Homework due May 9, 2021 07:01 +03

Changes in R since the creation of this material have altered the randomization code. You will need to include the following line in your code before you call `set.seed(N)` in order to obtain the correct answers:

```
RNGkind(sample.kind = "Rounding")
```

Load the following dataset:

```
library(GSE5859Subset)
data(GSE5859Subset)
```

And define the outcome and predictors. To make the problem more difficult, we will only consider autosomal genes:

```
y = factor(sampleInfo$group)
X = t(geneExpression)
out = which(geneAnnotation$CHR%in%c("chrX","chrY"))
X = X[,-out]
```

Note, you will also need to load the following package:

```
library(caret)
```

kNN and Cross Validation Exercises #1

1/1 point (graded)

Set the seed to 1, `set.seed(1)`, then use the `createFolds()` function in the **caret** package to create 10 folds of `y`.

What is the 2nd entry in the fold 3?

15

✓ **Answer:** 15

15

Explanation

```
library(caret)
set.seed(1)
idx = createFolds(y, k=10)
idx[[3]][2]
sapply(idx,function(ind) table(y[ind])) ##make sure every f
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

kNN and Cross Validation Exercises #2

1/1 point (graded)

For the following questions we are going to use kNN. We are going to consider a smaller set of predictors by *filtering* genes using t-tests. Specifically, we will perform a t-test and select the m genes with the smallest p-values.

Let $m = 8$ and $k = 5$ and train kNN by leaving out the second fold, `idx[[2]]`.

How many mistakes do we make on the test set? Remember it is indispensable that you perform the ttest on the training data.

1

✓ **Answer:** 1

1

Explanation

```
library(class)
library(genefilter)
m=8
k=5
ind = idx[[2]]
pvals = rowttests(t(X[-ind,]),factor(y[-ind]))$p.val
ind2 = order(pvals)[1:m]
predict=knn(X[-ind,ind2],X[ind,ind2],y[-ind],k=k)
sum(predict!=y[ind])
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

kNN and Cross Validation Exercises #3

1/1 point (graded)

Now run the code for kNN and Cross Validation Exercises #2 for all 10 folds and keep track of the errors. What is our error rate (number of errors divided by number of predictions) ?

0.375

✓ **Answer:** 0.375

0.375

Explanation

```
library(class)
library(genefilter)
m=8
k=5
result = sapply(idx,function(ind){
  pvals = rowttests(t(X[-ind,]),factor(y[-ind]))$p.val
  ind2 = order(pvals)[1:m]
  predict=knn(X[-ind,ind2],X[ind,ind2],y[-ind],k=k)
  sum(predict!=y[ind])
})
sum(result)/length(y)
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

kNN and Cross Validation Exercises #4

2/2 points (graded)

Now we are going to select the best values of k and m. Use the `expand.grid()` function to try out the following values:

```
ms=2^c(1:11)
ks=seq(1,9,2)
params = expand.grid(k=ks,m=ms)
```

Now use `sapply()` or a for loop to obtain error rates for each of these pairs of parameters. Which pair of parameters minimizes the error rate?

k=

3 ✓ **Answer:** 3

3

m=

1024

✓ **Answer:** 1024

1024

Explanation

```
errors = apply(params,1,function(param){
  k =  param[1]
  m =  param[2]
  result = sapply(idx,function(ind){
    pvals = rowttests(t(X[-ind,]),factor(y[-ind]))$p.val
    ind2 = order(pvals)[1:m]
    predict=knn(X[-ind,ind2],X[ind,ind2],y[-ind],k=k)
    sum(predict!=y[ind])
  })
  sum(result)/length(y)
  })
params[which.min(errors),]
##make a plot and confirm its just one min:
errors = matrix(errors,5,11)
library(rafalib)
mypar(1,1)
matplot(ms,t(errors),type="l",log="x")
legend("topright",as.character(ks),lty=seq_along(ks),col=seq_along(ks))
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

kNN and Cross Validation Exercises #5

1/1 point (graded)

Repeat question kNN and Cross Validation Exercises #4 but now perform the t-test filtering before the cross validation. Note how this biases the entire result and gives us much lower estimated error rates.

What is the minimum error rate?

0.08333333

✓ **Answer:** 0.08333333

0.08333333

Explanation

```
pvals = rowttests(t(X),factor(y))$p.val
errors = apply(params,1,function(param){
  k =  param[1]
  m =  param[2]
  result = sapply(idx,function(ind){
    ind2 = order(pvals)[1:m]
    predict=knn(X[-ind,ind2],X[ind,ind2],y[-ind],k=k)
    sum(predict!=y[ind])
  })
  sum(result)/length(y)
  })
min(errors)
##make a plot and compare to previous question
errors = matrix(errors,5,11)
library(rafalib)
mypar(1,1)
matplot(ms,t(errors),type="l",log="x")
legend("topright",as.character(ks),lty=seq_along(ks),col=se
```

Note how this biases the entire result and gives us much lower estimated error rates. The filtering must be applied without the test set data.

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

kNN and Cross Validation Exercises #6

1/1 point (graded)

Repeat the cross-validation we performed in question kNN and Cross Validation Exercises #4, but now instead of defining `y` as `sampleInfo$group`, use:

```
y = factor(as.numeric(format( sampleInfo$date, "%m")=="06")
```

What is the minimum error rate now?

0

✓ **Answer:** 0

0

Explanation

```
errors = apply(params,1,function(param){
  k =  param[1]
  m =  param[2]
  result = sapply(idx,function(ind){
    pvals = rowttests(t(X[-ind,]),factor(y[-ind]))$p.val
    ind2 = order(pvals)[1:m]
    predict=knn(X[-ind,ind2],X[ind,ind2],y[-ind],k=k)
    sum(predict!=y[ind])
  })
  sum(result)/length(y)
  })
min(errors)
##make a plot and confirm its just one min:
errors = matrix(errors,5,11)
library(rafalib)
mypar(1,1)
matplot(ms,t(errors),type="l",log="x")
legend("topright",as.character(ks),lty=seq_along(ks),col=se
```

Note that we achieve much lower error rate when predicting date than when predicting the group. Because group is confounded with date, it is very possible that these predictors have no information about group and that our lower 0.5 error rates are due to the confounding with date. We will learn more about this in the batch effects section.

Submit

You have used 1 of 5 attempts

ℹ Answers are displayed within the problem