STAT 542: Final Project

Karthik Vasu (kvasu2), Yining Lu (yining13)

Due: 05/05/2022

Contents

```
Xtrain = train[1:6000,2:785]
Ytrain = train[1:6000,1]

Xtest = test[,2:785]
Ytest = test[,1]
```

Summary Statistics

Data table

```
## Ytrain
##   0   1   2   3   4   5   6   7   8   9
## 630 603 586 613 586 592 616 619 573 582
```

```
## Ytest
##    0    1    2    3    4    5    6    7    8    9
## 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
```

Ensemble Model

```
# first stage with svm
library(kernlab)
k = 1
index = list()
svm.fit = list()
mXtrain = as.matrix(Xtrain)
for (i in 1:10) {
```

```r
  index[[i]] = which(Ytrain == i-1)
}
for (i in 1:10) {
    ind = index[[i]]
    y = rep(0, nrow(mXtrain))
    y[ind] = 1
    svm.fit[[i]] = ksvm(mXtrain, y, type="C-svc", kernel="rbfdot",
                        C=10, scaled=c(), prob.model = TRUE)
    #print(i)
}
```

```r
# get probabilities for each class
prob = matrix(NA, nrow=nrow(mXtrain), ncol=10)
for (i in 1:10) {
  prob[,i] = predict(svm.fit[[i]], Xtrain, type="probabilities")[,2]
}
```

```r
# second stage with random forest
set.seed(2)
library(randomForest)
```

```
## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.
```

```r
rf.fit = randomForest(as.factor(Ytrain)~., data=data.frame(prob, Ytrain), proximity=T)
print(rf.fit)
```

```
##
## Call:
##  randomForest(formula = as.factor(Ytrain) ~ ., data = data.frame(prob,      Ytrain), proximity = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.03%
## Confusion matrix:
##     0   1   2   3   4   5   6   7   8   9 class.error
## 0 622   0   0   1   0   0   7   0   0   0 0.012698413
## 1   0 602   0   1   0   0   0   0   0   0 0.001658375
## 2   1   0 572   1   9   0   3   0   0   0 0.023890785
## 3   2   0   0 607   2   0   2   0   0   0 0.009787928
## 4   0   0   5   2 577   0   2   0   0   0 0.015358362
## 5   0   0   0   0   0 591   0   1   0   0 0.001689189
## 6   6   0   8   2   2   0 598   0   0   0 0.029220779
## 7   0   0   0   0   0   0   0 617   0   2 0.003231018
## 8   0   0   0   0   0   0   0   0 573   0 0.000000000
## 9   0   0   0   0   0   0   0   3   0 579 0.005154639
```

```r
prob_pred = matrix(NA, nrow=nrow(Xtest), ncol=10)
for (i in 1:10) {
  prob_pred[,i] = predict(svm.fit[[i]], Xtest, type="probabilities")[,2]
}
```

```r
# error for ensemble model
Ytest_pred_en <- predict(rf.fit, data.frame(prob_pred, Ytest))
# confusion matrix
confuse_en = table(Ytest, Ytest_pred_en)
# mis-classification rate
misclass_rate_en = rep(NA, 10)
number = rep(NA, 10)
for (i in 1:10) {
  misclass_rate_en[i] = 1 - confuse_en[i,i]/sum(confuse_en[,i])
}
class = 0:9
misclass_en = cbind(class, round(misclass_rate_en,3))
# overall mis-classification rate
overall_misclass_en = mean(Ytest_pred_en != Ytest)
confuse_en
```

```
##      Ytest_pred_en
## Ytest   0   1   2   3   4   5   6   7   8   9
##     0 798   5  10  23   5   4 144   0  11   0
##     1   2 980   0   8   2   1   7   0   0   0
##     2  18   4 813   9  62   2  91   0   1   0
##     3  38  28  16 860  29   0  28   0   1   0
##     4   0   4 125  30 754   1  83   0   3   0
##     5   1   0   1   1   0 941   5  27   2  22
##     6 121   4 101  29  63   1 667   0  13   1
##     7   0   0   0   0   0  45   6 898   0  51
##     8   3   0   8   3   2  18  22   5 939   0
##     9   0   0   0   0   0  13   1  37   0 949
```

```r
misclass_en
```

```
##       class
##  [1,]     0 0.187
##  [2,]     1 0.044
##  [3,]     2 0.243
##  [4,]     3 0.107
##  [5,]     4 0.178
##  [6,]     5 0.083
##  [7,]     6 0.367
##  [8,]     7 0.071
##  [9,]     8 0.032
## [10,]     9 0.072
```

```r
overall_misclass_en
```

```
## [1] 0.1401
```

```r
# error for pure svm
Ytest_pred_ksvm = rep(NA, nrow(Xtest))
for (i in 1:nrow(Xtest)) {
  Ytest_pred_ksvm[i] = which.max(prob_pred[i,])-1
}
```

```
# confusion matrix
confuse_ksvm = table(Ytest, Ytest_pred_ksvm)
# mis-classification rate
misclass_rate_ksvm = rep(NA, 10)
number = rep(NA, 10)
for (i in 1:10) {
  misclass_rate_ksvm[i] = 1 - confuse_ksvm[i,i]/sum(confuse_ksvm[,i])
}
class = 0:9
misclass_ksvm = cbind(class, round(misclass_rate_ksvm,3))
# overall mis-classification rate
overall_misclass_ksvm = mean(Ytest_pred_ksvm != Ytest)
confuse_ksvm
```

```
##      Ytest_pred_ksvm
## Ytest   0   1   2   3   4   5   6   7   8   9
##     0 830   3   8  29   5   6 106   0  13   0
##     1   3 973   0  17   2   1   4   0   0   0
##     2  20   2 780  12  99   2  83   0   2   0
##     3  33  12  11 886  38   0  16   0   4   0
##     4   1   3  74  29 809   0  80   0   4   0
##     5   0   0   0   0   0 931   1  41   3  24
##     6 152   0  85  29  72   1 645   0  16   0
##     7   0   0   0   0   0  28   0 922   0  50
##     8   4   0   6   3   3   6  12   4 962   0
##     9   0   0   1   0   0  11   0  43   1 944
```

```
misclass_ksvm
```

```
##       class
##  [1,]     0 0.204
##  [2,]     1 0.020
##  [3,]     2 0.192
##  [4,]     3 0.118
##  [5,]     4 0.213
##  [6,]     5 0.056
##  [7,]     6 0.319
##  [8,]     7 0.087
##  [9,]     8 0.043
## [10,]     9 0.073
```

```
overall_misclass_ksvm
```

```
## [1] 0.1318
```

Save