Kaggle Project Report

Stats 202: Data Mining and Analysis

Alex Kim, Suraj Bulchand, Carina Zhang

December 6, 2018

Kaggle Team and Results

Kaggle Team Name: Alex (team mergers not enabled on Kaggle)
Team Members: Alex Kim, Suraj Bulchand, Carina Zhang

Public Leaderboard Score: 0.68333
Private Leaderboard Score: 0.70714

Model Description

Our optimal model was a support vector machine with a radial kernel, trained on all given predictors except for *Id*. More specifically, our optimal model had a *cost* hyperparameter of **1** and a *gamma* hyperparameter of **0.1**.

We reached these values by cross-validating several hyperparameter combinations. Namely, we cross-validated over the following hyperparameter spaces:

$$\text{cost} \in \{0.001, 0.01, 0.1, 1, 1.5, 2, 3\}$$
$$\text{gamma} \in \{0.01, 0.05, 0.1, 0.25, 0.5, 1\}$$

We fiddled somewhat with these ranges to find better local minima, but did not achieve different results than the ranges above.

To estimate the error of our model, we divided the given *train_data.csv* into random training and test sets with a 50:50 ratio. Using this validation approach, we achieved a training error rate of **0.232** and a test error rate of **0.350**, where the error rate is defined as the number of incorrect predictions divided by the number of responses predicted.

Although the predictor space is 10-dimensional (and thus hard to visualize), we hypothesize that the two classes are separated by a roughly ellipsoidal boundary. We hold this hypothesis because each one-dimensional plot of a predictor versus the response shows that the responses are highly mixed, with slightly different ranges. We note that a circular decision boundary has similar projections on each predictor axis. Because the radial SVM kernel performed optimally out of all of our models, we have reason to believe that the true shape is somewhat ellipsoidal.

```
library(tidyverse)
library(e1071)
```

Importing the Data

```r
# Initial import
train_data <- read_csv("data/train_data.csv")
test_data <- read_csv("data/test_data.csv")

# Remove ID's; create ID vector for test
train_data <- train_data[-12]
test_ids <- as.vector(test_data$Id)
test_data <- test_data[-11]

# Modify training data
train_data$Status <- as.factor(if_else(train_data$Status == 1, 1, -1))
```

Dividing Data into Training and Test Sets

```r
set.seed(1)

subtrain_size <- floor(0.5 * nrow(train_data))
subtrain_indexes <- sample(x = 1:nrow(train_data), size = subtrain_size)

subtrain <- train_data[subtrain_indexes,]
subtest <- train_data[-subtrain_indexes,]

rm(subtrain_size, subtrain_indexes)
```

Cross Validation on Radial SVM

```r
set.seed(1)

# Hyperparameter ranges
cost_range <- c(0.001, 0.01, 0.1, 1, 5, 10, 50, 100)
gamma_range <- c(0.01, 0.1, 0.5, 1, 2, 3)

# Model fitting
radial_tune_output <- tune(svm, Status ~ ., data = subtrain, kernel = "radial", ranges = list(cost = cos
best_radial_svm <- radial_tune_output$best.model
best_radial_svm
```

```
##
## Call:
## best.tune(method = svm, train.x = Status ~ ., data = subtrain,
##     ranges = list(cost = cost_range, gamma = gamma_range), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##        gamma:  0.1
##
## Number of Support Vectors:  406
```

```r
# Training error
predictions <- predict(best_radial_svm, subtrain)
mean(predictions != subtrain$Status)
```

```
## [1] 0.232
```

```r
# Test error
predictions <- predict(best_radial_svm, subtest)
mean(predictions != subtest$Status)
```

```
## [1] 0.35
```

Kaggle Submission on SVM Model

```r
set.seed(1)

cost_range <- c(0.001, 0.01, 0.1, 1, 1.5, 2, 3)
gamma_range <- c(0.01, 0.05, 0.1, 0.25, 0.5, 1)

# Train SVM
radial_tune_output <- tune(svm, Status ~ ., data = train_data, kernel = "radial", ranges = list(cost =
best_radial_svm <- radial_tune_output$best.model

radial_tune_output
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     1   0.1
##
## - best performance: 0.323
```

```r
# Prediction
predictions <- predict(best_radial_svm, test_data)

# Generate file for Kaggle
predictions <- if_else(predictions == 1, TRUE, FALSE)
kaggle_submission <- tibble(Id = test_ids, Category = predictions)
write_csv(kaggle_submission, "data/submission.csv")
```