Doctoral Thesis

Submitted to the faculty at Virginia Tech

Lucas Roberts

Department of Statistics

Virginia Tech

rlucas7@vt.edu

August 28, 2014

Abstract

In this thesis proposal we provide a historical survey of the statistics and computer science research in decision trees. We review both the greedy and Bayesian approaches to decision tree induction and propose a novel modification to previous Bayesian decision tree methods. Our approach facilitates covariate selection explicitly in the model, something not present in most previous research. Covariate selection is facilitated by a transform, which we define, that allows us to use priors from linear models. Using this transform, we are able to modify many common approaches to variable selection in the linear model and bring these methods to bear on the problem of explicit covariate selection in decision tree models. We also provide theoretical guidelines, including a theorem, which gives necessary and sufficient conditions for consistency of decision trees in infinite dimensional spaces. Our examples and case studies use both simulated and real data cases with moderate to large numbers of covariates. The examples give support to the claim that, in large dimensional covariate datasets, if decision tree methods are to be applied, the approach presented here is to be preferred.

Contents

List of Figures iv

List of Tables v

List of Symbols vi

List of Abbreviations vii

1 Introduction 1

1.1 Related Work 2

1.1.1 An Historical Path Through the Literature 2

1.1.2 A Brief Overview of Variable Selection Methods 7

2 Preliminaries 13

2.1 Greedy Induction 13

2.1.1 Impurity Functions 13

2.1.2 Induction 14

2.1.3 A Simple Example 16

2.2 Bayesian Approaches 17

2.2.1 The CGM approach 17

2.2.2 Integrated Likelihood 19

2.2.3 The Process Prior 19

2.2.4 Node Likelihoods and Priors 21

2.2.5 A Bayesian Zero Inflated Poisson Model 23

2.3 Previous Variable Selection 24

2.4 Derivations 25

2.4.1 ZIP Derivations 25

3 Dirichlet Variable Selection For Decision Trees: The DiVaS method 27

3.1 Related Work 29

3.2 Model Details 31

3.3 Ensuring Consistent Classifiers 34

3.4 A Simulated Example 39

3.4.1 Choosing Covariates 43

4 A Case Study **45**

4.1 Discussion of Results 48

5 Additive Logistic Variable Selection: The ALoVaS method **49**

5.1 Normal Distributions Transformed to the Unit Simplex: The ALoVaS method. . . 49

5.2 A Simple Sampler Approach 52

5.2.1 The General Strategy 52

5.3 A Stochastic Search Variable Selection Approach 57

5.4 A Lasso Prior 58

6 A Case study using the ALoVaS Model **60**

6.1 The Need For Sparsity 60

6.2 The internet ads dataset revisited 60

6.3 Simulation Results 60

6.4 Conclusions 60

7 Synthesis: comparing the DiVaS and ALoVaS methods **60**

7.1 Theoretical differences and a simulation study 60

7.2 Practical differences and a simulation study 62

7.3 Conclusions and Recommendations 64

8 Discussion **65**

9 References **72**

List of Figures

1 A simple decision tree . . . 17
2 A shatter coefficient plot . . . 35
3 Maximum depth of samplers trees with maximum depth set at 2 . . . 37
4 Maximum depth of samplers trees with maximum depth set at 3 . . . 37
5 Maximum depth of samplers trees with maximum depth set at 4 . . . 38
6 Maximum depth of samplers trees with maximum depth set at 5 . . . 38
7 Maximum depth of samplers trees with maximum depth set at 6 . . . 39
8 Maximum depth of samplers trees with maximum depth set at 7 . . . 39
9 Maximum depth of samplers trees with maximum depth set at 10 . . . 40
10 A plot of the true DGP . . . 41
11 The tree found by a greedy optimization . . . 41
12 Best tree using the weighted method . . . 42
13 The best tree found by the CGM method . . . 42
14 Covariate inclusion probabilities for the 100 covariate example . . . 43
15 Covariate inclusion probabilities for the 400 covariate example . . . 43
16 The greedy algorithm tree for the internet ads dataset . . . 46
17 The CGM algorithm tree for the internet ads dataset . . . 46
18 The weighted method tree for the internet ads dataset . . . 47
19 Estimated covariate weights for the internet ads dataset . . . 47
20 ALN plot with a zero mean vector . . . 51
21 ALN plot with a negative one mean vector . . . 51
22 ALN plot with a mean vector $(2, 0)^T$ . . . 52
23 ALN plot with mean vector $(2, 0)^T$ . . . 52
24 ALN plot with a mean vector of $(-2, 2)^T$ . . . 53
25 ALN plot $\Sigma = \text{Diag}(0.01,100)$ . . . 53
26 ALN plot Σ numerically singular . . . 54
27 Similar to the case in Figure 25 but with the variances reversed . . . 54
28 ALN plot with a zero vector mean and $\Sigma = \text{Diag}(100, 100)$ . . . 55
29 ALN plot approximating the CGM model . . . 55
30 Results for the zero mean prior . . . 57

31 Results for the informative prior 57
32 write the caption here. 62
33 write the caption here. 62

List of Tables

1 A simple decision tree example data 16
2 Misclassification probabilities for $d = 100$ 42
3 Misclassification Probabilities of the three tree fitting methods for $d = 400$ 44
4 Empirical covariate selection with 100 simulations 45
5 Misclassification probabilities for internet ads dataset 46
6 The set J of covariates using Equation 52 48

List of Symbols

$\lvert\underline{x}\rvert_\nu = \left(\sum_{i=1}^d \lvert x_i\rvert^\nu\right)^{1/\nu}$	The ν norm if the vector $\underline{x}$ 8
$\underline{a}$	A column vector of entries a_i for $i = 1, \ldots, d$ 8
$\mathbb{1}[A]$	Indicator function for the set A 9
$N_{[a,b]}(A\lvert B, C)$	Normal density on A, with mean B, and variance C, truncated to $[a, b]$ 11
$\sim$	Distributed as, or, Distributed according to the density 11
$\propto$	Proportional to 11
$R(\mathcal{T}_i)$	The risk of the ith tree 15
Inv-Gamma$(A\lvert B, C)$	Inverse gamma density on A, with shape B, and scale C 21
Multinomial$(A\lvert B, C)$	Multinomial mass function on A, with count B and probabilities C 22
Dirichlet$(A\lvert B)$	Dirichlet density on A with concentration parameters B 22
$s(\mathcal{A}, n)$	The shatter coefficient for sets $\mathcal{A}$ on sample size n 35
$V_\mathcal{A}$	The VC dimension for functions in the class $\mathcal{A}$ 35
$v_n = (1/n)\sum_{i=1}^n \mathbb{1}(y_i = \hat{\mathcal{T}}(X_i))$	The empirical error of the classifier 36
$v = \Pr(\mathcal{T}(X) = y)$	The theoretical (Bayes) error of the classifier 36
$\mathcal{H}(x)$	The entropy function 36
$J(\underline{y}\lvert\underline{x})$	The Jacobian of the transform from $\underline{y}$ to $\underline{x}$ 49
$\mathbb{R}$	The real number system 49
$\mathbb{S}^d$	The d-dimensional simplex 49
$Diag(a_i)$	A square, diagonal matrix with diagonal entries a_i 50
$\mathcal{O}(A)$	Big "O" of A 51
$\odot$	Hadamard product of two vectors (element-wise product) 52
$MVN(\underline{a}\lvert\underline{b}, C)$	Multivariate normal on $\underline{a}$, with mean $\underline{b}$, and covariance C 53
$S\beta(c\lvert\alpha, \beta, a, b)$	Scaled beta distribution on c with parameters α, β, scaled to the support $[a, b]$. 53
Φ, (Φ^{-1})	Cumulative density (quantile) function for the standard normal 54
$Unif(A, B)$	Continuous uniform density on the region $[A, B]$ 55
Exponential(λ)	Exponential density with rate λ 58
Laplace(μ, σ)	Laplace density with location μ and *scale* σ 58
Inv-Wishart$(A\lvert B, C)$	Inverse Wishart distribution on A, with degrees of freedom B and matrix C . . . **??**
tr(A)	The trace of the matrix A, the sum of the diagonal elements. . . . . **??**

List of Abbreviations

CS	Computer science	1
GLM	Generalized linear model	1
GLMM	Generalized linear mixed model	1
UCI	University of California Irvine	2
ALN	Additive logistic regression	2
CART	Classification and Regression Tree	3
CGM	Chipman, George and McCulloch	4
DMS	Denison, Mallick and Smith	4
MH	Metropolis-Hastings	4
MCMC	Markov chain Monte Carlo	4
FS	Forward selection	7
SSVS	Stochastic search variable selection	9
RJ-MCMC	Reversible jump Markov chain Monte Carlo	10
LAR	Least angle regression	10
LASSO	Least absolute shrinkage and selection operator	10
UUP	Uniform uncertainty principle	11
MSE	Mean squared error	16
VIMP	Variable importance	15
ZIP	Zero-inflated Poisson	25
GP	Gaussian process	28
VC	Vapnik-Chervonenkis	35
DGP	Data generating process	40
MVN	Multivariate normal	57
ZINB	Zero-inflated negative binomial	**??**

1 Introduction

This thesis outlines variable selection as a genre of research in the intersecting domains of statistics, computer science (CS) and other data sciences, including several related subfields of CS and statistics. Our goal is to develop methods to perform *explicit* variable selection within a decision tree model. We emphasize "explicit" because there are several *ad hoc* methods that are currently common in applied practice. These *ad hoc* methods perform variable selection using decision trees, usually with little theoretical justification, and no notion of measure on the individual variables.

Towards our goal, we give an historical overview of decision trees, surveying literature from both CS and statistics, spanning seven decades. While an exhaustive literature review would be a Herculean task, we seek to examine a representative sample of literature to give the reader sufficient knowledge of the choices and strategies applied by researchers. What will emerge is a confluence of several fields including mathematics, statistics, CS and related subdomains applying their own methods of research into this diverse subdomain of study.

We are fundamentally looking toward variable selection as the goal, but a reasonable question to ask is "who cares?"; why should we think that variable selection is worth studying? Moreover, why should we study variable selection in this very specific subdomain of decision trees? The simplest answer is that datasets are getting bigger. Cheap computing power and the march towards an interconnected web of business, socialization and life has nearly automated many data collection processes. This automation has created a 21st century gold rush into any academic pursuit that trains an individual to work with data. Thus, big datasets are here to stay.

Big data can mean a large number of observations, or a large number of variables. In this thesis we are mainly concerned with a large number of variables. Specifically, we study methods to choose subsets of variables from a decision tree model in reasonable ways.

There are many well known data models, perhaps the most common is the linear model. A straight forward extension to the linear model is the transformed linear model, known as the generalized linear model (hereafter abbreviated GLM). Further extensions allow for random effects, known as GLMMs (the extra M stands for 'mixed'). The earliest developed variable selection techniques are usually considered to be the forward, backward, and stepwise techniques. These three techniques were originally studied for linear regression models in the 1960s. Examining historically, as we do in the next subsection, we will see that decision trees were also one of the first forms of variable selection when the landmark paper by Morgan and Sonquist [74] was published.

This groundbreaking work would not be fully appreciated in the research community for nearly two decades. Finally in the 1990s and the first decade of this century, the methods employed in this original paper would be in widespread use, in both academia and industry.

This introductory chapter only outlines the material to be discussed in the subsequent thesis. We give a literature review of decision trees in the proceeding two subsections of this chapter. The final subsection of this chapter provides a brief literature review of variable selection methods. Chapter 2 gives an overview of the various decision tree models discussed in this thesis and gives some technical details about the models. Chapter 3 presents variable selection methods for decision trees using a Dirichlet prior type prior approach for covariate selection. This chapter includes a class of methods for variable selection proposed by the author of this thesis. Chapter 4 presents case studies of decision tree variable selection methods using simulated data and using data taken from the UCI machine learning repository [38]. Chapter 5 presents an alternative approach using normal distributions as a variable selection distribution and transforming to a probability scale using a transform we call the ALN transform. This chapter shows how the class of methods from Chapter 3 can be used to solve the decision tree variable selection problem using non-Dirichlet priors for the probability of selecting a covariate. Chapter 6 outlines the research I propose to conduct during my remaining time at Virginia Tech. This proposed research includes four regularization type approaches to variable selection in decision trees. Each proposed approach is briefly outlined over 1-2 pages. Chapter 7 provides a tentative timeline for the completion of this proposed research.

1.1 Related Work

This section details two important aspects of this thesis: decision trees and variable selection. Little work has focused on variable selection in decision trees and, for this reason, we separate the literature review into two components. In subsequent chapters, we provide more details on the few implementations of variable selection in decision tree problems. We begin with an historical review of the decision tree literature. Then, in a subsequent subsection, we survey the literature on the variable selection problem.

1.1.1 An Historical Path Through the Literature

In this section we review decision tree literature from an historical perspective. We begin with one of the earliest decision tree papers in the statistics literature, the paper by Morgan and Sonquist

[74]. Their proposal suggested that the linear model is often an inadequate method to handle complex survey data analysis. The authors outline several complications with survey data, including high correlations among the covariates, known to cause instability in linear models, and complex interactions, which make the linear model difficult to interpret and complicated to calculate. Furthermore, their paper cites only one previous author who had a similar but not the same approach. They mention an applied statistics paper from 1959 by Belson [2], also an economist. Belson's work certainly predates Morgan and Sonquist's, but the aim of both papers appears to be to partition recursively complex survey data, applying an empirical and simple approach instead of a theoretical approach to analyzing large complicated data.

After the pioneering work of Morgan and Sonquist and that of Belson, the next researcher to begin looking at decision tree methods of data analysis was Kass [61]. Although Kass' paper does not present a graphic of a decision tree, the method he proposes is that of recursive partitioning of the data. He gives suggestions and recommendations for how to carry out this procedure and notes the need for further research in this area. Kass follows this paper up with another paper in 1980 [62]. Kass [62] studies recursive partitioning again but this time specifically on categorical data. Conclusions and results are similar to those in Kass [61]. It is worth noting that although Breiman, Friedman, Olshen and Stone's 1984 CART book [11] is often considered the work that introduced decision trees into the statistics literature, here we have noted at least four papers prior to the 1984 publication of this book and we by no means claim to be exhaustive in our review. Moreover, it is equally surprising that Breiman noted that they published the CART book as a book because the authors thought that no statistics journal would publish the work [41].

The work of Breiman et al. [11] was pioneering in several aspects. The last chapter contains a theoretical proof of the consistency of decision trees as the number of observations, typically denoted n, grows arbitrarily large. Breiman et al. provided a new framework, which involved growing a full tree and then pruning back to optimality. This is the first instance of a consistent stopping rule in the decision tree literature. In addition, the book presents specifics of algorithmic implementation. Also many practical issues such as stopping criteria are thoroughly discussed, indicating why the full growth and then pruning approach is to be preferred over simpler stopping rules previously proposed. These aspects and more make the book excellent reading for theoretical statisticians and applied data analysts. Moreover, the low cost of the book and the practical interpretability of decision trees made the method popular among researchers in several fields.

Other work around the same time included the pioneering work of Quinlan [81] and his text-

book [82]. Both Quinlan [81]and Quinlan [82] discuss induction learning for decision trees. Quinlan's research differs from the statistical approaches in several aspects, the most fundamental differences are Quinlan uses multiway splits compared to only binary decision tree splits in the previous statistics literature and Quinlan uses an information theoretic approach to justify decision trees whereas the statistics literature takes a nonparametric approach.

Attempting to improve the prediction errors of CART trees, Loh and Vanichsetakul proposed using linear combinations of covariates as splitting rules instead of the simple axis orthogonal splits of the CART methodology. The fundamental differences between the CART method and the work of Loh and Vanichsetakul are: 1) multiple splits are possible at each node; 2) a direct stopping rule; 3) estimating missing values; 4) splits are linear combinations and may contain both categorical and continuous covariates; 5) no invariance to monotonic transformations; 6) Computationally faster than Breiman et al.'s method [70]. Furthermore, Loh and Vanichsetakul used statistic inference approaches such as F ratios to choose splits and to stop the tree from growing. This work was criticized by Breiman and Friedman [12] on several aspects, the most important of which are the lack of robustness, no proof of a consistent stopping rule, and lack of invariance to monotonic transformations.

In the subsequent decade much research was done, both empirically and theoretically regarding the efficacy of decision tree methods. The next major extension was done in the year 1998. Two groundbreaking papers were published, one by Chipman, George and McCulloch [21], and another by Denison, Mallick and Smith [28], hereafter referred to as CGM and DMS respectively. Both articles brought Bayesian computational techniques to bear on the problem of decision tree induction. Much Bayesian computational work was accomplished following the groundbreaking Gibbs sampler first published in 1990 by Gelfand and Smith [43], such as Metropolis-Hastings (MH) samplers [54, 19] and reversible jump methods [52]. The papers by CGM and DMS brought the 8+ years of research in Markov chain Monte Carlo (MCMC)methods into decision tree search methods. CGM proposed a novel process prior and gave a set of proposal functions that exhibit useful cancellation in the MH ratio. In a similar vein, DMS proposed a reversible jump algorithm with a similar proposal function that appears to mix more efficiently while searching the space of trees. These two papers are the genesis of our developments in the current thesis.

Around the same time (the 1990's), the machine learning community was experiencing rapid growth. During this time of rapid growth, Leo Breiman helped to bridge the gap between the statistics community and the machine learning community, publishing fundamental work propos-

ing the bagging method [9]. During the same year, 1998, Ho proposed a similar generalization known as the random subspace method [55]. Both algorithms use resampling methods similar to the bootstrap [35, 36], but build a decision tree on each subsampled dataset and then aggregate the predictions from the resulting trees to improve prediction and stability of the estimator. Although more refined and developed, a similar approach is the random forest model [10]. The random forest model is the subject of current research into the theoretical properties of the resulting collection of trees and their predictions [6, 5]. This model is very similar to bagging and differs primarily in implementation details. Also, the practical performance has empirically been shown to outperform the bagging method [10].

The authors Chipman, George, and McCulloch (CGM) [21] did further further fundamental research in developing Bayesian decision trees. In 2000 CGM formulated another modification of their previous model, this time proposing another clever prior that encouraged shrinkage and sharing of information by nodes close together in the tree [22]. Then in 2002, these authors' proposed another modification to their previous work suggesting that perhaps constant models in each terminal node were not effective or general enough to effectively model observed data. Instead, they suggested to allow GLM's in each terminal node and called the resulting models *treed* models [23], which generalize classic decision tree models to include linear or generalized linear models within each terminal node. It is worth noting that the previous models proposed using constant functions are in fact special cases of treed models, they are linear regression models with only an intercept term in each terminal node. Furthermore, the amount of available data decreases the further into the tree you traverse, so either tree regularization or node model regularization is necessary to combat the "small n, big p" problem that occurs in terminal nodes of large trees.

Several further refinements to the Bayesian approach were proposed during the subsequent decade. Wu, Tjelmland and West [93] offered an improved proposal function that contains a radical restart move that grows a new tree from scratch when the current chain is stuck in a local maximum. This is accomplished by a "pinball prior", which is one of the unique and practically useful aspects of the paper. As a further improvement, to aid the mixing of the Markov chain Monte Carlo and to aid the chain in traversing from one local maximum to another, Leman, Chen and Levine proposed a new MCMC algorithm called the multiset sampler [66]. The multiset sampler is able to achieve a move from one local max to another by allowing the chain to be in two states of the Markov chain at a single instant. The multiset sampler was originally developed for evolutionary inference in the reconstruction of phylogenetic trees, however this specific application was referred

to as the evolutionary forest algorithm. The extension from phylogenetic trees to CART trees is fairly straightforward. Moreover, Gramacy and Lee extended the treed model to include Gaussian processes and gave an application to simulation of rocket boosters [49]. Gramacy and Taddy [50] provide an R package that implements the ideas in Gramacy and Lee [49] and provides several extensions and special cases.

Several other important works deserve to be mentioned during this timeframe (the 2000's). In the computer science domain, Dobra advanced a scalable algorithm to do decision tree induction called SECRET [30]. In addition to providing scalable algorithms for decision tree induction Dobra, offered a novel modification to decision trees: a probabilistic split at each node. This creates fuzzy classifiers and fuzzy regions in the covariate space around which splits are made. The probabilistic splits create regions of splits in the covariate space. The same phenomenon is observed in bagged trees [9], although Dobra's approach is simpler and preserves the interpretability of the single tree. Along different lines, Gray and Fan proposed genetic algorithms to build decision trees in their TARGET algorithm [51]. Genetic algorithms are roughly similar to the population approaches [16] except they rely on an analogy with evolutionary processes to guide their development.

During the same decade (the 2000's), the stigmergic approach to decision tree induction was investigated empirically. The stigmergic approach generally works by agents, or a population approach, whereby each agent is able to communicate with other agents through the environment in which the agent acts. In the case of the ant colony optimization algorithm [33], stigmergy is achieved by ants leaving pheromone trails that influence the behavior of subsequent ant agents. While ant colony optimization approaches do not generally yield the best performance in the training data, they are generally competitive with other algorithms in test data prediction and provide differing and often insightful trees [32]. Several authors have modified algorithms to build decision trees using the antminer system[79] [67]. The antminer system is publicly available in Matlab code at the link http://www.antminerplus.com/ .

Two additional statistical approaches have been advocated recently. The EARTH algorithm and the GUIDE algorithm [31] [69]. EARTH is an algorithm that nonparametrically selects covariates to include in the model. Doksum, Tang, and Tsui provide a theorem that shows the covariate selection consistency of the EARTH algorithm as the sample size, $n \to \infty$, and as the dimensionality of the data, $d \to \infty$.

The GUIDE algorithm [69], proposed by Loh, is similar to the original work of Loh and Vanich-

setakul [70]. The GUIDE algorithm is a general unbiased interaction detector that claims to have superior performance at detecting interactions and incorporating those into the model by splits that are linear combinations of covariates. Unfortunately, there is no guarantee of consistency of the GUIDE algorithm.

Decision trees have long been applied to survival data. The explosion of cheap genome sequencing and generally cheap data collection and storage has made variable selection methods increasingly useful in the context of survival trees. The work of Ishwaran, Kogalur, Gorodeski, Minn, and Lauer defines a new quantity called a maximal subtree and use the inherent topology of the tree and the maximal subtree to measure variable importance [59]. Ishwaran et al. advocate a bootstrapping approach to variable selection in the context of random survival forests. The authors noted good empirical performance and provide probability approximations to calculations necessary in their simulations.

Finally, Taddy, Gramacy and Polson have extended the decision tree literature into the time series domain [86]. The authors propose to embed decision trees into a dynamic stochastic process. The authors suggest that the underlying tree of the model is updated by alternating grow, split, and do nothing moves. Taddy et al. also provide an illustrative example using car crash data. Time series applications of decision trees appears to be a fruitful area of research.

1.1.2 A Brief Overview of Variable Selection Methods

In order to understand the methods we will employ in further chapters, we will give a brief overview of variable selection methods for linear models. We focus on linear models because the majority of the research into variable selection has been conducted on these models. Of course extensions have been done for certain methods on GLMs, however this material is usually more complicated and specialized. Also these extensions have not been applied to all the methods we discuss. In subsequent chapters, we will see how the material we introduce here can be applied to decision trees using an appropriately defined transform.

Perhaps the earliest variable selection method, besides the modest proposal of Morgan and Sonquist [74], is the forward selection method (FS). The forward selection method and many variations appear in the early 1960's from several references making it very difficult to identify the person who originally proposed this method. References in other languages are not included in this review, further obfuscating the designation of first proposal. The FS method is well know to run into difficulties when several covariates are highly correlated with each other [71]. There

are several nice benefits to the FS method, such as computational feasibility and readily available, high quality computer codes that implement this technique. Nonetheless, several researchers have pointed out the sometimes dubious nature of the resultant output [53].

A related method to forward search is backwards search, which operates analogously to the forward search, except the model starts with all covariates in the model. At each iteration the variable with the lowest correlation with the response is removed from the model. Also, the stepwise method devised by Efroymson [37] represents a middle ground between forward and backward search by sequentially adding and deleting variables. The stepwise method tries at each step to include or exclude a variable, based upon an F statistic value. The backward and stepwise searches are also known to encounter similar difficulties as FS [72] does. Highly correlated covariates may produce dubious results. A nice overview of all three methods, along with several other subset selection approaches can be found in Miller [72]. The second edition of Miller's book contains many updates, including chapters on Bayesian and regularization methods.

Ridge regression is another popular approach used in subset selection problems [56]. The ridge regression estimator is defined as

$$\underline{\hat{\beta}} = \underset{\underline{\beta}}{argmin}\ (\underline{y} - X\underline{\beta})^T\Sigma^{-1}(\underline{y} - X^T\underline{\beta}) + \lambda\underline{\beta}^T\Gamma\underline{\beta}, \tag{1}$$

for $\lambda \geq 0$ some scalar (constant). The objective function (1) has the closed form solution $\underline{\hat{\beta}} = (X^TX + \lambda\Gamma)^{-1}X^T\underline{y}$ and Γ is a matrix chosen to be conformable for addition with X^TX. Ridge regression combats the effect of high correlation among the columns of X, allowing the matrix (X^TX) to be inverted. This method is often most useful when $d < n$ and regression coefficients are desired. Note that estimated $\underline{\hat{\beta}}$ vectors with zero entries are unattainable in this penalized regression method when Γ is a full rank matrix.

Frank and Friedman [39] proposed another penalized regression approach to ridge regression called bridge regression. Frank and Friedman also gave an optimization algorithm that solves the objective function. The objective function to solve was defined as

$$\underline{\hat{\beta}} = \underset{\underline{\beta}}{argmin}\ \ (\underline{y} - X\underline{\beta})^T\Sigma^{-1}(\underline{y} - X^T\underline{\beta}) + \lambda|\underline{\beta}|_\nu^\nu, \tag{2}$$

where the second term denotes a ν norm and $\nu \geq 0$ is a specified constant. This technique later became known as "bridge" regression, because the objective function bridges between several well known estimators by choosing various values of ν. For example $\nu = 0, 1, 2$ correspond to subset

selection, the lasso, and ridge regression, respectively. We note that the $\nu = 0$ case is interpreted as $\lim_{\nu \to 0} |\underline{\beta}|_\nu^\nu = |\underline{\beta}|_0$, the limiting case of the ν-norm.

During the 1990's Bayesian approaches became practical because of advances in computational statistics, especially developments in Gibbs sampling and MH algorithms. These advances lead to several researchers proposing Bayesian variable selection techniques. The first of these advances in the variable selection literature was the stochastic search variable selection (SSVS) approach of George and McCulloch [45]. The SSVS approach relies on

$$\delta(x_0) = \lim_{\sigma \to 0} \phi(x_0; \sigma) = d\mathbb{1}[x \geq x_0], \tag{3}$$

where $\phi(a; b)$ denotes a Gaussian density evaluated at the point a, with standard deviation b and mean zero [47, 73]. The notation $\delta(x_0)$ will be used to denote the Dirac delta functional. Essentially we are trying to determine if $\beta_j = 0$, or if $\beta_j \neq 0$ and we might wish to assign a point mass probability to $\beta_j = 0$. Thus, a reasonable approximation is to use a two component mixture of normal distributions, with one normal having a large variance compared to the other. Using a latent variable representation, George and McCulloch also gave a Gibbs sampling algorithm that samples subsets of predictors and thereby provides variable selections. The main drawback is that George and McCulloch only offered SSVS for the Gaussian linear model. Extensions in the literature indicate the method can be applied to GLMs and to problems where the number of covariates is larger than the number of observations, such as gene selection [95, 46].

Historically, the 1990s was a fruitful period of research and often rediscovery of methods proposed earlier but were not yet computationally feasible. Breiman advocated better subset selection using the nonnegative garrote [8, 96]. As indicated by the title, Breiman's procedure [8] selected better subsets compared to backward search and subset selection. The subsets selected are the non zero values of the estimated coefficients, conventionally denoted as $\underline{\hat{\beta}}$. In the non-negative garrote problem these estimates of $\underline{\beta}$ are obtained by solving the objective function

$$\underset{\forall j: c_j \geq 0}{argmin} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{d} c_j \hat{\beta}_j x_{ij})^2 + \lambda \sum_{j=1}^{d} c_j, \tag{4}$$

where λ is a specified constant, or estimated by some other means. The estimate $\hat{\beta}_j$ denotes the jth least squares estimate. Alternately we can use a estimate of β obtained by means other than least squares. Once $\hat{c}_j$ is estimated, the non-negative garrote estimated is defined as $\hat{\beta}_j^{\text{NNG}} = \hat{c}_j \hat{\beta}_j^{\text{LS}}$. Under an orthogonal covariate assumption ($X^T X = I$), there are closed form solutions to the

objective function (4) that have nice interpretations as hard thresholding rules. The threshold is determined as a function of the constraint multiplier λ. Breiman compared the non-negative garrote method against subset selection and backward search procedures, indicating positive results for the non-negative garrote. Yuan and Lin [96] and Xiong [94] give further results for the non-negative garrote. Yuan and Lin used the theory of duality to give oracle type properties of the nonnegative garrote estimator. Briefly, the oracle property states that $\Pr(\hat{\beta} = \hat{\beta}_{\text{global}}) \rightarrow 1$ as $\lambda_n \rightarrow \lambda$. In other words, this means the local estimator becomes the global estimator for a suitably constructed sequence of regularization parameters. Xiong studied iterating the non-negative garrote procedure and also how the degrees of freedom influence the prediction risk of estimates.

We now move to discuss a Bayesian approach to variable selection in the context of sampling methods. Besides the method proposed by George and McCulloch [45], there is another popular method applied to variable selection problems in a Bayesian context. This alternate method is known as reversible jump Markov chain Monte Carlo (RJ-MCMC). This method was first suggested by Green [52]. Green showed that mixtures of normal distributions can be modeled using the RJ-MCMC sampler. Specifically, the RJ-MCMC algorithm eliminates the need to specify the number of mixtures in the normal distribution, effectively eliminating tuning parameters from normal mixture distribution problems.

One of the most fruitful areas of research in the last 15 years has been the lasso, which stands for *least absolute selection and shrinkage operator*. Motivated by the non-negative garrotte, the lasso was proposed by Tibshirani in 1996 [87]. Many subsequent papers give properties of the lasso, or proposed alternate methods to solve the objective function. The objective function is defined as

$$\underset{\underline{\beta}}{argmin} \sum_{i=1}^{n}(y_i - \sum_{j=1}^{d} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{d} |\beta_j|_1, \tag{5}$$

where the $\lambda \geq 0$, is a constant or tuning parameter. Tibshirani solved the optimization by doing a grid search across several values of λ. Unhappy with the computational complexity of the linearly constrained quadratic programming optimization approach suggested by Tibshirani [87], Efron et al. [34] proposed the least angle regression algorithm, hereafter referred to as LAR, to solve the lasso optimization problem. The LAR algorithm uses a homotopy method to solve the objective function (Equation 5). Along similar lines, both Osborne, Presnell and Turlach, and Zhao and Yu used duality theory to prove properties of the lasso estimators [77, 97]. Bunea, Tsybakov, and Wegkamp [13] and Candes and Plan [14] have also derived oracle and optimality properties

of the lasso problem estimators. These optimality results state that lasso solutions are within a constant factor of the true values, with the constant usually being a number of the form $1 + \epsilon$ and $1 \geq \epsilon \geq 0$. Tibshirani [87] originally noted a Bayesian approach to interpreting Equation 5. Park and Casella [78] gave further Bayesian lasso results including a marginal maximum likelihood (empirical Bayes) method for estimating the tuning parameter λ. An empirical Bayes argument is used to justify this method of estimating λ. Finally, Zhou and Hastie [98] combined the penalties of the lasso and of ridge regression and called this objective function the elastic net. The elastic net can be interpreted as a convex combination of a lasso ($|\underline{\beta}|_1$) penalty and a ridge regression ($|\underline{\beta}|_2^2$) penalty. Zhou and Hastie [98] provided a transformation to convert the elastic net problem into a lasso problem so that the LARs algorithm can be used to solve the elastic net objective function efficiently.

The flurry of regularization papers on the lasso and the non-negative garrote methods inspired Candes and Tao [15]. Candes and Tao [15] advocated an alternate method of estimating regressors called the Dantzig selector. The Dantzig selector estimates a sparse vector of coefficients, denoted $\hat{\underline{\beta}}^{\mathrm{D}}$, by solving the objective function

$$\underset{\underline{\beta}}{argmin}\ |\underline{\beta}|_1 + \lambda |X^T(\underline{y} - X\underline{\beta}) - k_p\sigma|_\infty. \tag{6}$$

Candes and Tao [15] suggested that this objective function be reformulated as a linear program. Linear programs have several highly reliable software applications to estimate the optimum. Candes and Tao [15] also gave a primal-dual interior point algorithm to solve the objective function with publicly available Matlab code. The authors emphasized a uniform uncertainty principle (UUP) and derived oracle and optimality results based on the UUP. Bickel, Ritov, and Tsybakov [7] and Koltchinksii [63] provided further theoretical analysis of the Dantzig selector, including optimality results and oracle inequalities under different conditions than Candes and Tao [15].

The last estimator we discuss is the horseshoe estimator, arising from the horseshoe prior. This prior was proposed by Gelman as a way to combat a numerical difficulty in MCMC sampling [44]. Carvalho, Polson, and Scott [18, 17] describe the general setup. The horseshoe estimator arises via the hierarchical probability representation described in Equations 7-10 beneath

$$\underline{y}|\underline{\beta} \sim N(X\underline{\beta}, \sigma^2 I) \tag{7}$$

$$\beta_j|\lambda_j, \tau \sim N(0, \lambda_j^2\tau^2) \tag{8}$$

$$\pi(\lambda_j) \propto \frac{1}{1+\lambda_j^2}\mathbb{1}[\lambda_j \geq 0]. \tag{9}$$

$$\pi(\tau) \propto \frac{1}{1+\tau^2}\mathbb{1}[\tau \geq 0]. \tag{10}$$

It is worth noting that the local scale priors λ_j for $j = 1, \ldots, d$ are on the standard deviation scale and this prior is one of a general class of half-t densities [44, 80]. Carvalho et al. [17] showed that the horseshoe prior has several appealing properties, including sparsity properties shared by the lasso estimator, while also having other desirable properties *not* shared by the lasso. Carvalho et al. [18, 17] gave examples demonstrating the utility of this method for variable selection and indicated superior performance compared to the lasso in the datasets examined. Moreover, Polson [80] argued that the half-t prior, or the half Cauchy prior as a special case, should become the reference (default) prior for variable selection problems. Polson justified this argument based on the many desirable properties of the horseshoe, some of which are not shared with other estimators such as the lasso.

The attentive reader may notice that the majority of material in this subsection pertains to linear regression models. Decision tree models are inherently nonlinear in nature so it is reasonable to wonder: Will we use any of the material discussed in this section? The short answer is "yes," but not in the way of extensions. Our approach will be more in terms of similarity of motivation and spirit. We explain the decision tree model we work with in the next chapter and show how we will exploit sparsity in said model by using modifications to distributions to encourage sparsity. Moreover, we will compare our proposed methods against many of the methods described in this section throughout this thesis.

We conclude this section by noting some review papers on the variable selection problem within the literature. O'Hara and Silanpää [76] gave a recent notable Bayesian survey. This paper covered many of the methods discussed above in some detail and compared them all from a Bayesian perspective. Miller [72] gave a book length treatment on variable selection methods, with the second edition of the book including some Bayesian methods and the lasso, but not all methods discussed in this section. In particular, Miller [72] described the forward, backward and stepwise searches in detail and gave several useful references to the literature. From the CS literature, the review by Dash and Liu [27] provided a useful reference into the CS field developments in variable selection. Dash and Liu [27] also provided a useful framework to compare subset/variable selection approaches. George [46] provided a good overview and major references in the variable or subset selection field current to the year 2000.

2 Preliminaries

This chapter provides an overview of the models used for decision tree induction and variable selection. We begin by discussing the earliest methods of induction: greedy algorithms. With the advance of time (about a decade) greedy induction approaches became fully developed high quality fortran codes. Bayesian approaches to building decision trees were not possible until advances in computing power and the rediscovery of sampling based approaches to Bayesian statistics became popular again. While the work of Breiman et al [11] always contained a Bayesian flavor, no probability measure over trees was ever proposed. That is until the groundbreaking works of Denison, Mallick and Smith [28] and Chipman, George, and Mcculloch [21].

During the same time (the 1990s and the 2000s), researchers in Machine Learning and Statistics were developing theory beyond that published by Breiman et al. [11], that ensured decision trees were consistent. In this case consistency means the error rate of the tree converges to the Bayes error. The initial impetus for this work stemmed from the landmark paper of Vapnik and Chervonenkis [90], and subsequent work [89], though it would take over a decade for the full power of their combinatorial approach to be widely used in both Machine Learning and Statistics.

This chapter collects necessary preliminaries for the reader to understand the developments in subsequent chapters. Therefore the reader may omit this chapter on a first reading and refer back to the necessary subsections when subsequent chapters recall the preliminaries. Subsection 2.1 and subsection 2.2 review the necessary material on greedy and Bayesian approaches to decision tree induction. The reader who is not familiar with these two approaches to decision trees should read these two subsections before moving on to the rest of the thesis.

2.1 Greedy Induction

In this section we answer the question: how does greedy induction work? Once the reader has completed this section they should, with sufficient time, be able to reproduce computer codes that would induct decision trees using greedy strategies. We begin with impurity functions.

2.1.1 Impurity Functions

An impurity function corresponds to a likelihood function in statistics or to an objective function in machine learning. This function measures in some sense, how many good observations lie in a

node of the tree and how many bad observations lie in a node of the tree. Formally an impurity function, denoted ϕ must satisfy three axioms.

1. $\phi(1/2, 1/2) \geq \phi(p, 1-p)$ for any $p \in [0, 1]$.
2. $\phi(0, 1) = \phi(1, 0) = 0$.
3. $\phi(p, 1-p)$ non-decreasing for $p \in [0, 1/2]$ and non-increasing for $p \in [1/2, 1]$.

The impurity functions given in Breiman et al. [11] are:

1. Entropy: $\phi(p, 1-p) = -plog(p) - (1-p)log(1-p)$.
2. Gini: $\phi(p, 1-p) = 2p(1-p)$.
3. Misclass probability: $\phi(p, 1-p) = min(p, 1-p)$.

If a regression tree is to be built/inducted, in place of an impurity function, we use the mean squared error criteria denoted here as

$$\text{MSE} : \phi(y_i, \bar{y}) = \sum_i (y_i - \bar{y})^2.$$

The greedy approach to decision tree induction always tries to optimize impurity functions. The optimization problems are known to be NP-Hard (for an introduction to complexity theory c.f. Garey and Johnson [42]). Briefly NP-Hard optimization problems are considered some of the hardest optimization problems to solve. These optimization problems increase exponentially in the size of the problem. Therefore, exact methods would nearly always take too long to compute and thus greedy strategies are used to approximate the global optimum, assuming one exists, with a local optimum.

2.1.2 Induction

The induction of decision trees proceeds by solving the objective function

$$\underset{t,s}{argmax} \Delta\phi = \phi - \pi_L \phi_L - \pi_R \phi_R. \tag{11}$$

Here the variable t and s scroll over all covariates in the data and all midpoints between successive observations (or at observed points) of the tth covariate respectively. The proportions π_L and π_R

represent the number of data points going into the node to the left (L) and right (R) of the current node if the chosen split is on covariate t and observation s, and similarly defined subscripted impurity functions. Thus, if there are 100 observations in the current node and as a result of the split on covariate t, at observation s, 70 observations go into the left child node and 30 observations go into the right child node, then the two proportions are $(\pi_L, \pi_R) = (0.7, 0.3)$.

The tree induction process continues until no more data points are incorrectly classified, or a predetermined stopping rule is met. Once the full tree has been built, the second stage of the process now starts. This is known as the pruning stage. Now that the full tree is grown, we progressively prune back terminal nodes of the tree until the root node occurs. Several related approaches have been proposed in the literature to select the optimal tree via pruning. The most common is to select the tree using the regularized risk estimate given in Equation 12

$$R(\mathcal{T}_i, \alpha) = R(\mathcal{T}_i) + \alpha|\mathcal{T}_i|. \tag{12}$$

Here the α parameter is a regularization parameter with larger values of α given greater penalty to the number of terminal nodes in the tree, here denoted $|\mathcal{T}_i|$. The notation $R(\mathcal{T}_i)$ denotes the risk of the tree, which is usually calculated as the sum of squared errors across all terminal nodes in a regression setting or the sum of the impurity function values in each terminal of the tree in the classification case. The choice of α is done over a grid of positive values on holdout data, using a cross validation approach, and the value of α leading to the smallest regularized risk ($R(\mathcal{T}_i, \alpha)$) is chosen.

Discussion of consistency of this pruning rule can be found in Devroye et al. [29], Breiman et al. [11], Gey [48], and Suavé and Tuleau-Malot [83]. All the theoretical results require controlling the complexity of the decision tree, $|\mathcal{T}|$, and allowing the number of data points $n \to \infty$. However, the results in Devroye et al. [29] also give explicit bounds on the error of decision tree classifiers for finite values of n.

The pruning rule discussed here, and the induction process overall, is an implicit form of model selection. This is implicit because the variables that are selected are considered important and those variables not selected are considered not important. Furthermore, no measure of importance on each variable is defined, so it is difficult to rank variables based on importance. A remedy, called variable importance (abbreviated VIMP) was proposed in the literature by Breiman [10], but this is in the context of random forests and not for a single decision tree. We proposed several different methods to perform explicit variable selection for Bayesian decision trees in later chapters.

2.1.3 A Simple Example

In this subsection we work through a simple example to give the reader a flavor of the calculations necessary to induct a decision tree.

Consider the following data

i	y_i	x_1	x_2
1	1	2	3
2	2	5	6
3	5	8	9

Table 1: A simple decision tree example data.

We have 3 observations and two covariates within each observation. The response is a continuous random variable, so this will be a regression tree approach. We will examine potential split points by looking at the midpoints between two observed values of the covariates. We begin by sorting the data in increasing order for both covariates. Fortunately, in this case, the data is already in such a sorted order for both covariates, so no sorting is necessary. We now examine the possibility of a split point on x_1. Using the MSE impurity we calculate ϕ, ϕ_L, and ϕ_R. A sum of squared error calculation shows $\phi = 26/3$ which is a constant for all calculations we perform.

Now for a split between observation 1 and 2:

$$\Delta\phi = 26/3 - (1/3)(1-1)^2 - (2/3)((2-7/2)^2 + (5-7/2)^2) = 26/3 - 9/3 = 17/3$$

and for a split between observation 2 and 3:

$$\Delta\phi = 26/3 - (2/3)((1-3/2)^2 + (2-3/2)^2) - (5-5)^2(1/3) = 26/3 - 1/3 = 25/3$$

Now, because all the data is sorted, the same $\Delta\phi$ values will result for potential splits on x_2. Therefore, we (somewhat arbitrarily) choose to split on the covariate with the smaller index (x_1). Thus, our first split is on the value $\{x_1 : x_1 \leq 6.5\}$ and the tree at this point looks like that in Figure 1. Note the mean of the values in each terminal node is given in Figure 1. The mean value would be the predicted value for observations falling into the given terminal node.

We would then continue calculating $\Delta\phi$s for the resulting data that falls into each of the two resultant terminal nodes, continuing until there is only one observation in each terminal node, or

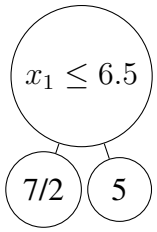


Figure 1: The decision tree after the first split.

until there is some specified number of observations in each terminal node. Although any number is possible, the specified minimum number of observations in each terminal node is usually 5. Once this process is completed, the induction step is finished, and the pruning process begins.

2.2 Bayesian Approaches

This section describes the Bayesian approach to decision trees. The methods described in the previous chapter provided an algorithm to fit a decision tree using a greedy algorithm. Besides the observed error, there is nothing to describe the fit of the model, or to provide a measure over decision trees. This section provides both of these quantities. We begin by defining the model and calculating necessary quantities for the algorithm. Furthermore, there is no explicit model selection, which will be the main contribution of this thesis.

2.2.1 The CGM approach

We begin by defining notation and measures on each quantity of the tree. We assume the tree topology and split rules are conditionally independent. Based on fundamentals of probability we have the following relations:

$$\Pr(\mathcal{T}_i|\underline{y}, X) \propto \Pr(\mathcal{T}_i)\Pr(\underline{y}|\mathcal{T}_i, X) \tag{13}$$

$$\propto \Pr(\mathcal{T}_i)\int_{\Theta}\Pr(\underline{y}|\mathcal{T}_i, X, \theta)\pi(\theta)d\theta, \tag{14}$$

where $\Pr(\mathcal{T}_i)$, denotes the prior measure on trees and $\Pr(\underline{y}|\mathcal{T}_i, X)$ denotes the integrated likelihood of the tree. Finally, $\Pr(\underline{y}|\mathcal{T}_i, X, \theta)$, and$\pi(\theta)$, denote the tree likelihood and prior measure on node

parameters respectively. This is the conditional decomposition defined by CGM [21]. We now proceed to define the aspects of the model described in CGM's paper [21]. The model has two main components, the tree $\mathcal{T}$ with b terminal nodes, and the parameters in each terminal node $(\theta_1, \ldots, \theta_b)$. The two main likelihoods in each terminal node are the normal and the multinomial, for continuous and categorical responses, respectively. Denote the responses in each terminal node as the vector of vectors $Y \equiv (Y_1, \ldots, Y_b)$. Then $Y_i = (y_{i1}, \ldots y_{in_i})$, and the main relation is the independence breakdown

$$f(Y|\mathcal{T}, X, \theta) = \prod_{i=1}^{b} f(Y_i|\mathcal{T}, X, \theta_i) = \prod_{i=1}^{b} \prod_{j=1}^{n_i} f(y_{ij}|\mathcal{T}, X, \theta_i). \tag{15}$$

The two likelihoods are now given by

$$f(y_{ij}|\mathcal{T}, X, \theta_i) = N(\mu_i, \sigma_i), \tag{16}$$

and the multinomial likelihood is

$$f(y_{i1}, \ldots, y_{in_i}|\mathcal{T}, X, \theta_i) = \prod_{j=1}^{n_i} \prod_{k=1}^{K} p_{ik}^{\mathbb{1}(y_{ij}=k)}. \tag{17}$$

In Equation 17, p_{ik} denotes the probability of being in category k in terminal node i and $\mathbb{1}(A)$ denotes the indicator function for the set A.

We now proceed to define the tree prior. We start with a tree consisting of a single node, the root node. We then imagine the tree growing by randomly choosing terminal nodes to split on. To grow a tree we must specify two functions, the growing function and the splitting function. The splitting function is denoted as $p_{\text{split}}(\eta, \mathcal{T})$ and the rule function is defined as $p_{\text{rule}}(\rho|\eta, \mathcal{T})$. The rule function provides a criteria to determine which of the two children nodes observed data go into. If the observed covariate value is less than the rule value, then observations go into the left child node. Similarly, if the observed covariate value is greater than the rule value, then observations go into the right child node. Growing the tree consists of creating two new children from a terminal node and assigning a rule to the terminal node (now a parent of two terminal nodes).

The probability measure on the potential splits of the tree is defined as

$$p_{\text{split}}(\eta, \mathcal{T}) = \alpha(1 + d_\eta)^{-\beta}, \;\; \alpha > 0, \beta \geq 0, \tag{18}$$

where d_η denotes the depth of the node η and α, and β are scalars.

$$p_{\text{rule}}(\rho|\eta, \mathcal{T}) \propto \underbrace{\Pr(\text{split on covariate})}_{=(1)} \underbrace{\Pr(\text{split on a value given a covariate})}_{=(2)}. \tag{19}$$

Here CGM recommends using a uniform prior on (1), and splitting uniformly amongst the splitting values (in (2)) that do not result in an empty terminal node. While we choose the same proposal mechanism for quantity (2), the main point of this thesis is to examine and propose alternate specifications for (1). The data sets modeled in CGM [21] and DMS [28], and other modifications in the literature, deal with data with a small number of predictors. In this thesis we are concerned with a large number of predictors, so we will focus on variable selection, which will ultimately explain how we specify quantity (1) in Equation 19.

2.2.2 Integrated Likelihood

We will now focus on the integrated likelihood, which is the quantity

$$\Pr(Y_i|\mathcal{T}, X) = \int_{\Theta} \Pr(Y_i|\mathcal{T}_i, X, \theta)\pi(\theta)d\theta. \tag{20}$$

To evaluate the integral in Equation 20 we must first define a prior, denoted $\pi(\theta)$, for the parameters in each terminal node. There are two possible priors for the case of the normal likelihood that will result in a conjugate prior/posterior. These are normals and normals and gammas, or normals and inverse gammas depending upon the given parametrization.

2.2.3 The Process Prior

Assuming we have a closed form solution for the integral in Equation 20, we can use Bayes' rule to determine

$$\Pr(\mathcal{T}|Y, X) \propto \Pr(Y|X, \mathcal{T}) \Pr(\mathcal{T}). \tag{21}$$

We now have an effective means of searching the posterior space over trees to determine the high posterior trees. We can do so by using the Metropolis-Hastings rule

$$\mathcal{T}^{i+1} = \begin{cases} \mathcal{T}^*, & \text{with probability } \alpha(\mathcal{T}^*, \mathcal{T}^i) = \min\left(\frac{q(\mathcal{T}^*,\mathcal{T}^i)}{q(\mathcal{T}^i,\mathcal{T}^*)} \frac{\Pr(Y|X,\mathcal{T}^*)}{\Pr(Y|X,\mathcal{T}^i)} \frac{\Pr(\mathcal{T}^*)}{\Pr(\mathcal{T}^i)}, 1\right) \\ \mathcal{T}^i, & \text{with probability } 1 - \alpha(\mathcal{T}^*, \mathcal{T}^i). \end{cases} \tag{22}$$

To evaluate the normalization constant would require summing Equation 21 across all possible trees. This is a sum with $\mathcal{O}(nd\frac{4^h}{h^{3/2}})$ terms, with h denoting the maximum height of the trees, n denoting the number of observations, and d denoting the number of covariates. This is an infeasible sum for most data sets, and for all data sets examined in this thesis. For the function $q(-|-)$, which

is called the proposal function, we use q to propose a new tree $\mathcal{T}^*$. In Equation 46, $q(\mathcal{T}|\mathcal{T}^*)$ denotes proposing a new tree $\mathcal{T}^*$, starting from the current tree $\mathcal{T}$. Our proposal mechanism is as follows:

- The grow step chooses at random one of the terminal nodes and proposes to append two new child nodes with a certain probability that could depend on the tree depth, splitting on a chosen covariate.

- The prune step works in reverse of the grow. A terminal node is selected at random and that node and the node's sibling are pruned to the immediate parent of the two child nodes.

- The change step randomly picks an internal node and attempts to change the split rule at the node with that of another observation, possibly on a different covariate.

- The swap step randomly selects an internal node that is not the root node and proposes to swap the split rules of the parent-child pair. If both child nodes' split on the same covariate, then both children and the parent node's rules are swapped.

- The rotate step randomly chooses a left or right rotation move. Then this step randomly chooses an admissible internal node and rotates.

The rotate operation for binary trees was first introduced in Sleater and Tarjan [84] and was introduced into Bayesian decision trees in GL [49]. A good introduction and several practical uses of the rotate move can be found in Cormen, Lieserson, Rivest and Stein [24]. The proposal of Gramacy and Lee [49] only allows a rotate move for the specific case when a swap move is proposed and the parent child pair both split on the same covariate. We modify this and allow rotate to be a separate operation of the transition kernel and not a special swap move case. The proposal mechanism of CGM uses the grow, prune, change and swap moves only. We also allow swap moves in our proposal. In addition, neither of these papers included weights on each covariate in their examples or model specifications. They sampled each covariate and split value uniformly, at random.

The probability measure on the tree is defined as

$$\Pr(\mathcal{T}) = \prod_{\eta \in \mathcal{T}} p_{\text{rule}}(\rho|\eta, \mathcal{T}) p_{\text{split}}(\eta, \mathcal{T}). \tag{23}$$

The probability measure on each split, here denoted $p_{\text{split}}(\eta, \mathcal{T})$, uses Equation 18. Similarly the measure on each rule, here denoted $p_{\text{rule}}(\rho|\eta, \mathcal{T})$, uses Equation 19. All that is left to specify is

the likelihood model in each node and the prior structure for the parameters in each node. This is done in the next subsection.

2.2.4 Node Likelihoods and Priors

CGM discuss three models. Two of the models use Gaussian priors and Gaussian likelihoods and one of the models uses a Dirichlet prior and a multinomial likelihood. The two Gaussian models differ in that one has a single variance and the other has a different variance for each node. As noted by Lee [65], in a greedy optimization context, sometimes the data suggest a different model than either a Gaussian or a multinomial-Dirichlet. If the experiment suggests analyzing data using an alternate model, the Bayesian context easily handles these alterations, once the corresponding likelihood and prior are specified. In the case of Lee [65], a zero inflated poisson (ZIP) model was proposed to analyze the solder data. Our Bayesian model can easily handle extensions such as this and also permits covariate selection, provided the integrated likelihood is available in closed form.

We begin with the Gaussian likelihood and Gaussian prior model. We define the likelihood as

$$N[y_{ij}|\mu_i, \sigma^2]. \tag{24}$$

Also, we define the prior for μ_i as

$$N[\mu_i|\bar{\mu}, \sigma^2]. \tag{25}$$

Furthermore, we define the prior for σ^2 as

$$\text{Inv-Gamma}(\sigma^2|\alpha, \beta). \tag{26}$$

All that remains is to evaluate the integral

$$\prod_{i=1}^{b} \int_0^\infty \int_{-\infty}^\infty \prod_{j=1}^{n_i} N[y_{ij}|\mu_i, \sigma^2] N[\mu_i|\bar{\mu}, \sigma^2] \text{Inv-Gamma}(\sigma^2|\nu/2, \nu\lambda/2) d\mu_i d\sigma^2. \tag{27}$$

For this specific prior and likelihood we can explicitly calculate the marginal likelihood. Being able to marginalize the node parameters explicitly allows us to implement a Metropolis-Hastings algorithm without resorting to complicated, specialized algorithms, or numerical integrations. Straightforward analytic manipulations yield the solution to Equation 27 written here in Equation 28

$$\frac{c a^{b/2}}{\prod_{i=1}^{b} \sqrt{n_i + a}} \times \left(\sum_{i=1}^{b} \left(\sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 \right) + \frac{(\bar{y}_i - \bar{\mu})^2 (n_i a)}{n_i + a} \right)^{-(\nu+n)/2}. \tag{28}$$

Assuming instead that the variances might change from node to node, then the stated model is misspecified. Let us denote the variance in each node as σ_i^2 and keep all other notations from the stated model specification. Then the model is specified using

$$N[y_{ij}|\mu_i, \sigma_i^2]. \tag{29}$$

Also, we define the prior for μ_i as

$$N[\mu_i|\bar{\mu}, \sigma_i^2]. \tag{30}$$

Furthermore, we define the prior for the σ_i^2s as

$$\text{Inv-Gamma}(\sigma_i^2|\nu/2, \nu\lambda/2) \tag{31}$$

and now we evaluate the integral equation

$$\prod_{i=1}^{b} \int_0^\infty \int_{-\infty}^\infty \prod_{j=1}^{n_i} N[y_{ij}||\mu_i, \sigma_i^2] N[\mu_i|\bar{\mu}, \sigma_i^2] \text{Inv-Gamma}(\sigma_i^2|\nu/2, \nu\lambda/2) d\mu_i d\sigma_i^2. \tag{32}$$

The result of computing the integrals in Equation 32 is

$$\prod_{i=1}^{b} \pi^{n_i/2} (\lambda\nu)^{\nu/2} \sqrt{\frac{a}{n_i + a}} \frac{\Gamma((n_i + \nu)/2)}{\Gamma(\nu/2)} \times \left(\sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 + \frac{(\bar{y}_i - \bar{\mu})^2 (n_i a)}{n_i + a} + \nu\lambda \right)^{(n_i+\nu)/2}. \tag{33}$$

These are the two "regression" models for Bayesian decision trees given in CGM [21].

The classification model discussed in CGM [21] defines the likelihood, prior, and integrated likelihood as

$$y_{i1}, \ldots, y_{in_i}|\mathcal{T} \sim \text{Multinomial}(Y_i|\underline{n}, \underline{p}), \tag{34}$$

$$\underline{p}|\mathcal{T} \sim \text{Dirichlet}(\underline{p}|\underline{\alpha}), \tag{35}$$

and

$$\Pr(Y|\mathcal{T}, X) = \left(\frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \right)^b \prod_{i=1}^{b} \left(\frac{\prod_{k=1}^K \Gamma(n_{ik} + \alpha_k)}{\Gamma(n_i + \sum_{k=1}^K \alpha_k)} \right), \tag{36}$$

respectively.

If we wanted to model the data using a different data generating process, for example a ZIP. We could do so by specifying a different likelihood, prior, and computing the integrated likelihood. For a ZIP model this is possible using gamma priors for the rate (λ) and beta priors for zero inflation components (ϕ).

2.2.5 A Bayesian Zero Inflated Poisson Model

Lee and Jin [65] reconsidered impurity functions in light of the connection to likelihood functions. Lee and Jin [65] proposed to use likelihood functions instead of impurity functions that model the data generating process. Towards this end they considered the soldering data from Chambers and Hastie [20]. The response of interest in this case is a collection of counts on manufactured circuit boards. This response has many zero's and Lee and Jin [65] propose using a zero inflated (ZIP) poisson likelihood to model the measured counts. Lee and Jin [65] optimized using a greedy algorithm and they found the fit and holdout prediction to be better using the ZIP model in each terminal node. If we are to use a Bayesian approach to this problem we need to define the likelihood, the prior, and the integrated likelihood. We now define these three quantities.

The likelihood for a single observation is

$$f(y|\lambda,\phi) \propto \mathbb{1}(y=0)\left(\phi+(1-\phi)\exp\left(-\lambda\right)\right)+\mathbb{1}(y>0)\left(\exp\left(-\lambda\right)\frac{\lambda^y}{y!}\right). \tag{37}$$

The priors for λ and ϕ are

$$\pi(\phi,\lambda) \propto \underbrace{\frac{\phi^{\alpha-1}(1-\phi)^{\beta-1}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}}_{=\text{A beta prior}} \times \underbrace{\frac{\lambda^{\alpha_\lambda-1}\exp\left(-\lambda/\beta_\lambda\right)}{\Gamma(\alpha_\lambda)\beta_\lambda^{\alpha_\lambda}}}_{=\text{A gamma prior}}. \tag{38}$$

Now we need to calculate the integrated likelihood, which means we must evaluate

$$\begin{aligned}&\int_0^1\int_0^\infty\left(\mathbb{1}(y=0)\left(\phi+(1-\phi)\exp\left(-\lambda\right)\right)+\mathbb{1}(y>0)\left(\exp\left(-\lambda\right)\frac{\lambda^y}{y!}\right)\right) \\ &\frac{\phi^{\alpha-1}(1-\phi)^{\beta-1}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\times\frac{\lambda^{\alpha_\lambda-1}\exp\left(-\lambda/\beta_\lambda\right)}{\Gamma(\alpha_\lambda)\beta_\lambda^{\alpha_\lambda}}d\lambda d\phi.\end{aligned} \tag{39}$$

Let j index the observed zero counts. Furthermore, let $\bar{y}_+$ denote the average of the non-zero counts and n_0 and n_+ denote the number of zeros and non-zeros in the data respectively. Now we assume that the observations are $i.i.d.$ and simple calculations lead to the conclusion that

$$\begin{aligned}\Pr(Y|X,\mathcal{T}) = &\left[\sum_{j=0}^{n_0}\binom{n_0}{j}\frac{\Gamma(\alpha+\beta)\Gamma(\alpha+j)\Gamma(n_0+\beta-j)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha+\beta+n_0)}\left(\frac{n_0-j+\beta_\lambda^{-1}}{\beta_\lambda}\right)^{\alpha_\lambda}\right] \\ &+\frac{\Gamma(\alpha_\lambda+n_+\bar{y}_+)}{\Gamma(\alpha_\lambda)\beta_\lambda^{\alpha_\lambda}}\left(n_++1/\beta_\lambda\right)^{\alpha_\lambda+n_+\bar{y}_+}.\end{aligned} \tag{40}$$

2.3 Previous Variable Selection

Previous approaches to variable selection have focused primarily on the linear model or GLM or GLMM models, all of which we briefly reviewed in Chapter 1. Hereafter we will refer to all three types of models as linear. The correspondence between variable selection in linear models and in Bayesian decision trees is a simple correspondence between zeros in the linear model, or zero means of the normal and transformed values on the unit simplex. This correspondence will be detailed in the next subsection of this chapter.

Ishwaran et al. [59] propose a modification to Variable important (VIMP) criteria that allows some very basic theory to describe the VIMP's analytical properties. VIMP was first proposed by Breiman [10] as a method to assess which covariates are important in a dataset. The difficulty with both Ishwaran et al's method and Breiman's method is that they are both very much black box techniques. VIMP basically takes a split in a decision tree on a specific covariate and randomly decides if the observation should go to the left or the right child node. This is done for a collection of observations and the random predictions are averaged against the actual predictions. The resulting numeric estimates for each covariate are the VIMP estimates. A ranking of these values provides a ranking of the covariates.

A similar method of ranking covariates was proposed by Taddy, Gramacy and Polson [86]. Taddy, Gramacy and Polson proposed a Bayesian approach and used the Bayesian equivalent to $\Delta\phi$ from this chapter. In the Bayesian approach the quantity $\Delta\phi$ really has little meaning, because we are using an MCMC approach to search across trees. The resulting estimates of the covariate's importance will be basically the same as the greedy approach. Something we find undesirable for many reasons. These methods are worth comparing against our approach and we will do so in future work.

$$\Delta\phi = \int \phi ds - \pi_L \int \phi ds - \pi_R \int \phi_R ds. \tag{41}$$

Taddy, Gramacy and Polson propose using samples and estimating the integrals with Monte Carlo approximations. In this case the approach is using a greedy measure on a Bayesian problem, something we find confusing.

2.4 Derivations

This subsection contains the mathematical details for calculating the integrals to get the closed form solutions for the integrated likelihood equations in this chapter.

2.4.1 ZIP Derivations

In this section we provide the derivation for the integrated likelihood for the Bayes ZIP (zero inflated poisson) tree model. Now let us define some notation: j will index either all observations or only the observed zero count observations if the upper limit is n_0 then j will index observed zero counts only, if the upper index limit is n then all observations are indexed. Also j' will index the non-zero observations. The total number of non-zero observations is denoted n_+, so that $n_+ + n_0 = n$. Finally let $\bar{y}_{i+}$ denote the sample mean of the non-zero count observations in terminal node i.

$$\begin{aligned}
\Pr(Y|X,\mathcal{T}) &= \prod_{i=1}^{b}\int_0^1\int_0^\infty \prod_{j=1}^{n_i}\left[\mathbb{1}[y_{ij}=0](\phi+(1-\phi)\exp(-\lambda)) + \mathbb{1}[y_{ij}>0]\frac{\exp(-\lambda)\lambda^{y_{ij}}}{y_{ij}!}\right]\pi(\phi_i,\lambda_i)d\lambda_i d\phi_i \\
&= \prod_{i=1}^{b}\int_0^1\int_0^\infty \underbrace{\prod_{j=1}^{n_0}(\phi+(1-\phi)\exp(-\lambda))\pi(\phi_i,\lambda_i)d\lambda_i d\phi_i}_{=(1)} \\
&+ \prod_{i=1}^{b}\int_0^1\int_0^\infty \underbrace{\prod_{j'=1}^{n_+}\frac{\exp(-\lambda)\lambda^{y_{ij'}}}{y_{ij'}!}\pi(\phi_i,\lambda_i)d\lambda_i d\phi_i}_{=(2)}.
\end{aligned}$$

We will first tackle (1), then tackle (2).

$$\begin{aligned}
(1) &= \int_0^1\int_0^\infty \prod_{j=1}^{n_0}(\phi+(1-\phi)\exp(-\lambda))\pi(\phi_i,\lambda_i)d\lambda_i d\phi_i \\
&= \int_0^1\int_0^\infty (\phi+(1-\phi)\exp(-\lambda))^{n_0}\pi(\phi_i,\lambda_i)d\lambda_i d\phi_i \\
&= \int_0^1\int_0^\infty \sum_{j=1}^{n_0}\binom{n_0}{j}\phi^j(1-\phi)^{n_0-j}\exp(-(n_0-j)\lambda))\pi(\phi_i)\pi(\lambda_i)d\lambda_i d\phi_i.
\end{aligned}$$

Now we take $\pi(\phi_i)$ to be a beta(α, β) prior and $\pi(\lambda_i)$ to be a gamma$(\alpha_\lambda, \beta_\lambda)$ prior. This simplifies matters greatly.

$$\int_0^1 \int_0^\infty \sum_{j=1}^{n_0} \binom{n_0}{j} \phi^j (1-\phi)^{n_0-j} \exp\left(-(n_0-j)\lambda\right)) \frac{\Gamma(\alpha+\beta)\phi^{\alpha-1}(1-\phi)^{\beta-1}}{\Gamma(\alpha)\Gamma(\beta)} \frac{\lambda^{\alpha_\lambda-1}\exp\left(-\lambda/\beta_\lambda\right)}{\Gamma(\alpha_\lambda)\beta_\lambda^{\alpha_\lambda}} d\lambda_i d\phi_i$$

$$= \sum_{j=1}^{n_0} \binom{n_0}{j} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha_\lambda)\beta_\lambda^{\alpha_\lambda}} \underbrace{\int_0^1 \phi^{j+\alpha-1}(1-\phi)^{\beta+n_0-j-1} d\phi_i}_{\text{a beta kernel}} \underbrace{\int_0^\infty \lambda^{\alpha_\lambda-1}\exp\left(-(n_0-j+\beta_\lambda^{-1})\lambda\right) d\lambda_i}_{\text{a gamma kernel}}$$

$$= \underbrace{\sum_{j=1}^{n_0} \binom{n_0}{j} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(\alpha_\lambda)\beta_\lambda^{\alpha_\lambda}} \frac{\Gamma(\alpha+j)\Gamma(\beta+n_0-j)\Gamma(\alpha_\lambda)}{\Gamma(\alpha+\beta+n_0)} (n_0-j+\beta_\lambda^{-1})^{\alpha_\lambda}}_{=(1)}.$$

Now with the first piece simplified we move on to piece (2).

$$\begin{aligned}
(2) &= \int_0^1 \int_0^\infty \prod_{j'=1}^{n_+} \frac{\exp\left(-\lambda\right)\lambda^{y_{ij'}}}{y_{ij'}!} \pi(\phi_i, \lambda_i) d\lambda_i d\phi_i \\
&= \int_0^\infty \prod_{j'=1}^{n_+} \frac{\exp\left(-\lambda\right)\lambda^{y_{ij'}}}{y_{ij'}!} \pi(\lambda_i) d\lambda_i \\
&= \int_0^\infty \frac{\exp\left(-n_+\lambda\right)\lambda^{n_+\bar{y}_{i+}}}{\prod_{j'=1}^{n_+} y_{ij'}!} \pi(\lambda_i) d\lambda_i \\
&= \int_0^\infty \frac{\exp\left(-n_+\lambda\right)\lambda^{n_+\bar{y}_{i+}}}{\prod_{j'=1}^{n_+} y_{ij'}!} \frac{\lambda^{\alpha_\lambda-1}\exp\left(-\lambda/\beta_\lambda\right)}{\Gamma(\alpha_\lambda)} d\lambda_i \\
&= \frac{\int_0^\infty \exp\left(-(n_+ + \beta_\lambda^{-1})\lambda\right)\lambda^{n_+\bar{y}_{i+}+\alpha_\lambda-1} d\lambda_i}{\Gamma(\alpha_\lambda)\prod_{j'=1}^{n_+} y_{ij'}!} \\
&= \underbrace{\frac{\Gamma(n_+\bar{y}_{i+}+\alpha_\lambda)(n_+ + \beta_\lambda^{-1})^{n_+\bar{y}_{i+}+\alpha_\lambda}}{\Gamma(\alpha_\lambda)\prod_{j'=1}^{n_+} y_{ij'}!}}_{=(2)}.
\end{aligned}$$

And the result is shown.

3 Dirichlet Variable Selection For Decision Trees: The DiVaS method

Many data analysis procedures, originally designed for inference in low dimensional spaces, create difficulties when dealing with high dimensional data. Often data analysts use low dimensional intuition and apply similar logic to high dimensional data, which can often lead to erroneous conclusions. The approach outlined in this thesis not only moves around the high dimensional space efficiently, but also indicates with a high degree of probability which covariates are useful in determining the response at the tree nodes. In addition, we retrieve summary data indicating which dimensions are useful assisting the data analyst in the interpretation of results. Our primary focus in this thesis is on inferring a structural relationship between dimensions and the response of interest, with a preference towards interpretability.

There are perhaps two predominant approaches to dealing with high dimensional data. The first attempts to only include dimensions that are important based on some metric of choice, usually with a filter run over the data to screen out some dimensions. Examples of this are principal components analysis [88] and various other approaches surveyed by Dash and Liu [27]. The second approach attempts to fit a model to the best dimensions sequentially. Examples of this approach include forward, backward and stagewise searches for regression models [71] [75] [3]. The lasso [87] is a problem which can be considered in either category, depending upon the approach taken. If we use the LARS algorithm to fit the model for all values of λ then the method can be considered iterative. If instead, we fit a model for the lasso objective function choosing a single value of λ, for example by cross validation, then the method can be though of as a filtering approach. Our approach is similar to both approaches and operates by choosing a middle ground between these two extremes. As a byproduct of choosing this middle ground we are able to search the space of models more effectively and give intuitive summary results indicating which covariates are useful, while returning models to the analyst for use in prediction and inference.

In this chapter we follow a Bayesian framework. We define a new prior measure over trees, which facilitates comparison across trees based on a global search of the tree space, and automatically induces coherent covariate selection. We assume a working familiarity with Markov chain Monte Carlo (MCMC) methods. For a good review of MCMC approaches to machine learning problems, we recommend the paper by Andrieu, De Freitas, Doucet and Jordan [1]. We show through an example that greedily grown decision trees can fail to recover good trees in high di-

mensional data. Moreover, we find that previously implemented Bayesian approaches to decision tree learning have been applied only to low dimensional spaces [21] [49] [28]. Applying the same approaches to high dimensional data is inappropriate and we demonstrate this through examples. These methods fail to move around the large dimensional space efficiently causing poor mixing and only finding local optima.

Nearly all the previous approaches to decision tree induction have not explicitly incorporated model selection into the tree framework [21] [49] [28] [11] [82]. A notable exception is the recent paper by Gramacy and Taddy [50] where binary indicators are used to perform model selection within a terminal node. Specifically, Gramacy and Taddy define a mixture of a Gaussian process (GP) and the GP's limiting linear model, and use a point mass mixture to choose between the GP and the limiting linear model within each terminal node. This is in contrast with our approach, which aims to perform variable selection at the level of the decision tree itself and not for each model in each terminal node. Another notable exception is the work of Ishwaran, Kogalur, Gorodeski, Minn, and Lauer, who define a new quantity called a maximal subtree, and use the inherent topology of the tree and the maximal subtree to measure variable importance [59]. Ishwaran et al. work in the context of random survival forests, built using bootstrapped data. Most researchers have used implicit selection, for which predictors are chosen at the end of the greedy induction strategy as the predictors which are useful. Those not selected are considered not useful. Currently, most tree algorithms rely on a pruning rule for determining which covariates are useful. This implicit approach has the drawback of only providing "yes/no" answers to inclusion, and does not allow one to order the dimensions in some meaningful fashion. In our work we explicitly model these probabilities of inclusion and exclusion in the decision tree. This not only allows for useful predictor selection when choosing models, but also improves the efficiency of searching in large dimensions. We note that we include a prune function in our proposal and this does affect the selection of which covariates are useful. However, in large dimensions, the more efficient way to improve the efficiency of the searching algorithm is to treat covariates differently based on their utility in the model.

This chapter follows from the original Bayesian decision tree model of Chipman, George and Mcculloch [21] (referred to hereafter as CGM), and Gramacy and Lee [49](hereafter referred to as GL). Specifically, we propose a generalization of the models used by CGM and of GL. We allow rotate moves at all permissible nodes of the tree and varying selection probabilities on each covariate. In this manuscript we confine ourselves to categorical response data. All results can

easily be extended and applied similarly to continuous data. In addition, we study the approach of CGM to large dimensional datasets. To our knowledge, the largest dimensions of data that the aforementioned approaches have been studied on are at most 15 dimensions. We find our method to be more effective than the CGM approach for large dimensional data. We also study the efficacy of a general rotate move in the transition kernel of the Markov chain.

The remainder of this paper unfolds as follows. Section 3.1 reviews previous greedy and Bayesian approaches to decision tree learning. Section 3.2 shows our model and describes how this is a modification of the algorithm proposed by CGM [21]. In this and the next chapter we place emphasis on the algorithmic details of our procedure and only briefly describe some of the mathematical properties. Section 3.3 states necessary conditions for consistency of the decision trees we propose. Section 3.4 shows how high dimensional data can create problems for both greedy and Bayesian approaches that have been previously proposed, and gives an example. Section 4 applies our approach to a dataset taken from the UCI machine learning repository. In Section 4.1 we state conclusions and point towards future work.

3.1 Related Work

Some of the most well known approaches to decision tree learning are two approaches invented in the 1980's; these are Breiman, Friedman, Olshen, and Stone's cart method [11] and Quinlan's C4.5 algorithms and variants [82]. The work of Breiman et al. [11] showed the practical and theoretical approaches to decision tree induction. In Breiman et al. [11], the theoretical framework for discussing decision trees is given in terms of general impurity functions, allowing results to apply to specific impurities suggested by Quinlan [82]. For a theoretical treatment of consistency, the work of Devroye, Lugosi, and Gyöfi [29] provided the foundation and application of the theory to certain tree methods.

The consistency of tree classifiers was also discussed by Breiman et al. [11]. They showed that, under fairly general conditions, tree classifiers are asymptotically consistent, meaning trees can recapture the correct classifier if they are given enough data. The practical problem with this is that we usually only have a finite amount of data. Although we do not overcome this practical limitation, the results in Section 3.3 improve the rate of convergence by exploiting sparsity. Quinlan's approach modified the impurity functions proposed by Breiman [11], and also allowed for multiway splits at nodes in the tree [82]. Empirically, we have found that the entropy impurity

function approach of Quinlan [82] performs better if there is a difference, although there is usually little or no difference in terms of the final tree.

In our probabilistic approach, we quantify *a priori* uncertainty in the classifier with a prior probability measure over trees. By defining a prior over trees, we induce a posterior distribution over trees via Bayes' rule shown in Equation 42, where θ denotes parameters in the model and D denotes observed data.

$$\Pr(\theta|D) = \frac{\Pr(D|\theta)\Pr(\theta)}{\Pr(D)} \tag{42}$$

The posterior allows us to *search* the space of trees, which is distinctly different from the greedy induction approach. In the Bayesian approach to decision tree induction several researchers have proposed various methods for searching over the tree space. The space of trees is a large discrete space, so MCMC algorithms are usually applied to search this space. CGM [21] proposed a clever process prior on the tree space, denoted $\Pr(\mathcal{T})$, to induce a posterior which moves between a varying number of dimensions. Let $\mathcal{T}$ denote the tree and θ the collection of parameters in each terminal node (mean or class probabilities) and D denotes data. Applying Bayes' rule, we have the result

$$\Pr(\mathcal{T}|D) = \frac{\int \Pr(D|\theta,\mathcal{T})\Pr(\theta|\mathcal{T})\Pr(\mathcal{T})d\theta}{\Pr(D)}. \tag{43}$$

Note that the integral in the numerator of Equation 43 treats the tree prior as a constant with respect to θ. The integrated product of the other two quantities in the numerator is called the integrated likelihood and will be referred to in Section 3.2. In the same year, Denison, Mallick, and Smith [28] proposed a reversible jump algorithm that has also shown success in searching tree space. As modifications to the work of CGM [21], GL [49] proposed a Gaussian process model in each terminal node, along with modifying the transition kernel to include rotations in one special case of swap moves.

Bagging and boosting are two other common methods that might be used for performing model selection in the context of decision tree induction [9] [40]. Often analysts will use these algorithms as black boxes and use the frequency of selected covariates in the resulting trees as measures of importance of decision trees. In the bagging context and the related random subspace method [55], one randomly subsamples either observations or dimensions and builds trees on this subsampled data. One of the difficulties with data containing many predictors is that, when we randomly sample with replacement collections of predictors to build trees, we select approximately 63% of the dimensions [35]. This is usually still a large feature space inheriting the same problems we initially

faced. Bagging and boosting methods are primarily used in the context of prediction. As initially stated, our primary motivation is on inferring a mechanistic relationship between dimensions and the response of interest with a preference towards simple and easily understood models. Hence, inference is the primary goal, with prediction being secondary.

3.2 Model Details

In this section, after we define our approach, we detail the transition kernels proposed by CGM [21] and by GL [49]. We highlight the differences between their approaches and our approach.

Let us define the prior measure over a decision tree as

$$\Pr(\mathcal{T}) = \prod_{\eta \in N} \Pr{}_{\text{split}}(\eta) \Pr{}_{\text{rule}}(\eta, s_k),$$

where η ranges over all nodes in the tree, and the two probabilities are probabilities of a split at the node η, and the specific rule at node η, respectively. Also, s_k denotes the randomly selected covariate. The probability of a split in a tree is

$$\Pr{}_{\text{split}}(\eta) = \alpha(1 + d_\eta)^{-\beta},$$

where the quantities $\alpha > 0$ and $\beta \geq 0$ are parameters, and $d_\eta = \lfloor lg(\eta) \rfloor$ is the depth of the node η under the usual binary heap node numbering (cf. Cormen, Lieserson, Rivest and Stein [24]). In this document lg denotes a base 2 logarithm, and log denotes a base e logarithm.

For our model we rely on the multinomial Dirichlet conjugate pair

$$\Pr(\underline{p}) \Pr(c_i|\underline{p}) = \underbrace{K \prod_{i=1}^{d} p_i^{\alpha_i - 1}}_{\text{Dirichlet}} \times \underbrace{\binom{n}{c_1, \ldots, c_d} \prod_{i=1}^{d} p_i^{c_i}}_{\text{Multinomial}},$$

where the c_i's are counts and K is a normalizing constant.

The integrated likelihood quantity from the numerator of Equation 43 is a multinomial Dirichlet conjugate pair. Therefore, the integral is available in closed form. We take the Dirichlet parameters (α_is) to be a vector of 1s although other alternatives are possible. The $\Pr_{\text{rule}}(\eta, s_k)$ is decomposed into two components, one selecting the jth covariate to use to propose a split at the node, and a second component selecting a specific split value. This corresponds to the conditional probability decomposition

$$\Pr{}_{\text{rule}}(\eta, s_k) = \underbrace{\Pr{}_{\text{cov}}(s_k)}_{\text{multinomial}} \Pr{}_{\text{splitvalue}}(\eta|s_k). \tag{44}$$

The multinomial distribution in Equation 44 to be a multinomial and we define a prior of choosing covariates to split on as a Dirichlet distribution. These two probability densities are conjugate pairs. This is brought about by assuming the structure $\Pr_{\text{cov}}(s_k) \equiv \Pr_{\text{cov}}(s_k|s_{k-1}, s_{k-2}, \ldots, s_1)$. To our knowledge, this refinement of the CGM model has not been employed in the literature. The work of Ishwaran et al. used the exact same breakdown as Equation 44 but proposed a different probability density on the covariates. Also Ishwaran et al.'s method was used in the context of random survival trees, decision trees made using bootstrap sampled survival data [59].

When we sample a Markov chain we collect trees sampled under the dynamics of Equation 46, a Metropolis-Hastings rule [19]. We often find that the observed counts of splits on each covariate drown out the information from the prior. This is an undesirable effect of the proposed model, because the observed split counts are not observed *a priori*, but during the course of the MCMC algorithm. To combat this, we find it necessary to define the parameter $\widetilde{\alpha}$. This parameter is defined by setting the prior expectation of the probability of splitting proportional to the expectation of the unobserved split probabilities on a covariate written here in Equation 45

$$\widetilde{\alpha} = \frac{C \sum_{j=1}^{d} \alpha_j}{\sum_{i,j} s_{ij}}. \tag{45}$$

The α_j are the current concentration parameters of the Dirichlet distribution and s_{ij} denotes the observed frequency of splits on dimension j at iteration number i. This implies that the posterior for the covariate weights follows a modified Dirichlet distribution,

$$\Pr(\underline{p}|\alpha, \underline{s}_i) = K(\widetilde{\alpha}) \binom{n}{s_{i1}, \ldots, s_{id}} \prod_{j=1}^{d} p_j^{\widetilde{\alpha} s_{ij}}.$$

Here $K(\widetilde{\alpha})$ is a normalizing constant. The extent to which the parameter $\widetilde{\alpha}$ (Equation 45) will impact the chain will depend on the length of each Markov chain. The parameter $\widetilde{\alpha}$ has a greater impact on chains that are run for a greater number of iterations.

The constant C in Equation 45 must be greater than zero and is a specified constant governing the ratio of exploration and exploitation conducted on the covariates of the Markov chain. All MCMC algorithms have the property of exploring the space until a region of high posterior probability is found. Once found, the algorithm exploits the local concavity of this probability region. The extent to which an MCMC algorithm exploits and explores will determine how well the posterior distribution is recovered. The higher the value of C, the more exploration the Markov chain will conduct, and the smaller the value, the more exploitation. In addition, for data sets of differing

dimensions, we may wish to use the dimensionality to guide the choice of the exploration and exploitation. Larger dimensional data sets will require a larger value of C to encourage exploration. If the analyst has good prior information in choosing α_j's, i.e. which covariates are useful, then the chain should move more towards exploitation, encouraging the Markov chain to find just the right splits and not focus too much on selecting the right covariates.

The approaches of CGM and GL used discrete uniform probability masses on each covariate and on each available split value in the current node. We modify this by allowing the selection of covariates to vary from covariate to covariate, while keeping the discrete uniform prior on split values. We take the prior measure on the covariates to be a Dirichlet distribution and the observed counts for splits in the tree from the Markov chain as the pseudo counts of a multinomial likelihood. Briefly, the benefits are:

- Improved searching of the Markov chain.
- Better inferential and predictive trees.
- Easier interpretation of classifiers.

We discuss the model and specification of parameters further in Section 3.4 and in Chapter 4.

The transition kernel describes the dynamics governing the state transitions of a Markov chain. To understand this let us introduce some notation, let X' denote the current state of the Markov chain, and X some new future state. Then $k(X'|X)$ denotes the probability of moving from the current state X' into the new state X, which may be viewed as a conditional density or mass function. In this paper, the states of the Markov chain are trees, denoted as$\mathcal{T}'$ (current) and $\mathcal{T}$ (new). The transition kernel $k(\mathcal{T}'|\mathcal{T})$ is defined by the Metropolis-Hastings rule

$$k(\mathcal{T}'|\mathcal{T}) = min\left(1, \frac{\Pr(\mathcal{T}|D)q(\mathcal{T}|\mathcal{T}')}{\Pr(\mathcal{T}'|D)q(\mathcal{T}'|\mathcal{T})}\right). \tag{46}$$

In Equation 46, $q(\mathcal{T}|\mathcal{T}')$ denotes a proposal function proposing a new tree $\mathcal{T}'$, starting from the current tree $\mathcal{T}$. Our proposal mechanism is as follows:

- The grow step chooses at random one of the terminal nodes and proposes to append two new child nodes with a certain probability that could depend on the tree depth, splitting on a chosen covariate.
- The prune step works in reverse of the grow, a terminal node is selected at random and that node and the node's sibling are pruned to the immediate parent of the two child nodes.

- The change step randomly picks an internal node and attempts to change the split rule at the node with that of another observation, possibly on a different covariate.
- The swap step randomly selects an internal node that is not the root node and proposes to swap the split rules of the parent-child pair. If both child nodes split on the same covariate, then both children and the parent node's rules are swapped.
- The rotate step randomly chooses a left or right rotation move. Then it randomly chooses an admissible internal node and rotates.

Sleater and Tarjan [84] introduced the rotate operation for binary trees in computer science. GL [49] first introduced the rotate operation into the Bayesian decision tree literature. A good introduction and several practical uses of the rotate move can be found in Cormen, Lieserson, Rivest and Stein [24]. The proposal of Gramacy and Lee [49] only allows a rotate move for the specific case when a swap move is proposed and the parent child pair both split on the same covariate. We modify this and allow rotate to be a separate operation of the transition kernel and not a special swap move case. The proposal mechanism of CGM uses the grow, prune, change and swap moves only. We also allow swap moves in our proposal. In addition, neither of these papers included weights on each covariate in their examples or model specifications. They sampled each covariate and split value uniformly at random.

3.3 Ensuring Consistent Classifiers

In this section we aim to answer two questions. The first question is what is the behavior of our method as the number of dimensions $d \to \infty$. We generally tend to think of data as having a fixed dimensionality, but as data storage costs decrease, the number of variables tends to increase. We assume that the data has a large number of dimensions and use the theoretical quantity $d \to \infty$ as a guidepost. We use a working example throughout this section, a stump tree, which is a tree containing only one split rule. The second working example is a decision tree with an arbitrary number of splits. Through these two examples we see when the theory discussed in this section works and when the theory fails.

We begin with a few definitions originally provided by Vapnik and Chervonekis [90], and subsequently studied by many others [85] [29]. Let us define the shatter coefficient as the maximum number of sets obtained by intersecting a finite collection of points with functions from a specified

class of functions and denote this as $s(\mathcal{A}, n)$. For our work it will suffice to look at half spaces on $\mathbb{R}^d$, which have shatter coefficient 2^n for $n < d$, but once $n > d$, the shatter coefficient no longer grows exponentially, instead growing polynomially in n. We say a class of functions $\mathcal{A}$ has Vapnik-Chervonekis (VC) dimension $V_{\mathcal{A}}$, where $V_{\mathcal{A}}$ denotes the largest value in the *exponent* of the shatter coefficient, such that the exponent equals the sample size. Formally, let us define this value as $V_{\mathcal{A}} \stackrel{def}{=} max\{n : s(\mathcal{A}, n) = 2^n\}$. Intuitively, the VC dimension denotes a phase change in the shattering dynamics as a function of the sample size, as seen, for example, in Figure 2. This phase change occurs when the growth in the shatter coefficient as a function of n, the sample size, changes from exponential to polynomial. For half spaces on $\mathbb{R}^d$, $V_{\mathcal{A}} = d$ [29]. We will use the shatter coefficient and VC dimension in error bounds stated in this section.

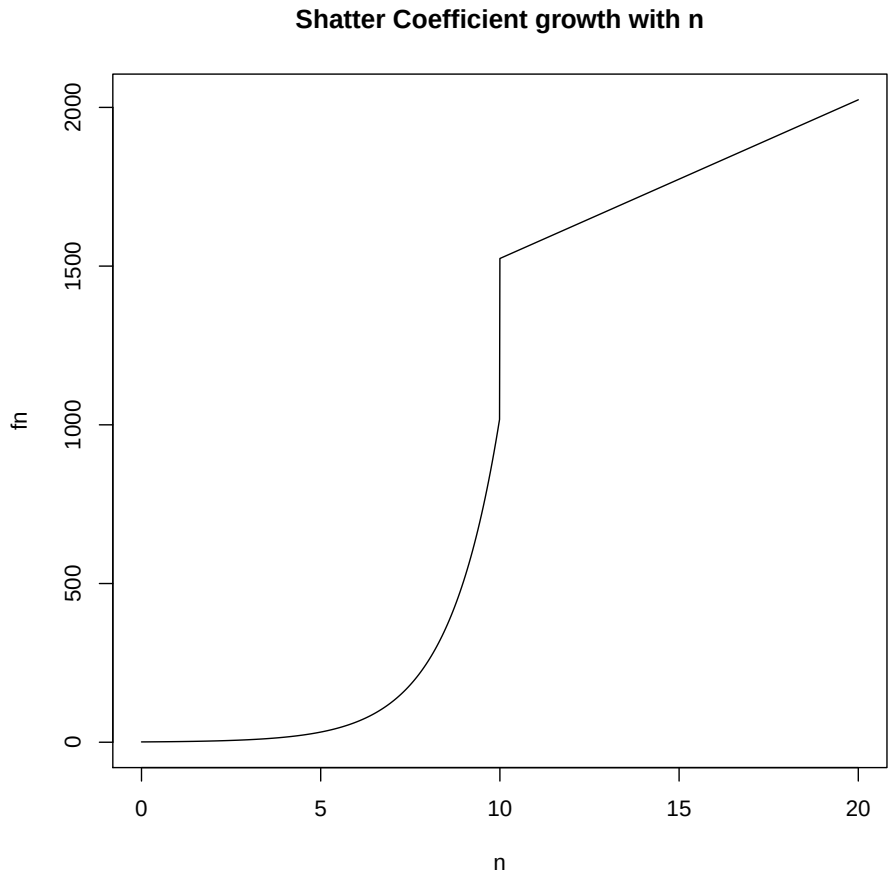


Figure 2: The shatter coefficient function for half spaces in $\mathbb{R}^d$, as a function of n. In this plot $d = 10$.

Devroye, Györfi, and Lugosi [29] provided the following result:

Theorem 1

For $0 < k^2/(k-1) < n\varepsilon^2$,

$$\Pr\left(\underset{A\in\mathcal{A}}{sup}|v_n - v| > \varepsilon\right) \leq 4ks(\mathcal{A}, n)\exp\left[\frac{-n\varepsilon^2}{2k^2}\right]. \tag{47}$$

The notation $v_n \overset{def}{=} (1/n)\sum_{i=1}^{n} \mathbb{1}(y_i = \hat{\mathcal{T}}(X_i))$ denotes the empirical error of the estimated classifier and the notation $v \overset{def}{=} \Pr(\mathcal{T}(X) = y)$ denotes the Bayes error of the data. This result tells us that if the family of classifiers has a finite VC dimension, or equivalently a finite shatter coefficient, then the classifier is consistent. Here k and ε are specified constants, usually we take $k = 2$ and $\varepsilon = 0.01$. Now we know that a halfspace split in $\mathbb{R}^d$ has $V_{\mathcal{A}} = d$. Let us define $\mathcal{H}(x) = -xlog(x) - (1-x)log(1-x))$ for $0 \leq x \leq 1$ with $\mathcal{H}(1) = \mathcal{H}(0) = 0$. A useful inequality is

$$s(\mathcal{A}, n) \leq \exp[n\mathcal{H}(V_{\mathcal{A}}/n)], \tag{48}$$

for $V_{\mathcal{A}} < 2n$. We see that, for $d < 2n$, this inequality holds and, thus, the right hand side of inequality 47 will go to 0, as $n \to \infty$. For our stump, the tree will capture the Bayes error as $n \to \infty$, if the model is correct. We also see that, for $n < d$, the shatter coefficient for half spaces is 2^n. If $d \to \infty$ and $n \to \infty$, but $n < d$, we have no consistency guarantee, according to this bound. In the second case, let us consider a decision tree of arbitrary depth. Theorem 13.5 of Devroye, Györfi, and Lugosi tells us that for $\mathcal{A} = \{\mathcal{A}_1 \cap \mathcal{A}_2\}$,

$$s(\mathcal{A}, n) \leq s(\mathcal{A}_1, n)s(\mathcal{A}_2, n). \tag{49}$$

For decision trees with two splits on halfspaces, the shatter coefficient is bounded above by $2^{2n} = 4^n$, for sample sizes less than d. For sample sizes larger than d we will have consistency as $n \to \infty$, provided that the decision tree does not grow arbitrarily large. As a simple requirement we ensure that the trees do not grow arbitrarily large, by ensuring that trees are shallower than K, for some constant K. Note that this constraint was also required by Denison, Mallick and Smith's reversible jump algorithm [28]. Our simulation studies indicate that there is little to no sensitivity in the specification of K. Figures 3 -9 show the results of simulating from the posterior distribution over trees and varying the maximum depth, K. We evaluate for values of $K = 2, 3, 4, 5, 6, 7, 10$. For small values of K such as 2 or 3, the setting of maximum depth appears to limit the sampler from exploring possible trees. However, once K is set larger than 6 there appears to be little impact restricting the sampler from exploring certain trees. Of course, for any given data set the value of

K that is acceptable will likely change, so our conclusion is that K should be large enough. Our assumption is $K = 20$ or $K = 30$ should be enough for most problems. Simply stated, K should be large. Too small values of K can cause problems. A value of $K = 10$ appears sufficient for our purposes and causes no problems.

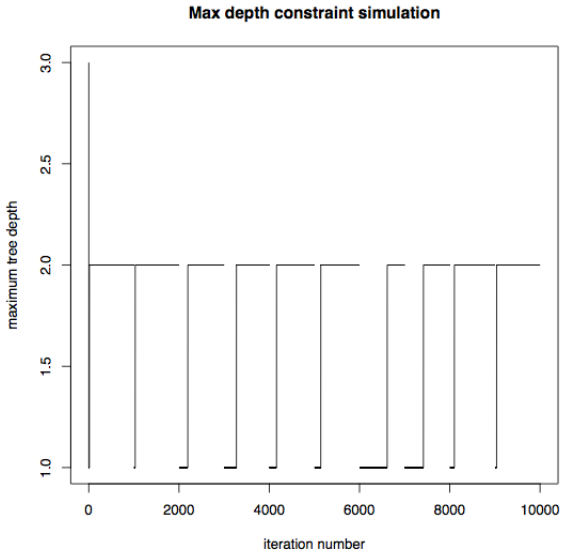


Figure 3: Maximum depth of samplers trees with maximum depth set at 2.

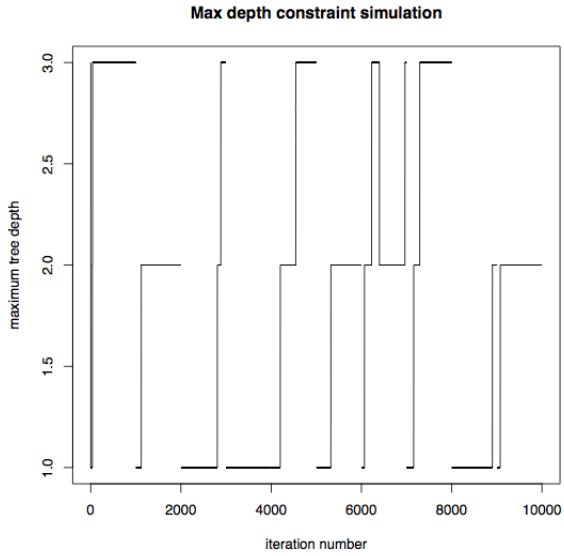


Figure 4: Maximum depth of samplers trees with maximum depth set at 3.

Up to this point, the material in this section is nothing new, but contained fully in the work of Devroye, Györfi, and Lugosi. Now let us define the sparse shatter coefficient as the shatter coefficient using only dimensions where $S_j = 1$, denoted $s(\mathcal{A}, n|S_{j=1}^d))$ and $S_j \in \{0, 1\}$. We now use these sparse definitions to carry over results into cases where sparsity holds in classification problems.

Theorem 2

For $0 < k^2/(k-1) < n\varepsilon^2$ and $\mathbb{E}(S_j) = p_j$, the bound

$$\Pr\left(\sup_{A\in\mathcal{A}}|v_n - v| > \varepsilon\right) \leq 4ks(\mathcal{A}, n|S_{j=1}^d)) \exp\left[\frac{-n\varepsilon^2}{2k^2}\right] \tag{50}$$

holds, if and only if $\sum_{j=1}^d p_j < \infty$, where d can be finite or infinite. In the case $d < \infty$ everything

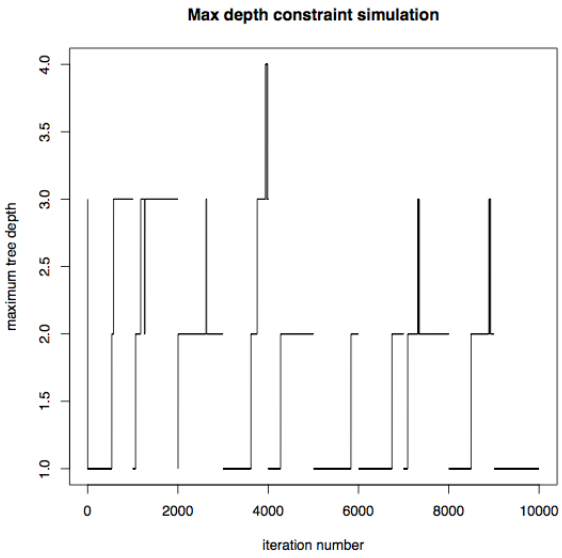


Figure 5: Maximum depth of samplers trees with maximum depth set at 4.

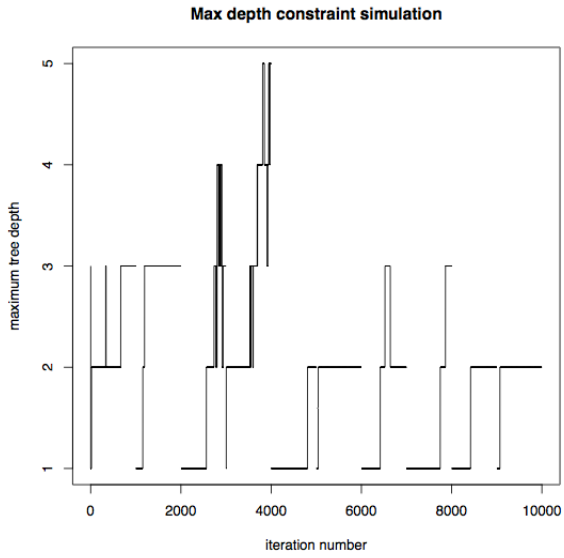


Figure 6: Maximum depth of samplers trees with maximum depth set at 5.

looks nearly the same and the proof follows verbatim from the derivation in Devroye, Györfi, and Lugosi. The case $d = \infty$ is delicate and requires Kolmogorov's three series theorem [68].

The second error bound (Equation 50) differs from the first (Equation 47) in that now we are only using some subset of the covariates to classify the observations. Explicitly, the number of effective dimensions, denoted as d^*, can be bounded almost surely, but the total number of dimensions $d \to \infty$. Each $S_j \in \{0, 1\}$, so if $\Pr(S_j = 1) = 1$, then we are back into the case of the first theorem, which is non sparse shattering. In our case, we relax the assumption that all $\Pr(S_j = 1) = 1$ and instead allow $\Pr(S_j = 1) = p_j$, for $0 \leq p_j \leq 1$. This can be thought of as a convex relaxation of a non-convex, computationally difficult optimization problem. The sum $\sum_j^d S_j$ is now a random sum, since each S_j is a random variable. Also, this random sum is the sparse VC dimension for splits of the form $(\infty, a_i]$, the splits we use to construct our classifiers. Passing to the limit as $d \to \infty$, we want the random infinite series to converge, otherwise we will not have a consistent classifier. The Kolmogorov three series theorem tells us that convergence almost surely of the random series $\sum_{j=1}^{\infty} S_j$ occurs if and only if the series $\sum_{j=1}^{\infty} p_j$ converges, thus necessary and sufficient conditions are $\sum_{j=1}^{\infty} p_j < \infty$.

We have necessary and sufficient conditions for convergence and therefore for consistency in

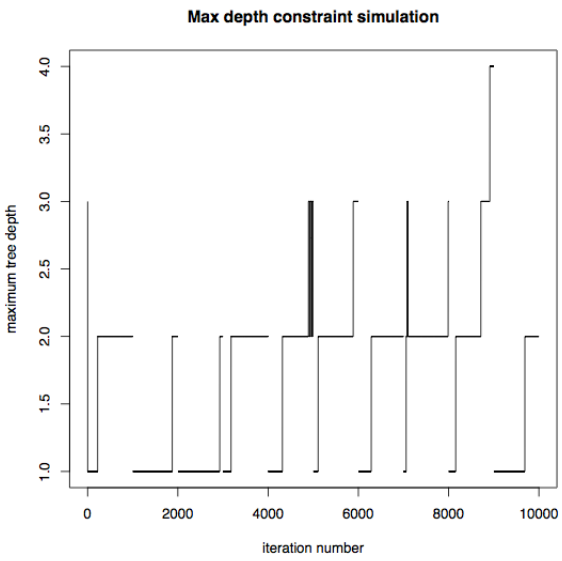


Figure 7: Maximum depth of samplers trees with maximum depth set at 6.

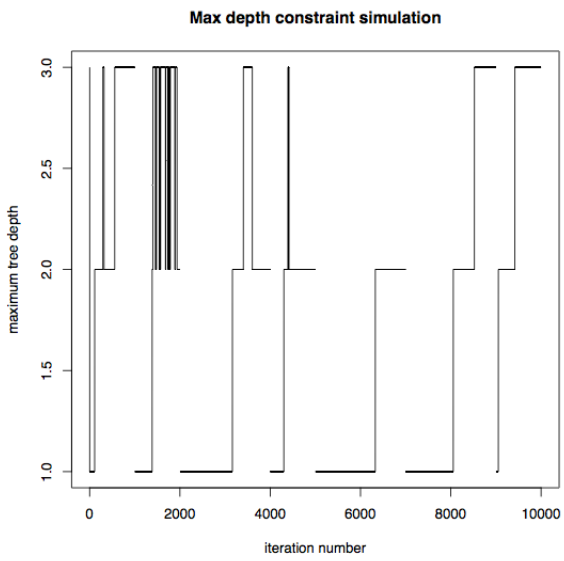


Figure 8: Maximum depth of samplers trees with maximum depth set at 7.

infinite dimensional data. We need only ensure that $\sum_{j=1}^{\infty} p_j < \infty$. Provided $\sum_{j=1}^{\infty} p_j < \infty$ the stump tree will be consistent with infinite dimensional data as $n \to \infty$ Furthermore, the tree with arbitrary depth will also be consistent, provided that the tree does not grow too deep, less than 2^K for some large but finite K. Intuitively this result should make sense. For infinite dimensional data, we must have most dimensions be irrelevant for the classification task, in order for the resulting classifiers to be consistent. In other words, most $p_j \to 0$ and some $p_{j'} \to c > 0$ $(0 < c < 1)$ as $d \to \infty$. Our approach, without the $\widetilde{\alpha}$, ensures precisely this condition. Furthermore, this is not a weakness of our approach, because this theoretical result holds for $d \to \infty$. In our real data cases d is finite and usually fixed. The $\widetilde{\alpha}$ is a practical aspect of our approach and is used to improve exploration and exploitation in fixed, finite dimensional data.

3.4 A Simulated Example

This section details a simulated example where we know which covariates are important and which are not important. Of course, this will never be the case in practice, yet, for verification, we find this setting useful.

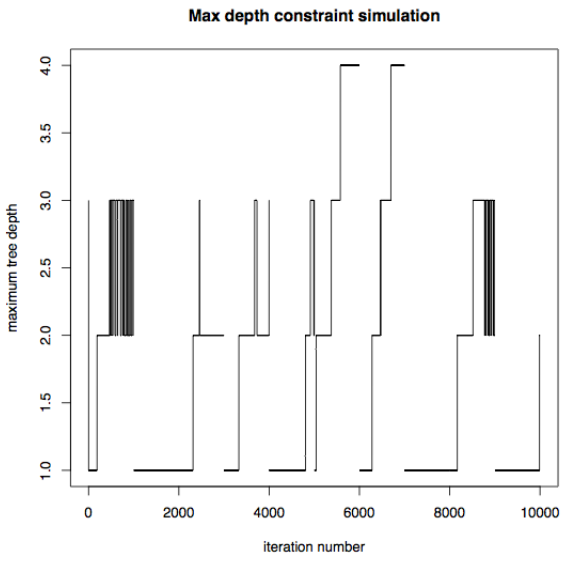


Figure 9: Maximum depth of samplers trees with maximum depth set at 10.

For our example, we note that a greedily built decision tree with a linear classifier can universally fail, if we are willing to make d large enough. To accomplish this for each covariate after d^*, we will simulate from a mixture of two normals, where each mixing probability is $1/2$, and the means of the two normals are taken from a continuous uniform distribution on a large enough region (l, u). If we can generate data where each covariate has observations in two groups that are separated enough and some of the responses are also separated, then we will incorrectly select a split rule based on this covariate, when in reality this covariate is independent of the response by construction. This will occur more frequently for larger d. Fortunately, our method will still select the correct covariates and as a result, select the correct tree.

Let us define the data generating process (DGP) as that given in Figure 10.

The probability of being the majority class in any of the four regions of the Figure 10 is 90%. Therefore no perfect classifier exists, yet very good classifiers are possible. We next generate additional covariates independently according to the mixture strategy described in the previous paragraph, with the continuous uniform supported on the interval $(0, 20)$. We look at the cases of $d = 100$ and 400. The trees generated by the three tree methods are shown in Figures 11-13. The best tree found by our method is the generative tree of the data, and therefore this classifier achieves the Bayes' error rate for this example. We simulate several Markov chains are simulated to guard against trapping in local optima. This is not a new strategy for MCMC with many local

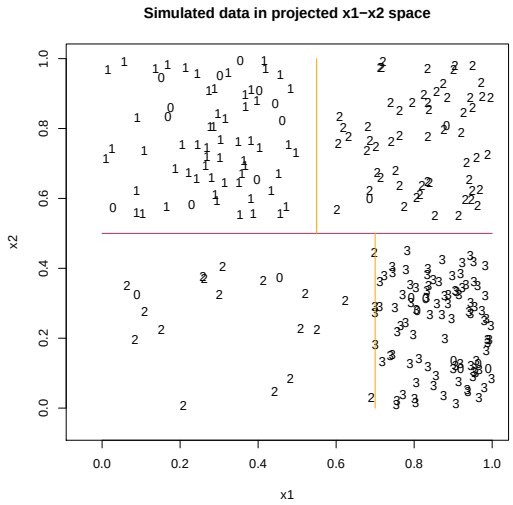


Figure 10: A plot of the true DGP

optima and was proposed by CGM [21] for decision tree MCMC samplers to aid exploration of the decision tree space and to prevent getting stuck in one of the many local optima. We tried a few chain lengths (500, 1000, and 2000) of each chain and reran 10 chains on each data set. The results were the same, regardless of the chain length.

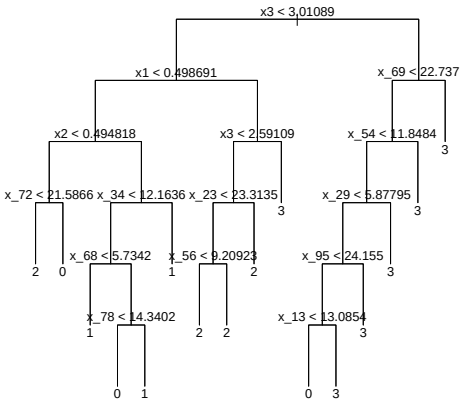


Figure 11: The tree found by a greedy optimization

To compare the three methods, we calculate misclassification probabilities for each of the three methods. To calculate misclassification probabilities we took a random sample of 250 observations as hold out test data. The misclassification probabilities are calculated by dropping all data points

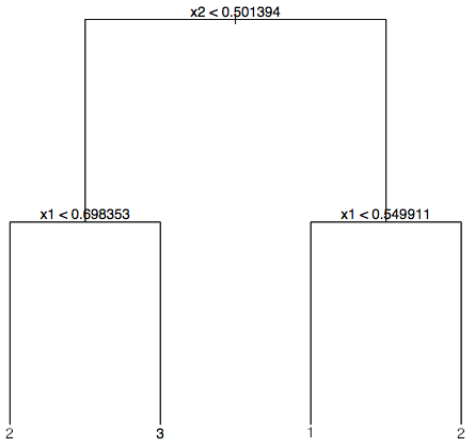


Figure 12: Best tree using the weighted method

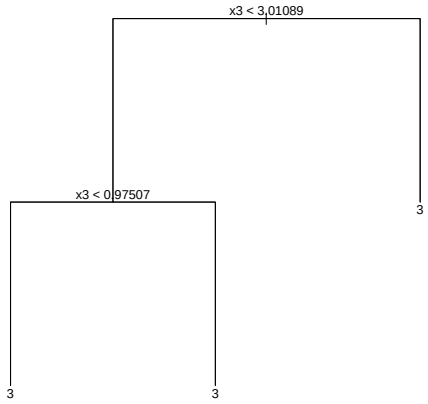


Figure 13: The best tree found by the CGM method

Method	Misclass Prob.
Greedy	11.6%
CGM	62.4%
Weighted	10.8%

Table 2: Misclassification probabilities of the three tree fitting methods for $d = 100$.

from the test data set through the resulting trees. We fit the trees using only the 250 observations from the training data. The tables of misclassification probabilities are displayed in Tables 2 and 3.

We see that the pruning rule fails to collect the correct covariates as important in the greedily grown tree. The problem is that the deterministic procedure is overwhelmed by candidate splits and thus overfits, in fact the misclassification probability *within* the training data is 4.8%, better than the Bayes rate! However, this method performed poorly on hold out data compared to the weighted method, as shown in Tables 2 and 3. The greedy method eventually finds some suboptimal splits. Of course, in this example we know the truth, and these extra splits are far from the truth. In addition, we see that the CGM approach fails on these trees. Much like the greedy method, the CGM approach is also overwhelmed with possible splits. Each new covariate adds exponentially more possible trees. This plethora of trees often allows one to find many locally optimal trees, but

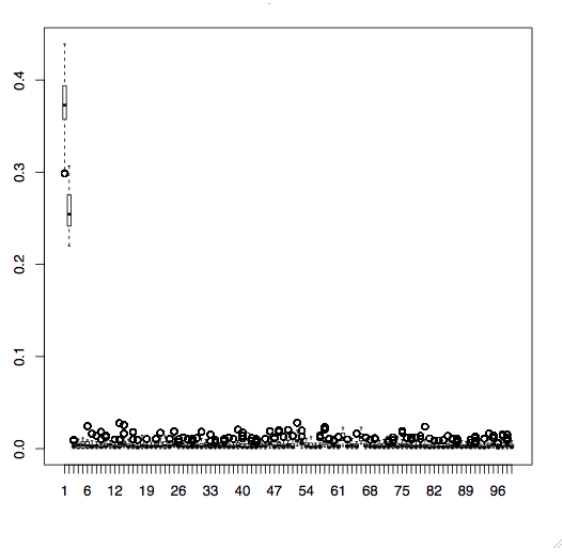


Figure 14: Covariate weights on the 100 dimensional example. Note the first two covariates are selected with high probability and the rest with miniscule probability.

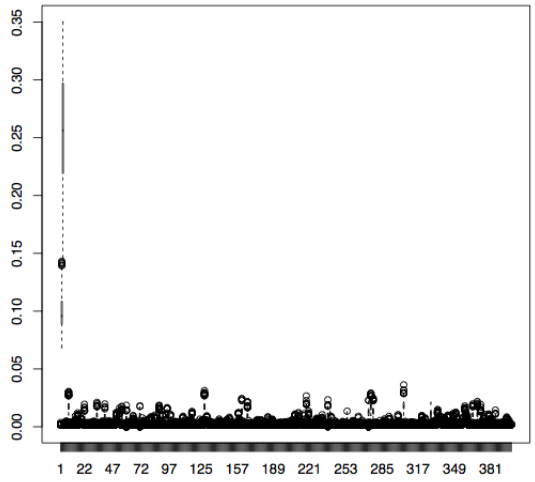


Figure 15: Covariate weights on the 400 dimension simulated example. Note the first two covariates are selected with high probability and the rest with negligible probability.

at the cost of exponentially more search time. A benefit of our method is the improved search by effectively ignoring certain covariates and preventing some trapping of the chain in local optima. By allowing non-uniform probabilities of selecting covariates, the algorithm spends more time searching through covariates that have good splits, while not spending time in other covariates that have mostly useless splits. This behavior leads to a more efficient search of the tree space and, consequently, better predictive trees than those found by the CGM search approach.A s indicated by the results in Table 3, the trees built using our weighted approach are better or competitive in prediction but simpler for inference compared with those found by a greedy search.

3.4.1 Choosing Covariates

A natural question to ask is: How should one choose covariates? This question is important because there are often covariates which have estimated probabilities which might be considered large enough to be important but also could be small enough to be considered not important. We

Method	Misclass Prob.
Greedy	24.8%
CGM	57.2%
Weighted	10.0%

Table 3: Misclassification Probabilities of the three tree fitting methods for $d = 400$.

approach this question from the context of the learning achieved during the sampling. We call the prior moment estimator defined here by Equation 51 as

$$\Pr(\text{covariate } j \text{ included}) = \frac{\alpha_j}{\sum_{j=1}^{d} \alpha_j}. \tag{51}$$

Nonetheless, the prior moment estimator does not take into account the learning performed by the MCMC algorithm. We now discuss a method that accounts for the learning occurring as the MCMC algorithm progresses.

Note that Equation 51 is simply the maximum likelihood estimator of the *a priori* probabilities of each covariate being selected to split on in the decision tree. To choose which covariates to use (in some subsequent model building), one might select those covariates where the *a posteriori* probabilities are larger than the *a priori* probabilities. This then indicates that information on the covariates was learned during the course of the MCMC algorithm. Formally, select the set of covariates for inclusion according to the rule

$$J = \left\{ \text{all } j : \frac{\alpha'_j}{\sum_{j=1}^{d} \alpha'_j} > \frac{\alpha_j}{\sum_{j=1}^{d} \alpha_j} \right\}, \tag{52}$$

where the $\alpha'_j = \alpha_j + \widetilde{\alpha} s_j$ denote posterior concentration parameters for each covariate.

Other methods for covariate selection are possible but we do not discuss them here. We present a small simulation study to determine the efficacy of the proposed method. We simulate data as described in the earlier portion of this section, where we know the specified covariates that should be selected, and those that should not. We ran the simulation 100 times and present the average percentage correctly selected by using the rule in Equation 52 for our weighted method. We also compared our weighted method results to simple frequency estimates using bootstrap samples and using the CGM method for fitting decision trees. The results are presented in Table 4. We fit a collection of trees using a bootstrap resample of 100 and using 1000 trees and calculate the number of times the correct covariates are selected. These entries are the subscripted '100' and '1000' entries in Table 4.

	Weights	Boot_{100}	Boot_{1000}	Greedy	CGM
$\Pr$(covariate 1 selected)	0.32140	0.12405	0.12664	0.12178	0.13333
$\Pr$(covariate 2 selected)	0.33806	0.10795	0.10835	0.12376	0.06667
$\Pr$(covariates 1 and 2 selected)	0.15429	0.08712	0.08365	0.08614	0.06667
$\Pr$(any other covariates selected)	0.34054	0.76799	0.76502	0.75446	0.80000

Table 4: The empirical performance of the tree fitting methods on the variable selection problem using 100 simulations.

From Table 4 it is clear that the weighted method outperforms the four other methods in correctly selecting the important covariates. Also, our weighted method selects the correct covariates jointly better than the other methods. Moreover, our weighted method selects incorrect covariates less often. whereas the other methods have a higher frequency of incorrectly selecting covariates. It is important to note that theoretically we do not want the entries in the last line to approach zero. If we did this then the MCMC would get immediately stuck in a local maximum of the likelihood space and would not find many good local maxima. Our weighted method is able to more evenly balance the search for new local maxima with the search for the correct splits and tree topology within the basin of attraction of the current local maximum. This leads to improved searching of the tree space and leads to better trees in terms of inference, which are competitive in terms of prediction.

4 A Case Study

In this section we compare our method against the greedy approach and the CGM approach using the publicly available internet ads dataset from the UCI data repository [38]. The internet ads data set contains 1558 covariates, some numeric and some categorical. The data set was first used in a paper by Kushmerick [64] and has since been used in several statistical classification studies. The UCI machine learning repository contains a more complete listing of papers using this dataset.

As noted by Kushmerick [64], there are several reasons for wanting to remove advertisements from webpages. Some reasons are: Images tend to dominate a pages total download time, users dislike paying for services indirectly through advertisers, preferring direct payment for services rendered, and malicious software can be unintentionally placed on a user's machine through images

masked as advertisements.

We used a random subset of internet advertisements data to fit a greedy tree, a CGM tree and a tree fit using our weighted method. We first removed all observations that contain missing values. We then took a 50% random sample of training and test data. The resulting trees are shown in Figures 16-18. The misclassification probabilities are calculated by dropping all data points from the test data set through the trees built using the training data. The resulting trees are fit using only the training data.

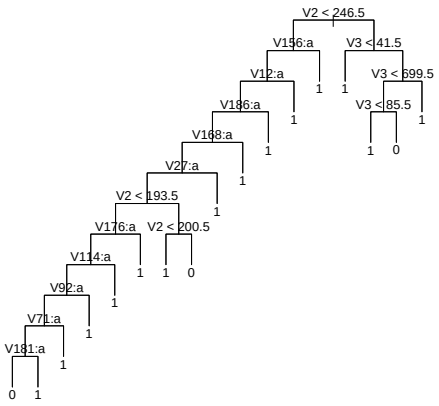


Figure 16: The greedy tree.

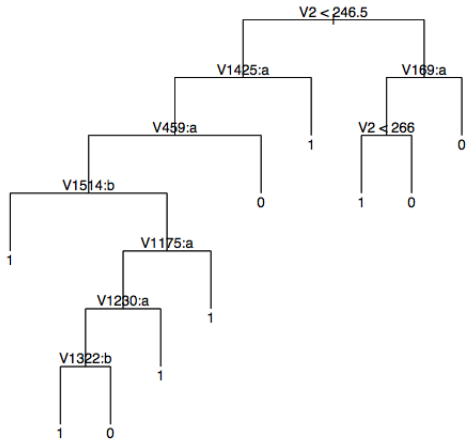


Figure 17: The tree found by the CGM algorithm on the internet ads dataset.

Method	Misclass Prob.
Greedy	5.578%
CGM	7.2%
Weighted	5.598%

Table 5: Misclassification probabilities of the three tree fitting methods on the internet ads dataset.

The best tree is fit using our weighted method. The tree is best both in terms of simplicity, and in terms of misclassification probabilities, see Table 5. The CGM approach is not as good at predicting as those trees found by either the greedy or our weighted approach. Amongst the weighted and greedy approaches, the tree found by the greedy algorithm is far more complex than

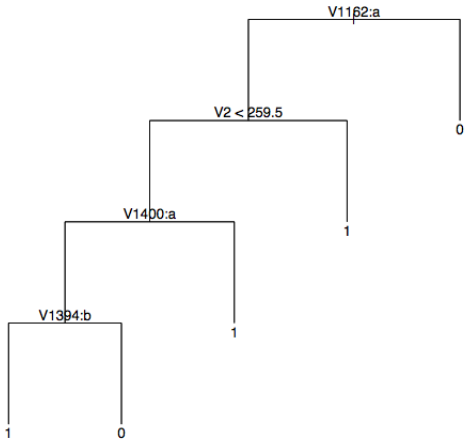


Figure 18: Best tree using the weighted method on the internet ads dataset.

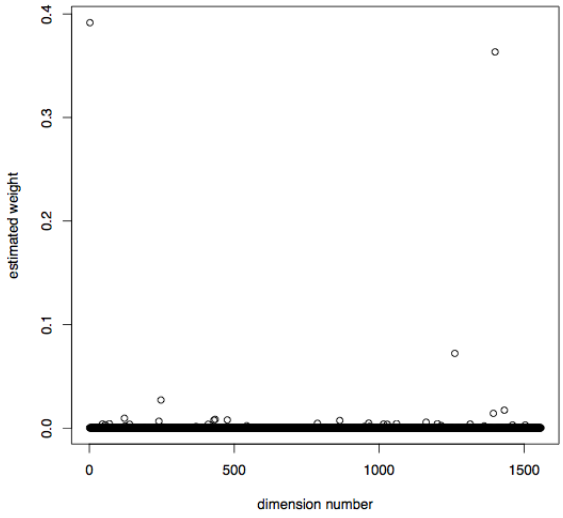


Figure 19: Estimated covariate weights using the internet ads dataset.

the tree found by our weighted method. The weighted method is essentially as good in terms of prediction error as the greedy method and is simpler to understand and interpret. Moreover, the greedy method and the CGM method do not provide output explicitly indicating which covariates are useful in the model.

Figure 19 presents the estimated probability of selection from our model. In this case there are three covariates that are clearly important in the model. The covariates selected according to the rule from Equation 52 are listed in decreasing order in Table 6. The covariates used in the tree in Figure 18 correspond to a subset of these covariates and are starred in the table. The starred covariates are: the width of the image, the url tag "tkaine+bars", the url tag "www.irish-times.com", and the url tag "click". The tag "click" is not particularly surprising because many ads are in the form of images that pop up a secondary webpage if you click on the image. The "Irish Times" is a newspaper and a perusal of their current webpage indicates that they have many images that accompany news articles and we can reasonable assume the website was similar when the dataset was created. The width field is also not surprising, wide windows facilitate better views of images. It is not very clear what the field "tkaine+bars" means. Perhaps some of the images come from a collection of webpages during Tim Kaine's race for mayor of Richmond, Virginia, in

Cov Num	2*	1400*	1261	247	1432	1394*	121	434	476	430	864	240
Rank	607.78*	563.94*	111.91	42.00	26.78	21.99*	14.66	12.97	12.12	11.84	11.28	10.15
Cov Num	1162*	964	787	1060	69	1201	45	1015	1314	139	410	1028
Rank	8.74*	7.33	7.05	6.77	6.48	6.48	5.92	5.92	5.92	5.64	5.64	5.64
Cov Num	55	1460	1504	1214	543	123	1362	967	368	949	253	813
Rank	4.79	4.51	4.51	3.66	3.38	3.10	3.10	2.54	2.26	2.26	1.41	1.13

Table 6: The set J of possibly included covariates from Equation 52. Covariates are listed in decreasing order of the value used to include them in J. Stars indicate covariates used in Figure 18. The rank is a numeric value that follows from Equation 52.

1998. Admittedly, more information on this field would be desirable.

4.1 Discussion of Results

In this chapter we studied the application of MCMC data partitioning models to large dimensional data. We showed that, in large dimensional data sets, the methods used on low dimensional examples will not work effectively. We provided a simple modification that greatly improves the accuracy and utility of the partitioning scheme in large dimensional datasets.

When running MCMC algorithms on large dimensional data,we spend more time searching the covariates that are important (high probability) and ignoring the covariates that are not useful (the low probability ones). Moreover, the values of the probabilities for each covariate are on a probability scale providing intuitive interpretation of the values sampled from the posterior distribution of weights. Sparsifying our tree search procedure provides us with several benefits. Firstly, we are able to separate the covariates into groups of high and low probability. Secondly, this gives us simplified interpretation of the tree models by searching for simple trees and eases inference for the analyst. Thirdly, we explore the state space more efficiently than the nonsparse greedy approach or the CGM search approach.

Finally, we note that our approach to sparsity is different from the approach in Chipman, George and McCulloch [22]. In their work the goal was to model using short trees, so sparsity was achieved as a byproduct of constraining the complexity of the decision tree. Furthermore, a simple tree was their primary goal and no measure of usefulness on each covariate was desired. The inferential difficulties of the pruning rule noted in the introduction still apply to the Chipman

et al. [22] method, whereas our dimension weighting does not have this difficulty.

5 Additive Logistic Variable Selection: The ALoVaS method

5.1 Normal Distributions Transformed to the Unit Simplex: The ALoVaS method.

This chapter outlines the additive logistic transformation, which transforms a d dimensional multivariate normal distribution onto the unit simplex in $d+1$ dimensions. Note that the unit simplex in $d+1$ dimensions actually lies in a subspace of d dimensions because of the constraint that the sum of the probabilities equals one.

The goal of this chapter is to find a transform that moves the space $\mathbb{R}^d$ to the simplex $\mathbb{S}^d$. The simplex $\mathbb{S}^d$ is a space defined by the constraints $\{x_i : 0 < x_i < 1, \sum_{j=1}^{d} x_j < 1\}$, and the extra term $x_{d+1} = 1 - \sum_{j=1}^{d} x_j$ ensures the total sums to 1.

Define the notation $\underline{y}$, for the normal random variables that reside in the $\mathbb{R}^d$ dimensional space. Define the notation $\underline{x}$ for the vector that resides on $\mathbb{S}^d$, the simplex in d dimensions. We use an underline to indicate that the stated quantity is a column vector and capital greek letters (and I) will denote matrices (the identity matrix)

$$x_i = \frac{e^{y_i}}{1 + \sum_{j=1}^{d} e^{y_j}}. \tag{53}$$

The Jacobian of the transform is defined as

$$J(\underline{y}|\underline{x}) = \left(\prod_{j=1}^{d+1} x_j \right)^{-1}. \tag{54}$$

It is important to note that $\underline{y} \in \mathbb{R}^d$, whereas $\underline{x} \in \mathbb{S}^d$. The d dimensional normal has the usual parameters and density

$$f_{\underline{y}}(\underline{y}|\Sigma, \mu) = (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left(-1/2(\underline{y} - \underline{\mu})^T \Sigma^{-1} (\underline{y} - \underline{\mu})\right). \tag{55}$$

Upon applying the transformation defined by Equation 53, we arrive at the additive logistic normal (ALN) distribution, with density

$$f_{\underline{x}}(\underline{x}|\Sigma,\mu) = \frac{(2\pi)^{-d/2}}{\sqrt{|\Sigma|}\prod_{j=1}^{d+1} x_j} \exp\left(-1/2(\log(\underline{x}_{(d+1)}/x_{d+1}) - \underline{\mu})^T\Sigma^{-1}(\log(\underline{x}_{(d+1)}/x_{d+1}) - \underline{\mu})\right). \tag{56}$$

The vector notation $\underline{x}_{(d+1)}$ denotes the vector in d dimensions that has the $d+1$ entry removed from the vector $\underline{x}$. It is important to note that this density function is defined on the space $\mathbb{S}^d$ and *not* on the space $\mathbb{R}^d$.

A useful property of this transform is that we can handle probabilities defined on the d dimensional simplex while working with a normal distribution. This is a common and comfortable probability distribution for most statisticians and applied scientists. We now wish to understand how the specification of the mean vector $\underline{\mu}$ and the covariance matrix Σ impact the structure of the ALN density. From simulation we can formulate the following conclusions:

- With $\Sigma = I$, increasing the mean vector in the positive direction in any one of the d components individually corresponds to shifting density towards the corner of the simplex associated with that covariate.
- With $\Sigma = I$, increasing the mean vector in the negative in *all* d components corresponds to shifting density towards the $d+1$ corner of the simplex.
- Keeping $\underline{\mu} = \underline{0}$, adjusting any of the variances corresponds to shifting towards a projected space of $\mathbb{S}^d$.
- With $\underline{\mu} = \underline{0}$, making one variance small corresponds to the shifting density towards the median of the simplex associated with the remaining d dimensions.
- Making the Σ matrix approximately singular and moving $\underline{\mu}$ in the negative direction for all components places most of the probability density along the median of simplex associated with first d dimensions.
- If $\Sigma = \text{Diag}(\sigma_j^2)$, as the σ_j^2 entries become smaller, the probabilities approach the CGM specification.

Using the transformation defined in Equation 53 and the fact that it is relatively simple to simulate from the vector normal distribution, the ALN density can be simulated.

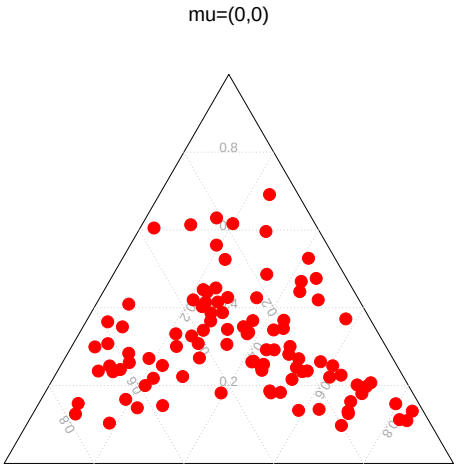


Figure 20: In this figure, $\underline{\mu}$ has all zero entries, with $\Sigma = I$, corresponding to the equiprobable case. Each probability is approximately $1/(d+1)$.

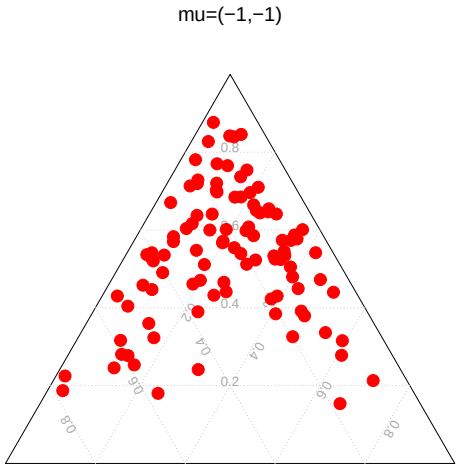


Figure 21: In this figure, $\underline{\mu} = (-1,-1)^T$, with $\Sigma = I$, corresponds to moving towards a sparser set of covariates.

The simulations indicate that we can scale everything at approximately an $\mathcal{O}(d)$ rate because of the diagonality of the variance covariance matrix, no matrix inversions are required. The step that remains is to link this portion of the model with the (currently) observed data in the tree, so that better trees are favored over trees that are worse, when each is observed in the Markov chain. We will first work in the space $\mathbb{R}^d$ and then translate to probabilities by using Equation 53.

If we are to focus on the means and a collection of variances in a multivariate normal i.i.d. model, then we must fully specify the likelihood and the prior to form our posterior.

The likelihood of the tree is defined as follows: the tree's selected covariates are counted and summed across all observed splits, leading to a likelihood taking on discrete values for the observed data. Let us define a multiplier that can take on an arbitrary positive or negative value in a compact region. We then multiply the counts of splits on each covariate by this quantity, effectively creating a mean which can take on arbitrary values in $\mathbb{R}$.

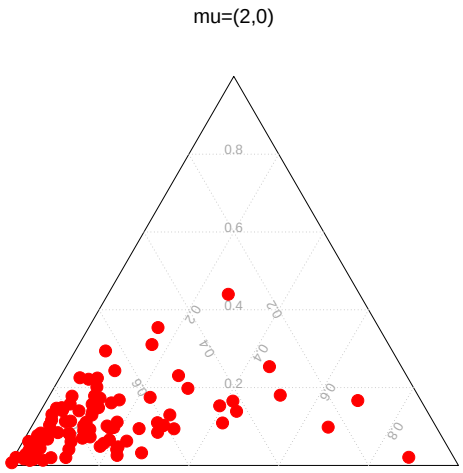


Figure 22: $\underline{\mu} = (2, 0)^T$, with $\Sigma = I$, moves the density towards one corner of the simplex.

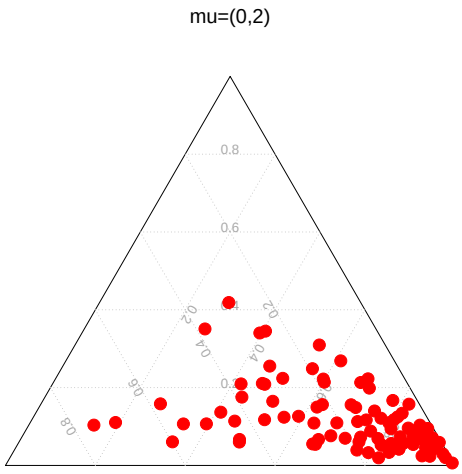


Figure 23: $\underline{\mu} = (0, 2)^T$, with $\Sigma = I$, moves the density towards the other corner of the simplex.

5.2 A Simple Sampler Approach

This subsection derives the full conditional densities for each necessary update in the Gibbs loop to sample posterior weights on each dimension in the CGM decision tree sampler. Throughout this subsection we will use the notation $\odot$ to denote a Hadamard product of two matrices or vectors.

5.2.1 The General Strategy

There are many problems with using a Dirichlet prior and a Multinomial conjugate likelihood. The two most glaring problems are the implicit prior assumption of same scales on each covariate and the fact that all covariances or, equivalently, correlations must be negative. If the generative model of the data is a linear model with an interaction term and a decision tree model is fit to the data, then several splits will occur on the two interacting covariates. These splits will occur alternately until the curvature is sufficiently approximated [59]. This situation indicates a positive correlation between the two covariates. Higher order interactions will result in similar positive correlations between collections of covariates. Therefore we conclude the Dirichlet density as a posterior for the probability of selecting a covariate is an inferior model. Moreover, the initial

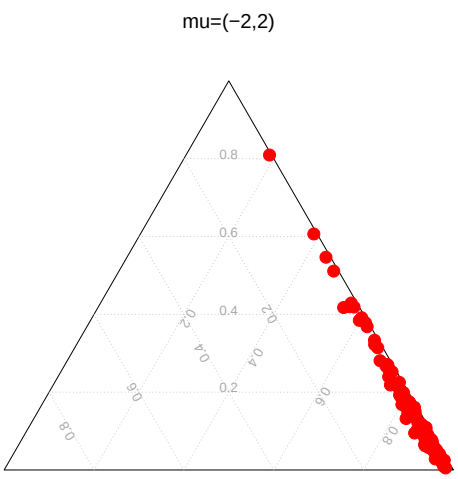


Figure 24: $\underline{\mu} = (-2, 2)^T$, with $\Sigma = I$, corresponds to most probability mass along a corner of the simplex and is a sparse representation.

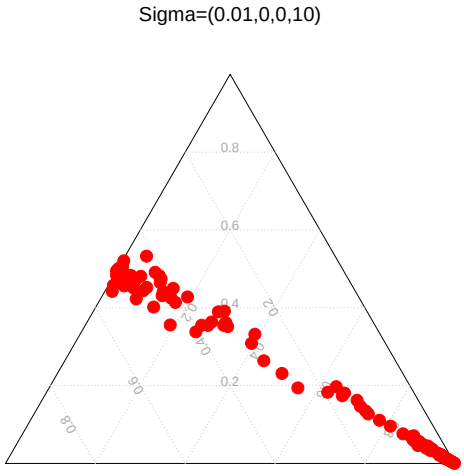


Figure 25: $\underline{\mu} = \underline{0}$, with $\Sigma = \text{diag}(0.01, 100)$, corresponds to most density lying on a one dimensional subspace (the second covariate in the $\mathbb{R}$ space).

motivation for modeling data with a decision tree was to handle survey data that contained many complex interactions that would be too computationally expensive to evaluate using linear model methods [74].

The likelihood will be denoted by

$$MVN(\underline{c} \odot \underline{s}|\underline{\mu}, \Sigma = \text{Diag}(\sigma_j^2)). \tag{57}$$

The prior is also a normal

$$\pi(\underline{\mu}|\underline{\mu}_p, \Sigma) = MVN(\underline{\mu}|\underline{\mu}_p, \Sigma). \tag{58}$$

The priors on the variances are $i.i.d.\sim$ Inv-Gamma$(\sigma_j^2|a_j, b_j)$. Finally the priors on the c_js are $i.i.d.$ scaled beta's with densities

$$S\beta(c_j|\alpha_j, \beta_j, -a, a) \equiv \pi(c_j|\alpha_j, \beta_j, -a, a) = \frac{(c_j + a)^{\alpha_j - 1}(a - c_j)^{\beta_j - 1}\Gamma(\alpha_j + \beta_j)}{(2a)^{\alpha_j + \beta_j - 1}\Gamma(\alpha_j)\Gamma(\beta_j)}. \tag{59}$$

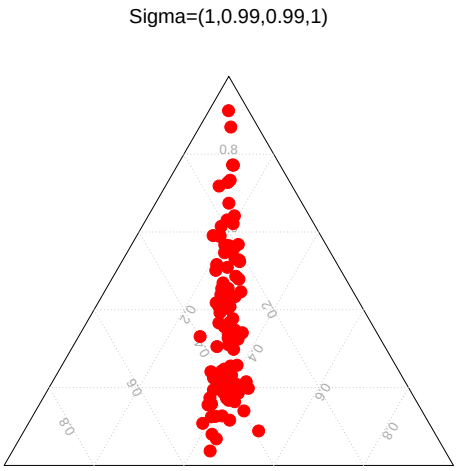


Figure 26: Here Σ is approximately singular and most of the probability mass in concentrated along the $d+1$th dimension in the $\mathbb{R}^{d+1}$ space.

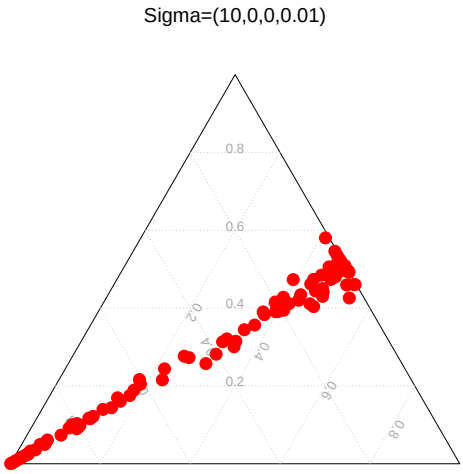


Figure 27: Similar to the case in Figure 25 but with the variances reversed.

A special case of these scaled beta densities is when $\alpha_j = \beta_j = 1$, which yields uniform r.v.'s on the region $[-a, a]$. For each $j = 1, \ldots, d$, we first calculate $s_j = \epsilon + \sum_{\forall \eta \in \mathcal{T}} \mathbb{1}[\text{split on covariate j at node } \eta]$, for some fixed $\epsilon > 0$. Through basic Bayesian calculations we find that the full conditionals are

$$\pi(c_j|\mu_j, \sigma_j^2, a) \sim N_{[-a,a]}(c_j s_j|\mu_j, \sigma_j^2)(\text{a normal truncated to the region } (-a, a)), \tag{60}$$

$$\pi(\sigma_j^2|\mu_j, c_j, a) \sim \text{Inv-Gamma}(\sigma_j^2|2a_j + 2, \frac{(c_j s_j - \mu_j)^2 + (\mu_j - \mu_j^p)^2 + b_j}{2}), \text{ and} \tag{61}$$

$$\pi(\mu_j|\mu_j^p, c_j, \sigma_j^2) \sim N[c_j s_j + \mu_j^p, \sigma_j^2]. \tag{62}$$

Ideally we would like everything to be a Gibbs step. This can be accomplished numerically using numerical approximations to the cumulative density of the standard normal distribution, here denoted Φ, and Φ^{-1}, the quantile function of the standard normal cumulative density. However, we can accomplish this directly using the technique of parameter expansion set forth in Damien and Walker [26], which we review here for completeness.

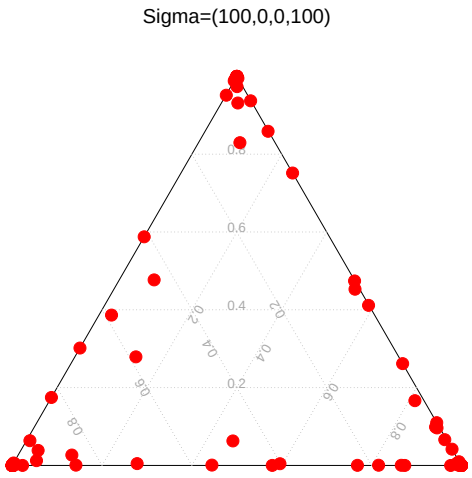


Figure 28: $\underline{\mu} = \underline{0}$, with $\Sigma = \text{diag}(100, 100)$, corresponds to encouraging sparse representations *a priori*.

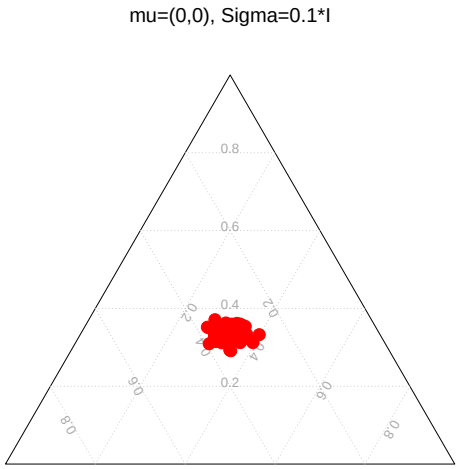


Figure 29: Here $\Sigma = 0.1I$, and $\underline{\mu} = \underline{0}$, corresponds to roughly the CGM specification.

We begin with the joint density of two random variables

$$f(x, y) \propto \mathbb{1}[0, \exp(-x^2/2)](y). \tag{63}$$

The marginal of x is a standard normal density. Through elementary probability calculations we find

$$f(y|X = x) \propto Unif(0, \exp(-x^2/2)). \tag{64}$$

and

$$f(x|Y = y) \propto Unif(-\sqrt{-2log(y)}, \sqrt{-2log(y)}) \tag{65}$$

The region upon which X is defined arises from the solution of the inequalities $\{0 \leq y \leq \exp(-x^2/2)\}$ for X, which is a quadratic equation in X. Similarly, if we have a truncated distribution truncated to the region $[a, b]$, we write the joint density

$$f(x, y) = Unif((0 \leq y \leq \exp(-x^2/2)) \times (a, b)). \tag{66}$$

Solving the resulting system of inequalities leads to the set

$$\{-\sqrt{-2log(y)}, \sqrt{-2log(y)}\} \cap \{a, b\}, \tag{67}$$

which results in the region for x

$$\{\max(a, -\sqrt{-2log(y)}), \min(b, \sqrt{-2log(y)})\}. \tag{68}$$

Simulating from a truncated normal is equivalent to simulating from two uniforms on the appropriate regions and evaluating a natural logarithm and a square root.

A special case of these scaled beta densities is when $\alpha_j = \beta_j = 1$, which yields uniform random variables on the region $[-a, a]$. For each $j = 1, \ldots, d$, we first calculate $s_j = \epsilon + \sum_{\forall \eta \in \mathcal{T}} \mathbb{1}[\text{split on covariate j at node } \eta]$, for some fixed $\epsilon > 0$. Through basic Bayesian calculations we find the full conditionals in closed form

$$\pi(u|c_j s_j, \mu_j, \sigma_j^2) = \text{Unif}(0, \exp\,(\frac{-(c_j s_j - \mu_j)^2}{2\sigma_j^2})), \tag{69}$$

$$\pi(c_j|u) = \text{Unif}(\max(-a, -\sqrt{-2log(u)}), \min(a, \sqrt{-2log(u)})), \tag{70}$$

$$\pi(\sigma_j^2|\mu_j, c_j, a) \sim \text{Inv-Gamma}(\sigma_j^2|2a_j + 2, \frac{(c_j s_j - \mu_j)^2 + (\mu_j - \mu_j^p)^2 + b_j}{2}), \text{ and} \tag{71}$$

$$\pi(\mu_j|\mu_j^p, c_j, \sigma_j^2) \sim N[c_j s_j + \mu_j^p, \sigma_j^2]. \tag{72}$$

The Gibbs sampling step, nested within the MH sampler, proceeds by sampling from Equations 69-72 At each iteration, once samples are drawn in sequence from the distributions given in Equations 69- 72, we take the posterior samples of μ_j and transform these onto the $[0, 1]$ scale, by using the ALN transform given in Equation 53.

We show results from a preliminary coding of the stated algorithm. We use two specifications for the prior means $\underline{\mu}^p$. One specification uses $\underline{\mu}^p = \underline{0}$ (Figure 30) and another uses $\underline{\mu}^p = (-2, -2, 2, \ldots, 2)$ (Figure 31). The difference in the results of sampled weights indicates that, if we can move from a negative to a positive value for the prior mean, we can greatly influence the selection of covariates. The graphic of posterior weights shown in Figure 31 is the correct set of weights for the data.

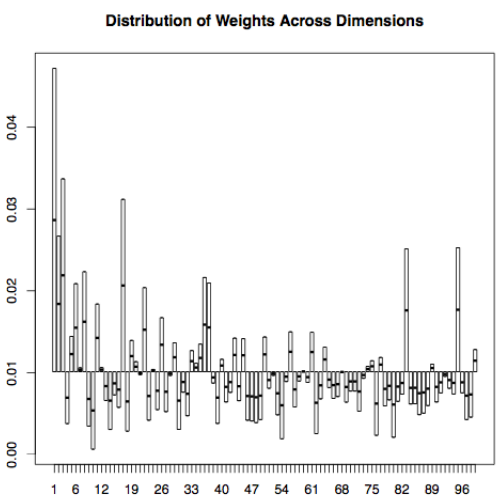


Figure 30: A zero mean prior. Note that the two covariates that should have large probabilities are covariates 1 and 2.

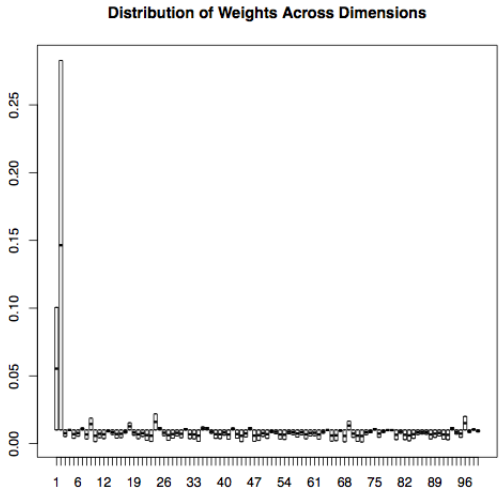


Figure 31: An informative prior. The first two prior means are 2 and the remaining prior means are set at -2.

5.3 A Stochastic Search Variable Selection Approach

Using the results of George and McCulloch we can derive Gibbs updates for each of the parameters and have as a special case the approach of CGM if all the dimension indicators are selected to be the point masses at zero. In this case the ALN transform puts probability $1/(d+1)$ on each dimension.

We investigate several priors for use in variable selection with decision trees. They are:

- A stochastic search approach (method of George and McCulloch and Cui and George) [25, 45].
- A lasso prior on the means of the MVN, probably using parameter expansion to get Gibbs updates [78].
- A multivariate half-Cauchy prior using parameter expansion (Huang and Wand method) [57, 80, 18, 17].
- A 'local' prior approach (Valen Johnson and David Rossell JRSS B 2010 approach) [60].

We will discuss the preliminary details of each method in sequence in this chapter.

Variable selection with SSVS facilitates a Bayesian approach to variable selection in decision trees. Prior methods have examined bootstrapping approaches and used some complicated math to

allow the statistician to peer inside the black box method known as randomForest [59, 58], often with little insight or understanding. The SSVS prior allows us to "test" whether the constant prior probability of selecting a covariate for all dimensions is appropriate for the given dataset. This means we can test, for any dataset, whether the CGM model of covariate selection is appropriate.

We will implement and compare our method against other methods such as observed frequency, a naïve method, as well as the maximal subtree approach of Ishwaran et al. [59, 58]. The SSVS approach involves a prior on the normal means of the form

$$\pi(\mu_i|\tau_i, p_i, c_i) \propto p_i N(\mu_i, \tau_i^2) + (1 - p_i)N(\mu_i, c_i^2\tau_i^2). \tag{73}$$

Here the $c_i > 1$ and $1 > \tau_i > 0$ is small. Usually $p_i \equiv 1/2$ but it is also possible to put a prior on these hyper-parameters. The advantage of a mixture prior in this form is that Gibbs samples are readily available for each of the desired quantities facilitating fast sampling.

5.4 A Lasso Prior

In this section we propose to use parameter expansion to facilitate Gibbs sampling within the sampling framework for the posterior means for each covariate in the normal space. We use the ALN transform to then convert to probabilities. The key idea we use for parameter expansion is the scale mixture of normals representation of a Laplace prior. This representation is stated here for reference.

If $V \sim$ Exponential(1) and $Z \sim N(0, 1)$ independent of V, then $X = \mu + b\sqrt{2V}Z \sim \text{Laplace}(\mu, b)$.

Equivalently we can write this as the hierarchy

$$X|V \sim N[\mu, \sigma^2 2V], \tag{74}$$

$$V \sim \text{Exponential}(1), \text{ and} \tag{75}$$

$$X \sim \text{Laplace}(\mu, \sigma) \tag{76}$$

What we want to examine here is whether shrinkage along an L_1 penalty will achieve similar results to the SSVS approach and give us meaningful results. The belief here is the same as in

the SSVS approach, shrinkage will shrink means toward zero in the multivariate normal. These sampled values from the multivariate normal will then be transformed into something on the $[0, 1]$ (probability) scale using the ALN transform. Values of zero, or near zero, on the normal space transform into values of approximately $1/(d+1)$. The probabilities with values around $1/(d+1)$ correspond to covariates we are indifferent about. A confounding aspect of interpreting the probabilities is that they must all sum to 1. Therefore if some values are excessively small this may push all the other probabilities to be larger. If something like this happens it is then difficult to determine which covariates are non-informative. Further simulations regarding selecting covariates using the rule of Equation 52 will be conducted using the ALN transform and will be done on several simulated and real data examples.

The following result will be useful for parameter expansion. If

$X|a \sim \text{Inv} - \text{Gamma}(\nu/2, \nu/a)$ and $a \sim \text{Inv} - \text{Gamma}(1/2, A^2)$, then $\sqrt{X} \sim \text{Half} - \text{Cauchy}$.

$$
\begin{aligned}
f(x) &= \int_0^\infty \underbrace{\frac{\nu^{\nu/2}}{a^{\nu/2}\Gamma(\nu/2)x^{(\nu/2)+1}} \exp\left(-\frac{\nu}{ax}\right)}_{=f(x|a)} \underbrace{\frac{\exp\left(-\frac{1}{aA^2}\right)}{A\sqrt{\pi}a^{3/2}}}_{=f(a)} da \\
&= \frac{\nu^{\nu/2}}{A\sqrt{\pi}\Gamma(\nu/2)x^{(\nu+2)/2}} \underbrace{\int_0^\infty a^{-\nu/2-1/2-1} \exp\left(-(1/a)(\nu/x + 1/A^2)\right) da}_{=\text{Inv-Gamma kernel}} \\
&= \frac{\nu^{\nu/2}\Gamma((\nu+1)/2)}{A\sqrt{\pi}\Gamma(\nu/2)x^{(\nu+2)/2}(\nu/x + 1/A^2)^{(\nu+1)/2}} \\
&\propto \frac{1}{\sqrt{x}x^{(\nu+1)/2}(\nu/x + 1/A^2)^{(\nu+1)/2}} \\
&= \frac{1}{\sqrt{x}(\nu + x/A^2)^{(\nu+1)/2}} \\
&= \frac{1}{\sqrt{x}(\nu + x/A^2)^{(\nu+1)/2}}.
\end{aligned}
$$

Making the change of variable $x = y^2$, which implies $dx/(2\sqrt{x}) = dy$, shows us that

$$
f(y) \propto (1 + (y/A)^2/\nu)^{-(\nu+1)/2}, \tag{77}
$$

for $y > 0$ which is the definition of a half-Cauchy density.

Huang and Wand [57] indicate that using an inverse-Wishart prior on the variances, with each standard deviation having a half-Cauchy prior, gives a scaled beta marginal for each correlation. At this point it is worth recalling that the t and half-t distributions have as special cases, the Cauchy

and half-Cauchy distributions when the degrees of freedom parameter (ν) in the t and half-t is set to $\nu = 1$.

6 A Case study using the ALoVaS Model

6.1 The Need For Sparsity

6.2 The internet ads dataset revisited

6.3 Simulation Results

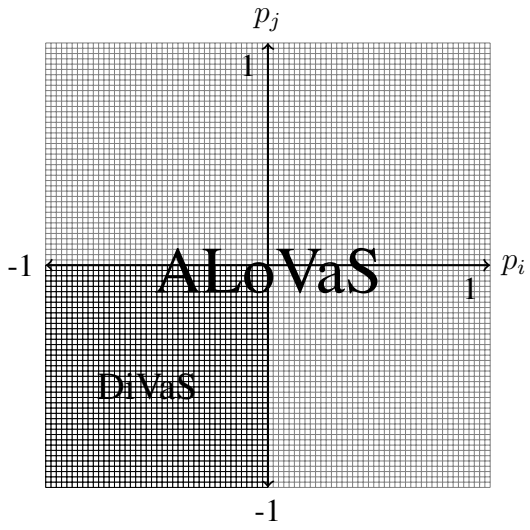
6.4 Conclusions

7 Synthesis: comparing the DiVaS and ALoVaS methods

In this section we compare the DiVaS and ALoVaS models. While both methods seem relatively similar, both methods performing variable selection on Bayesian decision trees, there are fundamental practical and theoretical differences between the two methods.

7.1 Theoretical differences and a simulation study

One of the most apparent theoretical differences between the DiVaS and ALoVaS models is the way the two models handle the correlation between covariate selection probabilities. One of the properties of the Dirichlet distribution is that $Corr(p_i, p_j) < 0$ for $i \neq j$. Thus if the prior puts zero probability on non-negative covariate selection probability correlations then the posterior will also place zero probability on these non-negative correlations. This section elucidates the differences between the two methods and compares the similarities.



One might be tempted to conclude that this is a element of minutæ that has no practical consequence but consider a linear model data generating process that contains an interaction between two variables, x_1 and x_2

$$y_i = \beta x_1 x_2 + \epsilon_i, \tag{78}$$

where $\epsilon_i \sim N(0, \sigma^2)$. In this case the coefficient β controls the degree of curvature in the x_1, x_2 space. The positive covariate selection probability results from using a constant function in each terminal node to approximate a curved two dimensional surface. The constant function requires alternating splits between x_1 and x_2 to approximate the curving of the x_1, x_2 space. Thus, we see that there are really two important pieces to the model specification when fitting decision trees: specifying a tree model and specifying a terminal node model. Misspecification of 1 can lead to errors in the other indicating that the two are also not independent.

A nice property of the ALoVaS method is that the if p_i, p_j be the two covariate selection probabilities, then if these p_i, p_j have a logistic normal density the covariance between the two densities is

$$Cov(log(p_j/p_k), log(p_l/p_m)) = \sigma_{jl} + \sigma_{km} - \sigma_{jm} - \sigma_{kl}, \tag{79}$$

where σ_{ij} is the covariance parameter for the normal random variable before the application of the ALT. Clearly the four $\sigma_{ij} \geq 0$ in Equation 79 so that the covariance could be positive or negative.

Figure 32: write the caption here.

Figure 33: write the caption here.

At this point it is worth noting that the idea of an automatic interaction detector was first proposed in the CHAID method [61]. The CHAID method claims to automatically detect interactions, yet popular documents [92] claim that CHAID and other decision tree methods cannot actually perform automatic interaction detection. The validity of this claim is not the point of this paragraph but it should be clear to the reader how one could visually inspect a fitted tree and determine if interaction is plausible. Moreover, with the ALoVaS method automatic interaction detection is plausible, by comparing the sampled values of $Cov(log(p_j/p_k), log(p_l/p_m))$, or equivalently $Corr(p_i, p_j)$, against the point zero one gets an indication of the degree of interaction. Additionally, we cannot determine if the interaction is positive or negative, only that an interaction is present. If there was a desire to determine the sign of the interaction then the analyst would need to provide estimates of the parameters in Equation 79 or alternately, we might estimate directly $Corr(p_i, p_j)$ with the Bayesian approach we take in the ALoVaS method, either approach is feasible.

We now present a small simulation study. We sampled data according to the model describing in Equation 78. We simulated $n = 500$ observations from this model for a training and another for a test, or hold-out data set. We simulated data under the model with $\beta \in \{5, -5\}$. The trees fitted under a greedy optimization for the two interaction scenarios are shown in Figure 32.

We ran the ALoVaS and DiVaS algorithms for $11,000$ samples discarding the first 1000 as a burn-in sample. Convergence statistics were calculated using the coda package in R with **??** statistics indicating a minimal sample size of 3764 and 3271 for the ALoVaS and DiVaS chains respectively. The trees with the largest integrated likelihood for the two chains are presented in Figure 33.

7.2 Practical differences and a simulation study

While the theoretical differences are important to academics and applied researchers who want to understand the limitations of a specific method, the questions most applied researchers tend to have

are more practical in nature. We address the practical questions in this section.

Practically speaking the ALoVaS and the DiVaS methods are different in terms of how you can think about selecting a variable to split on, or to remove, from the current decision tree. While the DiVaS method allows us to *quantify* these different scenarios it does not allow us to think about these values in a manner we are familiar with, whereas the ALoVaS method allows us a nice, linear framework to understand these differences. The primary difficulty lies in no standard, intuitive, basis to describe the simplex. Aitchison **??** discussed several bases, none of which are intuitive, whereas Euclidean geometric space is intuitive, ubiquitous, and all statisticians can be expected to have experience with the Euclidean space.

Apart from this there are other more practical differences between the DiVaS and ALoVaS methods. For example, the ALoVaS method takes much longer to simulate. This additional simulation time should be no surprise given the additional number of parameters to simulate in the ALoVaS model relative to the DiVaS model. In the ALoVaS model we typically have means, variances, and covariances to simulate one for each covariate and then global ones as well. This difference is one reason why the sampling takes longer. A second reason comes from the methods used to simulate the parameters from the DiVaS and ALoVaS methods. While the Dirichlet density is complicated because of the relationship with the gamma density there are highly optimized sampling codes for the Dirichlet density which have been used in production since at least the early 1990s and can be considered stable and highly optimized. In contrast to this, the lasso ALoVaS model requires sampling from a generalized inverse Gaussian density which is complicated to sample from. While several algorithms exist it is unclear what is the optimal algorithm for a given scenario. Nevertheless, there are well known accept-reject methods to sample the generalized inverse Gaussian density but they require more work to draw a sample (1.58 samples on average) than more straightforward sampling methods like the polar method most commonly used to sample the Dirichlet density. In addition the sampling for the generalized inverse Gaussian density typically requires evaluating special functions like the Bessel function adding to the computational time at each iteration.

A further practical difficulty that distinguishes the two methods is the correlations between the covariate selection probabilities. Recall the DiVaS method requires that the correlations between covariate selection probabilities are negative *a priori*. Whereas the ALoVaS method allows the correlation between any two covariate selection probabilities to be any value in the range $[-1, 1]$, the typical area of support for a correlation. The practical difficulty then with the DiVaS method is

that it is impossible to know whether the covariate selection probabilities are negative, positive, or zero. Thus, using the DiVaS method seems to be asking for model misspecification errors. Moreover, if the model is linear with significant interactions, as shown earlier in this chapter and we estimate the response with a decision tree, then we will have positive covariate selection probabilities. In fact any model having a curvature in two or more of the covariates implies that when we build a decision tree we will have a positive covariate selection probability regardless of whether the generative model is linear or not, curvature of the model indicates positive covariate selection probabilities.

A reasonable question the reader may ask now is: if positive correlations of the covariate selection probabilities are a result of curvature when do negative correlation occur? In this case we would need covariates to split in the decision tree but we would want to select either x_1 or x_2 but not both. As one simple example this might occur if the generative model was

$$y_i = \beta_1 x_1 + \beta_2 x_2 + \epsilon_i \tag{80}$$

for $i = 1, \ldots, n$ and the two covariates x_1 and x_2 were such that $Corr(x_1, x_2) \neq 0$. In this case we see that knowledge of one covariate indicates knowledge of the other covariate as well, if not exact then at least approximately. In this case we would need the generative model to be linear because the correlation is a bilinear operator. In this case, we could choose to split on $x_1 < c$ for some constant c or we could choose to split on $x_2 < c'$ for some value c' because the two covariates are correlated, knowledge of one implies approximate knowledge of the other and therefore we would only need to split on one and not the other covariate. Thus in this case if covariate x_1 is split on for example, the prevalence of a split on covariate x_2 would not only be unlikely, but it would also be unnecessary, provided $|Corr(x_1, x_2)|$ was large enough.

7.3 Conclusions and Recommendations

The similarities and the differences between the ALoVaS and DiVaS methods having been pointed out, we now turn to the conclusions and recommendations we have for the use of the two methods.

While the DiVaS method is a theoretically sound method, several theoretical and practical difficulties arise when trying to implement the method. In contrast the ALoVaS method has few theoretical difficulties and the practical difficulties are primarily in terms of time to simulate which will typically not be an issue for scientific problems and most problems with no dynamic, or

time varying component. Therefore, problems involving decision trees where the data occur on a timescale less than 1 day and especially less than an hour will likely not find use in the DiVaS and ALoVaS methods. Those needing these short timeframe models are likely to find some use in the time varying decision trees proposed by Gramacy, Taddy, and Polson [86]. Moreover, the approach of the DiVaS method is more of an ad-hoc method lacking a unifying idea whereas the ALoVaS method has one underlying concept, the ALT to transform from a linear scale to a probability scale.

The conclusion then seems straightforward if one is to choose between the DiVaS and the ALoVaS methods one would choose the ALoVaS method because of the theoretical soundness, the conceptual link with linear methods, and for the ease of implementation. The key drawbacks being the run-time of the algorithm and the need to choose one of the regularization techniques discussed in this text or one of the plethora of other regularization techniques available in the statistics literature. However as indicated in Section **??** the choice of which ALoVaS method to use is dependent largely on the underlying DGP, the knowledge of which would preclude the model fitting exercise. Also, our simulation studies in Section **??** largely indicate that within a given sparsity level, the choice of which type of ALoVaS method is largely irrelevant. To reiterate, the ALoVaS method, of any type is to be preferred to the DiVaS method.

8 Discussion

In this thesis we demonstrated two methods of variable selection for Bayesian decision trees and the necessity of these methods in comparison to currently used methods like the approaches of Chipman et al. [21] and Denison et al. [28]. Moreover, we showed the drawbacks of simplistic methods of variable selection like the pruning rule and bootstrapping. While we focused in this thesis on the variable selection aspects of the study of decision trees there are several avenues forward from here and this chapter discusses those paths.

The ability to provide good fitting decision trees with accurate predictions is of vital interest to practitioners in several fields. The use of simplistic models in the terminal nodes like a constant mean model or a highest class probability model will inevitably lead to model misspecification in some datasets like data with a large occurrence of zeros. The solution to this type of model misspecification problem is to use a zero-inflated model in each of the terminal nodes. One of the requirements for the Metropolis-Hastings algorithm proposed by CGM is that the parameters can be integrated over as shown in Equation 43. For the zero-inflated model this can be accomplished at

least to the point of a finite series which can be calculated exactly if the number of zero observations are not too large. If the number of zero observations is too large a numerical approximation would suffice. Future work will apply this model to several simulated and real datasets. These will include the solder data analyzed by [65], the Nematode and vine root data collected by Giese et al. []. Moreover some of these datasets could find use in applying the ALoVaS method to maintain a shallow tree and to select variables while using an appropriate terminal node model like a zero-inflated model.

A second area of further research is into the equivalence of the SSVS, lasso, horseshoe, and perhaps even the generalized double Pareto model proposed by Bhattacharya et al. [4]. By writing the negative log posteriors in proportional form we find that the horseshoe prior and the lasso prior have similar forms when expanding the final term involving the logarithm. The appropriate Taylor expansion here is

$$log\left(1+(\lambda/\sigma)^2/p\right) = (\lambda/\sigma)^2/p + (\lambda/\sigma)^4/2p^2 + \cdots + (\lambda/\sigma)^{2k}/kp + \ldots \tag{81}$$

By expanding the negative log posterior in this way we are able to see the equivalence asymptotically of the parameter expanded lasso and the horses posteriors. It is reasonable to assume a similar approximation will hold for the generalized double Pareto prior whose density is given in Bhattacharya et al. [4]. While Carvalho et al. [18] emphasize the importance of both the singularity, or at least positive point mass probability at zero, as well as the heavy tail behavior it is clear the difference in the two posteriors will be governed by a power exponential density where the variance is proportional to $1/p$. Thus in high-dimensional models the difference will be negligible and the sparsity patterns will be similar. This seems somewhat at odds with the claims of Carvalho et al. and the claims of Bhattacharya et al. and is an area in need of further clarification. Moreover, it remains to be seen if there is a similar asymptotic relationship with the stochastic search method. We posit this to be the case but the difficulty lies in specifying the appropriate limiting form of the prior to give us the SSVS model while simultaneously encapsulating the lasso and horseshoe priors.

Appendix A: Algorithms Pseudo Code

Define $\mathcal{T}^0$ as the initialized tree, n as the number of iterations of each MCMC chain, $\underline{p}^0$ as the initialized probability weights and $\underline{\alpha}'$ as the initialized pseudo-counts for the splits in each dimension, s are the observed split counts from each sampled tree. As defaults we take $\underline{p}^0 \propto \underline{1}$, and $\underline{\alpha}' \propto \underline{1}$. Also N is the tree likelihood of the new or proposed tree and O is the old tree likelihood, both are on the log scale. Finally, b denotes the number of terminal nodes in the current ($\mathcal{T}^i$) decision tree.

Algorithm 8.1: SAMPLER$(n, \underline{p}^0, \mathcal{T}^0, \alpha^0, C)$

for $i \leftarrow 1$ **to** n

$X \leftarrow Discrete\_Uniform\,(1,5)\,;$

case $(X = 1)$ $\mathcal{T}' \leftarrow Grow(\mathcal{T}^i)$;

case $(X = 2)$ $\mathcal{T}' \leftarrow Prune(\mathcal{T}^i)$;

case $(X = 3)$ $\mathcal{T}' \leftarrow Change(\mathcal{T}^i)$;

case $(X = 4)$ $\mathcal{T}' \leftarrow Swap(\mathcal{T}^i)$;

case $(X = 5)$ $\mathcal{T}' \leftarrow Rotate(\mathcal{T}^i)$;

$N \leftarrow log(\Pr(D|\mathcal{T}'))$;

$O \leftarrow log(\Pr(D|\mathcal{T}^{i-1}))$;

case $(X = 1)$ $log(R) \leftarrow N - O + log(b)$;

case $(X = 2)$ $log(R) \leftarrow N - O + log(b+1)$;

case $(X = 3)$ $log(R) \leftarrow N - O + log(p_j) \; - \; log(p_{j'})$;

case $(X = 4$ **or** $X = 5)$ $log(R) \leftarrow N \; - \; O$;

$U \leftarrow Continuous\_Uniform(0,1)$;

if $\{log(U) < log(R)\}$ $\mathcal{T}^i \leftarrow \mathcal{T}'$;

else $\mathcal{T}^i \leftarrow \mathcal{T}^{i-1}$;

$\underline{\alpha}^i \leftarrow \underline{\alpha}^{i-1} + \widetilde{\alpha}\underline{s}$;

$\underline{p}^i \leftarrow Dirichlet(\underline{\alpha}^i)$;

$\widetilde{\alpha} \leftarrow C \sum_{j=1}^{d} \alpha_j / \sum_{i,j} s_{ij}$;

The second pseudo-code listing contains the simple sampler approach from Chapter 5.1, the notation is similar to the first pseudocode.

Algorithm 8.2: SIMPLE SAMPLER$(n, \underline{p}^0, \mathcal{T}^0, \alpha^0, C)$

for $i \leftarrow 1$ **to** n

$X \leftarrow Discrete\_Uniform\,(1,5)\,;$

case $(X = 1)$ $\mathcal{T}' \leftarrow Grow(\mathcal{T}^i)$;

case $(X = 2)$ $\mathcal{T}' \leftarrow Prune(\mathcal{T}^i)$;

case $(X = 3)$ $\mathcal{T}' \leftarrow Change(\mathcal{T}^i)$;

case $(X = 4)$ $\mathcal{T}' \leftarrow Swap(\mathcal{T}^i)$;

case $(X = 5)$ $\mathcal{T}' \leftarrow Rotate(\mathcal{T}^i)$;

$N \leftarrow log(\Pr(D|\mathcal{T}'))$;

$O \leftarrow log(\Pr(D|\mathcal{T}^{i-1}))$;

case $(X = 1)$ $log(R) \leftarrow N - O + log(b)$;

case $(X = 2)$ $log(R) \leftarrow N - O + log(b + 1)$;

case $(X = 3)$ $log(R) \leftarrow N - O + log(p_j) \; - \; log(p_{j'})$;

case $(X = 4$ **or** $X = 5)$ $log(R) \leftarrow N \; - \; O$;

$U \leftarrow Continuous\_Uniform(0, 1)$;

if $\{log(U) < log(R)\}$ $\mathcal{T}^i \leftarrow \mathcal{T}'$;

else $\mathcal{T}^i \leftarrow \mathcal{T}^{i-1}$;

for $j \leftarrow 1$ **to** d

$u_j \leftarrow Unif(0, \exp\,(-c_j s_j - \mu_j)^2/2\sigma_j^2)$

$c_j \leftarrow Unif(max(-a, -\sqrt{-2log(u_j)}), min(a, \sqrt{-2log(u_j)}))$

$\sigma_j^2 \leftarrow Inv - Gamma(2a_j + 2, ((c_j s_j - \mu_j)^2 + (\mu_j - \mu_j^p)^2)/2 + b_j)$

$\mu_j \leftarrow N[c_j s_j + \mu_j^p, \sigma_j^2]$

$p_j \leftarrow \exp(\mu_j)/(1 + \sum_{k=1}^{d} \exp(\mu_k))$

EndFor

$p_{d+1} \leftarrow 1 - \sum_{k=1}^{d} p_k$

Appendix B: Non-negative Garrote Solutions when $X^T X = I$.

In this section we derive the non-negative garrote estimators under orthogonal designs. Recall a design matrix X is called orthogonal, or more properly, orthonormal, if $X^T X = I$. This implies

that $\sum_i x_{ij}^2 = 1$ and $\sum_i x_{ij}x_{ik} = 0$ for $j \neq k$.

Recall the non-negative garrote objective function is

$$\underset{\forall j: c_j \geq 0}{argmin} \ \sum_{i=1}^{n}(y_i - \sum_{j=1}^{d} c_j \hat{\beta}_j x_{ij})^2 + \lambda \sum_{j=1}^{d} c_j, \tag{82}$$

Now for simplicity of exposition we will take $d = 2$ and also assume the y_i have had their mean subtracted from each observation, removing the intercept term from the model. This gives us the objective

$$\underset{\forall j: c_j \geq 0}{argmin} \ \underbrace{\frac{1}{2}\sum_{i=1}^{n}(y_i - c_1\hat{\beta}_1 x_{i1} - c_2\hat{\beta}_2 x_{i2})^2 + \lambda(c_1 + c_2)}_{=f(c_1,c_2)}, \tag{83}$$

Now we optimize over c_j by taking derivatives. This results in a system of 2 linear equations, the details follow:

$$\frac{\partial f}{\partial c_1} = \sum_i (y_i - c_1\hat{\beta}_1 x_{i1} - c_2\hat{\beta}_2 x_{i2})(-\hat{\beta}_1 x_{i1}) + \lambda \overset{\text{set}}{=} 0 \tag{84}$$

$$\frac{\partial f}{\partial c_2} = \sum_i (y_i - c_1\hat{\beta}_1 x_{i1} - c_2\hat{\beta}_2 x_{i2})(-\hat{\beta}_2 x_{i2}) + \lambda \overset{\text{set}}{=} 0 \tag{85}$$

Multiplying the extra terms through gives us the equations

$$\frac{\partial f}{\partial c_1} = \sum_i (-y_i\hat{\beta}_1 x_{i1} + c_1\hat{\beta}_1 x_{i1}\hat{\beta}_1 x_{i1} + c_2\hat{\beta}_2 x_{i2}\hat{\beta}_1 x_{i1}) + \lambda \overset{\text{set}}{=} 0 \tag{86}$$

$$\frac{\partial f}{\partial c_2} = \sum_i (-y_i\hat{\beta}_2 x_{i2} + c_1\hat{\beta}_1 x_{i1}\hat{\beta}_2 x_{i2} + c_2\hat{\beta}_2\hat{\beta}_2 x_{i2}x_{i2}) + \lambda \overset{\text{set}}{=} 0 \tag{87}$$

Now after some algebra we get the analogs to the "normal" equations in least squares

$$\hat{\beta}_1 \sum_i y_i x_{i1} = \sum_i (c_1\hat{\beta}_1 x_{i1}^2 + c_2\hat{\beta}_2 x_{i1}x_{i2}) + \lambda \tag{88}$$

$$\hat{\beta}_2 \sum_i y_i x_{i2} = \sum_i (c_1\hat{\beta}_1 x_{i1}x_{i2} + c_2\hat{\beta}_2 x_{i2}^2) + \lambda \tag{89}$$

Now applying the sum through to the RHS of both equations and applying the orthonormal conditions we get

$$\hat{\beta}_1 \sum_i y_i x_{i1} = c_1 \hat{\beta}_1^2 + \lambda \tag{90}$$

$$\hat{\beta}_2 \sum_i y_i x_{i2} = c_2 \hat{\beta}_2^2 + \lambda \tag{91}$$

The orthonormal conditions imply $\sum_i x_{ij}^2 = 1$, so we divide the $\sum_i y_i x_{ij}$ terms by this "1" term. Noting that if the x_{ij}'s are centered about their means we have

$$\hat{\beta}_j = \frac{\sum_i x_{ij} y_i}{\sum_i x_{ij}^2} \tag{92}$$

Solving for c_1 and c_2 gives the equations

$$1 - \frac{\lambda}{\hat{\beta}_1^2} = c_1 \tag{93}$$

$$1 - \frac{\lambda}{\hat{\beta}_2^2} = c_2 \tag{94}$$

and noting that $c_j \geq 0$ implies we take the positive part. These are the closed form solutions given in [8]. The reader can now easily generalize to the case with $d > 2$ covariates. □

The interested reader may work out the analogous results for the constraint $\sum_{j=1}^{d} c_j^2 \leq s$ for some constant s. The closed form solution in the orthonormal X case is also given in [8].

Appendix C

Using the result from Villa and Escobar [91], which states.

Theorem

Suppose $M_{x|y}(t) = C_1(t) \exp[C_2(t)Y]$ and there exists a $\delta > 0$ such that for $t \in (-\delta, \delta)$, $|C_i(t)| < \infty$ and $|M_y(C_2(t))| < \infty$ assuming $M_y(t)$ exists, then $M_x(t) = C_1(t) M_y(C_2(t))$. In more common MGF notation we have $M_x(t) = \mathbb{E}_y(M_{x|y}(t))$. See Villa and Escobar for the proof.

We an use this result to understand the mixture of normals. To do this we recall the forms of the MGFs for three common distributions:

$$M(t) = \exp\left(\mu t + \frac{\sigma^2 t^2}{2}\right) \tag{95}$$

$$M(t) = \frac{1}{1-\lambda t} \tag{96}$$

$$M(t) = \frac{\exp(\mu t)}{1-b^2t^2} \tag{97}$$

In MGF's 95-97, the means of the densities are μ, λ, and μ respectively. Also, the variances of the densities are σ^2, λ^2, and b^2.

Now applying the Theorem of Villa and Escobar we have

$$\begin{aligned}
\mathbb{E}\left(M_{x|v}(t)\right) &= \mathbb{E}\left(\exp\left(\mu t + 2v\sigma^2t^2\right)\right) \\
&= \int_0^\infty \exp\left(-v + \mu t + 2v\sigma^2t^2\right) dv \\
&= \exp(\mu t)\int_0^\infty \exp\left(-v + 2v\sigma^2t^2\right) dv \\
&= \exp(\mu t)\int_0^\infty \exp\left(-v(1-2\sigma^2t^2)\right) dv \\
&= \frac{\exp(\mu t)}{1-2\sigma^2t^2}\underbrace{\int_0^\infty (1-2\sigma^2t^2)\exp\left(-v(1-2\sigma^2t^2)\right) dv}_{=1\text{, because it is an exponential pdf}} \\
&= \frac{\exp(\mu t)}{1-2\sigma^2t^2}
\end{aligned}$$

The result is shown.

9 References

[1] C. Andrieu, N. De Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43, 2003.

[2] W. Belson. Matching and prediction on the principle of biological classification. *Applied statistics*, pages 65–75, 1959.

[3] K. Berk. Comparing subset regression procedures. *Technometrics*, 20(1):1–6, 1978.

[4] A. Bhattacharya, D. Pati, N. S. Pillai, and D. B. Dunson. Bayesian shrinkage. *arXiv preprint arXiv:1212.6088*, 2012.

[5] G. Biau. Analysis of a random forests model. *The Journal of Machine Learning Research*, 98888:1063–1095, 2012.

[6] G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *The Journal of Machine Learning Research*, 9:2015–2033, 2008.

[7] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of LASSO and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

[8] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

[9] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[10] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[11] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[12] L. Breiman and J. H. Friedman. Comment. *Journal of the American Statistical Association*, 83(403):725–727, 1988.

[13] F. Bunea, A. Tsybakov, and M. Wegkamp. Sparsity oracle inequalities for the LASSO. *Electronic Journal of Statistics*, 1:169–194, 2007.

[14] E. Candes and Y. Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *Information Theory, IEEE Transactions on*, 57(4):2342–2359, 2011.

[15] E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[16] O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

[17] C. Carvalho, N. Polson, and J. Scott. Handling sparsity via the horseshoe. In *Journal of Machine Learning Research, W&CP*. Citeseer, 2009.

[18] C. Carvalho, N. Polson, and J. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[19] G. Casella and C. Robert. *Monte Carlo statistical methods*, volume 2. Springer New York, 1999.

[20] J. M. Chambers, T. Hastie, et al. *Statistical models in S*. Chapman & Hall London, 1992.

[21] H. Chipman, E. George, and R. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, pages 935–948, 1998.

[22] H. Chipman, E. George, and R. McCulloch. Hierarchical priors for Bayesian CART shrinkage. *Statistics and Computing*, 10(1):17–24, 2000.

[23] H. Chipman, E. George, and R. McCulloch. Bayesian treed models. *Machine Learning*, 48(1):299–320, 2002.

[24] T. Cormen, C. Lieserson, R. Rivest, and S. C. *Introduction to algorithms*. The MIT press, 2001.

[25] W. Cui and E. I. George. Empirical Bayes vs. fully Bayes variable selection. *Journal of Statistical Planning and Inference*, 138(4):888–900, 2008.

[26] P. Damien and S. G. Walker. Sampling truncated normal, beta, and gamma densities. *Journal of Computational and Graphical Statistics*, 10(2), 2001.

[27] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156, 1997.

[28] D. Denison, B. Mallick, and A. Smith. A Bayesian CART algorithm. *Biometrika*, 85(2):363–377, 1998.

[29] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Verlag, 1996.

[30] A. Dobra and J. Gehrke. SECRET: a scalable linear regression tree algorithm. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 481–487. ACM, 2002.

[31] K. Doksum, S. Tang, and K.-W. Tsui. Nonparametric variable selection: the EARTH algorithm. *Journal of the American Statistical Association*, 103(484):1609–1620, 2008.

[32] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.

[33] M. Dorigo and T. Stützle. *Ant Colony Optimization*. Bradford Bks. BRADFORD BOOK, 2004.

[34] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[35] B. Efron and R. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, pages 548–560, 1997.

[36] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*, volume 57. Chapman & Hall/CRC, 1994.

[37] M. Efroymson. Multiple regression analysis. *Mathematical methods for digital computers*, 1:191–203, 1960.

[38] A. Frank and A. Asuncion. UCI Machine Learning Repository, 2010.

[39] I. E. Frank and J. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[40] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[41] J. Friedman. Remembering Leo Breiman. *arXiv preprint arXiv:1101.0934*, 2011.

[42] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman New York, 1979.

[43] A. E. Gelfand and A. F. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990.

[44] A. Gelman. Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–533, 2006.

[45] E. George and R. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[46] E. I. George. The variable selection problem. *Journal of the American Statistical Association*, 95(452):1304–1308, 2000.

[47] J. Geweke. Variable selection and model comparison in regression. *Bayesian statistics*, 5:609–620, 1996.

[48] S. Gey and E. Nedelec. Model selection for CART regression trees. *Information Theory, IEEE Transactions on*, 51(2):658–670, 2005.

[49] R. Gramacy and H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

[50] R. Gramacy and M. Taddy. Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models. *Journal of Statistical Software*, 33:1–48, 2012.

[51] J. Gray and G. Fan. Classification tree analysis using TARGET. *Computational Statistics & Data Analysis*, 52(3):1362–1372, 2008.

[52] P. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[53] H. Halvorson. Regression and systems of equations analysis on the IBM-650. *Journal of Farm Economics*, 42(5):1450–1458, 1960.

[54] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[55] T. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, 1998.

[56] A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[57] A. Huang and M. Wand. Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, 8(1):1–14, 2013.

[58] H. Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519–537, 2007.

[59] H. Ishwaran, U. B. Kogalur, E. Z. Gorodeski, A. J. Minn, and M. S. Lauer. High-dimensional variable selection for survival data. *Journal of the American Statistical Association*, 105(489):205–217, 2010.

[60] V. E. Johnson and D. Rossell. On the use of non-local prior densities in Bayesian hypothesis tests. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(2):143–170, 2010.

[61] G. Kass. Significance testing in automatic interaction detection (AID). *Applied Statistics*, pages 178–189, 1975.

[62] G. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, pages 119–127, 1980.

[63] V. Koltchinskii. The Dantzig selector and sparsity oracle inequalities. *Bernoulli*, 15(3):799–828, 2009.

[64] N. Kushmerick. Learning to remove internet advertisements. In *Proceedings of the third annual conference on Autonomous Agents*, pages 175–181. ACM, 1999.

[65] S. Lee and S. Jin. Decision tree approaches for zero-inflated count data. *Journal of applied statistics*, 33(8):853–865, 2006.

[66] S. Leman, Y. Chen, and M. Lavine. The multiset sampler. *Journal of the American Statistical Association*, 104(487):1029–1041, 2009.

[67] B. Liu, H. A. Abbas, and B. McKay. Classification rule discovery with ant colony optimization. In *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, pages 83–88. IEEE, 2003.

[68] M. Loeve. Probability theory, volume i. *Graduate Texts in Mathematics, Springer-Verlag, New York (fourth edition 1977)*, 1977.

[69] W.-Y. Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386, 2002.

[70] W.-Y. Loh and N. Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *Journal of the American Statistical Association*, 83(403):715–725, 1988.

[71] A. Miller. Selection of subsets of regression variables. *Journal of the Royal Statistical Society. Series A (General)*, pages 389–425, 1984.

[72] A. Miller. *Subset selection in regression*. Chapman & Hall/CRC, 2002.

[73] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

[74] J. Morgan and J. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, pages 415–434, 1963.

[75] R. Myers. *Classical and modern regression with applications*, volume 2. Duxbury Press Belmont, CA, 1990.

[76] R. O'Hara and M. Sillanpää. A review of Bayesian variable selection methods: what, how and which. *Bayesian Analysis*, 4(1):85–117, 2009.

[77] M. Osborne, B. Presnell, and B. Turlach. On the LASSO and its dual. *Journal of Computational and Graphical statistics*, 9(2):319–337, 2000.

[78] T. Park and G. Casella. The Bayesian LASSO. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[79] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas. Data mining with an ant colony optimization algorithm. *Evolutionary Computation, IEEE Transactions on*, 6(4):321–332, 2002.

[80] N. Polson and J. Scott. On the half-Cauchy prior for a global scale parameter. *arXiv preprint arXiv:1104.4937*, 2011.

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

[82] J. Quinlan. *C4.5: Programs for machine learning*. Morgan kaufmann, 1993.

[83] M. Sauvé and C. Tuleau-Malot. Variable selection through CART. *Arxiv preprint arXiv:1101.0689*, 2011.

[84] D. Sleator and R. Tarjan. Self-adjusting binary search trees. *Journal of the ACM (JACM)*, 32(3):652–686, 1985.

[85] J. Steele. Empirical discrepancies and subadditive processes. *The Annals of Probability*, 6(1):118–127, 1978.

[86] M. A. Taddy, R. B. Gramacy, and N. G. Polson. Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106(493):109–123, 2011.

[87] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[88] M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[89] V. Vapnik. *The nature of statistical learning theory*. Springer Verlag, 2000.

[90] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[91] E. R. Villa and L. A. Escobar. Using moment generating functions to derive mixture distributions. *The American Statistician*, 60(1), 2006.

[92] B. D. Ville. *Decision Tree for Business Intelligence and Data Mining*. SAS Publishing, 2006.

[93] Y. Wu, H. Tjelmeland, and M. West. Bayesian CART: Prior specification and posterior simulation. *Journal of Computational and Graphical Statistics*, 16(1):44–66, 2007.

[94] S. Xiong. Some notes on the nonnegative garrote. *Technometrics*, 52(3):349–361, 2010.

[95] N. Yi, V. George, and D. B. Allison. Stochastic search variable selection for identifying multiple quantitative trait loci. *Genetics*, 164(3):1129–1138, 2003.

[96] M. Yuan and Y. Lin. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007.

[97] P. Zhao and B. Yu. On model selection consistency of LASSO. *Journal of Machine Learning Research*, 7(2):2541, 2007.

[98] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.