

# STAT 380:

## Variable Selection, Ridge, and Lasso Regression

An Example using .... Data

**United Arab Emirates University**

# Variable Selection in Regression

# Forward variable selection



- The first blue point is the variable with the lowest p-value.

# Forward Selection

❑ Only intercept is considered in the very first model. Thereafter, one variable is **added** to the model at a time. Based on the selected model selection criteria.

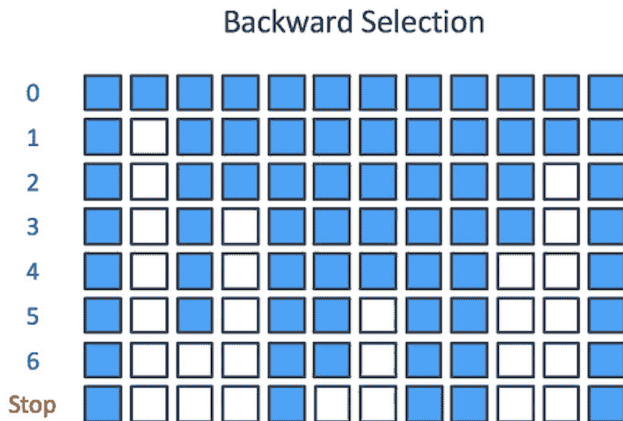
Minimum AIC: Comparing all the models adding one more variable

Minimum BIC: Comparing all the models adding one more variable

Include the variable in the model that has Minimum p-value of the corresponding regression coefficient when adding it to the existing model..

❑ The inclusion of the variables is stopped when a pre-determined p-value/AIC/BIC is achieved.

# Backward variable selection



- The first white point is the variable with the highest p-value.

# Backward Selection

- ❑ All the variables are considered in the very first model. Thereafter, one variable is removed from the model at a time. Based on the selected model selection criteria.

Minimum AIC: Comparing all the models removing one more variable

Minimum BIC: Comparing all the models removing one more variable

Each Step remove the variable that has Maximum p- value of the corresponding coefficient.

- ❑ The Elimination of the variables are stopped when a pre determined p-value/AIC/BIC is achieved.

the data set named  
'surgical'

Available in the R - package  
`library(olsrr)`

□ A dataset containing data about survival of patients undergoing liver operation. \*Kutner, MH, Nachtsheim CJ, Neter J and Li W., 2004

It is a R data-frame with 54 rows and 8 covariates and response is the 'Survival time'

*bcs* : blood clotting score

*pindex* : prognostic index

*enzyme\_test* : enzyme function test score

*liver\_test* : liver function test score

*age* : In years

*Gender* : indicator variable for gender (0 = male, 1 = female)

*alc\_mod* : indicator variable for history of alcohol use (0 = None, 1 = Moderate)

*alc\_heavy* : indicator variable for history of alcohol use (0 = None, 1 = Heavy)



# Forward Selection: R code

```
library(olsrr)
data(surgical)
model <- lm(y ~ ., data = surgical)
step_forward<-ols_step_forward_p(model, details = TRUE)
step_forward
plot(step_forward)
>>>
```

### Elimination Summary

Step	Variable	R-Square	Adj.	C(p)	AIC	RMSE
	Removed		R-Square			
1	alc_mod	0.7818	0.7486	7.0141	734.4068	199.2637
2	gender	0.7814	0.7535	5.0870	732.4942	197.2921
3	age	0.7809	0.7581	3.1925	730.6204	195.4544

# Backward Selection: R code

```
model <- lm(y ~ ., data = surgical)
step_backward<-ols_step_backward_p(model, details = TRUE)
step_backward
plot(step_backward)
```

### Selection Summary

Step	Variable Entered	R-Square	Adj. R-Square	C(p)	AIC	RMSE
1	liver_test	0.4545	0.4440	62.5119	771.8753	296.2992
2	alc_heavy	0.5667	0.5498	41.3681	761.4394	266.6484
3	enzyme_test	0.6590	0.6385	24.3379	750.5089	238.9145
4	pindex	0.7501	0.7297	7.5373	735.7146	206.5835
5	bcs	0.7809	0.7581	3.1925	730.6204	195.4544

# Selection the best combination comparing all the subsets

☐ All the  $2^p$  models are fitted if there are  $p$  variables in total.

Minimum AIC: Comparing all the models best model is chosen based on the AIC/BIC/RMSE/ $R^2$  criterion.

```
```{r }  
# stepwise aic regression  
model <- lm(y ~ ., data = surgical)  
ols_step_both_aic(model, details = TRUE)  
  
```
```

```
```{r }  
model <- lm(y ~., data = surgical)  
allPossible <- ols_step_all_possible(model)  
allPossible  
plot(allPossible)  
  
```
```

# Penalized Regression

## Penalized Regression: Matrix Notation

Let  $(y_i, \mathbf{x}_i) \in \mathbb{R} \times \mathbb{R}^p$  for  $i = 1, \dots, n$  are observed data. A Penalized Regression estimator is via following minimization (with respect to  $\beta$ ) problem:

$$\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \text{ Penalty Function}.$$



## Penalized Regression: idea behind

- They are also known as shrinkage methods or regularization models.
- We would like to continue using linear regression models, but we need to **adjust** them to be usable with big or high-dimensional datasets.
- We introduce a **penalty** for too many or too large coefficients.
- We can fit a model containing all  $p$  predictors using a technique that **constrains** or **regularizes** the coefficient estimates, or equivalently, that **shrinks** the coefficient estimates towards zero.

## Penalized regression: when?

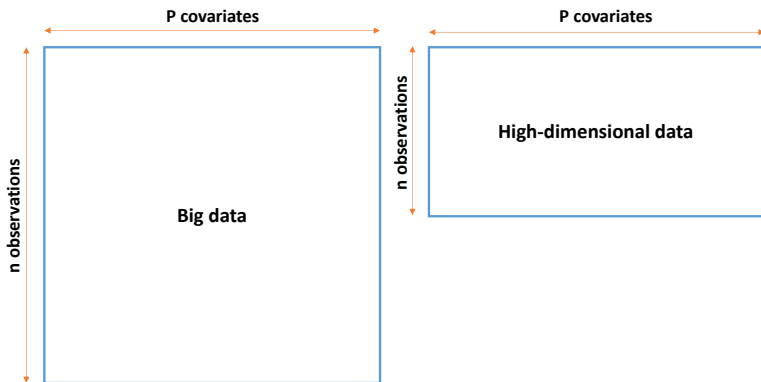
Regularization methods can be used when at least one of the following conditions is met

- large number of variables
- more variables than observations  $n \ll p$
- strong multicollinearity
- a sparse solution is wanted/needed (feature selection)
- "The word 'high-dimensional' refers to a situation where the number of unknown parameters which are to be estimated is one or several orders of magnitude larger than the number of samples in the data."<sup>2</sup>

---

<sup>2</sup>Peter Bühlmann, Sara van de Geer - Statistics for High-Dimensional Data, Springer 2011

# Big data vs. high-dimensional data



# Examples of high-dimensional data

Typically, high-dimensional data arise in a number of settings:

- genomics (microarrays, proteomics)
- signal processing
- image analysis
- market basket data and portfolio allocation
- industry (3d-printing)

## MSE of a predictor: remeber

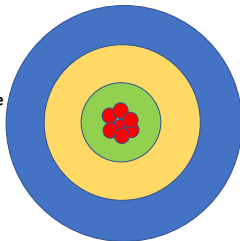
We use the MSE together with cross validation to assess our model fit.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \left( \hat{Y}_i - Y_i \right)^2$$

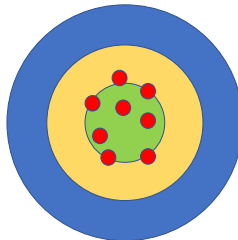
Or more exactly, the mean squared prediction error.

# Bias-variance trade-off

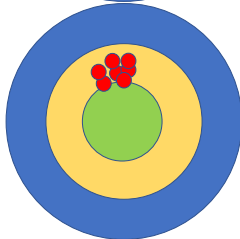
Low variance  
Low bias



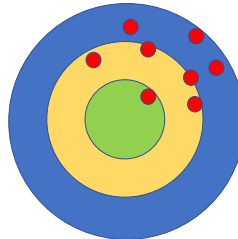
High variance  
Low bias



Low variance  
High bias



High variance  
High bias



# Ridge Regression

## Definition of ridge regression

- The ridge estimate is defined by

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2$$

subject to  $\sum_{j=1}^p \beta_j^2 \leq t.$

- Or equivalently in *Lagrangian form*

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\},$$

with  $\lambda \geq 0$ .



## Ridge Regression: Matrix Notation

Let  $(y_i, \mathbf{x}_i) \in \mathbb{R} \times \mathbb{R}^p$  for  $i = 1, \dots, n$  are observed data. A Ridge Regression estimator  $\hat{\beta}_{\text{ridge}, \lambda}$  is via following minimization problem:

$$\hat{\beta}_{\text{ridge}, \lambda} = \text{Argmin}_{\beta} \left[ \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|^2 \right].$$

,  $\lambda > 0$ .

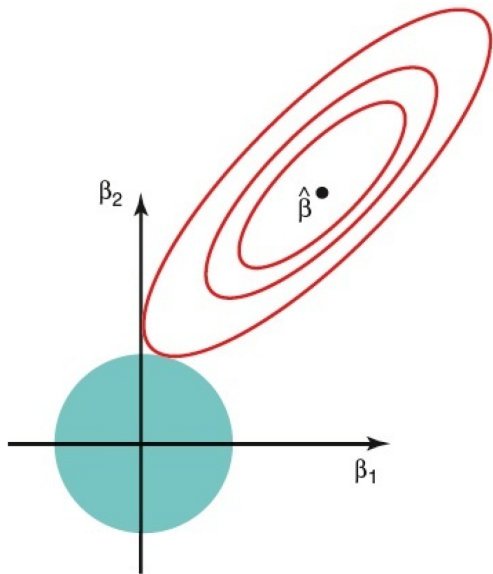
$$\hat{\beta}_{\text{ridge}, \lambda} = (\mathbf{X}^T \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^T \mathbf{y},$$

# The role of $\lambda$

## Ridge Regression: The objective function

$$\|\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 .$$

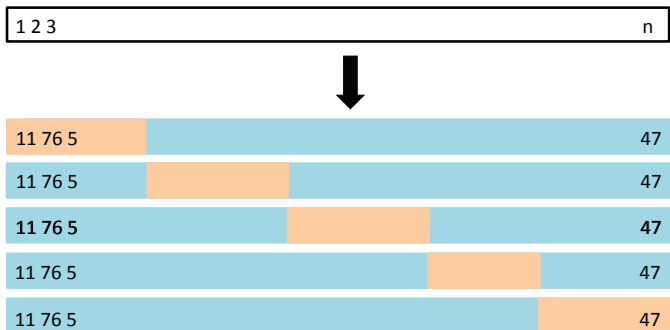
$$\lambda > 0.$$



□ Ridge regression does not exclude variables, but reduces effect estimates to near zero. It is therefore not suited for finding parsimonious models.

We use the `glmnet` R package to fit it.

## 5-fold cross-validation: A schematic display



A set of  $n$  observations is randomly split into 5 non-overlapping groups. Each of these fifths acts as a validation set, and the remainder as a training set. The test error is estimated by averaging the 5 resulting MSE estimates.

# Selection of $\lambda$

```

library(glmnet)
data("surgical") # from the package library(olsrr)
names(surgical[,1:8])
# alpha=0 for fitting a Ridge Regression model
fit_ridge<-glmnet(x =as.matrix(surgical[,1:8]) , y =surgical$y, alpha = 0 )
fit_ridge

plot(fit_ridge, xvar = "lambda", label = TRUE)

# Cross validation to Choose the optimal \Lambda Parameter
cvfit <- cv.glmnet(x =as.matrix(surgical[,1:8]) , y =surgical$y, alpha=0, nfolds = 10)
plot(cvfit)
Ridge_opt_Lambda.model <- glmnet(x=as.matrix(surgical[,1:8]), y=surgical$y,
                                alpha = 0,
                                lambda = cvfit$lambda.min)
Ridge_opt_Lambda.model

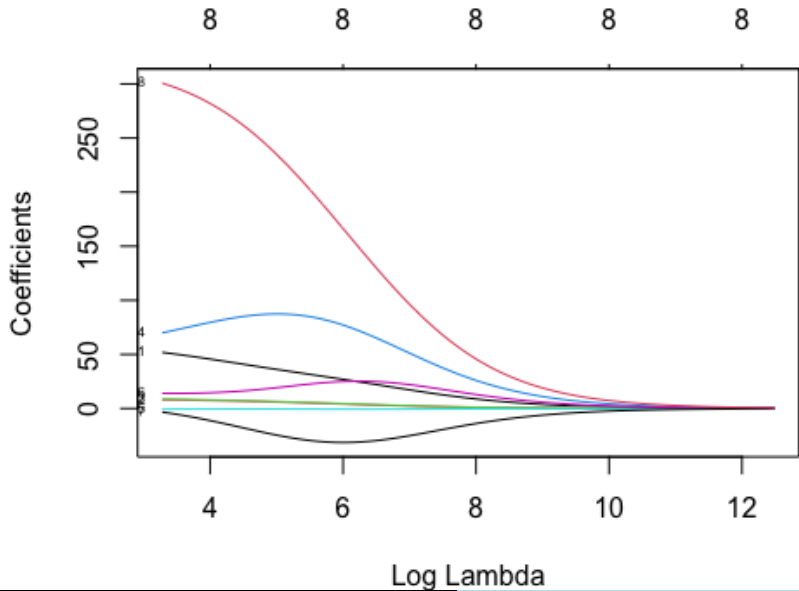
#Alternatively
coef(cvfit, s = "lambda.min")

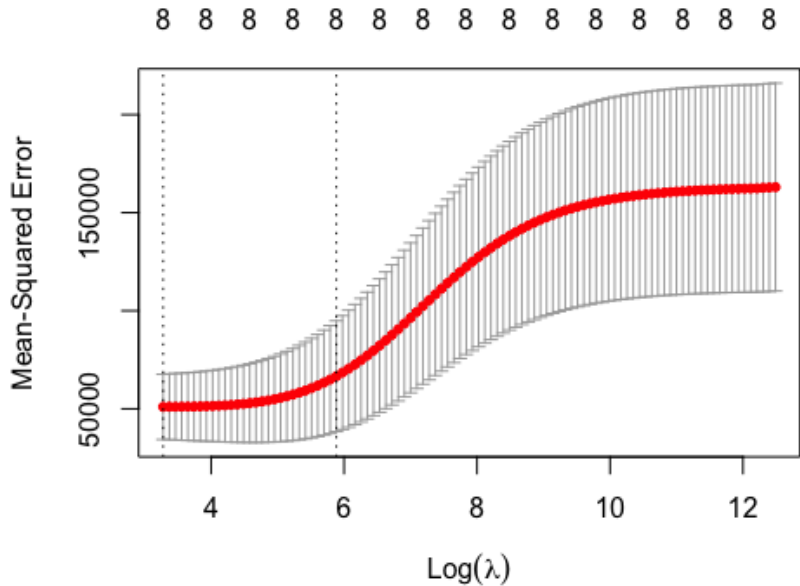
```

# Shrinkage Effect

$$\hat{\beta}_{\text{ridge},\lambda} = (\mathbf{X}^T \mathbf{X} + \lambda I_p)^{-1} \mathbf{X}^T \mathbf{y},$$







```
Call: glmnet(x = as.matrix(surgical[, 1:8]), y = surgical$y, alpha = 0, lambda
= cvfit$lambda.min)
```

```
      Df %Dev Lambda
1  8 77.87  26.54
9 x 1 sparse Matrix of class "dgCMatrix"

              1
(Intercept) -1002.4910407
bcs          51.8370557
pindex       8.0497572
enzyme_test  8.7789615
liver_test   69.9033668
age          -0.6486542
gender       13.9924291
alc_mod      -3.2776238
alc_heavy    300.7347029
```

# LASSO Regression

# Lasso regression

**L**east **A**bsolute **S**hrinkage and **S**election **O**perator introduced by Tibshirani in 1996.

Advantages:

- statistical accuracy in prediction
- variable selection
- computational feasibility

Does not perform well, when groups with high collinearity are present.



## LASSO Regression: Matrix Notation

Let  $(y_i, \mathbf{x}_i) \in \mathbb{R} \times \mathbb{R}^p$  for  $i = 1, \dots, n$  are observed data. A LASSO Regression estimator  $\hat{\beta}_{\text{lasso}, \lambda}$  is via following minimization problem:

$$\hat{\beta}_{\text{lasso}, \lambda} = \underset{\beta}{\text{Argmin}} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=1}^p |\beta_j|$$

, Or

$$\hat{\beta}_{\text{lasso}, \lambda} = \underset{\beta}{\text{Argmin}} \left( \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \|\beta\|_{L_1} \right).$$

$$\lambda > 0$$

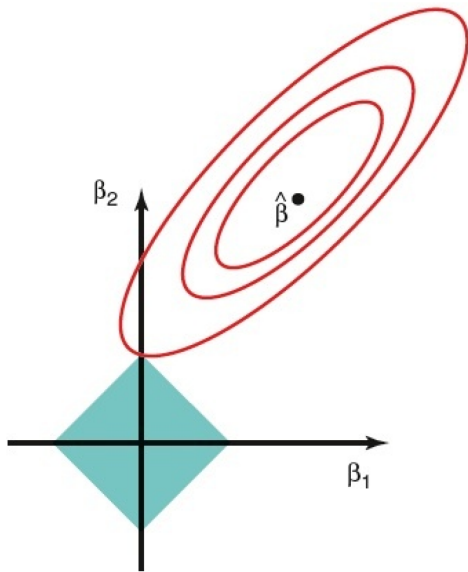
# The role of $\lambda$

## Ridge Regression: The objective function

$$\|\mathbf{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}\|^2 + \lambda \|\tilde{\boldsymbol{\beta}}\|_{L_1}.$$

$$\lambda > 0$$





□ Lasso regression can estimate a regression coefficient to be Exactly zero,. Therefore it has the ability to perform variable selection. It is often used to find parsimonious models.

We use the `glmnet` R package to fit it.

# Selection of $\lambda$

```

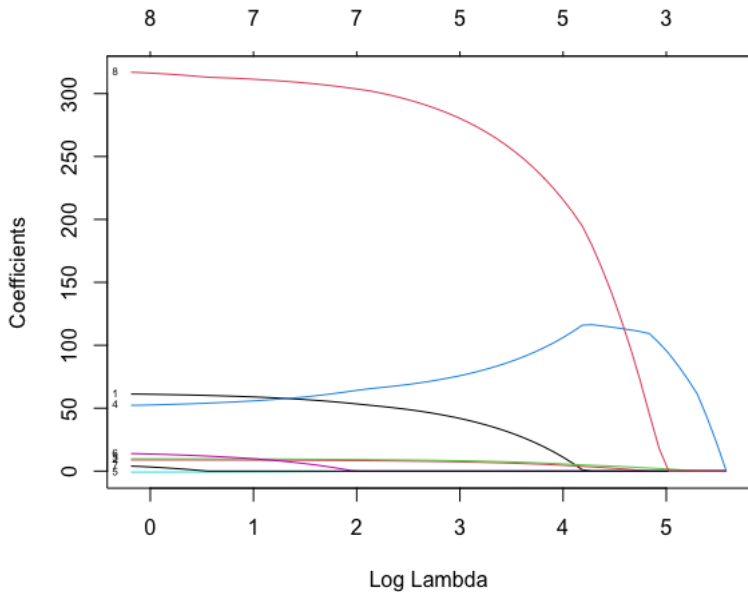
library(glmnet)
data("surgical") # from the package library(olsrr)
names(surgical[,1:8])
# alpha=0 for fitting a Ridge Regression model
fit_lasso<-glmnet(x =as.matrix(surgical[,1:8]) , y =surgical$y, alpha = 1 )
fit_lasso

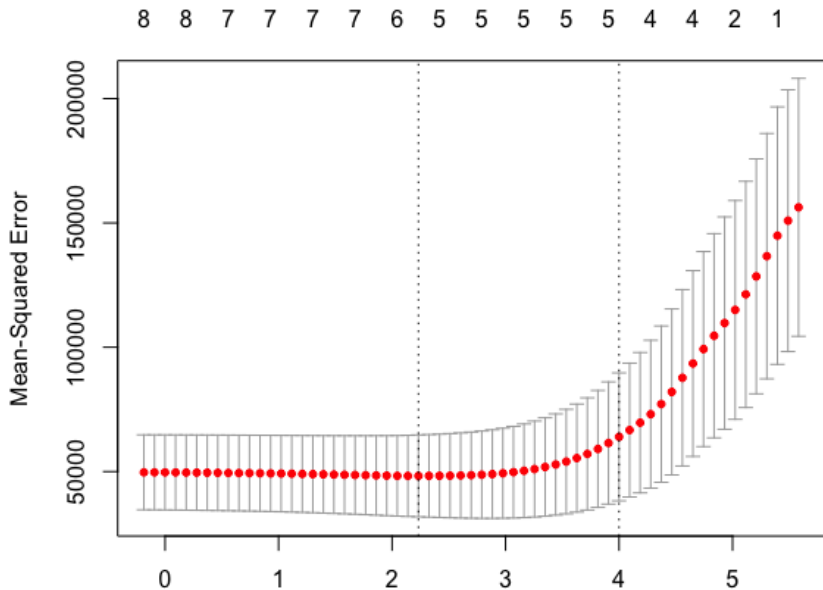
plot(fit_lasso, xvar = "lambda", label = TRUE)

# Cross validation to Choose the optimal \Lambda Parameter
cvfit <- cv.glmnet(x =as.matrix(surgical[,1:8]) , y =surgical$y,alpha=1, nfolds = 10)
plot(cvfit)
Lasso_opt_Lambda.model <- glmnet(x=as.matrix(surgical[,1:8]), y=surgical$y,
                                alpha = 1,
                                lambda = cvfit$lambda.min)
Lasso_opt_Lambda.model

#Alternatively
coef(cvfit, s = "lambda.min")

```





```
Call: glmnet(x = as.matrix(surgical[, 1:8]), y = surgical$y, alpha = 1, lambda
= cvfit$lambda.min)
```

```
      Df %Dev Lambda
1  5 77.86  9.32
9 x 1 sparse Matrix of class "dgCMatrix"
      1
(Intercept) -1049.793801
bcs          51.473401
pindex       8.189280
enzyme_test   9.058995
liver_test    66.362568
age           .
gender        .
alc_mod       .
alc_heavy     300.440816
```

## Regression with Elastic Net Penalty



## Elastic net: Ideas behind

- To address the problems of lasso and ridge, the elastic net was created. It generalizes both and combines the  $l_1$  and  $l_2$  penalties.
- The elastic net **combines** the benefits of both lasso and ridge Regression.
  - It will shrink coefficients for groups of highly correlated variables like Ridge
  - It will set variables to zero like Lasso
  - It can give more than  $n$  non-zero coefficients in the  $n < p$  case

## Elastic net definition

The Elastic Net estimate is given by

$$\hat{\beta}^{elnet} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2$$

subject to  $\sum_{j=1}^p \left( \alpha |\beta_j| + (1 - \alpha) \beta_j^2 \right) \leq t.$

Or equivalently in *Lagrangian form*

$$\hat{\beta}^{elnet} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p \left( \alpha |\beta_j| + (1 - \alpha) \beta_j^2 \right) \right\}$$

with  $\lambda \geq 0$  and  $\alpha \in [0, 1]$ .

## Regression with Elastic Net Penalty: Matrix Notation

Let  $(y_i, \mathbf{x}_i) \in \mathbb{R} \times \mathbb{R}^p$  for  $i = 1, \dots, n$  are observed data. A Elastic Net Regression estimator  $\hat{\beta}_{\text{EN}, \lambda}$  is via following minimization problem:

$$\hat{\beta}_{\text{EN}, \lambda} = \underset{\beta}{\text{Argmin}} \left\| \mathbf{y} - \mathbf{X}\beta \right\|^2 + \lambda \left[ \alpha \|\beta\|_{L_1} + (1 - \alpha) \|\beta\|^2 \right].$$

$\lambda > 0$ , and  $0 \leq \alpha \leq 1$ .

Try: <https://glmnet.stanford.edu/articles/glmnet.html>

# Geneset MicroArray Data

☐ The covariates are the allele frequencies of 200 Gene sets for 120 subjects. (Scheetz et al., (2006)

It represents the data of 120 rats with 200 gene probes.

Response a 120-dimensional vector of, which represents the expression level of 'TRIM32' gene.

We want to identify which of the other genes are significantly responsible for the gene counts of the 'TRIM32'.

☐ Therefore, In terms of the regression terminology:  $n = 120$  and  $p = 200$ .

The data is available in the R library (library('flare'))

```

#library(flares)# for the eyedata
load(url("https://github.com/subhadippal2019/STAT380UAEU/raw/main/eyedata.Rdata"))
```

```{r }
fit_lasso_Gene<-glmnet(x =eyeData$X , y =eyeData$y, alpha = 1 )
fit_lasso_Gene

plot(fit_lasso_Gene, xvar = "lambda", label = TRUE)
# Cross validation to Choose the optimal \Lambda Parameter
cvfit <- cv.glmnet(x =eyeData$X , y =eyeData$y,alpha=1, nfolds = 10)

# the optimal value for lambda for which minimum MSE is achieved is cvfit$lambda.min
paste0("Optimum Lambda=", cvfit$lambda.min)
# Fitting the model with optimum value of Lambda

plot(cvfit)
Lasso_opt_Lambda.model <- glmnet(x =eyeData$X , y =eyeData$y,
                                alpha = 1, lambda = cvfit$lambda.min)
#Printing the model Parameters.
Lasso_opt_Lambda.model$beta
#Alternatively
coef(cvfit, s = "lambda.min")

```

Thank You