Gov 2018 Problem Set 2: Bayesian Additive Regression Trees (Solutions)

Your name here:

Friday February 25, 2022

Please upload answers to all questions (including computational questions and any related graphics and R code *with comments*, in .rmd *and* knitted pdf forms to Gradescope by Friday March 4, 2022 at 11:59 pm. For any problems that require calculations, please show your work. Finally, when doing simulations or draws from a random distribution in `R`, please set your seed immediately using `set.seed(2022)`.

Introduction

This problem is based off of Montgomery, Jacob M., Santiago Olivella, Joshua D. Potter, and Brian F. Crisp. 2015. "An Informed Forensics Approach to Detecting Vote Irregularities". *Political Analysis*.

This paper proposes an *informed forensics* approach to identifying fraud in a large cross-national setting.

- Data: a cross-national data set spanning seventy countries and six decades containing three sets of variables.
  1. Outcome (Y_i): a proxy measure of electoral fraud constructed with the NELDA data set (Hyde and Marinov 2012) using Item Response Theory (IRT) model
  2. Predictor set X_i^f: *forensic indicators* of anomalous vote distributions
  3. Predictor set X_i^c: *contextual risk factors* that have been identified in past research as increasing the likelihood of fraud.
- Method: The authors fit a Bayesian additive regression trees (BART) model developed by Chipman, George and McCulloch (2010) to combine forensic indicators and contextual risk factors.

Overview of data

	Name(s)	Description
Descriptive	`uid`	Unique id for observation $i \in \{1, \ldots, 598\}$.
	`country`	Name of country of observation i.
	`year`	Year of observation i.
Outcome	`fraud.score`	A proxy measure of electoral fraud (continuous).
Forensic indicators	`Uniform.LastDigit`	Test statistic of relative frequencies of numbers in the last digit (uniform violation).
	`DistanceLastPair`	Distance of last two digits.
	`Mean.SecondDigit`	Test statistic of relative frequencies of numbers in second digit (mean).
	`Benford.SecondDigit`	Second digit (Benford violation).
Contextual risk factors	`inequality.(var-name)`	Economic inequality (Gini coefficient). Indicator variables for each of quartiles {[20.1,27.3], (27.3,32.7], (32.7,43], (43,60.1]} and `NA`.

Name(s)	Description
`regime.(var-name)`	Average score from prior election, discretized into regime type. Indicator variables for each of {New Democracy, Old Democracy, Anocracy, Autocracy} and `NA`.
`avgmag.(var-name)`	Mean of all districts' magnitudes. Indicator variables for each of quartiles {[1,1.03], (1.03,6.17], (6.17,11.1], (11.1,150]} and `NA`.
`gdpchange.(var-name)`	Change in GDP per capita in the prior year. Indicator variables for each of quartiles {[-32.1,1.81], (1.81,3.56], (3.56,5.58], (5.58,34.8]} and `NA`.
`commision.(var-name)`	Level of independence of electoral commission (government/mixed/independent). Indicator variables for each of {0, 1, 2}.
`instability.0`	Indicator variable for political instability.
`fract`	Fearon's index of ethnic fractionalization.
`urban`	Percent of population living in urban centers.
`turnout`	Percent registered voters casting ballots in national election.

Note that the descriptive variables are not included in the model. See Table B1 of Appendix B for more details of each variable.

A quick review of ensemble methods for trees

Recall that there exist various methods to combine a set of tree models (*ensemble methods*), to produce a powerful committee using the outputs of many weak classifiers.

- Boosting: fit a sequence of single trees, using each tree to fit data variation not explained by earlier trees in the sequence, and then combine these trees to correct bias.
- Bagging (= bootstrap aggregation): fit a large number of independent trees for each of bootstrap samples, and average these noisy trees to reduce variance.
- Random forests: a modification of bagging that averages a large collection of de-correlated trees created with a random selection of the input variables.

In this problem set, we will use a Bayesian additive regression trees (BART) model, a sum-of-trees model motivated by a boosting algorithm. BART is a Bayesian approach for a sum of trees where each tree explains a small and different portion by imposing a regularization prior (see Chipman, George and McCulloch 2010 for more details). According to the authors, "one crucial advantage of using BART is that Bayesian estimation techniques allow the model to produce measures of uncertainty regarding not only the model's parameters (which are often not of direct interest), but also of more usual quantities of interest, such as the partial dependence of the outcome of interest across any one combination of covariates. (Montgomery et al. 2010, Appendix F)'' In light of this advantage, the authors answer the following question along with its uncertainty (e.g., 80% credible intervals): which particular variables are most important in allowing the fitted BART model to identify electoral fraud (as captured by our NELDA-based proxy)?

Throughout the questions, we will use the data from `forensic.RData`, and a set of tuning parameters that the original paper has used. Note that your result may look somewhat different from the results from the paper due to the random split of training and test data and the use of a different package — the authors used `BayesTree` package for their analysis while we will use `BART`, a recently published package that is relatively computationally efficient (see this link for more details). **Bonus points for efforts to tune the parameters for better performance.**

Question 1

In this question, we will fit a continuous BART model with forensic indicators.

(a) Load the data, set the random seed and randomly split the data into two sets: a training set with 500 observations, and test set with 98 observations.

(b) Using `fraud.score` as the outcome variable, and forensic indicators as predictors (a total 4 variables), fit the BART model with your training set using a call to `wbart()`.

(c) Check the convergence with `plot(your-fitted-model$sigma, type = "l")`.

(d) Create a scatter plot of your test set where the y-axis is the outcome variable (`fraud.score`) and the x-axis is its predictions using the fitted model. Add a 45 degree line as a reference.

(e) Report the following statistics (RMSE, MAE, MAPE, MAD, MEAPE) of the fitted model with test set.

Let $e_i = \mid \hat{y}_i - y_i \mid$ denote the absolute error, and $a_i = \frac{e_i}{|y_i|} \times 100$ denote the absolute percentage error where y_i is the true outcome value and $\hat{y}_i$ is the prediction.

- Root mean squared error (RMSE) $= \sqrt{\frac{\sum_i e_i^2}{n}}$
- Mean absolute error (MAE) $= \frac{\sum_i e_i}{n}$
- Mean absolute proportional error (MAPE) $= \frac{\sum_i a_i}{n}$
- Median absolute deviation (MAD) $= \text{med}(\mathbf{e})$
- Median absolute proportional error (MEAPE) $= \text{med}(\mathbf{a})$

Note that n here would be the number of observations in the test set ($= 98$), $\mathbf{e} = (e_1, \ldots, e_n)$, and $\mathbf{a} = (a_1, \ldots, a_n)$

Hint: Sample codes are as below.

```
## (a)
seed.num = 2022
load("forensic.RData")
set.seed(seed.num)
# TODO: randomly split the data into two sets

# 80% of the data to train
train = sample_frac(Hitters, 0.8)

# 20% to test
test = setdiff(Hitters, train)

# Making X's using same function as before
x_train = model.matrix(Salary~., train)[,-1]
x_test = model.matrix(Salary~., test)[,-1]

# Making Y's
y_train = train$Salary
y_test = test$Salary

## (b)
library(BART)
set.seed(seed.num)
# TODO: specify the arguments (x.train, y.train, x.test)
your-fitted-model <- wbart(x.train = NULL,
                           y.train = NULL,
                           x.test = NULL,
```

```
                            k = 2, power = 0.5, base = 0.95, # parameters from the paper
                            sigdf = 10, sigquant = 0.75, nskip = 50000,
                            ndpost = 5000, ntree = 100, sigest = 0.35,
                            keepevery = 10, printevery = 2000)

## (c)
# TODO: specify your model
plot(your-fitted-model$sigma, type = "l")
abline(v = 50000, lwd = 2, col = "red")

## (d)
# TODO: create a scatter plot

## (e)
# Create a function for computing the test statistics
stats.model <- function(true.y, predict.y) {
  if (length(true.y)!=length(predict.y)) stop("Check the inputs.")
  # TODO: specify the following objects
  n = NULL
  e = NULL
  a = NULL
  rmse = NULL
  mae = NULL
  mape = NULL
  mad = NULL
  meape = NULL
  res = c(rmse, mae, mape, mad, meape)
  names(res) = c("rmse", "mae", "mape", "mad", "meape")
  return(res)
}

# TODO: specify the arguments
stats.model(your-test-outcome, your-fitted-model$yhat.test.mean)
```

Question 2

Repeat (b) - (e) with two different models: a contextual risk factors model (total 27 predictors) and informed forensics model (a total 31 predictors). You may use the same tuning parameters as before. Compare the statistics of the three models and briefly discuss the results in words.

Question 3

Now, we will investigate a potential concern with these results: "given that roughly five out of six of our observations take on the lowest fraud score, these fit statistics may simply be capturing the models' accurate prediction of cases with no obvious fraudulent activities (496).'' (A case of *high specificity but low sensitivity*)

(a) Create a histogram of the outcome variable of your test data and check the distribution. Specifically, how many observations in your test data have the lowest fraud score ($= -0.4062667$)?

(b) Create a binary version of the outcome variable (`fraud.score`) with the threshold at 0 for both of your train and test set.

(c) Fit a probit BART using this binary outcomes with informed forensics model (a total 31 predictors).

(d) Create a 2x2 table of the predicted test labels and true test labels. What does this tell you about the aforementioned concern?

Hint: Sample codes for (c) are as below.

```
## (c)
set.seed(seed.num)
# TODO: specify the arguments (x.train, y.train, x.test)
binary.mod <- pbart(x.train = NULL,
                    y.train = NULL,
                    x.test = NULL,
                    k = 2, power = 0.5, base = 0.95,
                    nskip = 50000, ndpost = 5000, ntree = 100,
                    keepevery = 10, printevery = 2000)
```

Question 4

Which particular variables are most important in allowing the fitted informed forensic model to identify electoral fraud? We will answer this question using the following approach from the paper.

"When the number of trees is low, BART tends to rely on the most relevant predictors when building trees, as predictors are forced to compete to improve the model's fit (Chipman, George, and McCulloch 2010). As a result, we can re-estimate the BART model using a small number of trees (viz. 10) and calculate the average share of splitting rules that involve each variable (497).''

(a) Fit the informed forensics model (total 31 predictors) using the entire data set.

(b) Replicate Figure 2 using the item `varcount` from the fitted output. Briefly discuss the results.

Hint:

- `varcount`: a matrix with ndpost rows and nrow(x.train) columns. Each row is for a draw. For each variable (corresponding to the columns), the total count of the number of times that variable is used in a tree decision rule (over all trees) is given.
- Sample codes are as below.

```
## (a)
set.seed(seed.num)
# TODO: specify the arguments (x.train, y.train)
bart.full.mod <- wbart(x.train = NULL,
                       y.train = NULL,
                       k = 2, power = 0.5, base = 0.95,
                       sigdf = 10, sigquant = 0.75, nskip = 50000,
                       ndpost = 5000, ntree = 100, sigest = 0.35,
                       keepevery = 10, printevery = 2000)

## (b)
## Get nr. of variables (a.k.a. features) used
numVars <- length(colMeans(bart.full.mod$varcount))

## Calculate relative frequency of feature usage
## in splitting rules, and sort
values <- colMeans(prop.table(bart.full.mod$varcount,1))
ordering <- order(values)

## More legible names to variables (TODO: you may need to change the order)
names <- c("Uniform last digit", "Distance last pair", "Mean 2nd digit",
           "Benford 2nd digit", "Ethnolinguistic fract.",  "Urbanization",
           "Turnout", "Regime type (anocracy)", "Regime type (autocracy)",
           "Regime type (missing)", "Regime type (new democracy)",
```

```
          "Regime  type (old democracy)", "Inequality (1st quart.)",
          "Inequality (2nd quart.)", "Inequality (3rd quart.)", "Inequality (4th quart.)",
          "Inequality (missing)", "Average district magnitude (1st quart.)",
          "Average district magnitude (2nd quart.)", "Average. district magnitude (3rd quart.)",
          "Average district magnitude (4th quart.)", "Average district magnitude (missing)",
          "Change in GDP (1st quart.)", "Change in GDP (2nd quart.)", "Change in GDP (3rd quart.)",
          "Change in GDP (4th quart.)", "Change in GDP (missing)", "Gov. Electoral Commission",
          "Mixed Electoral Commission", "Indep. Electoral Commission", "Regime Instability")

## Create Graph
par(yaxt="n", mfrow=c(1,1), mar=c(2.5,14,.5,.5), bty="n", mgp=c(1,0,0), tcl=0)
plot(values[ordering], 1:numVars, ylab="", pch=19, main=""
     , xlab="Posterior average inclusion rate"
     ,xlim=c(0.02,0.065))
par(las=1)
mtext(names[ordering], at=1:numVars, cex=.8, side=2, line=.25)
for(i in 1:numVars){
  abline(h=i, lty=2, col="gray80")
}
points(values[ordering], 1:numVars, ylab="", pch=19)
```