Lab7: Tree-based Methods

MAT43 Statistical Machine Learning

Silvia Montagna

5/22/2018

We return to binary regression with the National Election Study data (see Lab 3 for descriptions of some of the variables and initial model fitting).

[*The following code will read in the data. Remove this text and modify the code chunk options so that the code does not appear in the output.*]

```
# Data are at http://www.stat.columbia.edu/~gelman/arm/examples/nes

nes <- read.dta("nes5200_processed_voters_realideo.dta",
                    convert.factors = F)

# Filter data to include year, age, income, female, race, white, black,
# religion, south, state, region, martial_status, party afilliation
# (as in LAB3) and ideology (as in LAB3)

nes_clean = nes %>%
  select(presvote, year, age, income, race, white, black, gender, religion, south, state, regi
  filter(presvote %in% 1:2) %>%
  mutate(female = gender - 1,
         black = as.integer(race == 2),
         vote = as.integer(presvote == 2))

# Remove NA's
nes_clean = nes_clean[!is.na(rowSums(nes_clean)), ] %>%
  select(-presvote) %>%
  select(-gender)

# Convert variables that are coded as numerical as factors (state, region, etc.)
nes_clean = nes_clean %>% mutate(vote = factor(vote),
                                 year = factor(year),
                                 female = factor(female),
                                 race = factor(race),
                                 white = factor(white),
                                 black = factor(black),
                                 religion = factor(religion),
                                 south = factor(south),
                                 state = factor(state),
                                 region = factor(region),
                                 martial_status = factor(martial_status),
                                 partyid3 = factor(partyid3),
```

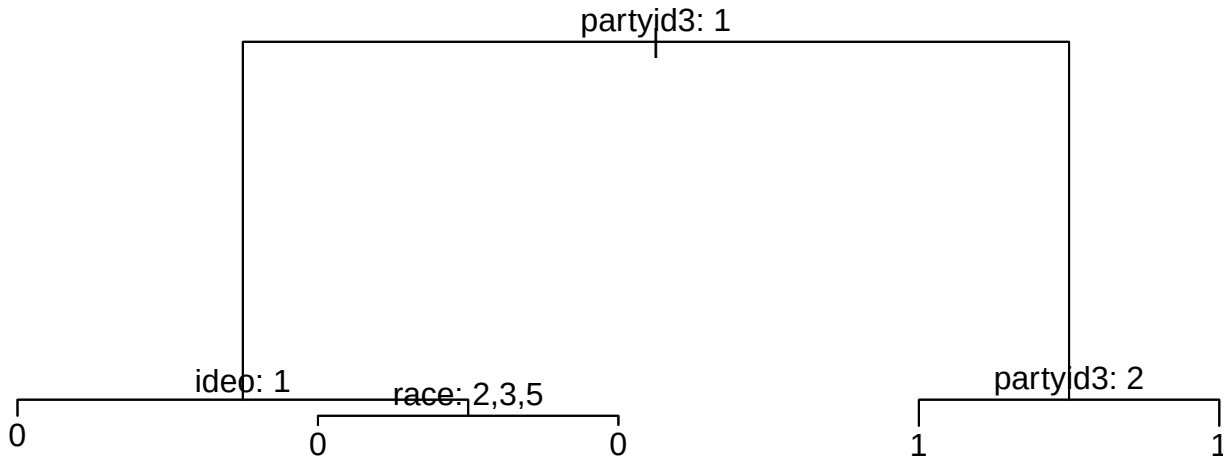
```
                                    ideo = factor(ideo))

# Create a random split 50% sample for test and training
set.seed(42)
df = data.frame(nes_clean)
idx = sample(nrow(df), nrow(df))[1:(nrow(df)/2)]
train = df[idx, ]
test = df[-idx, ]

test = test[test$partyid3 != 9, ]
# Note the variable state has more than 50 levels. You may decide how to handle
# this; i.e., assume 1:50 are US and other territories and use 1:50???
# Just document what you do (states are not sorted alphabetically)
# Discuss how this limits your modeling
```

1. Using the the training data, fit a tree model to the data to predict the probability of voting republican in the election and prune. Comment on the selected tree - which variables are important? Are there interesting interactions? Provide graphics or tables to highlight findings.

```
set.seed(8675309)
library(tree)
tree.vote = tree(vote ~ . -state, data = train)
tree.vote
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 2290 3171.00 0 ( 0.51921 0.48079 )
##    2) partyid3: 1 1188  946.40 0 ( 0.86364 0.13636 )
##      4) ideo: 1 587  270.70 0 ( 0.93867 0.06133 ) *
##      5) ideo: 3,5 601  617.20 0 ( 0.79035 0.20965 )
##       10) race: 2,3,5 179   93.22 0 ( 0.92737 0.07263 ) *
##       11) race: 1,4 422  490.40 0 ( 0.73223 0.26777 ) *
##    3) partyid3: 2,3 1102  923.60 1 ( 0.14791 0.85209 )
##      6) partyid3: 2 142  195.80 1 ( 0.45775 0.54225 ) *
##      7) partyid3: 3 960  632.90 1 ( 0.10208 0.89792 ) *
```

```
plot(tree.vote); text(tree.vote, pretty = 10)
```



2. Using the the training data, fit a random forest model to the data to predict the probability of voting Republican in the election. Comment on the results - which variables are important? What insights does the model provide? Support with graphics if possible.

3. Repeat 2, but using boosting.

4. Repeat 2, but using BART. Comment on any partial dependence plots or other output that is of interest in explaining the model.

5. Using `gam` from `mgcv` fit a generalized additive model to predict the probability of voting Republican using smoothing splines for fitting examining nonlinear functions of the continuous variables. Are there any interactions that you might expect will be important? In `mgcv` you may allow different curves for levels of a factor using the `by` option: `race + s(age, by = race)`. Random intercepts for say `state`, may be obtained via `s(state, bs = "re")`. Using residuals, residual deviance, AIC, or other options find a predictive model that seems to be reasonable for the training data, exploring non-linearity, random intercepts and slopes. Provide a brief description of how you came up with your final model and describe what insights about voting it provides.

6. Using the models from 1-5, determine the error rate for each model for predicting on the test data.

7. Provide a summary of your findings. Your comments should address benefits and advantages for the different methods. Which method has the best predictive accuracy? Which provides the most interpretability or insight into quantifying factors? In explaining your findings and insights provide graphs and tables that help quantify uncertainty and illustrate effects of the different characteristics.