HW3-XGBOOST

Santanu Mukherjee

2022-04-13

R Markdown

XGBOOST

The code and results for XGBOOST are displayed below. The findings are:

1. For the same max depth, the RMSE has consistently decreased when correlated predictor has been added.

2. The R-squared hasn't changed much and it has always been in the range of 0.80 - 0.86 even if more correlated predictor has been added.

```
##
# Lets try the same experiment but using boosted trees:

library(mlbench)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
set.seed(200)
simulated <- mlbench.friedman1(200, sd = 1)
simulated <- cbind(simulated$x, simulated$y)
simulated <- as.data.frame(simulated)
colnames(simulated)[ncol(simulated)] <- "y"

library(xgboost)
library(caTools)

simulated$duplicate1 = NULL
simulated$duplicate2 = NULL

ind = createDataPartition(simulated$y, p = 0.8, list=FALSE)
train.df = simulated[ind,]
test.df  = simulated[-ind,]
control.parm = trainControl(method = "CV", number = 10, savePredictions = TRUE, classProbs = TRUE)
param.grid = expand.grid(eta = 0.1, colsample_bytree = c(0.5,0.7), max_depth=c(3,6), nrounds = 100,
                         gamma=1, min_child_weight= 2, subsample = 0.5)


#Model1 XG boost

model1.xgbost = train(y~., data = train.df, method = "xgbTree", trControl = control.parm,
                      tuneGrid = param.grid)
```

```
## Warning in train.default(x, y, weights = w, ...): cannnot compute class
## probabilities for regression
```

```
model1.xgbost
```

```
## eXtreme Gradient Boosting
##
## 160 samples
##  10 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
## Resampling results across tuning parameters:
##
##   max_depth  colsample_bytree  RMSE      Rsquared   MAE
##   3          0.5               2.096792  0.8363651  1.753181
##   3          0.7               1.930588  0.8595791  1.523414
##   6          0.5               2.256791  0.8222974  1.862401
##   6          0.7               2.035353  0.8452091  1.705371
##
## Tuning parameter 'nrounds' was held constant at a value of 100
## Tuning
##  'min_child_weight' was held constant at a value of 2
## Tuning
##  parameter 'subsample' was held constant at a value of 0.5
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nrounds = 100, max_depth = 3, eta
##  = 0.1, gamma = 1, colsample_bytree = 0.7, min_child_weight = 2 and subsample
##  = 0.5.
```

```
## Plot of Important Variables
#summary(model1.xgbost)

# Now we add correlated predictors one at a time - duplicate1
simulated$duplicate1 = simulated$V1 + rnorm(200) * 0.1

ind = createDataPartition(simulated$y, p = 0.8, list=FALSE)
train.df = simulated[ind,]
test.df  = simulated[-ind,]

model2.xgbost = train(y~., data = train.df, method = "xgbTree", trControl = control.parm,
                      tuneGrid = param.grid)
```

```
## Warning in train.default(x, y, weights = w, ...): cannnot compute class
## probabilities for regression
```

```
model2.xgbost
```

```
## eXtreme Gradient Boosting
##
## 160 samples
##  11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
## Resampling results across tuning parameters:
##
##   max_depth  colsample_bytree  RMSE      Rsquared   MAE
##   3          0.5               2.144012  0.8375386  1.688685
##   3          0.7               1.959826  0.8644575  1.577921
##   6          0.5               2.219594  0.8261830  1.771485
##   6          0.7               2.207315  0.8106613  1.756022
##
## Tuning parameter 'nrounds' was held constant at a value of 100
## Tuning
##  'min_child_weight' was held constant at a value of 2
## Tuning
##  parameter 'subsample' was held constant at a value of 0.5
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nrounds = 100, max_depth = 3, eta
##  = 0.1, gamma = 1, colsample_bytree = 0.7, min_child_weight = 2 and subsample
##  = 0.5.
```

```
## Plot of Important Variables
summary(model2.xgbost)
```

```
##               Length Class              Mode
## handle             1 xgb.Booster.handle externalptr
## raw           107560 -none-             raw
## niter              1 -none-             numeric
## call               5 -none-             call
## params             8 -none-             list
## callbacks          1 -none-             list
## feature_names     11 -none-             character
## nfeatures          1 -none-             numeric
## xNames            11 -none-             character
## problemType        1 -none-             character
## tuneValue          7 data.frame         list
## obsLevels          1 -none-             logical
## param              0 -none-             list
```

```
# adding another correlated predictor this time - duplicate2

simulated$duplicate2 = simulated$V1 + rnorm(200) * 0.1

ind = createDataPartition(simulated$y, p = 0.8, list=FALSE)
train.df = simulated[ind,]
test.df  = simulated[-ind,]

model3.xgbost = train(y~., data = train.df, method = "xgbTree", trControl = control.parm,
                      tuneGrid = param.grid)
```

```
## Warning in train.default(x, y, weights = w, ...): cannnot compute class
## probabilities for regression
```

```
model3.xgbost
```

```
## eXtreme Gradient Boosting
##
## 160 samples
##  12 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
## Resampling results across tuning parameters:
##
##   max_depth  colsample_bytree  RMSE      Rsquared   MAE
##   3          0.5               2.100123  0.8330513  1.670128
##   3          0.7               2.007407  0.8468859  1.632185
##   6          0.5               2.252669  0.8010067  1.845682
##   6          0.7               2.069108  0.8394404  1.681876
##
## Tuning parameter 'nrounds' was held constant at a value of 100
## Tuning
##  'min_child_weight' was held constant at a value of 2
## Tuning
##  parameter 'subsample' was held constant at a value of 0.5
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nrounds = 100, max_depth = 3, eta
##  = 0.1, gamma = 1, colsample_bytree = 0.7, min_child_weight = 2 and subsample
##  = 0.5.
```

```
## Plot of Important Variables
summary(model3.xgbost)
```

```
##               Length Class              Mode
## handle             1 xgb.Booster.handle externalptr
## raw           111138 -none-             raw
## niter              1 -none-             numeric
## call               5 -none-             call
## params             8 -none-             list
## callbacks          1 -none-             list
## feature_names     12 -none-             character
## nfeatures          1 -none-             numeric
## xNames            12 -none-             character
## problemType        1 -none-             character
## tuneValue          7 data.frame         list
## obsLevels          1 -none-             logical
## param              0 -none-             list
```