Topics on High-Dimensional Data Analytics

LD Learning using Regularization

Kamran Paynabar, Ph.D.
Associate Professor
School of Industrial and Systems Engineering

Ridge Regression

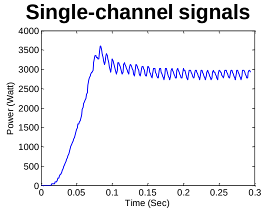
Learning Objectives

- Define regularization and its importance in estimation
- Define ridge regression
- How to use L_2 regularization to avoid variance inflation.

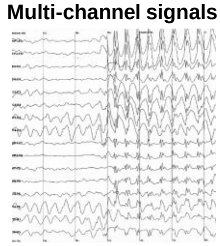
Sparse Information and Regularization

- High-dimensional data usually have a low dimensional structure.
- Important information of HD data is embedded in a few dimensions (sparse) and the rest are non-informative and noise

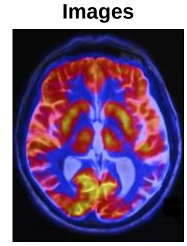
Single-channel signals



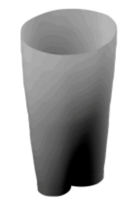
Multi-channel signals



Images



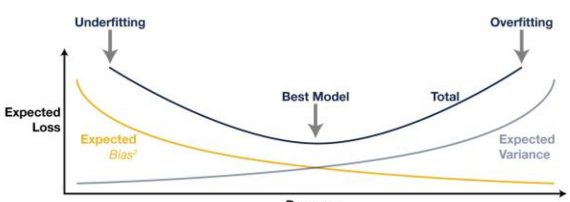
Point Cloud



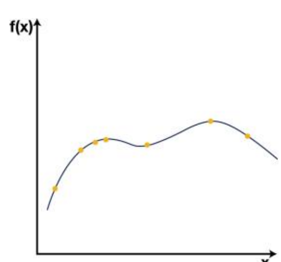
- Regularization helps identify the spares informative features and remove the noise.

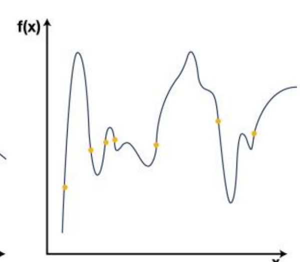
Model Complexity and Regularization

- Regularization can help adjust model complexity to avoid overfitting.



Model Complexity





General Form of Regularization

- Typically a regularized estimation is performed by solving the following optimization problem:

$$\min\{L(\theta; Z_i) + \lambda P(\theta)\}$$

- $L(\theta; Z_i)$ is a loss function (or negative likelihood function)
- $P(\theta)$ is a penalty function.
- λ adjusts the level of regularization.
- Examples include Ridge, Lasso, Non-negative Garrote, Adaptive Lasso, Group Lasso, etc.

$$\min_{\beta_i} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Ridge Regression

Assuming the observations are centered, Ridge estimates can be computed by

$$\min_{\beta_i} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

The Ridge objective function in a matrix form is

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

$$\hat{\boldsymbol{\beta}}^{ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_p\right)^{-1} \mathbf{X}^T \mathbf{y}$$

- Estimated Ridge coefficients are shrunken towards zero
- $\lambda \geq 0$ is the tuning parameter controls the amount of shrinkage
- λ is chosen based on some prediction criteria (MSE) using CV or independent validation data set

Ridge Regression

An equivalent formulation is given by

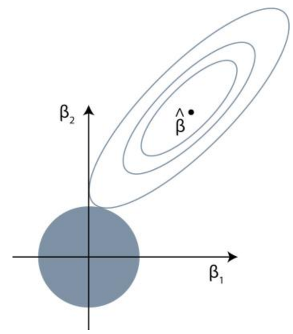
$$\min_{\beta_i} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

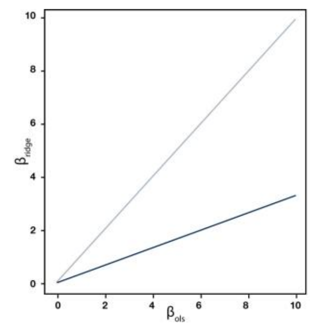
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \le s$$

Estimated Ridge coefficients

- Are linear in y
- Are biased
- Have smaller variance

$$\hat{\boldsymbol{\beta}}^{ridge} = \left(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_p \right)^{-1} \mathbf{X}^T \mathbf{y}$$



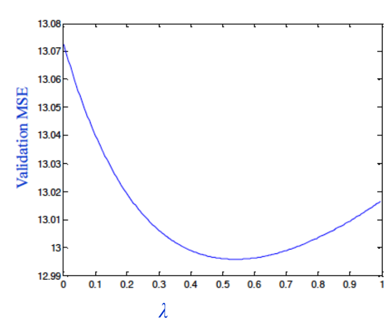


Example

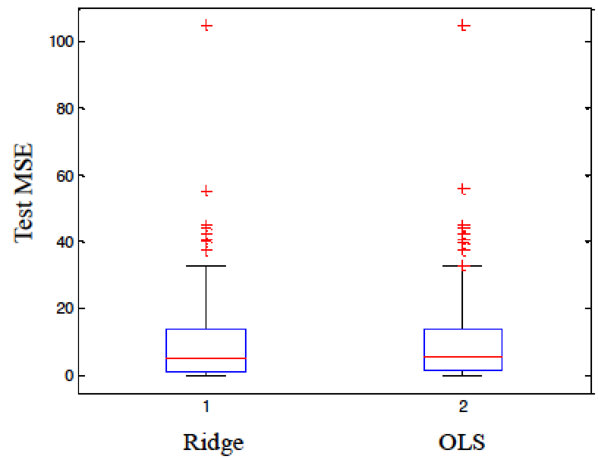
- A sample of 100 centered training data including one response and two independent variables is simulated.
- X1 and X2 are highly correlated.
- Two other independent samples are simulated for validation and test
- Both OLS and Ridge Regression are applied to estimate regression coefficients
- This procedure is repeated 100 times

$$y_i = x_{i1} + x_{i2} + \varepsilon_i; \quad \varepsilon_i \sim N(0, 0.5)$$

$$\min_{\beta_j} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

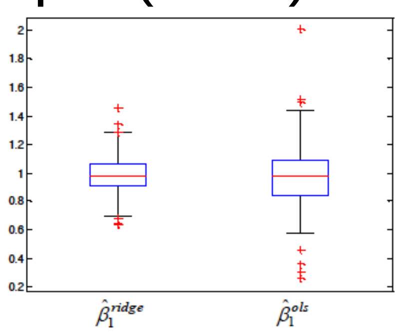


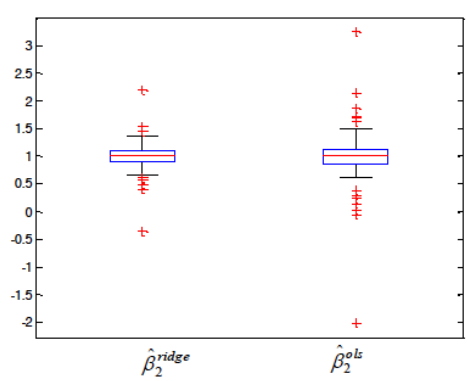
Example



Test MSE for Ridge = 7.1583 Test MSE for OLS = 7.4709

Example (cont.)





Topics on High-Dimensional Data Analytics

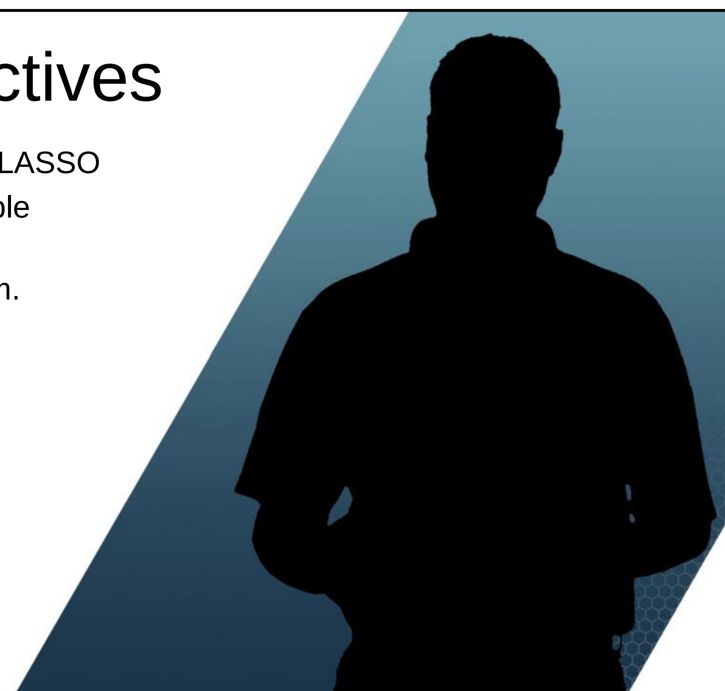
LD Learning using Regularization

Kamran Paynabar, Ph.D.
Associate Professor
School of Industrial and Systems Engineering

LASSO

Learning Objectives

- Define L1 regularization and LASSO
- How to use LASSO for variable selection
- How to solve LASSO problem.



Variable (Feature) Selection

To determine a smaller subset of predictors that exhibit the strongest correlation with response. Traditional variable selection methods include:

- Forward selection
- Backward elimination
- Stepwise selection

The selection process is discrete and thus unstable.

Variable (Feature) Selection using Regularization

- Regularized regression provides with a more stable variable selection procedure compared with traditional variable selection methods.
- If an estimated coefficient for a predictor is zero then the predictor may not have information about the response variable.
- Ridge regression can control model complexity by reducing the variance.
- However, the L2 norm or ridge regression cannot be used for variable selection. That is the solution to the ridge problem is likely non-zero.
- L0 and L1 norm, however, can provide sparse solutions where most of non-informative variables have coefficients equal to zero.

Variable Selection Using L_0 Norm

Suppose we know that the number of non-zero coefficients is at most K. Then using L_0 a regularized (penalized) OLS can be written as

$$\hat{\beta} = \underset{\|\beta\|_0 \leq K}{\operatorname{argmin}} \|y - X\beta\|_2^2$$

- L_0 norm is the total number of nonzero elements in a vector.
- This problem is combinatorial. The best subset selection approach can be used to solve it.
- Since this approach requires solving an OLS for all possible subsets, it is not efficient for cases with a large number of predictors.
- An alternative solution is find an appropriate convex approximation by replacing L_0 norm with L_1 norm.

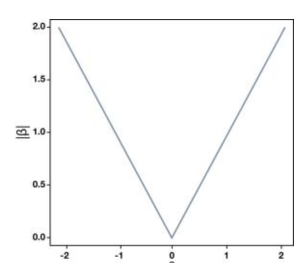
LASSO: Least Absolute Shrinkage and Selection Operator

Assuming the observations are centered, lasso estimates can be computed by (Chen, Donoho and Saunders 1996; Tibshirani 1996)

$$\hat{\beta} = \underset{\|\beta\|_1 \leq K}{\operatorname{argmin}} \|y - X\beta\|_2^2$$

Equivalently $\hat{\beta} = \operatorname{argmin} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$

- Shrinkage: estimated lasso coefficients are shrunken towards zero
- Sparsity: some fitted coefficients are exactly zero
- Continuous variable selection
- λ≥0 is the tuning parameter controls the amount of sparsity

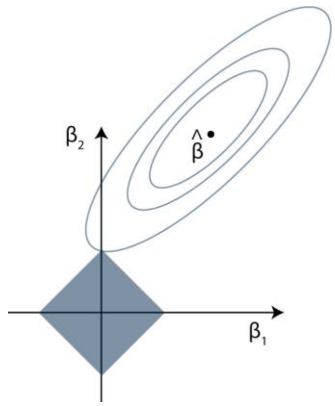


L1 penalty

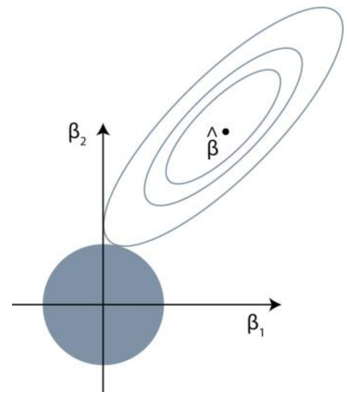
LASSO

$$\min_{\beta_i} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

$$\text{subject to} \quad \sum_{j=1}^{p} \left| \beta_j \right| \le s$$



LASSO



Ridge

LASSO with Orthonormal Predictors

$$\min_{\beta_j} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \left| \beta_j \right|$$

When X is orthonormal the above minimization model is reduced to

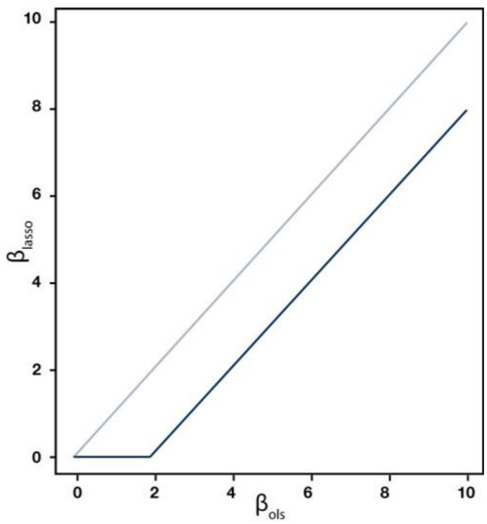
$$\min_{\beta_j} \left(\beta_j - \hat{\beta}_j^{ols} \right)^2 + \lambda \left| \beta_j \right|$$

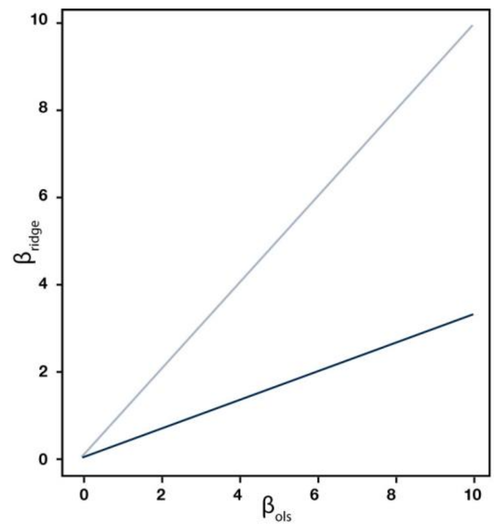
The closed-from solution is given by

$$\begin{aligned} \hat{\beta}_j^{\text{lasso}} &= \begin{cases} \hat{\beta}_j^{\text{ols}} - \frac{\lambda}{2} & \text{if } \hat{\beta}_j^{\text{ols}} > \frac{\lambda}{2} \\ 0 & \text{if } |\hat{\beta}_j^{\text{ols}}| \le \frac{\lambda}{2} \\ \hat{\beta}_j^{\text{ols}} + \frac{\lambda}{2} & \text{if } \hat{\beta}_j^{\text{ols}} < -\frac{\lambda}{2} \end{cases} \\ &= \text{sign}(\hat{\beta}_j^{\text{ols}}) \cdot \left(|\hat{\beta}_j^{\text{ols}}| - \frac{\lambda}{2} \right)_+ \end{aligned}$$

- Lasso shrinks large coefficients by a constant. Lasso truncates small coefficients to zero.

Ridge Vs. LASSO



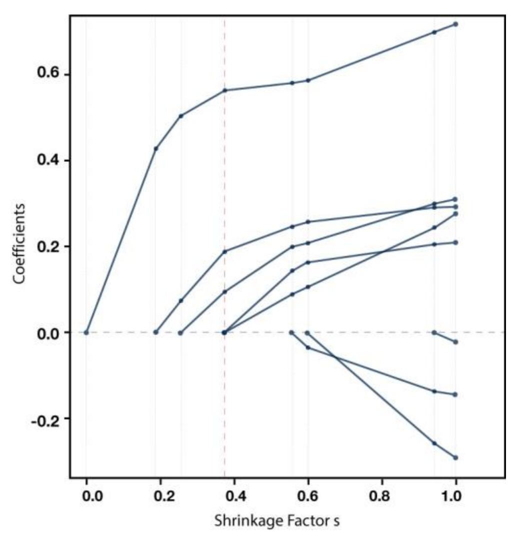


Computation

Convex optimization: Lasso can be solved using convex programming

Piecewise linear solution path by LARS: (Efron et al., 2004)

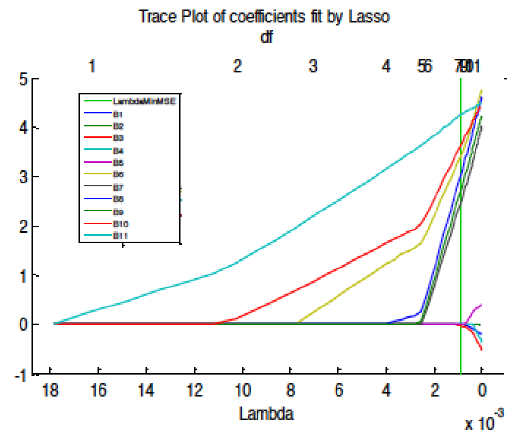
- The path of solution is piecewise linear in λ
- Cost is approximately one least-square calculation



Example

For a regression setting with a 11 predictors, 100 observations are generated such that 5 predictors are not informative.

- Lasso is used to identify important variables and fit a prediction model.
- A 10-fold cross validation is used to choose the tuning parameter.



MSE= 0.2714

Example

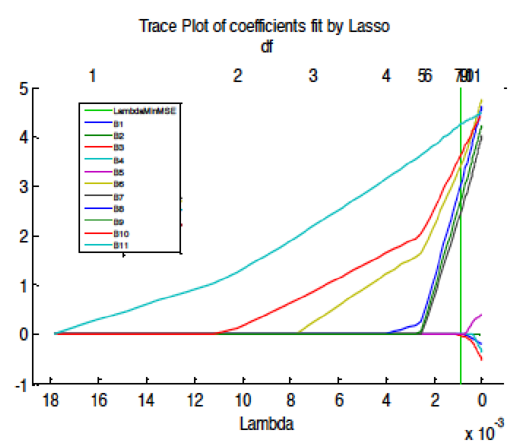
```
clear all
load predictors

for i=1:100
   T(i)=X(i,:)*[0 0 0 0 0 4*ones(1,6)]'+normrnd(0,0.5);
end

[B,FitInfo] = lasso(X,T,'CV',10,'Standardize', 0 , 'Alpha' ,1 )
ax = lassoPlot(B,FitInfo, 'PlotType', 'Lambda')

B(:,FitInfo.IndexMinMSE)

FitInfo.MSE(FitInfo.IndexMinMSE)
```



Topics on High-Dimensional Data Analytics

LD Learning using Regularization

Kamran Paynabar, Ph.D.

Associate Professor

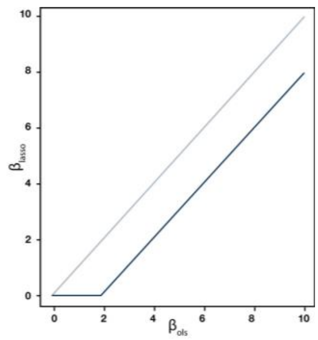
School of Industrial and Systems Engineering

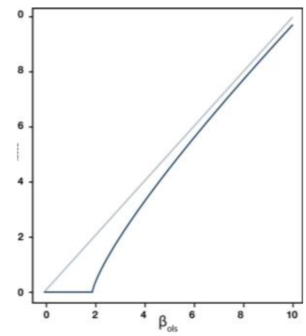
NNG and Adaptive LASSO

Learning Objectives

- Define Non-Negative Garrote and Adaptive LASSO
- Apply Adaptive LASSO and Non-Negative Garrote

Constant Shrinkage in Lasso





- Lasso shrinks all large coefficients by a constant;
- A better shrinkage would consider the magnitude of coefficients in shrinkage;
- Lasso is not consistent in variable selection.

Non-Negative Garrote

Consider lasso with the following change of variables

$$\min_{d_j} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} d_j \hat{\beta}_j^{ols} x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} d_j$$

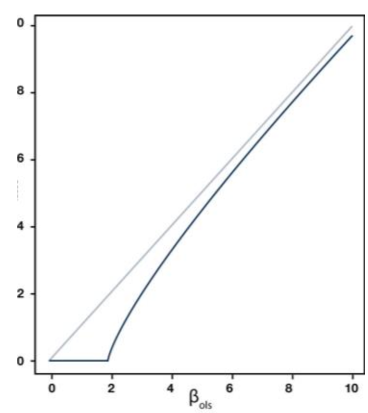
$$\text{subject to} \quad d_j \geq 0$$

- Final Estimates are

$$\hat{\beta}_j = \hat{d}_j \hat{\beta}_j^{ols}$$

- NNG is proposed by Breiman (1995)
- Closed-form solution in orthonormal case

$$\hat{\beta}_j^{\text{garrote}} = \text{sign}(\hat{\beta}_j^{\text{ols}}) \cdot \left(|\hat{\beta}_j^{\text{ols}}| - \frac{1}{2} \frac{\lambda}{|\hat{\beta}_j^{\text{ols}}|} \right)_+$$



Adaptive LASSO

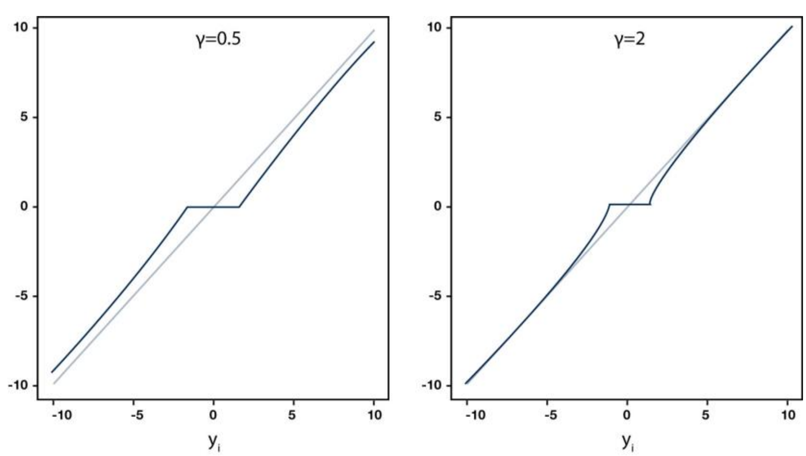
Assuming the observations are centered, adaptive lasso estimates can be computed by

$$\beta^{\text{Alasso}} = \text{argmin}\|y - X\beta\|_2^2 + \lambda_n \sum_{j=1}^{p} \widehat{w}_j |\beta_j|$$

- where $\widehat{w}_j$ is a weight defined by $\widehat{w}_j = \frac{1}{\left|\widehat{\beta}_j^{ols}\right|^{\gamma}}$, $\widehat{w}_j = \frac{1}{\left|\widehat{\beta}_j^{ridge}\right|^{\gamma}}$, etc.

- Adaptive lasso (Zou, 2007) is very similar to NNG.

Adaptive LASSO



$$\beta^{\text{Alasso}} = \min\|y - X\beta\|_2^2 + \lambda_n \sum_{j=1}^{p} \widehat{w}_j |\beta_j|$$

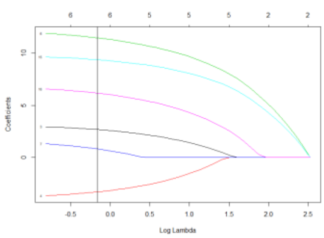
Example

True model

$$y_i = 6.94x_3 - 4.03x_6 + 1.90x_{11} + 3.23x_{14} + 12.26x_{18} + 9.99x_{20} + \epsilon_i$$

$$y \in \mathbb{R}^{100}, X \in \mathbb{R}^{100\times 20}, \epsilon \in \mathbb{R}^{100}$$

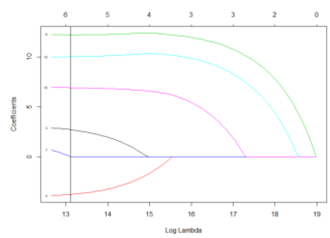
Lasso



$\lambda = 0.84, \log\lambda = -0.17$

$mse = 4.12$

Adaptive lasso - ols

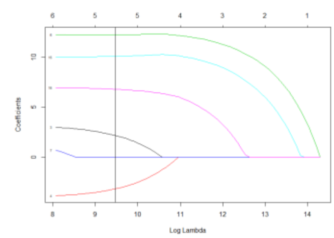


$\gamma = 2$

$\lambda = 499{,}558.56 \log\lambda = 13.12$

$mse = 3.19$

Adaptive lasso - ridge



$\gamma = 2$

$\lambda = 12{,}892.95, \log\lambda = 9.46$

$mse = 4.73$

Example

True model

$$y_i = -8.18x_1 + 4.17x_2 + 9.23x_5 + 3.90x_8 + 8.56x_{10} + 7.40x_{14} + \epsilon_i$$

$$y \in \mathbb{R}^{100}, X \in \mathbb{R}^{100\times 20}, \epsilon \in \mathbb{R}^{100}$$

Using CV, to find the optimal lambda we get:

	True beta	Lasso	Alasso - ols	Alasso - ridge
3	6.94	6.45	6.90	6.91
6	-4.03	-3.47	-3.73	-3.76
11	1.90	1.33	0.93	1.00
14	3.23	2.61	2.73	2.76
18	12.26	11.72	12.22	12.23
20	9.99	9.38	9.93	9.93

Example

```
# Data generation
p = 20      #Number of parameters
n = 100     #Number of observaations
x = rnorm(p*n,0,1)
X = matrix(x,nrow = n)
rm(x)
n0 = sample(1:p,6)
beta = rep(0,p)
beta[n0] = rnorm(6,0,5)
y = X%*%beta + rnorm(n,0,0.5)
```

```
# Lasso
lasso = cv.glmnet(X, y, family = "gaussian", alpha = 1, intercept = FALSE)
lambda = lasso$lambda.min
coef.lasso = matrix(coef(lasso, s = lambda))[2:(p+1)]
lasso = glmnet(X, y, family = "gaussian", alpha = 1, intercept = FALSE)
plot(lasso, xvar = "lambda", label = TRUE, main = "Lasso")
abline(v = log(lambda))
# Adaptive lasso
gamma = 2
b.ols = solve(t(X)%*%X)%*%t(X)%*%y
ridge = cv.glmnet(X, y, family = "gaussian", alpha = 0, intercept = FALSE)
l.ridge = ridge$lambda.min
b.ridge = matrix(coef(ridge, s = l.ridge))[2:(p+1)]
w1 = 1/abs(b.ols)^gamma
w2 = 1/abs(b.ridge)^gamma
alasso1 = cv.glmnet(X, y, family = "gaussian", alpha = 1, intercept = FALSE, penalty.factor = w1)
alasso2 = cv.glmnet(X, y, family = "gaussian", alpha = 1, intercept = FALSE, penalty.factor = w2)
lambda1 = alasso1$lambda.min
lambda2 = alasso2$lambda.min
coef.alasso1 = matrix(coef(alasso1, s = lambda1))[2:(p+1)]
coef.alasso2 = matrix(coef(alasso2, s = lambda2))[2:(p+1)]
alasso1 = glmnet(X, y, family = "gaussian", alpha = 1, intercept = FALSE, penalty.factor = w1)
alasso2 = glmnet(X, y, family = "gaussian", alpha = 1, intercept = FALSE, penalty.factor = w2)
plot(alasso1, xvar = "lambda", label = TRUE, main = "Adaptive lassso - ols")
abline(v=log(lambda1))
plot(alasso2, xvar = "lambda", label = TRUE, main = "Adaptive lassso - ridge")
abline(v=log(lambda2))
```

Topics on High-Dimensional Data Analytics

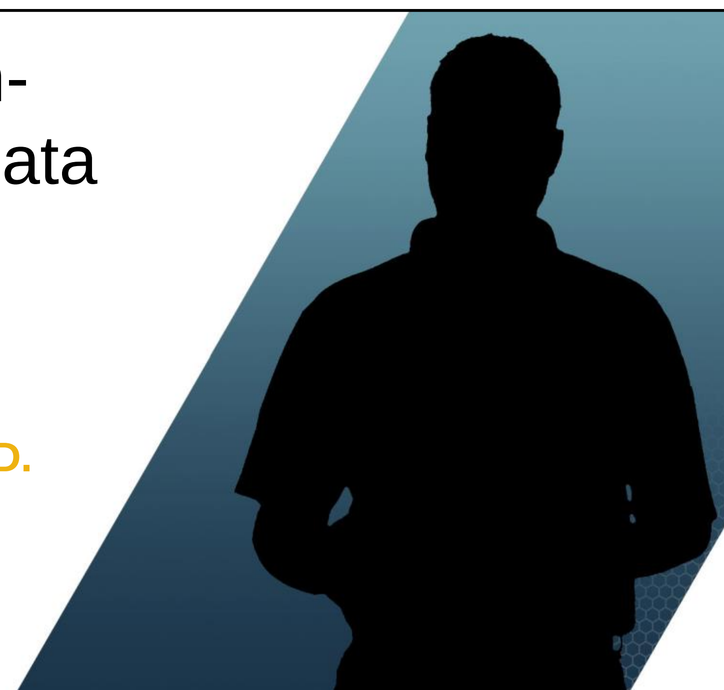
LD Learning using Regularization

Kamran Paynabar, Ph.D.

Associate Professor

School of Industrial and Systems Engineering

Elastic Net



Learning Objectives

- Create Elastic Net penalty.
- Define the benefits of Elastic Net
- Apply Elastic Net for variable selection

Elastic Net

Assuming the observations are centered, Elastic Net estimates can be computed by (Zou and Hastie, 2006)

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$$

- L1 penalty: encourages sparsity
- L2 penalty:
  - Helps perform variable selection when important variable is more than *n*
  - Alleviates the drawback of multicolinearity

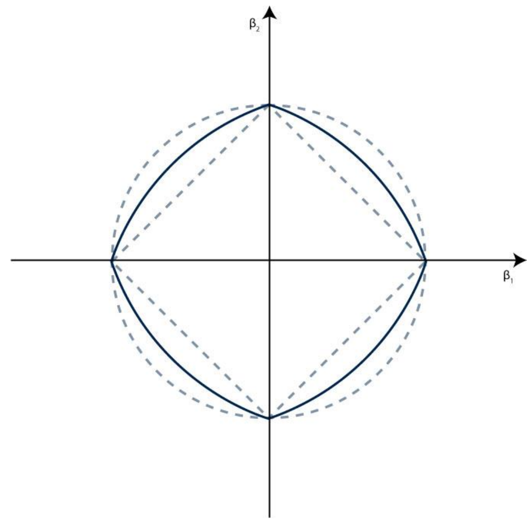
Given λ_2 the solution path of elastic net coefficients is piecewise linear in λ_1

$$\mathbf{X}^*_{(n+p)\times p} = (1+\lambda_2)^{-\frac{1}{2}} \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix}, \quad \mathbf{y}^*_{(n+p)} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix}$$

Elastic Net

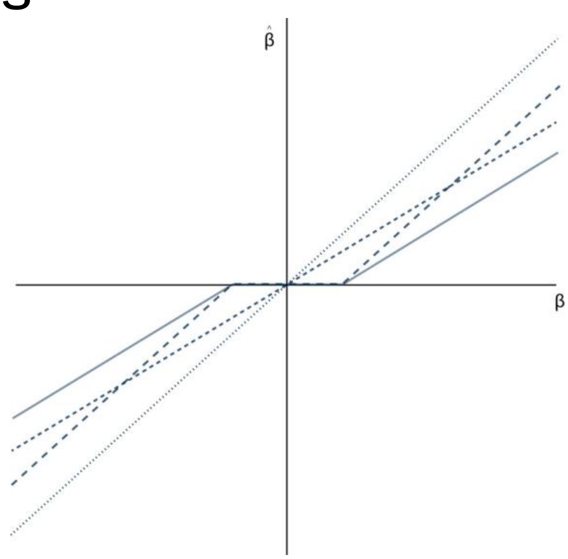
$$\min_{\beta_j} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

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



Elastic Net Estimates

$$\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|_2^2 + \lambda_1 \left\| \boldsymbol{\beta} \right\|_1 + \lambda_1 \left\| \boldsymbol{\beta} \right\|_2^2$$

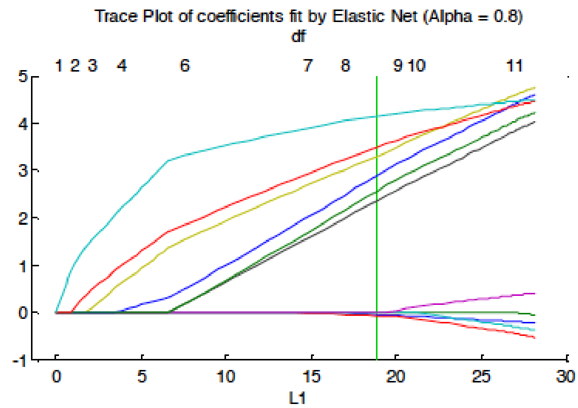


(Zou and Hastie, 2006)

Example

For a regression setting with a 11 predictors, 100 observations are generated such that 5 predictors are not informative.

- Elastic Net is used to identify important variables and fit a prediction model.
- A 10-fold cross validation is used to choose the tuning parameter.



MSE= 0.2714

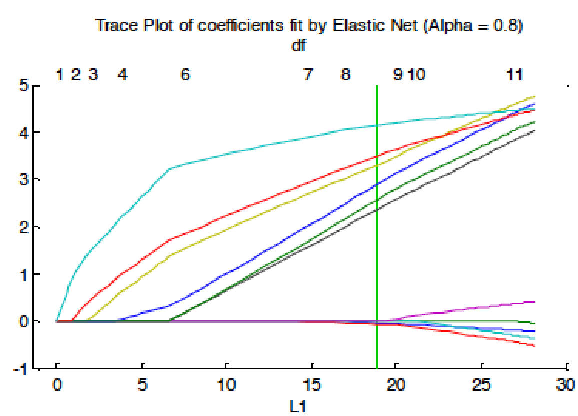
Example

```
clear all
load predictors

for i=1:100
   T(i)=X(i,:)*[0 0 0 0 0 4*ones(1,6)]'+normrnd(0,0.5);
end


[B1,FitInfo1] = lasso(X,T,'CV',10,'Standardize', 0 , 'Alpha' ,0.8)
ax1 = lassoPlot(B1,FitInfo1)

B1(:,FitInfo1.IndexMinMSE)
FitInfo1.MSE(FitInfo1.IndexMinMSE)
```



MSE= 0.2714

Topics on High-Dimensional Data Analytics

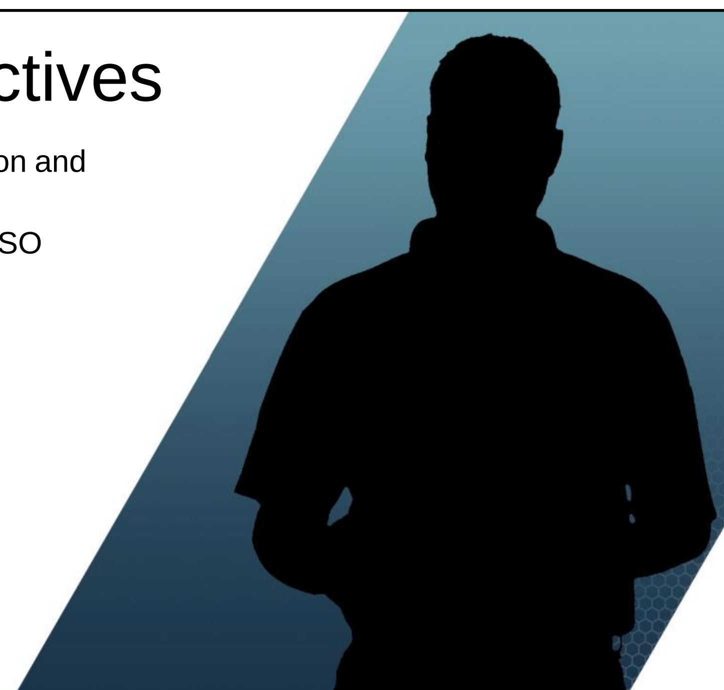
LD Learning using Regularization

Kamran Paynabar, Ph.D.
Associate Professor
School of Industrial and Systems Engineering

Group Lasso

Learning Objectives

- Define group variable selection and sensor selection.
- How to define the group LASSO problem
- Apply group LASSO



Group Variable Selection

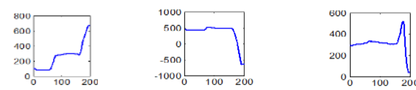
- In some cases, features or variables are naturally partitioned into grouped variables, for example
  - Each factor in ANOVA is a group variable that includes dummy variables representing the levels of the factor.
  - The features extracted from a signal can form a grouped variable.

Extract features from each sensing signal (e.g., using PCA, etc)

↓

Group the features based on their sensors

Select important sensors via group non-negative garrote or group lasso



$$(\underbrace{x_{11},\ldots,x_{1p_1}}_{\mathbf{x}_1},\underbrace{x_{21},\ldots,x_{2p_2}}_{\mathbf{x}_2},\ldots,\underbrace{x_{K1},\ldots,x_{Kp_K}}_{\mathbf{x}_K})$$

Group LASSO

Assuming the observations are centered, group lasso estimates can be computed by (Yuan and Lin, 2008)

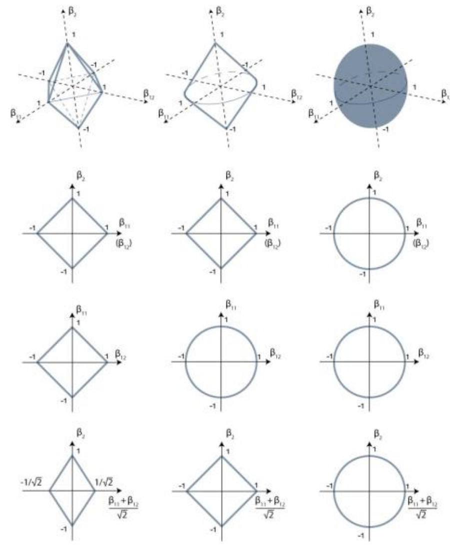
$$\min_{\hat{\boldsymbol{\beta}}_k} \frac{1}{2}\left\|\mathbf{y}-\sum_{k=1}^{K}\mathbf{X}_k\hat{\boldsymbol{\beta}}_k\right\|^2+\lambda_1\sum_{k=1}^{K}\left\|\boldsymbol{\beta}_k\right\|$$

where

$$\left\|\boldsymbol{\beta}_k\right\|=\sqrt{\sum_{j=1}^{p_k}\beta_{kj}^2}$$

$$(\underbrace{x_{11},\ldots,x_{1p_1}}_{\mathbf{X}_1},\underbrace{x_{21},\ldots,x_{2p_2}}_{\mathbf{X}_2},\ldots,\underbrace{x_{K1},\ldots,x_{Kp_K}}_{\mathbf{X}_K})$$

Group LASSO



LASSO | Grouped LASSO | Ridge

(Yuan and Lin, 2008)

Other Group Variable Selection Methods

$$\min_{\hat{\boldsymbol{\beta}}_k} \frac{1}{2}\left\| \mathbf{y} - \sum_{k=1}^{K} \mathbf{X}_k \hat{\boldsymbol{\beta}}_k \right\|^2 + J(\lambda_t, \hat{\boldsymbol{\beta}}_k)$$

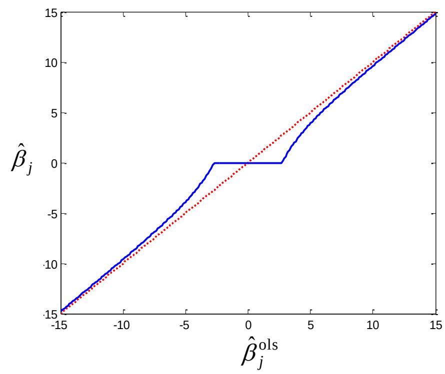
Group Lasso L_2 (Yuan and Lin, 2008) $\quad J(\lambda_t, \beta_{kj}) = \lambda_1 \sum_{k=1}^{K} \|\boldsymbol{\beta}_k\| \quad \|\boldsymbol{\beta}_k\| = \sqrt{\sum_{j=1}^{p_k} \beta_{kj}^2}$

Group Lasso L_∞ (Zhao et al., 2009) $\quad J(\lambda_t, \beta_{kj}) = \lambda_1 \sum_{k=1}^{K} \|\boldsymbol{\beta}_k\|_\infty \quad \|\boldsymbol{\beta}_k\|_\infty = \max_j \{|\beta_{kj}|\}$

Group Non-negative Garrote (Yuan and Lin, 2009) $\quad J(\lambda_t, \beta_{kj}) = \lambda_1 \sum_{k=1}^{K} d_k \quad \boldsymbol{\beta}_k = d_k \hat{\boldsymbol{\beta}}_k^{\text{ols}}; d_k \geq 0$

Group NNG: Orthogonal Basis Case

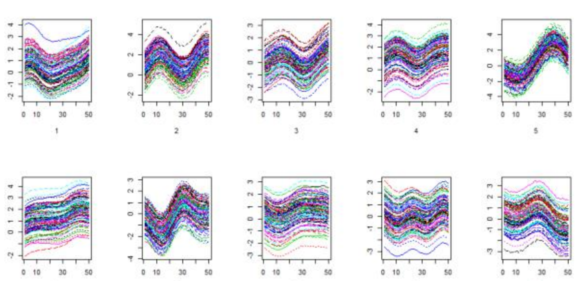
In case of orthogonal predictors (e.g., wavelets) there is a closed-form solution in form of soft-thresholding estimates



$$\hat{d}_k = \left(1 - \lambda \Big/ \sum_{j=1}^{p_k} (\hat{\beta}_{kj}^{\text{ols}})^2\right)_+$$

Example – Function to Scalar Regression

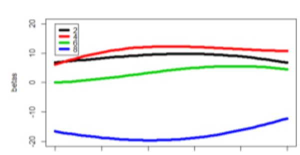
Covariates



Output

$$y_i = \int_0^1 x_{i2}(t)\beta_2(t)dt + \int_0^1 x_{i4}(t)\beta_4(t)dt + \int_0^1 x_{i6}(t)\beta_6(t)dt + \int_0^1 x_{i8}(t)\beta_8(t)dt + \epsilon_i$$

betas



Example – Function to Scalar Regression

$$y_i = \sum_{j=1}^{10} \int_0^1 x_{ij}(t)\beta_j(t)dt + \epsilon_i$$

We can use b-splines to reduce the dimensionality:

$$\beta_j(t) = \sum_{k=1}^{10} b_{kj}\theta_{kj}(t) = \boldsymbol{\theta_j^T b_j}$$

With this, we have:

$$\int_0^1 x_{ij}(t)\beta_j(t)dt = \int_0^1 x_{ij}(t)\boldsymbol{\theta_j^T}(t)dt\boldsymbol{b_j} = \boldsymbol{z_{ij}b_j}$$

Therefore:

$$y_i = \boldsymbol{z_{i1}b_1} + \boldsymbol{z_{i2}b_2} + \cdots + \boldsymbol{z_{i10}b_{10}} + \boldsymbol{\epsilon_i}$$

Our goal is to estimate $\boldsymbol{b_j} \in \mathbb{R}^{10\times 1}$, using group lasso. The problem we want to solve is:

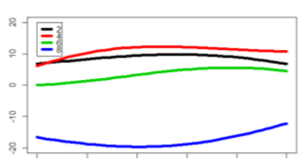
$$\min_{\boldsymbol{b}} ||\boldsymbol{y} - \boldsymbol{Zb}||_2^2 + \sum_{j=1}^{10} \left|\left|\boldsymbol{b_j}\right|\right|_{\boldsymbol{2}}$$

Example – Function to Scalar Regression

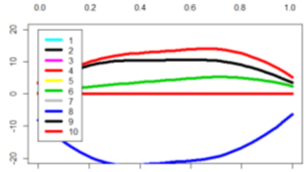
$$y_i = \sum_{j=1}^{10} \int_0^1 x_{ij}(t)\beta_j(t)dt + \epsilon_i$$

Prediction using group lasso:

True



Predicted



Learned B-spline coefficients:

	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10
coef1	0.00	3.43	0.00	3.44	0.00	0.45	0.00	-8.13	0.00	0.00
coef2	0.00	5.05	0.00	5.44	0.00	0.86	0.00	-12.04	0.00	0.00
coef3	0.01	7.78	0.00	8.61	0.00	1.67	0.00	-17.60	0.00	0.00
coef4	0.01	10.52	0.00	12.10	0.00	2.90	0.00	-22.73	0.00	0.00
coef5	0.01	10.50	0.00	12.90	0.00	3.77	0.00	-21.90	0.00	0.00
coef6	0.01	10.53	0.00	13.71	0.00	4.67	0.00	-20.91	0.00	0.00
coef7	0.01	10.71	0.00	14.61	0.00	5.57	0.00	-20.08	0.00	0.00
coef8	0.01	8.01	0.00	11.45	0.00	4.72	0.00	-14.74	0.00	0.00
coef9	0.01	5.23	0.00	7.78	0.00	3.40	0.00	-9.71	0.00	0.00
coef10	0.00	3.48	0.00	5.24	0.00	2.40	0.00	-6.30	0.00	0.00

Example – Function to Scalar Regression

```
# Data generation
p = 10     #Number of parameters
tp = 4    #Number of true parameters
n = 50    #Length of observations
m = 100    #Number of observations
snr = 200
ds = 0.2
```

```
# Covariates
x_1 = list()
for(i in 1:p)
{ x = seq(0,1,length=n)
  E = as.matrix(dist(x, diag=T, upper=T))
  Sigma = exp(-10*E^2)
  eig = eigen(Sigma)
  Sigma.sqrt = eig$vec%*%diag(sqrt(eig$val+10^(-10)))%*%t(eig$vec)
  mean1 = Sigma.sqrt%*%rnorm(n)
  S_noise = exp(-0.1*E^2)
  eig_noise = eigen(S_noise)
  S.sqrt_noise = eig_noise$vec%*%diag(sqrt(eig_noise$val+10^(-10)))%*%t(eig_noise$vec)
  noise = S.sqrt_noise%*%rnorm(n)
  signal = mean1 + noise
  var = var(signal)
  ds1 = sqrt(var/snr)
  S.sqrt_err = diag(n)*drop(ds1)
  x1 = matrix(0,m,n)
  for(j in 1:(m)) {
    noise = S.sqrt_noise%*%rnorm(n)
    error = S.sqrt_err%*%rnorm(n)
    x1[j,] = mean1 + noise + error  }
  x_1[[i]] = x1 }
```