

```

library(gbm)

## Loading required package: survival
## Loading required package: lattice
## Loading required package: splines
## Loading required package: parallel
## Loaded gbm 2.1.1

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(e1071)
library(caret)

## Loading required package: ggplot2

##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster

# Load feature data
load("C:/Users/sh355/Documents/GitHub/spr2017-proj3-
group10/data/siftFeatures.RData")
# set working directory
setwd("C:/Users/sh355/Documents/GitHub/spr2017-proj3-group10/data")
# Load label
label<-read.csv("labels.csv",header = T)

# transpose feature data
sift_features<-t(sift_features)
y<-as.factor(label$V1)
y1<-label$V1

# set as a dataframe
dat<-data.frame(y,sift_features)
dat2<-data.frame(y1,sift_features)

```

```

names<-c("labels",paste0('Feature',1:5000))
colnames(dat2)<-names

# split the data into testset and trainset
indice<-sample(1:nrow(dat),size=500,replace = F)
testset<-dat[indice,] # Contains 500 observations
testset2<-dat2[indice,]
trainset<-dat[-indice,] # Contains 1500 observations
trainset2<-dat2[-indice,]

source("C:/Users/sh355/Documents/GitHub/spr2017-proj3-group10/lib/train.R")
source("C:/Users/sh355/Documents/GitHub/spr2017-proj3-group10/lib/test.R")

gbm1 = gbm(labels ~ .,data=trainset2,distribution = "adaboost",
            n.trees = 512,
            bag.fraction = 0.5,interaction.depth=3,train.fraction =
1,cv.folds = 5)

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 144: Feature144 has no variation.

## Warning in gbm.fit(x, y, offset = offset, distribution = distribution, w =
## w, : variable 2233: Feature2233 has no variation.

# predict train
trainset2$predicted <-predict(gbm1,as.data.frame(trainset2),
                             n.trees=512,type="response")

# Create Confusion Matrix
confusionMatrix(data=as.numeric(trainset2$predicted> 0.5),
                reference=trainset2$labels,
                positive='1')

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##           0 577 188
##           1 172 563
##
##               Accuracy : 0.76
##               95% CI : (0.7376, 0.7814)
##       No Information Rate : 0.5007
##       P-Value [Acc > NIR] : <2e-16
##
##               Kappa : 0.52
##  Mcnemar's Test P-Value : 0.4292
##
##               Sensitivity : 0.7497
##               Specificity : 0.7704
##               Pos Pred Value : 0.7660

```

```

##          Neg Pred Value : 0.7542
##          Prevalence : 0.5007
##          Detection Rate : 0.3753
##          Detection Prevalence : 0.4900
##          Balanced Accuracy : 0.7600
##
##          'Positive' Class : 1
##

# train accuracy 0.7856

# predict train
testset2$predicted <- predict(gbm1, as.data.frame(testset2),
                             n.trees=512, type="response")

# Create Confusion Matrix
confusionMatrix(data=as.numeric(testset2$predicted > 0.5),
                 reference=testset2$labels,
                 positive='1')

## Confusion Matrix and Statistics
##
##          Reference
## Prediction  0   1
##          0 179  75
##          1  72 174
##
##          Accuracy : 0.706
##          95% CI : (0.6639, 0.7456)
##          No Information Rate : 0.502
##          P-Value [Acc > NIR] : <2e-16
##
##          Kappa : 0.412
##          Mcnemar's Test P-Value : 0.869
##
##          Sensitivity : 0.6988
##          Specificity : 0.7131
##          Pos Pred Value : 0.7073
##          Neg Pred Value : 0.7047
##          Prevalence : 0.4980
##          Detection Rate : 0.3480
##          Detection Prevalence : 0.4920
##          Balanced Accuracy : 0.7060
##
##          'Positive' Class : 1
##

# Train pca+SVM model
train_svm<-train(trainset[, -1], trainset[, 1])

```

```
# SVM training accuracy
train_svm$training_accuracy

## [1] 0.98

# predict test on pca+SVM
result<-test(train_svm$fit, testset[,-1], train_svm$pca_fit)

# accuracy on testset
1-sum((as.numeric(result)-1)!=testset[,1])/500

## [1] 0.812
```