Simulation on Pulsar data

Conner McCloney

11/10/2019

Simulating and comparing error rates for K-fold Cross Validation, Leave One Out Cross Validation, and Support Vector Classification on the Pulsar classification data set. For this dataset, all three methods perform on average almost equivalently.

```
#k-fold CV at K=10
suppressMessages(library(boot))
pulsar <- read.csv("pulsar_stars.csv")
cv.error.10=rep(0,10)
for (i in 1:10){
  glm.fit=glm(target_class~.,data=pulsar)
  cv.error.10[i]=cv.glm(pulsar,glm.fit,K=10)$delta[1]
}
cv.error.10
```

```
##  [1] 0.02640191 0.02641446 0.02639544 0.02641056 0.02641312 0.02640184
##  [7] 0.02641136 0.02641101 0.02639539 0.02641079
```

```
#Leave One-Out CV
n <- 200
true.pulsar.n <- floor((length(pulsar$target_class[pulsar$target_class == 1]) / length(pulsar$target_cla
false.pulsar.n <- floor((length(pulsar$target_class[pulsar$target_class == 0]) / length(pulsar$target_cl
true.pulsar <- subset(pulsar, pulsar$target_class == 1)
false.pulsar <- subset(pulsar, pulsar$target_class == 0)

cv.error=rep(0,10)
for (i in 1:10){
  true.index <- sample(1:nrow(true.pulsar),true.pulsar.n)
  false.index <- sample(1:nrow(false.pulsar),false.pulsar.n)
  #Create new dataset of size n that has same true/false positive proportion
  #as original dataset
  new.data <- rbind(true.pulsar[true.index,],false.pulsar[false.index,])
  glm.fit=glm(target_class~.,data=new.data)
  cv.error[i]=cv.glm(new.data,glm.fit)$delta[1]
}
cv.error
```

```
##  [1] 0.03363685 0.02634641 0.03085726 0.03696694 0.02265951 0.02829977
##  [7] 0.02543120 0.03923869 0.03030938 0.03484637
```

```
#Support Vector Machine
for (i in 1:10){
  true.index <- sample(1:nrow(true.pulsar),true.pulsar.n)
  false.index <- sample(1:nrow(false.pulsar),false.pulsar.n)
  #Create new dataset of size n that has same true/false positive proportion
  #as original dataset
  new.data <- rbind(true.pulsar[true.index,],false.pulsar[false.index,])
  new.data$target_class <- factor(new.data$target_class)

  suppressMessages(library(e1071))
```

```
  tune.out=tune(svm,target_class~., data=new.data, kernel="linear", ranges=list(cost=c(0.001, 0.01, 0.1
  bestmodel=tune.out$best.model

  ypred=predict(bestmodel, pulsar)
  print(table(predict=ypred, truth=pulsar$target_class))
}
```

```
##        truth
## predict     0     1
##       0 16195   383
##       1    64  1256
##        truth
## predict     0     1
##       0 16049   314
##       1   210  1325
##        truth
## predict     0     1
##       0 16208   424
##       1    51  1215
##        truth
## predict     0     1
##       0 16233   576
##       1    26  1063
##        truth
## predict     0     1
##       0 16128   234
##       1   131  1405
##        truth
## predict     0     1
##       0 16092   219
##       1   167  1420
##        truth
## predict     0     1
##       0 16099   346
##       1   160  1293
##        truth
## predict     0     1
##       0 16091   267
##       1   168  1372
##        truth
## predict     0     1
##       0 16111   295
##       1   148  1344
##        truth
## predict     0     1
##       0 16222   483
##       1    37  1156
```