Practical 3: The epigenetic clock: Building a prediction rule

Verena Zuber, Jelena Besevic, Alpha Forna, and Saredo Said

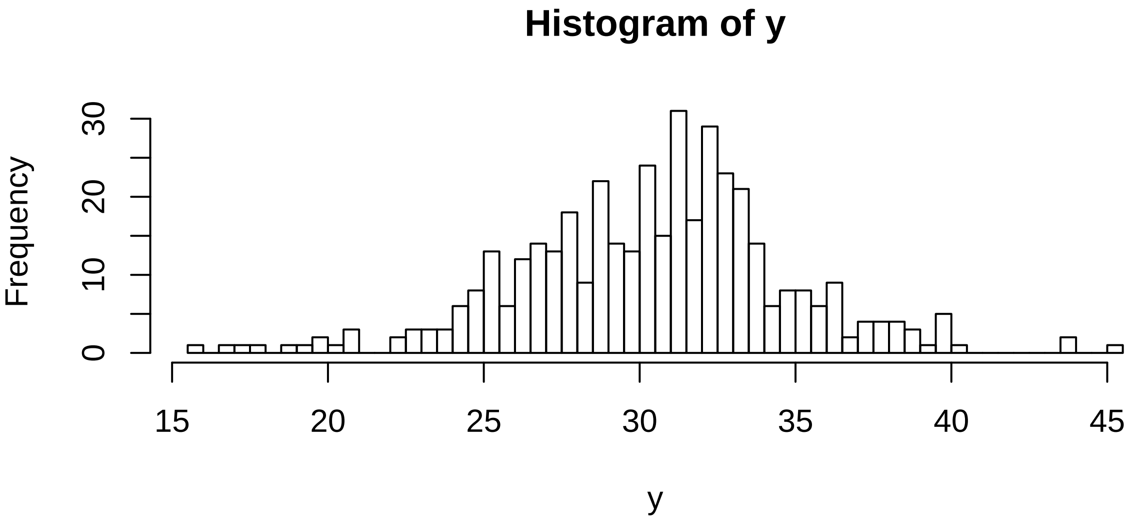
7/2/2019

Part 1: The epigenetic clock: A predictive signature for ageing using penalised regression

In this practical we consider again the same data on $n = 409$ healthy mice and methylation of $p = 3,663$ conserved methylation sites as last week. Our goal is to train our own epigenetic clock and use the methylation data to predict the biological age of mice.

Load the dataset, that contains the methylation matrix as predictor matrix and the age of the mice (in months) stored in the vector y. Familiarise yourself with the dataset using the following commands

```
load("data_epigenetic_clock_control")
#alternatively try load("data_epigenetic_clock_control.dms")
y = control_mice$y_control
hist(y,breaks=50)
```



```
x = control_mice$x_control
dim(x)
```

```
## [1]  409 3663
```

Question 1.1

First load the glmnet package and fit a lasso regression, where you use y as the outcome and x as predictor matrix (always make sure x is a matrix and not a dataframe). For the first question set lambda to a fixed value equal 0.9. Run the lasso and see how many beta-coefficient are unequal zero and thus included in the model.

```
library(glmnet)
```

Question 1.2

When performing penalised regression it is not advised to set the regularisation parameter before seeing the data. It is good practice to perform cross-validation (cv) to set the regularisation parameter. Use the cv.glmnet function and the option type.measure = “mse” to optimise the mean squared error (mse). Find the lambda parameter that minimises the cv mse using the value $lambda.min. What is the lambda parameter that is largest, but has a mse that is within one standard error of the minimum mse using the value $lambda.1se?

Question 1.3

Redo the cv and check if the optimal lambda parameters are the same. What can you do to control the random number generator in R?

Question 1.4

Fit the two lasso models, one with the lambda that optimises the mse, the second with the largest lambda that is within one standard error of the minimum mse. How many variables are included in each model and discuss the impact of the regularisation.

Question 1.5

Fit a cv to define the optimal regularisation for the ridge regression. What are the optimal lambda parameter for the minimum cv mse and 1 standard error within the minimum? Fit a ridge regression with the respective parameter.

Question 1.6

Fit a cv to define the two optimal regularisation parameter for elastic net regression. Focus on the largest lambda within 1 standard error of the minimum. This will provide the sparsest model (fewest predictors) that is almost as good as the one with the minimum cv mse. What are the optimal lambda and alpha parameter? Use the following code to search the optimal combination of lambda and alpha on a grid.

```
set.seed(1234)
a = seq(0.05, 0.95, 0.05)
search = foreach(i = a, .combine = rbind)%do%{
            cv = cv.glmnet(x,y,family = "gaussian", type.measure = "mse", alpha = i)
            data.frame(cvm = cv$cvm[cv$lambda == cv$lambda.1se], lambda.1se = cv$lambda.1se, alpha = i)
}
elasticnet.cv = search[search$cvm == min(search$cvm), ]
elasticnet.cv
```

```
##         cvm lambda.1se alpha
## 16 10.84984 0.08795974   0.8
```

Finally fit an elastic net model using the optimal lambda and alpha regularisation parameter.

Part 2: Which of the 3 models (ridge, lasso and elastic net) builds the better prediction rule?

In the second part we perform a cv to compare how well the three different models can predict new data. To this end we use the

```
library(crossval)
```

package for which we need to write a prediction function. Please see here how to define a prediction function for a linear regression model.

```
predfun.lm = function(train.x, train.y, test.x, test.y){
    #fit the model and build a prediction rule
    lm.fit = lm(train.y ~ ., data=train.x)
    #predict the new observation based on the test data and the prediction rule
    ynew = predict(lm.fit, test.x )

    #compute mse as squared difference between predicted and observed outcome
    out = mean( (ynew - test.y)^2 )
    return( out )

}
```

Use this code fragment to write a prediction function for the following

Question 2.1

Lasso (with the regularisation parameter lambda as set in Question 1.2 to be the largest lambda that has a mse that is within one standard error of the minimum mse using the value $lambda.1se)

Question 2.2

Ridge (with lambda as $lambda.1se in Question 1.5)

Question 2.3

Elastic net (with lambda and alpha as $lambda.1se in Question 1.6)

Question 2.4

Use the crossval package to perform a k-fold cross validation with k=5 folds. For each of the three methods output the mean and standard error of the cv test error and discuss which method generalises best to new data.

Part 3 (optional): Differential expression using shrinkage t-score

The third question considers the response of breast cancer patients to treatment. The aim is to identify differentially expressed genes in 2 responder groups (pathologic complete response or minimal residual cancer burden [RCB-I] defining excellent response, vs moderate or extensive residual cancer burden [RCB-II/III] defining lesser response.

For the original publication please see https://jamanetwork.com/journals/jama/fullarticle/899864

Load the dataset including $n = 414$ patients and the predictor matrix including expression of $p = 22,283$ genes.

```
load("JAMA2011_breast_cancer")
#alternatively try load("JAMA2011_breast_cancer.dms")
y = data_bc$rcb
table(y)
```

```
## y
##   0   1
## 117 297
```

```
x = data_bc$x
dim(x)
```

```
## [1]   414 22283
```

Question 3.1

Compute first the sample variance estimate of each gene using the var.shrink function in the

```r
library(corpcor)
```

package using the option lambda=0 for no shrinkage and save it in a vector. Then compute the shrinkage variance using the same function but do not specify the shrinkage parameter, so that lambda is estimated from the data. Plot two boxplots of the 1000 smallest variances for the sample and the shrinkage variance to visualise the impact of the shrinkage.

Question 3.2

Next compute for each of the $p = 22,283$ genes the shrinkage t-score using the function shrinkt.stat() in the

```r
library(st)
```

st package. Save this to a vector named shrinkt.

Question 3.3

Finally perform a multiple testing correction on the shrinkage t-statistic saved in the shrinkt object using the fdrtool package. The great advantage of the fdrtool is that it works also on t-statistics like the shrinkage t-score. Specify statistic = "normal" as option in the fdrtool to fit a Normal-distribution as Null distribution for the shrinkage t-score. Use the local fdr and a threshold of 0.2.

How many genes do you identify as differentially expressed between excellent and non-response to treatment?