Bayesian Ensemble of Regression Trees for Multinomial Probit and Quantile Regression

Bereket P. Kindo
University of South Carolina

Follow this and additional works at: https://scholarcommons.sc.edu/etd
Part of the Statistics and Probability Commons

Recommended Citation

This Open Access Dissertation is brought to you by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact dillarda@mailbox.sc.edu.
Bayesian ensemble of regression trees for multinomial probit and quantile regression

by

Bereket P. Kindo

Bachelor of Science
University of Eastern Africa, Baraton 2005

Master of Arts
Ball State University 2009

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in
Statistics
College of Arts and Sciences
University of South Carolina
2016

Accepted by:
Edsel A. Peña, Major Professor
Timothy Hanson, Committee Member
Xianzheng Huang, Committee Member
Marco Valtorta, Committee Member
Cheryl L. Addy, Vice Provost and Dean of the Graduate School
DEDICATION

To my wife, my daughter, my family and all the wonderful teachers I’ve had.
ACKNOWLEDGMENTS

I am grateful to have worked under the supervision of Dr. Edsel Peña and Dr. Hao Wang. I am very thankful for their guidance during this dissertation work and for instilling in me the joy of research. I also thank Dr. Timothy Hanson for his guidance on the work that constituted the last chapter this work. Moreover, I would like to thank Dr. Edsel Peña for organizing weekly research meetings and the research group which included Professor James Lynch, Dr. A.K.M Rahman, Dr. Piaomu Liu and Lillian Wanda. The insightful discussions and comments in these meetings were instrumental in shaping this dissertation as well as exposing me to a broader area of Statistics. I extend my gratitude to the members of the dissertation committee: Dr. Timothy Hanson, Dr. Xianzheng Huang and Dr. Marco Valtorta for their valuable comments.

I would like to thank my professors in the Department of Statistics for classes I attended have greatly benefited me. I am also thankful for the financial support accorded to me by the Department of Statistics and the graduate school at the University of South Carolina.

I also acknowledge the support I received through National Science Foundation Grant DMS1106435 and National Institutes of Health Grants R01CA154731, RR17698, and 1P30GM103336-01A1.

I also thank Professor Edward I. George for his 2012 Palmetto Lecture at the University of South Carolina, which partly motivated this research.

I thank my wife Christine Kindo, my daughter Hannah Kindo, and my parents and siblings for all the support and motivation.
Abstract

This dissertation proposes multinomial probit Bayesian additive regression trees (MPBART), ordered multiclass Bayesian additive classification trees (O-MBACT) and Bayesian quantile additive regression trees (BayesQArt) as extensions of BART - Bayesian additive regression trees for tackling multinomial choice, multiclass classification, ordinal regression and quantile regression problems. The proposed models exhibit very good predictive performances. In particular, ranking among the top performing procedures when non-linear relationships exist between the response and the predictors. The proposed procedures can readily be applied on data sets with the number of predictors larger than the number of observations.

MPBART is sufficiently flexible to allow inclusion of predictors that describe the observed units as well as the available choice alternatives and it can also be used as a general multiclass classification procedure. Through two simulation studies and four real data examples, we show that MPBART exhibits very good out-of-sample predictive performance in comparison to other discrete choice and multiclass classification methods. To implement MPBART, the R package mpbart is freely available from CRAN repositories.

When ordered gradation is exhibited by a multinomial response, ordinal regression is an appealing framework. Ensemble of trees models, while widely used for binary classification, multiclass classification and continuous response regression, have not been extensively applied to solve ordinal regression problems. This work fills this void with Bayesian sum of regression trees. The predictive performance of our ordered Bayesian ensemble of trees model is illustrated through simulation studies and real
data applications.

Ensemble of regression trees have become popular statistical tools for the estimation of conditional mean given a set of predictors. However, quantile regression trees and their ensembles have not yet garnered much attention despite the increasing popularity of the linear quantile regression model. This work proposes a Bayesian quantile additive regression trees model that shows very good predictive performance illustrated using simulation studies and real data applications. Further extension to tackle binary classification problems is also considered.
# Table of Contents

Dedication ........................................ iii

Acknowledgments ................................. iv

Abstract .......................................... v

List of Tables .................................... x

List of Figures .................................. xiv

Chapter 1 Introduction ........................ 1
  1.1 Review of multiclass classification procedures ............... 3

Chapter 2 Multinomial probit Bayesian additive regression trees ¹ ........................................ 16
  2.1 Introduction ................................ 16
  2.2 MPBART: Multinomial probit Bayesian additive regression trees . 19
  2.3 Prior specifications and posterior computation ................... 20
  2.4 Synthetic data examples ................................ 25
  2.5 Real data examples ................................ 29
  2.6 Conclusion ................................ 31

Chapter 3 Multiclass Bayesian Additive Classification Trees for Ordered Response ................. 35
3.1 Introduction ......................................................... 35
3.2 Ordered multiclass Bayesian additive classification trees
(O-MBACT) ................................................................. 38
3.3 Prior specifications and posterior computation for ordered multi-
class Bayesian additive classification trees ......................... 39
3.4 Simulation study for ordered MBACT ............................. 42
3.5 Real data illustration ................................................. 44
3.6 Conclusion ............................................................. 45

CHAPTER 4  BAYESIAN QUANTILE ADDITIVE REGRESSION TREES ² ........................................ 49
4.1 Introduction ............................................................ 49
4.2 Bayesian quantile additive regression trees ......................... 53
4.3 Prior specifications .................................................. 54
4.4 Posterior updating scheme ......................................... 56
4.5 Data analysis .......................................................... 59
4.6 Simulation study ...................................................... 59
4.7 Real data examples .................................................. 62
4.8 Binary classification extension ..................................... 65
4.9 Binary classification simulation study ............................. 67
4.10 Real data for binary classification ................................. 69
4.11 Conclusion ............................................................. 70

CHAPTER 5  CONCLUSIONS ................................................. 72

BIBLIOGRAPHY .......................................................... 74
Table 1.1 Pre-specified costs for the loss function $L(y,a)$. 4

Table 2.1 List of competing classifiers, the $R$ packages utilized, and tuning parameters that are chosen by cross-validation. The abbreviations in the first column stand for the procedures mentioned in the second paragraph of this chapter. 27

Table 2.2 Comparison of MPBART and the procedures listed in Table 2.1 on the first simulation study generated via (2.10) and the waveform recognition example (2.12). Training and test data sets of each 500 observations are used for the first simulation study. Training and test data sets of 300 and 500 observations, respectively are used for the waveform recognition example. Average test error rates (with standard errors in parentheses) are reported on 20 replications. 28

Table 2.3 Comparison results on the fishing mode and choice of travel mode data sets. Classification error rates (with standard errors in parentheses) are reported. 30

Table 2.4 Classification error rates and standard errors (in parentheses) for vertebral column and forensic glass data sets. 31

Table 2.5 Travel mode data selected tuning parameters. The number of posterior samples is 2000 with the first 500 discarded. 32

Table 2.6 Fishing mode data selected tuning parameters. The number of posterior samples is 2000 with the first 500 discarded. 32

Table 2.7 Waveform recognition dataset selected tuning parameters. The number of posterior samples is 1300 with the first 300 discarded. 33

Table 2.8 Friedman multiclass classification data set selected tuning parameters. The number of posterior samples is 1000 with the first 500 discarded. 34
Table 3.1 Commonly used link functions for ordinal cumulative link models, where \( \Phi (\cdot) \) denotes the cumulative distribution function of the standard normal distribution. 36

Table 3.2 Average MADs (Mean Absolute Deviations) as defined in (3.13) with standard errors in parentheses for Nonlinear Ordered Simulated Data 43

Table 3.3 Selected tuning parameters for the nonlinear ordered simulated data. The number of trees used is 200 with posterior sample of size 3,000 after discarding the initial 1,000. 44

Table 3.4 Description of ordered real data sets. 45

Table 3.5 Results of ordered real data sets Wisconsin and Abalone. This table shows MADs defined in (3.13), and standard errors (in parentheses) on test data sets over the number of replications mentioned in the last row of Table 3.4. 46

Table 3.6 Results of ordered real data sets Triazines and Ailerons. This table shows MADs defined in (3.13), and standard errors (in parentheses) on test data sets over the number of replications mentioned in the last row of Table 3.4. 46

Table 3.7 Selected tuning parameters for Ailerons dataset. The other tuning parameters that are not selected via cross validation are \( n_T = 200, \kappa = 3.0 \). The number or posterior samples is 3000 with initial 1000 discarded. 47

Table 3.8 Selected tuning parameters for Triazines dataset. The other tuning parameters that are not selected via cross validation are \( n_T = 200, \kappa = 3.0 \). The number or posterior samples is 3000 with initial 1000 discarded. 47

Table 3.9 Selected tuning parameters for Wisconsin dataset. The other tuning parameters that are not selected via cross validation are \( n_T = 200, \kappa = 3.0 \). The number or posterior samples is 3000 with initial 1000 discarded. 47

Table 3.10 Selected tuning parameters for Abalone dataset. The other tuning parameters that are not selected via cross validation are \( n_T = 200, \kappa = 3.0 \). The number or posterior samples is 3000 with initial 1000 discarded. 48
Table 4.1  First quantile regression simulation study results: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses over 20 replications. 60

Table 4.2  Selected tuning parameters for the first simulation study for quantile regression. Tuning parameters $n_0 = 5$, $d = 10$ are used as the number of minimum observations in a terminal node and the maximum depth of each regression tree in the sum. The number of posterior samples is 30,000 with the first 10,000 discarded, and the number of trees in the sum is equal to 50. 61

Table 4.3  Second quantile regression simulation study results: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses based on 20 replications. 62

Table 4.4  Selected tuning parameters for the second simulation study for quantile regression. Tuning parameters $n_0$ denotes the number of minimum observations in a terminal node. The maximum depth allowed for any tree in the sum is 3 and the value of $\psi_2$ used is 3.0. The number of posterior samples is 30,000 with the first 10,000 discarded, and the number of trees in the sum is equal to 200. 63

Table 4.5  Ozone data set: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses based on 5 consecutive splits of the data as they appear in the R package datasets. 63

Table 4.6  Selected tuning parameters for the Ozone data quantile additive regression trees real data example. 64

Table 4.7  Auto insurance claims data set: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses based on 10 splits. 66

Table 4.8  Selected tuning parameters for the Auto insurance claims data quantile additive regression trees real data example. 66

Table 4.9  Simulation study for binary classification: test data averages of test data classification error rate, area under the ROC curve (AUC), and their standard errors in parentheses based on 20 replications. 68
Table 4.10 Cancer classification results: test data averages of classification error rate, area under the ROC curve (AUC), and their standard errors.

Table 4.11 Cancer classification results: tuning parameters selected via cross validation. The number of trees used is 200 and a value of 3.0 is used for \( \kappa \), and \( \psi_2 \). We have limited the depth of each tree to \( d \), and the minimum number of observations allowed in each terminal node to \( n_0 \). Posterior sample size of 10,000 is used with the first 3,000 discarded.
List of Figures

Figure 1.1 Illustration of GROW move \( T^{(1)} \to T^{(2)} \), CHANGE move \( T^{(2)} \to T^{(3)} \), SWAP move \( T^{(3)} \to T^{(4)} \) and PRUNE move \( T^{(4)} \to T^{(5)} \). 11

Figure 3.1 This figure illustrates the nonlinearity of the relationship between the predictors and ordinal response of the simulated data. 43

Figure 4.1 Asymmetric Laplace distribution with \( \mu = 0 \) and values of \( \tau \in \{0.25, 0.50, 0.83\} \). 51

Figure 4.2 Predicted conditional quantiles against the actual response for the first simulation study. 60

Figure 4.3 Circle in square - two dimensional illustration of class boundaries of the binary classification simulation study for Bayesian quantile additive regression trees. 69
Classification and regression are essential parts of many fields of studies such as engineering, agriculture, biomedical science, social science and business where the objectives of classifying units into one of several categories and/or prediction of a continuous target variable arise. An ample understanding of the influence of predictors on different quantiles of the response variable is also an integral part of business decisions and scientific studies that have of late depended heavily on insights obtained from data. Moreover, technological advancements in data collection and storage have resulted in the need to analyze large data sets, and hence the need for statistical procedures that can be used to analyze them. This dissertation proposes the modeling of multinomial choice, multiclass classification, ordinal regression and quantile regression problems using ensemble of Bayesian regression trees. The proposed models are amenable to large data sets in which the number of predictors may be larger than the number of observations.

Consider the following four real data examples to motivate multiclass classification, ordinal classification, multinomial choice and the quantile regression problems considered in this dissertation.

- **Example 1. Multiclass classification.** Classification of an aerial image of an urban area as either “trees”, “grass”, “soil”, “concrete”, “asphalt”, “buildings”, “cars”, “pools” or “shadows” using attributes that describe the image such as color, brightness, compactness, texture and shape. This data set is available at the machine learning repository of the University of California at Irvine (Bache
and Lichman, 2013) and it is analyzed in Johnson and Xie (2013).

- **Example 2. Ordinal classification.** Financial bonds issued by governments (local or national) and corporations often receive a credit rating which classifies the creditworthiness of the issuing organization into ordered multinomial gradations. Typically, a firm’s financial information which includes debt level, profit, and revenue along with other macroeconomic variables are used as input variables. In this example, there is a natural ordering of the classes, for example, Standard and Poor’s rating agency currently uses the gradations AAA, AA, A, BBB, BB, B, and so forth to discriminate the credit risks associated with financial obligations; with the class AAA indicating the highest level of creditworthiness.

- **Example 3. Multinomial choice data.** An example of multinomial choice data is the fishing mode choice data set in which an individual chooses from the fishing modes “beach”, “pier”, “boat” and “charter” (Kling and Thomson, 1996; Herriges and Kling, 1999). Multinomial choice modeling deals with understanding how such a choice is made given predictors that describe the choice alternatives as well as predictors that describe the decision maker. In the fishing mode data set, the individual specific predictor, describing the decision maker, is monthly income of the individual. The choice specific predictors, describing the available choice alternatives, are expected catch rates per hour and price for each mode of fishing.

- **Example 4. Quantile regression.** Quantile regression complements least squares regression by using conditional quantiles in contrast to conditional means to explain the relationship between a set of predictors and a dependent variable. For example, the auto insurance claims data set in Qian et al. (2015) consists of predictors that describe insurance policyholders and a response variable that
records claim amounts paid by the insurer. This response variable in this example is right skewed making quantile regression an appealing framework to understand the relationship between higher claims and the predictors.

The remaining sections of this chapter outline a general formulation of the multiclass classification problem followed by a review of some widely used statistical procedures for multiclass classification procedures; discuss the use of multinomial probit models for discrete choice modeling and ordinal regression; and a review of statistical procedures used for obtaining conditional quantiles.

1.1 REVIEW OF MULTICLASS CLASSIFICATION PROCEDURES

Multiclass classification deals with classifying units into one of the $K > 2$ classes. Given a data set with $p$ predictors, say $x \in \mathbb{R}^p$, and dependent variable $y \in \{1, 2, \ldots, K\}$, multiclass classifier obtains a function $h : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} = \{1, 2, \ldots, K\}$ and $\mathcal{X} = \mathbb{R}^p$. We first outline a mathematical framework for the multiclass classification problem that takes into account the severity of a mis-classification. It is natural to think that no cost will be incurred if a unit is classified correctly and that a nonzero cost be associated with a mis-classification. Moreover, in multiclass classification setting, not every mis-classification results in an equal amount of loss. For example, suppose that a tumor can be treated with noninvasive medical prescriptions. A mis-diagnosis of such a tumor as requiring surgical removal is associated with a higher cost than a mis-diagnosis as requiring noninvasive therapy. With such scenarios in mind, we outline a “cost efficient” mathematical formulation of a multiclass classification problem.

Consider a training data set $(y_i, x_i)$ for $i = 1, \ldots, n$, where $y_i \in \mathcal{Y} = \{1, \ldots, K\}$ and $x_i \in \mathcal{X} = \mathbb{R}^p$ denote the observed class and predictors for the $i^{th}$ individual or unit, respectively. We make the assumption that $(y_i, x_i), i = 1, \ldots, n$ are observations from independent and identically distributed random variables $(Y_i, X_i), i = 1, \ldots, n$
Table 1.1 Pre-specified costs for the loss function $L(y, a)$.

<table>
<thead>
<tr>
<th>Loss</th>
<th>Prediction a</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>$C_{11}$</td>
</tr>
<tr>
<td>2</td>
<td>$C_{21}$</td>
</tr>
<tr>
<td>True Class y</td>
<td>:</td>
</tr>
<tr>
<td>K</td>
<td>$C_{K1}$</td>
</tr>
</tbody>
</table>

that are governed by the probability distribution $P(Y, X)$. Our goal is to classify a new test unit with a vector of predictors $x \in \mathcal{X}$ into one of $K$ classes. Hence, a multiclass classification problem deals with developing a classifier or function $\rho(x) = \rho(x; Y, X) : \mathcal{X} \times (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{Y}$.

Suppose that $y \in \mathcal{Y}$ is the unobserved true class of a test unit that has features $x$. If we know $p_k(x) = P(y = k|x)$, the probability that a unit belongs to class $k$ given its features, for $k = 1, \ldots, K$, or have a “good” estimator $\hat{p}_k(x)$ of $p_k(x)$, we can utilize these probabilities to make “cost-efficient” class predictions. Suppose that if the prediction or action $\rho(x) = a \in \mathcal{Y}$ is taken when, in fact, the true class is $y = l$, a pre-specified cost $C_{la} \geq 0$ is incurred. That is, there is a cost function $C(y, a)$ that assigns a pre-specified loss to every combination of action $a \in \mathcal{Y}$ and true class $y \in \mathcal{Y}$; $C(y, a) : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$. The pre-specified cost combinations are described in Table 1.1.

The idea behind “cost-efficient” decision is to utilize the class probability estimates so that the overall average cost is minimized. Table 1.1 can equivalently be expressed via the loss function

$$L(y, a) = \sum_{l=1}^{K} \sum_{m=1}^{K} C_{lm} I(y = l, a = m).$$

(1.1)

where $I(\cdot)$ is the usual indicator function. If the mis-classification costs are all equal, and the correct classification costs are all zero (i.e., $C_{lm} = C > 0$ for $l \neq m$, and $C_{ll} =$
0; \ l, \ m \in \{1, \ldots, K\}\), the loss function simplifies to
\[ L(y, a) = C I(y \neq a). \] (1.2)

In order to arrive at the classification (prediction) of the test unit, we seek to minimize the average loss associated with the decision \( \rho(x) \). That is, we minimize \( E_{(Y,X)} L(Y, \rho(X)) \) which we can rewrite as

\[
E_{(Y,X)} L(Y, \rho(X)) = E_X E_{Y|x} L(Y, \rho(X)) \\
= E_X \sum_{l=1}^{K} \sum_{m=1}^{K} C_{lm} E(I(\rho(X) = m)I(Y = l) \mid X = x) \\
= E_X \sum_{l=1}^{K} \sum_{m=1}^{K} C_{lm} I(\rho(x) = m)P(Y = l|x).
\]

The optimal action, in this case, is the class \( m \) that minimizes the quantity

\[
\sum_{l=1}^{K} C_{lm} I(\rho(x) = m)P(Y = l|x).
\]

Note that

\[
\arg \min_{m} \sum_{l=1}^{K} C_{lm} I(\rho(x) = m)P(Y = l|x) = \arg \min_{m} \sum_{l=1}^{K} C_{lm} p_l(x).
\]

Hence, with a “good” estimator \( \hat{p}_l(x) \) of \( p_l(x) \) obtained by learning on the training data set, we predict the test unit to belong to the class

\[
\hat{y} = \arg \min_{m} \sum_{l=1}^{K} C_{lm} \hat{p}_l(x).
\] (1.3)

**Some multiclass classification procedures**

We now review some of the widely used regression and classification procedures that can be utilized to tackle multiclass classification problems, namely the multinomial logistic regression, k-nearest neighbors classifier, linear and quadratic discrimination analyses, and the tree based classification procedures: classification and regression
trees (CART) and random forests. In multinomial logistic regression the probability that a unit belongs to the $k^{th}$ class is assumed to depend on a linear combination of the predictors as

$$\log \frac{P(y = k \mid x)}{P(y = K \mid x)} = \beta_{k0} + x^\top \beta_k,$$

for $k = 1, \ldots, K - 1$, where $x \in \mathbb{R}^p$, and $y \in \{1, 2, \ldots, K\}$ is the multiclass response. Multinomial logistic maximum likelihood estimates are

$$\hat{\beta} = \arg \max_{(\beta_0, \beta)} \sum_{i=1}^n \sum_{k=1}^K I(y_i = k) \log [P(y_i = k \mid x_i, \ldots, \beta_0, \beta_k)],$$

where $\beta_0 = (\beta_{10}, \ldots, \beta_{K-1,0})$, and $\beta = (\beta_1, \ldots, \beta_{K-1})$. When the number of predictors is larger than the sample size, regularized multinomial logistic regression (Zou and Hastie, 2005; Friedman et al., 2010) can be used to shrink some of the parameter estimates to zero, hence performing a variable selection. The regularized parameter estimates are

$$\hat{\beta} = \arg \max_{\beta} \sum_{i=1}^n \sum_{k=1}^K I(y_i = k) \log [P(y_i = k \mid x_i, \ldots, \beta_0, \beta_k)] + p_\lambda(\beta),$$

where the lasso regularization penalty is obtained when $\alpha = 1$ in the penalty function

$$p_\lambda(\beta) = \lambda \sum_{j=1}^p \sum_{k=1}^{K-1} \left\{ \alpha \mid \beta_{kj} \mid + (1 - \alpha) \beta_{kj}^2 \right\}.$$

KNN, k-Nearest Neighbors (Fix and Hodges Jr, 1952; Cover and Hart, 1967), is one of the oldest nonparametric classification techniques based on the intuition that, as described by Thomas Cover in Cover (20, March 1982), “things that look alike must be alike.” For each unit in the test data set, KNN picks the most represented class of the $k$ closest/nearest training predictors as the predicted class, where $k$ is an integer and the closeness of a point is measured in terms of Euclidean distance. The use and properties of KNN in multiclass classification have been studied in Cover and Hart (1967); Bay (1998); Athitsos and Sclaroff (2005).

Quadratic Discriminant Analysis (QDA) is based on the assumptions that the prior probability that a unit belongs to class $k$ is $\pi_k$ with $\sum_{k=1}^K \pi_k = 1$, and that, given
class \( y = k \); the vector of predictors \( \mathbf{x} \) follows a multivariate normal distribution, 
\( \mathbf{x} | y = k \sim N(\mu_k, \Sigma_k) \) for \( k = 1, \ldots, K \). Applying Bayes Theorem, the posterior probability of belonging to class \( k \) is obtained by

\[
P(y = k | \mathbf{x}) = \frac{f(\mathbf{x} | y = k) \pi_k}{\sum_{i=1}^{K} f(\mathbf{x} | y = i) \pi_i}.
\]

Linear Discriminant Analysis (LDA) is a special case of QDA obtained by setting \( \Sigma_k = \Sigma \) for all \( k = 1, \ldots, K \). Details on LDA and QDA are found in Duda et al. (2012); Friedman et al. (2001), among others.

For binary response classification, Support Vector Machines (SVM) (Cortes and Vapnik, 1995; Vapnik, 2013) finds an optimal hyperplane in the predictor space that seeks to separate one response class from the other. When the data are separable, two unique hyperplanes can be obtained by maximizing the distance between the closest two points (support points) in either class. The separating hyperplane is one that is equidistant from the two hyperplanes. When the data are not completely separable, some perturbation is induced so that some points are allowed to be on the “wrong” side of the hyperplane. Nonlinear hyperplanes are induced in the predictor space by projecting linear hyperplanes obtained in an extended predictor space with the extension done via basis expansions which include the use of polynomial, spline, wavelet, and reproducing kernel Hilbert space expansions, among others. Extensive study of SVM is found in Vapnik (2013); Friedman et al. (2001). The multiclass extension of SVM is often implemented by reduction of the multiclass problem into a sequence of binary classification problems. Hsu and Lin (2002); Lee et al. (2004); Zhang (2004) have studied the theoretical and application aspects of multi-category support vector machines which does not reduce the problem into several binary classification problems.
**Regression trees and their ensembles**

Ensemble of regression trees have shown good predictive performance in cases where the number of predictors is larger than the number of samples and when the predictors are related to the response in a non-linear fashion. Classification and Regression Trees (CART) (Breiman et al., 1984; Quinlan, 1986) is the building block of classical ensemble of trees models such as random forest and gradient boosting (Breiman, 2001; Friedman, 2001). CART works by creating recursive binary partitions of the predictor space that take the form \( \{ x_j > c(x_j) \} \) versus \( \{ x_j \leq c(x_j) \} \), where \( x_j \) is the \( j^{\text{th}} \) predictor, and \( c(x_j) \) a constant that lies within the range of \( x_j \). The partitioning of the space is followed by fitting simple functions (e.g., constant function) within each resulting partition. For each recursive split, the selection of a predictor and splitting value is done so that a pre-specified loss is minimized (i.e., maximum information gain is obtained). CART is considered one of the earliest nonlinear and nonparametric classification and regression methods that handles both binary and multiclass classification problems. Advances in computing power have enabled the use of ensemble of classification and regression trees to construct powerful techniques such as random forests and gradient boosting machine.

Random Forest (Breiman, 2001, 2000) is an ensemble learning method in which each classifier is a classification and regression tree generated in the following manner.  

(i) A bootstrap sample of the training data is selected to fit a CART like model without pruning. The partitions of the training data are termed “bootstrap” and “out-of-bootstrap” samples, where the “bootstrap” portion is used to construct the classification and regression tree. (ii) At each node, a subset of the predictors is randomly selected in order to determine the best predictor-splitting value pair. The predictor and splitting value pair that maximize the information gain is then used to grow the tree. (iii) Each learner (tree) votes on a class membership, and the majority vote is taken as the Random Forest prediction.
Bayesian CART and BART

Bayesian CART is formulated as an analog of classification and regression trees in Chipman et al. (1998) and further utilized as a building block of a more advanced regression technique BART - Bayesian Additive Regression Trees Chipman et al. (2010). The Bayesian CART model is described as follows:

\[ y_i = g(x_i, T, M) + \epsilon_i, \quad \epsilon_i \sim \text{N} \left(0, \sigma^2\right) \]

where \( T \) is a regression tree, \( M \) is the set of terminal node parameters of the tree \( T \), and \( g(\cdot) \) is a function that assigns a given vector of predictors \( x \) to one of the terminal node parameters in \( M \). The tree \( T \), the set of terminal node values \( M \) and \( \sigma^2 \) are the parameters of the Bayesian CART model each with prior specification and posterior updating scheme. See Figure 1.1 for a pictorial display of a regression tree. The initial node and all other internal nodes are characterized by a splitting variable \( v \) and a splitting point \( s \). For example, if the splitting variable is \( v = x_1 \), for a splitting point \( s \), the resulting left child node contains observations which have values of \( x_1 < s \) and the right node contains observations for which the values of \( x_1 \geq s \). The prior on \( T \) is governed by a “tree generating stochastic process” which stipulates certain rules for partitioning the predictor space in order to construct a regression tree. Given a tree \( T \), the terminal node parameters are assigned a Gaussian prior, the prior on \( \sigma^2 \) is specified through an Inverse Gamma distribution. These priors together with a Bayesian CART model likelihood allow the computation of the posteriors of the Bayesian CART model parameters. The choice of conjugate priors for the terminal node parameters and the scale parameter greatly simplifies the Metropolis-Hastings algorithm that is used to obtain a posterior update of the tree structure (Chipman et al., 1998) since \( M \) and \( \sigma \) can analytically be integrated out to get

\[ f \left( \mathbf{Y} \mid T, \mathbf{X} \right) = \int_{\Theta=M,\sigma} f \left( \mathbf{Y} \mid \mathbf{X}, M, T, \sigma \right) p \left( M \mid T, \sigma \right) p \left( \sigma \right) d\Theta. \quad (1.7) \]
The posterior updating scheme of the tree structure in Chipman et al. (1998) iteratively updates a tree $T$ using the moves GROW, PRUNE, CHANGE and SWAP. The GROW move randomly selects a terminal node and splits it into left and right nodes. The PRUNE serves as the reverse move of GROW. The CHANGE move selects an internal node and assigns a new splitting variable and splitting value, while the SWAP rule interchanges the splitting rules of two internal nodes that are parent and child. These rules are pictorially displayed in Figure 1.1.

BART - Bayesian Additive regression trees Chipman et al. (2010) uses the Bayesian CART as a building block to form a sum of regression trees. The BART model is briefly described as follows.

Consider the model

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \text{N}(0, \sigma^2).$$

BART uses sum of regression trees to approximate the functional relationship between the predictors and the response vector as $f(x) \approx \sum_{j=1}^{n_T} g(x; T_j, M_j)$. The parameters of BART are $\sigma^2$, and the $n_T$ regression trees along with their associated sets of terminal node parameters $(T_j, M_j)$, for $j = 1, \ldots, n_T$. BART uses prior specifications that aid in shrinking the effect of each individual tree in the sum. The weak trees as an ensemble, build a model that has excellent predictive accuracy. BART is utilized with great success in many applied settings that include phishing detection (Abu-Nimeh et al., 2007); credit risk modeling (Zhang and Härdle, 2010); electrical and computer engineering (Yu et al., 2010); and survival analysis (Bonato et al., 2011) among others.

**Multinomial probit models**

To review the linear multinomial probit model, consider an individual who is faced with a choice of $K > 2$ alternatives. Suppose $y \in \{1, \ldots, K\}$ is the choice made
Figure 1.1 Illustration of GROW move ($T^{(1)} \rightarrow T^{(2)}$), CHANGE move ($T^{(2)} \rightarrow T^{(3)}$), SWAP move ($T^{(3)} \rightarrow T^{(4)}$) and PRUNE move ($T^{(4)} \rightarrow T^{(5)}$).
by the individual, \( x \in \mathbb{R}^p \) a set of predictors that describe the individual (e.g., income, gender, age, etc.), and \( w_k \in \mathbb{R}^d \) set of predictors that describe the \( k \)th choice alternative. In the linear multinomial probit modeling approach, each individual is assumed to construct an unobserved latent variable \( z_k \) (also known as utility in the econometrics literature) for each alternative and select the alternative that results in the highest utility. That is, suppose

\[
    z_k = x^T \beta_k + w_k^T \gamma_k + \epsilon_k, \tag{1.8}
\]

for \( k = 1, \ldots, K \), where \( \epsilon = (\epsilon_1, \ldots, \epsilon_K) \sim N(0, \Sigma) \), then the predicted alternative is \( \hat{y} = \arg \max_j \{z_j, j = 1, \ldots, K\} \).

The representation in 1.8, however, exhibits the following identifiability issues. Adding a constant to its right hand side does not alter the induced choice alternative, that is \( P(y = k | z = z^*, x) = P(y = k | z = z^* + c, x) \). Subtracting the last latent model from the first \( K - 1 \) resolves this identifiability issue (Nobile, 1998; Geweke et al., 1994), resulting in the model

\[
    \tilde{z}_k = x^T \tilde{\beta}_k + \tilde{w}_k^T \tilde{\gamma}_k + \tilde{\epsilon}_k, \tag{1.9}
\]

where \( \tilde{z}_k = z_k - z_K \), \( \tilde{w}_k = w_k - w_K \), \( \tilde{\epsilon} = (\tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_{K-1}) \sim N(0, \tilde{\Sigma}) \). The other identifiability issue is that multiplying the right hand side of (1.9) does not alter the implied choice. This issue can be resolved by setting the first diagonal element of \( \tilde{\Sigma} \) to one (Geweke et al., 1994; McCulloch et al., 2000; Imai and van Dyk, 2005) or by restricting the trace of the latent model covariance matrix to \( \text{tr} \left( \tilde{\Sigma} \right) = K \) (Burgette and Nordheim, 2012). With these modifications, to ensure identifiability, we have

\[
    \hat{y} = \begin{cases} 
        \arg \max_j \{\tilde{z}_j, j = 1, \ldots, K - 1\}, & \text{if } \max(\tilde{z}) > 0 \\
        K, & \text{if } \max(\tilde{z}) \leq 0,
    \end{cases} \tag{1.10}
\]

where \( \max(\tilde{z}) \) is the maximum element of the vector \( \tilde{z} = (\tilde{z}_1, \ldots, \tilde{z}_{K-1}) \). For an observable data consisting of \( \{y_i, x_i, w_i; i = 1, \ldots, n\} \), the likelihood function of the
linear multinomial probit model is thus

\[ L(\tilde{\theta}, \tilde{\Sigma} \mid Y, X, W) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left[ P(y_i = k \mid x_i, w_i, \tilde{\theta}, \tilde{\Sigma}) \right]^{I(y_i = k)}, \quad (1.11) \]

where

\[ P(y_i = k \mid x_i, w_i, \tilde{\theta}, \tilde{\Sigma}) = P[\tilde{z}_{ik} > 0, \cap_{j=1, j\neq k}^{K-1} (\tilde{z}_{ik} > \tilde{z}_{ij}) \mid x_i, w_i, \tilde{\theta}, \tilde{\Sigma}], \quad (1.12) \]

for \( k = 1, \ldots, K - 1 \),

\[ P(y_i = K \mid x_i, w_i, \tilde{\theta}, \tilde{\Sigma}) = P[\cap_{j=1}^{K} (\tilde{z}_{ik} \leq 0) \mid x_i, w_i, \tilde{\theta}, \tilde{\Sigma}], \quad (1.13) \]

and \( \tilde{\theta} = \{ (\tilde{\beta}_k, \tilde{\gamma}_k) \mid k = 1, \ldots, K - 1 \} \). The estimation of the parameters \( \tilde{\theta} \) and \( \tilde{\Sigma} \) in the Bayesian framework is described in McCulloch and Rossi (1994); McCulloch et al. (2000); Imai and van Dyk (2005); Burgette and Nordheim (2012). In particular, Imai and van Dyk (2005) introduce a “working parameter” \( \alpha \) such that

\[ z_k^* = x^T \beta_k^* + \tilde{w}_k^T \gamma_k^* + \epsilon_k^*, \quad (1.14) \]

where \( z_k^* = \alpha \tilde{z}_k, \gamma_k^* = \alpha \tilde{\gamma}_k, \epsilon^* = \alpha \tilde{\epsilon}, \epsilon^* \sim N(0, \Sigma^*) \), \( \Sigma^* = \alpha^2 \tilde{\Sigma} \) and use the priors

\[ \tilde{\theta} \sim N(0, A^{-1}) \quad \text{and} \quad p(\tilde{\Sigma}) \propto |\tilde{\Sigma}|^{-(v+K)/2} \text{tr} \left( S\tilde{\Sigma}^{-1} \right)^{-v(K-1)/2} \quad (1.15) \]

with the prior on \( \tilde{\theta} \) and \( \tilde{\Sigma} \) assumed to be independent. Their posterior sampling scheme cycles through (i) updating \( (\alpha^2, Z^* \mid Y, \tilde{\theta}, \tilde{\Sigma}) \); (ii) updating \( (\alpha^2, \tilde{\theta} \mid Y, Z^*, \tilde{\Sigma}) \); and (iii) updating \( (\alpha^2, \tilde{\Sigma} \mid Y, Z^*, \Sigma^*, \tilde{\theta}) \). This sampling scheme uses the “marginal data augmentation” strategy of van Dyk (2010) in which sampling of \( \alpha \) jointly with the other parameters is conducted in every Gibbs sampling step.

Data sets with ordinal responses commonly appear in serveral disciplines. In the case of ordinal probit models, the observable data typically is \( \{(y_i, x_i) \mid i = 1, \ldots, n\} \), with the response exhibiting inherent ordering. The linear ordinal probit model assumes that, for each observation \( i \), there is an unobserved continuous random variable
\( z_i \) which depends on a linear combination of the predictors \( x_i \), such that the induced response is given by

\[
\hat{y}_i = \begin{cases} 
1, & \text{if } z_i \in (\gamma_0, \gamma_1] \\
k, & \text{if } z_i \in (\gamma_{k-1}, \gamma_k], \text{ for } k = 2, \ldots, K - 1 \\
K, & \text{if } z_i \in (\gamma_{K-1}, \gamma_K). 
\end{cases} 
\] (1.16)

The cutoff points satisfy the ordering \(-\infty = \gamma_0 < \gamma_1 < \ldots < \gamma_K = \infty\) and that \( \gamma_0 = 0 \). Inference on the parameters of the ordinal probit model in the Bayesian framework is outlined in Albert and Chib (1993); Cowles (1996); Chen and Dey (2000); Johnson and Albert (2006).

**Quantile regression**

At times data analyses seek to study the different segments of a distribution. In such cases, quantile regression may provide added insight to the least squares regression. Since the seminal work of Koenker and Bassett Jr (1978), quantile regression has been widely used in many areas of the sciences and social sciences (Meinshausen, 2006; Friederichs and Hense, 2007; Burgette et al., 2011; Fitzenberger et al., 2013). Consider the data \( \{(y_i, x_i); i = 1, \ldots, n\} \), with \( y_i \in \mathbb{R} \) representing the response and \( x_i \) the covariates of the \( i^{\text{th}} \) observation. Linear quantile regression estimates the \( \tau \text{-th} \) conditional quantile \( F^{-1}_\tau (y|x) \) using a linear combination of the covariates, where \( F^{-1}(\cdot) \) is the inverse cumulative function. The minimization

\[
\hat{\beta}_\tau = \arg \min_{\beta} \sum_{i=1}^{n} \rho_\tau \left( y_i - x_i^T \beta \right), 
\] (1.17)

where \( \rho_\tau (\omega) = \omega (\tau - I \{\omega < 0\}) \) yields the parameter estimates. The quantile regression problem can also be formulated as maximization of a likelihood obtained via the asymmetric Laplace distribution (Yu and Moyeed, 2001) which is utilized in Kozumi and Kobayashi (2011) to solve the quantile regression problem in the Bayesian framework.
Outline of the remaining chapters

The next three chapters formulate, propose, and test Bayesian ensemble of regression trees models that can be used to analyze the types of example data sets described earlier in this Chapter. Multinomial probit Bayesian additive regression trees (MP-BART) model is proposed in Chapter 2. Chapter 3 extends BART - Bayesian additive regression trees to tackle ordinal regression, while Chapter 4 proposes Bayesian quantile additive regression trees model for estimation of conditional quantiles. Concluding remarks are provided in Chapter 5.
2.1 Introduction

Multinomial probit (MNP) model for discrete choice modeling is often used in economics, market research, political sciences and transportation. It models the choices made by agents given their demographic characteristics and/or the features of the available choice alternatives. In this article, we focus on cases where there are at least three choices. Examples include the study of consumer purchasing behavior (McCulloch et al., 2000; Imai and van Dyk, 2005); voting behavior in multi-party elections (Quinn et al., 1999); and choice of different modes of transportation (Bolduc, 1999). Details of the MNP model in which choices depend on predictors in a linear fashion is studied in McFadden (1974, 1989); Keane (1992); McCulloch and Rossi (1994); Nobile (1998); McCulloch et al. (2000); Imai and van Dyk (2005); Train (2009); Burgette and Nordheim (2012) among others.

Among widely used multinomial choice modeling procedures are the multinomial logit model (McFadden, 1974; Train, 2009) and multinomial probit model (McFadden, 1989; McCulloch and Rossi, 1994; Imai and van Dyk, 2005). The former relies on an

---

Reprinted here with permission of publisher.
assumption that a choice outcome is independent of removal (or introduction) of an irrelevant choice alternative while the latter including MPBART does not make this restrictive assumption. In the multinomial probit regression framework, it is assumed that each decision maker faced with \( K \geq 3 \) alternatives uses a \((K - 1)\) vector of latent variables in order to arrive at their choice. Alternative \( k \) is chosen if the \( k^{th} \) entry of the latent vector is positive and greater than the other entries, for \( k = 1, \ldots, (K - 1) \).

If none of the entries of the latent vector are positive, then the “reference” alternative \( K \) is chosen.

MPBART can also be used as a multiclass classification procedure to classify units into one of \( K \geq 3 \) classes based on their observed characteristics. Multiclass classification is common in many disciplines. In biology, tumors are classified into tumor sub-types based on their gene expression profiles (Khan et al., 2001). In environmental sciences, clouds are classified as clear, liquid clouds, or ice clouds based on their radiance profiles (Lee et al., 2004). Other areas of multiclass classification application include text recognition, spectral imaging, chemistry, and forensic science (Li et al., 2004; Fauvel et al., 2006; Evett and Spiehler, 1987; Vergara et al., 2012).

The effect of predictors on the response may be linear or non-linear, of much or little significance, and at times magnified with interactions. When such complicated relationships exist, models that use ensemble of trees often provide appealing framework since variable selection and inclusion of interactions are intrinsic in construction of trees. Some popular “tree-based” classification methods include CART (Breiman et al., 1984; Quinlan, 1986), Bayesian CART (Chipman et al., 1998), random forests (Breiman, 2001), and gradient boosting (Friedman, 2001). There is a gap in the literature for “tree based” statistical procedures that directly deal with the MNP model in which choice specific predictors can readily be incorporated. This article, thus, seeks to fill that void using Bayesian tree ensembles for multinomial probit regression.

A newcomer to the “tree-based” family is the Bayesian additive regression trees
(BART) (Chipman et al., 2010). The innovative idea of BART is to approximate an unknown function $f(x)$ for predicting a continuous variable $z$ given values of input $x$ using a sum-of-trees model

$$f(x) \approx \sum_{j=1}^{n_T} g(x, T_j, M_j),$$

where $g(x, T_j, M_j)$ is the $j^{th}$ tree that consists of sets of partition rules $T_j$ and parameters $M_j$ associated with its terminal nodes. Conceptually, the sum-of-trees structure makes BART adaptive to complicated nonlinear and interaction effects, and the use of Bayesian regularization prior on regression trees minimizes the risk of over-fitting. Empirically, a variety of experiments and applications of BART has confirmed that it has robust and accurate out-of-sample prediction performance (Liu and Zhou, 2007; Chipman et al., 2010; Abu-Nimeh et al., 2007; Bonato et al., 2011). The standard BART further extends to binary classification problems and shows competitive classification performance (Zhang and Härdle, 2010; Chipman et al., 2010).

The success of BART on predicting continuous and binary variables naturally motivates the question of whether the sum-of-trees structure also helps in predicting multinomial choices and classes, thus, we are interested in the utility of the sum-of-trees for discrete choice modeling. We utilize a Bayesian probit model formulation in Albert and Chib (1993); McCulloch and Rossi (1994); McCulloch et al. (2000); Imai and van Dyk (2005) in conjunction with the idea of sum-of-trees regression to propose multinomial probit Bayesian additive regression trees (MPBART). Through a comprehensive simulation study with various data generating schemes, we find that it is a serious contender in its predictive performance to existing multinomial choice models and multiclass classification methods and that it usually ranks among the topmost when a nonlinear relationship exists between the predictors and choice alternatives.

A related work to this article is Agarwal et al. (2014), which utilizes BART for the purpose of satellite image classification. Their multiclass classification procedure
combines binary BART and one-versus-all technique of transforming a multiclass problem to a series of binary classification problems. Our work is different from theirs in that we consider the problem within the traditional multinomial probit regression framework rather than the one-versus-all framework.

The chapter proceeds as follows. Section 2.2 formally outlines the multinomial probit model in general and MPBART in particular along with the associated data structure, Section 2.3 delves into the prior specifications and posterior computation for MPBART. Sections 2.4 and 2.5 use simulated data sets and real data examples, respectively to illustrate the predictive performance of MPBART. Section 2.6 closes the article with concluding remarks.

2.2 MPBART: Multinomial probit Bayesian additive regression trees

Suppose we have a data set \((y_i, X_i)\) for \(i = 1, \ldots, n\), where \(y_i \in \{1, \ldots, K\}\) denotes the available choice alternatives and \(X_i\) the predictors for the \(i^{th}\) observation. We are interested in estimating the conditional choice probability \(p(y_i = k \mid X_i)\) for \(k = 1, \ldots, K\). The observed choice \(y_i\) can be viewed as arising from a vector of latent variables \(z_i \in \mathbb{R}^{K-1}\) as in Albert and Chib (1993); Geweke et al. (1994); Imai and van Dyk (2005) via

\[
y_i(z_i) = \begin{cases} 
  k & \text{if } \max(z_i) = z_{ik} > 0, \\
  K & \text{if } \max(z_i) < 0,
\end{cases} \tag{2.1}
\]

for \(k = 1, \ldots, (K - 1)\), where \(\max(z_i)\) denotes the largest element of \(z_i = (z_{i1}, \ldots, z_{i,K-1})'\). The latent vector \(z_i\) depends on \(X_i\) as follows:

\[
z_i = G(X_i; T, M) + \epsilon_i \quad \text{for } i = 1, \ldots, n, \tag{2.2}
\]

where \(G(X_i; T, M) = (G_1(X_i; T, M), \ldots, G_{K-1}(X_i; T, M))'\) is a vector of \(K - 1\) regression functions and \(\epsilon_i = (\epsilon_{i1}, \ldots, \epsilon_{i,K-1})' \sim N(0, \Sigma)\).
The predictors for the \( i \)th observation are comprised of two components \( \mathbf{v}_i \) and \( \mathbf{W}_i \) (i.e., \( \mathbf{X}_i = (\mathbf{v}_i, \mathbf{W}_i) \)). The first component is a vector of \( q \) demographic variables \( \mathbf{v}_i \in \mathbb{R}^q \) that describe the subject. The second component \( \mathbf{W}_i = (\mathbf{w}_{i1}, \ldots, \mathbf{w}_{i(K-1)}) \), where \( \mathbf{w}_{ik} \in \mathbb{R}^r \), is a matrix of \( r \) predictors that vary along the choice alternatives in relation to the reference choice. For example, in a market research scenario, the price of the choices faced by individuals in a study is a choice specific predictor that varies along alternatives and the difference between the prices of \( k \)th choice and the reference choice \( K \) will be part of \( \mathbf{w}_{ik} \), for \( k = 1, \ldots, (K-1) \).

The tree splitting rules of the \( k \)th sum of trees
\[
G_k(\mathbf{X}_i; \mathbf{T}, \mathbf{M}) = \sum_{j=1}^{n_T} g(\mathbf{X}_i, T_{kj}, M_{kj}), \quad \text{for } k = 1, \ldots, (K-1) \quad (2.3)
\]
depend on \( \mathbf{X}_i \) through \( \mathbf{x}_{ik} = (\mathbf{v}_i, \mathbf{w}_{ik}) \). The \( j \)th tree of the \( k \)th sum of trees, \( g(\cdot, T_{kj}, M_{kj}) \), consists of \( T_{kj} \), a set of partition rules based on the predictor space, and \( M_{kj} = \{\mu_{kj,l}, l = 1, \ldots, b_{kj}\} \), a set of parameters associated with the terminal nodes. The partition rules \( T_{kj} \) are recursive binary splits of the form \( \{x < s\} \) versus \( \{x \geq s\} \), where \( x \) is one of the predictors that make up \( \mathbf{x}_{ik} \), and \( s \) is a value in the range of \( x \).

The complete set of parameters of MPBART (2.1)–(2.3) is thus
\[
\{(T_{kj}, M_{kj})_{k=1,\ldots,(K-1),j=1,\ldots,n_T}, \Sigma\},
\]
where \( M_{kj} \) denotes the collection of terminal nodes of the \( j \)th tree in the \( k \)th sum-of-trees.

2.3 Prior specifications and posterior computation

Prior specifications

The \( \Sigma \) prior:

The MNP model specification in (2.2) exhibits a well documented identifiability issue, for example the multiplication of both sides of (2.2) by a positive constant does
not alter the implied choice outcome (Keane, 1992; McCulloch and Rossi, 1994; McCulloch et al., 2000; Nobile, 1998). To circumvent this issue, McCulloch and Rossi (1994); McCulloch et al. (2000); Imai and van Dyk (2005) among others restrict the first diagonal element of $\Sigma$ to equal one, while Burgette and Nordheim (2012) restricts the trace of $\Sigma$ to equal $K$. We implement the latter.

Consider an augmented latent model

$$\tilde{z}_i = G(X_i; T, \tilde{M}) + \tilde{\epsilon}_i,$$  \hspace{1cm} (2.4)

where $\tilde{z}_i = \alpha z_i$, $\tilde{\epsilon}_i = \alpha \epsilon_i$, $\tilde{\epsilon}_i \sim N(0, \tilde{\Sigma})$, $\tilde{\Sigma} = \alpha^2 \Sigma$ and $\tilde{M}_{kj} = \{\alpha \mu_{kj}, l = 1, \ldots, b_{kj}\}$. Following Imai and van Dyk (2005); Burgette and Nordheim (2012), we place the prior $p(\Sigma) = \int p(\Sigma, \alpha^2) p(\alpha^2 | \Sigma) \, d\alpha^2 \propto |\Sigma|^{-(v+K)/2} \left( \text{tr} \left[ S\Sigma^{-1} \right] \right)^{-\nu(K-1)/2}$, with a restriction $\text{tr}(\Sigma) = K$; a constrained inverse Wishart distribution induced by

$$\Sigma \sim \text{Inv-Wish} \left( \nu, \alpha_0^2 S \right) \text{ and } \alpha^2 | \Sigma \sim \alpha_0^2 \text{tr}[S\Sigma^{-1}] / \chi^2_v(K).$$

**The $T_{kj}$ prior:**

As in Chipman et al. (1998) and Chipman et al. (2010), the prior on a single tree $T_{kj}$ is specified through a “tree-generating stochastic process” apriori independent of $\Sigma$. The tree prior consists of (i) the probability of splitting a terminal node, (ii) the distribution of the splitting variable if the node has to split, and (iii) the distribution of the splitting rule given the splitting variable. For step (i), the probability that a terminal node $\eta$ splits is given by

$$\frac{\gamma}{(1 + d_{\eta})^\beta}, \quad \gamma \in (0, 1), \beta \in [0, \infty),$$

where $d_{\eta}$ is the depth of the node. A small $\gamma$ and a big $\beta$ result in a tree with small number of terminal nodes. In other words, influence of individual trees in the
sum can be controlled by carefully choosing $\gamma$ and $\beta$. For step (ii), the splitting variable is uniformly selected from all possible predictors, representing a prior belief of equal level of importance placed on each predictor. For step (iii), given a splitting predictor, the splitting value $s$ is taken to be a random sample from discrete uniform distribution of the set of observed values of the selected predictor, provided that such a value does not result in an empty partition.

**The $\mu_{kjl}|T_{kj}$ prior:**

Given a tree $T_{kj}$ with $b_{kj}$ terminal nodes, the prior distribution on the terminal node parameters is taken to be

$$
\mu_{kjl} \mid T_{kj} \sim \text{iid} N(\mu_k, \tau_k^2) \text{ for } k = 1, \ldots, (K - 1).
$$

For binary classification problems (i.e., $K = 2$), Chipman et al. (2010) propose choosing $\mu_1 = 0$ and $\tau_1 = 3/(r \sqrt{n_T})$ so that the sum-of-tree effect $\sum_{j=1}^{n_T} g(x, T_{1j}, \mu_{1j})$ assigns high probability to the interval $(-3, 3)$. We extend their method to the multinomial probit setting by assuming $\mu_k = 0$ and $\tau_k = 3/(r \sqrt{n_T})$ for all $k$. The hyper-parameters $r$ and $n_T$ play the role of adjusting the level of shrinkage on the contribution of each individual tree. Default values $r = 2$ and $n_T = 200$ are recommended by Chipman et al. (2010) which we also find reasonable in the multinomial probit setup.

**Posterior computation**

Our posterior sampling scheme relies on the partial marginal data augmentation strategy van Dyk (2010). Marginal data augmentation (MDA) and partial marginal data augmentation (Meng and van Dyk, 1999; Imai and van Dyk, 2005; van Dyk, 2010; Burgette and Nordheim, 2012) introduce a “working parameter” that is identifiable given an augmented data, but not identifiable given the observed data. By strate-
gically augmenting the data, MDA and partial MDA result in a computationally tractable posterior distribution and an MCMC chain with improved convergence.

Our posterior computing is accomplished via cycling through the following three steps (for convenience the intermediate draws are flagged with an asterisk).

(i) Sample from \((z, \alpha^2) \mid T, M, \Sigma, y\) by obtaining random draws of \(p\{(z_i)_{i=1,\ldots,n} \mid T, M, \Sigma, y\}\), and \((\alpha^*)^2 \sim p\{\alpha^2 \mid \Sigma, M, T, (z_i)_{i=1,\ldots,n}\} = p\{\alpha^2 \mid \Sigma\}\) followed by transforming to obtain \(\tilde{z}_i^* = \alpha^*z_i\) for all \(i\).

(ii) Sample from \((T, \tilde{M}^*) \sim p\{(T, \tilde{M}^*) \mid (\tilde{z}_i^*)_{i=1,\ldots,n}, \Sigma, (\alpha^*)^2, y\}\) followed by recording \(M = \tilde{M}^*/\alpha^*\).

(iii) Sample from \((\Sigma, \alpha^2) \sim p\{(\Sigma, \alpha^2) \mid T, \tilde{M}^*, (\tilde{z}_i^*)_{i=1,\ldots,n}, y\}\) by random draws of \(p\{\tilde{\Sigma}^* \mid T, \tilde{M}^*, (\tilde{z}_i^*)_{i=1,\ldots,n}, y\}\) followed by transforming \(\tilde{\Sigma}^*\) to \((\Sigma, \alpha^2)\).

Our algorithm utilizes a “partial marginalization” strategy since the working parameter \(\alpha^2\) is updated in steps (i) and (iii), but not in (ii) (cf. the marginalization strategy in Imai and van Dyk (2005) where the working parameter is updated in every step).

The first part of obtaining a sample from (i) is iterative random draws of truncated normals from the conditional distribution \(z_{ik} \mid z_{i(-k)}, T, M, \Sigma \sim N(m_{ik}, \psi_k)\) with \(\max\{0, \max(z_{i(-k)})\}\) as a lower truncation point if \(y_i = k\) and as an upper truncation point if \(y_i \neq k\), where \(z_{i(-k)}\) represents the vector \(z_i\) without the \(k^{th}\) entry. The conditional first moment and variance \(m_{ik}\), and \(\psi_k\) are given by

\[
m_{ik} = G_k(X_i; T, M) + \sigma_{k(-k)}\Sigma_{(-k)(-k)}^{-1}[z_{i(-k)} - G_k(-k)(X_i; T, M)],
\]

\[
\psi_k = \sigma_{kk} - \sigma_{k(-k)}\Sigma_{(-k)(-k)}^{-1}\sigma_{k(-k)},
\]

where \(\sigma_{k(-k)}\) is the \(k^{th}\) column of \(\Sigma\) that excludes \(\sigma_{kk}\) and \(\Sigma_{(-k)(-k)}\) is the matrix \(\Sigma\) that excludes the \(k^{th}\) column and row.
For (ii), we sample \( (T_{kj}, \tilde{M}_{kj}^*) \) for \( k = 1, \ldots, (K-1) \), \( j = 1, \ldots, n_T \) via the following. Given all the trees and their terminal node parameters but the \( j \)th tree in the \( k \)th sum of trees, \( \tilde{\Sigma}, \tilde{z}_{i(-k)}^* \) and \((\alpha^*)^2\), we observe that

\[
\tilde{z}_{ik}^+ = g(X_i, T_{kj}, \tilde{M}_{kj}) + \tilde{e}_{ik}^+, \quad \tilde{e}_{ik}^+ \sim N(0, \tilde{\psi}_k), \tag{2.6}
\]

where

\[
\tilde{z}_{ik}^+ = \tilde{z}_{ik}^* - \sum_{l \neq j} g(X_i, T_{kl}, \tilde{M}_{kl}) - \tilde{\sigma}_{k(-k)} \tilde{\Sigma}_{(-k)(-k)}^{-1} [\tilde{z}_{i(-k)}^* - G_{(-k)}(X_i; T, \tilde{M})]
\]

and \( \tilde{\psi}_k = (\alpha^*)^2 \tilde{\psi}_k \). We use the back-fitting algorithm, also used in Chipman et al. (2010), to obtain posterior samples of \( (T_{kj}, \tilde{M}_{kj}^*) \) by considering (2.6) as the single tree model of Chipman et al. (1998). Finally, the posterior sample in (iii) is

\[
\tilde{\Sigma}^* \sim \text{Inv-Wish} \left( \nu + n, \hat{S} + \sum_{i=1}^n [\tilde{z}_i^* - G(X_i; T, \tilde{M}^*)] [\tilde{z}_i^* - G(X_i; T, \tilde{M}^*)]' \right)
\]

then taking \( \alpha^2 \) as \( \text{tr}(\tilde{\Sigma}^*)/K \) and transforming to obtain \( \Sigma = \tilde{\Sigma}^*/\alpha^2 \).

**Posterior-based prediction**

In our Bayesian setting, predictions of future observations \( y^* \) at new values \( X^* \) are based upon the posterior predictive distribution \( p(y^* \mid y) = \int p(y^* \mid X, \Theta, y)p(\Theta \mid y)dy \), where \( \Theta \) consists of all unknown parameters of MPBART. For a given loss function, predictions of \( y^* \) are made using the optimal choice \( a \in \{1, \ldots, K\} \) that minimizes the expected posterior predictive loss

\[
E_{y^* \mid y} L(y^*, a) = \int L(y^*, a) p(y^* \mid y)dy^*,
\]

where \( L(y^*, a) \) is the loss function of using class \( a \) to predict the unknown choice outcome \( y^* \). We assume that the loss function \( L(y, a) \) assigns a pre-specified non-negative loss to every combination of action \( a \in \{1, \ldots, K\} \) and true choice \( y \in \{1, \ldots, K\} \). These pre-specified loss combinations are described in Table 1.1 and can
equivalently be expressed as

\[ L(y, a) = \sum_{l=1}^{K} \sum_{m=1}^{K} C_{lm} I(y = l, a = m), \tag{2.7} \]

where \( I(\cdot) \) is the usual indicator function.

Under the loss function (2.7), the expected posterior predictive loss is given by

\[ E_{y^* | Y} L(y^*, a) = \sum_{l=1}^{K} C_{la} p(y^* = l \mid y). \tag{2.8} \]

We assume that the costs associated with a wrong prediction are all equal to the constant \( C \) and correct prediction costs equal to 0 (i.e., \( C_{lm} = C > 0 \) for \( l \neq m \), and \( C_{ll} = 0 \)). Then the expected posterior predictive loss (2.8) simplifies to

\[ E_{y^* | Y} L(y^*, a) = C \{ 1 - p(y^* = a \mid y) \}, \]

which is minimized at

\[ a = \arg \max_k \{ p(y^* = k \mid y), k = 1, \ldots, K \}. \tag{2.9} \]

The posterior predictive distribution \( p(y^* = l \mid y) \) does not have closed form representation and is thus approximated using Monte Carlo samples drawn from the posterior distributions \( p(\Theta \mid y) \). Once computed, they enable the estimation of the predictions 2.9 through a search over the space \( a \in \{1, \ldots, K\} \).

### 2.4 Synthetic data examples

**A simulation study in the multinomial choice framework**

In this three choice simulation study, we use a function similar to the one used in Friedman (1991) to induce a non-linear relationship between five choice specific predictors \( w_k \in \mathbb{R}^5, k = 1, 2, 3 \) and the choice alternatives. The choice specific predictors are from i.i.d Unif[0, 1]. In addition, we include a predictor \( v \sim \text{Unif}[0, 2] \) that describes the observed unit. Suppose that

\[ f(u) = 20 \sin(\pi u_1 u_2) - 20(u_3 - 0.5)^2 + 10u_4 + 5u_5, \]

\[ g(v) = 8v, \]

and

\[
\begin{bmatrix}
  z_1 \\
  z_2
\end{bmatrix} = \begin{bmatrix}
  f(w_1 - w_3) + g(v) \\
  f(w_2 - w_3) + g(v)
\end{bmatrix} + \epsilon, \quad \epsilon \sim \mathcal{N} \left( 0, \begin{bmatrix}
  1 & 0.5 \\
  0.5 & 1
\end{bmatrix} \right). \tag{2.10}
\]
The response variable is then recorded using

\[ y(z) = \begin{cases} 
  k & \text{if } \max(z) = z_k > 0, \\
  3 & \text{if } \max(z) < 0,
\end{cases} \quad \text{for } k = 1, 2. \]

This true model contains linear, nonlinear, and interaction effects, making it interesting benchmark data set. We are mainly interested in how well MPBART is able to predict the choices on a test data. Hence, we simulate training and test data sets of 500 observations each and compare the predictive performance on the test data for MPBART, Bayesian multinomial probit model (Bayes-MNP) in Imai and van Dyk (2005), the Multinomial logit (MNL) model in Train (2009); McFadden (1974), and the following multiclass classification procedures: support vector machines with linear (SVM-L) and radial (SVM-R) kernels in Cortes and Vapnik (1995); Vapnik (2013), random forest (RF) in Breiman (2001), linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) in Duda et al. (2012); Friedman et al. (2001), multinomial logistic regression (MNL) in McFadden (1974), classification and regression trees (CART) in Breiman et al. (1984); Quinlan (1986), neural networks (NNET) in Ripley (2007), K-nearest neighbors (KNN) in Cover and Hart (1967) and One vs. All BART (OvA-BART) in Agarwal et al. (2014). We note that for the multiclass classification procedures, a choice specific predictor makes up three separate predictors in this simulation study, one describing each of the choices, putting the total number of predictors for this simulation study at sixteen. For each competing procedure and MPBART, we selected the tuning parameters via a 10-fold cross-validation based on the training data. Table 2.1 lists the names of these competing procedures, the corresponding \textit{R} packages utilized and tuning parameters.

The comparison metric we use in this example and all that follow is test error rate

\[ \frac{1}{m} \sum_{i=1}^{m} I(\hat{y}_i \neq y_i), \tag{2.11} \]

where \( y_i \) and \( \hat{y}_i \) are the actual and predicted classes for the \( i \)th observation in a given test data set of size \( m \). This metric makes use of the loss function in (2.7) with a
Table 2.1  List of competing classifiers, the \( R \) packages utilized, and tuning parameters that are chosen by cross-validation. The abbreviations in the first column stand for the procedures mentioned in the second paragraph of this chapter.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>( R ) Package</th>
<th>Tuning parameter(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>randomForest</td>
<td>mtry</td>
</tr>
<tr>
<td>CART</td>
<td>rpart</td>
<td>no tuning parameters</td>
</tr>
<tr>
<td>SVM-L</td>
<td>kernlab</td>
<td>( C )</td>
</tr>
<tr>
<td>SVM-R</td>
<td>kernlab</td>
<td>( C ) and ( \sigma )</td>
</tr>
<tr>
<td>QDA</td>
<td>MASS</td>
<td>no tuning parameters</td>
</tr>
<tr>
<td>LDA</td>
<td>MASS</td>
<td>no tuning parameters</td>
</tr>
<tr>
<td>NNET</td>
<td>nnet</td>
<td>size and decay</td>
</tr>
<tr>
<td>MNL</td>
<td>mlogit</td>
<td>no tuning parameters</td>
</tr>
<tr>
<td>KNN</td>
<td>caret</td>
<td>( k )</td>
</tr>
<tr>
<td>OvA-BART</td>
<td>dbarts</td>
<td>( k ), power, base</td>
</tr>
</tbody>
</table>

misclassification cost of \( C_{lm} = 1 \) and a cost of \( C_{ll} = 0 \) for a correct prediction. As can be seen from Table 2.2, MPBART exhibits a very good out-of-sample predictive accuracy. This is not surprising given the data generating scheme with nonlinear effects.

**A simulation study for multiclass classification**

In this simulation study the waveform recognition problem in Breiman et al. (1984), often used as a benchmark artificial data in multiclass classification studies (Gama et al., 2003; Hastie and Tibshirani, 1996; Keerthi et al., 2005) is employed. The model has 21 predictors and a multiclass response with 3 classes. For each observation, the \( i^{th} \) predictor \( x_i \) is generated from

\[
 x_i = \begin{cases} 
 u h_1(i) + (1 - u)h_2(i) + \epsilon_i, & \text{if } y = 1, \\
 u h_1(i) + (1 - u)h_3(i) + \epsilon_i, & \text{if } y = 2, \\
 u h_2(i) + (1 - u)h_3(i) + \epsilon_i, & \text{if } y = 3,
\end{cases}
\]  

(2.12)

where \( i = 1, \ldots, 21, u \sim \text{UNIF}[0, 1], \epsilon_i \sim \text{N}(0, 1) \), and \( h_i \) are three waveform functions: \( h_1(i) = \max(6 - |i - 11|, 0) \), \( h_2(i) = h_1(i - 4) \), and \( h_3(i) = h_1(i + 4) \).
Table 2.2  Comparison of MPBART and the procedures listed in Table 2.1 on the first simulation study generated via (2.10) and the waveform recognition example (2.12). Training and test data sets of each 500 observations are used for the first simulation study. Training and test data sets of 300 and 500 observations, respectively are used for the waveform recognition example. Average test error rates (with standard errors in parentheses) are reported on 20 replications.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Simulation Study - I</th>
<th>Waveform Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test Error Rate</td>
<td>Rank</td>
</tr>
<tr>
<td>MPBART</td>
<td>0.2725 (0.0060)</td>
<td>1</td>
</tr>
<tr>
<td>Bayes-MNP</td>
<td>0.3976 (0.0065)</td>
<td>7</td>
</tr>
<tr>
<td>MNL</td>
<td>0.3921 (0.0064)</td>
<td>6</td>
</tr>
<tr>
<td>RF</td>
<td>0.4023 (0.0059)</td>
<td>8</td>
</tr>
<tr>
<td>CART</td>
<td>0.4791 (0.0080)</td>
<td>12</td>
</tr>
<tr>
<td>SVM-L</td>
<td>0.4072 (0.0058)</td>
<td>9</td>
</tr>
<tr>
<td>SVM-R</td>
<td>0.3254 (0.0057)</td>
<td>3</td>
</tr>
<tr>
<td>LDA</td>
<td>0.4095 (0.0064)</td>
<td>10</td>
</tr>
<tr>
<td>QDA</td>
<td>0.3381 (0.0045)</td>
<td>4</td>
</tr>
<tr>
<td>NNET</td>
<td>0.2917 (0.0065)</td>
<td>2</td>
</tr>
<tr>
<td>KNN</td>
<td>0.4195 (0.0070)</td>
<td>11</td>
</tr>
<tr>
<td>OvA-BART</td>
<td>0.3908 (0.0059)</td>
<td>5</td>
</tr>
</tbody>
</table>

We generate 20 replications of training and test data sets with 300 and 500 observations, respectively from (2.12) and compare MPBART with classifiers listed in Table 2.1. Our choice of sample sizes is the same as those in Hastie and Tibshirani (1996) so the results can be compared with them. Table 2.2 summarizes the average error rates and standard errors in parentheses based on 20 simulations. For LDA, QDA and CART, the error rates are consistent with those reported in Table 1 of Hastie and Tibshirani (1996). MPBART is among best for this data generating scheme exhibiting low average test error rate. Note that Hastie and Tibshirani (1996) report an error rate of 0.157 on test data sets achieved by penalized mixture discriminant analysis.
2.5 Real data examples

Multinomial choice data examples

Two discrete choice data sets, dealing with fishing and travel mode choices, are used to illustrate MPBART. Fishing mode choice data is a survey of 1,182 individuals who reported their most recent saltwater fishing modes as either “beach”, “pier”, “boat” or “charter”. The choice specific variables in this data set are expected catch rates per hour and price for each mode of fishing, while the individual specific predictor is monthly income. Details of this data are in Kling and Thomson (1996); Herriges and Kling (1999) and we use the version of data available in the R package mlogit. The second data records the choice of travel mode between Sydney and Melbourne, Australia as either “air”, “train”, “bus” or “car” (Greene, 2003; Kleiber and Zeileis, 2008). It includes 210 individuals’ choice of travel and the following choice specific predictors: general cost associated with the travel mode choice, waiting time at a terminal (with zero recorded for a travel choice of “car”), cost of travel mode and travel time. In addition, the individual specific predictors logarithms of household income, and traveling party size are used. We use the version of the data set in the R package AER (Kleiber and Zeileis, 2008).

After splitting the fishing mode data into ten and the travel mode data into five nearly equal random folds, we implement the procedures MPBART, Bayesian multinomial probit model (Bayes-MNP), the Multinomial logit (MNL) and the multiclass classification procedures listed in Table 2.1 with one fold set aside as a test data and the remaining folds utilized for training the models. Table 2.3 reports the average test error rates along with their standard errors. MPBART is again among the procedures with the lowest error rates.
Table 2.3 Comparison results on the fishing mode and choice of travel mode data sets. Classification error rates (with standard errors in parentheses) are reported.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Fishing Mode</th>
<th></th>
<th>Travel Mode</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test Error Rate</td>
<td>Rank</td>
<td>Test Error Rate</td>
<td>Rank</td>
</tr>
<tr>
<td>MPBART</td>
<td>0.3960 (0.0160)</td>
<td>1</td>
<td>0.0571 (0.0086)</td>
<td>2</td>
</tr>
<tr>
<td>Bayes-MNP</td>
<td>0.5546 (0.0171)</td>
<td>10</td>
<td>0.3286 (0.0394)</td>
<td>10</td>
</tr>
<tr>
<td>MNL</td>
<td>0.5600 (0.0160)</td>
<td>11</td>
<td>0.3143 (0.0332)</td>
<td>9</td>
</tr>
<tr>
<td>RF</td>
<td>0.4746 (0.0148)</td>
<td>3</td>
<td>0.0429 (0.0089)</td>
<td>1</td>
</tr>
<tr>
<td>CART</td>
<td>0.5372 (0.0147)</td>
<td>8</td>
<td>0.1048 (0.0161)</td>
<td>3</td>
</tr>
<tr>
<td>SVML</td>
<td>0.5034 (0.0139)</td>
<td>6</td>
<td>0.2143 (0.0345)</td>
<td>7</td>
</tr>
<tr>
<td>SVMR</td>
<td>0.4882 (0.0194)</td>
<td>4</td>
<td>0.1381 (0.0254)</td>
<td>5</td>
</tr>
<tr>
<td>LDA</td>
<td>0.4975 (0.0193)</td>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>NNET</td>
<td>0.5211 (0.0064)</td>
<td>7</td>
<td>0.3048 (0.0739)</td>
<td>8</td>
</tr>
<tr>
<td>KNN</td>
<td>0.5406 (0.0189)</td>
<td>9</td>
<td>0.1810 (0.0358)</td>
<td>6</td>
</tr>
<tr>
<td>OvA-BART</td>
<td>0.4434 (0.0144)</td>
<td>2</td>
<td>0.1143 (0.0158)</td>
<td>4</td>
</tr>
</tbody>
</table>

Multiclass classification data examples

Forensic glass and vertebral column classification data sets, both of which are publicly available at the University of California at Irvine (UCI) machine learning data repository (Bache and Lichman, 2013), are used to illustrate MPBART as a multiclass classification procedure in comparison to the multiclass classification procedures listed in Table 2.1. The forensic glass classification data set consists of 9 features collected on 214 glass samples classified as one of the 6 glass types: building windows float processed, building windows non-float processed, vehicle windows float processed, containers, tableware, or headlamps. The vertebral column data contains 310 patients diagnosed either as normal, having Disk Hernia or Spondylolisthesis. This data set records the pathology of the human vertebral column, whose main function is the protection of the spine, and its dependence on the characteristics of the pelvis and lumbar spine. Further detail on the data set is available in da Rocha Neto et al. (2011); Calle-Alonso et al. (2013).

In our analysis, we split the forensic glass and vertebral column data sets into five and ten nearly equal random folds, respectively. One fold is set aside as test data and
Table 2.4  Classification error rates and standard errors (in parentheses) for vertebral column and forensic glass data sets.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Vertebral Column</th>
<th>Forensic Glass</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Test Error Rate</td>
<td>Rank</td>
</tr>
<tr>
<td>MPBART</td>
<td>0.1466 (0.0324)</td>
<td>1</td>
</tr>
<tr>
<td>RF</td>
<td>0.1645 (0.0265)</td>
<td>4</td>
</tr>
<tr>
<td>CART</td>
<td>0.1839 (0.0160)</td>
<td>8</td>
</tr>
<tr>
<td>SVML</td>
<td>0.1484 (0.0285)</td>
<td>2</td>
</tr>
<tr>
<td>SVMR</td>
<td>0.1742 (0.0216)</td>
<td>6</td>
</tr>
<tr>
<td>LDA</td>
<td>0.1968 (0.0335)</td>
<td>0</td>
</tr>
<tr>
<td>QDA</td>
<td>0.1548 (0.0254)</td>
<td>3</td>
</tr>
<tr>
<td>NNET</td>
<td>0.2161 (0.0259)</td>
<td>10</td>
</tr>
<tr>
<td>MNL</td>
<td>0.6129 (0.0304)</td>
<td>11</td>
</tr>
<tr>
<td>KNN</td>
<td>0.1806 (0.0334)</td>
<td>7</td>
</tr>
<tr>
<td>OvA-BART</td>
<td>0.1645 (0.0282)</td>
<td>5</td>
</tr>
</tbody>
</table>

the classification methods in Table 2.1 and MPBART are trained on the remaining folds. Table 2.4 shows the average classification error rates with standard errors in parenthesis. QDA could not be implemented in this data set since the representation of observations classified as tableware is very small. For the same reason, we only considered five-fold partitioning of the forensic glass data. MPBART, RF and OvA-BART are the top performing procedures in terms of having the lowest classification error.

2.6 Conclusion

We have proposed and tested through simulation studies and real data examples the utility of Bayesian ensemble of trees for multinomial probit regression and multiclass classification. Regression trees and their ensembles are widely used for the purpose of classification. However, their use in multinomial probit regression which allows the introduction of choice specific predictors is less explored. MPBART fills that gap in the literature. It exhibits very good predictive performance in a range of examples and is among the best when the relationship between the predictors and choice response
Table 2.5  Travel mode data selected tuning parameters. The number of posterior samples is 2000 with the first 500 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>β</th>
<th>γ</th>
<th>n₀</th>
<th>base choice</th>
<th>s</th>
<th>pb</th>
<th>pbd</th>
<th>r</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.99</td>
<td>5</td>
<td>car</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.0476</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.95</td>
<td>2</td>
<td>bus</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.0952</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.95</td>
<td>2</td>
<td>train</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.0238</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.99</td>
<td>2</td>
<td>bus</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.0476</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.95</td>
<td>5</td>
<td>car</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.0714</td>
</tr>
</tbody>
</table>

Table 2.6  Fishing mode data selected tuning parameters. The number of posterior samples is 2000 with the first 500 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>β</th>
<th>γ</th>
<th>n₀</th>
<th>base choice</th>
<th>s</th>
<th>pb</th>
<th>pbd</th>
<th>kfac</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.99</td>
<td>10</td>
<td>charter</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.3782</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.95</td>
<td>5</td>
<td>charter</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.3361</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.95</td>
<td>10</td>
<td>charter</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.3983</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.95</td>
<td>20</td>
<td>charter</td>
<td>5</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.4153</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0.99</td>
<td>10</td>
<td>boat</td>
<td>5</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.5000</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0.95</td>
<td>5</td>
<td>boat</td>
<td>3</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.3814</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.99</td>
<td>5</td>
<td>boat</td>
<td>4</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.3475</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0.99</td>
<td>2</td>
<td>boat</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.3559</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.95</td>
<td>2</td>
<td>beach</td>
<td>5</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.3898</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0.99</td>
<td>2</td>
<td>charter</td>
<td>3</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.4576</td>
</tr>
</tbody>
</table>

is nonlinear. The software implementation of MPBART is freely available as an R package `mpbart`.

**SELECTED TUNING PARAMETERS**

For the simulation studies and real data examples, the MPBART tuning parameters selected via cross-validation are in this section. The selected scale matrix of the prior on $\Sigma$ is $S = sI_{K-1}$, where $I_{K-1}$ an identity matrix of size $K - 1$. The column $n₀$ indicates the minimum number of observations allowed in a terminal node.
Table 2.7  Waveform recognition dataset selected tuning parameters. The number of posterior samples is 1300 with the first 300 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>β</th>
<th>γ</th>
<th>n₀</th>
<th>base choice</th>
<th>s</th>
<th>pb</th>
<th>pbd</th>
<th>kfac</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0.99</td>
<td>30</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1620</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.99</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1460</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.99</td>
<td>15</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1920</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.99</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1740</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.99</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1740</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0.99</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1680</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>0.99</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1420</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>0.99</td>
<td>15</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1460</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>0.99</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1480</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0.99</td>
<td>20</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1460</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>0.99</td>
<td>15</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.2000</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>0.9</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1780</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>0.99</td>
<td>20</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1520</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>0.99</td>
<td>15</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1400</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>0.9</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1420</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>0.99</td>
<td>30</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1940</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>0.99</td>
<td>20</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1360</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>0.99</td>
<td>30</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1220</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>0.99</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1520</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>0.99</td>
<td>20</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>2</td>
<td>0.1640</td>
</tr>
</tbody>
</table>
Table 2.8  Friedman multiclass classification data set selected tuning parameters. The number of posterior samples is 1000 with the first 500 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>β</th>
<th>γ</th>
<th>n₀</th>
<th>s</th>
<th>pb</th>
<th>pbd</th>
<th>r</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.99</td>
<td>5</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.342</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.99</td>
<td>20</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.342</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.99</td>
<td>10</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.32</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.99</td>
<td>20</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.364</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.99</td>
<td>15</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.366</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.95</td>
<td>30</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.382</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.95</td>
<td>30</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.338</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.99</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.342</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.99</td>
<td>5</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.342</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.99</td>
<td>5</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.294</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.99</td>
<td>5</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.38</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.99</td>
<td>10</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.346</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0.95</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.35</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0.95</td>
<td>5</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.392</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0.99</td>
<td>15</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.38</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0.99</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.362</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0.99</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.282</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0.99</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.282</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>0.99</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.354</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>0.95</td>
<td>5</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.36</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>0.99</td>
<td>10</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.334</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0.99</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.324</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>0.99</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>0.392</td>
</tr>
</tbody>
</table>
Chapter 3
Multiclass Bayesian Additive Classification Trees for Ordered Response

3.1 Introduction

Sometimes a natural ordering is exhibited by a variable. For example, a college student may be classified into one of the ordered gradations: Freshman, Sophomore, Junior, or Senior. A corporate bond can be classified into one of the ordered gradations: Junk grade, Low grade, Medium grade, or High grade. A consumer survey may ask if the satisfaction of a customer is one of the ordered gradations: Low, Medium, or High. This chapter deals with the utility of Bayesian ensemble of trees to estimate the probability that a unit belongs to one of $K > 2$ ordered classes. That is, estimate $p_j(x) = P(y = j|x)$ for $j = 1, \ldots, K$, where $x \in \mathbb{R}^p$ and $y \in \{1, 2, \ldots, K\}$ are a set of predictors and the ordered multiclass response.

A detailed classical treatment of ordinal categorical data analysis is found in Agresti (2010); Tutz and Hennevogl (1996) mainly dealing with a function of the linear combination of the predictors being used to estimate the categorical class probabilities. Among the early works dealing with ordinal regression is McCullagh (1980) using a link function to relate a linear predictor to the cumulative probabilities. Similarly, several Bayesian procedures are proposed including Albert and Chib (1993); Cowles (1996); Chen and Dey (2000); Johnson and Albert (2006).

The link functions that are often used are the logit, log-log, complementary log-log and the probit functions (see Table 3.1 listing the mathematical forms of these link
Table 3.1  Commonly used link functions for ordinal cumulative link models, where $\Phi (\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

<table>
<thead>
<tr>
<th>Name</th>
<th>Link function $h(\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>logit</td>
<td>$\log \left( \frac{\omega}{1-\omega} \right)$</td>
</tr>
<tr>
<td>log-log</td>
<td>$- \log (- \log (\omega))$</td>
</tr>
<tr>
<td>complementary log-log</td>
<td>$- \log (- \log (1 - \omega))$</td>
</tr>
<tr>
<td>probit</td>
<td>$\Phi^{-1}(\omega)$</td>
</tr>
</tbody>
</table>

functions. Suppose $h(\cdot)$ is a generic link function, and $P(y \leq k|x) = \sum_{j=1}^{k} p_j(x)$ for $k = 1, \ldots, K$ is the cumulative conditional probability that a unit belonging to at most the $k^{th}$ ordinal class, then the linear ordinal regression model is represented as

$$h(P(y \leq k|x)) = \gamma_k - x^T \beta,$$  \hspace{1cm} (3.1)

where $\gamma_0 = -\infty, \gamma_1 = 0, \ldots, \gamma_K = +\infty$ with order restriction $\gamma_k < \gamma_{k+1}$ are threshold parameters. The underlying model assumes that there exists a latent continuous random variable whose range is subdivided into $K$ contiguous partitions by the threshold parameters, where a class $y = k$ is associated with a latent variable $z$ in the range $(\gamma_{k-1}, \gamma_k]$. The ordinal classification model in (3.1) is further extended to modeling longitudinal ordinal data and repeated measures ordinal data in Kauermann (2000); Singer et al. (2004), while Kauermann and Tutz (2003, 2000) employ a semi-parametric modeling approach which allows some predictors to non-linearly impact the ordinal response.

Ensemble of regression trees models such as random forest (Breiman, 2001), gradient boosting (Friedman, 2001), Bayesian additive regression trees (Chipman et al., 2010) have shown to have very good predictive performance when intricate relationships exist between predictors and responses and when the number of predictors in the data is larger than the number of samples available. Often, practitioners ignore the natural ordering in the response variable when using these statistical procedures. The literature is sparse with respect to ensemble of trees models that seek to extract
information from the ordinal nature of such data sets.

In the case of a single tree CART model (Breiman et al., 1984), tree splitting criteria that consider increasing mis-classification cost as the actual class is farther from the predicted class are considered in Piccarreta (2008) through the ordinal impurity criterion (Piccarreta, 2001). We briefly describe the CART Gini impurity criterion used for nominal classification followed by ordinal Gini impurity. For a terminal node $t$ of a classification tree, its Gini impurity is

$$\text{GI}_t = \sum_{k=1}^{K} \sum_{l=1}^{K} C_{kl} \pi(k|t) \pi(l|t)$$

where

$$\pi(k|t) = \frac{\sum_{i=1}^{n_t} I(y_i = k)}{n_t}$$

for $k \in \{1, \ldots, K\}$ is the proportion of observations that belong to ordinal class $k$ in a node $t$ which has $n_t$ number of observations, and $C_{kl}$ is the mis-classification cost associated with classifying an observation that belongs to class $l$ as belonging to class $k$. If a parent tree node $p$ is split into left and right nodes $l$ and $r$, then CART seeks to minimize the increase in impurity, $\text{ GI}_l + \text{ GI}_r - \text{ GI}_p$. Piccarreta (2008) uses the mis-classification cost $C_{kl} = s_k - s_l$, where $s_1 < s_2 < \ldots < s_K$ are scores associated with the ordinal classes. In a slightly different approach, Xia et al. (2006) consider the ranking impurity measure which they define as

$$\text{RI}_t = \sum_{k=1}^{K} \sum_{l=1}^{K} (k - l) N(k|t) N(l|t),$$

where $N(k|t)$ is the number of observations in node $t$ that belong to the ordinal category $k$. The conditional regression trees approach of Hothorn et al. (2006) also incorporates the flexibility to model ordinal response data. In contrast to CART (Breiman et al., 1984; Quinlan, 1986) which search the best variable and splitting value combination that minimize a splitting criterion, Hothorn et al. (2006) proposed a two step tree splitting algorithm: the first step identifying the splitting predictor and the second step selecting a splitting value given the selected predictor. In the first variable selection step, a statistical test for independence between the response
variable and any of the predictors is conducted to select a predictor with the strongest
association with the response. In this step, a statistical test that takes into account
the ordinal nature of the response can be constructed (Hothorn et al., 2006; Janitz et
al., 2016). The second step selects the best splitting value by minimizing a pre-
specified criterion.

The few works that consider ensemble of trees models for ordinal classification
include Archer and Mas (2009) which utilizes bootstrap aggregation of regression trees
whose splitting rule is based on the ordinal Gini impurity measure; the conditional
trees of Hothorn et al. (2006) are also used as building blocks in an ensemble of trees
ordinal classification (Janitz et al., 2016).

Bayesian CART (Chipman et al., 1998; Denison et al., 1998) and their ensembles
Chipman et al. (1998); Pratola (2013); Kapelner and Bleich (2013); Pratola et al.
(2014); Kindo et al. (2016) have shown very good predictive accuracy. This work,
thus, deals with leveraging the flexibility of Bayesian additive regression trees for the
purpose of ordinal classification.

The remaining sections of this Chapter are outlined as follows. Section 3.2 sets up
the model ordered multiclass Bayesian additive classification trees (O-MBACT) and
Section 3.3 delves into the prior specifications and posterior computations. Sections
3.4 and 3.5 apply O-MBACT to simulated data set and real data sets, respectively.
Section 3.6 provides concluding remarks.

3.2 Ordered multiclass Bayesian additive classification trees

(O-MBACT)

The observable data for ordinal regression considered in this chapter is \((y_i, x_i)\), \(i = 1, \ldots, n\), where \(x_i \in \mathbb{R}^p\) and \(y_i\) represent the predictors and the ordinal response,
respectively. We consider modeling the conditional cumulative probability as

\[ z_i = G(x_i; T, M) + \epsilon_i, \quad \epsilon_i \sim N(0, 1), \tag{3.4} \]

which can also be equivalently represented as

\[ \Phi^{-1}(P(y_i \leq k | x_i)) = \gamma_k - G(x_i; T, M), \tag{3.5} \]

for \( i = 1, \ldots, n, \ k = 1, \ldots, K-1 \), where \( \Phi(\cdot) \) is the standard normal cumulative distribution function, and \( G(x_i; T, M) \) a sum of classification trees,

\[ G(x_i; T, M) = g_1(x_i; T_1, M_1) + \ldots + g_{n_T}(x_i; T_{n_T}, M_{n_T}). \tag{3.6} \]

The parameter \( T = (T_1, \ldots, T_{n_T}) \) denotes the structure of the trees in the sum whose corresponding terminal node parameters are denoted by \( M = (\theta_1, \ldots, \theta_{n_T}) \), where \( \theta_j = (\theta_{j1}, \ldots, \theta_{jm_j}) \) is the vector of terminal node parameters of the \( j^{th} \) tree in the sum that has \( m_j \) terminal nodes. The model in (3.4) can be motivated by assuming that the values taken by the response \( y \) arise from an unobserved latent variable \( z \). Specifically, \( y_i = k \) if \( z_i \in (\gamma_{k-1}, \gamma_k] \) for \( k = 1, \ldots, K \). Here, we take \( \gamma_0 = -\infty, \ \gamma_K = +\infty, \ \gamma_1 = 0, \) and \( \gamma_k < \gamma_{k+1} \) for the sake of identifiability. Thus, the parameters of ordered multiclass Bayesian additive classification trees are the bin/threshold parameters \( \gamma = (\gamma_0, \ldots, \gamma_K) \), the tree structures \( T \), and the terminal node parameters \( M \).

### 3.3 Prior specifications and posterior computation for ordered multiclass Bayesian additive classification trees

Uniform noninformative prior are placed on the threshold parameters \( \gamma \), and the prior distribution on the regression tree, say \( T_j \) in the sum is specified through the “tree generating stochastic process” of Chipman et al. (1998). For any two trees in the sum, their priors are assumed to be independent. We refer the reader to Chapter
1.1 and Chipman et al. (1998, 2010); Kindo et al. (2016) for further details. For the
priors on the terminal node parameters, we assume that they are independent of the
tree priors and that given a tree \( T_j \), its terminal node parameters \( \theta_{j1}, \ldots, \theta_{jm} \) follow
independent Gaussian distributions. That is,

\[
\theta_{jl} | T_j \sim N \left( 0, \sigma_0^2 \right),
\]

where \( \sigma_0 = \frac{3}{\kappa \sqrt{n_T}} \). This prior specification is selected so that the overall contribution
of the prior distribution on all the trees in the sum places high probability on the
interval \((-3, 3)\). Default values of \( k = 2 \) and \( n_T = 200 \) have shown good results in
most situations (Chipman et al., 2010; Kindo et al., 2016).

We now seek to find the posterior distribution of the parameters which are gov-
erned by the model in (3.4). The joint posterior distribution of all the parameters in
the model, \( p(M, T, \gamma | x, y) \), up to a proportionality constant, is given by

\[
p(M, T) p(\gamma) \prod_{i=1}^{n} \sum_{k=1}^{K} I(y_i = k) P(y_i = k | M, T, \gamma, z),
\]

where

\[
P(y_i = k | M, T, \gamma, z) = \left[ \Phi (\gamma_k - G(x_i, M, T)) - \Phi (\gamma_{k-1} - G(x_i, M, T)) \right],
\]

and \( p(M, T) \) is the joint prior of the tree structure and the terminal node parameters,
and \( p(\gamma) \) the prior on the bin parameters \( \gamma \). Since a closed form computation of the
joint posterior (3.7) is difficult, we implement a Gibbs sampler algorithm. Note that
the joint distribution \( p(M, T, \gamma, z | x, y) \) after placing a non-informative prior on \( \gamma \) is
proportional to

\[
\prod_{i=1}^{n} \left[ \phi (z_i - G(x_i, M, T)) \left\{ \sum_{k=1}^{K} I(y_i = k) I(\gamma_{k-1} < z_i \leq \gamma_k) \right\} \right].
\]

Thus, the conditional posterior distributions of \( z_1, \ldots, z_n \), given \( M, T, \gamma, x, y \), are

\[
\begin{align*}
    z_i | M, T, \gamma, y_i = k & \sim N \left( G(x_i, M, T), 1 \right) I(\gamma_{k-1} < z_i \leq \gamma_k) \\
    \text{for } k = 1, \ldots, K,
\end{align*}
\]

40
and the conditional density of $\gamma_k$, given $\{\gamma_j, j \neq k\}, M, T, \gamma, x, y$, is proportional to
\[
\prod_{i=1}^{n} [I\{y_i = k\}I\{\gamma_{k-1} < z_i \leq \gamma_k\} + I\{y_i = k + 1\}I\{\gamma_k < z_i \leq \gamma_{k+1}\}]. \quad (3.10)
\]
The Gibbs sampler is implemented by cycling through draws from (3.10), (3.9), and $n_T$ successive draws of $(T_j, M_j) \mid (T_{-j}, M_{-j}), z, y, x$ enumerated below, where $(T_{-j}, M_{-j})$ denotes all trees and their terminal node parameters in the sum with the exception of the $j^{th}$ tree.

i. Observe from (3.10) that $\gamma_k$, given $\{\gamma_j, j \neq k\}, M, T, \gamma, x, y$, can be sampled from a uniform distribution with the upper bound
\[
b = \min(\min_i \{z_i : y_i = k + 1\}, \gamma_{k+1}),
\]
and lower bound
\[
a = \max(\max_i \{z_i : y_i = k\}, \gamma_{k-1}).
\]

ii. Draws of $z_i, i = 1, \ldots, n$, from truncated normal distribution given in (3.9).

iii. The $n_T$ successive draws of
\[
(T_j, M_j) \mid (T_{-j}, M_{-j}), z, y, x \quad (3.11)
\]
are done by rewriting (3.4) as
\[
\gamma_k - z = \sum_{l=1}^{n_T} g(x; T_l, M_l),
\]
and further rearranging to obtain
\[
\gamma_k - z - \sum_{l \neq j} g(x; T_l, M_l) = g(x; T_j, M_j). \quad (3.12)
\]
Now, the left hand side of (3.12) can be considered as a residual $R$ associated with the $j^{th}$ tree in the sum-of-trees. Hence, we can sample from (3.11) by considering the residuals as the dependent variable of the single Bayesian CART model in Chipman et al. (1998). Similar ‘Bayesian back-fitting’ algorithms are implemented in Chipman et al. (2010); Kindo et al. (2016).
3.4 Simulation study for ordered MBACT

In this section, we simulate a 5-class data set in which the response is ordered, and nonlinearly related to the predictors. Only twenty of the fifty predictors \( x_1, \ldots x_{50} \) which are independent and identically distributed standard normal random variables, are related to the multiclass response. The simulation setup is as follows: Define random variables \( X_1^*, X_2^* \), and \( Y \) as

\[
X_1^* = \max \{ \min \{ x_1, x_2, x_3 \}, \min \{ x_4, x_5, x_6 \}, \min \{ x_7, x_8, x_9 \} \}, \quad X_2^* = \min \{ \max \{ x_{11}, x_{12}, x_{13} \}, \max \{ x_{14}, x_{15}, x_{16} \}, \max \{ x_{17}, x_{18}, x_{19} \} \}, \text{ and} \]

\[
Y = aX_1^* + bX_2^* + \sin(cx_{10}) + \cos(dx_{20}),
\]

where \( a, b, c, \) and \( d \) are real numbers. \( Y \) is then discretized using monotone class boundary points in order to obtain ordered multiclass responses. The boundary points that separate class labels are calculated using the \( Y \)-values in the training data in such a way that the number of observations belonging to each of the classes are nearly equal. For the simulation study in this section, we used data generating parameters: \( a = 0.25, b = 0.30, c = 0.20, d = 0.10 \), and 500 instances in the training as well as the test data sets. A two-dimensional projection of the response data is shown in Figure 3.1 to illustrate the nonlinearity of the relationship between the predictors and the ordinal response.

The predictive performance of each procedure is evaluated using the Mean Absolute Deviation (MAD) criterion defined as

\[
\frac{1}{m} \sum_{j=1}^{m} | \hat{\omega}_j - \omega_j |,
\]

where \( \hat{\omega}_j \) and \( \omega_j \) are the predicted and actual ordered classes of the \( j^{th} \) observation in a given test data set, respectively. The idea is that the cost of making a prediction error is higher for those predictions farther from the truth. The test error rates and standard errors (in parentheses) in Table 3.2 are based on 50 replications of
Figure 3.1  This figure illustrates the nonlinearity of the relationship between the predictors and ordinal response of the simulated data.

Table 3.2  Average MADs (Mean Absolute Deviations) as defined in (3.13) with standard errors in parentheses for Nonlinear Ordered Simulated Data

<table>
<thead>
<tr>
<th>Methodology</th>
<th>TestMAD</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>1.2596(0.0106)</td>
<td>5</td>
</tr>
<tr>
<td>CART</td>
<td>1.3363(0.0238)</td>
<td>8</td>
</tr>
<tr>
<td>SVM.L</td>
<td>1.2982(0.0118)</td>
<td>7</td>
</tr>
<tr>
<td>SVM.R</td>
<td>1.2854(0.0094)</td>
<td>6</td>
</tr>
<tr>
<td>NNET</td>
<td>1.2002(0.0116)</td>
<td>2</td>
</tr>
<tr>
<td>Multinom</td>
<td>1.2128(0.0097)</td>
<td>4</td>
</tr>
<tr>
<td>KNN</td>
<td>1.6566(0.0248)</td>
<td>10</td>
</tr>
<tr>
<td>QDA</td>
<td>1.4446(0.0121)</td>
<td>9</td>
</tr>
<tr>
<td>LDA</td>
<td>1.2125(0.0100)</td>
<td>3</td>
</tr>
<tr>
<td>O-MBACT</td>
<td>0.6984(0.0107)</td>
<td>1</td>
</tr>
</tbody>
</table>

training-test pairs. As we see from the results of this simulation study and real data applications of Section 3.5, ordered MBACT has a good predictive performance as measured by MAD when the response is of ordinal type.
Table 3.3  Selected tuning parameters for the nonlinear ordered simulated data. The number of trees used is 200 with posterior sample of size 3,000 after discarding the initial 1,000.

<table>
<thead>
<tr>
<th>Fold</th>
<th>α</th>
<th>β</th>
<th>Test MAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9</td>
<td>2.0</td>
<td>0.6560</td>
</tr>
<tr>
<td>2</td>
<td>0.9</td>
<td>3.0</td>
<td>0.6280</td>
</tr>
<tr>
<td>3</td>
<td>0.95</td>
<td>3.0</td>
<td>0.7100</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>2.0</td>
<td>0.6420</td>
</tr>
<tr>
<td>5</td>
<td>0.99</td>
<td>2.0</td>
<td>0.6300</td>
</tr>
<tr>
<td>6</td>
<td>0.99</td>
<td>2.0</td>
<td>0.6400</td>
</tr>
<tr>
<td>7</td>
<td>0.99</td>
<td>2.0</td>
<td>0.6760</td>
</tr>
<tr>
<td>8</td>
<td>0.9</td>
<td>2.0</td>
<td>0.6920</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>2.0</td>
<td>0.6720</td>
</tr>
<tr>
<td>10</td>
<td>0.95</td>
<td>2.0</td>
<td>0.6480</td>
</tr>
<tr>
<td>11</td>
<td>0.95</td>
<td>3.0</td>
<td>0.7020</td>
</tr>
<tr>
<td>12</td>
<td>0.99</td>
<td>3.0</td>
<td>0.6700</td>
</tr>
<tr>
<td>13</td>
<td>0.95</td>
<td>3.0</td>
<td>0.6960</td>
</tr>
<tr>
<td>14</td>
<td>0.9</td>
<td>2.0</td>
<td>0.6180</td>
</tr>
<tr>
<td>15</td>
<td>0.95</td>
<td>3.0</td>
<td>0.6560</td>
</tr>
<tr>
<td>16</td>
<td>0.99</td>
<td>2.0</td>
<td>0.6440</td>
</tr>
<tr>
<td>17</td>
<td>0.95</td>
<td>2.0</td>
<td>0.6540</td>
</tr>
<tr>
<td>18</td>
<td>0.95</td>
<td>2.0</td>
<td>0.5940</td>
</tr>
<tr>
<td>19</td>
<td>0.95</td>
<td>3.0</td>
<td>0.6480</td>
</tr>
<tr>
<td>20</td>
<td>0.95</td>
<td>3.0</td>
<td>0.7440</td>
</tr>
</tbody>
</table>

3.5 Real data illustration

In this section, we compare the predictive performance of ordered MBACT, and other competing methods on modified real data sets. The data sets are modified so that the response vector is ordinal. That is, the data sets in their original form had continuous numerical responses, but we discretize the continuous responses using bins/buckets by creating monotone class boundaries as described in Chu and Ghahramani (2005).

Four data sets used for comparing O-MBACT with some existing classification methodologies are Ailerons, Triazines, Wisconsin breast cancer, and Abalone from the UCI machine learning repository (Bache and Lichman, 2013) with brief descriptions as follows. Ailerons data set is used to predict the control action while the F16
Table 3.4  Description of ordered real data sets.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Ailerons</th>
<th>Triazines</th>
<th>Wisconsin</th>
<th>Abalone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predictors(Numeric, Nominal)</td>
<td>40(40, 0)</td>
<td>60(60,0)</td>
<td>32(32,0)</td>
<td>8(7,1)</td>
</tr>
<tr>
<td>Training Cases</td>
<td>6438</td>
<td>149</td>
<td>155</td>
<td>3759</td>
</tr>
<tr>
<td>Test Cases</td>
<td>716</td>
<td>58</td>
<td>39</td>
<td>418</td>
</tr>
<tr>
<td>Number of Replications Done</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

fighter aircraft is airborne. The predictors describe the status of the aircraft. Triazines data set was used to study the quantitative structure-activity relationship of 2,3-diamino-6,6-dimethyl-5-phenyl-dihydrotriazine derivatives and their inhibition of dihydrofolate reductase (see Hirst et al. (1994) for details). The data set Wisconsin breast cancer has 32 predictors and a response which is either time to recurrence if a breast cancer patient returns due to a recurrence of the disease, or time to being free of cancer if the disease does not recur. Abalone data set relates the age of abalone to the attributes of a mollusk such as gender, weight, shell weight, and so forth. An accurate measure of the age involves observing the number of rings which is done after cutting the shell through the cone. A summary of the data sets is in Table 3.4.

The response vector for each data set is discretized in order to obtain a 5-class ordered classification problem. The results in Tables 3.5 and 3.6 are the average mean absolute deviations(MADs) (3.13) and standard errors on the test data sets over the number of replications mentioned in the last row of Table 3.4.

3.6 Conclusion

In this chapter, we have introduced Bayesian ordinal classification procedure O-MBACT: Ordered multiclass Bayesian additive classification trees as ordinal classification extensions of BART: Bayesian Additive Regression Trees and binary BART (Chipman et al., 2010; Zhang and Härdle, 2010). O-MBACT is suitable for modeling response vectors which exhibit inherent ordering. O-MBACT performs particularly well when the relationship between the predictors and the response is nonlinear and
Table 3.5 Results of ordered real data sets Wisconsin and Abalone. This table shows MADs defined in (3.13), and standard errors (in parentheses) on test data sets over the number of replications mentioned in the last row of Table 3.4.

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Wisconsin</th>
<th></th>
<th>Abalone</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>1.2530 (0.0620)</td>
<td>3</td>
<td>0.6696 (0.0134)</td>
<td>6</td>
</tr>
<tr>
<td>CART</td>
<td>1.5304 (0.0578)</td>
<td>8</td>
<td>0.7409 (0.0166)</td>
<td>8</td>
</tr>
<tr>
<td>SVMLinear</td>
<td>1.5108 (0.0376)</td>
<td>7</td>
<td>0.6701 (0.0111)</td>
<td>7</td>
</tr>
<tr>
<td>SVMRadial</td>
<td>1.2794 (0.0519)</td>
<td>4</td>
<td>0.6380 (0.0153)</td>
<td>3</td>
</tr>
<tr>
<td>NNET</td>
<td>1.5104 (0.0615)</td>
<td>6</td>
<td>0.6071 (0.0165)</td>
<td>1</td>
</tr>
<tr>
<td>Multinom</td>
<td>1.3719 (0.0477)</td>
<td>5</td>
<td>0.6550 (0.0082)</td>
<td>4</td>
</tr>
<tr>
<td>KNN</td>
<td>1.2424 (0.1329)</td>
<td>2</td>
<td>0.6655 (0.0172)</td>
<td>5</td>
</tr>
<tr>
<td>O-MBACT</td>
<td>1.1356 (0.0865)</td>
<td>1</td>
<td>0.6280 (0.0126)</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.6 Results of ordered real data sets Triazines and Ailerons. This table shows MADs defined in (3.13), and standard errors (in parentheses) on test data sets over the number of replications mentioned in the last row of Table 3.4.

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Triazines</th>
<th></th>
<th>Ailerons</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>0.9417 (0.0581)</td>
<td>1</td>
<td>0.4091 (0.0047)</td>
<td>3</td>
</tr>
<tr>
<td>CART</td>
<td>1.3852 (0.1223)</td>
<td>7</td>
<td>0.8203 (0.0118)</td>
<td>6</td>
</tr>
<tr>
<td>SVMLinear</td>
<td>1.1558 (0.0612)</td>
<td>6</td>
<td>0.4501 (0.0146)</td>
<td>5</td>
</tr>
<tr>
<td>SVMRadial</td>
<td>1.1354 (0.0753)</td>
<td>4</td>
<td>0.9639 (0.0083)</td>
<td>7</td>
</tr>
<tr>
<td>NNET</td>
<td>1.1930 (0.0869)</td>
<td>8</td>
<td>0.4379 (0.0220)</td>
<td>4</td>
</tr>
<tr>
<td>Multinom</td>
<td>1.0523 (0.1267)</td>
<td>3</td>
<td>0.4087 (0.0074)</td>
<td>2</td>
</tr>
<tr>
<td>KNN</td>
<td>1.1411 (0.0957)</td>
<td>5</td>
<td>1.1608 (0.0110)</td>
<td>8</td>
</tr>
<tr>
<td>O-MBACT</td>
<td>1.0172 (0.0711)</td>
<td>2</td>
<td>0.3668 (0.0068)</td>
<td>1</td>
</tr>
</tbody>
</table>

it is robust to existence of unimportant predictors.
Table 3.7  Selected tuning parameters for Ailerons dataset. The other tuning parameters that are not selected via cross validation are $n_T = 200$, $\kappa = 3.0$. The number or posterior samples is 3000 with initial 1000 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>Test MAD</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.333799</td>
<td>0.9</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>0.371508</td>
<td>0.95</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>0.389665</td>
<td>0.8</td>
<td>2.0</td>
</tr>
<tr>
<td>4</td>
<td>0.361732</td>
<td>0.8</td>
<td>2.0</td>
</tr>
<tr>
<td>5</td>
<td>0.338462</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>0.404196</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>7</td>
<td>0.363636</td>
<td>0.8</td>
<td>3.0</td>
</tr>
<tr>
<td>8</td>
<td>0.38042</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>9</td>
<td>0.363636</td>
<td>0.9</td>
<td>2.0</td>
</tr>
<tr>
<td>10</td>
<td>0.360839</td>
<td>0.95</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 3.8  Selected tuning parameters for Triazines dataset. The other tuning parameters that are not selected via cross validation are $n_T = 200$, $\kappa = 3.0$. The number or posterior samples is 3000 with initial 1000 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>Test MAD</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8157895</td>
<td>0.95</td>
<td>3.0</td>
</tr>
<tr>
<td>2</td>
<td>1.1891892</td>
<td>0.80</td>
<td>3.0</td>
</tr>
<tr>
<td>3</td>
<td>1.1621622</td>
<td>0.80</td>
<td>2.0</td>
</tr>
<tr>
<td>4</td>
<td>0.9189189</td>
<td>0.80</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>1.0000000</td>
<td>0.90</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 3.9  Selected tuning parameters for Wisconsin dataset. The other tuning parameters that are not selected via cross validation are $n_T = 200$, $\kappa = 3.0$. The number or posterior samples is 3000 with initial 1000 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>Test MAD</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9487179</td>
<td>0.80</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>1.0256410</td>
<td>0.95</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>1.0769231</td>
<td>0.99</td>
<td>3.0</td>
</tr>
<tr>
<td>4</td>
<td>1.1794872</td>
<td>0.95</td>
<td>3.0</td>
</tr>
<tr>
<td>5</td>
<td>1.4473684</td>
<td>0.99</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Table 3.10  Selected tuning parameters for Abalone dataset. The other tuning parameters that are not selected via cross validation are $n_T = 200$, $\kappa = 3.0$. The number or posterior samples is 3000 with initial 1000 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>Test MAD</th>
<th>$\alpha$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.677033</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>0.564593</td>
<td>0.99</td>
<td>3.0</td>
</tr>
<tr>
<td>3</td>
<td>0.691388</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>0.643541</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>0.598086</td>
<td>0.99</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>0.605263</td>
<td>0.99</td>
<td>1.0</td>
</tr>
<tr>
<td>7</td>
<td>0.665072</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>8</td>
<td>0.618705</td>
<td>0.99</td>
<td>3.0</td>
</tr>
<tr>
<td>9</td>
<td>0.609113</td>
<td>0.9</td>
<td>2.0</td>
</tr>
<tr>
<td>10</td>
<td>0.606715</td>
<td>0.99</td>
<td>2.0</td>
</tr>
</tbody>
</table>
4.1 Introduction

Quantile regression gives a comprehensive picture of the relationship between a response variable and a set of predictors. It is particularly appealing when the inferential interest lies in the probabilistic properties of extreme observations conditional on a set of predictors. Such objectives arise in various disciplines: in environmental sciences, Friederichs and Hense (2007) study the probabilistic properties of extreme precipitation events, while Pedersen (2015) model the tail distribution of stock and bond returns. In an epidemiological study, Burgette et al. (2011) use penalized quantile regression to explore covariates that affect the lower tail of the distribution of birth weight of babies. When the distribution of the dependent variable is skewed, the desire for robustness in the presence of extreme observations makes quantile regression a preferred approach. Examples include the study of tourist expense patterns in Marrocu et al. (2015) and wage distribution in Buchinsky (1995).

Extensive work in the theory and application of linear quantile regression can be found in Koenker and Bassett Jr (1978); Koenker (1994); Buchinsky (1998); Tsai (2012); Cole and Green (1992). Suppose we have a data set \((y_i, x_i)\) for \(i = 1, \ldots, n\), where \(y_i \in \mathcal{R}\) and \(x_i \in \mathcal{R}^d\) denote the observed response and predictors for the

---

\(^2\) Kindo, B. P., Wang, H., Hanson, T., & Peña, E. A. Bayesian quantile additive regression trees. Submitted to Bayesian Analysis. 10/12/2016
ith observation, respectively. Analogous to the use of the mean function \( E(y|x) \) used in least squares regression to explain the relationship between the response and predictors, quantile regression uses the \( \tau \)th quantile function \( Q(y|x, \tau) \), where \( \tau \in (0, 1) \). The \( \tau \)th quantile of a random variable \( Y \) with distribution \( F \) is defined as \( Q(\tau) = \inf \{ y : F(y) \geq \tau \} \), where \( F(\cdot) \) denotes the cumulative distribution function. Thus, for a given quantile value \( \tau \), quantile regression seeks to estimate \( Q(x, \tau) = \inf \{ y : F(y|x) \geq \tau \} \). The linear quantile regression problem in particular is described as the minimization problem

\[
\hat{\beta}_\tau = \arg \min_{\beta} \sum_{i=1}^{n} \rho_\tau \left( y_i - x_i^T \beta \right),
\]

where \( \rho_\tau(\omega) = \omega (\tau - I \{\omega < 0\}) \) is usually termed as the “check loss” function. The error distribution is left largely unspecified except that its \( \tau \)th quantile equals zero. The work in Koenker and Bassett Jr (1978) spearheaded the use of quantile regression as a robust alternative to mean regression. More recently, \( l_1 \) regularized quantile regression with simultaneous variable selection and parameter estimation is studied in Zou and Yuan (2008); Belloni and Chernozhukov (2011).

An alternative, yet equivalent, formulation of (4.1) assumes that the random errors follow the asymmetric Laplace distribution (Yu and Moyeed, 2001; Kozumi and Kobayashi, 2011; Sriram et al., 2013). If a random variable \( Y \) follows an asymmetric Laplace distribution \( ALD(y; \tau, \mu) \) with location parameter \( \mu \in \mathbb{R} \), its density function is given by

\[
f_\tau(y; \mu) = \tau (1 - \tau) \exp \left\{ -\rho_\tau(y - \mu) \right\} ,
\]

where \( \tau \in (0, 1), \rho_\tau(\omega) = \omega (\tau - I \{\omega < 0\}) \) for \( \omega \in \mathbb{R} \). A special case of (4.2) with \( \tau = 0.5 \) is the Laplace double exponential distribution. Figure 4.1 shows the plots of the probability density functions of asymmetric Laplace distributions for fixed location parameter \( \mu = 0 \), and values of \( \tau \in \{0.25, 0.50, 0.83\} \). The expectation and
variance of $Y \sim \text{ALD}(\tau, \mu = 0)$ are

$$
E(Y) = \frac{1 - 2\tau}{\tau(1 - \tau)} \text{ and } \text{Var}(Y) = \frac{1 - 2\tau + 2\tau^2}{\tau^2(1 - \tau)^2},
$$

while its characteristic function is $\psi_Y(t) = \left[\frac{1}{2} \vartheta_2^2 t^2 - \vartheta_1 ti + 1\right]^{-1}$, where $\vartheta_1 = \frac{1 - 2\tau}{\tau(1 - \tau)}$ and $\vartheta_2^2 = \frac{2}{\tau(1 - \tau)}$.

Some Bayesian approaches to the quantile regression problem in general and median regression in particular have been considered in Yu and Moyeed (2001); Dunson and Taylor (2005); Taddy and Kottas (2012); Hanson and Johnson (2002); Kozumi and Kobayashi (2011); Kottas and Gelfand (2001); Reich et al. (2010) either by assuming asymmetric Laplace, Dirichlet process mixtures, Polya trees, or Gaussian mixture approximations as the distribution of the random error terms. In particular, Kozumi and Kobayashi (2011) outline a Gibbs sampler for Bayesian quantile regression based on a mixture representation of the asymmetric Laplace distribution. With the intention of utilizing their approach, we paraphrase their finding which they show using the equality of characteristic functions. If the random variables $V$ and $Z$ which follow the standard exponential and Gaussian distributions, respectively, are
mutually independent, then \( W = \vartheta_1 V + \vartheta_2 \sqrt{V} Z \) is equal in distribution to the asymmetric Laplace distribution \( \text{ALD}(\tau, \mu = 0) \), where \( \vartheta_1 = \frac{1 - 2\tau}{\tau(1 - \tau)} \) and \( \vartheta_2 = \frac{2}{\tau(1 - \tau)} \). Such representation allows a formulation of an efficient algorithm to estimate regression quantiles in a Bayesian framework that involves simulations from the Gaussian and Generalized Inverse Gaussian distributions.

In comparison to least squares regression trees, quantile regression trees or their ensembles have not yet garnered much attention. However, sporadic works in the literature exist including the single tree quantile regression model of Chaudhuri and Loh (2002) and the quantile regression forests model in Meinshausen (2006) which extends on the idea of random forests (Breiman, 2001). In the quantile regression forests model of Meinshausen (2006), all of the observations that lie in a regression tree terminal node are used for estimation, while a summary statistic (typically the average) of the observations in a terminal node are used by random forests. At the core of the quantile regression forests is the empirical estimation of the conditional cumulative density function \( F(y|x) = P(Y \leq y|x) \) so that \( \hat{Q}(x, \tau) = \inf \{ y : \hat{F}(y|x) \geq \tau \} \), where \( \hat{F} \) is an estimator of \( F \).

Bayesian regression trees and their ensembles are shown to have enhanced predictive performance in the framework of least squares regression, and binary and multiclass classification (Chipman et al., 1998, 2010; Abu-Nimeh et al., 2007; Zhang and Härdle, 2010; Pratola et al., 2014; Kapelner and Bleich, 2013; Kindo et al., 2016). In particular, BART - Bayesian additive regression trees (Chipman et al., 2010) estimates the conditional mean of a response given a set of predictors by using a sum of regression trees model

\[
y = \sum_{j=1}^{n_T} g_j(x; T_j, M_j) + \epsilon, \quad \text{where } \epsilon \sim N(0, \sigma^2). \tag{4.4}
\]

BART is specified through priors on the regression trees via a “tree generating stochastic process” that favors shallow trees and prior specifications on terminal node parameters that strategically shrink the influence of individual trees. BART has been
utilized in many applications with great predictive performance (Abu-Nimeh et al., 2007; Zhang and Härdle, 2010; Wu et al., 2010; He et al., 2009; Liu et al., 2015).

In this chapter we explore the utility of an ensemble of Bayesian regression trees to garner a comprehensive view of the dependence between a response and predictors. Thus, we propose a fully Bayesian framework for construction of quantile regression trees and their ensembles to complement the linear Bayesian quantile regression of Kozumi and Kobayashi (2011); Yu and Moyeed (2001) and quantile regression forests of Meinshausen (2006). We note that, at the time of this writing, we are not aware of a Bayesian counterpart in the literature to the frequentist quantile regression tree.

The remaining parts of this chapter are outlined as follows. Section 4.2 sets the framework for Bayesian quantile additive regression trees including the prior specifications on all the parameters of the model and posterior computations. Section 4.5 delves into the implementation of the model with simulation studies and real data applications. Section 4.8 extends Bayesian quantile additive trees to tackle binary classification problems along with a simulation study and real data application. Section 4.11 provides concluding remarks.

4.2 Bayesian quantile additive regression trees

In this section we outline the model specifications for Bayesian quantile additive regression trees. Specifically, let the observable data be \((y_i, x_i)\) for \(i = 1, \ldots, n\), where \(y_i \in \mathbb{R}\) and \(x_i \in \mathbb{R}^d\) denoting the response and predictors for the \(i^\text{th}\) observation. Consider the model

\[
y_i = G(x_i; T, M) + \vartheta_1 \nu_i + \vartheta_2 \phi^{1/2} \sqrt{\nu_i} z_i,
\]

\[
G(x_i; T, M) = \sum_{j=1}^{n_T} g(x_i; T_j, M_j) \tag{4.5}
\]

\[
p(\nu_i|\phi) = \frac{1}{\phi} \exp \left\{ -\frac{\nu_i}{\phi} \right\},
\]

\[
p(z_i) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} z_i^2 \right\},
\]
where $T_j$ and $M_j$ are the $j^{th}$ tree in the sum and its associated terminal node parameters, and $(T, M) = \{(T_j, M_j) : j = 1, \ldots, n_T\}$. Note that $v_1 \nu_i + v_2 \sqrt{\nu_i} z_i = \phi [v_1 \tilde{\nu}_i + v_2 \sqrt{\tilde{\nu}_i} z_i]$, where $\nu_i \sim \text{Exp}(1)$ and the quantity in the square brackets is the mixture representation of the asymmetric Laplace distribution.

4.3 Prior specifications

We assume that the priors on any two distinct trees in the sum are independent and the prior on $\phi$ is independent of the tree priors. That is, $(T_j, M_j) \perp (T_{j'}, M_{j'})$ for $j \neq j'$, and $(M, T) \perp \phi$. Further assuming that given a tree, say the $j^{th}$ tree $T_j$, the priors on its $m_j$ terminal node parameters are independent enables writing the prior distribution on $(T, M, \phi)$ as

$$p(T, M, \phi) = \prod_{j=1}^{n_T} p(T_j, M_j) p(\phi)$$

$$= \prod_{j=1}^{n_T} [p(T_j) p(M_j|T_j)] p(\phi) \quad (4.6)$$

$$= \prod_{j=1}^{n_T} [p(T_j) \prod_{k=1}^{m_j} p(\mu_{jk}|T_j)] p(\phi),$$

where $n_T$ is the number of trees in the sum and $m_j$ is the number of terminal nodes of tree $T_j$ (i.e., $M_j = (\mu_{j1}, \ldots, \mu_{jm_j})$).

The prior $p(T_j)$ is specified through a “tree generating stochastic process” of Chipman et al. (1998). This process is governed by tree splitting rule that creates non-overlapping partitions of the predictor space by selecting a splitting variable followed by a splitting value given the selected variable. Once a terminal node is randomly selected for use in the binary partitioning of the predictor space, a splitting variable is randomly chosen followed by a random selection of a value in the range of the selected predictor with the condition that no empty partition is created. Furthermore, the probability that a terminal node $\eta$ with depth $d_\eta$ (number of ancestor nodes) splits
is given by

\[ p_{\text{SPLIT}}(\eta) = \begin{cases} 
1 & \text{if } d_{\eta} = 0 \\
\frac{\psi_1}{(1+d_{\eta})^{\psi_2}} & \text{if } d_{\eta} > 0, 
\end{cases} \tag{4.7} \]

where \( \psi_1 \in (0,1), \psi_2 \in [0,\infty) \). The splitting probability in (4.7), and the choice of \( \psi_1 \) and \( \psi_2 \) play a crucial role of regulating the influence of individual trees in the sum. For example, higher values of \( \psi_2 \) and lower values of \( \psi_1 \) result in shallow trees in general.

Given a tree \( T_j \), the prior on the terminal node parameters is a Gaussian distribution \( \mu_{jk}|T_j \sim N(\mu_0, \sigma^2_0) \) for \( k = 1, \ldots, m_j \). In the model representation given in (4.5), the overall contribution of the prior distributions of the terminal node parameters on \( E(y|x) \) and \( \text{Var}(y|x) \) are \( n_T\mu_0 \) and \( n_T\sigma^2_0 \). The hyper-parameters \( \mu_0 \) and \( \sigma^2_0 \) are selected so that the overall effect induced by the prior distributions is in the interval \((\min(y), \max(y))\) with high probability. A convenient aspect of the quantile function is its invariance to a monotone transformation. In particular, we use the transformation \( \tilde{y} = h(y) = \frac{y - \min(y)}{(\max(y) - \min(y))} - 0.5 \) for which we have \( Q(y, \tau) = h^{-1}(Q(\tilde{y}, \tau)) \). Taking \( \tilde{y} \) as the dependent variable in (4.5) along with priors \( \mu_{jk}|T_j \sim N(\mu_0 = 0, \sigma^2_0 = \frac{1}{2\kappa\sqrt{n_T}}) \), we ensure that the transformed response is in the interval \((-0.5, 0.5)\). This choice of the hyper-parameters also ensures that the effect of the prior distributions on the terminal nodes places high probability to the same interval. We find that a value of \( \kappa \) between 2 and 3 gives reasonable results. Note that the larger the number of trees in the sum, the smaller the prior variance placed on the terminal node parameters effectively shrinking the influence of individual trees to zero. Finally, the prior on \( \phi \) is specified as an Inverse-Gamma distribution \( \phi \sim IG\left(\frac{\alpha}{2}, \frac{\beta}{2}\right) \).
4.4 Posterior updating scheme

The posterior updating scheme cycles through the following three posterior draws: a draw from

\[ p(V | T, M, Y, \phi) \]  

(4.8)

followed by consecutive updates of the \(j\)th tree and its terminal node parameters for \(j = 1, \ldots, n_T\) accomplished by a draw from

\[ p\{ (T_j, M_j) | M_{(-j)}, T_{(-j)}, \phi, X, Y \}, \]  

(4.9)

with \((T_{(-j)}, M_{(-j)})\) denoting all the trees and their terminal node parameters in the sum excluding the \(j\)th tree; and finally a draw from

\[ p\{ \phi | M, T, X, Y \}, \]  

(4.10)

where \(V = (\nu_1, \ldots, \nu_n)^T\), \(Y = (y_1, \ldots, y_n)^T\), and \(X = (x_1, \ldots, x_n)^T\). The posterior draw in (4.8) is \(n\) sequential samples from the Generalized Inverse Gaussian distribution

\[ p(\nu_i | T, M, \phi, y_i, x_i) \propto \nu_i^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \left[ \delta_1 \nu_i^{-1} + \delta_2 \nu_i \right] \right\}, \]  

(4.11)

where \(\delta_1 = \frac{(y_i - G(x_i; T, M))^2}{\frac{\partial^2}{\partial \phi^2}}\) and \(\delta_2 = \frac{2 \theta_1^2 + \theta_2^2}{\frac{\partial^2}{\partial \phi^2}}\). To describe the draw in (4.9), we re-write (4.5) as

\[ \omega_i \equiv y_i - \sum_{l \neq j} g(x_i; T_l, M_l) - \partial_1 \nu_i = g(x_i; T_j, M_j) + \phi \frac{1}{2} \theta_2 \sqrt{\nu_i} z_i \]  

(4.12)

so that \(\omega_i | x_i, \nu_i, T_{(-j)}, M_{(-j)}, \phi \sim N(g(x_i; T_j, M_j), \phi \theta_2^2 \nu_i)\). A Metropolis-Hastings algorithm is utilized to update the tree \(T_j\) with \(W = (\omega_1, \ldots, \omega_n)^T\) considered a residual psuedo-response variable. A similar Bayesian “back-fitting” algorithm is implemented in Chipman et al. (2010); Kindo et al. (2016).

For ease of explanation of the Metropolis-Hasting algorithm, we pursue a slight modification of notation as follows. Suppose that \(W_k = (\omega_{k1}, \ldots, \omega_{kn_k})^T\) is a vector of residuals that lie in the \(k\)th terminal node of the regression tree \(T_j\) which has
While its reverse counterpart PRUNE move randomly selects and collapses a pair of terminal nodes, and that \( \mathbf{X}_k = (x_{k1}, \ldots, x_{kn_k})^T \) denotes the corresponding set of predictors. Likewise, \( \mathbf{V}_k = (\nu_{k1}, \ldots, \nu_{kn_k})^T \) and \( \mathbf{Z}_k = (z_{k1}, \ldots, z_{kn_k})^T \) denote the components of the mixture representation of asymmetric Laplace error term corresponding to the observations in the \( k \)th terminal node. With this notation, we write \( \mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{m_j})^T \), \( \mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_{m_j})^T \), and \( \mathbf{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_{m_j})^T \), where \( n = n_1 + \ldots + n_{m_j} \). Similar notation is used in Chipman et al. (1998). We can then write the likelihood function of the single residual tree in (4.12) as

\[
f(\mathbf{W}|\mathbf{X}, \mathbf{V}, \phi, T_j, M_j) = \prod_{k=1}^{m_j} f(\mathbf{W}_k|\mathbf{X}_k, \mathbf{V}_k, \phi, T_j, M_j), \tag{4.13}
\]

where

\[
f(\mathbf{W}_k|\mathbf{X}_k, \mathbf{V}_k, \phi, T_j, M_j) = f(\mathbf{W}_k|\mu_{jk}, \mathbf{V}_k, \phi)
= \left[ \frac{1}{\sqrt{2\pi \varphi_2 \phi^2}} \right]^{n_k} \prod_{l=1}^{n_k} \nu_{kl}^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2\varphi_2 \phi} \sum_{l=1}^{n_k} \frac{(\omega_{kl} - \mu_{jk})^2}{\nu_{kl}} \right\}. \tag{4.14}
\]

With the prior specification \( \mu_{jk} \sim N(\mu_0 = 0, \sigma_0^2 = \frac{1}{2\sqrt{m_l}}) \), we have

\[
\int f(\mathbf{W}_k, M_j|\mathbf{X}_k, T_j, \mathbf{V}_k, \phi) dM_j \\
= \int f(\mathbf{W}_k|\mathbf{X}_k, T_j, \mathbf{V}_k, \phi) p(\mu_{jk}) d\mu_{jk} \\
= \left[ \frac{1}{\sqrt{2\pi \varphi_2 \phi^2}} \right]^{n_k} \prod_{l=1}^{n_k} \nu_{kl}^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2\varphi_2 \phi} \sum_{l=1}^{n_k} \frac{\omega_{kl}^2}{\nu_{kl}} \right\} \times \frac{\varphi_2^2 \phi}{\varphi_2^2 \phi + \sigma_0^2 \sum_{l=1}^{n_k} \nu_{kl}^{-1}} \exp \left\{ -\frac{\sigma_0^2 \sum_{l=1}^{n_k} \omega_{kl}^2 \nu_{kl}^{-1}}{2\varphi_2^2 \phi + \sigma_0^2 \sum_{l=1}^{n_k} \nu_{kl}^{-1}} \right\}. \tag{4.15}
\]

To draw from \( p \left\{ (T_j, M_j)|M_{(-j)}, T_{(-j)}, \phi, \mathbf{X}, \mathbf{Y} \right\} \), we first obtain a tree \( T_j^* \) as a candidate update to \( T_j \) accepted with a probability

\[
\min \left\{ 1, \frac{q(T_j^*, T_j)p(\mathbf{W}|\mathbf{X}, \mathbf{V}, T_j^*, \phi)p(T_j^*)}{q(T_j, T_j^*)p(\mathbf{W}|\mathbf{X}, \mathbf{V}, T_j, \phi)p(T_j)} \right\}. \tag{4.16}
\]

The transition kernel \( q(\cdot, \cdot) \) assigns probabilities of 0.25, 0.25, 0.40 and 0.10 to the moves GROW, PRUNE, SWAP, and CHANGE respectively. The GROW move randomly selects a terminal node and proposes a binary split with probability of (4.7) while its reverse counterpart PRUNE move randomly selects and collapses a pair of
terminal node parameters originating from the same parent node. The CHANGE move randomly selects a non-terminal node and changes the splitting variable and value. It affects terminal nodes that are descendants of the node where CHANGE move is applied. However, this move does not change the number of terminal and non-terminal nodes. The SWAP move interchanges the splitting rule of a parent and child non-terminal nodes.

For illustrative purposes, we elaborate on the calculation of the ratio

\[ \frac{p(W|X, V, T^*_j, \phi)}{p(W|X, V, T_j, \phi)}, \tag{4.17} \]

which is a component of (4.16). For the fittingly named GROW move, when a terminal node with \( n_p \) observation splits to left and right nodes of size \( n_l \) and \( n_r \) (the subscripts \( p, l \) and \( r \) denoting “parent”, and “left” and “right” child nodes), (4.17) simplifies through cancellations since a GROW move only affects the terminal node that is being split. That is,

\[ \frac{p(W|X, V, T^*_j, \phi)}{p(W|X, V, T_j, \phi)} = \frac{p(W_l|X_l, V_l, T^*_j, \phi)p(W_r|X_r, V_r, T^*_j, \phi)}{p(W_p|X_p, V_p, T_j, \phi)} \tag{4.18} \]

which equals

\[
\frac{\varphi}{\bigr(\varphi^2 + \sigma^2 B_p\bigr) \times \exp\left\{ \frac{\sigma^2}{2\varphi} \left( \frac{A_r^2}{\varphi^2 + \sigma^2 B_r} + \frac{A_l^2}{\varphi^2 + \sigma^2 B_l} - \frac{A_p^2}{\varphi^2 + \sigma^2 B_p} \right) \right\}} \]

where \( B_k = \sum_{i=1}^{n_k} \nu_{kl}^1 \) and \( A_k = \sum_{i=1}^{n_k} \omega_{kl} \nu_{kl}^1 \) whose dependence on \( V_k \) and \( W_k \) is suppressed for conciseness. Given an updated tree \( T_j \), its terminal node parameters \( M_j = (\mu_{jk}; k = 1, \ldots, m_j) \) are updated by drawing from \( p(\mu_{jk}|T_j, V, \phi, W, X) \) which upto a proportionality constant is given by

\[ \exp\left\{ -\frac{1}{2} \left( \frac{\varphi^2 + \sigma^2 \sum_{i=1}^{n_k} \nu_{kl}^1}{\varphi^2 + \sigma^2 \varphi} \right) \left[ \mu_{jk} - \frac{\sigma^2 \sum_{i=1}^{n_k} \omega_{kl} \nu_{kl}^1}{\varphi^2 + \sigma^2 \sum_{i=1}^{n_k} \nu_{kl}^1} \right]^2 \right\}, \tag{4.19} \]

indicating a sample from a Gaussian distribution.
In order to update the scale parameter $\phi$, we revert to the original notation of the quantile sum of trees in (4.5), then draw from Inverse-Gamma distribution

$$
p(\phi|M, T, Y, X, V) \propto \phi^{-\frac{n}{2} - \frac{\alpha}{2} - 1} \exp \left\{ -\frac{1}{\phi} \left[ \frac{\beta}{2} + \frac{\sum_{i=1}^{n} (y_i - G(x_i) - \vartheta_1 \nu_i)^2}{2 \vartheta_2^2 \nu_i} \right] \right\}.
$$

(4.20)

4.5 Data analysis

4.6 Simulation study

We now conduct two simulation studies. The first uses the function $f : \mathbb{R}^{10} \to \mathbb{R}$ given by

$$
f(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + 0(x_6 + x_7 + x_8 + x_9 + x_{10}),
$$

where $x_j \sim \text{Unif}(0, 1)$ for $j = 1, \ldots, 10$. This benchmark data generating function is used in Friedman (1991); Chipman et al. (2010); Gramacy and Lee (2012) among others. The response variable is simulated as $y = f(x) + \epsilon$, where $\epsilon \overset{d}{=} \pi \epsilon_1 + (1 - \pi) \epsilon_2$, $\pi \sim \text{Bern}(0.8)$, $\epsilon_1 \sim \text{N}(0, 1)$ and $\epsilon_2 \sim \text{N}(1, 4)$. Note that it includes non-linear, linear, interaction effects, as well as predictors that do not affect the response variable. The model evaluation metric used is the mean weighted absolute deviation given by

$$
\text{MWAD} = \frac{1}{n} \sum_{i=1}^{n} \rho_{\tau}(\hat{y}_i - y_i),
$$

(4.21)

where $\rho_{\tau}(\omega) = \omega (\tau - I\{\omega < 0\})$ is the “check” loss function, $\hat{y}_i$ is the estimated conditional $\tau$th quantile and $y_i$ is the actual response value of the $i$th observation in the evaluation data set. Twenty replications of training and test data sets of 100 observations each are simulated and test data set performance comparisons for Bayesian quantile additive regression trees (BayesQArt), Bayesian quantile regression with adaptive Lasso regularization (BayesQR.AL) in Alhamzawi et al. (2012); Li et al. (2010), linear regression quantiles with adaptive Lasso regularization (QReg.AL) in Zou and Yuan (2008) and quantile random forest (QRF) in Meinshausen (2006) are reported. Our proposed method shows very good predictive performance with lower mean weighted absolute deviation than the competing procedures in estimating
Table 4.1 First quantile regression simulation study results: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses over 20 replications.

<table>
<thead>
<tr>
<th>Method</th>
<th>(\tau = 0.25)</th>
<th>(\tau = 0.50)</th>
<th>(\tau = 0.75)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesQArt</td>
<td>0.7190 (0.0243)</td>
<td>0.9236 (0.0228)</td>
<td>0.6795 (0.0170)</td>
</tr>
<tr>
<td>QRF</td>
<td>0.9215 (0.0228)</td>
<td>1.1430 (0.0243)</td>
<td>1.0123 (0.0171)</td>
</tr>
<tr>
<td>BayesQR.AL</td>
<td>0.8577 (0.0065)</td>
<td>1.0274 (0.0069)</td>
<td>0.8298 (0.0083)</td>
</tr>
<tr>
<td>QReg.AL</td>
<td>0.8395 (0.0194)</td>
<td>1.0132 (0.0211)</td>
<td>0.8130 (0.0151)</td>
</tr>
</tbody>
</table>

Figure 4.2 Predicted conditional quantiles against the actual response for the first simulation study.

the 25\(^{th}\), 50\(^{th}\) and 75\(^{th}\) conditional quantiles as displayed in Table 4.1, underscoring its robustness to the presence of intricate relationships between predictors and the dependent variable. Figure 4.2 displays the predicted conditional quantiles against the actual response values and Table 4.2 displays the selected tuning parameters for each fold.

In the second simulation study, a data set with 30 predictors of which 10 do not impact the response in any form is generated. The data generating scheme is based
Table 4.2  Selected tuning parameters for the first simulation study for quantile regression. Tuning parameters $n_0 = 5$, $d = 10$ are used as the number of minimum observations in a terminal node and the maximum depth of each regression tree in the sum. The number of posterior samples is 30,000 with the first 10,000 discarded, and the number of trees in the sum is equal to 50.

<table>
<thead>
<tr>
<th>Fold</th>
<th>$\tau = 0.25$</th>
<th>$\tau = 0.50$</th>
<th>$\tau = 0.75$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\psi_1$ $\psi_2$ Test MWAD</td>
<td>$\psi_1$ $\psi_2$ Test MWAD</td>
<td>$\psi_1$ $\psi_2$ Test MWAD</td>
</tr>
<tr>
<td>1</td>
<td>0.85 2 0.6509</td>
<td>0.85 2 0.8716</td>
<td>0.90 3 0.6477</td>
</tr>
<tr>
<td>2</td>
<td>0.85 2 0.6088</td>
<td>0.90 3 0.8574</td>
<td>0.90 2 0.6857</td>
</tr>
<tr>
<td>3</td>
<td>0.90 3 0.8074</td>
<td>0.85 3 1.0364</td>
<td>0.85 3 0.7109</td>
</tr>
<tr>
<td>4</td>
<td>0.85 3 0.7507</td>
<td>0.90 2 0.8317</td>
<td>0.90 3 0.5634</td>
</tr>
<tr>
<td>5</td>
<td>0.90 3 0.6207</td>
<td>0.85 3 0.8488</td>
<td>0.85 3 0.5599</td>
</tr>
<tr>
<td>6</td>
<td>0.85 3 0.6524</td>
<td>0.85 3 0.8474</td>
<td>0.85 3 0.6176</td>
</tr>
<tr>
<td>7</td>
<td>0.90 2 0.6545</td>
<td>0.85 2 0.9868</td>
<td>0.85 3 0.7372</td>
</tr>
<tr>
<td>8</td>
<td>0.90 3 0.7937</td>
<td>0.85 2 0.7915</td>
<td>0.85 3 0.6822</td>
</tr>
<tr>
<td>9</td>
<td>0.90 2 0.7162</td>
<td>0.85 3 0.8445</td>
<td>0.90 3 0.6741</td>
</tr>
<tr>
<td>10</td>
<td>0.90 2 0.8369</td>
<td>0.90 3 0.9686</td>
<td>0.90 2 0.5975</td>
</tr>
<tr>
<td>11</td>
<td>0.90 3 0.7053</td>
<td>0.90 2 0.7708</td>
<td>0.90 2 0.5954</td>
</tr>
<tr>
<td>12</td>
<td>0.90 3 1.0129</td>
<td>0.90 3 1.0676</td>
<td>0.90 2 0.7612</td>
</tr>
<tr>
<td>13</td>
<td>0.85 2 0.6506</td>
<td>0.90 3 0.8308</td>
<td>0.85 3 0.6703</td>
</tr>
<tr>
<td>14</td>
<td>0.85 3 0.7243</td>
<td>0.85 3 0.9562</td>
<td>0.90 3 0.7861</td>
</tr>
<tr>
<td>15</td>
<td>0.90 2 0.6283</td>
<td>0.90 3 0.7602</td>
<td>0.85 2 0.6993</td>
</tr>
<tr>
<td>16</td>
<td>0.90 2 0.7706</td>
<td>0.90 3 1.0654</td>
<td>0.85 2 0.8529</td>
</tr>
<tr>
<td>17</td>
<td>0.90 3 0.7164</td>
<td>0.85 3 0.9411</td>
<td>0.85 3 0.7159</td>
</tr>
<tr>
<td>18</td>
<td>0.90 2 0.5406</td>
<td>0.90 2 0.7607</td>
<td>0.90 3 0.5791</td>
</tr>
<tr>
<td>19</td>
<td>0.90 2 0.6734</td>
<td>0.90 2 1.0316</td>
<td>0.90 3 0.6506</td>
</tr>
<tr>
<td>20</td>
<td>0.85 2 0.5866</td>
<td>0.90 2 0.8893</td>
<td>0.85 3 0.6515</td>
</tr>
</tbody>
</table>

on the heteroskedastic error model in He (1997)

$$y = x^T \beta + (x^T \gamma) \epsilon,$$

(4.22)

where $\epsilon \sim N(0, 1)$, $x \in \mathbb{R}^{30}$, $\beta = \begin{pmatrix} 1_{20 \times 1} \\ 0_{10 \times 1} \end{pmatrix}$, $\gamma = \begin{pmatrix} 1_{5 \times 1} \\ 0_{25 \times 1} \end{pmatrix}$, and $1_{m \times 1}$ and $0_{m \times 1}$ denoting column vectors of ones and zeros of size $m$. Each component of $x$ is generated independently from Unif$(0, 1)$. The result of this simulation study for estimation of 25th, 50th and 75th conditional quantiles is reported in Table 4.3 on twenty replications of training and test data sets of 100 observations each. The selected tuning parameters for each fold is displayed in Table 4.4. For the estimation of the 50th
Table 4.3 Second quantile regression simulation study results: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses based on 20 replications.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \tau = 0.25 )</th>
<th>( \tau = 0.50 )</th>
<th>( \tau = 0.75 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesQArt</td>
<td>0.4864 (0.0137)</td>
<td>0.5367 (0.0087)</td>
<td>0.3619 (0.0045)</td>
</tr>
<tr>
<td>QRF</td>
<td>0.4601 (0.0179)</td>
<td>0.6327 (0.0166)</td>
<td>0.5143 (0.0067)</td>
</tr>
<tr>
<td>BayesQR.AL</td>
<td>0.7601 (0.0042)</td>
<td>0.5083 (0.0028)</td>
<td>0.2549 (0.0014)</td>
</tr>
<tr>
<td>QReg.AL</td>
<td>0.7624 (0.0042)</td>
<td>0.5082 (0.0028)</td>
<td>0.2541 (0.0014)</td>
</tr>
</tbody>
</table>

and 75\textsuperscript{th} conditional quantiles, the linear models have better performance than our method or quantile random forest. This is expected given that the underlying data generating process assumes a linear relationship between the predictors and the dependent variable. Our method performs well showing better results than quantile random forests for the estimation of the 50\textsuperscript{th} and 75\textsuperscript{th} conditional quantiles.

4.7 Real data examples

The first data used for illustrating Bayesian quantile additive regression trees is the \texttt{airquality} data from the \texttt{R} package \texttt{datasets}. This data set records the ozone levels (in parts per billion) in New York from May to September 1973. The predictors used are a measure of solar radiation level, wind speed, maximum daily temperature, and month and day of measurement. We estimate the 25\textsuperscript{th}, 50\textsuperscript{th}, and 75\textsuperscript{th} conditional quantile ozone level using the competing statistical procedures in Section 4.6 and Bayesian quantile additive regression trees. After removing observations with missing records, we split the data into five nearly equal partitions. Table 4.5 reports the mean weighted absolute deviations and standard errors, and Table 4.6 records the selected tuning parameters for each fold.
Table 4.4  Selected tuning parameters for the second simulation study for quantile regression. Tuning parameters \( n_0 \) denotes the number of minimum observations in a terminal node. The maximum depth allowed for any tree in the sum is 3 and the value of \( \psi_2 \) used is 3.0. The number of posterior samples is 30,000 with the first 10,000 discarded, and the number of trees in the sum is equal to 200.

<table>
<thead>
<tr>
<th>Fold</th>
<th>( \tau = 0.25 )</th>
<th>( \tau = 0.50 )</th>
<th>( \tau = 0.75 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n_0 ) ( \psi_1 ) Test MWAD</td>
<td>( n_0 ) ( \psi_1 ) Test MWAD</td>
<td>( n_0 ) ( \psi_1 ) Test MWAD</td>
</tr>
<tr>
<td>1</td>
<td>20 0.90 0.5386</td>
<td>30 0.85 0.5222</td>
<td>30 0.90 0.3663</td>
</tr>
<tr>
<td>2</td>
<td>10 0.90 0.5755</td>
<td>20 0.90 0.5533</td>
<td>20 0.90 0.3756</td>
</tr>
<tr>
<td>3</td>
<td>10 0.90 0.5542</td>
<td>10 0.85 0.5746</td>
<td>20 0.85 0.3859</td>
</tr>
<tr>
<td>4</td>
<td>10 0.85 0.5243</td>
<td>30 0.90 0.6174</td>
<td>10 0.90 0.3873</td>
</tr>
<tr>
<td>5</td>
<td>20 0.90 0.4380</td>
<td>20 0.90 0.5054</td>
<td>20 0.90 0.3523</td>
</tr>
<tr>
<td>6</td>
<td>20 0.90 0.3857</td>
<td>20 0.85 0.4833</td>
<td>20 0.90 0.3579</td>
</tr>
<tr>
<td>7</td>
<td>30 0.85 0.5900</td>
<td>20 0.90 0.5820</td>
<td>20 0.90 0.3909</td>
</tr>
<tr>
<td>8</td>
<td>30 0.90 0.4739</td>
<td>20 0.90 0.5250</td>
<td>30 0.90 0.3521</td>
</tr>
<tr>
<td>9</td>
<td>30 0.90 0.4707</td>
<td>30 0.85 0.5531</td>
<td>20 0.85 0.3573</td>
</tr>
<tr>
<td>10</td>
<td>20 0.90 0.4534</td>
<td>30 0.85 0.5504</td>
<td>30 0.90 0.3678</td>
</tr>
<tr>
<td>11</td>
<td>30 0.85 0.4932</td>
<td>20 0.90 0.5153</td>
<td>10 0.90 0.3694</td>
</tr>
<tr>
<td>12</td>
<td>20 0.90 0.5594</td>
<td>20 0.90 0.6037</td>
<td>10 0.85 0.3913</td>
</tr>
<tr>
<td>13</td>
<td>20 0.90 0.4185</td>
<td>30 0.90 0.4957</td>
<td>20 0.90 0.3113</td>
</tr>
<tr>
<td>14</td>
<td>30 0.85 0.4900</td>
<td>10 0.90 0.5209</td>
<td>10 0.90 0.3772</td>
</tr>
<tr>
<td>15</td>
<td>10 0.90 0.3918</td>
<td>20 0.90 0.4914</td>
<td>30 0.85 0.3546</td>
</tr>
<tr>
<td>16</td>
<td>30 0.90 0.5095</td>
<td>30 0.90 0.5253</td>
<td>30 0.85 0.3391</td>
</tr>
<tr>
<td>17</td>
<td>30 0.85 0.4753</td>
<td>20 0.90 0.5300</td>
<td>20 0.90 0.3607</td>
</tr>
<tr>
<td>18</td>
<td>10 0.85 0.4931</td>
<td>20 0.90 0.5507</td>
<td>10 0.90 0.3520</td>
</tr>
<tr>
<td>19</td>
<td>20 0.90 0.3886</td>
<td>20 0.90 0.4753</td>
<td>20 0.90 0.3456</td>
</tr>
<tr>
<td>20</td>
<td>30 0.90 0.5040</td>
<td>20 0.90 0.5592</td>
<td>30 0.90 0.3425</td>
</tr>
</tbody>
</table>

Table 4.5  Ozone data set: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses based on 5 consecutive splits of the data as they appear in the \( R \) package \textit{datasets}.

<table>
<thead>
<tr>
<th></th>
<th>( \tau = 0.25 )</th>
<th>( \tau = 0.50 )</th>
<th>( \tau = 0.75 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesQArt</td>
<td>4.7855 (0.4967)</td>
<td>6.7465 (0.5402)</td>
<td>6.0362 (0.7636)</td>
</tr>
<tr>
<td>QRF</td>
<td>4.4835 (0.7453)</td>
<td>6.4043 (0.7731)</td>
<td>5.7270 (0.6917)</td>
</tr>
<tr>
<td>BayesQR.AL</td>
<td>5.9865 (0.4938)</td>
<td>7.6987 (0.6479)</td>
<td>7.5316 (0.7150)</td>
</tr>
<tr>
<td>QReg.AL</td>
<td>6.0649 (0.5171)</td>
<td>8.7799 (0.3412)</td>
<td>8.0962 (0.3104)</td>
</tr>
</tbody>
</table>
Table 4.6  Selected tuning parameters for the Ozone data quantile additive regression trees real data example.

<table>
<thead>
<tr>
<th>Fold</th>
<th>$n_0$</th>
<th>$d$</th>
<th>$\psi_1$</th>
<th>$\beta$</th>
<th>Test MWAD</th>
<th>$n_0$</th>
<th>$d$</th>
<th>$\psi_1$</th>
<th>$\beta$</th>
<th>Test MWAD</th>
<th>$n_0$</th>
<th>$d$</th>
<th>$\psi_1$</th>
<th>$\beta$</th>
<th>Test MWAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>3</td>
<td>0.99</td>
<td>2</td>
<td>3.8141</td>
<td>10</td>
<td>3</td>
<td>0.99</td>
<td>3</td>
<td>5.7566</td>
<td>5</td>
<td>10</td>
<td>0.99</td>
<td>3</td>
<td>4.3613</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0.90</td>
<td>2</td>
<td>5.1634</td>
<td>5</td>
<td>10</td>
<td>0.90</td>
<td>3</td>
<td>7.2889</td>
<td>5</td>
<td>10</td>
<td>0.95</td>
<td>3</td>
<td>6.0770</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>5</td>
<td>0.90</td>
<td>2</td>
<td>4.7468</td>
<td>20</td>
<td>10</td>
<td>0.90</td>
<td>3</td>
<td>5.6361</td>
<td>5</td>
<td>3</td>
<td>0.99</td>
<td>2</td>
<td>5.4546</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>3</td>
<td>0.99</td>
<td>2</td>
<td>6.4519</td>
<td>5</td>
<td>3</td>
<td>0.99</td>
<td>2</td>
<td>8.5540</td>
<td>5</td>
<td>5</td>
<td>0.95</td>
<td>3</td>
<td>8.8850</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>5</td>
<td>0.90</td>
<td>2</td>
<td>3.7511</td>
<td>20</td>
<td>3</td>
<td>0.9</td>
<td>2</td>
<td>6.4968</td>
<td>20</td>
<td>3</td>
<td>0.95</td>
<td>3</td>
<td>5.4031</td>
</tr>
</tbody>
</table>
The second real data set considered is the auto insurance data consisting of 2,812 auto insurance policyholders with 56 predictors along with an aggregate paid claim amount. This data set is available in the \textit{R} package \textit{HDtweedie} (Qian et al., 2015). Examples of the predictors are driver’s age, driver’s income, use of vehicle (commercial or not), vehicle type (either of 6 categories), and driver’s gender. The response variable is the aggregate claim amount and it is skewed with substantial policyholders having zero claims. When the claim amounts are non-zero, larger claim amounts tend to be reported.

Insurers are often interested in understanding the distribution of claim amounts conditional on a set of policyholder and policy characteristics with added emphasis on higher quantiles. At the time of sale of an insurance policy, whether the policyholder will experience claims is not known. If a future claim were to occur, its amount is also not known at the time of sale. Hence, insurers use estimates of future claims to appropriately price the insurance product and also to set aside sufficient amount of monetary reserves to pay future claims. Thus, we estimate the $90^{th}$ and $95^{th}$ conditional quantiles by splitting the data set into 10 nearly equal partitions each time using nine-tenth of the data for training and the remaining for testing Bayesian quantile additive regression trees and the statistical procedures in Section 4.6. Table 4.7 displays the predictive performances of each procedure and our method performs very well and Table 4.8 records the selected tuning parameters for each of the folds. Note that we are intentionally using the regularized versions of the procedures Bayesian linear quantile regression and the classical quantile regression since variable selection is a component of these procedures.

4.8 Binary classification extension

In this section, we extend Bayesian quantile additive regression trees to tackle binary classification problems. Kordas (2006); Benoit and Van den Poel (2012); Benoit
Table 4.7  Auto insurance claims data set: test data average mean weighted absolute deviations (MWADs) (4.21) and standard errors in parentheses based on 10 splits.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\tau = 0.90$</th>
<th>$\tau = 0.95$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesQArt</td>
<td>1.4487 (0.0690)</td>
<td>1.0440 (0.0571)</td>
</tr>
<tr>
<td>QRF</td>
<td>1.4862 (0.0676)</td>
<td>1.0656 (0.0522)</td>
</tr>
<tr>
<td>BayesQR.AL</td>
<td>1.4508 (0.0681)</td>
<td>1.0483 (0.0602)</td>
</tr>
<tr>
<td>QReg.AL</td>
<td>1.4542 (0.0671)</td>
<td>1.0559 (0.0603)</td>
</tr>
</tbody>
</table>

Table 4.8  Selected tuning parameters for the Auto insurance claims data quantile additive regression trees real data example.

<table>
<thead>
<tr>
<th>Fold</th>
<th>$n_0$</th>
<th>$\psi_1$</th>
<th>Test WMAD</th>
<th>$n_0$</th>
<th>$\psi_1$</th>
<th>Test WMAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>300</td>
<td>0.90</td>
<td>1.5994</td>
<td>100</td>
<td>0.95</td>
<td>1.2181</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>0.95</td>
<td>1.4960</td>
<td>300</td>
<td>0.95</td>
<td>1.0197</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>0.90</td>
<td>1.3919</td>
<td>300</td>
<td>0.95</td>
<td>0.9585</td>
</tr>
<tr>
<td>4</td>
<td>300</td>
<td>0.95</td>
<td>1.0042</td>
<td>300</td>
<td>0.90</td>
<td>0.6687</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>0.90</td>
<td>1.5854</td>
<td>50</td>
<td>0.90</td>
<td>1.0844</td>
</tr>
<tr>
<td>6</td>
<td>200</td>
<td>0.95</td>
<td>1.5794</td>
<td>200</td>
<td>0.95</td>
<td>1.1922</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>0.95</td>
<td>1.1075</td>
<td>300</td>
<td>0.90</td>
<td>0.8403</td>
</tr>
<tr>
<td>8</td>
<td>300</td>
<td>0.90</td>
<td>1.6124</td>
<td>300</td>
<td>0.95</td>
<td>1.2285</td>
</tr>
<tr>
<td>9</td>
<td>300</td>
<td>0.90</td>
<td>1.5405</td>
<td>300</td>
<td>0.95</td>
<td>1.1662</td>
</tr>
<tr>
<td>10</td>
<td>200</td>
<td>0.90</td>
<td>1.5701</td>
<td>300</td>
<td>0.95</td>
<td>1.0637</td>
</tr>
</tbody>
</table>

et al. (2013) among others consider the binary classification problem in a quantile regression framework. Suppose $y_i \in \{0, 1\}$ and $x_i \in \mathbb{R}^d$ are the binary response and the predictors for the $i^{th}$ observation. Suppose also that there is an unobserved latent variable $\tilde{y}_i$ for $i = 1, \ldots, n$ such that $y_i = 1$ if $\tilde{y}_i > 0$ and $y_i = 0$ otherwise. The goal of the classification problem is to obtain an estimate $\hat{P}(y_i = 1 \mid x_i)$ for $P(y_i = 1 \mid x_i)$. 
which we obtain via the hierarchical model

\[ y_i | \tilde{y}_i, \nu_i, T, M \sim \text{Bern} \left( P \left( \tilde{y}_i > 0 \mid \nu_i, T, M \right) \right) \]

\[ \tilde{y}_i \mid \nu_i, T, M \sim N \left( G (x_i; T, M) + \vartheta_1 \nu_i, \vartheta_2^2 \phi \nu_i \right) \]

\[ G (x_i; T, M) = \sum_{j=1}^{n_T} g (x_i; T_j, M_j) \quad (4.23) \]

\[ \nu_i \mid \phi \sim \text{Exp} (\phi), \]

\[ z_i \sim N (0, 1), \]

where \( G (x_i; T, M) \) is a sum of regression trees. The prior specifications for \( T, M \) are as specified in (4.5).

The posterior computation cycles through the following four steps:

(i) draws from \( p (\tilde{y}_i \mid y_i, x_i, \nu_i, T, M, \phi) \), for \( i = 1, \ldots, n \);

(ii) \( n \) sequential draws from \( p (\nu_i \mid T, M, \tilde{y}_i, y_i, \phi) \);

(iii) \( n_T \) sequential draws from \( p \left( (T_j, M_j) \mid M_{(-j)}, T_{(-j)}, \phi, X, Y, \tilde{Y} \right) \), for \( j = 1, \ldots, n_T \),

where \( M_{(-j)} \) and \( T_{(-j)} \) denote the \( n_T \) regression trees in the sum excluding the \( j^{th} \) tree; and

(iv) a draw from \( p \left( \phi \mid M, T, X, Y, \tilde{Y} \right) \).

The first step involves, for \( i = 1, \ldots, n \), a draw from truncated normal distribution

\[ \tilde{y}_i \mid y_i, x_i, \nu_i, T, M, \phi \sim N \left( G (x_i; T, M) + \vartheta_1 \nu_i, \vartheta_2^2 \phi \nu_i \right) I (y_i = 1, \tilde{y}_i \geq 0) \]

\[ + N \left( G (x_i; T, M) + \vartheta_1 \nu_i, \vartheta_2^2 \phi \nu_i \right) I (y_i = 0, \tilde{y}_i < 0). \quad (4.24) \]

The steps (ii) to (iv) are as described in Section 4.4 with \( \tilde{Y} = (\tilde{y}_1, \ldots, \tilde{y}_n)^T \) considered as the response vector of the Bayesian quantile additive regression model.

### 4.9 Binary classification simulation study

We simulate a binary classification data set with ten predictors using the data generating scheme known as “cicle” from the R package mlbench (Leisch and Dimitriadou,
Table 4.9 Simulation study for binary classification: test data averages of test data classification error rate, area under the ROC curve (AUC), and their standard errors in parentheses based on 20 replications.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Error Rate</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesQArt</td>
<td>0.2185 (0.0100)</td>
<td>0.8297 (0.0088)</td>
</tr>
<tr>
<td>RF</td>
<td>0.2600 (0.0112)</td>
<td>0.6363 (0.0130)</td>
</tr>
<tr>
<td>BayesQR.AL</td>
<td>0.5500 (0.0133)</td>
<td>0.4684 (0.0078)</td>
</tr>
</tbody>
</table>

2010) which has often been considered as a benchmark classification data set (Chung and Kim, 2015; Ishwaran, 2015; Rudnicki et al., 2015). Suppose \( x \in [-1,1]^d \), with \( x_j \sim \text{Unif}(-1,1), \ j = 1,\ldots,d \), where we take \( d = 10 \). The goal of this classification problem is to identify if the coordinate \((x_1,\ldots,x_d)\) in a \( d \)-dimensional hypercube with edges at all sign permutations of the coordinates \( \{\pm 1,\ldots,\pm 1\} \) lies outside of a hypersphere which lies inside the hypercube. That is, \( y = 1 \) if \( \sum_{j=1}^{d} x_j^2 > r^2 \), otherwise \( y = 0 \). The radius of the hypersphere, \( r \), is chosen so that there is nearly equal representation between the two classes. The class boundaries are non-linear, making it an interesting classification problem (see Figure 4.3 which shows the class boundary for the two dimensional case).

We simulate training and test data sets of size 100 each and report the averages of classification error rate and area under the ROC curve over twenty replications for binary Bayesian quantile additive regression trees (BayesQArt), binary Bayesian linear quantile regression (BayesQR), and random forests (RF). Note that the random forest procedure used for classification in this section is one described in Breiman (2001) and not the quantile random forest in Meinshausen (2006). For an evaluation data set with \( m \) observations, classification error rate is computed as

\[
\text{Error Rate} = \frac{1}{m} \sum_{i=1}^{m} I(y_i \neq \hat{y}_i),
\]

where \( I(\cdot) \) denotes the indicator function.
Figure 4.3 Circle in square - two dimensional illustration of class boundaries of the binary classification simulation study for Bayesian quantile additive regression trees.

4.10 Real data for binary classification

We consider a binary classification real data example in which the number of predictors is much larger than the sample size to illustrate the predictive performance of the binary extension of Bayesian quantile additive regression trees. The goal for this data set is to classify whether a patient has cancer (ovarian or prostate cancer) based on 10,000 predictors of which a portion is mass-spectra data and the other portion consisting of unimportant predictors. This data set is obtained from a data set named “arcene” at the UCI machine learning repository (Bache and Lichman, 2013). The training and validation data sets contain 200 patient samples. Additional details on this data set are in Guyon et al. (2007, 2008). We split the data into five nearly equal partitions and report the averages of test data classification error rate and area under the ROC curve for binary Bayesian quantile additive regression trees (BayesQArt) and random forests (RF). Results in Table 4.10 show that our proposed method handles regression problems in which the number of predictors is much larger than the number samples in the training data while exhibiting very good predictive
Table 4.10  Cancer classification results: test data averages of classification error rate, area under the ROC curve (AUC), and their standard errors.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Error Rate</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayesQArt</td>
<td>0.1650 (0.0170)</td>
<td>0.8712 (0.0165)</td>
</tr>
<tr>
<td>RF</td>
<td>0.1800 (0.0094)</td>
<td>0.8184 (0.0110)</td>
</tr>
</tbody>
</table>

Table 4.11  Cancer classification results: tuning parameters selected via cross validation. The number of trees used is 200 and a value of 3.0 is used for $\kappa$, and $\psi_2$. We have limited the depth of each tree to $d$, and the minimum number of observations allowed in each terminal node to $n_0$. Posterior sample size of 10,000 is used with the first 3,000 discarded.

<table>
<thead>
<tr>
<th>Fold</th>
<th>$d$</th>
<th>$n_0$</th>
<th>$\tau$</th>
<th>$\psi_1$</th>
<th>Error Rate</th>
<th>Test AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>30</td>
<td>0.5</td>
<td>0.99</td>
<td>0.1250</td>
<td>0.8951</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>20</td>
<td>0.5</td>
<td>0.95</td>
<td>0.2000</td>
<td>0.8827</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>50</td>
<td>0.5</td>
<td>0.95</td>
<td>0.2000</td>
<td>0.8207</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>30</td>
<td>0.5</td>
<td>0.95</td>
<td>0.1750</td>
<td>0.8465</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>30</td>
<td>0.5</td>
<td>0.99</td>
<td>0.1250</td>
<td>0.9110</td>
</tr>
</tbody>
</table>

performance. The selected tuning parameters for each fold is displayed in Table 4.11.

For this example, we only used the moves GROW and PRUNE to reduce the computational cost. An average computing time of 5.42 minutes is recorded using a 64-bit Windows personal computer with the specifications: Intel Core i5-2320, 3.0GHz and 6.0GB installed memory.

4.11 CONCLUSION

This chapter proposed a Bayesian sum of regression trees model for estimating conditional quantiles. The asymmetric Laplace distribution likelihood is employed with its mixture representation enabling tractable posterior computation of the regression trees in the sum and their terminal node parameters.

Simulation studies with data generating schemes that included linear, non-linear, interaction effects, as well as unimportant predictors, illustrated that Bayesian quantile additive regression trees has very good predictive performance. Real data appli-
cations dealing with insurance claims and ozone level prediction demonstrated that the proposed method complements existing powerful statistical procedures.

We also successfully extended and tested the proposed procedure to tackle binary classification problems. The proposed method exhibited very good out-of-sample classification accuracy in a simulation study characterized by a non-linear class boundary and a cancer classification example in which the number of predictors is about fifty times as much as the number of samples in the training data. The source code for the implementation of our proposed method, and the selected tuning parameters for the simulation studies and real data applications are at https://github.com/bpkindo/bayesqart.
CHAPTER 5

CONCLUSIONS

This dissertation proposed and tested Bayesian ensemble of regression trees for the purpose of multinomial probit and quantile regressions. Through simulation studies and real data applications, the predictive performances of the models proposed were compared to existing statistical procedures yielding competitive results. The flexibility of the models to extract information in the presence of intricate relationship of predictors with response variables was illustrated using simulation studies. Furthermore, some simulation studies were intentionally devised to include unimportant predictors in which our proposed models showed very good predictive performances. In addition, we have provided an R software implementation of our models which can freely be downloaded from the repositories http://www.github.com/bpkindo/mpbart and http://www.github.com/bayesqart.

Future works

Bayesian regression trees and their ensembles have recently gained popularity after the work in Chipman et al. (2010). Given the predictive performance of the multinomial response models and quantile regression models outlined in this dissertation, along with similar findings for binary classification and regression (Chipman et al., 2010; Yu et al., 2010; Kapelner and Bleich, 2013; Zhang and Härdle, 2010; Agarwal et al., 2014), it is natural to ask if such ensembles of Bayesian regression trees yield similar results in other areas of statistical applications. To mention a few, the utility of Bayesian regression trees is not yet explored for the purpose of clustering. Longitudinal data
appear in many application for which the development of Bayesian ensemble of trees is not yet extensively studied. Also, in situations where only partial information is available due to censoring or truncation, the question of whether ensemble of Bayesian trees would yield reasonable and competitive results is not yet extensively explored.
BIBLIOGRAPHY


Thomas M. Cover. Historical comments on nearest neighbor pattern classification. *Citation Classics*, 13, 20, March 1982.


## APPENDIX A

### EXCERT OF LICENSE TO MPBART PUBLISHED IN STAT

<table>
<thead>
<tr>
<th>JOHN WILEY AND SONS LICENSE TERMS AND CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jun 21, 2016</td>
</tr>
</tbody>
</table>

This Agreement between Berakem Kindo ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

<table>
<thead>
<tr>
<th>License Number</th>
<th>3893960873836</th>
</tr>
</thead>
<tbody>
<tr>
<td>License date</td>
<td>Jun 21, 2016</td>
</tr>
<tr>
<td>Licensed Content Publisher</td>
<td>John Wiley and Sons</td>
</tr>
<tr>
<td>Licensed Content Publication</td>
<td>Stat</td>
</tr>
<tr>
<td>Licensed Content Title</td>
<td>Multinomial probit Bayesian additive regression trees</td>
</tr>
<tr>
<td>Licensed Content Author</td>
<td>Berakem P. Kindo, Hao Wang, Edsel A. Pena</td>
</tr>
<tr>
<td>Licensed Content Date</td>
<td>Apr 4, 2016</td>
</tr>
<tr>
<td>Licensed Content Pages</td>
<td>13</td>
</tr>
<tr>
<td>Type of use</td>
<td>Dissertation/Thesis</td>
</tr>
<tr>
<td>Requestor type</td>
<td>Author of this Wiley article</td>
</tr>
<tr>
<td>Format</td>
<td>Print and electronic</td>
</tr>
<tr>
<td>Portion</td>
<td>Full article</td>
</tr>
<tr>
<td>Will you be translating?</td>
<td>No</td>
</tr>
<tr>
<td>Title of your thesis / dissertation</td>
<td>Bayesian ensemble of regression trees for multinomial probit and quantile regression</td>
</tr>
<tr>
<td>Expected completion date</td>
<td>Dec 2016</td>
</tr>
<tr>
<td>Expected size (number of pages)</td>
<td>100</td>
</tr>
<tr>
<td>Requestor Location</td>
<td>Berakem Kindo 7009 Shadow Ponce</td>
</tr>
</tbody>
</table>

SELLERSBURG, IN 47172
United States
Attn: Berakem Kindo

Publisher Tax ID | EUE8626007151 |
Billing Type | Invoice |
Billing Address | Berakem Kindo 7009 Shadow Ponce |

SELLERSBURG, IN 47172
United States
Attn: Berakem Kindo

Total 0.00 USD

### TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a "Wiley Company") or handled on behalf of a society with
## APPENDIX B

### SOFTWARE DOCUMENTATION FOR THE R PACKAGE

**MPBART - Multinomial Probit Bayesian Additive Regression Trees**

---

### Package ‘mpbart’

February 7, 2016

**Title**  
Multinomial Probit Bayesian Additive Regression Trees

**Version**  
0.2

**Description**  
Fit Multinomial Probit Bayesian Additive Regression Trees.

**Depends**  
R (>= 3.2.2), mlbench, bayesm, cvTools, mlogit

**License**  
GPL (>= 2)

**LazyData**  
true

**RoxygenNote**  
5.0.1

**NeedsCompilation**  
yes

**Author**  
Bereket Kindo [aut, cre]

**Maintainer**  
Bereket Kindo <bkkindo@gmail.com>

**Repository**  
CRAN

**Date/Publication**  
2016-02-07 12:39:20

### R topics documented:

<table>
<thead>
<tr>
<th>R packages</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpbart</td>
<td>1</td>
</tr>
<tr>
<td>rmpbart</td>
<td>5</td>
</tr>
</tbody>
</table>

### Index

**mpbart**  
Multinomial Probit Bayesian Additive Regression Trees

**Description**  
Multinomial probit modeling using Bayesian Additive Regression Trees,

**Usage**

```r
mpbart(formula, train.data, test.data = NULL, base = NULL, varying = NULL, sep = ".", Prior = NULL, Wcalc = NULL, seedvalue = NULL)
```
Arguments

- **formula**: response ~ choice specific covariates | demographic covariates. If there are no demographic variables use response ~ choice specific covariates | ~ 1. If there are no choice specific covariates, use response ~ 1 | demographic covariates.

- **train.data**: Training Data in wide format (for details on wide format, see documentation in R package `mlogit`).

- **test.data**: Test Data in wide format, typically without the response.

- **base**: base choice. Default is the highest class/choice.

- **varying**: The indices of the variables that are alternative specific.

- **sep**: The separator of the variable name and the alternative name in the choice specific covariates. For example a covariate name variable1.choice1 indicates a separator of dot (.)

- **Prior**: List of Priors for MPBART: e.g., `Prior = list(nu=p+2, V= diag(p - 1), ntrees=200, kfac=2.0, pbd=1.0, pb=0.5, beta = 2.0, alpha = 0.95, nc = 100, priorindep = FALSE, minobsnode = 10)`. The components of Prior are:
  - **nu**: "Mcmc"

- **Mcmc**: List of MCMC starting values, burn-in: e.g., `list(sigma0 = diag(p - 1), keep = 1, burn = 100, ndraws = 1000, keep_sigma_draws=FALSE)`

- **seedvalue**: random seed value, default of 99 will be used if null.

Value

class_prob_train: training data choice/class probabilities,

predicted_class_train: training data predicted choices/classes,

class_prob_test: test data choice/class probabilities,

predicted_class_test: test data predicted choices/classes,

sigma_sample: posterior samples of the latent variable covariance matrix.

Examples

```r
# Not run: library(mpbart)
set.seed(9)
data(Fishing)
table(Fishing$mode)
folds = cvFolds(n = nrow(Fishing), K = 10, R = 1, type = "random")
Fishing$fold = sample(folds$which)
Fishing$logincome = log(Fishing$income)
FishingTrain <- Fishing[which(Fishing$fold != 1),]
FishingTest <- Fishing[which(Fishing$fold == 1),]
burn <- 100
ndraws <- 200 # a higher number such as 1500 is better
```
```r
p = 4
# four choices
sigma0 <- diag(p-1)

Mcmc1 <- list(sigma0=sigma0, burn = burn, ndraws = ndraws)
Prior1 <- list( nu=p-1,
               V = .5*diag(p-1),
               ntrees = 5, # ntrees >= 50 is probably more appropriate
               kfac = 3.0,
               phi = 1.0,
               pb = 0.5,
               alpha = 0.95,
               beta = 3.0,
               nc = 100,
               priorindep = FALSE,
               minobsnode = 10)

out <- mpbart(as.factor(mode) ~ price + catch | logincome,
               train.data = FishingTrain,
               test.data = FishingTest,
               varying = 2:9,
               sep = ".",
               Prior = Prior1,
               Mcmc = Mcmc1,
               seedvalue = 99)

table(as.character(FishingTrain$mode), as.character(out$predicted_class_train))
table(as.character(FishingTest$mode), as.character(out$predicted_class_test))
test_err <- sum(as.character(FishingTest$mode) != as.character(out$predicted_class_test))/length(FishingTest$mode)
cat("test error : ", test_err)
```
```r
# V=(p+2)*diag(p-1),
# ntrees = 100,
# kfac = 2.0,
# pb = 1.0,
# ph = 0.5,
# alpha = 0.99,
# beta = 2.0,
# nc = 200,
# priorindep = FALSE)
#
# out <- mpbart(as.factor(y) ~ 1 | .,
# train.data = traindata,
# test.data = testdata,
# base = NULL,
# varying = NULL,
# sep = NULL,
# Prior = Prior1,
# Mnc = Mnc1,
# seedvalue = 99)
#
# # The above output can alternatively be obtained via:
# out <- mpbart(as.factor(y) ~ 1 | X1 + X2 + X3 + X4 + X5 + X6 +
# X7 + X8 + X9 + X11 + X12 + X13 +
# X14 + X15 + X16 + X17 + X18 + X19 +
# X20 + X21,
# train.data = traindata,
# test.data = testdata,
# base = NULL,
# varying = NULL,
# sep = NULL,
# Prior = Prior1,
# Mnc = Mnc1,
# seedvalue = 99)
#
# confusion matrix train
# table(traindata$y, out$predicted_class_train)
# table(traindata$y==out$predicted_class_train)/
# sum(table(traindata$y==out$predicted_class_train))
#
# confusion matrix test
# table(testdata$y, out$predicted_class_test)
# test_err <- sum(testdata$y != out$predicted_class_test)/
# sum(table(testdata$y == out$predicted_class_test))
# cat("test error : ", test_err )
# Not run: END
```
Multinomial Probit Bayesian Additive Regression Trees

Description
A function to implement multinomial probit regression via Bayesian Addition Regression Trees using partial marginal data augmentation.

Usage
```r
rmpbart(x.train, y.train, x.test = NULL, Prior = NULL, Mcmc = NULL, seedvalue = NULL)
```

Arguments
- `x.train`: Training data predictors.
- `y.train`: Training data observed classes.
- `x.test`: Test data predictors.
- `Prior`: List of Priors for MPBART: e.g., `Prior = list(nu=p+2, V=diag(p-1), ntrees=200, kfac=2.0, pb=1.0, Vbeta=2.0, alpha=0.95, n_tree_pos=100, minsample=10)`
- `Mcmc`: List of MCMC starting values, burn-in... e.g., `Mcmc = list(sigma0 = diag(p-1), keep = 1, burn = 100, ndraws = 1000, keep_sigma_draws=FALSE)`
- `seedvalue`: random seed value: e.g., `seedvalue = 99`

Examples
```r
set.seed(64)
library(mpbart)
p=3
train_wave = mlbench.waveform(50)
test_wave = mlbench.waveform(100)
traindata = data.frame(train_wave$x, y = train_wave$classes)
testdata = data.frame(test_wave$x, y = test_wave$classes)
x.train = data.frame(train_wave$x)
x.test = data.frame(test_wave$x)
y.train = train_wave$classes
sigma0 = diag(p-1)
burn = 100
ndraws = 200 # a higher number >=1000 is more appropriate.
Mcmc1=list(sigma0=sigma0, burn = burn, ndraws = ndraws)
Prior1 = list(nu=p+2,
  V=diag(p-1), ntrees=200)
```
V=(p+2)*diag(p-1),
ntrees = 5, #typically 200 trees is good
kfac = 2.0,
phd = 1.0,
ph = 0.5,
alpha = 0.99,
beta = 2.0,
nc = 200,
priorindep = FALSE)

out = rmpbart(x.train = x.train, y.train = y.train, x.test = x.test,
Prior = Prior, Mcmc=Mcmc, seedvalue = 99)

#confusion matrix train
table(y.train, out$predicted_class_train)
table(y.train==out$predicted_class_train)/sum(table(y.train==out$predicted_class_train))

#confusion matrix test
table(test_wave$classes, out$predicted_class_test)
test_err <- sum(test_wave$classes != out$predicted_class_test)/
sum(table(test_wave$classes == out$predicted_class_test))
cat("test error : ", test_err )
Appendix C

Software Documentation for Bayesian Quantile Additive Regression Trees

Package ‘BayesQArt’

Type Package
Title Bayesian quantile additive regression trees
Version 1.0
Date 2016-03-01
Author Bereket Kindo
Maintainer Bereket Kindo <bpkindo@gmail.com>
Description Fits Bayesian quantile additive regression trees
License GPL>=2
Imports Rcpp (>= 0.12.3)
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 5.0.1

R topics documented:

BayesQArt ................................................................. 1
BayesQArt_binary ..................................................... 3
do_rpg ................................................................. 5
wmad ................................................................. 5

Index 7

BayesQArt Fit Bayesian quantile additive regression trees for a continuous response

Description
Fit Bayesian quantile additive regression trees for a continuous response

Usage
BayesQArt(y, X, Xtest, quantile, burn, nd, m, min_obs_node, aa_parm, bb_parm, nc, pbd, gb, alpha, betap, kappa, maxdepth)
BayesQArt

Arguments

- \( y \) response variable,
- \( X \) a matrix of training covariates,
- \( X_{\text{test}} \) a matrix containing test data covariates,
- quantile the quantile value.
- burn number of initial posterior draws to be discarded,
- nd number of posterior draws,
- \( m \) number of trees in the sum,
- \( \min_{\text{obs node}} \) minimum number of observations needed in a terminal node,
- \( \alpha_{\text{parms}} \) the \( \alpha \) parameter for the prior distribution \( \phi \sim \text{IG}(\alpha/2, \beta/2) \),
- \( \beta_{\text{parms}} \) the \( \beta \) parameter for the prior distribution \( \phi \sim \text{IG}(\alpha/2, \beta/2) \),
- \( \text{nc} \) number of cutpoints potentially available for splitting a tree,
- \( \text{pbd} \) probability of birth/prune move,
- \( \text{pb} \) probability of birth given a birth/prune move,
- \( \text{alpha} \) parameter defining probability of splitting node \( \alpha (1+\text{d})\beta \),
- \( \text{betap} \) parameter defining probability of splitting node \( \alpha (1+\text{d})\beta \),
- \( \kappa \) parameter defining the prior on the terminal node parameters \( \mu_{jk} \sim N(0, \sigma^2_0) \),
- \( \text{maxdepth} \) maximum depth a tree can grow.

Value

- \( \text{pred}_{\text{train}} \) training data prediction,
- \( \text{pred}_{\text{test}} \) test data prediction.

Examples

```r
f = function(x){
  10*sin(pi*x[,1]*x[,2]) + 20*(x[,3]-.5)^2+10*x[,4]+5*x[,5]
}

n = 100;
p = 10;
np = 100;
x=matrix(runif(n*p),n,p)
xp=matrix(runif(np*p),np,p)
fy = f(x)
fpy = f(xp)
pis = rbinom(n,1,.8);
y=fy+pis*rnorm(n) + (1-pis)*rnorm(n,mean = 3,sd = 3);
set.seed(99)
out = BayesQArt(y=y,X=x, Xtest = xp, quantile = .5,
burn = 3000,
```
BayesQArt_binary

nd = 5000, 
m = 200, 
min.obs.node= 5, 
aa_parm = 2.0, 
bb_parm = 3.0, 
nc=50, 
plot=4, 
plot=0.5, 
alpha=0.95, 
beta=2.0, 
kappa = 2.0, 
maxdept = 3)

# actual vs. predicted plot
# training data
plot(fy, out$pred_train)
abline(0,1)

# testing data
plot(fpy, out$pred_test)
abline(0,1)

wmad(fpy,out$pred_test,0.5)

BayesQArt_binary 3

Implements Bayesian quantile additive regression trees for binary classification problems.

Description
Implements Bayesian quantile additive regression trees for binary classification problems.

Usage
BayesQArt_binary(y, X, Xtest, quantile, burn, nd, m, min.obs.node, aa_parm, 
bb_parm, nc, pbd, pb, alpha, betap, kappa, maxdepth)

Arguments
y the binary response,
X the training matrix of covariates,
Xtest the test data matrix of covariates,
quantile the quantile value.
burn number of initial posterior draws to be discarded,
nd number of posterior draws,
m number of trees in the sum,
min.obs.node minimum number of observations needed in a terminal node,
BayesQArt_binary

aa_parm the α parameter for the prior distribution $\phi \sim IG(\alpha/2, \beta/2)$,
bb_parm the β parameter for the prior distribution $\phi \sim IG(\alpha/2, \beta/2)$,
nc number of cutpoints potentially available for splitting a tree,
pbd probability of birth/prune move,
pb probability of birth given a birth/prune move,
alpha parameter defining probability of splitting node $\alpha(1 + d)^\beta$,
betap parameter defining probability of splitting node $\alpha(1 + d)^\beta$,
kappa parameter defining the prior on the terminal node parameters $\mu_{jk} \sim N(0, \sigma^2)$,
maxdepth maximum depth a tree can grow.
double quantile,

Value

pred_train training data average prediction of the latent variable,
pred_test test data average prediction of the latent variable.

Examples

library(mlbench)
library(AUC)
set.seed(54)
traindat <- mlbench.circle(200, 10)
testdat <- mlbench.circle(200, 10)
y <- as.numeric(traindat$classes) - 1
table(y)
x <- traindat$x
yp <- as.numeric(testdat$classes) - 1
table(yp)
xp <- testdat$x

binary_bayesqart <- BayesQArt_binary(y = y,
            X = x,
            Xtest = xp,
            quantile = 0.5,
            burn = 1000,
            nd = 2000,
            m = 200,
            min_obs_node = 20,
            aa_parm = 2.0,
            bb_parm = 2.0,
            nc = 100,
            pbd = 0.6,
            pb = 0.5,
            alpha = 0.95,
            betap=3.0,
            kappa = 3.0,
            maxdepth = 5)
```r
eps = 1e-15;
binary_bayesqart$pred_train = sapply( binary_bayesqart$pred_train, function(x) max(eps,x))
binary_bayesqart$pred_train = sapply( binary_bayesqart$pred_train, function(x) min(1-eps,x))
binary_bayesqart$pred_test = sapply( binary_bayesqart$pred_test, function(x) max(eps,x))
binary_bayesqart$pred_test = sapply( binary_bayesqart$pred_test, function(x) min(1-eps,x))

train_error <- sum(y != (binary_bayesqart$pred_train > 0.5))/sum(table(y == (binary_bayesqart$pred_train > 0.5)))
test_error <- sum(yp != (binary_bayesqart$pred_test > 0.5))/sum(table(yp == (binary_bayesqart$pred_test > 0.5)))
cat("test error rate: ", test_error)

train_auc <- auc(roc(binary_bayesqart$pred_train, as.factor(y)))
test_auc <- auc(roc(binary_bayesqart$pred_test, as.factor(yp)))
cat("test AUC: ", test_auc)
```

### do_rgig

Generate Generalized Inverse Gaussian random variable with density proportional to
\[ x^{(\lambda - 1)} \exp \left\{ \frac{-\chi}{x} - \psi x \right\} / 2 \]
satisfying \( \lambda > 0, \psi > 0, \chi := 0; \lambda = 0, \psi > 0, \chi > 0; \lambda < 0, \psi > 0, \chi > 0. \)

#### Description

Generate Generalized Inverse Gaussian random variable with density proportional to \( x^{(\lambda - 1)} \exp \left\{ \frac{-\chi}{x} - \psi x \right\} / 2 \)
satisfying \( \lambda > 0, \psi > 0, \chi := 0; \lambda = 0, \psi > 0, \chi > 0; \lambda < 0, \psi > 0, \chi > 0. \)

#### Usage

```r
do_rgig(lambda, chi, psi)
```

#### Arguments

- `lambda`: the shape parameter.
- `chi`: scale parameter.
- `psi`: scale parameter.

### wmad

A function to calculate weighted mean absolute deviation.

#### Description

A function to calculate weighted mean absolute deviation.

#### Usage

```r
wmad(y_true, y_hat, quant)
```
Arguments

y_true    actual response,
y_hat     conditional quantile prediction,
quart     quantile value.

Value

weighted mean absolute deviation, wmad.