Prediciton on Online News Popularity

Nanjun Wang
nw2359

08 November 2017

Contents

Load libraries

```
library(e1071)
library(mlbench)
library(caret)
library(ggplot2)
library(knitr)
```

Remove redundant varibales

```
load("/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/data/modifieddata.RData")
Y <- as.numeric(newsdata$log_shares)
X <- newsdata[,-c(1,49,50,51)]
categorical_var <- grep("is", names(X))
for(i in 1:length(categorical_var)){
  indicator <- categorical_var[i]
  X[,indicator] <- as.factor(X[,indicator])
}
#X <- scale(X[,-nearZeroVar(X)])
#X <- X[, -findCorrelation(cor(X), .8)]
#X <- as.data.frame(X)
#system.time(svmProfile <- rfe(X, Y,
#                   sizes = 10,
#                   rfeControl = rfeControl(functions = caretFuncs, number = 2),
#                   method = "svmLinear"))


correlationMatrix <- cor(X[,-categorical_var])
# find attributes that are highly corrected (ideally >0.75)
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.5)
# print indexes of highly correlated attributes
sink("/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/output/var_remove.txt")
names(X[,highlyCorrelated])
```

```
##  [1] "n_unique_tokens"            "log_n_tokens_content"
##  [3] "n_non_stop_words"           "data_channel_is_tech"
##  [5] "is_weekend"                 "global_sentiment_polarity"
##  [7] "log_LDA_02"                 "log_LDA_03"
##  [9] "global_rate_negative_words" "sqrt_num_imgs"
## [11] "global_rate_positive_words" "sqrt_kw_min_max"
## [13] "max_positive_polarity"      "data_channel_is_world"
```

```
sink()
X_cor_rm <- as.data.frame(X[,-highlyCorrelated])
```

Rank features by importance

```
#control <- trainControl(method="repeatedcv", number=5, repeats=2)
# train the model
#system.time( model <- train(x= X, y= Y,  method="lm", preProcess="scale", trControl=control, importanc
# estimate variable importance
#importance <- varImp(model, scale=FALSE)
# plot (importance)


categorical_ind <- grep("is", names(X_cor_rm))
ncols <- ncol(X_cor_rm)
col_ind <- 1:ncols
continuous_ind <- col_ind[-categorical_ind]
pearson <- NA
for(i in 1:length(continuous_ind)){
  indicator <- continuous_ind[i]
  pearson[i] <- abs( cor.test(X_cor_rm[,indicator], Y, method = "pearson")$estimate)
}
#pearson_rank <- order(pearson_scores<-unlist(pearson), decreasing = TRUE)
#pearson_sorted <- sort(pearson_scores,decreasing = TRUE)
pearson_df <- data.frame(variables = names(X_cor_rm[,continuous_ind]),
                         scores = pearson)


png("/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/output/varImp.png")
ggplot(data=pearson_df, aes(x=reorder(variables, scores), y=scores)) +
  geom_bar(stat = "identity" ,width = 0.5, color = "steelblue", fill = "steelblue")+coord_flip()
dev.off()
```

```
## pdf
##   2
```

Split Dataset

```
continuous_order <- order( pearson_df$scores, decreasing = TRUE)
continuous_16 <- pearson_df$variables[continuous_order[1:16]]
var_selected <- c(as.character( continuous_16), colnames(X_cor_rm [,categorical_ind]))
X_20 <- X_cor_rm[,var_selected]
mydata <- cbind(X_20, Y)
#smp_size <- floor(0.75*nrow(mydata))
```

```r
#set.seed(123)
#train_ind <- sample(seq_len(nrow(mydata)), size = smp_size)
#train <- mydata[train_ind, ]
#test <- mydata[-train_ind, ]
#write.csv(test, "/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/data/test.csv")
#write.csv(train, "/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/data/train.csv")
train <- read.csv("/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/data/train.csv")
test <- read.csv("/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/data/test.csv")
```

Tunning Support Vector Regression Model

```r
system.time(tuneResult <- tune(svm, Y~. , data = train[1:3000,],
              ranges = list(epsilon = seq(0.1,0.3,0.05), cost = 2^(2:4))
))
```

```
##    user  system elapsed
## 318.101   2.521 323.285
```

```r
print(tuneResult)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  epsilon cost
##      0.3    4
##
## - best performance: 0.8641523
```

```r
png("/Users/ouminamikun/Documents/Temporary/ADA/ADA_Project/output/cv_svm.png")
plot(tuneResult)
dev.off()
```

```
## pdf
##   2
```

Making Prediciton

```r
cate_ind <- grep("is", colnames(train))
for(i in 1:length(cate_ind)){
  indicator <- cate_ind[i]
  train[,indicator] <- as.numeric(train[,indicator])
}
system.time( model_linear <- svm(x= train[,-21], y = train[,21], kernel = "linear"))
```

```
##    user  system elapsed
## 522.321   2.857 526.983
```

```r
system.time( model_RBF <- svm(x= train[,-21], y = train[,21], kernel = "radial"))
```

```
##    user  system elapsed
```

```
## 238.393   2.638 245.510
```

```r
#tunedModel <- tuneResult$best.model

cat_ind <- grep("is", colnames(test))
for(i in 1:length(cat_ind)){
  indicator <- cat_ind[i]
  test[,indicator] <- as.numeric(test[,indicator])
}
pred_linear <- predict(model_linear, test[,-21])
pred_RBF <- predict(model_RBF, test[,-21])
RSS_linear <- sum((test$Y - pred_linear)^2)
RSS_RBF <- sum((test$Y - pred_RBF)^2)
RSS_df <- data.frame(Algorithms = c("SVM Linear", "SVM RBF"),
                     RSS = c(RSS_linear,RSS_RBF),
                     Training_Time = c(557.894,255.026))

kable(RSS_df)
```

Algorithms	RSS	Training_Time
SVM Linear	7417.649	557.894
SVM RBF	7182.982	255.026