Tutorials on R, Part II, An Example Review on Classification and Discriminant Analysis

Hoofar Pourzand, Email: rcc@rcc.psu.edu
Research Computing and Cyberinfrastructure (RCC)
Computer Building

July 2, 2013

Gene expression data and the information on the physical interaction network [1] is given in the ProjectTwo.Rdata file, which may be loaded into R. Expression Quantitative Trait Loci (eQTLs) are genomic loci that regulate expression levels of mRNAs or proteins [2]. For eQTLs, Quantitative Traits Locus(QTL) mapping methods are used. The QTL refer to , aka characteristics, that vary in degree and can be attributed to polygenic effects, i.e. product of two or more genes and their environment. Quantitative trait loci (QTLs) are in fact regions of DNA containing or linked to the genes that underlie a quantitative trait.

So the genetic architecture is associated with the quantitative traits and these associations if significant- will be of interest for us.
The complex genetics underlying inheritance of thousands of transcript levels, in a cross between two strains of protein types, is the larger picture we are studying here. And also what are the highly heritable transcripts/traits and how high can they become if muted is important- since we don't want to consider all mutations and only those that are "significant" in developing a prediction model that that doesn't "overfit". This will be studied with the use of LDA technique and comparison of its results with Logistic Regression.

1 Introduction to the Problem Statistics

The data is from 231 active genes, in 95 individuals over 6 time steps. The Physical Protein-Protein Interactions (PPI) are assumed to be responsible for the biological processes[3]. In this paper, the goal is to identify interacting genes and later validate the result with PPI data.

2 Method

Data preparation is the first step here[1].

[1] Test Data not used for this tutorial.

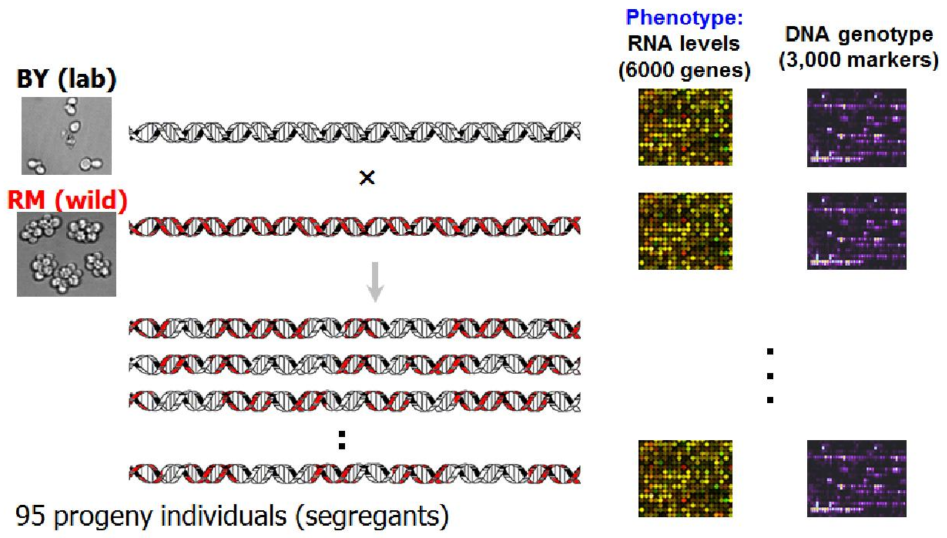

Figure 1: Brem and Kruglyak, The landscape of genetic complexity, 2005

Data	Size
Data.train	122 × 95 × 6
Network.train	122 × 122
Data.valid	53 × 95 × 6
Network.valid	53 × 53

Data	Number of genes	var Names
Train Data	122	Data.train/Network.train
Valid Data	53	Data.valid/Network.valid
Test Data	56	NA

Listing 1: Data Loading in R

```
#Hoofar_Pourzand Tutorial Two
setwd("/home/hoofar/Documents/projectTwo")
load("Project2.RData")  #Load the data
DATA.train #gene expression array
DATA.train[, 1:6] #gene expresssion for gene i,
        individual 1 at t= 1, 2, ...,6
Data.valid #gene expression matrix

Network.train #PPI network
Network.valid #PPI network matrix

Y.train = Network.train[lower.tri(Network.train)]
    #for the training data(convert PPI network to a vector)
Y.valid = Network.valid[lower.tri(Network.valid)]
```

Listing 2: Data Preparation in R

```
#X.train: 7381 by 5 matrix contains 5 predictors for each
gene pair in training data. Predictors: # mean of gene i
# variance of gene i, mean of gene j,  variance of gene j,
covariance between gene i and gene j
```

```
X.train = NULL
for (i in 1:(dim(DATA.train)[1]-1))
  for (j in (i+1):dim(DATA.train)[1])
    X.train = rbind( X.train,
    c(mean(DATA.train[i,]), mean(DATA.train[j,]),
    cov(DATA.train[i,],DATA.train[j,]),
    var(DATA.train[i,]), var(DATA.train[j,])) )

#X.valid : 1378 by 5 matrix contains 5 predictors for each gene pair in test data.
X.valid = NULL
for (i in 1:(dim(DATA.valid)[1]-1))
  for (j in (i+1):dim(DATA.valid)[1])
    X.valid = rbind( X.valid,
    c(mean(DATA.valid[i,]), mean(DATA.valid[j,]),
    cov(DATA.valid[i,],DATA.valid[j,]),
    var(DATA.valid[i,]), var(DATA.valid[j,])) )
```

For prediction we use the model:

$$\chi_{i,j} = f(mean(\xi_i), mean(\xi_j), var(\xi_i), var(\xi_j), cov(\xi_i, \xi_j)) \tag{1}$$

where, $\chi_{i,j}$ is the interaction between gene i, ξ_i, and gene j, ξ_j. The five predictors are the means and the variances and finally the covariance between the two genes.
It's always good to have a look at the data prior to analysis; Y.train is displayed in the figure below:

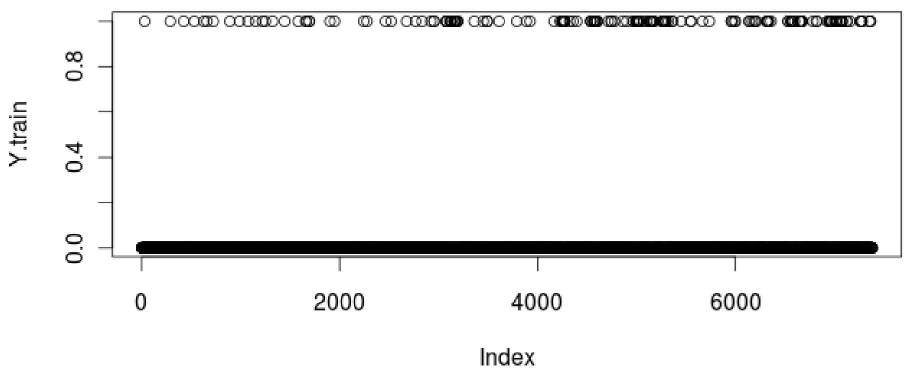

Figure 2: Binary output, Y.train

LDA (Linear Discriminant Analysis) compared with Logistic Regression (LR) is more sensitive to outliers and extremities in the distributions.(Pohar, 2004) As a precautionary step, we will study the outliers in the protein data. Also, it is developed for normally distribued explanatory variables.

It is there fore reasonable to expect LDA to give better resul i nteh case when the normality assumption are fulfillled.

Listing 3: in R

```
#Performing discriminant analysis using LDA, QDA and RDA
library(MASS)

pc.comp <- princomp(scale(X.train))$scores
pc.comp1 <- -1*pc.comp[,1];
pc.comp2 <- -1*pc.comp[,2];
pc.comp3 <- -1*pc.comp[,3];
pc.comp4 <- -1*pc.comp[,4];
pc.comp5 <- -1*pc.comp[,5];

lda.train <- lda(Y.train ~ pc.comp1+pc.comp2+pc.comp3+pc.comp4+pc.comp5);
#,data=Project2 removed, constant cov(i,j)removed for lda

lda.train.fit<- predict(lda.train)$class; #, Project2 for data.frame ommitted
```

Interpretation of the results: LDA makes some strong assumptions. For example, it assumes that the covariance matrix is identical for different classes and that the density is Gaussian. However, LDA may still be valid when the assumptions with regards to the density is violated. In certain cases, LDA may yield poor results. In the first example , we do have similar data sets which follow exactly the model assumptions of LDA. This means that the two classes, red and blue, actually have the same covariance matrix and they are generated by Gaussian distributions.

For the Gene Expression Data, the key element in developeing a good Classificatin model is to use all of the provided features. The problem is a multi-class classification problem, and linear discriminant analysis is suitable for this task. LDA is used, which was popular and successful in face recognition ten years ago (Wang, 2004). It appeared to have surprisingly good results on this problem, possibly because the two tasks are similar. The code is implemented in R.

Trying on feature reduction approach, using generally linear discriminant analysis has been generally preferred for this problem (Rachel Brem, 2005). This suggests that not all the provided features were discriminative. In order to further investigate the importance of the provided features with regards to this dataset, the identication rate of logistic regression classiers based on each category of features is computed separately. This clearly indicates that, some features are far more discriminative than the other features.

In the logistic regression model that the reader will develop this will be evident...

3 discriminant analysis using LDA, QDA and RDA

The Gene Data set has two types of PPI interactions in it. Either we have interaction, Y=1, or we don't, Y=0. Here are the prior probabilities estimated for both of the sample types, first for positive interaction between the proteins and second for no interaction:

$$\hat{\pi}_0 = .97, \hat{\pi}_1 = 197/7381 = 0.03 \tag{2}$$

Listing 4: in R

Now, we see that the first class has a prior probability estimated at 0.97. This means that among the data set, (about 7000 interactions possible), about 97% are no-interaction- these belong to class one- and the other 3% belong to class two.

The classification error rate on training data and test data, respectively.

Listing 5: in R

```
###classification Error
mean(abs(c(lda.train.fit)-Y.train))
```

```
> mean(abs(c(lda.train.fit)-Y.train))
[1] 0.9734453  %for training data
```

Listing 6: in R

```
####validation
lda.valid <- lda(Y.valid ~ X.valid);
lda.valid.fit <- predict(lda.valid)$class;
###classification Error
mean(abs(c(lda.valid.fit)-Y.valid))
```

```
> mean(abs(c(lda.valid.fit)-Y.valid))
[1] 0.969521  %for validation data- no PCA
```

Listing 7: in R

```
##Now with PCA
pc.comp <- princomp(scale(X.valid))$scores
pc.comp1 <- -1*pc.comp[,1];
pc.comp2 <- -1*pc.comp[,2];
pc.comp3 <- -1*pc.comp[,3];
pc.comp4 <- -1*pc.comp[,4];
pc.comp5 <- -1*pc.comp[,5];

lda.valid <- lda(Y.valid~ pc.comp1+pc.comp2+pc.comp3+pc.comp4+pc.comp5);
 #,data=Project2 removed, constant cov(i,j)removed for lda

lda.valid.fit<- predict(lda.valid)$class; #, Project2 for data.frame ommitted
```

```
> mean(abs(c(lda.valid.fit)-Y.valid))
[1] 0.969521 %for validation data-with PCA
```

Listing 8: in R

```
###classification Error
mean(abs(c(lda.valid.fit)-Y.valid))
library(mvtnorm)
#indices = 1:7381
j=1
```

```
X.1.train <- 0
for(i in 1:7381){
    if (Y.train[i]==0){
     X.1.train[j]<- X.train[i]
     j= j+1;
    }
}

j=1
X.2.train <- 0
for(i in 1:7381){
  if (Y.train[i]==1){
    X.2.train[j]<- X.train[i]
    j= j+1;
  }
}

f2<- dmvnorm(x , mean= mean(X.2.train), sigma =cov(cbind(X.2.train)))
#mean, sigma is set up for Y=1

f1<- - dmvnorm(x = c(X.1.train), mean= mean(X.1.train), sigma = cov(X.1.train))
pi<- c(0.97, 0.03)
p.lda <- pi[2]* f2/ (pi[1]*f1 +pi[2]*f2)#for Y=1

#Quadratic Discriminant Analysis
qda.train <- qda(Y.train ~ X.train);
qda.train.fit<-predict(qda.train)$class;


###classification Error
mean(abs(c(qda.train.fit)-Y.train))
```

```
> mean(abs(c(qda.train.fit)-Y.train))
[1] 0.9784582 %for training data
```

Listing 9: in R

```
###validation
qda.valid <- qda(Y.valid ~ X.valid);
qda.valid.fit<-predict(qda.valid)$class;
```

```
> ###validation
> mean(abs(c(qda.valid.fit)-Y.valid))
[1] 1.03193   %for validation data
```

Listing 10: in R

```
###classification Error
mean(abs(c(qda.valid.fit)-Y.valid))
#Regularized Discriminant Analysis

library(klaR)
```

```
lrda.train <- rda(Y.train ~ X.train,
      regularization = c(gamma = 0, lambda = 1));
qrda.train <- rda(Y.train ~ X.train,
      regularization = c(gamma = 0, lambda = 0));

lrda.train$error.rate;
```

```
> lrda.train$error.rate;
      APER    %for training data
0.02682563
```

Listing 11: in R

```
qrda.train$error.rate;
```

```
> qrda.train$error.rate;
      APER    %for training data
0.03075464
```

Listing 12: in R

```
rda.train.fit <- predict(lrda.train)$class;
```

```
> qrda.train
Call:
rda(formula = Y.train ~ X.train,
regularization = c(gamma = 0,
    lambda = 0))


Prior probabilities of groups:
         0          1
0.97330985 0.02669015
%for training data
Misclassification rate:
       apparent: 3.075 %
```

Listing 13: in R

```
lrda.valid <- rda(Y.valid ~ X.valid,
                  regularization = c(gamma = 0, lambda = 1));
qrda.valid <- rda(Y.valid ~ X.valid,
                  regularization = c(gamma = 0, lambda = 0));

lrda.valid$error.rate;
qrda.valid$error.rate;
```

```
> lrda.train
Call:
rda(formula = Y.train ~ X.train,
regularization = c(gamma = 0,
    lambda = 1))


Prior probabilities of groups:
         0          1
0.97330985 0.02669015

Misclassification rate:
       apparent: 2.683 %
```

Listing 14: in R

```
#missclasification rate PPI for rda
#mean(abs(rda.train.fit - Y.train))
```

```
> lrda.valid$error.rate;
      APER %for validation dat
0.03047896
```

Listing 15: in R

```
> qrda.valid$error.rate;
      APER %for validation dat
0.05950653
```

Listing 16: in R

```
> lrda.valid
Prior probabilities of groups:
         0          1
0.96879536 0.03120464

%for validation dat
Misclassification rate:
       apparent: 3.048 %
```

For the posterior probabilities of (i, j) interacting we know that the posterior probability of Y=1 is:

$$P.Ida(Y = 1|X = x) = \frac{\pi_2 f_2}{\pi_1 f_1 + \pi_2 f_2} \tag{3}$$

where f_k is the density of multivariate Gaussian distribution.
π_k is usually estimated simply from empirical frequencies of the training set:

$$\pi_k = \frac{SampleSizeofK}{TotalSampleSize} \tag{4}$$

Since we have the training data set, we may count what percentage of data come from a certain class. And finally the $f_k(x)$ is given by:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)} \tag{5}$$

The 100 most likely interacted pairs are listed below and the proportion of PPI pairs among them is also tabulated.

4 Exercises: logistic regression

- Report the classification error rate on training data and test data, respectively. Find posterior probabilities of (i,j) being interacted.

Listing 17: in R

```
#LOGISTIC REGRESSION
library(DAAG)
logit.lm=formula(log(Y.train)~log(X.train)
cv.binary(obj, rand=NULL, nfolds=5, print.details=TRUE)
```

Listing 18: in R

- Get the 100 most likely interacted pairs and report the proportion of PPI pairs among them.

Listing 19: in R

5 Exercises: linear regression for Y

- Report the classification error rate on training data and test data, respectively.
- Find the 100 most likely interacted pairs and report the proportion of PPI pairs among them.

Listing 20: in R

```
#LINEAR REGRESSION FOR Y
#fitting a multiple linear regression on the training data

fit.lm <- lm(Y.train ~ X.train[,1] +X.train[,2]+X.train[,3]+
                  X.train[,4]+X.train[,5])
fit.lm$fitted.values
```

```
> fit.lm$coefficients
 (Intercept) X.train[, 1] X.train[, 2]
-0.191468134  0.010838233  0.009042716

X.train[, 3] X.train[, 4] X.train[, 5]
0.031828382  0.002483849  0.001379675
```

Listing 21: in R

```
#MSE for Linear Model- full model
mse.lmmodel = mean((fit.lm$fitted.values-Y.train)^2)
```

```
> mse.lasso.fit
[1] 2742.704
%After a backward selection from full model:
Call:
lm(formula =Y.valid~X.valid[,1]+
   X.valid[, 2] + X.valid[, 3])

Coefficients:
(Intercept) X.valid[,1] X.valid[,2] X.valid[,3]
-0.186917      0.012546   0.007801  -0.034926
```

Listing 22: in R

```
# Assessing Outliers
outlierTest(fit.lm) # Bonferonni p-value for most extreme obs
qqPlot(fit.lm, main="QQ Plot") #qq plot for studentized resid
leveragePlots(fit.lm) # leverage plot
#########################
#Normality of Residuals
# qq plot for studentized resid
qqnorm(fit.lm$residuals, main="QQ Plot for residuals")
qqline(fit.lm$residuals)
# distribution of studentized residuals


# diagnostic plots on the test data
fit.lm.valid <- lm(Y.valid ~ X.valid[,1] +X.valid[,2]+X.valid[,3]+
                      X.valid[,4]+X.valid[,5])
```

```
step(fit.lm.valid, direction="backward");
# layout(matrix(c(1,2,3,4,5,6),3,2)) # optional 4 graphs/page
# plot(fit.lm.valid)

# K-fold cross-validation
library(DAAG)

cv.lm(df=data.frame(X.train), fit.lm, m=5, dots= FALSE)
                # 5 fold cross-validation
```

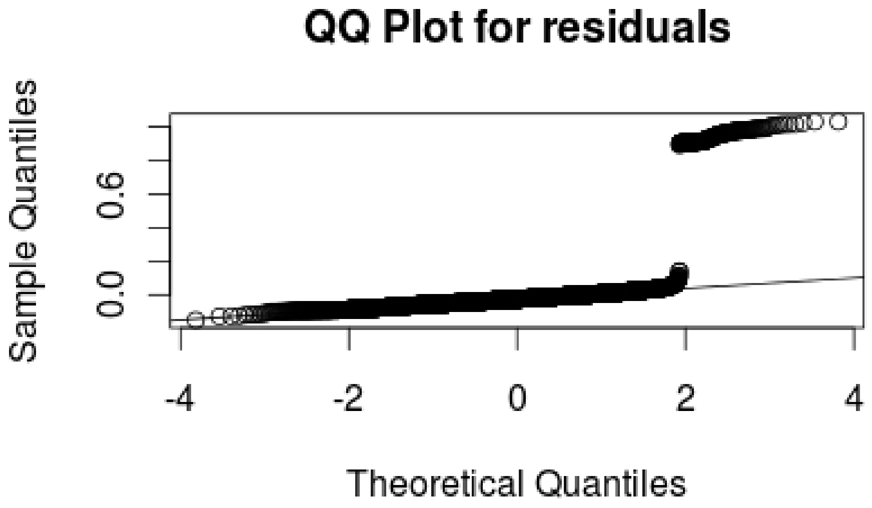

Figure 3: QQ plot for the Residuals,note the violation of Normality assumption

This example illustrates when LDA gets into. LDA separates the classes with hyper planes. This means that the classes have to pretty much be two separated masses, each occupying their share of the space. Then, one has to use more sophisticated density estimation for the classes if one wants to get a good result. This is yet not an example where LDA has seriously broken down. In all,it's always a good idea to look at the scatter plot before one chooses a method. Here the logistic and the LDA are exactly the same. But the scatter plot will often show whether a certain method is appropriate. If a scatter plot like shows that the data classes are not broken into separate pieces, then one can imagine LDA should work regardless, no matter how the positioning of the linear boundary is done (Freidman, 1984)

As mentioned before, LDA must be the better choice if we know the population is normally distributed. However, in practice, the assumptions are nearly always violated, and we have therefore tried to check the performance of both methods- LDA vs LR- with a carefully controlled simulation. Regarding, whether overlapping between classes is a positive wind for LDA or not, one should note that both LR and LDA are appropriate for the development of linear classification models, i.e.

models associated with linear boundaries between the classes. The larger the overlaps the less likely they make a good non-over-fitting model as one increases the number of predictors.

6 conclusion

Gene expression data and the information on the physical interaction network [1] is given in the ProjectTwo.Rdata file, which may be loaded into R. Expression Quantitative Trait Loci (eQTLs) are genomic loci that regulate expression levels of mRNAs or proteins [2]. For eQTLs, Quantitative Traits Locus(QTL) mapping methods are used. The QTL refer to , aka characteristics, that vary in degree and can be attributed to polygenic effects, i.e. product of two or more genes and their environment. Quantitative trait loci (QTLs) are in fact regions of DNA containing or linked to the genes that underlie a quantitative trait.

So the genetic architecture is associated with the quantitative traits and these associations if significant- will be of interest for us.
The complex genetics underlying inheritance of thousands of transcript levels, in a cross between two strains of protein types, is the larger picture we are studying here. And also what are the highly heritable transcripts/traits and how high can they become if muted is important- since we don't want to consider all mutations and only those that are "significant" in developing a prediction model that that doesn't "overfit". This will be studied with the use of LDA technique and comparison of its results with Logistic Regression.

References

[1] Brem and Kruglyak (2005) The landscape of genetic complexity across 5,700 gene expression traits in yeast, proceedings of the national academy of sciences, February 1. vol. 102 no. 5

[2] Gibson, Greg. Muse, Spencer V. (2009) A primer of Genome Science. 3rd ed. Sunderland: Sinaeur Associates, Inc., p229-232.

[3] Friedman, J.H. (1989): Regularized Discriminant Analysis. Journal of the American Statistical Association 84, 165-175.

[4] Ronning. Maximum Likelihood estimation of Dirchlet distributions (1989) Jounal of Statistical Computation and Simulation, 34(4):215-221.

[5] X. Wang and X. Tang (2004) Random sampling LDA for face recognition, Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages 259265 Vol.2.

[6] Maja Pohar, Mateja Blas, Sandra Turk (2004) Comparison of Logistic Regression and Linear Discriminant Analysis: A Simulation Study.

A R script