Prediction and Model Selection with LASSO, Ridge, Elastic Net

The percentage of body fat is an index which is used extensively in healthcare. Identifying this index's relationship to other health factors is of interest to health professionals. The dataset `FatData.csv` includes two measures of body fat percentage, 10 body circumference measurements, and other information such as weight, height, age, etc., recorded for 252 men. Each man's percentage of body fat was accurately estimated by an underwater weighing technique. Interest lies in modelling and predicting the body fat percentage measured by Brozek's index as a function of other variables. The explanation for each of the measured variables is below:

- `brozek` : Percent of body fat using Brozek's equation = $\max(0\,,\,457/Density - 414.2)$
- `siri` : Percent of body fat using Siri's equation = $\max(0\,,\,495/Density - 450)$
- `density` : Density (gm/cm^3)
- `age` : Age (yrs)
- `weight` : Weight (lbs)
- `height` : Height (inches)
- `adipos` : Adiposity index = Weight/(Height^2) (kg/m^2)
- `free` : Fat Free Weight = (1 - fraction of body fat using Brozek's index)×Weight (lbs)
- `neck` : Neck circumference (cm)
- `chest` : Chest circumference (cm)
- `abdom` : Abdomen circumference (cm) at the umbilicus and level with the iliac crest
- `hip` : Hip circumference (cm)
- `thigh` : Thigh circumference (cm)
- `knee` : Knee circumference (cm)
- `ankle` : Ankle circumference (cm)
- `biceps` : Extended biceps circumference (cm)
- `forearm` : Forearm circumference (cm)
- `wrist` : Wrist circumference (cm) distal to the styloid processes

Since we are building a model with percent of body fat using Brozek's index as the dependent/response variable, it makes sense to exclude Siri's index. We also exclude density, since it is part of the equation to get Brozek's index. Thus, if we know the density, we would not need to predict anything, and we would get the body fat percentage directly. We also exclude free, since the calculation of free includes the fraction of body fat using Brozek's index. Thus, the response is part of the equation for free, so we need to exclude it. Since we are ignoring any potential multicollinearity for now, we can leave the other variables in the model, although it is very obvious there are predictors that are extremely correlated like height and weight.

First, We take a look at the order of variables appearing in the model using LASSO vs forward stepwise.

```r
set.seed(444)
library(glmnet)
```

```
## Warning: package ’glmnet’ was built under R version 4.0.5
```

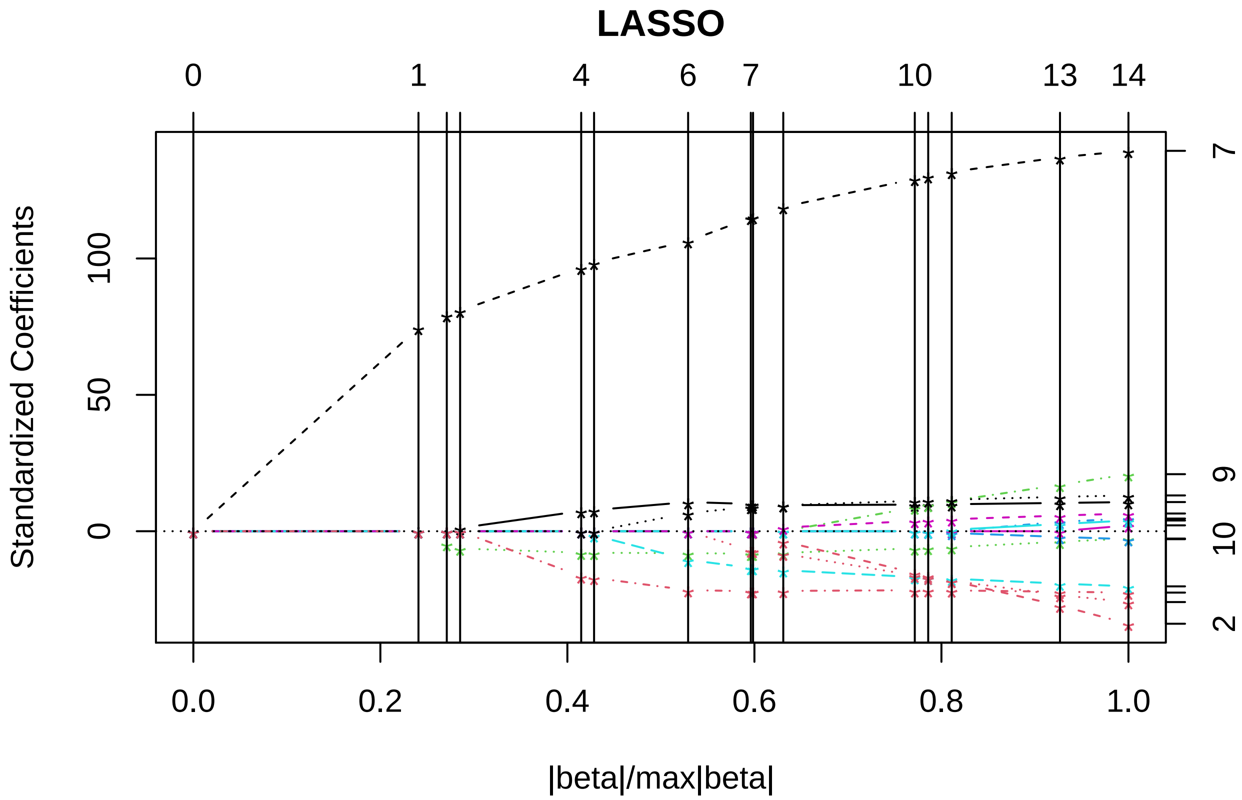
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```r
library(lars)
```

```
## Loaded lars 1.2
```

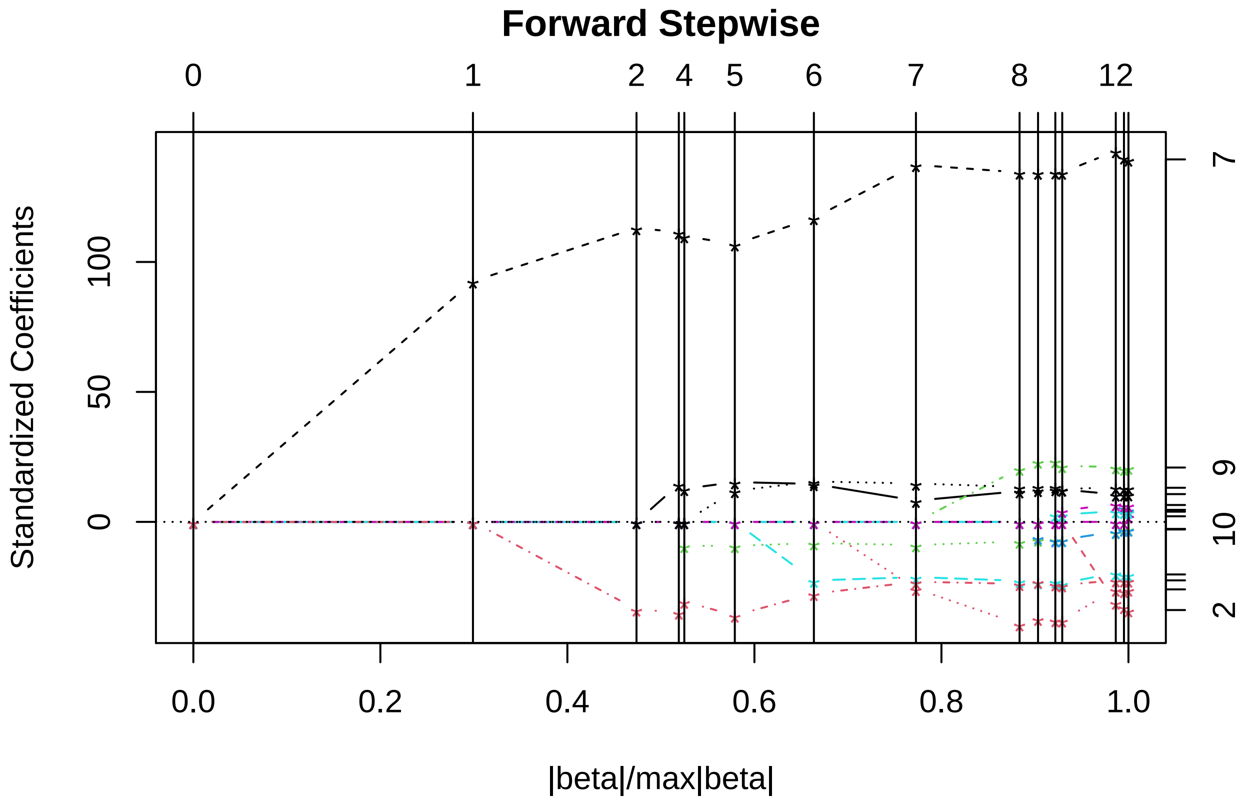
```r
fat <- read.csv("FatData.csv", header = TRUE)

Fat_Modelling = fat[1:225, ]

Fat_Test = fat[226:252, ]


Fat_Modelling <- subset(Fat_Modelling, select = -c(siri,density,free))
Fat_Test = subset(Fat_Test, select = -c(siri,density,free))

pen = 0.2

X <- model.matrix(Fat_Modelling$brozek ~ . , data = Fat_Modelling)[,-1]

Y <- Fat_Modelling[,"brozek"]
model.lasso = lars(X , Y, type="lasso")

plot(model.lasso)
```



```
model.lasso
```

```
##
## Call:
## lars(x = X, y = Y, type = "lasso")
## R-squared: 0.748
## Sequence of LASSO moves:
##      abdom height age wrist neck forearm hip weight biceps thigh knee ankle
## Var      7      3   1    14    5      13   8      2     12     9   10    11
## Step     1      2   3     4    5       6   7      8      9    10   11    12
##      adipos chest
## Var       4     6
## Step     13    14
```

```
model.step = lars(X , Y , type="step")

plot(model.step)
```



```
model.step
```

```
##
## Call:
## lars(x = X, y = Y, type = "step")
## R-squared: 0.748
## Sequence of Forward Stepwise moves:
##      abdom wrist age height forearm neck hip thigh knee ankle biceps weight
## Var      7    14   1      3      13    5   8     9   10    11     12      2
## Step     1     2   3      4       5    6   7     8    9    10     11     12
##      adipos chest
## Var       4     6
## Step     13    14
```

The order of variables that appear using LASSO is abdom, height, age, wrist, neck, forearm, hip, weight, biceps, thigh, knee, ankle, adipos, chest.

The order of variables that appear using forward stepwise is abdom, wrist, age, height, forearm, neck, hip, thigh, knee, ankle, biceps, weight, adipos, chest.

Both methods result in abdom appearing first, and chest appearing last. Both methods also have abdom, wrist, age, height in the top 4.

LASSO

```
set.seed(444)
library(glmnet)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.5
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
lambda <- 10^seq(-3, 3, length = 100)

lasso1 <- train(brozek ~., data = Fat_Modelling, method = "glmnet",
                trControl = trainControl("cv", number = 10),
                tuneGrid = expand.grid(alpha = 1, lambda = lambda))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
coef(lasso1$finalModel, lasso1$bestTune$lambda)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) -4.60240376
## age          0.06047639
## weight      -0.04959182
## height      -0.08727475
## adipos       0.03246926
## neck        -0.50435903
## chest        .
## abdom        0.84406328
## hip         -0.19777806
## thigh        0.17667606
## knee        -0.04208146
## ankle        0.05948302
## biceps       0.11133875
## forearm      0.40154056
## wrist       -1.61699034
```

```
lasso1$bestTune$lambda
```

```
## [1] 0.03764936
```

Using 10-fold cross validation, we find that the optimum value of the penalty factor for a LASSO model is $\lambda = 0.03764936$ The variables that are included in the model based on this penalty factor are age, weight, height, adipos, neck, abdom, hip, thigh, knee, ankle, biceps forearm, wrist.

Elastic Net

```r
library(caret)
library(glmnet)

X = as.matrix(Fat_Modelling[, -1])

cv.error = c()
lambda.cv = c()

alpha.values=seq(0.1, 0.9, by=0.1)
for(alpha in alpha.values) {
  set.seed(844)
  cv.ELN <- cv.glmnet(X, Fat_Modelling$brozek, alpha=alpha)
  cv.error = c(cv.error, min(cv.ELN$cvm))
  lambda.cv = c(lambda.cv, cv.ELN$lambda.min)
}


cv.error
```

```
## [1] 17.83399 17.80542 17.78860 17.77686 17.75830 17.75098 17.74797 17.74273
## [9] 17.73727
```

```r
indx = which.min(cv.error)
alpha.values[indx]
```

```
## [1] 0.9
```

```r
lambda.cv[indx]
```

```
## [1] 0.01779706
```

```r
fit.ELN.CrossValidated = glmnet(X, Fat_Modelling$brozek ,
                             alpha = alpha.values[indx] , lambda=lambda.cv[indx])
coef(fit.ELN.CrossValidated)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -7.251824975
## age          0.062119341
## weight      -0.065104321
## height      -0.064993020
## adipos       0.065737616
## neck        -0.532714157
## chest        0.004636124
## abdom        0.864239526
## hip         -0.221630418
## thigh        0.223764256
## knee        -0.066752112
## ankle        0.111699106
## biceps       0.131098606
## forearm      0.423538269
## wrist       -1.630836946
```

Using 10-fold cross validation, we find that the optimum value of the penalty factor for an Elastic Net model is 0.01779706. The optimum value for the mixing parameter is 0.9.

All the variables are included in the model based on the cross-validated values of the penalty factor and the mixing parameter.

Comparing LASSO, Ridge, Elastic Net and OLS

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(caret)
library(glmnet)

set.seed(444)

# Ridge

cv_lambda_ridge <- cv.glmnet(X, Y, alpha = 0)

cv_lambda_ridge
```

```
##
## Call:  cv.glmnet(x = X, y = Y, alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min 0.6172   100   19.44 1.850      14
## 1se 1.7175    89   21.21 1.834      14
```

```r
ridge.optimal <- glmnet(x = X, y = Y, alpha = 0, lambda = cv_lambda_ridge$lambda.min)



ridge.optimal.coef <- predict(ridge.optimal, type = "coefficients",
                      s = cv_lambda_ridge$lambda.min)[1:15, ]

ridge.optimal$beta
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                   s0
## age       0.10909704
## weight   -0.01362523
## height  -0.09447895
## adipos   0.30753369
## neck    -0.44874394
## chest    0.12253288
## abdom    0.47805452
## hip     -0.05369912
## thigh    0.18114424
## knee    -0.05893399
## ankle   -0.02314328
## biceps   0.05771830
## forearm  0.32513643
## wrist   -1.78098105
```

```
lambda <- 10^seq(-3, 3, length = 100)


ridge <- train(brozek ~., data = Fat_Modelling, method = "glmnet",
               trControl = trainControl("cv", number = 10),
               tuneGrid = expand.grid(alpha = 0, lambda = lambda))

coef(ridge$finalModel, ridge$bestTune$lambda)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) -5.53493690
## age          0.10907249
## weight      -0.01377796
## height      -0.09453964
## adipos       0.30630169
## neck        -0.44907827
## chest        0.12320297
## abdom        0.47816960
## hip         -0.05363803
## thigh        0.18140107
## knee        -0.05865400
## ankle       -0.02280547
## biceps       0.05793149
## forearm      0.32514080
## wrist       -1.78082175
```

```
predictionsr <- ridge %>% predict(Fat_Test)

RMSE = RMSE(predictionsr, Fat_Test$brozek)

RMSE
```

```
## [1] 3.986247
```

```
set.seed(444)


# Lasso
```

```r
lasso1 <- train(brozek ~., data = Fat_Modelling, method = "glmnet",
                trControl = trainControl("cv", number = 10),
                tuneGrid = expand.grid(alpha = 1, lambda = lambda))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```r
coef(lasso1$finalModel, lasso1$bestTune$lambda)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) -4.60240376
## age          0.06047639
## weight      -0.04959182
## height      -0.08727475
## adipos       0.03246926
## neck        -0.50435903
## chest        .
## abdom        0.84406328
## hip         -0.19777806
## thigh        0.17667606
## knee        -0.04208146
## ankle        0.05948302
## biceps       0.11133875
## forearm      0.40154056
## wrist       -1.61699034
```

```r
predictionsl <- lasso1 %>% predict(Fat_Test)

RMSE = RMSE(predictionsl, Fat_Test$brozek)

RMSE
```

```
## [1] 3.963672
```

```r
set.seed(444)
library(dplyr)

alpha = seq(from = 0.1, to = 0.9, by = 0.1)


y <- Fat_Modelling$brozek

cvenet <-  train(brozek ~ . , data = Fat_Modelling, method = "glmnet",
                trControl = trainControl(method = "cv", number = 10), tuneLength = 10)

p <- cvenet$results

p <- subset(p, alpha != 1)

min <- which.min(p$RMSE)
```

```r
p[84,]
```

```
##    alpha     lambda    RMSE  Rsquared      MAE    RMSESD RsquaredSD     MAESD
## 84   0.9 0.03515805 4.17032 0.7026285 3.413843 0.5046806  0.1182202 0.3335978
```

```r
#coef(cvenet$finalModel, 0.03515805)

# Elastic Net
predictionse <- cvenet %>% predict(Fat_Test)

RMSE = RMSE(predictionse, Fat_Test$brozek)

RMSE
```

```
## [1] 3.969896
```

```r
# OLS

ols <- lm(Fat_Modelling$brozek ~. , data = Fat_Modelling)

predols <- predict(ols, newdata = Fat_Test)

sqrt(mean((predols - Fat_Test$brozek)^2))
```

```
## [1] 4.073315
```

We see that the OLS model has the highest prediction error, while LASSO has the lowest prediction error. Thus, we choose LASSO with cross validated parameters for our best model.