Problem 8.2

Chen Bo Calvin Zhang

23/11/2020

We investigate a high-dimensional microarray gene expression data set.

```r
library("sda") # R package for shrinkage discriminant analysis
```

```
## Loading required package: entropy

## Loading required package: corpcor

## Loading required package: fdrtool
```

```r
# Singh et al. (2002) gene expression prostate cancer data
data(singh2002)

Xtrain = singh2002$x
Ytrain = singh2002$y

print(dim(Xtrain))
```

```
## [1]  102 6033
```

```r
print(levels(Ytrain))
```

```
## [1] "cancer"  "healthy"
```

First, let us write a predictor function for SDA that outputs the confusion matrix.

```r
predfun.sda = function(Xtrain, Ytrain, Xtest, Ytest, diagonal=FALSE)
{
  sda.out = sda(Xtrain, Ytrain, diagonal=diagonal, verbose=FALSE)
  ynew = predict(sda.out, Xtest, verbose=FALSE)$class

  cm = confusionMatrix(Ytest, ynew, negative="healthy")

  return (cm)
}
```

Now, perform K-fold cross validation with 5 folds and 50 repeats.

```r
library("crossval")

# DDA
cv.out = crossval(predfun.sda, Xtrain, Ytrain, K=5, B=50, diagonal=TRUE, verbose=FALSE)
print(cv.out$stat)
```

```
##    FP    TP    TN    FN
## 3.084 7.316 6.916 3.084
```

```r
print(diagnosticErrors(cv.out$stat))
```

```
##       acc      sens      spec       ppv       npv       lor
## 0.6976471 0.7034615 0.6916000 0.7034615 0.6916000 1.6714464
```

```r
# LDA
cv.out = crossval(predfun.sda, Xtrain, Ytrain, K=5, B=50, diagonal=FALSE, verbose=FALSE)
print(cv.out$stat)
```

```
##    FP    TP    TN    FN
## 3.196 7.304 6.804 3.096
```

```r
print(diagnosticErrors(cv.out$stat))
```

```
##       acc      sens      spec       ppv       npv       lor
## 0.6915686 0.7023077 0.6804000 0.6956190 0.6872727 1.6139218
```

Finally, select 100 genes from the 6033 genes and perform K-fold cross validation again.

```r
best100 = sda.ranking(Xtrain, Ytrain, verbose=FALSE, diagonal=TRUE, fdr=FALSE)[1:100,"idx"]
print(best100)
```

```
##   [1]  610 1720 3940  914  364  332 3647 4331  579 1068 1089 4546 3991 1113 1077
##  [16] 3375  735 4088 4073  739 4316 4518  702 3665 1557  694  921 4104  698 3282
##  [31] 4000 4549 2945 4981    2  721 3292 1130 1346 3600 3260 2856 1314 4396 3930
##  [46] 1589 2897 2370 3269 3017  298  292 4154 4040   11 3505  905  718 4552  452
##  [61] 1588  805 1659 3200 3208  684 4013  478  637 2968 1966 4515  377 1647 4492
##  [76] 3313 3242 3879 3585 4496  493 1491 5287 2811 2852 4671 3343 3961  641 5159
##  [91] 1572  341  913 4378 3696 1507 3917   78  731 1329
```

```r
# DDA
cv.out = crossval(predfun.sda, Xtrain[, best100], Ytrain, K=5, B=50, diagonal=TRUE, verbose=FALSE)
print(cv.out$stat)
```

```
##     FP     TP     TN     FN
##  0.000 10.196 10.000  0.204
```

```r
print(diagnosticErrors(cv.out$stat))
```

```
##       acc      sens      spec       ppv       npv       lor
## 0.9900000 0.9803846 1.0000000 1.0000000 0.9800078       Inf
```

```r
# LDA
cv.out = crossval(predfun.sda, Xtrain[, best100], Ytrain, K=5, B=50, diagonal=FALSE, verbose=FALSE)
print(cv.out$stat)
```

```
##     FP     TP     TN     FN
##  0.000 10.188 10.000  0.212
```

```r
print(diagnosticErrors(cv.out$stat))
```

```
##       acc      sens      spec       ppv       npv       lor
## 0.9896078 0.9796154 1.0000000 1.0000000 0.9792401       Inf
```