REPORT

수강과목	:	빅데이터통계분석
담당교수	:	선호근
학 과	:	통계학과
학 번	:	201611531
이 름	:	정호재
제출일자	:	2020.12.01.

Homework Assignment 05

The Due Date : By Thursday, December, 1st in class
Your solution should include R codes and the answer of each question.
You need to upload your R codes on http://plato.pusan.ac.kr for full credits.
You may collaborate on this problem but you must write up your own solution.

Perform a simulation study to compare some of classifiers including boosting methods and support vector machines. First, generate the simulation data such that

```
> library(mnormt)
> RNGkind(sample.kind = "Rounding")
> set.seed(1111)
> K <- 100; n <- 200; p <- 10
> x.tran <- x.test <- x.vald <- array(0, c(n, p, K))
> z <- rep(c(1,2,3), each=n/2)
> covm <- matrix(0.6, p, p); diag(covm) <- 1
> for (i in 1:K) {
+   t <- sample(1:p, 1); s <- sample(1:p, t)
+   mu <- rep(0,p); mu[s] <- runif(t,-1, 1)
+   x1 <- rmnorm(3*n/2, mu, covm)
+   x2 <- rmnorm(3*n/2, rep(0,p), covm)
+   x.tran[,,i] <- rbind(x1[z==1,], x2[z==1,])
+   x.test[,,i] <- rbind(x1[z==2,], x2[z==2,])
+   x.vald[,,i] <- rbind(x1[z==3,], x2[z==3,])
+ }
```

In this simulation data, we generate a training set (`x.tran`), a test set (`x.test`) and a validation set (`x.vald`), where the observations of each set consist of 100 cases (`y=1`) for the first 100 observations and 100 controls (`y=0`) for the other 100 observations, i.e., the sample size (`n`) is 200. The number of predictors (`p`) is 10 and the number of simulation replications (`K`) is 100. For the k-th simulation replicate, you have to build a classifier $c(x)$ from the training set (`x.tran[,,k]`) and then compute the classification error rate (CER) of the test set (`x.test[,,k]`). The validation set (`x.vald[,,k]`) should be used only for selection of the optimal tuning parameter(s), if necessary. The final CER of the classifier $c(x)$ should be averaged over 100 simulation replications.

1. Based on the training set, build a boosting classifier, using the **gbm** function of the '**gbm**' package. Consider 3 different boosting models, where the number of splits is `depth=1`, `depth=2` or `depth=3`. Fix the number of trees as `n.trees=1000` when you predict class labels of the test set. Use the prediction threshold of 0.5 for classification. What are the averaged CERs of the test sets for 3 boosting models? Note that you don't need to use the validation set in this question.

```
> q1
[1] 0.23765 0.23730 0.23870
```

왼쪽부터 depth = 1, depth = 2, depth = 3

```
> which(min(q1)==q1)
[1] 2
```

여기서 depth = 2 일 때 가장 좋았다.

2. Repeat Q1 except that we find the optimal number of trees between 1 and 1000, using the validation set. First, build a boosting classifier for the training set. Second, compute the CER of the validation set for each number of trees `n.trees=j`, where `j` starts from 1 to 1000. Find the optimal number of trees that has the smallest CER of the validation set. If you have multiple number of trees that have the smallest CER, choose the maximum number of trees for the optimal value. Finally, apply the boosting method with the optimal number of tree to the test set. What are the averaged CERs of the test sets for 3 boosting models?

```
> q2
[1] 0.22375  0.22760  0.23370
```

왼쪽부터 depth = 1, depth = 2, depth = 3

```
> which(min(q2)==q2)
[1] 1
```

여기서 depth = 1 일 때 가장 좋았다.

3. Build a logistic boosting method, using the `LogitBoost` function of the '`caTools`' package. Similar to Q2, first build a boosting classifier for the training set, where the number of boosting iteration should be fixed as `nIter=1000`. Second, compute the CER of the validation set for each number of iterations `nIter=j`, where `j` starts from 1 to 999 increased by 2, e.g., `j=1`, `j=3`, `j=5`, . . . , `j=999`. Consider only odd numbers of iterations because even numbers can make 50:50 for classification, which leads to the prediction result of `NA`. If you have multiple number of iterations that have the smallest CER, choose the maximum number of iterations for the optimal value. Finally, apply the logistic boosting method with the optimal number of boosting iteration to the test set. What is the averaged CER of the test sets for the logistic boosting method?

```
> q3
[1] 0.2522
```

4. Build a support vector machine (SVM) method, using the function `ksvm` of the '`kernlab`' package. Apply the linear kernel (`kernel='vanilladot'`) and the constrain parameter `C` which can have 26 different values such as `2^seq(-10,15)`. Do not change any other parameters of the function `ksvm`, i.e., just use the default values. First, build SVM to the training set for each value of `C`. Second, apply SVM to the validation set and then compute decision values (predicted values) of 200 validation samples. Next, sort 200 decision values from the smallest to the largest such that

$$\hat{y}_{[1]} < \hat{y}_{[2]} < \cdots < \hat{y}_{[199]} < \hat{y}_{[200]}$$

Each decision value will be used for the threshold of classification. For example, for the i-th validation sample

$$\hat{c}(x_i) = \begin{cases} 1 & \text{if } y_i > \hat{y}_{[j]} \\ 0 & \text{if } y_i \leq \hat{y}_{[j]} \end{cases}$$

where y_i is the decision value of the i-th validation sample and $\hat{y}_{[j]}$ is the j-th threshold for $j = 1, \ldots, 200$. Find both of the optimal value of the constrain parameter `C` and the optimal value of the threshold that can minimize the CER of validation sets. If you have multiple values of `C` that have the smallest CER, choose the minimum value of `C` for the optimal value. If you have multiple values of thresholds that have the smallest CER, compute the mean of the multiple thresholds and take it for the optimal value of the threshold. Finally, apply SVM with the optimal values of `C` and the threshold to the test set. What is the averaged CER of the test sets for SVM?

```
> vanilladot
[1] 0.19415
```

5. Repeat Q4 with 5 different kernels such as

   - Radial kernel (`kernel='rbfdot'`)
   - Polynomial kernel (`kernel='polydot'`)
   - Hyperbolic tangent kernel (`kernel='tanhdot'`)
   - Laplacian kernel (`kernel='laplacedot'`)
   - Bessel kernel (`kernel='besseldot'`).

   What are the averaged CERs of the test sets for 5 SVMs?

```
> rbfdot
[1] 0.21735
> polydot
[1] 0.19245
> tanhdot
[1] 0.2636
> laplacedot
[1] 0.20275
> besseldot
[1] 0.2212
```

6. Based on Q1 $\sim$ Q5, which classifier is the best in terms of the classification error rate?

Best classifer은 커널이 polydot인 ksvm이다.