Problem 10.2

Chen Bo Calvin Zhang

02/01/2021

Let us first load the data and the nexessary libraries.

```
# Singh et al. (2002) gene expression prostate cancer data
library("sda")
```

```
## Loading required package: entropy
```

```
## Loading required package: corpcor
```

```
## Loading required package: fdrtool
```

```
data(singh2002)
Xtrain = singh2002$x
Ytrain = singh2002$y

# load randomForest package
library("randomForest")
```

```
## randomForest 4.6-14
```

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

```
# load crossval package
library("crossval")
```

First, we need a predictor function for the crossval() function. We will have a random forset with 100 trees.

```
predfun.rf = function(train.x, train.y, test.x, test.y)
{
  rf.fit = randomForest(train.x, train.y, ntree=100)
  ynew = predict(rf.fit, test.x)

  # count false and true positives/negatives
  negative = levels(train.y)[2] # "healthy"
  cm = confusionMatrix(test.y, ynew, negative=negative)
  return(cm)
}
```

Now we run cross-validtion and look at the results.

```r
cv.out = crossval(predfun.rf, Xtrain, Ytrain, K=5, B=20, verbose=FALSE)
print(diagnosticErrors(cv.out$stat))
```

```
##       acc      sens      spec       ppv       npv       lor
## 0.9852941 0.9932692 0.9770000 0.9782197 0.9928862 8.7433048
```

Random forest performs very well, however it is very computationally intensive.