Homework 5: Tree-based Inference

Wen Li Teng

March 1 2020

1. GINI INDEX, CLASSIFICATION ERROR, AND CROSS-ENTROPY

According to James et al. (2013), when growing a decision tree, “either the Gini index or the cross-entropy are typically used to evaluate the quality of a particular split, since these two approaches are more sensitive to node purity than is the classification error rate.” Indeed, “classification error is not sufficiently sensitive for tree-growing, and in practice two other measures are preferable” - the Gini index and the cross-entropy. However, “any of these three approaches might be used when pruning the tree, but the classification error rate is preferable if prediction accuracy of the final pruned tree is the goal.”

2. ESTIMATING THE MODELS

```r
train <- read.csv("data/gss_train.csv")
test <- read.csv("data/gss_test.csv")
set.seed(1234)

library(caret)
library(e1071)
library(MLmetrics)
library(tidyverse)
library(pROC)
library(ranger)

train_control <- trainControl(method = "CV", number = 10)
train$colrac <- as.factor(train$colrac)
test$colrac <- as.factor(test$colrac)
```

Logistic Regression

```r
logit_mod <- train(colrac ~ .,
                   data = train,
                   method = "glm",
                   trControl = train_control)
```

Naive Bayes

```
nb_mod <- train(colrac ~ .,
                  data = train,
                  method = "nb",
                  trControl = train_control)
```

Elastic Net Regression

```
enet_mod <- train(colrac ~ .,
                  data = train,
                  method = "glmnet",
                  trControl = train_control,
                  tuneGrid = expand.grid(alpha = seq(0, 1, 0.1),
                                         lambda = seq(0.001, 0.1, 0.001)))
```

Decision Tree (CART)

```
cart_mod <- train(colrac ~ .,
                  data = train,
                  method = "rpart",
                  trControl = train_control)
```

Bagging

```
bag_mod <- train(colrac ~ .,
                  data = train,
                  method = "treebag",
                  trControl = train_control)
```

Random Forest

```
rfgrid <- expand.grid(
  .mtry = seq(20, 30, by = 2),
  .splitrule = "gini",
  .min.node.size = seq(3, 9, by = 2)
)

rf_mod <- train(colrac ~ .,
                data = train,
                method = "ranger",
                trControl = train_control,
                tuneGrid = rfgrid
                )
```

Boosting

```
boost_grid <-  expand.grid(interaction.depth = c(1, 5, 9),
                        n.trees = (1:30)*50,
                        shrinkage = 0.1,
                        n.minobsinnode = 20)

boost_mod <- train(colrac ~ .,
                  data = train,
                  method = "gbm",
                  trControl = train_control,
                  tuneGrid = boost_grid)
```

3. CROSS-VALIDATED ERROR RATE AND ROC/AUC ON TRAINING SET

Logistic Regression

```
logit_pred <- logit_mod %>% predict(train)
logit_cmat <- confusionMatrix(data = logit_pred, reference = train$colrac)
logit_err <- 1 - logit_cmat$overall['Accuracy']
logit_roc_obj <- roc(as.numeric(train$colrac), as.numeric(logit_pred))
logit_auc <- auc(logit_roc_obj)
```

Naive Bayes

```
nb_pred <- nb_mod %>% predict(train)
nb_cmat <- confusionMatrix(data = nb_pred, reference = train$colrac)
nb_err <- 1 - nb_cmat$overall['Accuracy']
nb_roc_obj <- roc(as.numeric(train$colrac), as.numeric(nb_pred))
nb_auc <- auc(nb_roc_obj)
```

Elastic Net Regression

```
enet_pred <- enet_mod %>% predict(train)
enet_cmat <- confusionMatrix(data = enet_pred, reference = train$colrac)
enet_err <- 1 - enet_cmat$overall['Accuracy']
enet_roc_obj <- roc(as.numeric(train$colrac), as.numeric(enet_pred))
enet_auc <- auc(enet_roc_obj)
```

Decision Tree (CART)

```r
cart_pred <- cart_mod %>% predict(train)
cart_cmat <- confusionMatrix(data = cart_pred, reference = train$colrac)
cart_err <- 1 - cart_cmat$overall['Accuracy']
cart_roc_obj <- roc(as.numeric(train$colrac), as.numeric(cart_pred))
cart_auc <- auc(cart_roc_obj)
```

Bagging

```r
bag_pred <- bag_mod %>% predict(train)
bag_cmat <- confusionMatrix(data = bag_pred, reference = train$colrac)
bag_err <- 1 - bag_cmat$overall['Accuracy']
bag_roc_obj <- roc(as.numeric(train$colrac), as.numeric(bag_pred))
bag_auc <- auc(bag_roc_obj)
```

Random Forest

```r
rf_pred <- rf_mod %>% predict(train)
rf_cmat <- confusionMatrix(data = rf_pred, reference = train$colrac)
rf_err <- 1 - rf_cmat$overall['Accuracy']
rf_roc_obj <- roc(as.numeric(train$colrac), as.numeric(rf_pred))
rf_auc <- auc(rf_roc_obj)
```

Boosting

```r
boost_pred <- boost_mod %>% predict(train)
boost_cmat <- confusionMatrix(data = boost_pred, reference = train$colrac)
boost_err <- 1 - boost_cmat$overall['Accuracy']
boost_roc_obj <- roc(as.numeric(train$colrac), as.numeric(boost_pred))
boost_auc <- auc(boost_roc_obj)
```

Comparison

```r
overall_auc <- c(logit_auc, nb_auc, enet_auc, cart_auc, bag_auc, rf_auc, boost_auc)
overall_err <- c(logit_err, nb_err, enet_err, cart_err, bag_err, rf_err, boost_err)
overall_mod <- c("logit", "naive bayes", "elastic net", "CART",
                 "bagging", "random forest", "boosting")
overall_comp <- cbind(overall_mod, overall_err, overall_auc)
rownames(overall_comp) <- c()

overall_comp %>%
  as.data.frame() %>%
  arrange(overall_err)
```

```
##     overall_mod          overall_err       overall_auc
## 1       bagging 0.000677506775067727 0.999354838709677
## 2 random forest  0.00135501355013545 0.998709677419355
## 3      boosting    0.160569105691057 0.836949058948046
## 4   elastic net    0.180216802168022 0.819124752657494
## 5         logit    0.182926829268293 0.816476002024757
## 6          CART    0.215447154471545 0.780536560673692
## 7   naive bayes    0.257452574525745 0.743124568588652
```

4. CHOICE OF BEST MODEL

Given that bagging results in the lowest overall error and higher overall AUC, bagging is the best model.

5. EVALUATION OF BEST MODEL ON TEST SET

Classification Error Rate

```
bag_pred_new <- bag_mod %>% predict(test)
bag_cmat_new <- confusionMatrix(data = bag_pred_new, reference = test$colrac)
bag_err_new <- 1 - bag_cmat_new$overall['Accuracy']
bag_err_new
```

```
##  Accuracy
## 0.2028398
```

ROC/AUC

```
bag_roc_obj_new <- roc(as.numeric(test$colrac), as.numeric(bag_pred_new))
bag_auc_new <- auc(bag_roc_obj_new)
bag_auc_new
```

```
## Area under the curve: 0.7908
```

Generalizability

```
bag <- c("train", "test")
err <- c(bag_err, bag_err_new)
auc <- c(bag_auc, bag_auc_new)
general <- cbind(bag, err, auc)
rownames(general) <- c()
general
```

```
##      bag     err                    auc
## [1,] "train" "0.000677506775067727" "0.999354838709677"
## [2,] "test"  "0.202839756592292"    "0.7908060244952"
```

The best model (bagging) appears to be of acceptable generalizability. From the train to the test sets, the error rate increased and the area under the curve decreased. Nonetheless, around 80% of the observations still belong to the most common class. Furthermore, this model performs better than a random classifier (which would have a AUC of 0.5). As such, the bagging model can still be generalizable.