Poster color decompostion PCA SVM

May 1, 2017

1) Inputs: feature dataframe (953 rows, 1000 columns) contains frequency of 1000 colors (numbers of pixels falling into each RGB value range) of each poster. Genre dataframe (953 rows, 1 column) contains genre label of movies. The dataset has been randomly split into training set (753 posters) and testing set (200 posters). 2) Using PCA to extract first 40 PCs that explain 95% 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, romance and scifi 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 around 65%.

Read in word appearance data of movies (data was generated using Python); Create training set and testing set

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

set.seed(1)
index <- sample(1:nrow(feature),200)

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

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

Principal Component Analysis using First 40 PCs

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

PCA.vectors <- PCA.train$rotation[,1:40]

PCA.score.train <- PCA.train$x[,1:40]

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

PCA.score.test <-
    test.scaled %*% PCA.train$rotation[,1:40]
```

```
library(e1071)
```

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

Use Support Vector Machine (Radial Basis Function) to classify Horror movies, Romance movies and Scifi 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^(-4:0),cost=10^(1:4)))

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)

table(test.df$Labels,preds)
```

```
##          preds
##           horror romance scifi
##   horror      40      12    14
##   romance      9      69     8
##   scifi       15      11    22
```

Accuracy on test set

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

```
## [1] 0.655
```