Lab #2: Classification with R

Biostat 216: Machine Learning in R for the Biomedical Sciences

The purpose of this lab is to get you working with R methods for some classification methods and generally getting more comfortable manipulating objects in R.

Simulating data:

We will simulate something like the weight vs. age data from a cubic function and add some noise and then apply a few methods.

1. Run the command `set.seed(24)` to set the random number generator (that way we should all generate the same set of pseudo-random predictors).
2. Simulate a multivariate (3 variables) predictor with 3 classes using the following

```
library(MASS) ## Note that MASS is already installed as part of base R.

nsamp <- 100

SMat <- matrix(c(1.0, 0.6, 0.3, 0.4, 1.0, 0.7, 0.1, 0.1, 1.0),
nrow=3, ncol=3)

datmat1 <- mvrnorm(n = nsamp, mu = rep(-1.0, 3), Sigma = SMat)
datmat2 <- mvrnorm(n = nsamp, mu = rep(1.0, 3), Sigma = SMat)
datmat3 <- mvrnorm(n = nsamp, mu = c(0.0,0.5,-1.0), Sigma = SMat)
```

3. Stack datamat1, datamat2, and datamat3 to generate a single 300x3 matrix called datamat.
4. Generate a `groups` factor such that the first 100 data points are labeled `marker1`, the second 100 `marker2`, and the third 100 `marker3`.
5. Combine `groups` and `datamat` into a data frame called `dat`.
6. Make pairwise plots of the predictors in the data (use google to find the name of the appropriate command)
7. Re-do the scatterplot so that get each data-point is color coded by `groups` by adding the `col=1+as.numeric(dat$groups)` argument
8. Install the package `scatterplot3d` and load it into R
9. Run the following:

```
s3d <- scatterplot3d(dat$marker1,dat$marker2,dat$marker3,
xlab="marker 1", ylab = "marker 2", zlab = "marker 3",
pch=19, color = as.numeric(dat$groups), main="3D
Scatterplot")

legend(s3d$xyz.convert(3.5, 2, 3), levels(dat$groups), col
= c("black","red","green"), pch = 19)
```

If you like you can try playing around with some of the plotting parameters. Look at the options in the help for `scatterplot3d` for many other parameters that you can play with.

Logistic regression:

10. Rearrange $p(X) = \frac{e^{\beta_0+\beta_1 X_1}}{1+e^{\beta_0+\beta_1 X_1}}$ to get $\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1$ and generalize both equations to p predictors
11. Subset the dataset so that there are only the `groups`: `gene1` and `gene2`. Call the new data `dat2`. You will need to run `dat2$groups <- factor(dat2$groups)`, or `dat2 <- droplevels(dat2)` to remove the unused level from the factor. Otherwise `R` will still think you have a gene3 group.
12. Fit a logistic regression with groups as the outcome and all 3 markers as predictors (additive)
13. Run `summary` on your model
14. Generate confidence intervals for your model
15. Generate odds-ratios for the coefficients of your model
16. Generate odds-ratio confidence intervals for the fitted parameters

Linear discriminant analysis:

17. Load the `MASS` package
18. Fit a linear discriminant analysis to the original data with 3 groups
19. Compare the estimated group means with the values used in the simulation
20. Examine the 2x2 table of predicted vs. actual observations in the training data
21. Examine the predictive accuracy in this training data
22. Simulate a new set of test data (call it `dat3`) under the same conditions, but use `set.seed(15)`.
23. Examine the 2x2 table of predicted vs. actual observations in the test data
24. Examine the predictive accuracy in the test data
25. Compare the accuracy of the results in the test vs. training data. Did one do better than the other? What do you think is going on?