^{-a} + o_p(r^{-a}),$$

where $C_f$ is a positive constant that depends on $f$, and $a$ is the same constant as in (A.10). The other two K-L divergences have similar representations, and so

$$\begin{aligned}\log(LR) &= C_f\left\{(m-r)\left[\frac{1}{(r+s)^a}-\frac{1}{r^a}\right]+(n-s)\left[\frac{1}{(r+s)^a}-\frac{1}{s^a}\right]\right\} \\ &\quad +o_p\left(\frac{(m-r)}{r^a}+\frac{(n-s)}{s^a}\right).\end{aligned}$$

Now we assume that $f \not\equiv g$, i.e., that $\int |f-g| > 0$. Define the mixture density $f_q = qf + (1-q)g$ and note that the KDE $\hat{f}_{X,Y}$ (based on $r+s$ observations) is consistent for $f_q$. On the other hand, $\hat{f}_X$ and $\hat{f}_Y$ are consistent for $f$ and $g$, respectively. For future reference we point out that $\int |f-g| > 0$ implies that $\int |f-f_q| > 0$ and $\int |g-f_q| > 0$.

The proof that $\log CVBF = \log LR(1+o_p(1))$ proceeds exactly as in the case $f \equiv g$. We have

$$\begin{aligned}\log(LR) &= (m-r)KL(f,f_q)+(n-s)KL(g,f_q) \\ &\quad +(m-r)\left[\log L_X(\hat{\alpha})/(m-r)-\int f(x)\log f(x)\,dx\right] \\ &\quad +(n-s)\left[\log L_Y(\hat{\beta})/(n-s)-\int g(x)\log g(x)\,dx\right] \\ &\quad -(m+n-r-s)\Bigg[\log L_0(\hat{h})/(m+n-r-s) \\ &\quad -\int f_{q_{m,n}}(x)\log f_q(x)\,dx\Bigg], \qquad \text{(A.15)}\end{aligned}$$

where $f_{q_{m,n}} \equiv q_{m,n}f + (1-q_{m,n})g$ and $q_{m,n} = m/(m+n)$. Since each of $\int |f-f_q|$ and $\int |g-f_q|$ is positive, it follows from Pinsker's inequality that $KL(f,f_q)$ and $KL(g,f_q)$ are both positive. Applying the same argument as in the case $f \equiv g$, each of the three terms in brackets is $o_p(1)$, and the result follows.

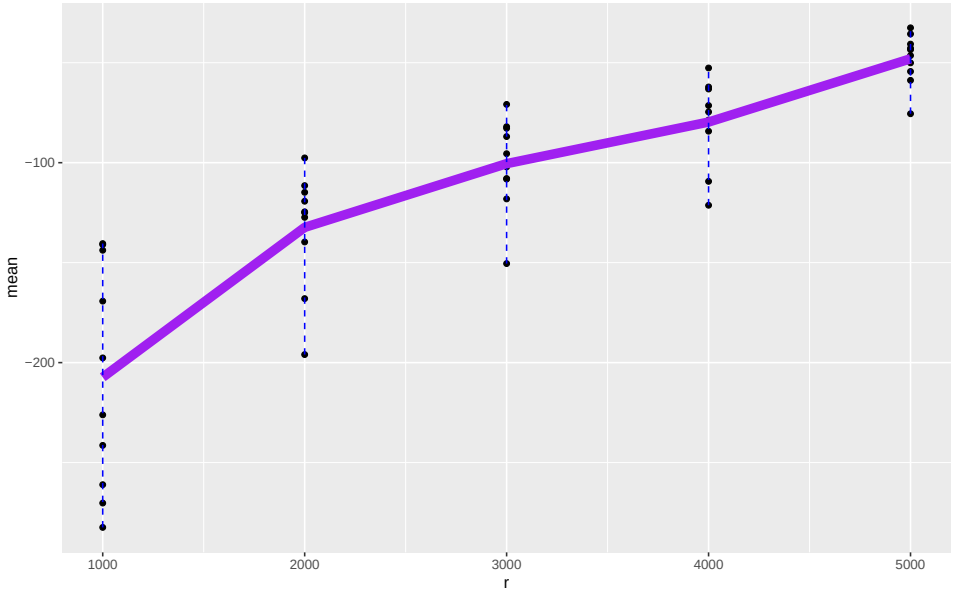
Remarks about Assumption A1 are in order. While this assumption may seem exceptionally strong, it may be defended in two ways. In Section 3.2 we give compelling numerical evidence that the assumption holds. Furthermore, the point is essentially moot since, as stated in the paper,

we use Laplace approximations in the simulations and data analyses, and hence our consistency result does not require Assumption A1 for the statistic used in all our numerical work.

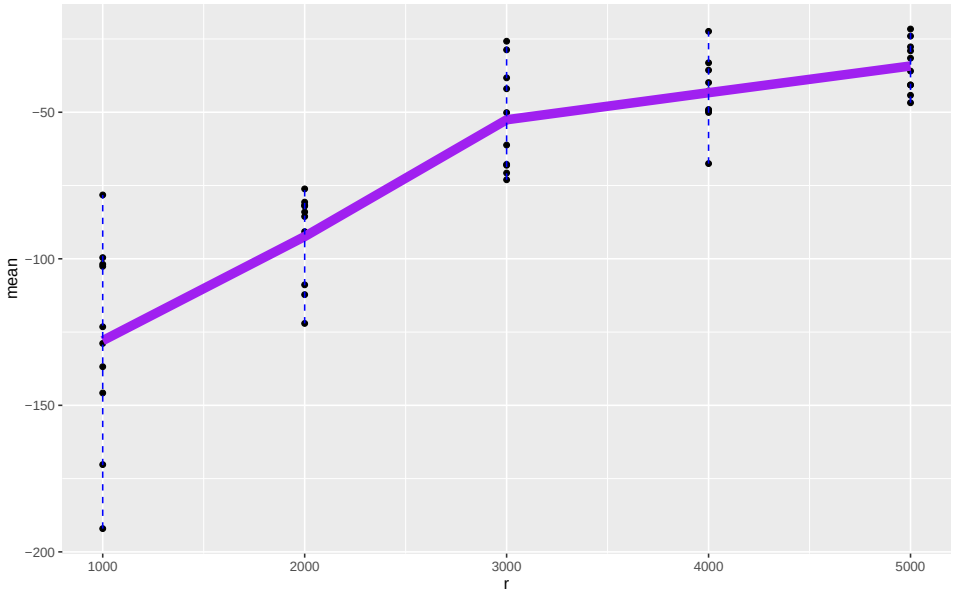
### A.6 Distributions of columns 23 and 29 of the Higgs boson data under the null hypothesis

The permutation-based method of Section 5.2 was applied to both the column 23 and 29 data. The number of permutations in each case was 10 for each different training set size. For the column 23 data, the largest log-Bayes factor for any training size was less than $-20$, which suggests that if the data had supported $f \equiv g$, we would have obtained a negative log-$CVBF$ with large magnitude.

The largest log-Bayes factor for column 29 was still less than $-30$. The similarity of these results with the column 23 results in Figure 8 of the main paper lend further credence to the notion that the data better support $f \equiv g$ than $f \not\equiv g$.




(a) Values of log-CVBF under the Null hypothesis using the permutation based procedure computed from column 29 of the Higgs boson data. The lines connect the averages of log-CVBF at different training set sizes.




(b) Values of log-CVBF under the Null hypothesis using the permutation based procedure computed from column 23 of the Higgs boson data. The lines connect the averages of log-CVBF at different training set sizes.

Figure A.5: *Log-CVBF values under the null hypothesis using the permutation-based procedure for columns 23 and 29 of the Higgs boson data. Our analysis suggests that the two classes in column 23 have the same distribution, and the classes in column 29 have different distributions.*