Statistical Methods for Bioinformatics
[I0U31a]
Assignment 03
Vijver Gene Expression Data Set

Hamed Borhani

May 3, 2016

1

-Large numbers of predictors. -Multiple collinearity: Since all of these genes are related to cancer, one could expect multicollinearity because there could be group of genes related to a specific process in the cell (e.g. a particular cell growth phase), or genes which regulate each other. -No significant correlation between individual genes and the phenotypes.

```
library(glmnet)
load("VIJVER.Rdata")
#dummy coding : DM->1 , NODM->0
phenotype = ifelse(data$meta == "DM", 1, 0)
data <- cbind(phenotype,data[,-1])
```

2

To evaluate the association between individual genes and the phenotype, we can calculate the correlation coefficient (e.g Pearson) between them and the check for high correlations:

```
cor <- cor(gdata)

>table((cor[1,] > 0.5)[-1])
>#[-1] to omit the correlation of phenotype with
   itself
FALSE
 4948
```

No correlation of higher than 0.5 or less than -0.5 (negative correlation) were found. (Pearson correlation coefficient)

This could be repeated for the correlation coefficients of higher than 0.4 as well (just to have some candidate genes to test for the significance)

```
>names(gdata)[cor[1,] > 0.4][-1]
>#[-1] to remove "phenotype" from the list
[1] "NM_003258" "NM_001168"
```

This could further be tested by fitting a logistic regression between the phenotype and each individual gene candidates:

```
># gene NM_003258
>logit.gene1 <- glm(phenotype ~ NM_003258, data =
   data, family = binomial(link = "logit"))
```

```
>summary(logit.gene1)

...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.04761    0.17153  -0.278    0.781
NM_003258    3.84847    0.68080   5.653 1.58e-08 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *
       0.05    .    0.1          1

(Dispersion parameter for binomial family taken to
   be 1)

    Null deviance: 255.15  on 187  degrees of
       freedom
Residual deviance: 212.87  on 186  degrees of
   freedom
AIC: 216.87
...
```

The coefficient of "NM_003258" gene is highly significant, suggesting a (negative) relation between this gene and the phenotype, the "Residual deviance" is also lower than the "Null deviance".

Same thing could be done for the other gene, "NM_001168":

```
># gene NM_001168
>logit.gene2 <- glm(phenotype ~ NM_001168, data =
   data, family = binomial(link = "logit"))
>summary(logit.gene2)

...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.05466    0.16974  -0.322    0.747
NM_001168    2.65441    0.51053   5.199    2e-07 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *
       0.05    .    0.1          1
```

```
(Dispersion parameter for binomial family taken to
   be 1)

    Null deviance: 255.15  on 187  degrees of
       freedom
Residual deviance: 222.26  on 186  degrees of
   freedom
AIC: 226.26
...
```

This process could also be done by computing the Spearman correlation coefficient, using Hmisc library which also provides the p-values:

```
library(Hmisc)
spear_corrs <- rep(NA, ncol(data)-1)
pvals <- rep(NA, ncol(data)-1)
for (i in 2:ncol(data)){
        spear_cor <- rcorr(cbind(data$phenotype,
           data[,i]), type = "spearman")
        spear_corrs[i] <- spear_cor$r[1,2]
        pvals[i] <- spear_cor$P[1,2]
}
>names(data)[which(spear_corrs > 0.4)]  #no
   correlation < -0.4
[1] "NM_003258"      "NM_002811"      "NM_012291"
        "Contig31288_RC" "NM_003981"      "NM_014176
   "
 [7] "NM_004701"      "NM_007057"      "NM_006461"
         "NM_001168"

>pvals[which(spear_corrs > 0.4)]
[1] 2.408695e-11 1.112723e-08 3.082709e-09 1.196568e
   -08 3.141902e-09 1.218462e-08 1.240741e-08
   3.141902e-09
 [9] 1.634228e-09 3.024594e-09
```

They all have significantly low p-values. But by looking at their correlation matrix, we can see that most of them are highly collinear.

3

By calculating the full correlation matrix, we can check if there is a collinearity between different genes, and also how sever it is.

```
diag(cor) <- NA          #to omit the diagonal
   elements in counting

>table(cor[-1,-1] > 0.5)
FALSE     TRUE
24266310   211446

>table(cor[-1,-1] > 0.9)
FALSE     TRUE
24477268      488
```

This shows that a high number of variables are collinear and 244 of them have correlation coefficients more than 0.9 which suggests sever multicollinearity of in the data.

It is indeed a challenge which could make the model non-stable, because the coefficient estimates would be sensitive to minor changes in the model. The model would also be difficult to interpret. A solution to this situation could be PCA or Factor Analysis.

4

```
library(glmnet)
x <- model.matrix(phenotype~., data)[,-1]
y <- data$phenotype

set.seed(1)
train <- sample(1:nrow(x), nrow(x)/2)
test <- (-train)
y.test <- y[test]
```

Ridge:

```
##choosing lambda tuning parameter by C.V
set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha =0)
bestlambda <- cv.out$lambda.min          #[1]
   59.90289
```

```
#doing a ridge regression
ridge.mod <- glmnet(x[train,], y[train], alpha = 0,
   lambda = 60)

#computing test MSE with this lambda
ridge.pred <- predict(ridge.mod, s = bestlambda,
   newx = x[test,])
>mean((ridge.pred - y.test)^2)
[1] 0.2393686

out <- glmnet(x, y, alpha = 0)
>predict(out, type = "coefficients", s = bestlambda)

(Intercept)          J00129 Contig29982_RC
   Contig42854 Contig42014_RC Contig27915_RC
   Contig20156_RC
  0.3821033016  -0.0001155809  -0.0007157701
     -0.0006814239  -0.0011455571  -0.0008669318
     -0.0012960925
Contig50634_RC Contig42615_RC Contig56678_RC
   Contig48659_RC Contig49388_RC  Contig1970_RC
   Contig26343_RC
 -0.0005948833   0.0008189745  -0.0003348213
    0.0010804055  -0.0001751950   0.0001944577
    -0.0007462841
Contig53047_RC Contig43945_RC    Contig19551
   Contig10437_RC Contig47230_RC Contig20749_RC
  0.0002187805   0.0015232195  -0.0001875428
     -0.0003775705   0.0003379700  -0.0008755412
  ...
```

Lasso:

```
#lasso
lasso.mod <- glmnet(x, y, alpha = 1)
plot(lasso.mod)
sqrt(sum(coef(lasso.mod)[-1,60]^2))

set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha =1)
```

```
plot(cv.out)
bestlambda <- cv.out$lambda.min          #[1]
   0.1203589

lasso.mod <- glmnet(x[train,], y[train], alpha = 1,
   lambda = bestlambda)
#computing test MSE with this lambda
lasso.pred <- predict(lasso.mod, s = bestlambda,
   newx = x[test,])
>mean((lasso.pred - y.test)^2)
[1] 0.2337597

out <- glmnet(x, y, alpha = 1, lambda = bestlambda)
lasso.coefs <- predict(out, type = "coefficients", s
    = bestlambda)

>lasso.coefs[lasso.coefs != 0]
[1]  0.439162475 -0.044044298  0.244576724
   -0.090362966 -0.057141054  0.135838330
   0.033935840  0.009716377
 [9]  0.064617344  0.013627813
```

Lasso selects 9 genes (+ intercept) as effective predictors. Lasso also has a lower test MSE (0.2337597) compared to ridge (0.2393686)

PCR:

```
library(pls)
set.seed(2)
pcr.fit <- pcr(phenotype~., data = data, scale =
   TRUE, validation = "CV")
summary(pcr.fit)

#PCR on training data
set.seed(1)
pcr.fit <- pcr(phenotype~., data = data, subset =
   train, scale = TRUE, validation = "CV")
validationplot(pcr.fit, val.type = "MSEP")

#test MSE
pcr.pred <- predict(pcr.fit, x[test,], ncomp = 10)
>mean((pcr.pred - y[test])^2)
```

```
[1] 0.2078954

#pcr on full dataset
pcr.fit <- pcr(y~x, scale = TRUE, ncomp = 10)

>summary(pcr.fit)
Data:   X dimension: 188 4948
        Y dimension: 188 1
Fit method: svdpc
Number of components considered: 10
TRAINING: % variance explained
   1 comps  2 comps  3 comps  4 comps  5 comps  6
      comps  7 comps  8 comps  9 comps  10 comps
X   11.728   19.231   25.988    29.92    33.60
   36.59    39.19    41.72    43.59     45.28
y    5.727    5.761    9.765    14.80    17.31
   21.18    21.23    22.12    22.43     25.59
```

Lowest CV error corresponds to the model with 10 components. This models results in a test MSE of 0.2078954 which is lower than lasso and ridge. But it only captures 45% of the predictors variance and 26% of response variable.