PCA and SVM on Movie Title and Overview

April 11, 2017

Here we accomplished several tasks: 1) From the dataframe of 527 Horror movies and 570 Romance movies prepared by python codes (please refer to Milestone_2_Part_1.pdf), we randomly chose 300 movies to be the test set, and the rest of them to be the training set; 2) Using PCA to extract first 150 PCs that explain 80% of the variance in data, and projecting the training data and testing data to get PC score in each sets; 3) Employing SVM with radial basis function to classify the horror and romance movies based on their PC scores. The parameter (gamma and cost) has been found through tuning; 4) The final predicting accuracy on test set using this model is 87%, which is a satisfactory result.

Read in data (data was generated using Python); Create training set and testing set

```
feature <- read.csv("feature.csv")
feature <- feature[,-1]
genre <- read.csv("genre.csv", header = FALSE)
genre <- genre[,-1]

set.seed(1)
index <- sample(1:nrow(feature),300)

test.genre <- genre[index]
test.feature <- feature[index,]

train.genre <- genre[-index]
train.feature <- feature[-index,]
```

Principal Component Analysis using First 150 PCs

```
PCA.train <- prcomp(train.feature)

PCA.vectors <- PCA.train$rotation[,1:150]

PCA.score.train <- PCA.train$x[,1:150]

test.scaled <-
    scale(test.feature, center=PCA.train$center, scale=PCA.train$scale)

PCA.score.test <-
    test.scaled %*% PCA.train$rotation[,1:150]
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.2.5
```

Use Support Vector Machine (Radial Basis Function) to classify Horror movies and Romance movies using PCs

```
train.df <- data.frame(train.genre,PCA.score.train)
colnames(train.df)[1]<- c("Labels")

test.df <- data.frame(test.genre,PCA.score.test)
colnames(test.df)[1]<- c("Labels")

tuned.params <-
    tune(svm, Labels~., kernel="radial", data=train.df, ranges=
             list(gamma=10^(-5:1),cost=10^(-3:2)))

gamma <- tuned.params$best.parameters$gamma
cost <- tuned.params$best.parameters$cost

model <- svm(Labels~., kernel="radial", data=train.df, gamma=gamma, cost=cost)

preds <- predict(model, newdata=test.df)

mean(test.df$Labels==preds)
```

```
## [1] 0.85
```