

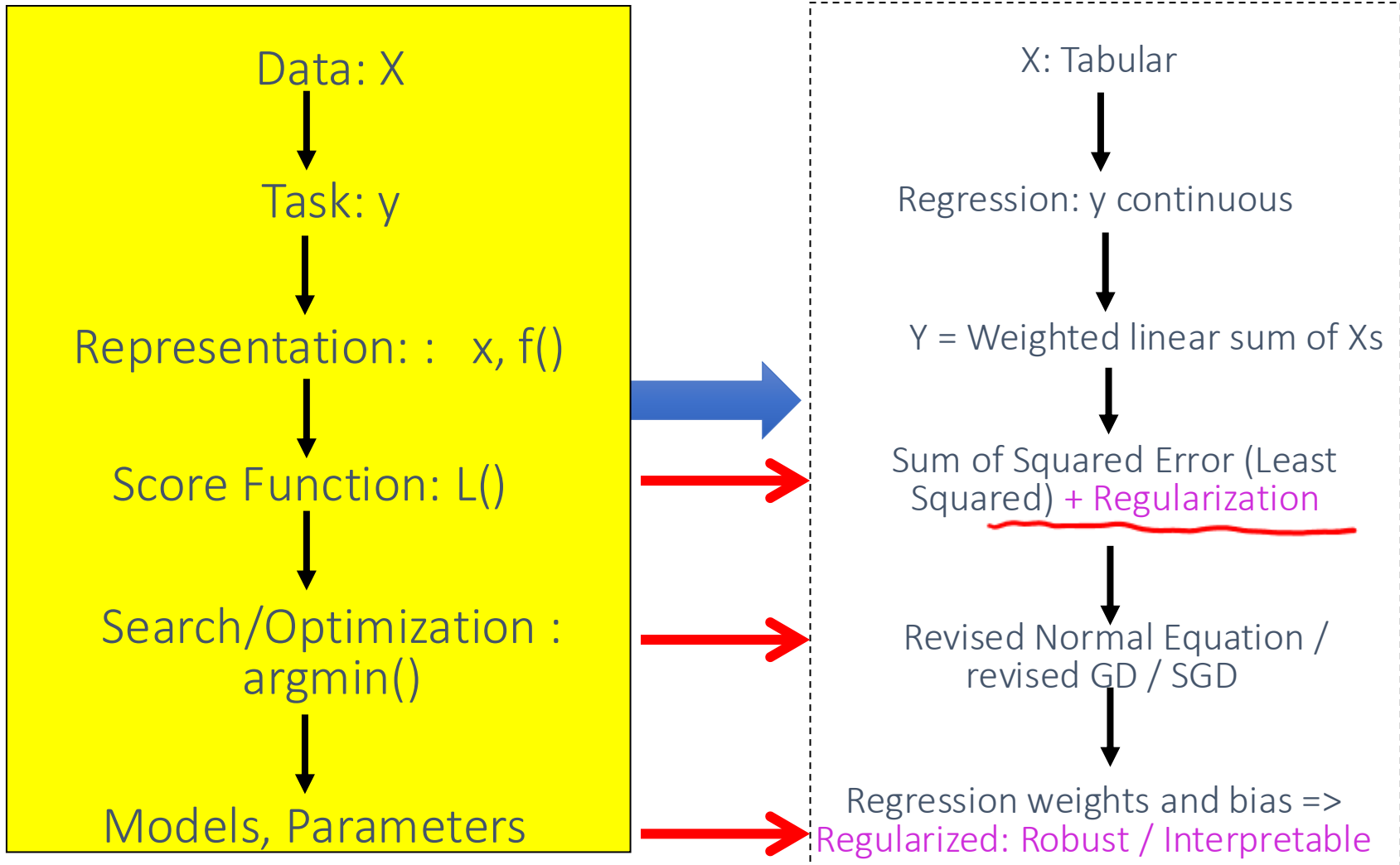
# UVA CS 4774: Machine Learning

## Lecture 7: Linear Regression Model with Regularizations

Dr. Yanjun Qi

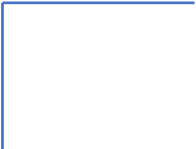
University of Virginia  
Department Of Computer Science

## Today: Regularized multivariate linear regression



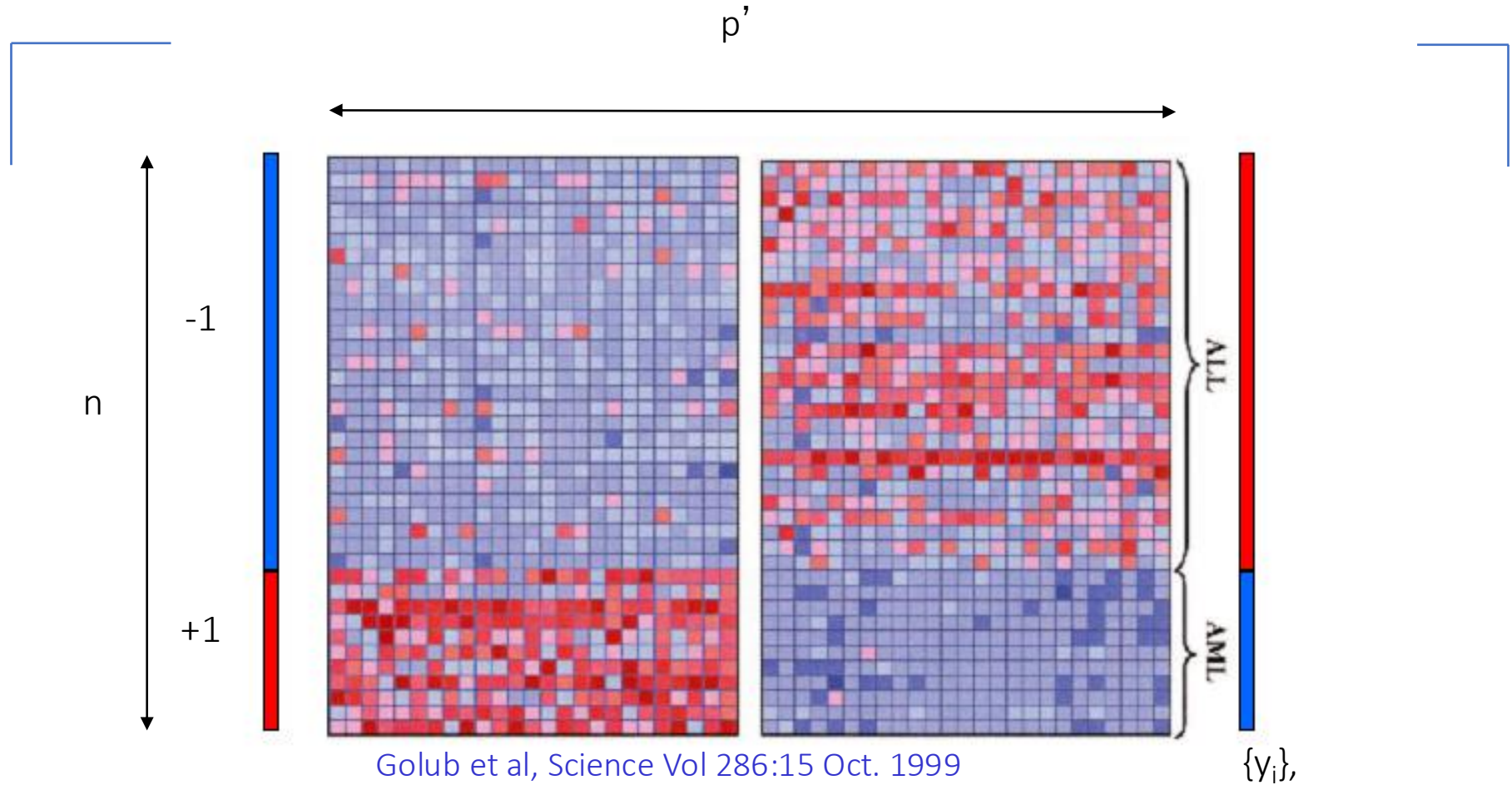
# We aim to make our trained model

- 1. Generalize Well
- 2. Computational Scalable and Efficient
- 3. Trustworthy: Robust / Interpretable
  - Especially for some domains, this is about trust!



Many real-world datasets have  $p$  larger than  $n$

# Example: Gene Expression based Cancer Diagnosis

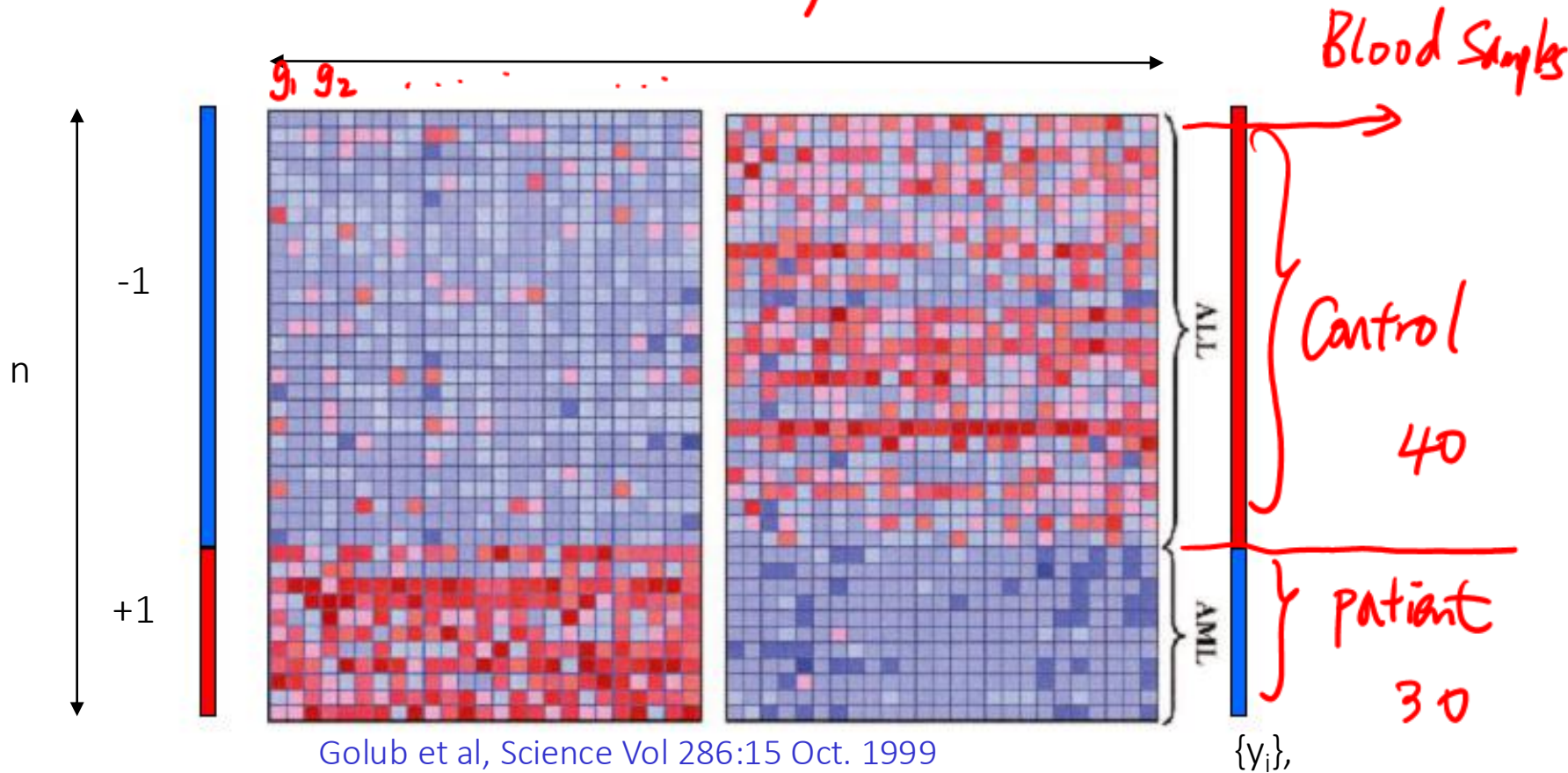


Golub et al, Science Vol 286:15 Oct. 1999

$$p \gg n$$

# Example: Gene Expression based Cancer Diagnosis

$p' \sim 7000$



# Example: Gene Expression based Cancer Diagnosis

- <https://www.kaggle.com/crawford/gene-expression/notebooks>

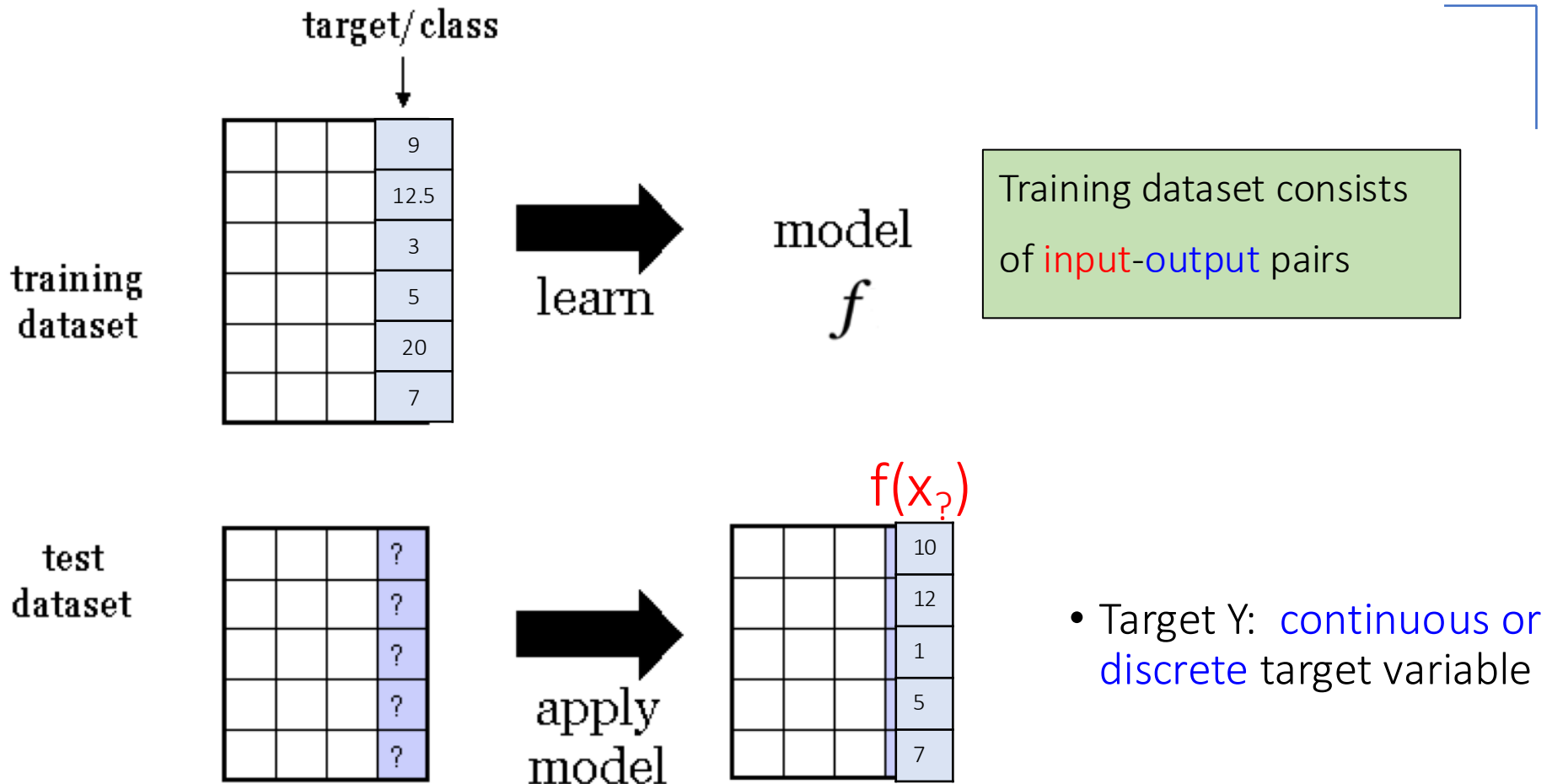
```
# Clean up the column names for Testing data
X_test.columns = X_test.iloc[1]
X_test = X_test.drop(["Gene Description", "Gene Accession Number"]).ap
ply(pd.to_numeric)
```

```
print(X_train.shape)
print(X_test.shape)
X_train.head()
```

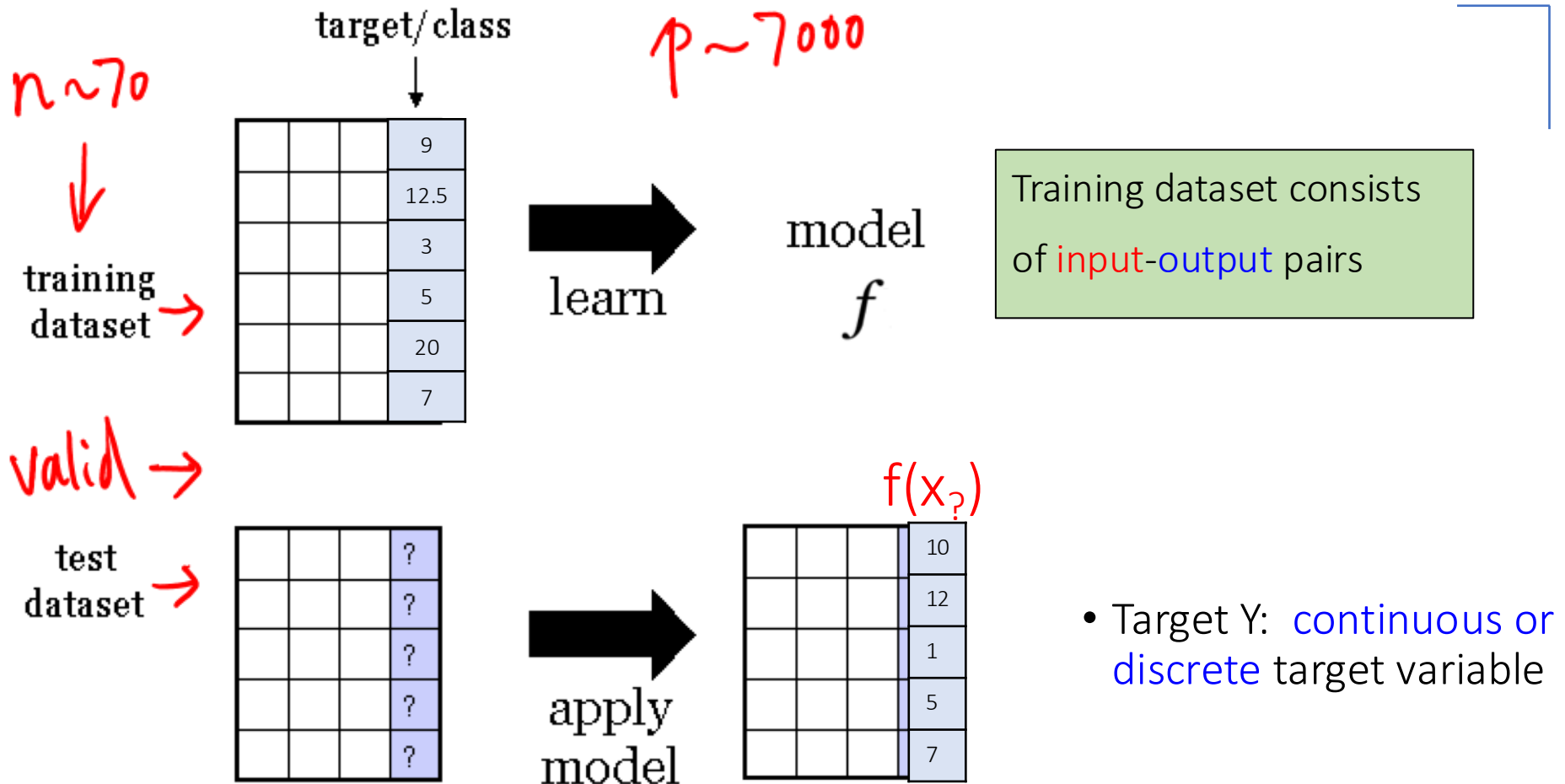
$p \sim 7129$   
 $n \sim 70$

```
(38, 7129)
(34, 7129)
```

# SUPERVISED Prediction Tasks



# SUPERVISED Prediction Tasks



# Summary of **Explicit** Feature Selection Methods:



- Filtering approach:

ranks features or feature subsets **independently of** the predictor.

- ...using **univariate** methods: consider **one** variable at a time
- ...using **multivariate** methods: consider **more than one** variables at a time

- Wrapper approach:

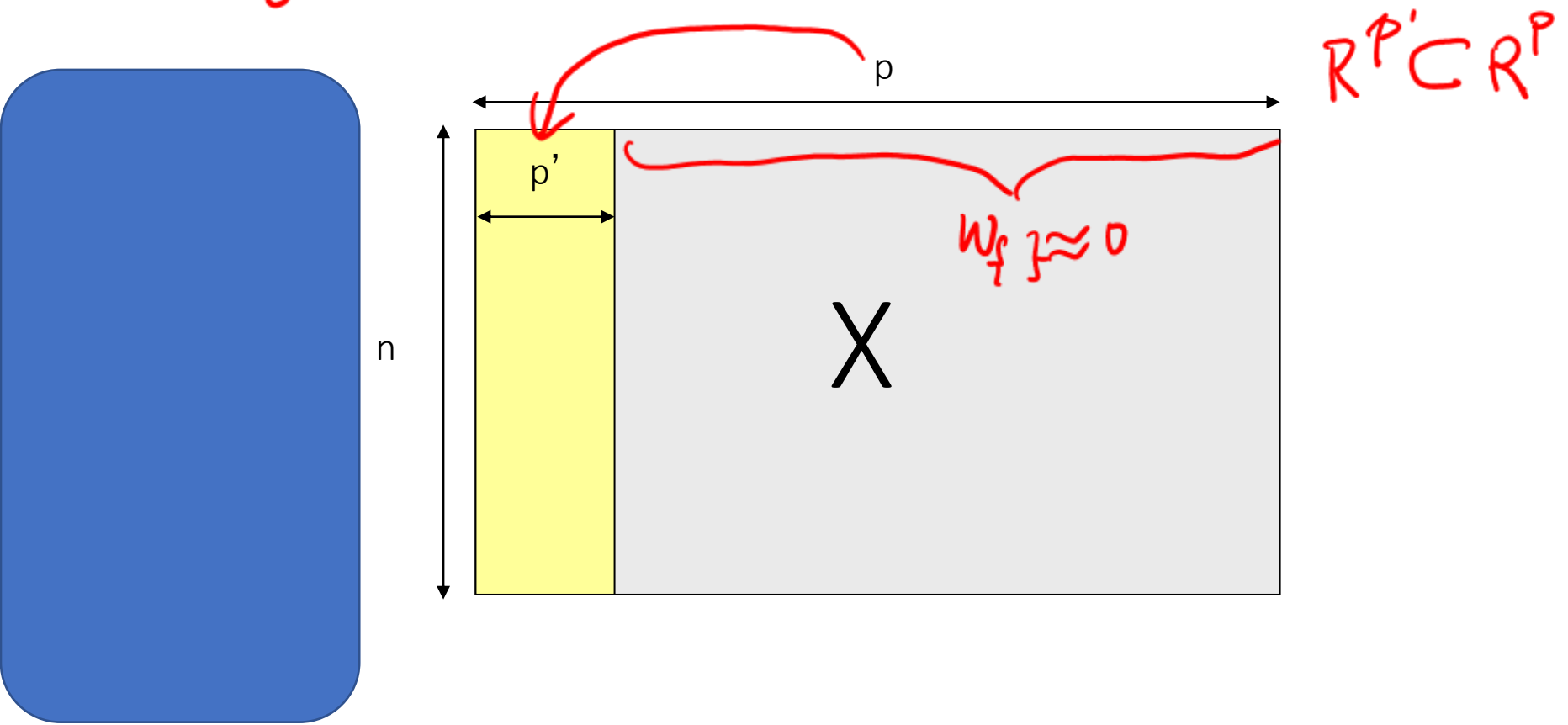
uses a **predictor to assess** features or feature subsets.

- Embedding approach:

uses a **predictor to build** a (single) model with a subset of features that are internally selected.

Large  $p$ , small  $n$ : How?

Regularization = Implicit feature selection



# Another Example: Application of Text Regression

<http://www.cs.cmu.edu/~nasmith/papers/joshi+das+gimpel+smith.naacl10.pdf>

## Movie Reviews and Revenues: An Experiment in Text Regression\*

**Mahesh Joshi   Dipanjan Das   Kevin Gimpel   Noah A. Smith**

Language Technologies Institute

Carnegie Mellon University

Pittsburgh, PA 15213, USA

`{maheshj, dipanjan, kgimpel, nasmith}@cs.cmu.edu`

### Abstract

We consider the problem of predicting a movie's opening weekend revenue. Previous work on this problem has used metadata about a movie—e.g., its genre, MPAA rating, and cast—with very limited work making use of text *about* the movie. In this paper, we use the text of film critics' reviews from several sources to predict opening weekend revenue. We describe a new dataset pairing movie reviews with metadata and revenue data, and show that review text can substitute for metadata, and even improve over it, for prediction.

Proceedings of  
HLT ' Human  
Language  
Technologies:

# I. The Story in Short

- ❖ Use metadata and critics' reviews to predict opening weekend revenues of movies
- ❖ Feature analysis shows what aspects of reviews predict box office success

$n=1,718$

## II. Data

- ❖ 1718 Movies, released 2005-2009
- ❖ Metadata (genre, rating, running time, actors, director, etc.): [www.metacritic.com](http://www.metacritic.com)
- ❖ Critics' reviews (~7K): Austin Chronicle, Boston Globe, Entertainment Weekly, LA Times, NY Times, Variety, Village Voice
- ❖ Opening weekend revenues and number of opening screens: [www.the-numbers.com](http://www.the-numbers.com)

# Predicting Revenue using Text



| Domain                      | train | dev | test | total |
|-----------------------------|-------|-----|------|-------|
| <i>Austin Chronicle</i>     | 306   | 94  | 62   | 462   |
| <i>Boston Globe</i>         | 461   | 154 | 116  | 731   |
| <i>LA Times</i>             | 610   | 2   | 13   | 625   |
| <i>Entertainment Weekly</i> | 644   | 208 | 187  | 1039  |
| <i>New York Times</i>       | 878   | 273 | 224  | 1375  |
| <i>Variety</i>              | 927   | 297 | 230  | 1454  |
| <i>Village Voice</i>        | 953   | 245 | 198  | 1396  |
| # movies                    | 1147  | 317 | 254  | 1718  |



Table 1: Total number of reviews from each domain for the training, development and test sets.

e.g., Movie Reviews and Revenues: An Experiment in Text Regression, (1.7k n / >3k features)

## IV. Features

e.g. counts  
of a ngram in  
the text

**I** Lexical n-grams (1,2,3)

**II** Part-of-speech n-grams (1,2,3)

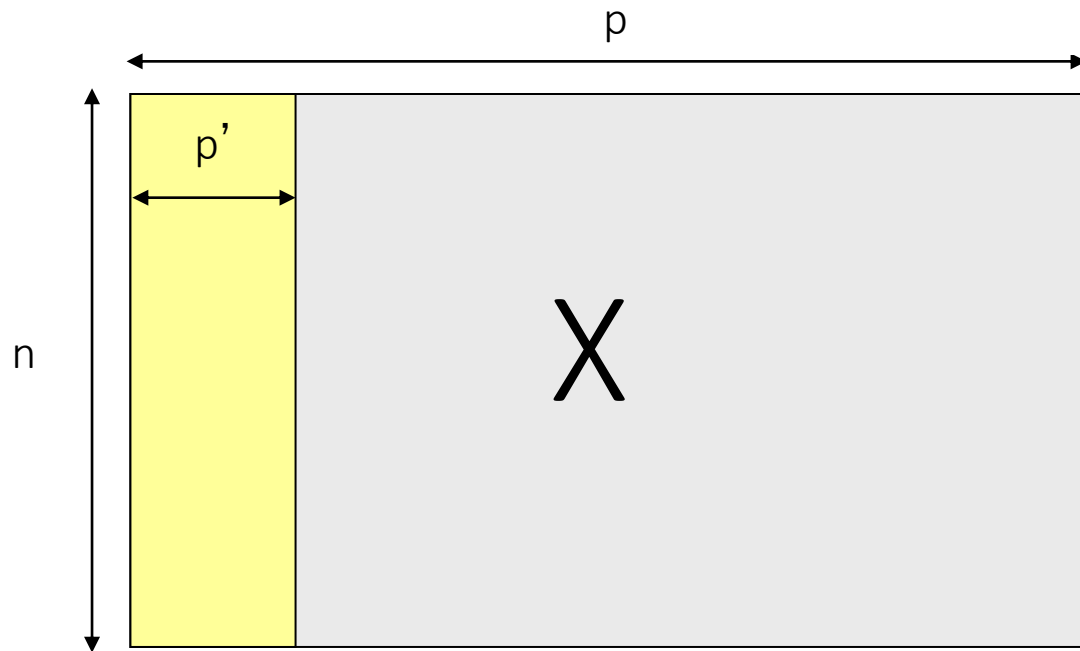
**III** Dependency relations (nsubj,advmod,...)

**Meta**

U.S. origin, running time, budget (log),  
# of opening screens, genre, MPAA  
rating, holiday release (summer,  
Christmas, Memorial day,...), star power  
(Oscar winners, high-grossing actors)

Large  $p$ , small  $n$ : How?

$p \rightarrow p'$



# Regularized multivariate linear regression

• Model:  $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$

- LR estimation:

$$\arg \min_{\beta} \sum \left( Y - \hat{Y} \right)^2$$

- LASSO estimation:

$$\arg \min \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Ridge regression estimation:

$$\arg \min_{\beta} \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Error on data

+

Regularization

17/54

# Regularized multivariate linear regression

• Model:  $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$

- LR estimation:

$$\arg \min_{\beta} \sum \left( Y - \hat{Y} \right)^2$$

- LASSO estimation:

$$\arg \min \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Ridge regression estimation:

$$\arg \min_{\beta} \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Error on data

+

Regularization

18/54

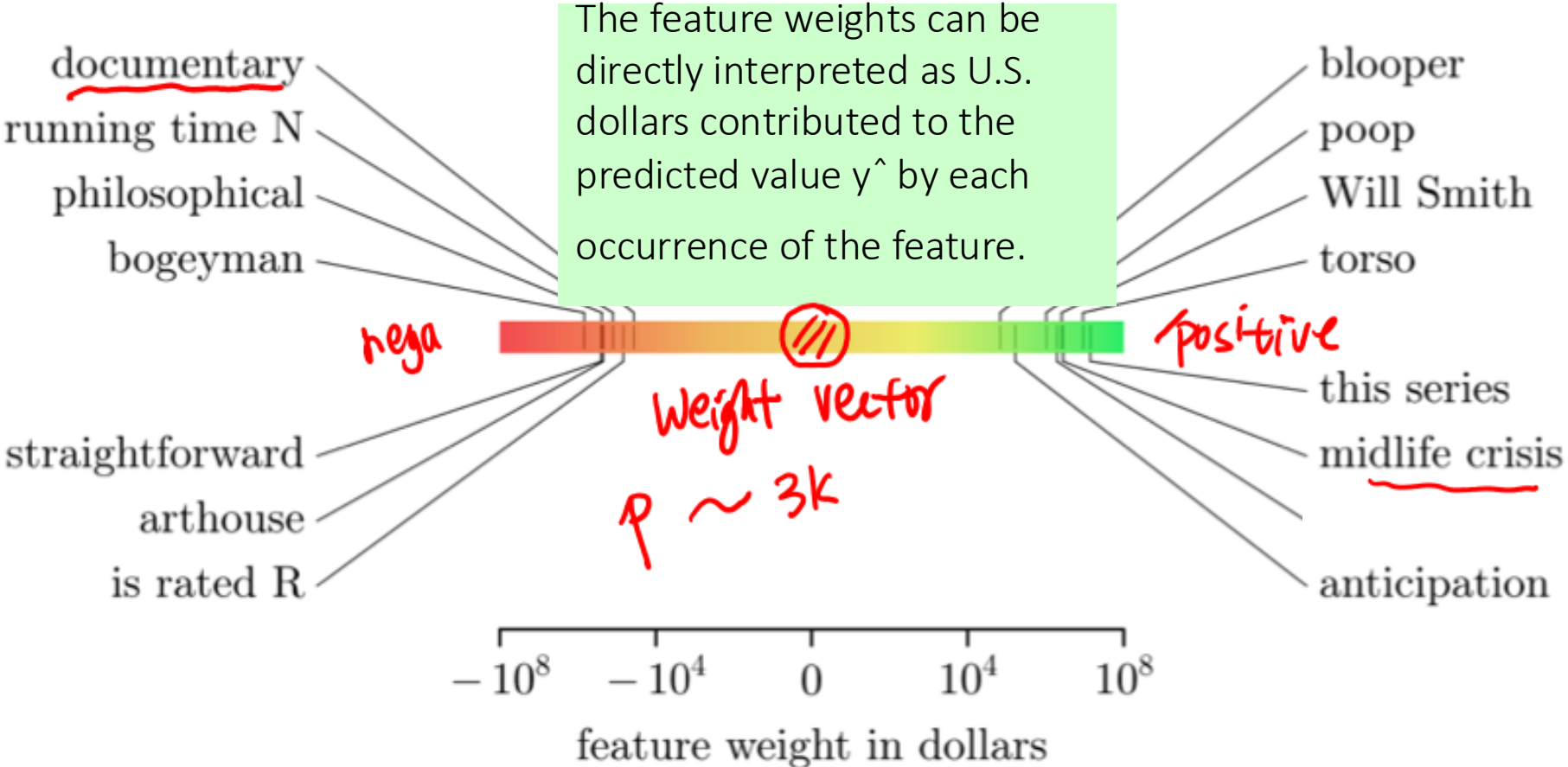
Early Application: Helping use  
movie Reviews and meta data to  
Predict Revenues

VIII. Get the Data!

[www.ark.cs.cmu.edu/movie\\$-data](http://www.ark.cs.cmu.edu/movie$-data)

V. What May Have Brought You

to movies



### III. Model

- ❖ Linear regression with the **elastic net** (Zou and Hastie, 2005)

$$\hat{\theta} = \underset{\theta=(\beta_0, \beta)}{\operatorname{argmin}} \frac{1}{2n} \sum_{i=1}^n \underbrace{\left( y_i - (\beta_0 + \mathbf{x}_i^\top \beta) \right)^2}_{\text{loss}} + \lambda P(\beta)$$

$\sum (y_i - \hat{y}_i)$  (red handwritten above the sum)  
 $\left( y_i - (\beta_0 + \mathbf{x}_i^\top \beta) \right)$  (red handwritten "loss" below the term)  
 $\beta_0$  (red handwritten "Bias" with an arrow pointing to it)  
 $\beta$  (red handwritten "weight" with an arrow pointing to it)

$$P(\beta) = \sum_{j=1}^p \left( \frac{1}{2} (1 - \alpha) \beta_j^2 + \alpha |\beta_j| \right)$$

$\|\vec{\beta}\|_2^2 = \sum_{i=1}^p (\beta_i)^2$  (red handwritten)  
 $\|\vec{\beta}\|_1 = \sum_{i=1}^p |\beta_i|$  (red handwritten)

Use linear regression to directly predict the opening weekend gross earnings, denoted as  $y$ , based on features  $x$  extracted from the movie metadata and/or the text of the reviews.

# Thank You



# UVA CS 4774: Machine Learning

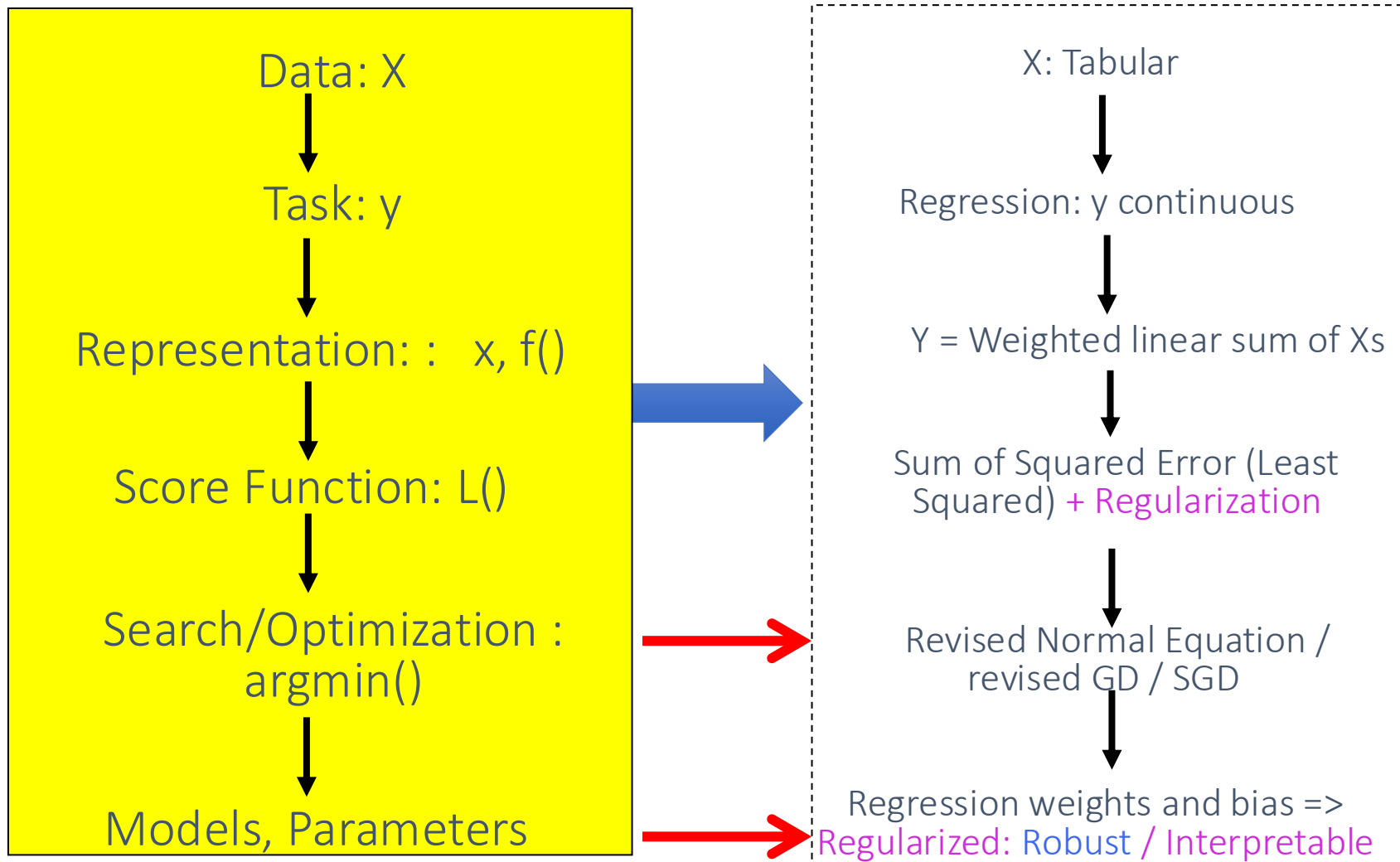
## Lecture 7: Linear Regression Model with Regularizations

Module (2)

Dr. Yanjun Qi

University of Virginia  
Department of Computer Science

# Today: Regularized multivariate linear regression



# Review: Normal equation for LR

- Write the cost function in matrix form:

$$J(\beta) = \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i^T \beta - y_i)^2$$

$$= \frac{1}{2} (X\beta - \bar{y})^T (X\beta - \bar{y})$$

$$= \frac{1}{2} (\beta^T X^T X \beta - \beta^T X^T \bar{y} - \bar{y}^T X \beta + \bar{y}^T \bar{y})$$

$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n^T & -- \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

To minimize  $J(\theta)$ , take derivative and set to zero:

$\Rightarrow$

$$X^T X \beta = X^T \bar{y}$$

The normal equations

$$\beta^* = \left( X^T X \right)^{-1} X^T \bar{y}$$

Dr. Yanjun Qi / UVA CS

Assume  
that  $X^T X$  is  
invertible

What if  $X$  has less than full column rank?  $\Rightarrow$  Not Invertible

$n < p$  :  
 $X^T X$  not invertible

$$\underbrace{X^T X}_{p \times n \quad n \times p} \in \underbrace{\mathbb{R}^{p \times p}}_{\text{full Rank} = p}$$
$$\text{Rank}(X^T X) \leq \min(\text{rank}(X^T), \text{rank}(X))$$
$$\text{Rank}(X^T X) \leq \text{rank}(X) \leq \underbrace{\min(n, p)}_{\text{when } n < p} \leq n$$

For any matrix  $A \in \mathbb{R}^{m \times n}$ , it turns out that the column rank of  $A$  is equal to the row rank of  $A$  (though we will not prove this), and so both quantities are referred to collectively as the **rank** of  $A$ , denoted as  $\text{rank}(A)$ . The following are some basic properties of the rank:

- For  $A \in \mathbb{R}^{m \times n}$ ,  $\text{rank}(A) \leq \min(m, n)$ . If  $\text{rank}(A) = \min(m, n)$ , then  $A$  is said to be **full rank**. ③
- For  $A \in \mathbb{R}^{m \times n}$ ,  $\text{rank}(A) = \text{rank}(A^T)$ . ②
- For  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$ . ①

What if  $X$  has less than full column rank?  $\Rightarrow$  Not Invertible

$n < p$  :  $X^T X$  not invertible

$X^T X \in \mathbb{R}^{p \times p}$  full rank =  $p$

$\text{Rank}(X^T X) \leq \min(\text{rank}(X^T), \text{rank}(X))$

$\text{Rank}(X^T X) \leq \text{rank}(X) \leq \min(n, p) \leq n$

when  $n < p$

# Ridge Regression / L2 Regularized Regression

$$\beta^* = (X^T X)^{-1} X^T \bar{y}$$



- If not **invertible**, a classical solution is to add a small positive element to diagonal

$$\beta^* = (X^T X + \lambda I)^{-1} X^T \bar{y}$$

# Ridge Regression / L2 Regularized Regression

$$\beta^* = \left( X^T X + \lambda I \right)^{-1} X^T \bar{y}$$

- Is the solution of

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n \left( Y_i - \hat{Y}_i \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

to minimize, take derivative and set to zero  
*gradient*

# Parameter Shrinkage by Ridge

$$\beta_{OLS} = \left( X^T X \right)^{-1} X^T \bar{y}$$

$$\beta_{Rg} = \left( \underline{X^T X} + \lambda I \right)^{-1} X^T \bar{y}$$

# Ridge Regression: Squared Loss+L2 penalty on weights

- $\lambda > 0$  penalizes each  $\beta_j$

$$\beta_{rg} \approx \underbrace{\frac{1}{1+\lambda}}_{>0} \beta_{OLS} \Rightarrow |\beta_{rg}| < |\beta_{OLS}|$$

- if  $\lambda = 0$  we get the least squares estimator;
- if  $\lambda \rightarrow \infty$ , then  $\beta_j$  to zero

# Summary: Parameter Shrinkage by Ridge

$$\beta_{OLS} = (X^T X)^{-1} X^T \bar{y}$$

When  $X^T X = I$

$$\beta_{OLS} = X^T y$$

$$\beta_{Rg} = (X^T X + \lambda I)^{-1} X^T \bar{y}$$
$$(I + \lambda I)^{-1}$$

When  $X^T X = I$

$$\beta_{Rg} = \frac{1}{1 + \lambda} X^T y = \frac{1}{1 + \lambda} \beta_{OLS}$$

When  $X^T X = I \Rightarrow \beta_{Rg} = \frac{1}{1 + \lambda} \beta_{OLS}$

Shrinkage

$$\lambda > 0$$

When  $X^T X$  is general: see advanced analysis.

Page 65 of ESL book @

[http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII\\_print10.pdf](http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf)

# Regularized multivariate linear regression

• Model:  $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$

- LR estimation:

$$\arg \min_{\beta} \sum_{i=1}^n \left( Y_i - \hat{Y}_i \right)^2$$

- LASSO estimation:

$$\arg \min_{\beta} \sum_{i=1}^n \left( Y_i - \hat{Y}_i \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Ridge regression estimation:

$$\arg \min_{\beta} \sum_{i=1}^n \left( Y_i - \hat{Y}_i \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$$

Error on data

+

Regularization

32/54

# Lasso (least absolute shrinkage and selection operator)

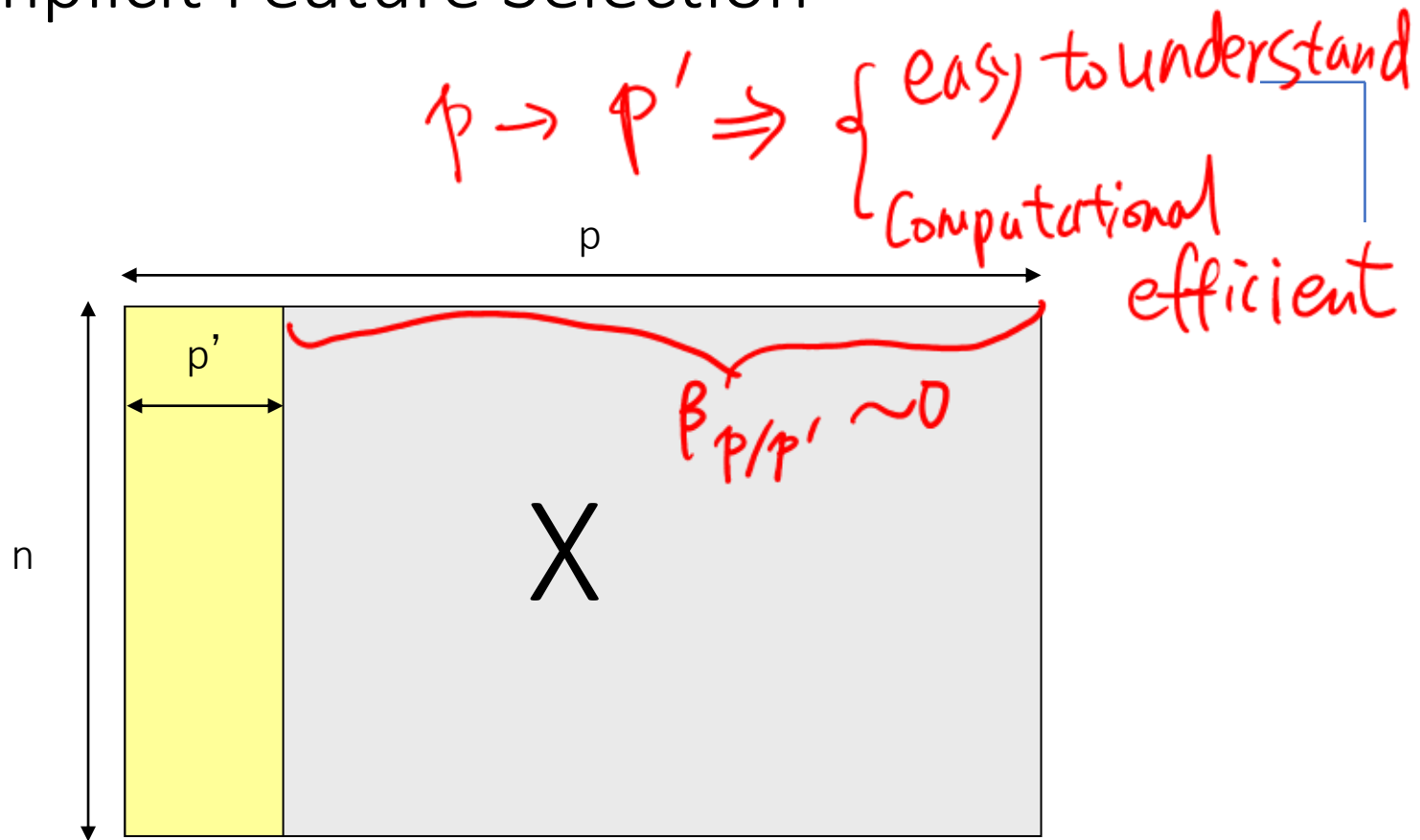
- Let us change that ridge penalty  $\sum \beta_j^2$

- Be replaced by  $\sum |\beta_j|$

- Due to the nature of the constraint, if tuning parameter is chosen large enough, then the lasso will set some coefficients exactly to zero.

$$\hat{\beta}^{lasso} = \operatorname{argmin} \sum_{i=1}^n \left( Y_i - \hat{Y}_i \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$$

# Lasso: Implicit Feature Selection



- LASSO does **shrinkage and variable selection** simultaneously for better prediction and model interpretation.

A regularizer tries to keep the parameters more normal/regular

# Common Regularizers

L2: Squared weights penalizes large values more

L1: Sum of weights will penalize small values more

$$\sum_j |\beta_j|$$

$$\sum_j \beta_j^2$$

Generally, we don't want huge weights

If weights are large, a small change in a feature can result in a large change in the prediction

Might also prefer weights of 0 for features that aren't so useful

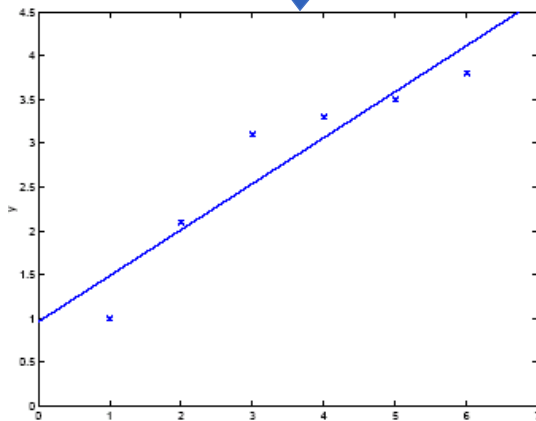
# How to Select $\lambda$ :

## Model Selection & Generalization

- **Generalisation**: learn function / hypothesis from **past data** in order to “explain”, “predict”, “model” or “control” **new** data examples
- **Underfitting**: when model is too simple, both training and test errors are large
- **Overfitting**: when model is too complex and test errors are large although training errors are small.
  - After learning knowledge, model tends to learn “**noise**”

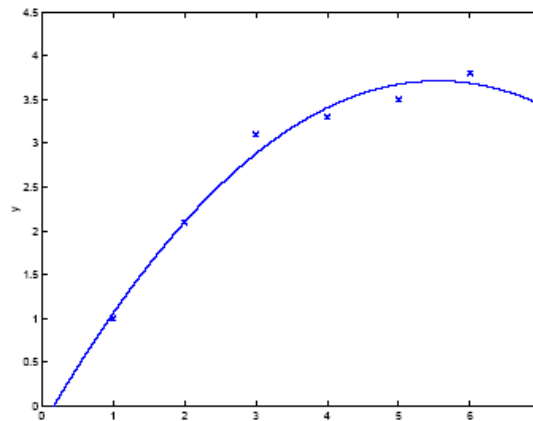
# Issue: Overfitting and underfitting

Under fit



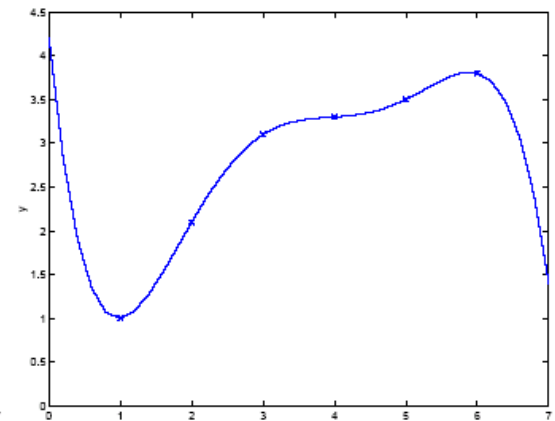
$$y = \theta_0 + \theta_1 x$$

Looks good



$$y = \theta_0 + \theta_1 x + \theta_2 x^2$$

Over fit



$$y = \sum_{j=0}^5 \theta_j x^j$$

Generalisation: learn function / hypothesis from past data in order to “explain”, “predict”, “model” or “control” new data examples

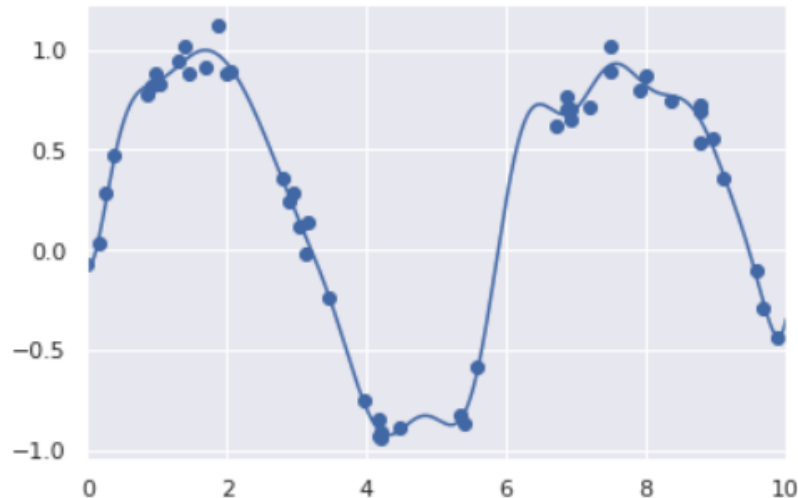
K-fold Cross Validation !!!!

<https://colab.research.google.com/drive/1BVcHUBYDO4AlwldcKmpmj5blAbf7lSJb?usp=sharing>

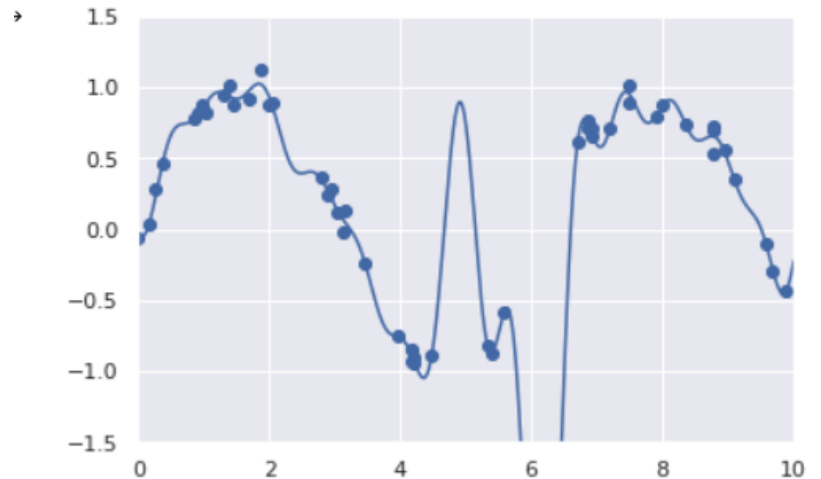
Modified from:

<https://jakevdp.github.io/PythonDataScienceHandbook/05.06-linear-regression.html>

```
gauss_model = make_pipeline(GaussianFeatures(20),  
                             LinearRegression())  
gauss_model.fit(x[:, np.newaxis], y)  
yfit = gauss_model.predict(xfit[:, np.newaxis])  
  
plt.scatter(x, y)  
plt.plot(xfit, yfit)  
plt.xlim(0, 10);
```

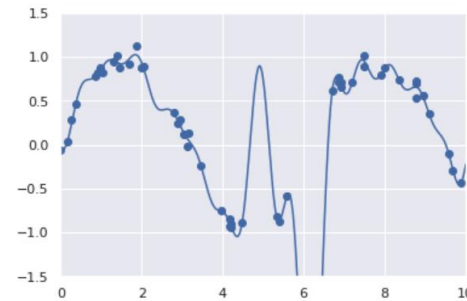


```
model = make_pipeline(GaussianFeatures(30),  
                       LinearRegression())  
model.fit(x[:, np.newaxis], y)  
  
plt.scatter(x, y)  
plt.plot(xfit, model.predict(xfit[:, np.newaxis]))  
  
plt.xlim(0, 10)  
plt.ylim(-1.5, 1.5);
```

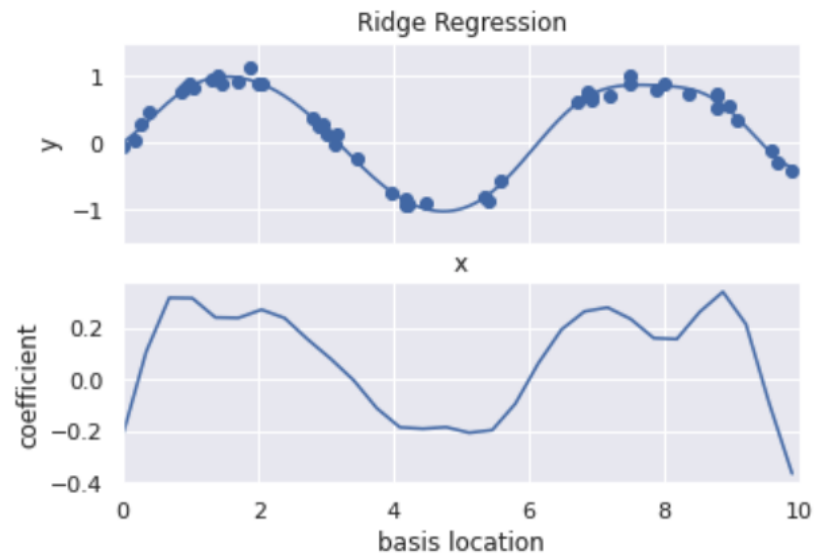


# Overfitting: Can be Handled by Regularization

A **regularizer** is an additional criteria to the loss function to make sure that we don't overfit. It's called a **regularizer** since it tries to keep the parameters more normal/regular



```
from sklearn.linear_model import Ridge
model = make_pipeline(GaussianFeatures(30), Ridge(alpha=0.1))
basis_plot(model, title='Ridge Regression')
```



# WHY and How to Select $\lambda$ ?

- 1. Generalization ability  
    ➔ k-folds CV to decide
- 2. Control the bias and Variance of the model (details in future lectures)

L2: Squared weights penalizes large values more

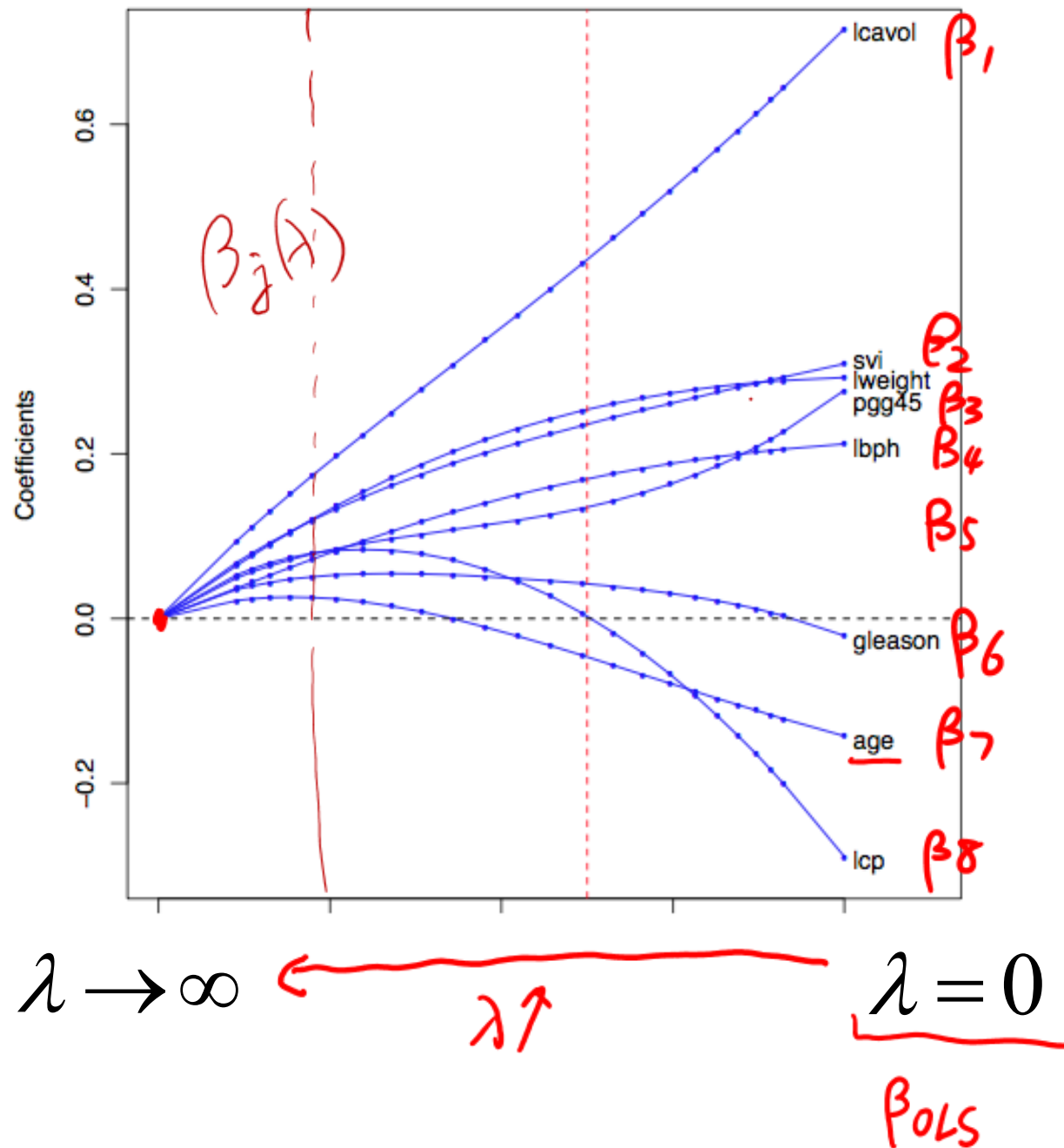
L1: Sum of weights will penalize small values more

$$\sum_j \left| \beta_j \right|$$

$$\sum_j \beta_j^2$$

# Regularization path of a Ridge Regression

when varying  $\lambda$ ,  
how  $\beta_j$  varies.

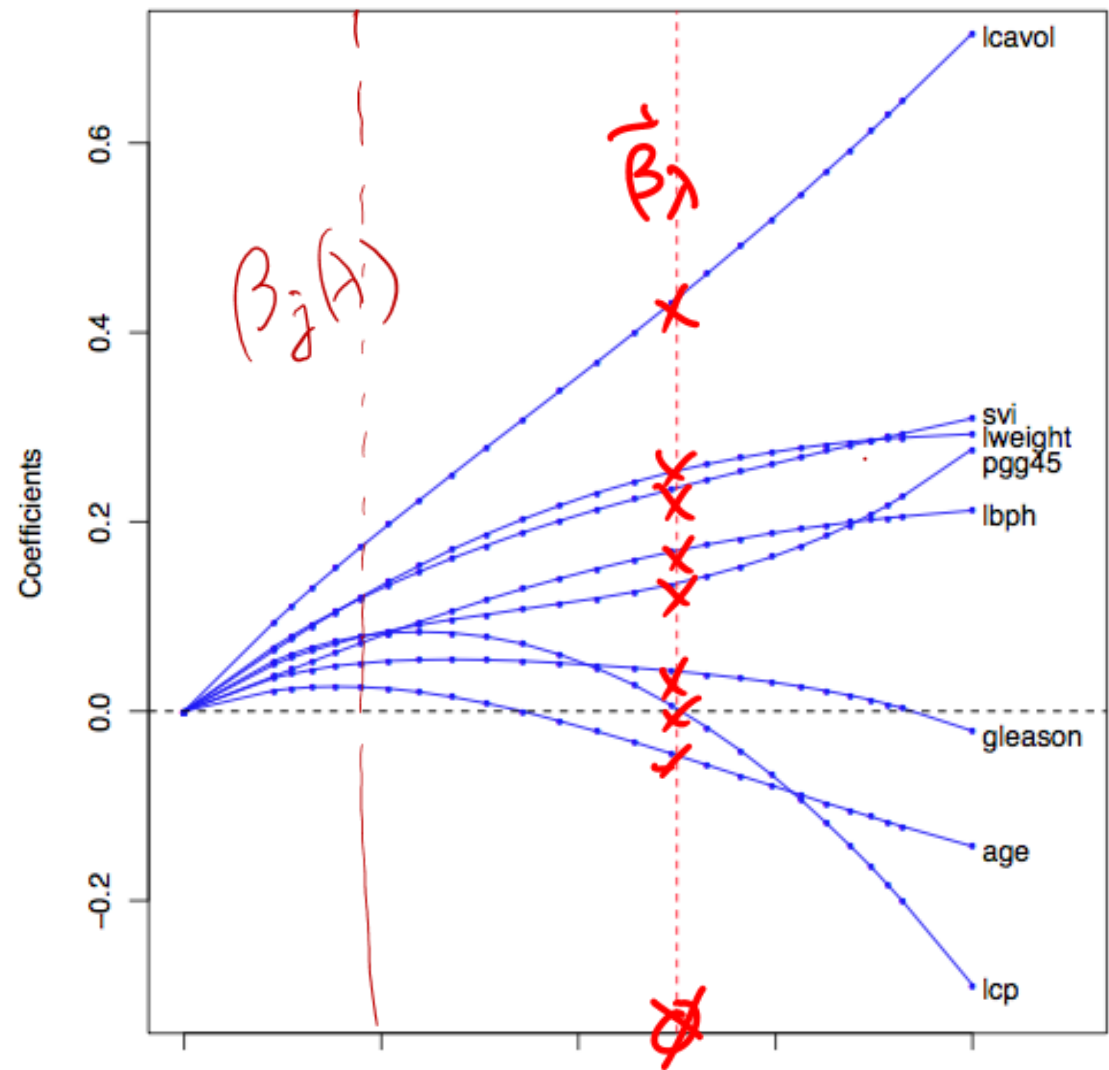


# Regularization path of a Ridge Regression

When  $X^T X = I$

$$\beta_{Rg} = \frac{1}{1 + \lambda} \beta_{OLS}$$

Weight Decay

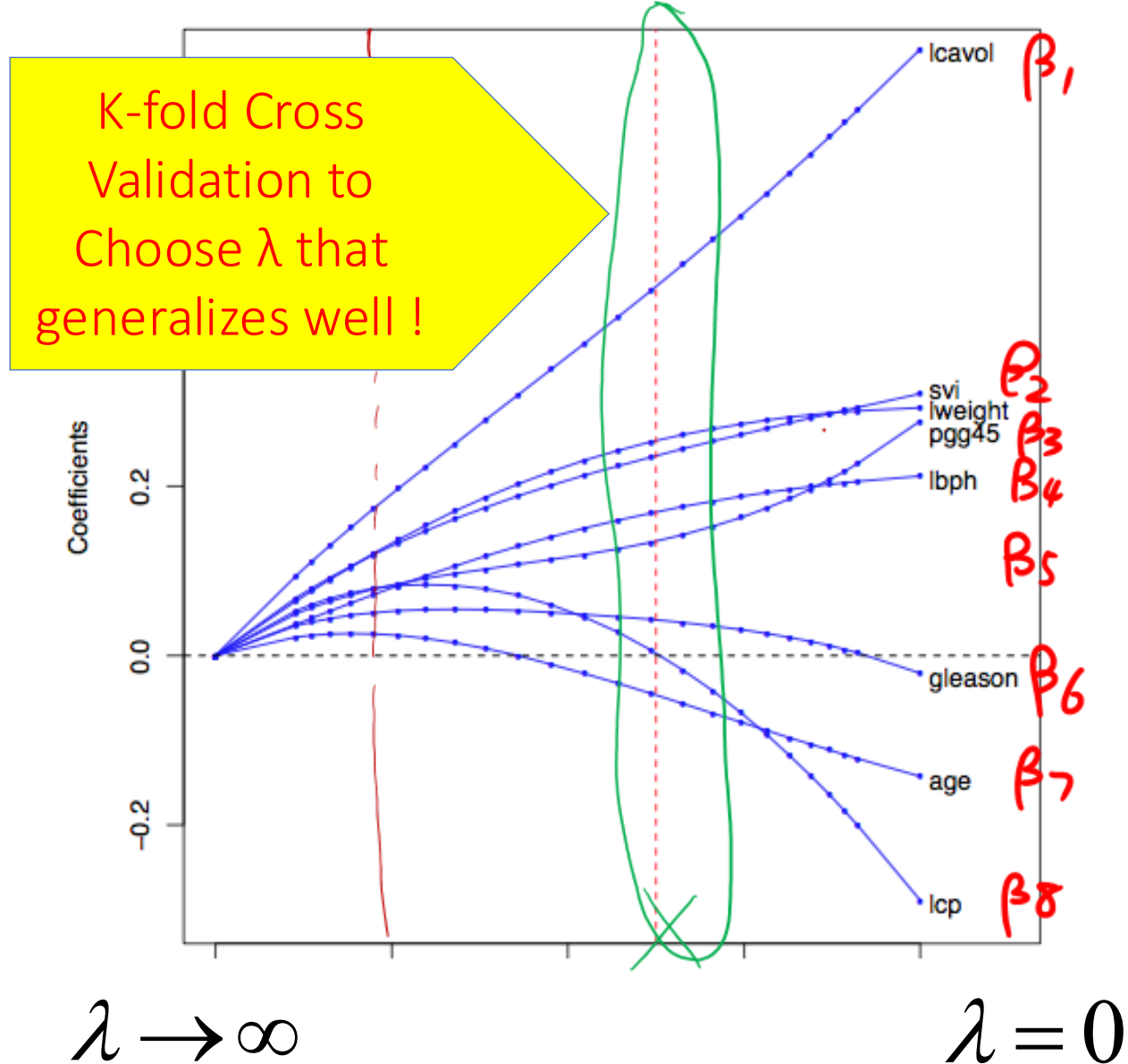


$\lambda \rightarrow \infty$

$\lambda = 0$

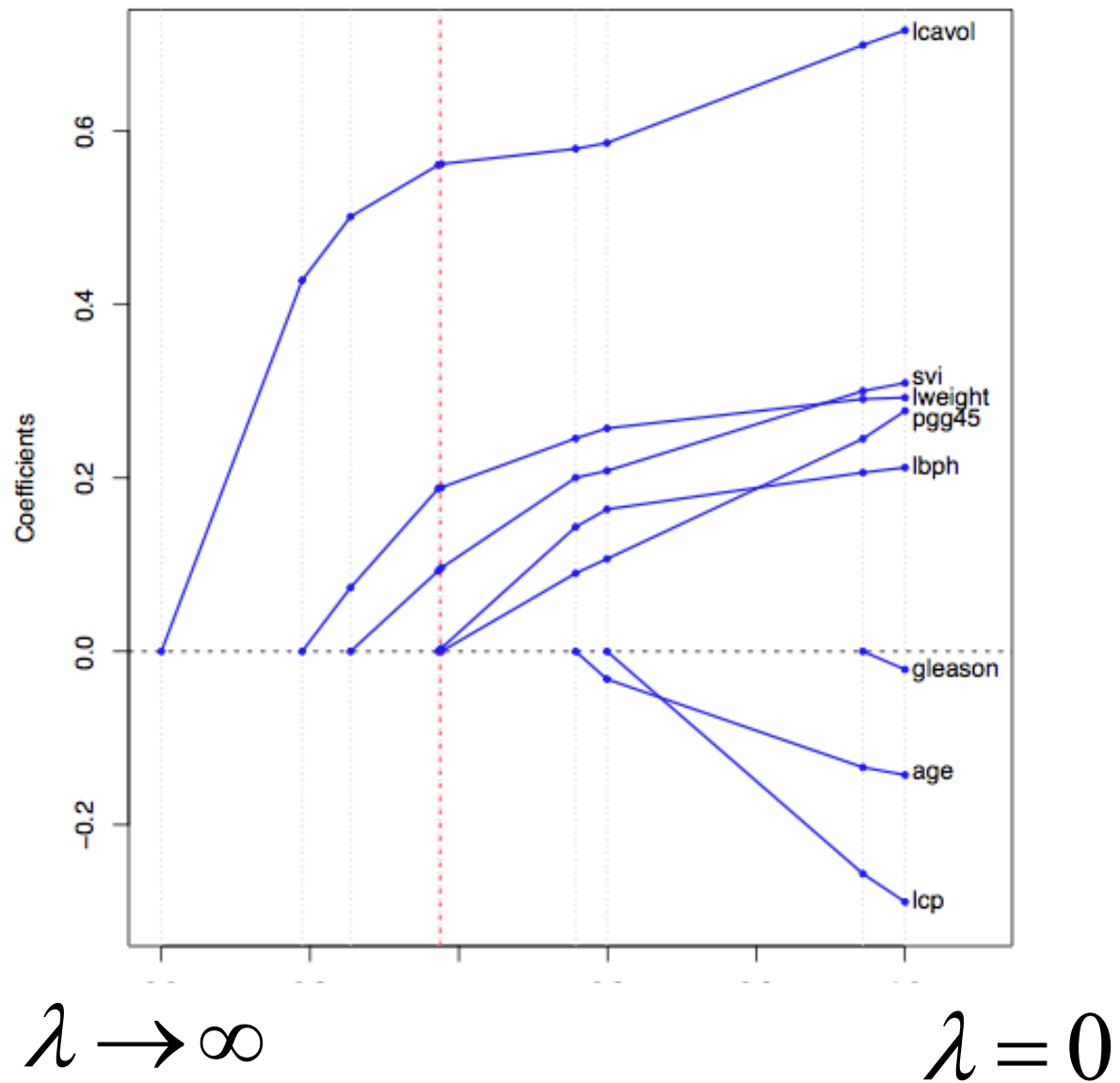
# Regularization path of a Ridge Regression

when varying  $\lambda$ ,  
how  $\beta_j$  varies.



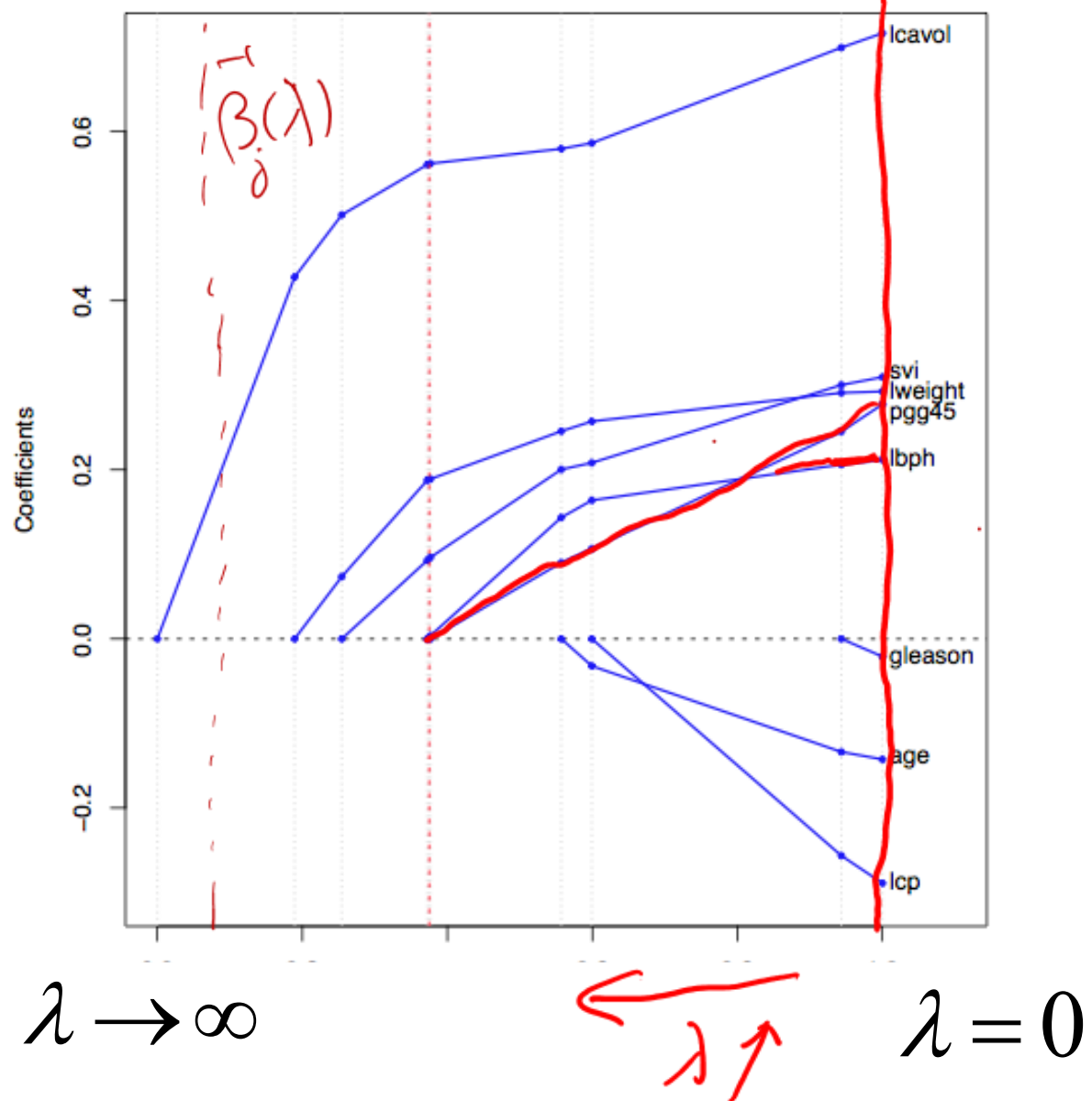
# Regularization path of a Lasso Regression

when varying  $\lambda$ ,  
how  $\beta_j$  varies.



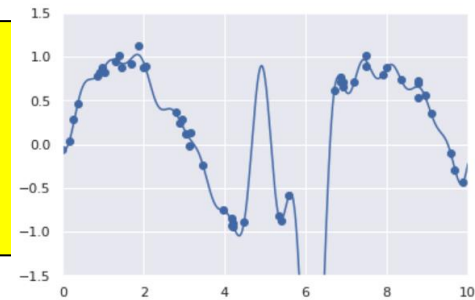
# Regularization path of a Lasso Regression

when varying  $\lambda$ ,  
how  $\beta_j$  varies.



# Overfitting: Can be Handled by Regularization

A **regularizer** is an additional criteria to the loss function to make sure that we don't overfit. It's called a **regularizer** since it tries to keep the parameters more normal/regular



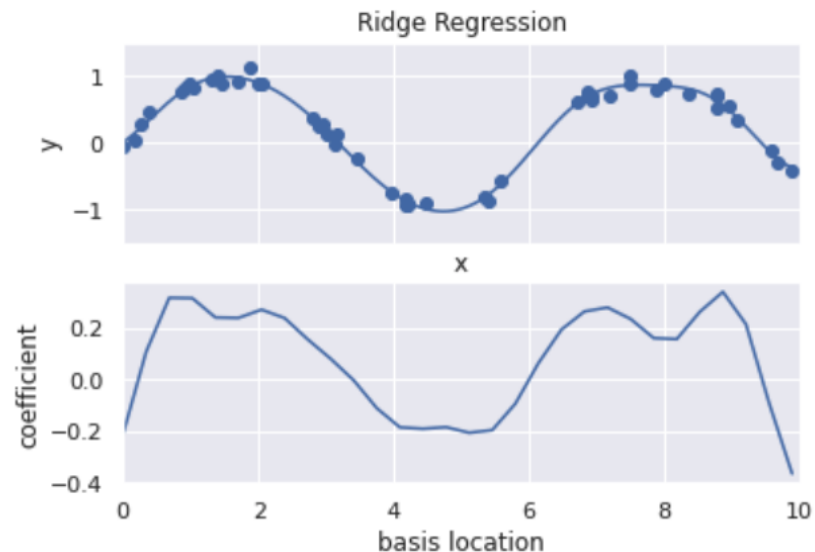
I will code-run:

<https://colab.research.google.com/drive/16LCQGg5Be6XH5yq9NwoVXcOFoAZIgQN?usp=sharing>

Adapted from:

<https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.06-Linear-Regression.ipynb#scrollTo=TNA3vumSulUH>

```
from sklearn.linear_model import Ridge
model = make_pipeline(GaussianFeatures(30), Ridge(alpha=0.1))
basis_plot(model, title='Ridge Regression')
```



# Thank You



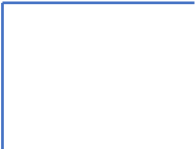
# UVA CS 4774: Machine Learning

## Lecture 7: Linear Regression Model with Regularizations

Module (3)

Dr. Yanjun Qi

University of Virginia  
Department of Computer Science



# Extra on Ridge and Lasso formulation and Geometric Interpretations

# Roadmap: Linear Regression with Regularizations

- 
- ✓ When  $p > n$ : How is Ordinary Least squares?
  - ✓ Ridge regression: squared loss with L2 regularization
  - ✓ Lasso regression: squared loss with L1 regularization
  - ✓ Elastic regression: squared loss with L1 AND L2 regularization
  - ✓ How to Choose Regularization Parameter

# Review: Normal equation for LR

- Write the cost function in matrix form:

$$\begin{aligned} J(\beta) &= \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i^T \beta - y_i)^2 \\ &= \frac{1}{2} (X\beta - \bar{y})^T (X\beta - \bar{y}) \\ &= \frac{1}{2} (\beta^T X^T X \beta - \beta^T X^T \bar{y} - \bar{y}^T X \beta + \bar{y}^T \bar{y}) \end{aligned}$$
$$\mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n^T & -- \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

To minimize  $J(\theta)$ , take derivative and set to zero:

$$\Rightarrow \boxed{X^T X \beta = X^T \bar{y}}$$

The normal equations

$$\beta^* = \left( X^T X \right)^{-1} X^T \bar{y}$$

Dr. Yanjun Qi / UVA CS

Assume  
that  $X^T X$  is  
invertible

# Comments on the normal equation

What if  $X$  has less than full column rank?

→ Not Invertible

$$\text{rank}(X_{n \times p}) = \min(n, p)$$

When  $p > n$

$$\text{rank}(X) < p$$

$$\cancel{(X^T X)} < r$$

$$\text{rank} \left( \underbrace{\begin{matrix} X^T & X \\ p \times n & n \times p \end{matrix}}_{p \times p} \right) \leq \min(r(X^T), r(X)) < p$$

For any matrix  $A \in \mathbb{R}^{m \times n}$ , it turns out that the column rank of  $A$  is equal to the row rank of  $A$  (though we will not prove this), and so both quantities are referred to collectively as the **rank** of  $A$ , denoted as  $\text{rank}(A)$ . The following are some basic properties of the rank:

- For  $A \in \mathbb{R}^{m \times n}$ ,  $\text{rank}(A) \leq \min(m, n)$ . If  $\text{rank}(A) = \min(m, n)$ , then  $A$  is said to be **full rank**.
- For  $A \in \mathbb{R}^{m \times n}$ ,  $\text{rank}(A) = \text{rank}(A^T)$ .
- For  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{n \times p}$ ,  $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$ .
- For  $A, B \in \mathbb{R}^{m \times n}$ ,  $\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$ .

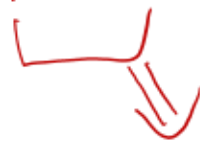
Page 11 Of  
Handout L2

$$\underbrace{X^T X}_{p \times p}$$

$$\text{rank}(X^T X) \leq \text{rank}(X) \leq \min(n, p)$$

When  $n < p$

$$\text{rank}(X^T X) < p$$



singular / not invertible

# Roadmap: Linear Regression with Regularizations

- ✓ When  $p > n$ : How is Ordinary Least squares?
- ✓ Ridge regression: squared loss with L2 regularization
- ✓ Lasso regression: squared loss with L1 regularization
- ✓ Elastic regression: squared loss with L1 AND L2 regularization
- ✓ How to Choose Regularization Parameter

# Review: Vector norms

A norm of a vector  $\|x\|$  is informally a measure of the “length” of the vector.

$$\|x\|_q = \left( \sum_{i=1}^n |x_i|^q \right)^{1/q} \quad q = 1, 2, \dots$$

- Common norms:  $L_1$ ,  $L_2$  (Euclidean)

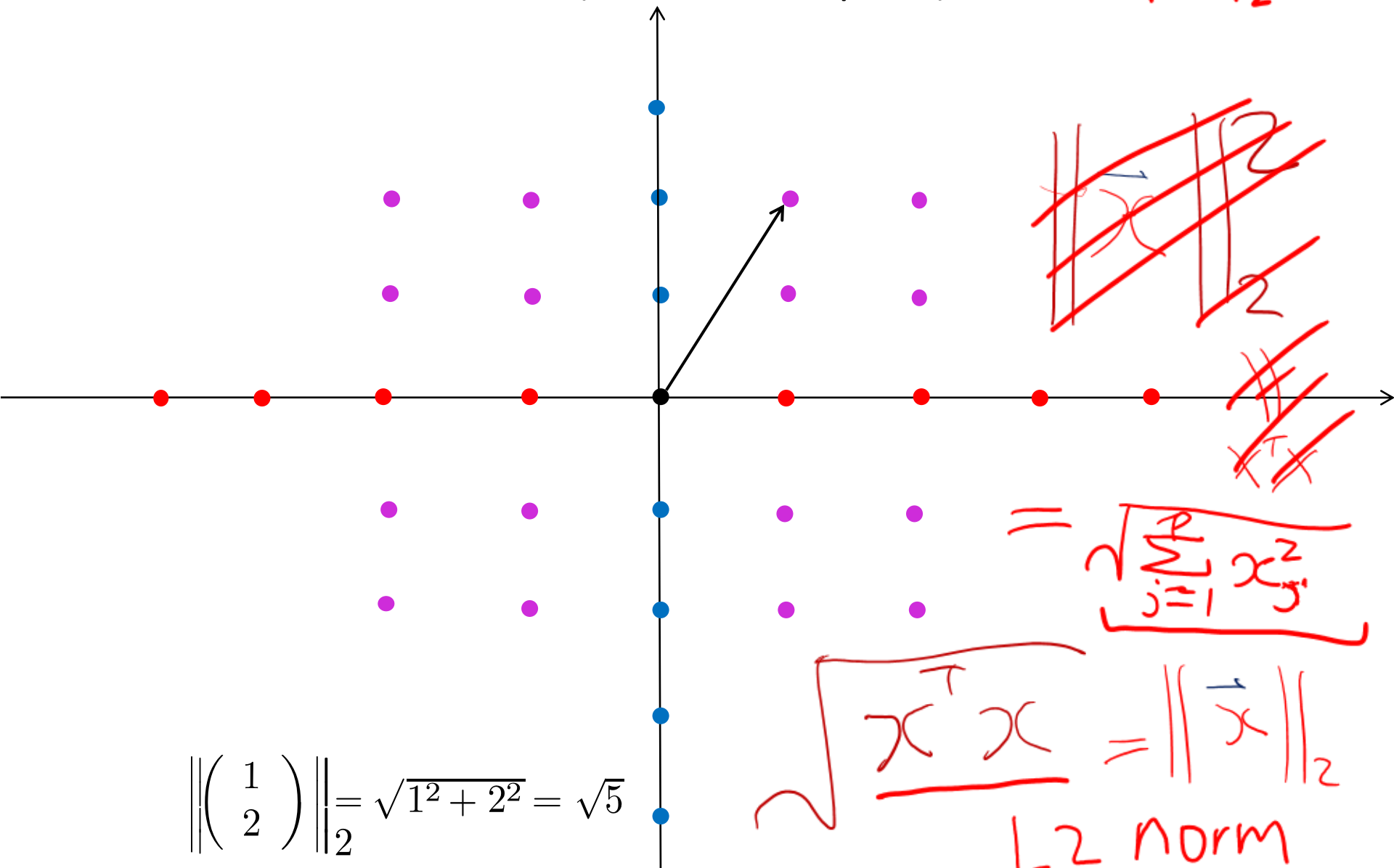
$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad \|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$

- $L_{\text{infinity}}$

$$\|x\|_{\infty} = \max_i |x_i|$$

# Review: Vector Norm (L2, when p=2)

$$\vec{x}^T \vec{x} = \|\vec{x}\|_2^2$$



$$\left\| \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right\|_2 = \sqrt{1^2 + 2^2} = \sqrt{5}$$

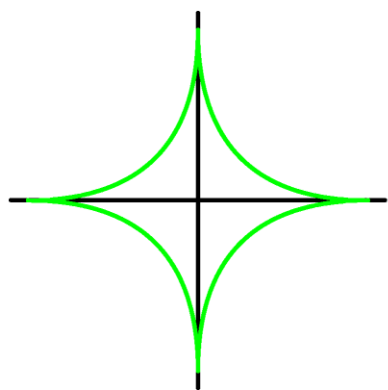
$$\sqrt{\vec{x}^T \vec{x}} = \|\vec{x}\|_2$$

L2 norm

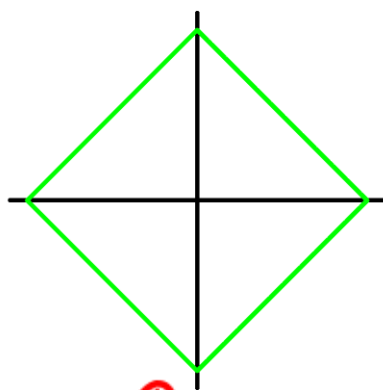
$$= \sqrt{\sum_{j=1}^p x_j^2}$$

# $p$ Norms

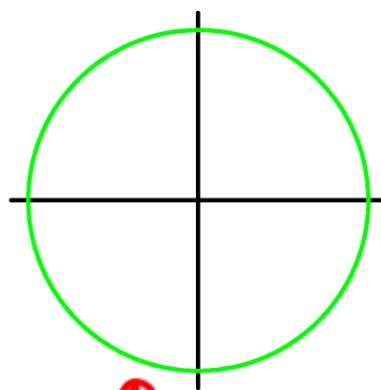
$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$$



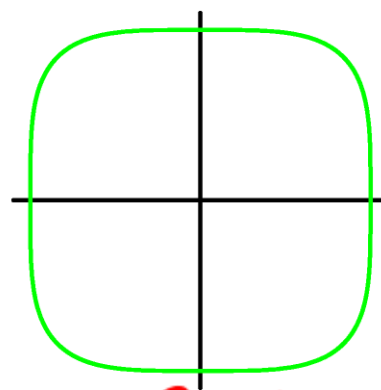
$p=0.5$



$p=1$   
diamond  
contour



$p=2$   
circle  
contour



$p=4$

# Ridge Regression / L2 Regularization

$$\hat{\beta}_{OLS} = \beta^* = (X^T X)^{-1} X^T \bar{y}$$


- If not **invertible**, a classical solution is to add a small positive element to diagonal

$$\lambda > 0$$

$$\beta^* = (X^T X + \lambda I)^{-1} X^T \bar{y}$$

# Extra: Positive Definite Matrix

- A symmetric matrix  $A \in \mathbb{S}^n$  is **positive definite** (PD) if for all non-zero vectors  $x \in \mathbb{R}^n$ ,  $x^T A x > 0$ . This is usually denoted  $A \succ 0$  (or just  $A > 0$ ), and often times the set of all positive definite matrices is denoted  $\mathbb{S}_{++}^n$ .
- A symmetric matrix  $A \in \mathbb{S}^n$  is **positive semidefinite** (PSD) if for all vectors  $x^T A x \geq 0$ . This is written  $A \succeq 0$  (or just  $A \geq 0$ ), and the set of all positive semidefinite matrices is often denoted  $\mathbb{S}_+^n$ .

One important property of positive definite matrices is that

- ➔ They are always full rank, and hence, invertible.
- ➔ Extra: See Proof at Page 17-18 of Linear-Algebra Handout

positive definite (PD)

$$\forall a \neq 0 \quad \underbrace{a^T (X^T \Sigma + \lambda I) a} > 0$$

$$= a^T X^T \Sigma a + \lambda a^T a$$

$$= \|\Sigma a\|_2^2 + \lambda \|a\|_2^2 > 0$$

$$\beta^* = \underbrace{(X^T X + \lambda I)^{-1}} \quad X^T \bar{y}$$

Extra: Positive Definite Matrix

$$\forall \vec{a} \neq 0, \quad \vec{a}^T A \vec{a} \geq 0 \Rightarrow A \succeq 0$$

$$\textcircled{1} \quad \begin{matrix} 1 \times p & p \times n & n \times p & p \times 1 \end{matrix} \quad \vec{a}^T X^T X \vec{a} = \underbrace{(X \vec{a})^T (X \vec{a})}_{\substack{n \times p \times p \times 1 \\ n \times 1}} = \|\bar{X} \vec{a}\|_2^2 \geq 0$$

[for any non-zero vector  $\vec{a} \in \mathbb{R}^p$ ]

$X^T X$  PSD

$$\textcircled{2} \quad \vec{a}^T \underbrace{(X^T X + \lambda I)}_{\text{PD} \rightarrow \text{invertible}} \vec{a} = \vec{a}^T X^T X \vec{a} + \lambda \vec{a}^T I \vec{a} = \|\bar{X} \vec{a}\|_2^2 + \lambda \|\vec{a}\|_2^2 > 0$$

$\lambda > 0, \vec{a} \neq 0$

# Ridge Regression / Squared Loss+L2

$$\beta^* = \left( X^T X + \lambda I \right)^{-1} X^T \bar{y}$$

- As the solution from


$$\hat{\beta}^{ridge} = \operatorname{argmin}_{\beta} (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$$

to minimize, take derivative and set to zero

$$\sum_{n \times p} \beta \rightarrow \hat{y}_{n \times 1}$$

# Ridge Regression / Squared Loss+L2

$$\beta^* = (X^T X + \lambda I)^{-1} X^T \bar{y}$$

- As the solution from

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \underbrace{\sum_{j=1}^n (y_j - \beta^T x_j)^2}_{\text{Squared Loss}} + \lambda \beta^T \beta$$

to minimize, take derivative and set to zero

By convention, the bias/intercept term is typically not regularized.  
Here we assume data has been centered ... therefore no bias term

$$\sum_{n \times p} \beta \rightarrow \hat{y}_{n \times 1}$$

# Ridge Regression / Squared Loss+L2

$$\beta^* = (X^T X + \lambda I)^{-1} X^T \bar{y}$$

- As the solution from

$$\sum_{j=1}^n (y_j - \beta^T x_j)^2$$



$$\hat{\beta}^{ridge} = \operatorname{argmin} (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$$

to minimize, take derivative and set to zero

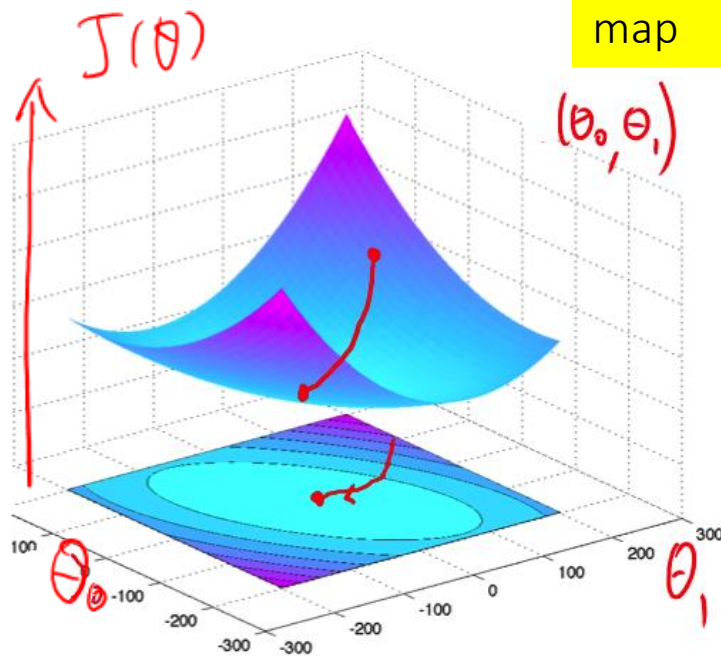
- Equivalently  $\hat{\beta}^{ridge} = \operatorname{argmin} (y - X\beta)^T (y - X\beta)$   
subject to  $\sum_{j=\{1..p\}} \beta_j^2 \leq s^2$

circle  
with radial  
s

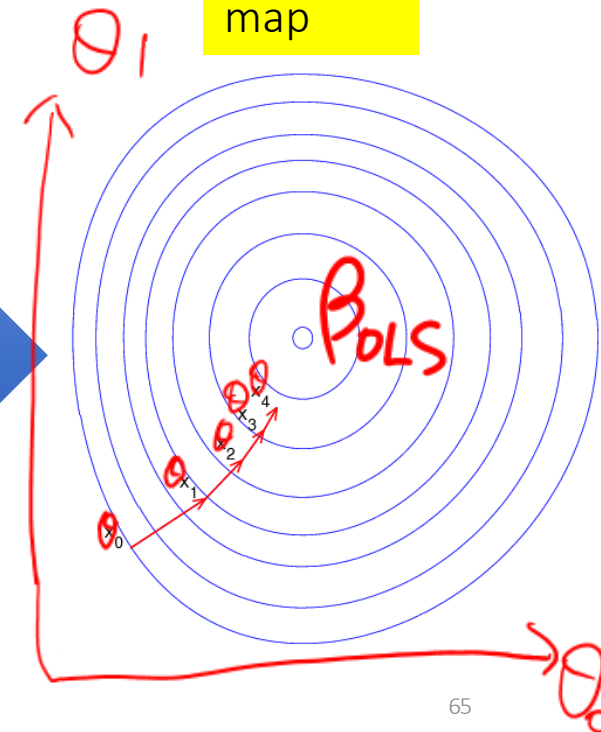
# Review



Surface  
map

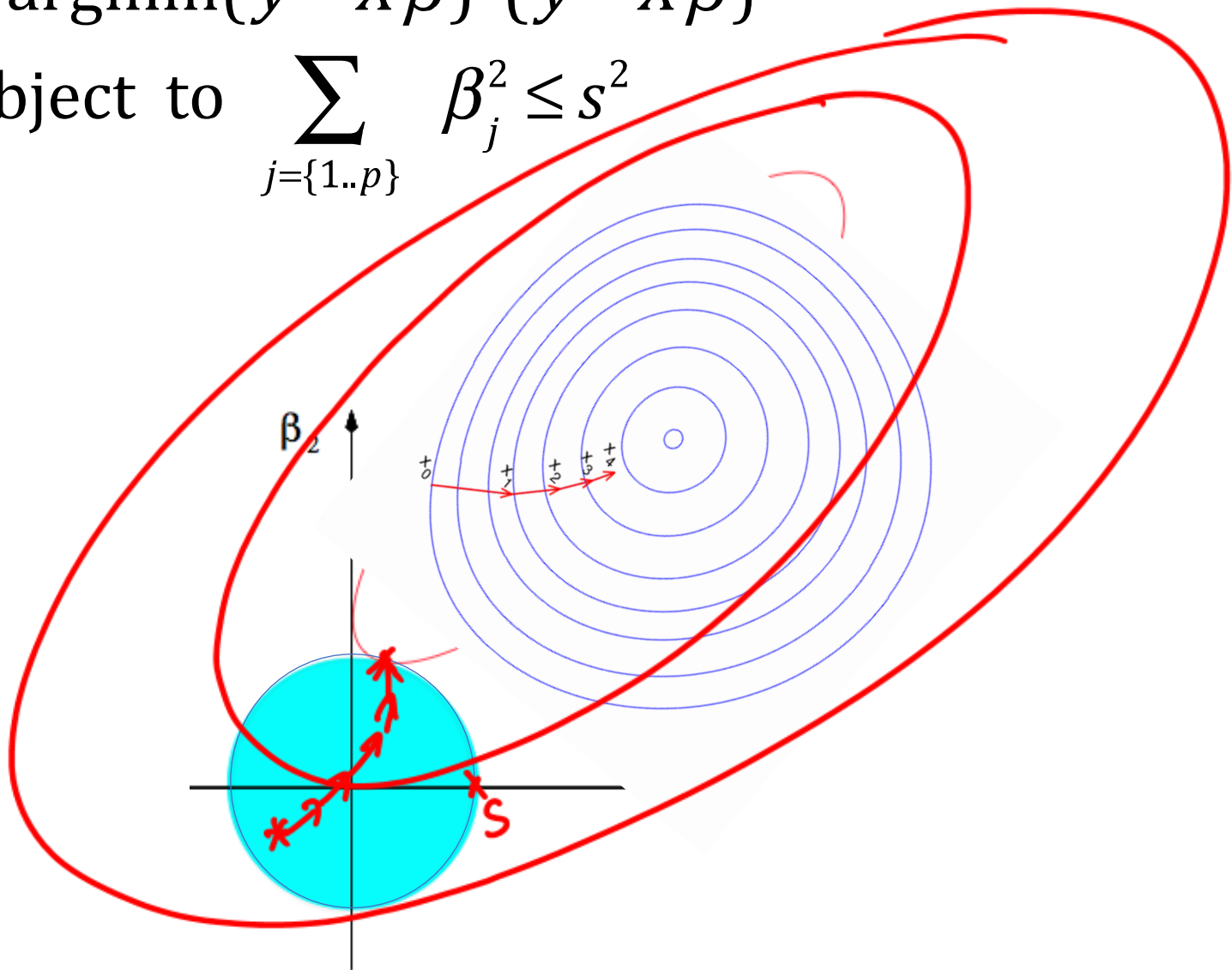


Contour  
map



$$\hat{\beta}^{ridge} = \operatorname{argmin} (y - X\beta)^T (y - X\beta)$$

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

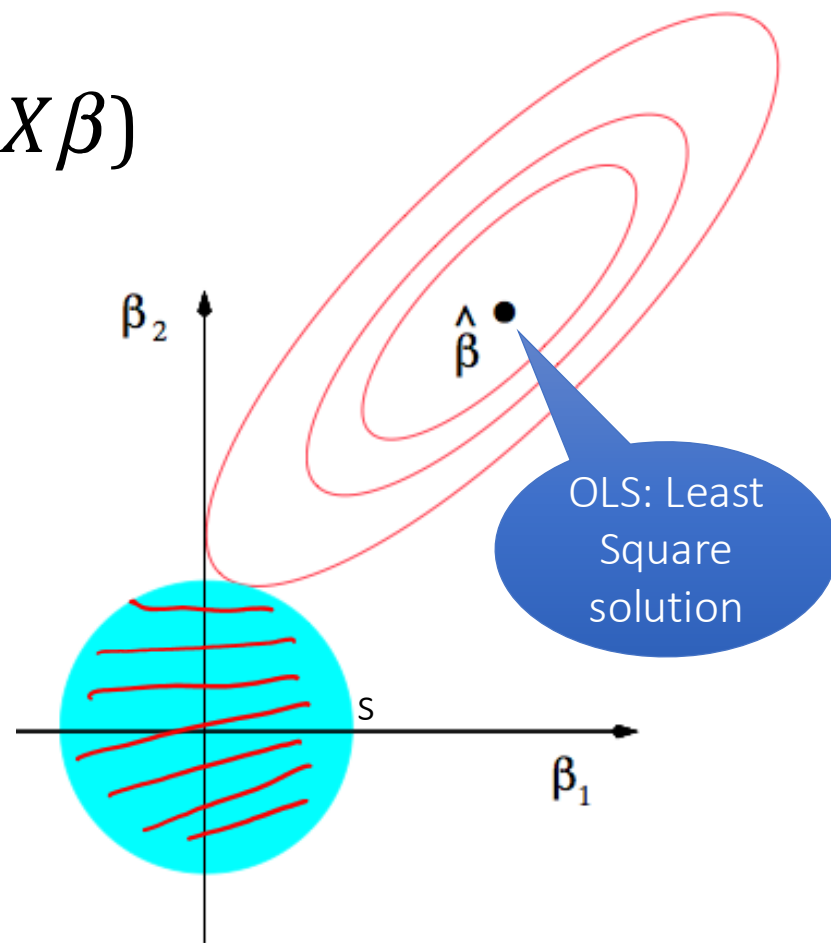


# Objective Function's Contour lines from Ridge Regression

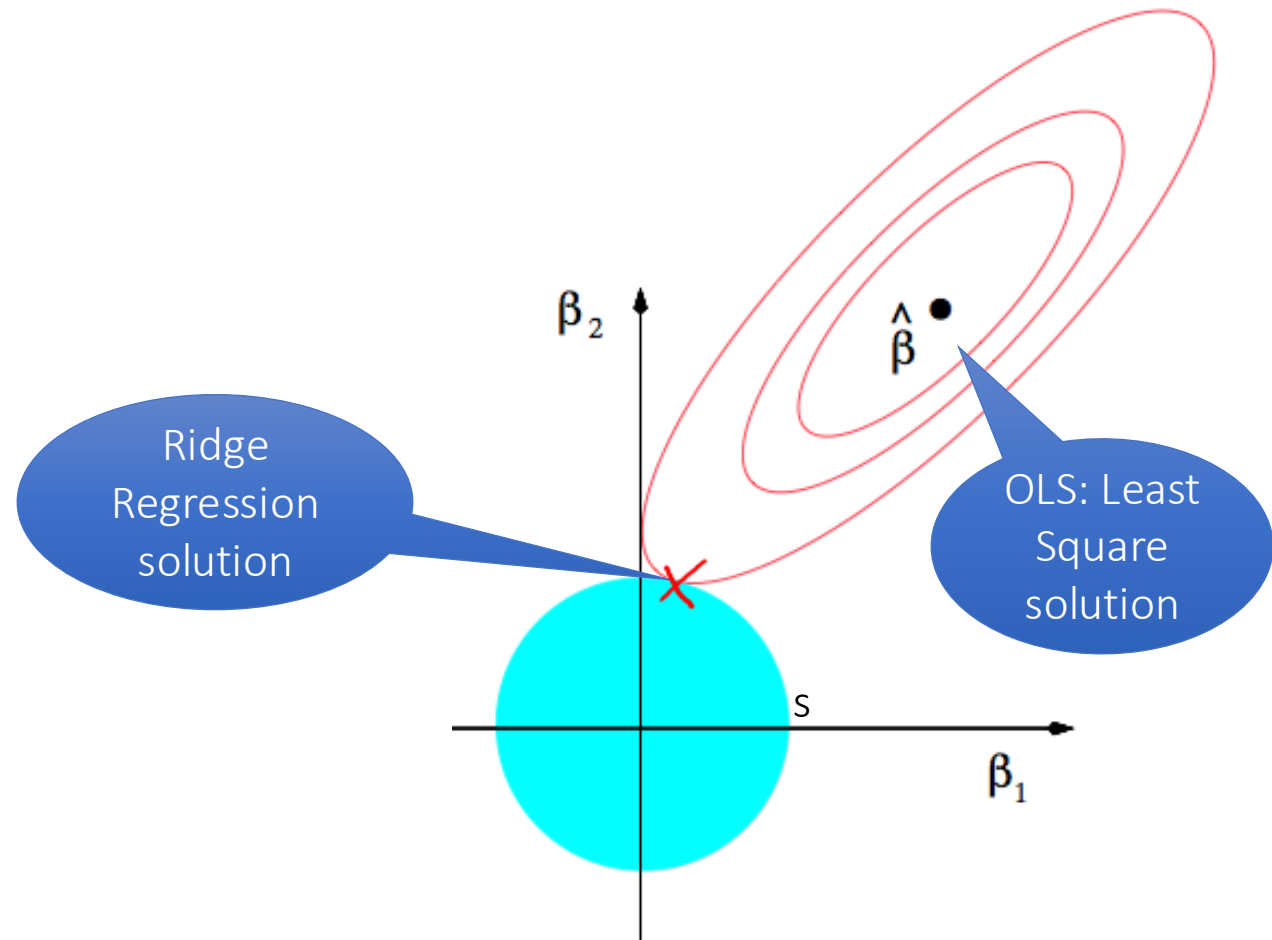
$$\hat{\beta}^{\text{ridge}} = \operatorname{argmin} (y - X\beta)^T (y - X\beta)$$

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

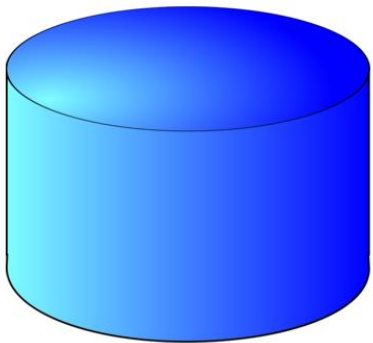
circle  
with radial  
s



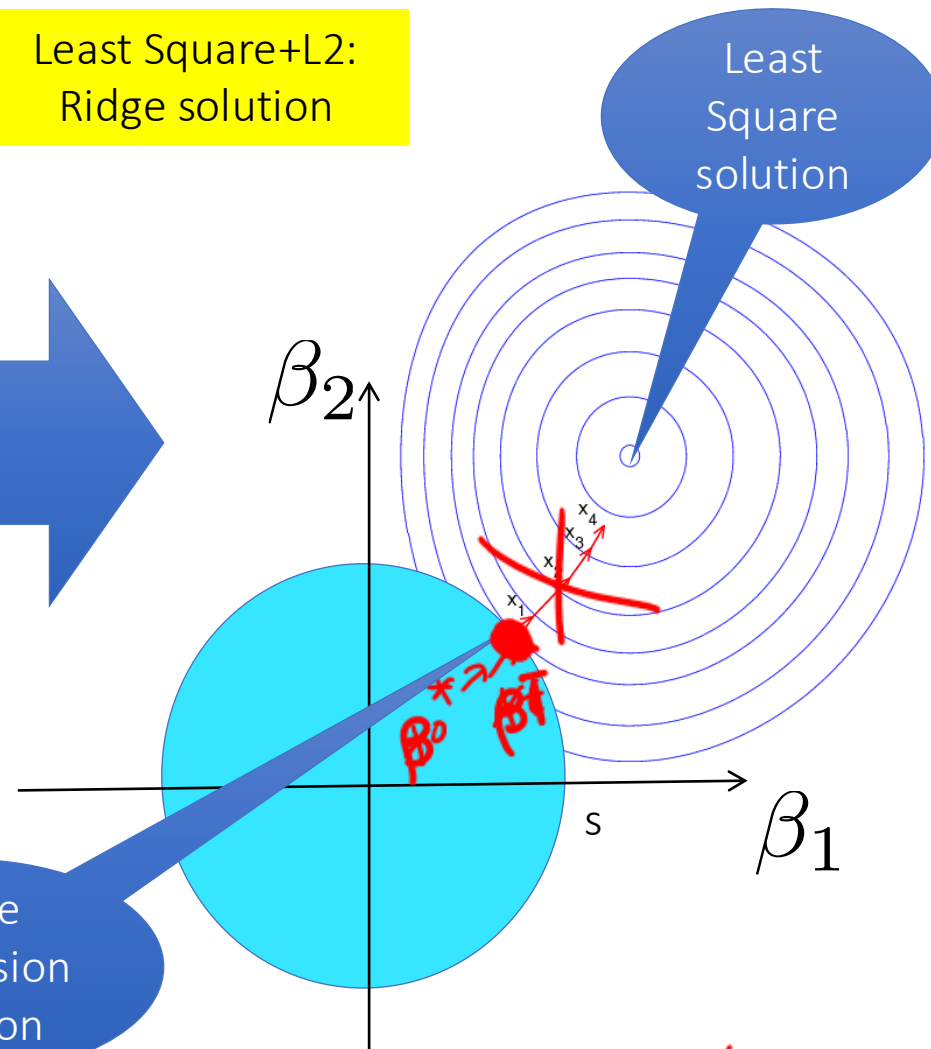
# Objective Function's Contour lines from Ridge Regression



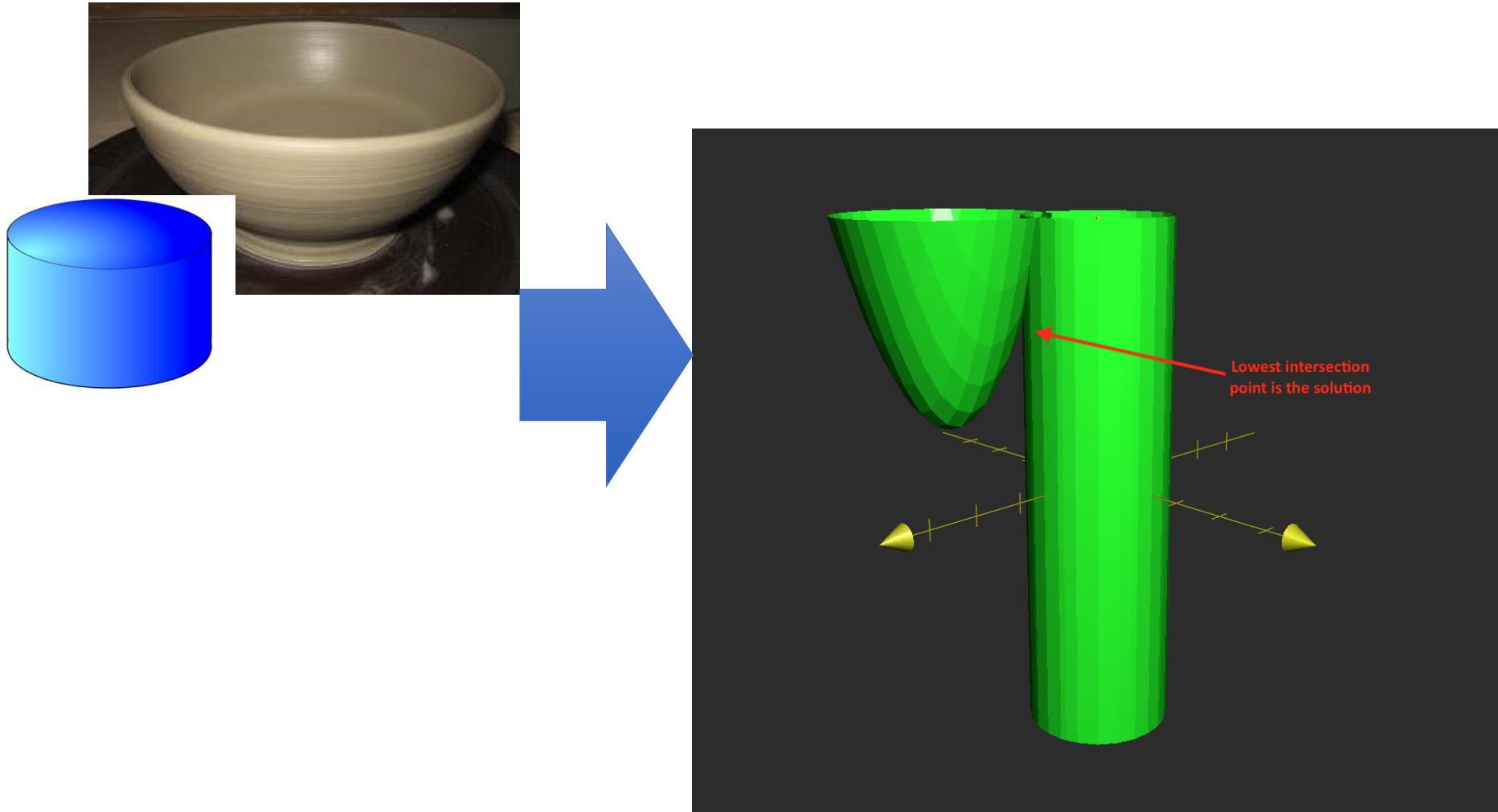
Least Square+L2:  
Ridge solution



Ridge  
Regression  
solution



must within the circle



# Parameter Shrinkage

$$\beta_{OLS} = (X^T X)^{-1} X^T \bar{y}$$

$$\beta_{Rg} = (X^T X + \lambda I)^{-1} X^T \bar{y}$$

when  $X^T X = I$   
 $\Rightarrow$

$$\beta_{OLS} = X^T \bar{y}$$

$\lambda > 0$

$\lambda > 0$

when  $X^T X = I$   
 $\Rightarrow$

$$\beta_{Rg} = \frac{1}{1+\lambda} X^T \bar{y} = \frac{1}{1+\lambda} \beta_{OLS}$$

When  $X^T X = I \Rightarrow \beta_{Rg} = \frac{1}{1+\lambda} \beta_{OLS}$  [Shrinkage]

When  $X^T X$  general case, see advanced analysis @

Page 65 of ESL book @

[http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII\\_print10.pdf](http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf)

## Extra: two forms of Ridge Regression

- Totally equivalent

$$\begin{cases} \textcircled{1} \arg\min_{\beta} J(\beta) + \lambda \beta^T \beta \\ \textcircled{2} \arg\min_{\beta} J(\beta), \text{ s.t. } \beta^T \beta \leq S^2 \end{cases}$$

Optimal solution  $\beta_{Rg}^*$  needs (necessary condition)

$$\left[ \lambda \left( \sum_j (\beta_{Rg})_j^2 - S^2 \right) = 0 \right] \Rightarrow S^2 = \sum_j (\beta_{Rg})_j^2 \quad \lambda > 0$$

When  $X^T X = I$ ,

$$S^2 = \sum_j (\beta_{Rg})_j^2 = \frac{1}{(1+\lambda)^2} \sum_j (\beta_{ols})_j^2 \Rightarrow S^2 \propto \frac{1}{(1+\lambda)^2}$$

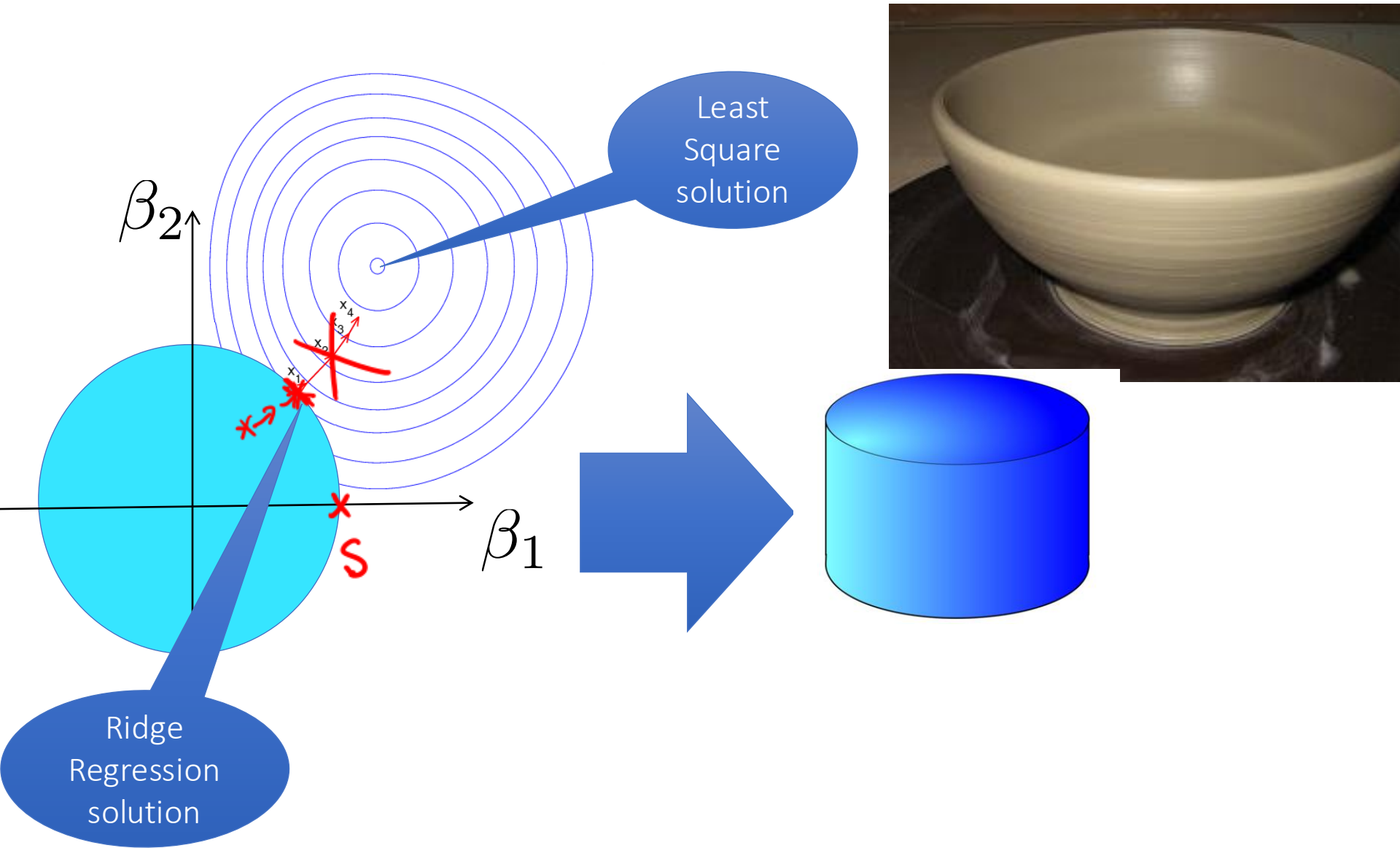
$$\lambda = \sqrt{\frac{\sum_j (\beta_{ols})_j^2}{S^2}} - 1$$

<http://stats.stackexchange.com/questions/190993/how-to-find-regression-coefficients-beta-in-ridge-regression>

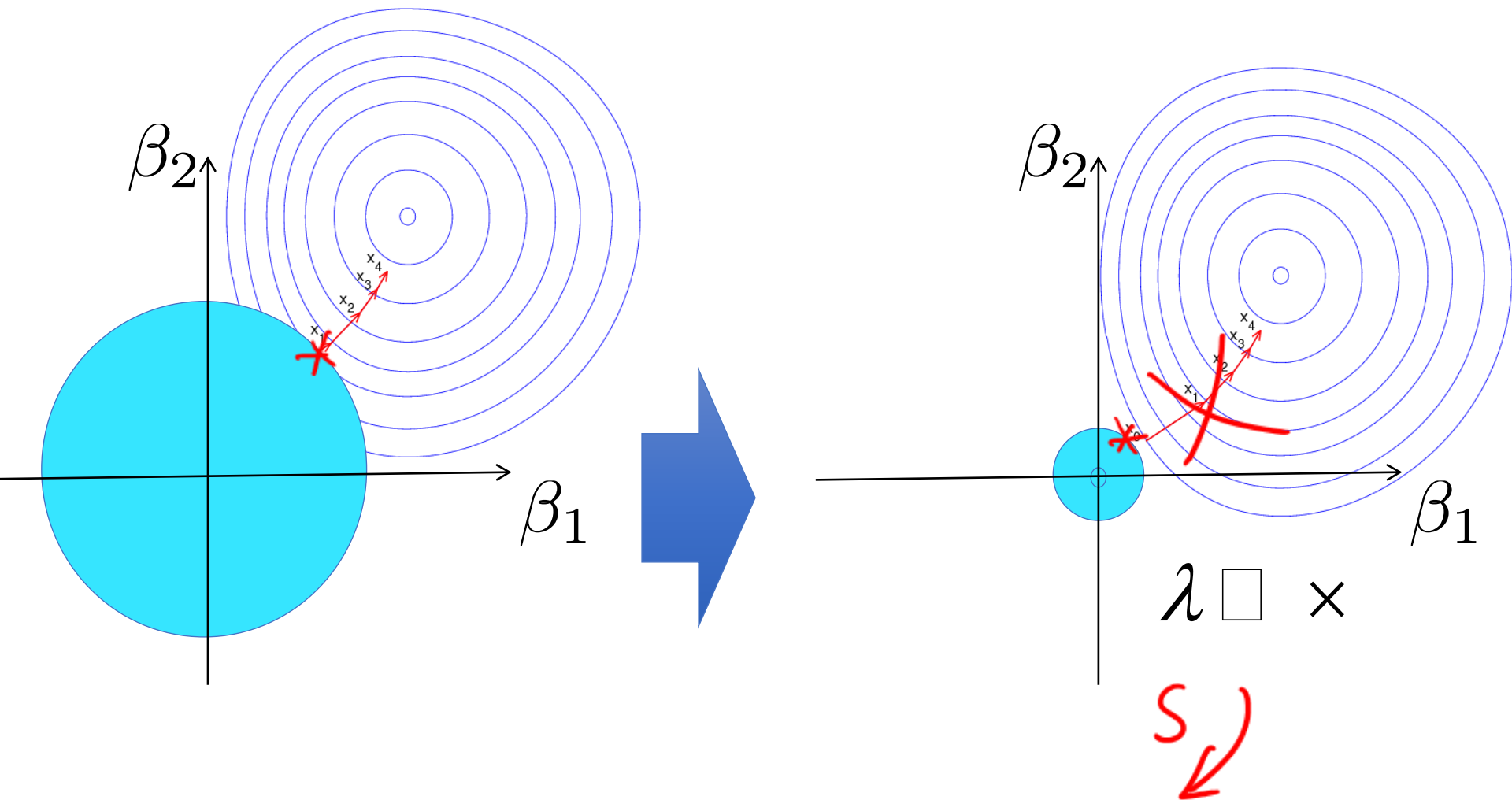
# Ridge Regression: Squared Loss+L2

- $\lambda > 0$  penalizes each  $\beta_j$
- if  $\lambda = 0$  we get the least squares estimator;
- if  $\lambda \rightarrow \infty$ , then  $\beta_j$  to zero

# ✓ Influence of Regularization Parameter



# ✓ Influence of Regularization Parameter

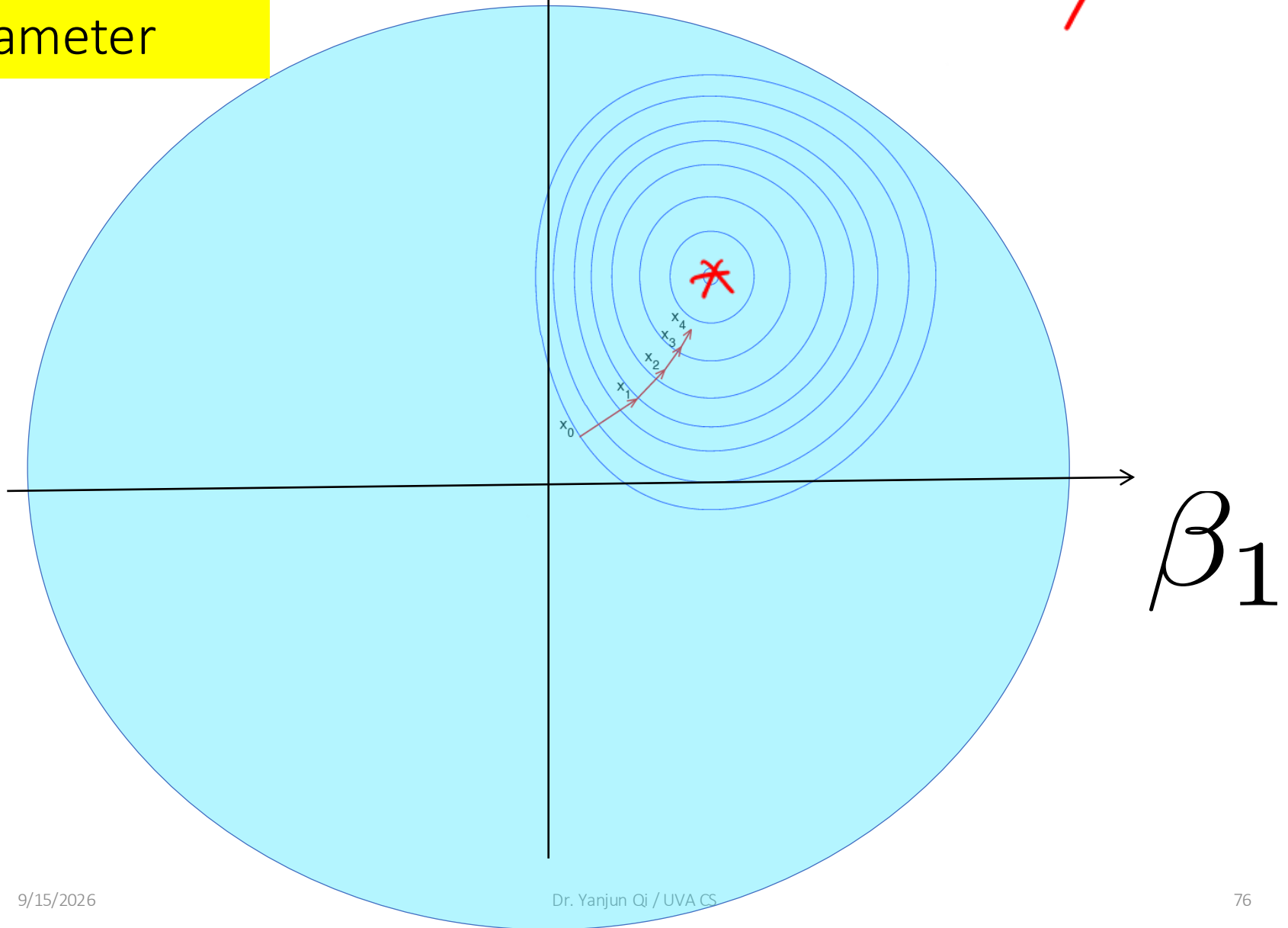


✓ Influence of  
Regularization  
Parameter

$\beta_2$

$\lambda \rightarrow 0$

$s$  ↗



# Roadmap: Linear Regression with Regularizations

- ✓ When  $p > n$ : How is Ordinary Least squares?
- ✓ Ridge regression: squared loss with L2 regularization
- ✓ Lasso regression: squared loss with L1 regularization
- ✓ Elastic regression: squared loss with L1 AND L2 regularization
- ✓ How to Choose Regularization Parameter

## (2) Lasso (least absolute shrinkage and selection operator) / Squared Loss+L1

- The lasso is a shrinkage method like ridge, but acts in a nonlinear manner on the outcome  $y$ .
- The lasso is defined by

$$\hat{\beta}^{lasso} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n (y_i - x_i^T \beta)^2$$



$$\text{subject to } \sum_j |\beta_j| \leq s$$



By convention, the bias/intercept term is typically not regularized.  
Here we assume data has been centered ... therefore no bias term

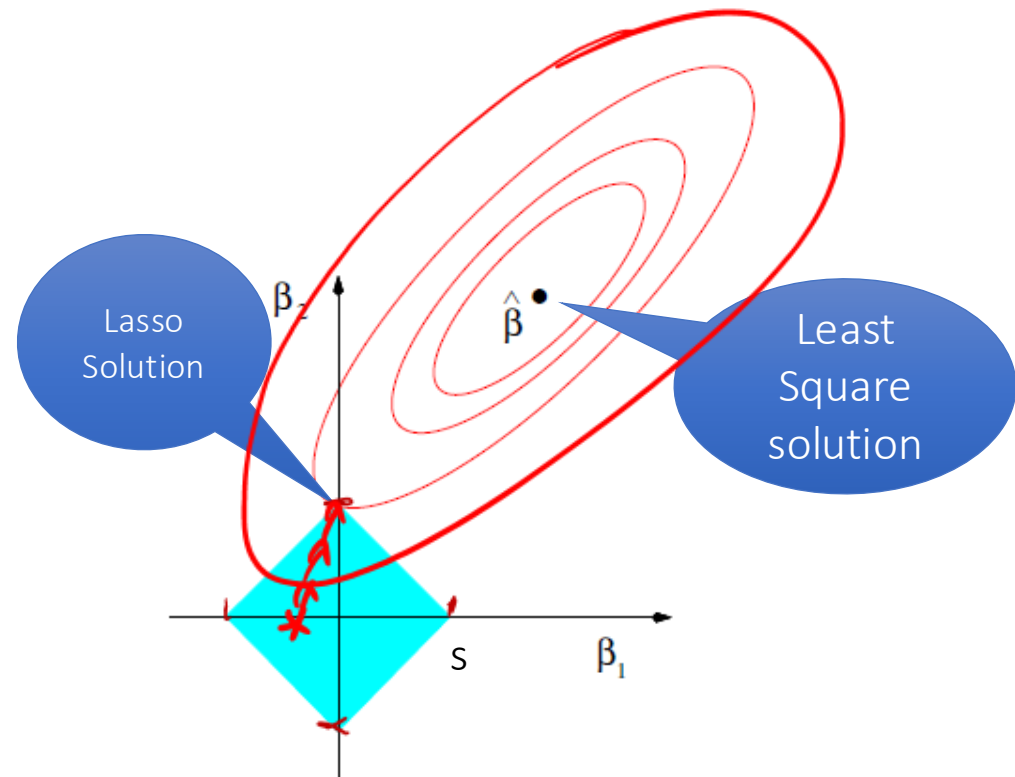
# Lasso (least absolute shrinkage and selection operator)

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

push  $\beta_j = 0$

$$\beta^{\text{lasso}} = [0, s, 0]^T$$

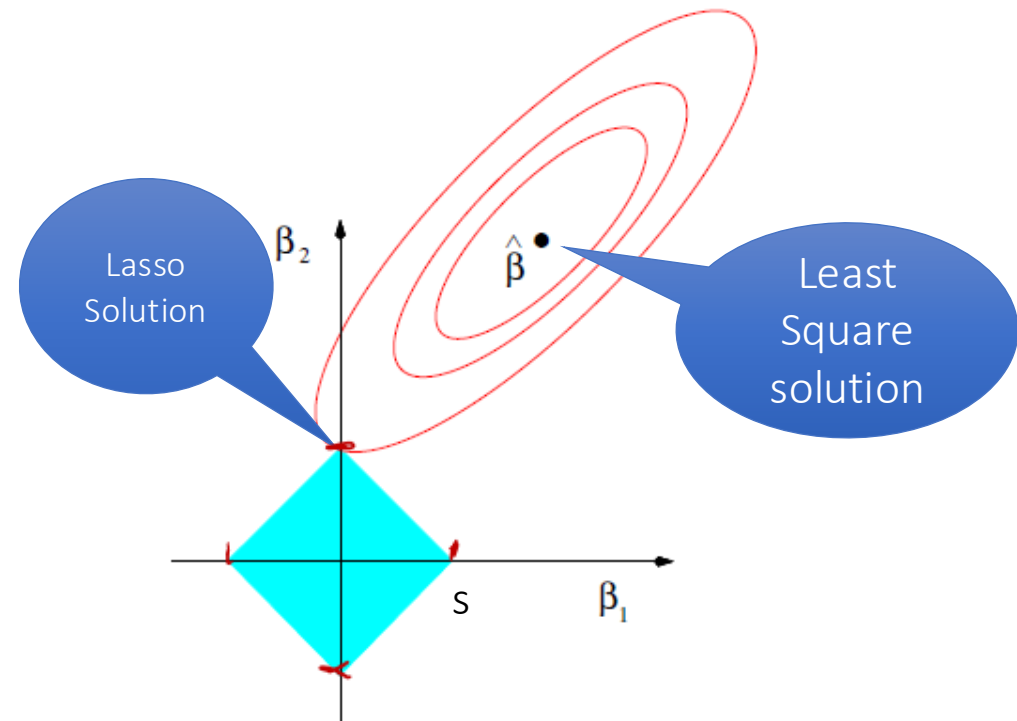
- Suppose in 2 dimension
- $\beta = (\beta_1, \beta_2)$
- $|\beta_1| + |\beta_2| = \text{const}$
- $|\beta_1| + |-\beta_2| = \text{const}$
- $|-\beta_1| + |\beta_2| = \text{const}$
- $|-\beta_1| + |-\beta_2| = \text{const}$



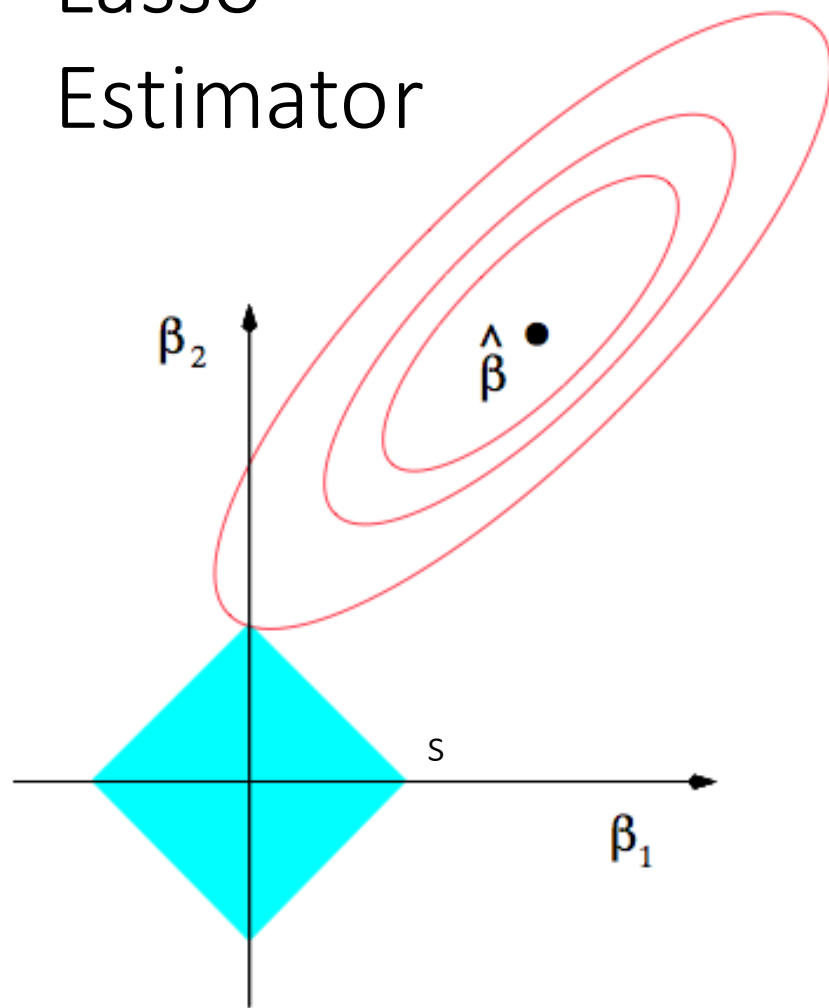
$$\hat{y} = \sum_{j=1}^p \beta_j x_j$$

when many  $\beta_j$  are zero  $\Rightarrow$  select feature

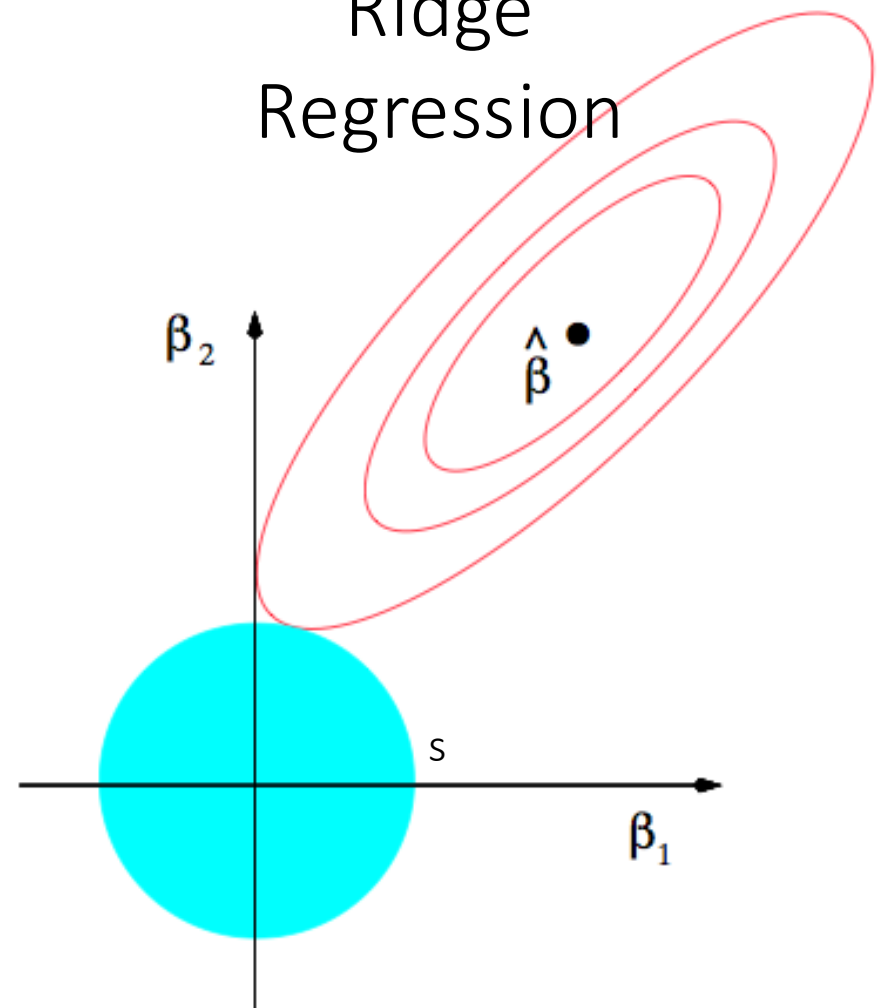
- In the Figure, the solution has eliminated the role of  $x_2$ , leading to sparsity



# Lasso Estimator



# Ridge Regression



**FIGURE 3.11.** Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions  $|\beta_1| + |\beta_2| \leq t$  and  $\beta_1^2 + \beta_2^2 \leq t^2$ , respectively, while the red ellipses are the contours of the least squares error function.

# Lasso (least absolute shrinkage and selection operator)

- Notice that ridge penalty is replaced

by

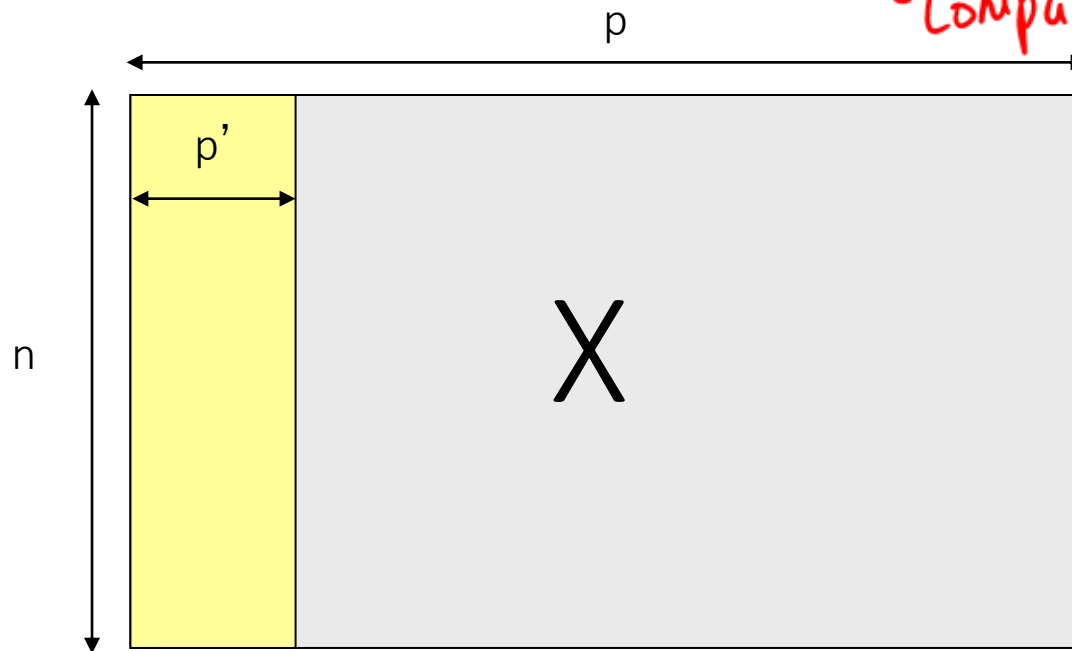
$$\sum |\beta_j|$$

$$\sum \beta_j^2$$

- Due to the nature of the constraint, if tuning parameter is chosen small enough, then the lasso will set some coefficients exactly to zero.

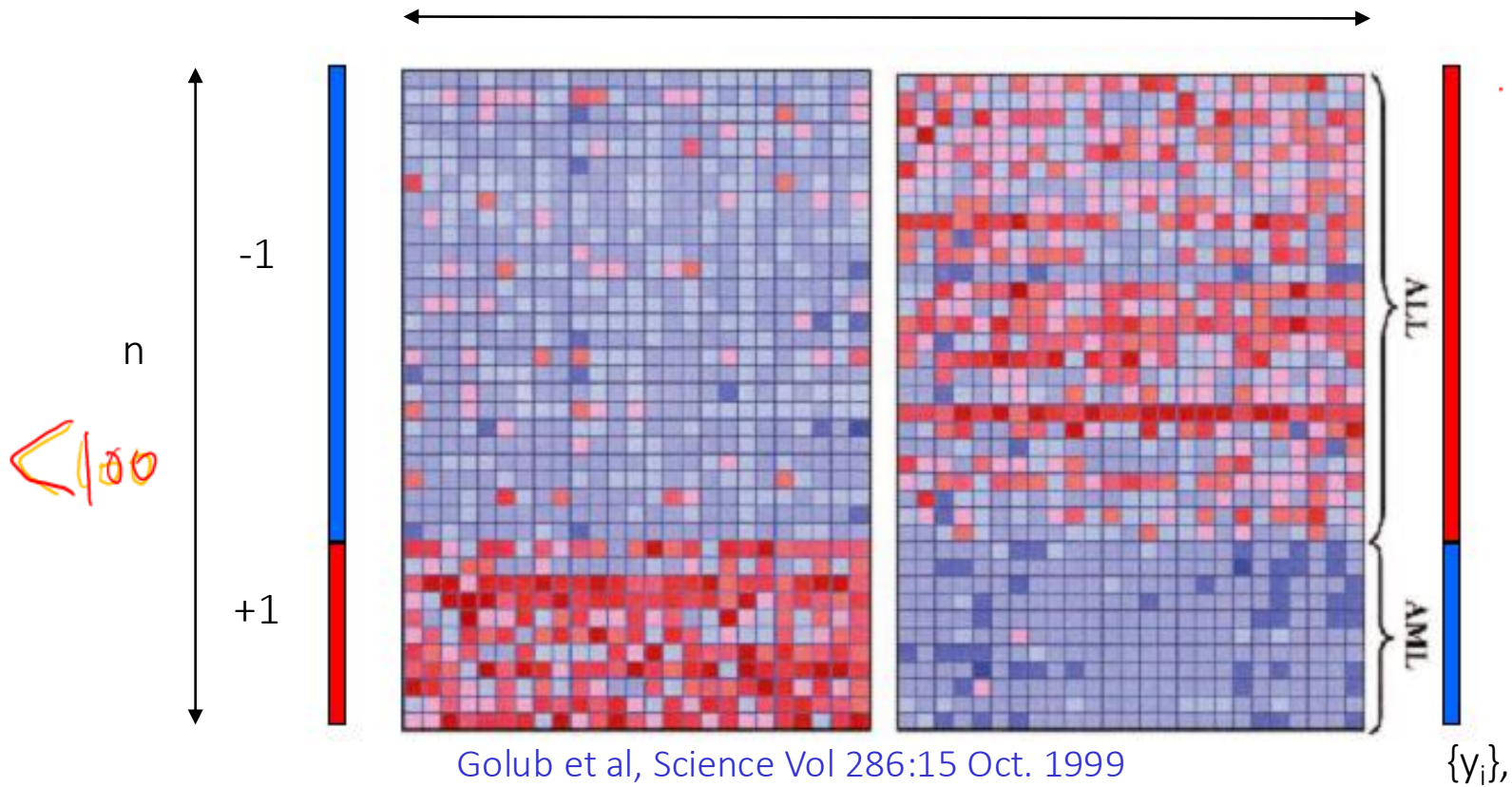
# Lasso: Implicit Feature Selection

$p \rightarrow p' \Rightarrow$  { easy to understand  
Computational  
efficient



# e.g., Leukemia Diagnosis

$p' > 30,000$



$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X} \mathbf{y}$$

when  $n < p$ ,  $O(p^3)$

$$\Rightarrow \begin{matrix} \mathbf{X}^T & \mathbf{X} \\ p \times n & n \times p \end{matrix} : O(np^2)$$

$$\Rightarrow \underbrace{(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1}}_{p \times p} : O(p^3)$$

$$\Rightarrow \mathbf{X} \mathbf{y} : O(np)$$

Computationally,

choose to  
make  $p \downarrow$   
if we can



operational mode  $\mathbf{X} \mathbf{\beta}^*$   
 $n \times p$   $p \times 1$   
 $O(n' p)$

# Roadmap: Linear Regression with Regularizations

- ✓ When  $p > n$ : How is Ordinary Least squares?
- ✓ Ridge regression: squared loss with L2 regularization
- ✓ Lasso regression: squared loss with L1 regularization
- ✓ Elastic regression: squared loss with L1 AND L2 regularization
- ✓ How to Choose Regularization Parameter

# Lasso for when $p > n$

- Prediction **accuracy and model interpretation** are two important aspects of regression models.
- LASSO does **shrinkage and variable selection** simultaneously for better prediction and model interpretation.

Disadvantage:

- In  $p > n$  case, lasso selects at most  $n$  variable before it saturates
- If there is a group of variables among which the pairwise correlations are very high, then lasso select one from the group

## => Hybrid of Ridge and Lasso : Elastic Net regularization

- L1 part of the penalty generates a sparse model
- L2 part of the penalty (extra):
  - Remove the limitation of the number of selected variables
  - Encouraging group effect
  - Stabilize the L1 regularization path

# Naïve elastic net

- For any non negative fixed  $\lambda_1$  and  $\lambda_2$ , naive elastic net criterion:

$$L(\lambda_1, \lambda_2, \beta) = |\mathbf{y} - \mathbf{X}\beta|^2 + \lambda_2 |\beta|^2 + \lambda_1 |\beta|_1,$$

$$|\beta|^2 = \sum_{j=1}^p \beta_j^2, \quad |\beta|_1 = \sum_{j=1}^p |\beta_j|.$$

- The naive elastic net estimator is the minimizer of above equation

$$\hat{\beta} = \arg \min_{\beta} \{L(\lambda_1, \lambda_2, \beta)\}.$$

# Naïve elastic net

- For any non negative fixed  $\lambda_1$  and  $\lambda_2$ , naive elastic net criterion:

$$L(\lambda_1, \lambda_2, \beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2 \|\beta\|^2 + \lambda_1 \|\beta\|_1,$$

$$\|\beta\|^2 = \sum_{j=1}^p \beta_j^2, \quad \|\beta\|_1 = \sum_{j=1}^p |\beta_j|.$$

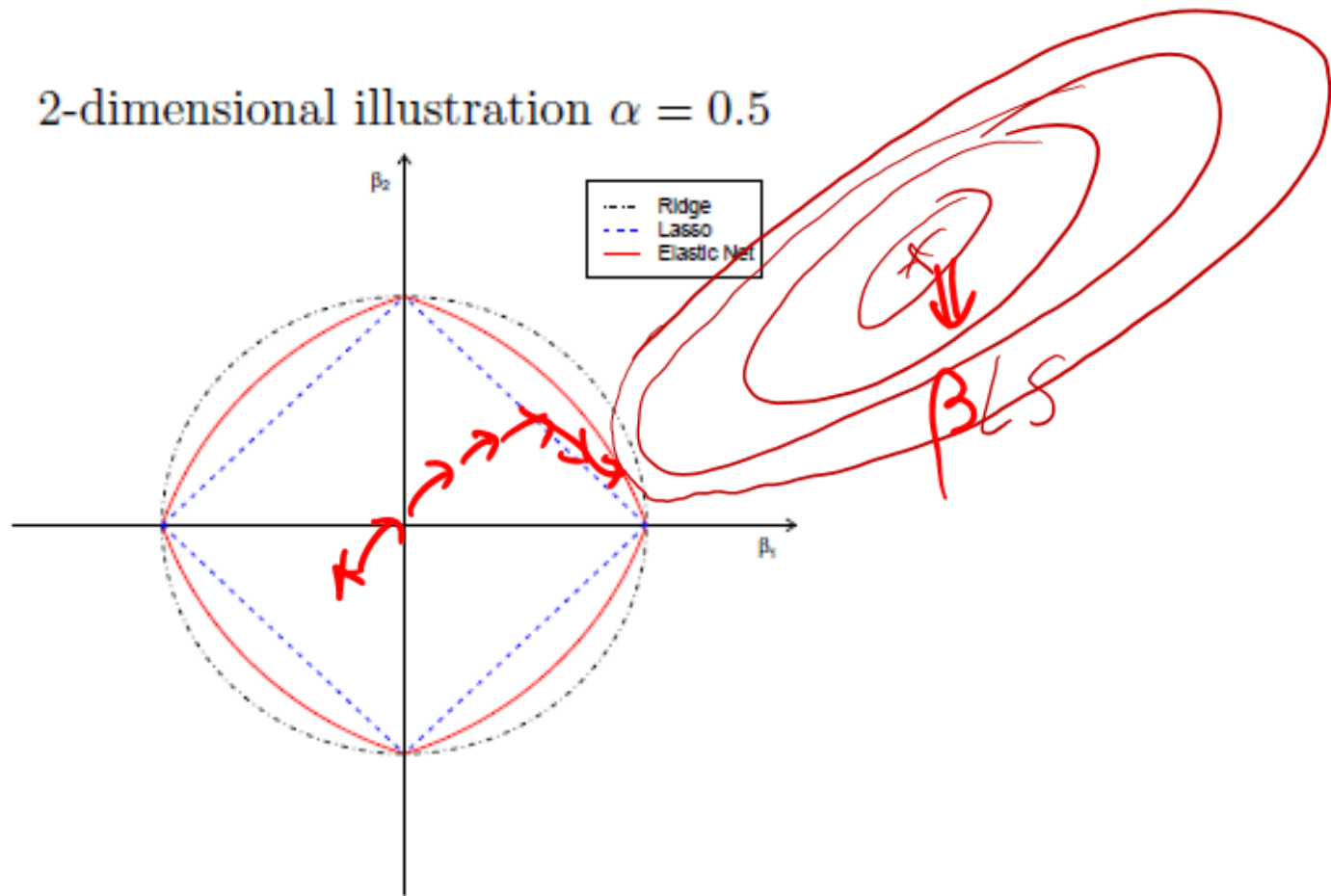
- The naive elastic net estimator is the minimizer of above

$$\hat{\beta} = \arg \min_{\beta} \{L(\lambda_1, \lambda_2, \beta)\}.$$

- Equivalently:  $\alpha = \lambda_2 / (\lambda_1 + \lambda_2)$

$$\hat{\beta} = \arg \min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2, \quad \text{subject to } (1 - \alpha) \|\beta\|_1 + \alpha \|\beta\|^2 \leq t \text{ for some } t.$$

# Geometry of elastic net



e.g. A Practical Application of Regression Model

# Movie Reviews and Revenues: An Experiment in Text Regression\*

**Mahesh Joshi   Dipanjan Das   Kevin Gimpel   Noah A. Smith**

Language Technologies Institute

Carnegie Mellon University

Pittsburgh, PA 15213, USA

`{maheshj, dipanjan, kgimpel, nasmith}@cs.cmu.edu`

## Abstract

We consider the problem of predicting a movie's opening weekend revenue. Previous work on this problem has used metadata about a movie—e.g., its genre, MPAA rating, and cast—with very limited work making use of text *about* the movie. In this paper, we use the text of film critics' reviews from several sources to predict opening weekend revenue. We describe a new dataset pairing movie reviews with metadata and revenue data, and show that review text can substitute for metadata, and even improve over it, for prediction.

Proceedings of  
HLT '2010  
Human  
Language  
Technologies:

### III. Model

- ❖ Linear regression with the elastic net (Zou and Hastie, 2005)

$$\hat{\theta} = \underset{\theta=(\beta_0, \beta)}{\operatorname{argmin}} \frac{1}{2n} \left[ \sum_{i=1}^n \left( y_i - (\beta_0 + \mathbf{x}_i^\top \beta) \right)^2 \right] + \lambda P(\beta)$$

$$P(\beta) = \sum_{j=1}^p \left( \frac{1}{2} (1 - \alpha) \beta_j^2 + \alpha |\beta_j| \right)$$

Use linear regression to directly predict the opening weekend gross earnings, denoted as  $y$ , based on features  $x$  extracted from the movie metadata and/or the text of the reviews.

|           | Feature                             | Weight (\$M) |
|-----------|-------------------------------------|--------------|
| rating    | pg                                  | +0.085       |
|           | <i>New York Times</i> : adult       | -0.236       |
|           | <i>New York Times</i> : rate_r      | -0.364       |
| sequels   | this_series                         | +13.925      |
|           | <i>LA Times</i> : the_franchise     | +5.112       |
|           | <i>Variety</i> : the_sequel         | +4.224       |
| people    | <i>Boston Globe</i> : will_smith    | +2.560       |
|           | <i>Variety</i> : brittany           | +1.128       |
|           | ^_producer_brian                    | +0.486       |
| genre     | <i>Variety</i> : testosterone       | +1.945       |
|           | <i>Ent. Weekly</i> : comedy_for     | +1.143       |
|           | <i>Variety</i> : a_horror           | +0.595       |
|           | documentary                         | -0.037       |
|           | independent                         | -0.127       |
| sentiment | <i>Boston Globe</i> : best_parts_of | +1.462       |
|           | <i>Boston Globe</i> : smart_enough  | +1.449       |
|           | <i>LA Times</i> : a_good_thing      | +1.117       |
|           | shame_\$                            | -0.098       |
|           | bogeyman                            | -0.689       |
| plot      | <i>Variety</i> : torso              | +9.054       |
|           | vehicle_in                          | +5.827       |
|           | superhero_\$                        | +2.020       |

An example of how real applications use the elastic net and its weights!

Here, the features are from the text-only model annotated in Table 2.

The feature weights can be directly interpreted as U.S. dollars contributed to the predicted value by each occurrence of the feature.

Sentiment-related text features are not as prominent as might be expected, and their overall proportion in the set of features with non-zero weights is quite small (estimated in preliminary trials at less than 15%). Phrases that refer to metadata are the more highly weighted and frequent ones.

Table 3: Highly weighted features categorized manually. ^ and \$ denote sentence boundaries.

|                  | Features               | Site | Total        |              | Per Screen                |              |
|------------------|------------------------|------|--------------|--------------|---------------------------|--------------|
|                  |                        |      | MAE (\$M)    | $r$          | MAE (\$K)                 | $r$          |
| meta             | Predict mean           |      | 11.672       | —            | 6.862                     | —            |
|                  | Predict median         |      | 10.521       | —            | 6.642                     | —            |
|                  | Best                   |      | 5.983        | 0.722        | 6.540                     | 0.272        |
| text             | I<br><i>see Tab. 3</i> | —    | 8.013        | 0.743        | 6.509                     | 0.222        |
|                  |                        | +    | 7.722        | 0.781        | 6.071                     | 0.466        |
|                  |                        | B    | 7.627        | 0.793        | 6.060                     | 0.411        |
|                  | I $\cup$ II            | —    | 8.060        | 0.743        | 6.542                     | 0.233        |
|                  |                        | +    | <b>7.420</b> | 0.761        | 6.240                     | 0.398        |
|                  |                        | B    | 7.447        | 0.778        | 6.299                     | 0.363        |
|                  | I $\cup$ III           | —    | 8.005        | 0.744        | 6.505                     | 0.223        |
|                  |                        | +    | 7.721        | 0.785        | 6.013                     | <b>0.473</b> |
|                  |                        | B    | 7.595        | <b>0.796</b> | <sup>†</sup> <b>6.010</b> | 0.421        |
| meta $\cup$ text | I                      | —    | 5.921        | <b>0.819</b> | 6.509                     | 0.222        |
|                  |                        | +    | 5.757        | 0.810        | 6.063                     | 0.470        |
|                  |                        | B    | 5.750        | <b>0.819</b> | 6.052                     | 0.414        |
|                  | I $\cup$ II            | —    | 5.952        | 0.818        | 6.542                     | 0.233        |
|                  |                        | +    | 5.752        | 0.800        | 6.230                     | 0.400        |
|                  |                        | B    | 5.740        | <b>0.819</b> | 6.276                     | 0.358        |
|                  | I $\cup$ III           | —    | 5.921        | <b>0.819</b> | 6.505                     | 0.223        |
|                  |                        | +    | <b>5.738</b> | 0.812        | 6.003                     | <b>0.477</b> |
|                  |                        | B    | 5.750        | <b>0.819</b> | <sup>†</sup> <b>5.998</b> | 0.423        |

Table 2: Test-set performance for various models, measured using mean absolute error (MAE) and Pearson’s correlation ( $r$ ), for two prediction tasks.

- I.  $n$ -grams. We considered unigrams, bigrams, and trigrams. A 25-word stoplist was used; bigrams and trigrams were only filtered if all words were stopwords.
- II. Part-of-speech  $n$ -grams. As with words, we added unigrams, bigrams, and trigrams. Tags were obtained from the Stanford part-of-speech tagger (Toutanova and Manning, 2000).
- III. Dependency relations. We used the Stanford parser (Klein and Manning, 2003) to parse the critic reviews and extract syntactic dependencies. The dependency relation features consist of just the relation part of a dependency triple  $\langle \text{relation, head word, modifier word} \rangle$ .

A combination of the meta and text features achieves the best performance both in terms of MAE and pearson  $r$ .

We consider three ways to combine the collection of reviews for a given movie. The first (“—”) simply concatenates all of a movie’s reviews into a single document before extracting features. The second (“+”) conjoins each feature with the source site (e.g., *New York Times*) from whose review it was extracted. A third version (denoted “B”) combines both the site-agnostic and site-specific features.

# More Ways for Measuring Regression Predictions: Correlation Coefficient

- Pearson correlation coefficient

$$r(x, y) = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}}$$

where  $\bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$  and  $\bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$ .

$$|r(x, y)| \leq 1$$

- For regression:  $r(\vec{y}_{\text{predicted}}, \vec{y}_{\text{known}})$

- Measuring the **linear** correlation between two sequences, x and y,
- giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation.

# Advantage of Elastic net (Extra)

- Native Elastic set can be converted to lasso with augmented data form

$$p \gg n \Rightarrow X_{n \times p} \text{ (when } n \ll p \text{)}$$

- In the augmented formulation,

$$\Rightarrow X^*_{(n+p) \times p}$$

- sample size  $n+p$  and  $X^*$  has rank  $p$
  - ➔ can potentially select all the predictors
- Naïve elastic net can perform automatic variable selection like lasso

# Summary:

## Regularized multivariate linear regression

• Model:  $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$

• LR estimation:  $\arg \min_{\beta} \sum \left( Y - \hat{Y} \right)^2$

• LASSO estimation:  $\arg \min \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \sum_{j=1}^p |\beta_j|$

• Ridge regression estimation:  $\arg \min_{\beta} \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2$

Error on data

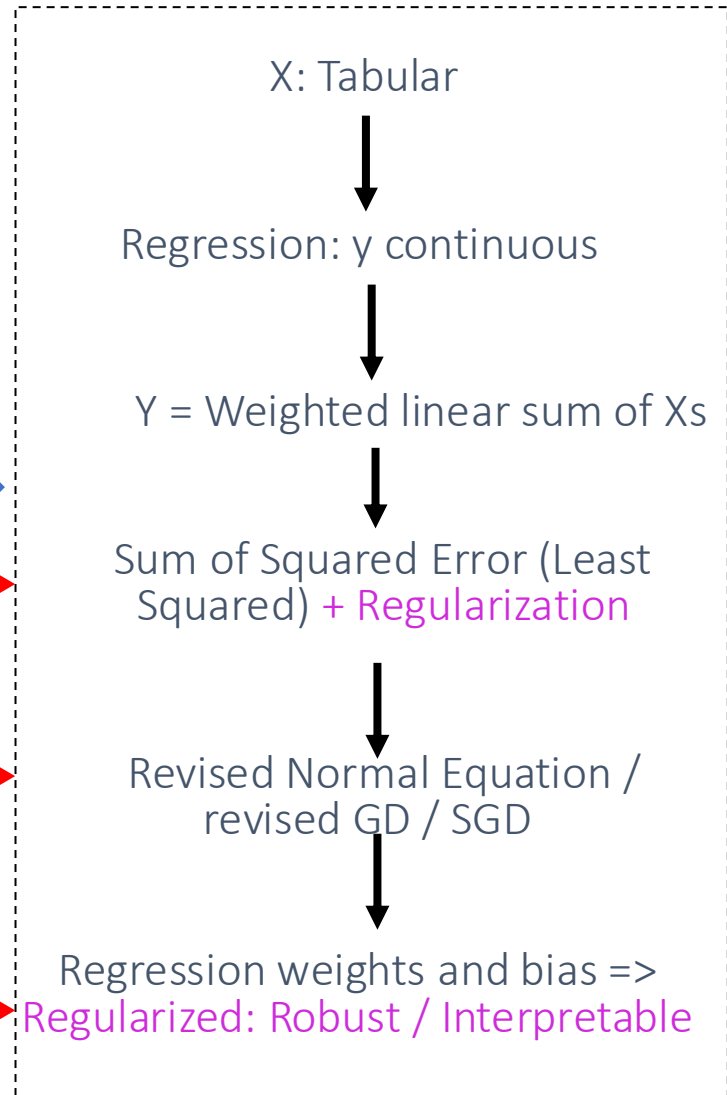
+

Regularization

98/54

## Today: Regularized multivariate linear regression

Data: X  
↓  
Task: y  
↓  
Representation: : x, f()  
↓  
Score Function: L()  
↓  
Search/Optimization : argmin()  
↓  
Models, Parameters



$$\min J(\beta) = \sum_{i=1}^n \left( Y - \hat{Y} \right)^2 + \lambda \left( \sum_{j=1}^p \beta_j^q \right)^{1/q}$$

## More: A family of shrinkage estimators

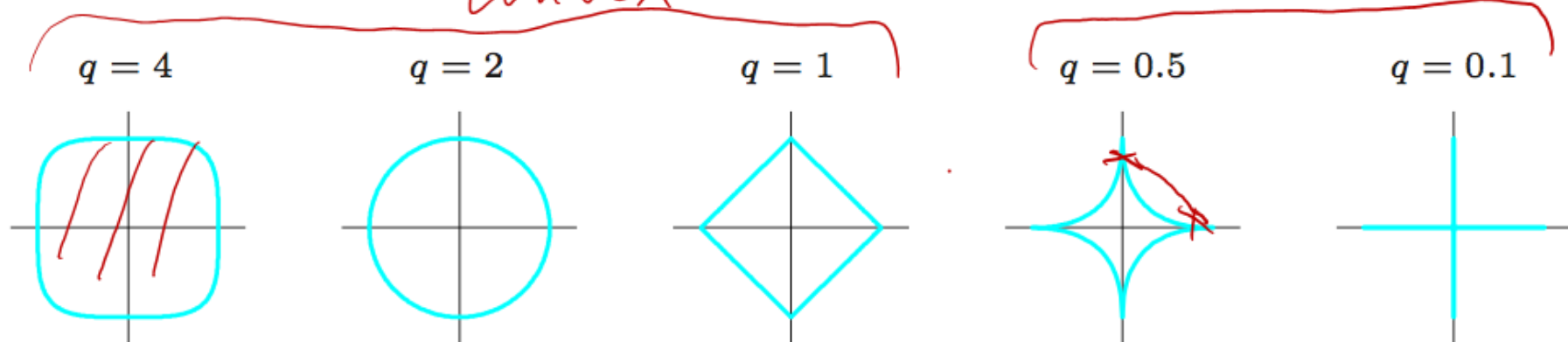
$$\beta = \arg \min_{\beta} \sum_{i=1}^N (y_i - x_i^T \beta)^2$$

$$\text{subject to } \sum_j |\beta_j|^q \leq s$$

- for  $q \geq 0$ , contours of constant value of  $\sum_j |\beta_j|^q$  are shown for the case of two inputs.

$$\sum_j |\beta_j|^q$$

*convex*



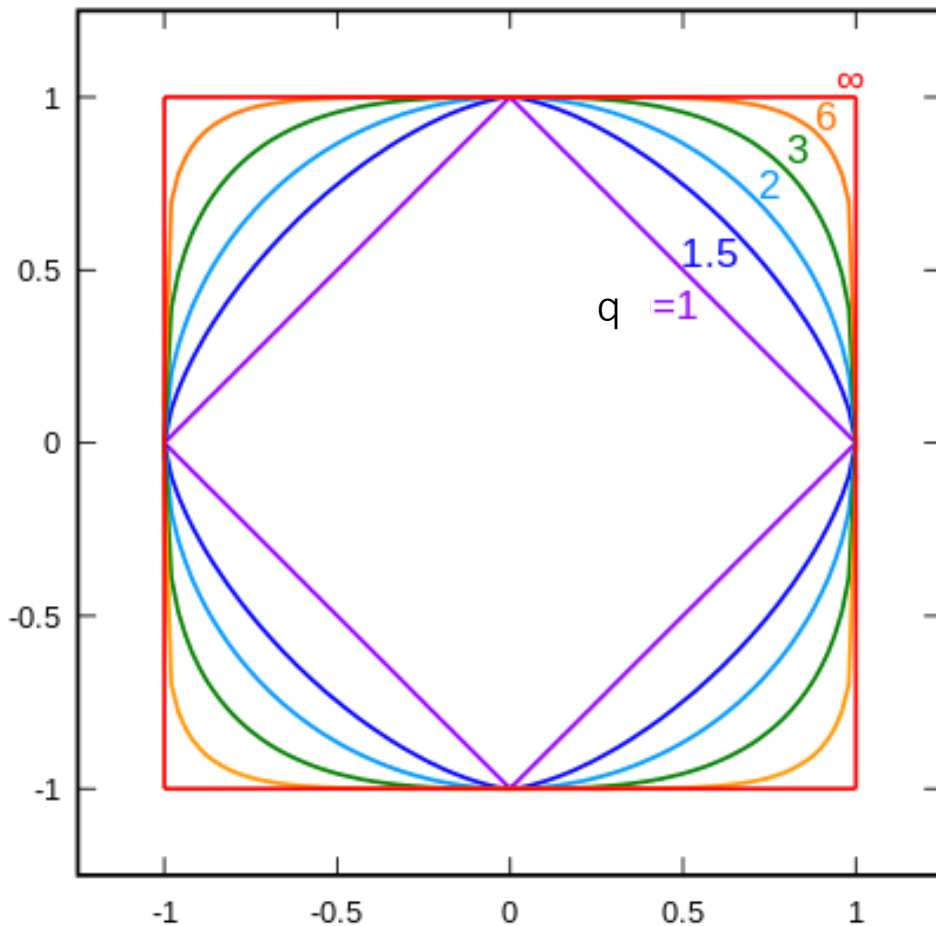
**FIGURE 3.12.** Contours of constant value of  $\sum_j |\beta_j|^q$  for given values of  $q$ .

$$\Sigma_{h \times p} \rightarrow \left( \begin{bmatrix} \Sigma_{h \times p} \\ \sqrt{\lambda_2} I_{p \times p} \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \textcircled{1}$$

norms visualized

$$\Sigma_{(h+p) \times p}^*$$

(2) group  $L_1$  norm



all p-norms penalize larger weights

$q < 2$  tends to create sparse (i.e. lots of 0 weights)

$q > 2$  tends to push for similar weights

We aim to make the learned model

- 1. Generalize Well

$$\mathcal{P} \rightarrow \mathcal{P}'$$

reduce model variance

- 2. Computationally Scalable and Efficient

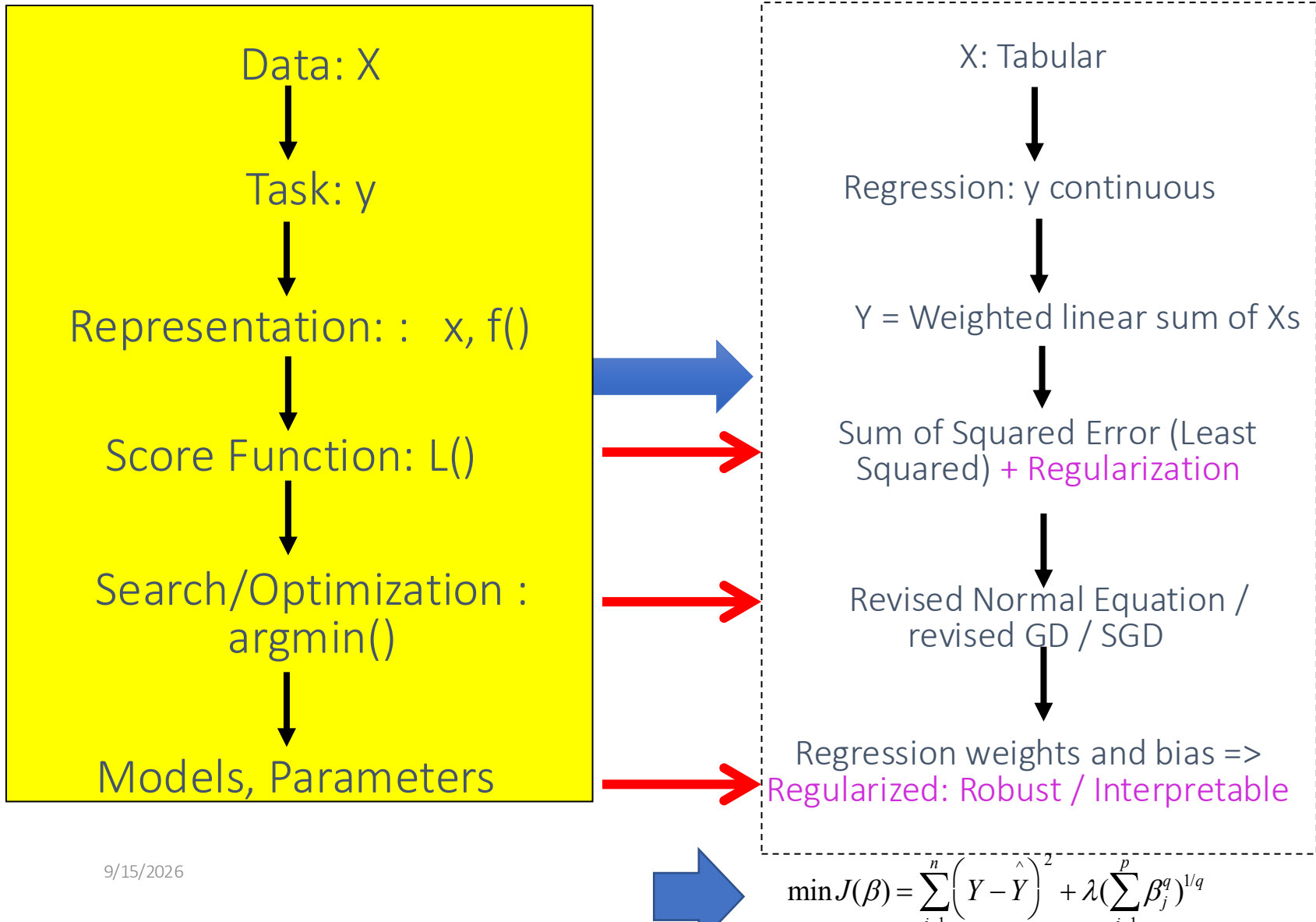
$$\frac{n \times \mathcal{P}'}{t_s}$$

- 3. Robust / Trustworthy / Interpretable
  - Especially for some domains, this is about trust!

# Roadmap: Linear Regression with Regularizations

- ✓ When  $p > n$ : How is Ordinary Least squares?
- ✓ Ridge regression: squared loss with L2 regularization
- ✓ Lasso regression: squared loss with L1 regularization
- ✓ Elastic regression: squared loss with L1 AND L2 regularization
- ➔ ✓ How to Choose Regularization Parameter

## Today: Regularized multivariate linear regression



Common regularizers

$$x_1 = x_2 \Rightarrow \beta_1 x_1 + \beta_2 x_2$$
$$\beta_1 + \beta_2 = 0$$

L2: Squared weights penalizes large values more

$$\sum_j |\beta_j|$$

L1: Sum of weights will penalize small values more

$$\sum_j \beta_j^2$$

Generally, we don't want huge weights

If weights are large, a small change in a feature can result in a large change in the prediction

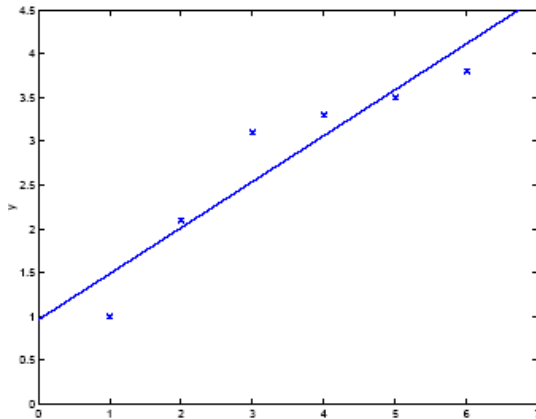
Might also prefer weights of 0 for features that aren't useful

# Model Selection & Generalization

- **Generalisation**: learn function / hypothesis from **past data** in order to “explain”, “predict”, “model” or “control” **new** data examples
- Underfitting: when model is too simple, both training and test errors are large
- Overfitting: when model is too complex and test errors are large although training errors are small.
  - After learning knowledge, model tends to learn “**noise**”

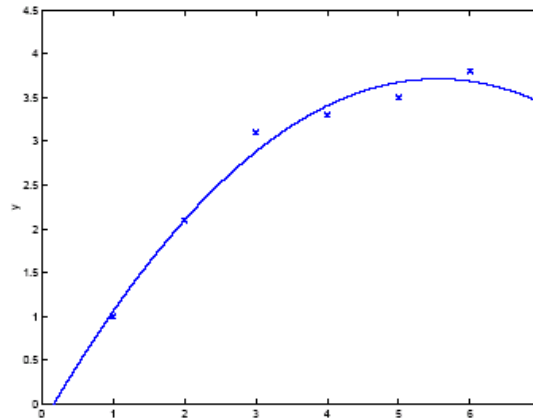
# Issue: Overfitting and underfitting

Under fit



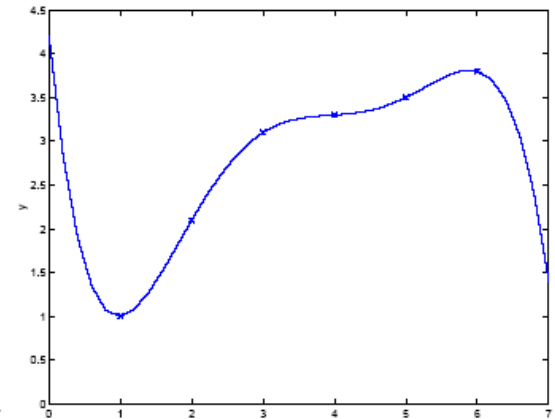
$$y = \theta_0 + \theta_1 x$$

Looks good



$$y = \theta_0 + \theta_1 x + \theta_2 x^2$$

Over fit



$$y = \sum_{j=0}^5 \theta_j x^j$$

Generalisation: learn function / hypothesis from past data in order to “explain”, “predict”, “model” or “control” new data examples

K-fold Cross Validation !!!!

# Overfitting: Handled by Regularization

A **regularizer** is an additional criteria to the loss function to make sure that we don't overfit

It's called a regularizer since it tries to keep the parameters more normal/regular

It is a bias on the model forces the learning to prefer certain types of weights over others, e.g.,

$$\hat{\beta}^{ridge} = \operatorname{argmin}_{\beta} \sum_{i=1}^n (y_i - x_i^T \beta)^2 + \lambda \beta^T \beta$$

# WHY and How to Select $\lambda$ ?

- 1. Generalization ability  
    ➔ k-folds CV to decide
- 2. Control the bias and Variance of the model (details in future lectures)

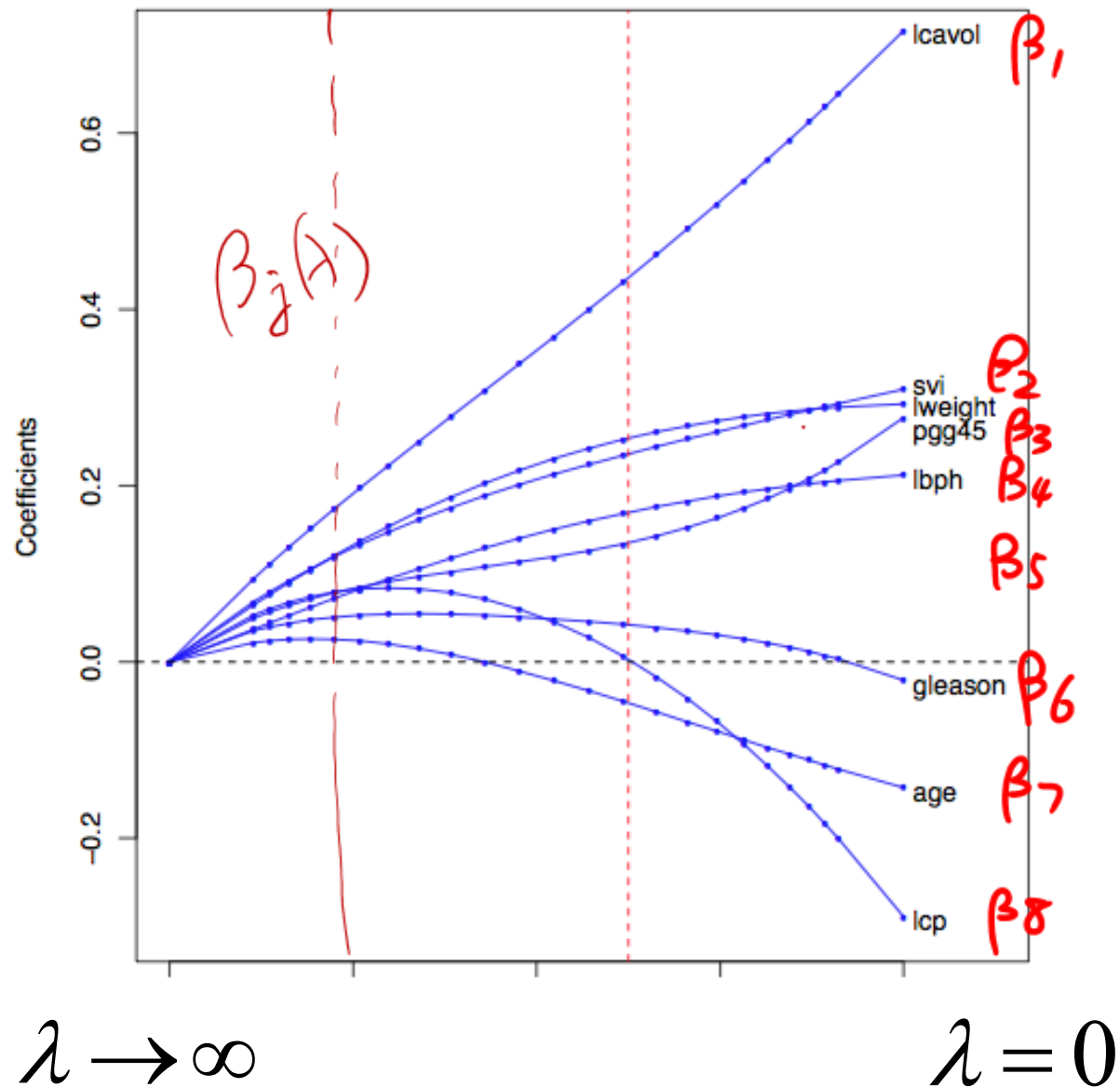
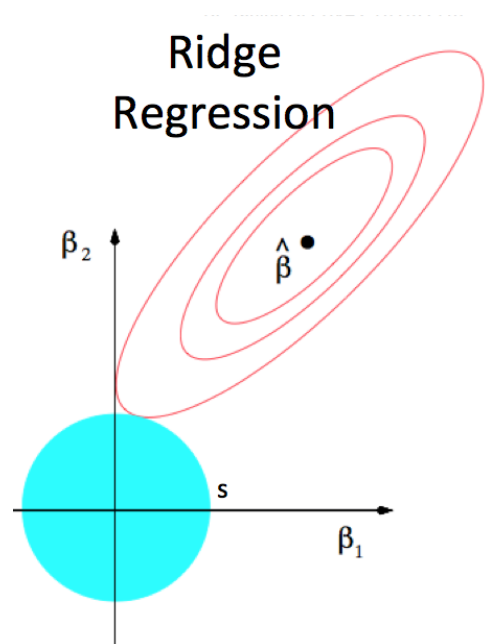
L2: Squared weights penalizes large values more

L1: Sum of weights will penalize small values more

$$\sum_j \left| \beta_j \right|$$

$$\sum_j \beta_j^2$$

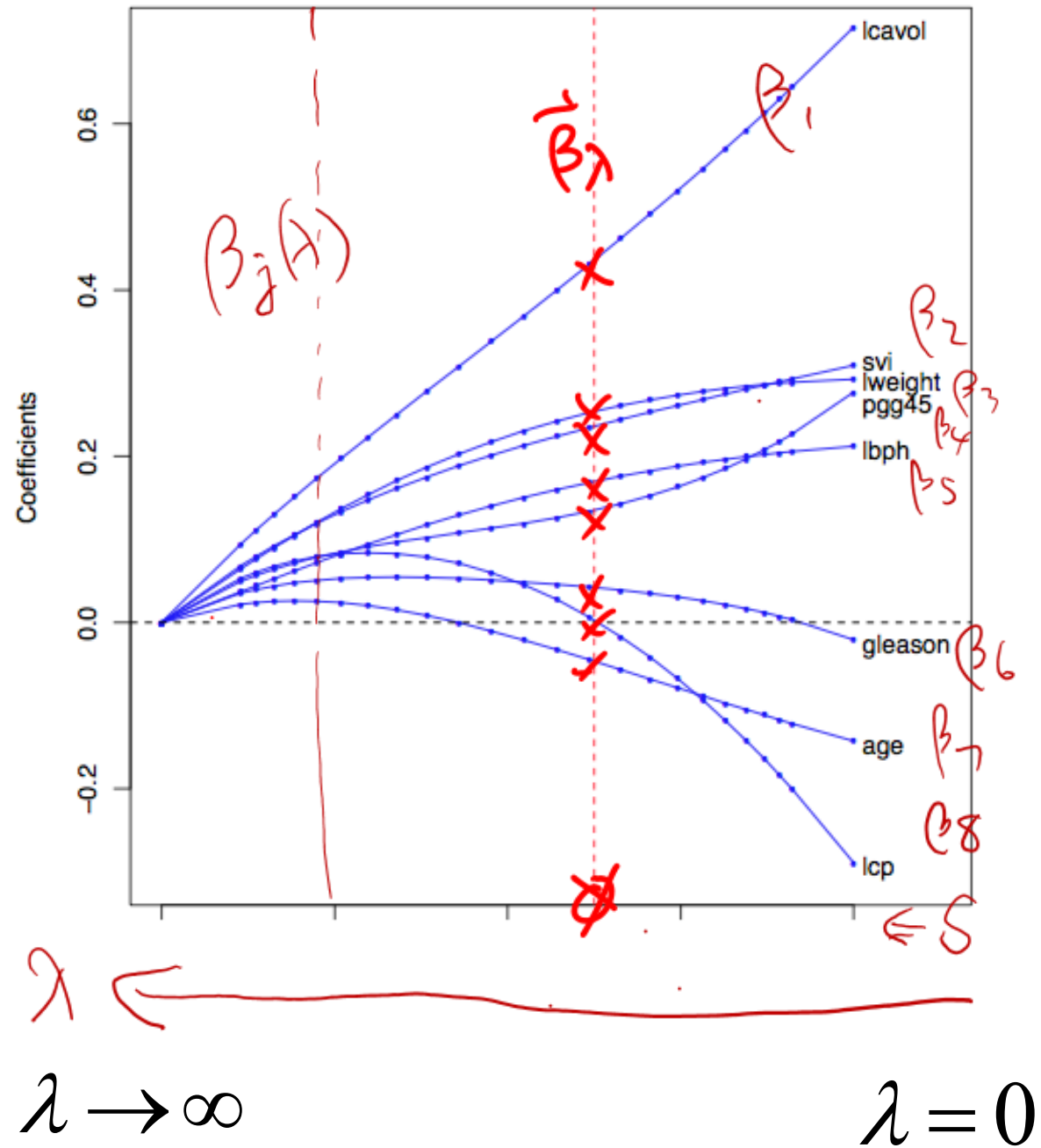
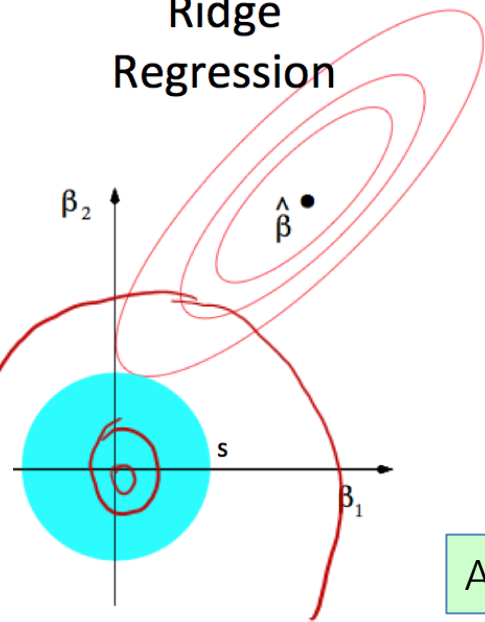
# Regularization path of a Ridge Regression



When  $\bar{X}'X = I \Rightarrow \frac{1}{1+\lambda} \beta_{OLS}$

Weight Decay

# Ridge Regression

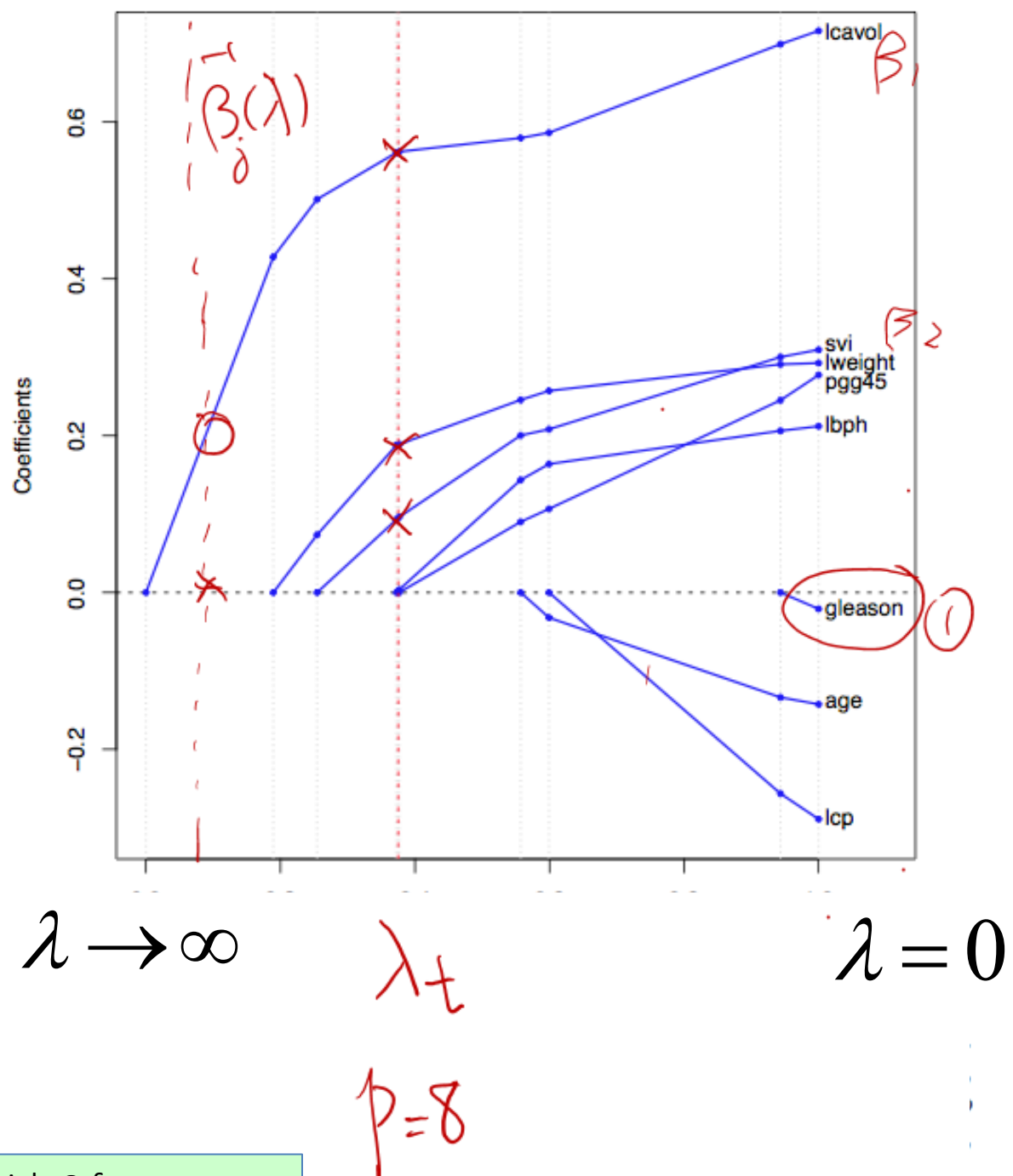
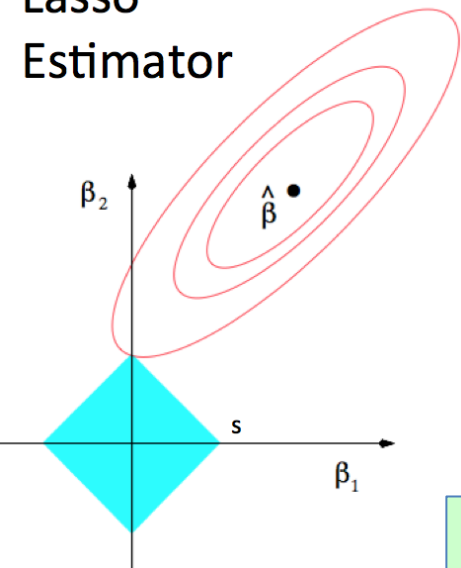


An example with 8 features

# Regularization path of a Lasso Regression

when varying  $\lambda$ ,  
how  $\beta_j$  varies.

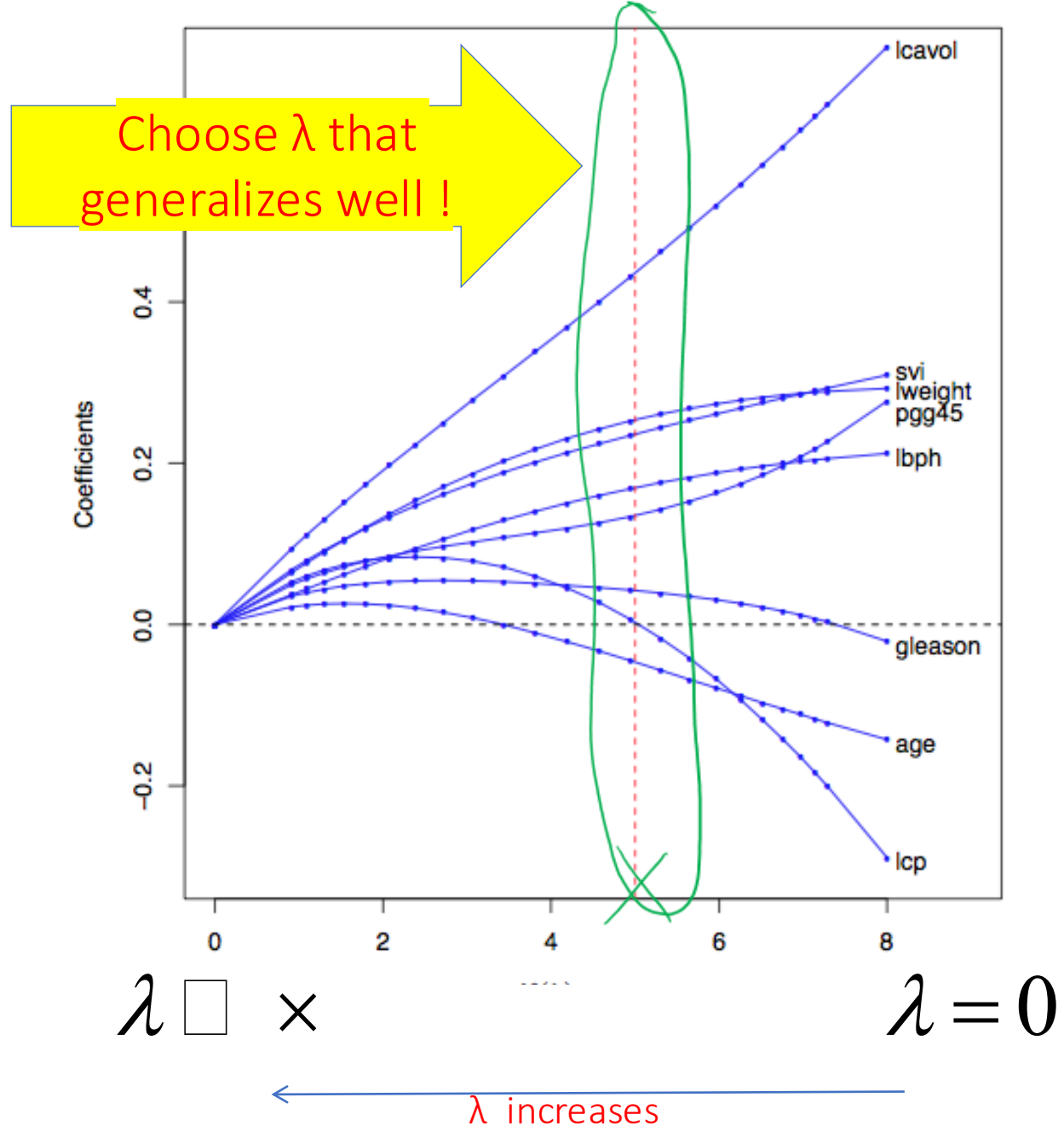
Lasso  
Estimator



An example with 8 features

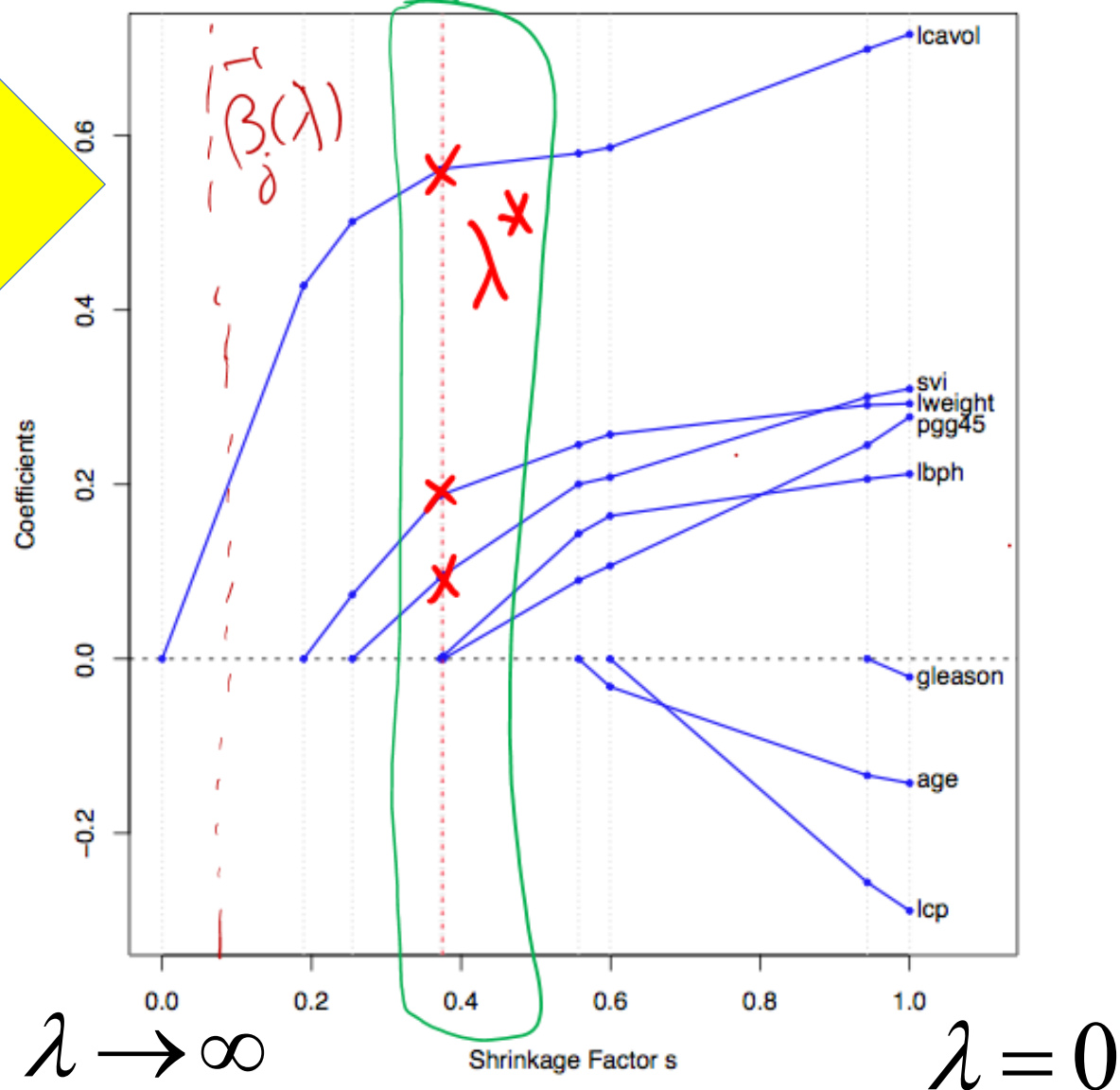
An example  
of  
Ridge Regression

when varying  
 $\lambda$ , how  $\beta_j$   
varies.



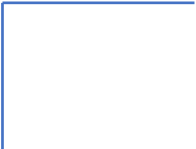
Choose  $\lambda$  that generalizes well !

when varying  $\lambda$ ,  
how  $\beta_j$  varies.



**FIGURE 3.10.** Profiles of lasso coefficients, as the tuning parameter  $t$  is varied. Coefficients are plotted versus  $s = t / \sum_1^p |\hat{\beta}_j|$ . A vertical line is drawn at  $s = 0.36$ , the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.

An example with 8 features



# Extra on Optimization of Regularized Regression Models

# Extra More Roadmap

- Optimization of regularized regressions:
  - See L6-extra slide
- Relation between  $\lambda$  and  $s$ 
  - See L6-extra slide
- Why Elastic Net has a few nice properties
  - See L6-extra slide

# References

- ❑ Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- ❑ Prof. Nando de Freitas's tutorial slide
- ❑ Regularization and variable selection via the elastic net, Hui Zou and Trevor Hastie, Stanford University, USA
- ❑ ESL book: Elements of Statistical Learning

# Extra Recap



- ❑ More about LR Model with Regularizations

- ❑ Ridge Regression

- ❑ Lasso Regression

- ❑ Extra: how to perform training

- ❑ Elastic net

- ❑ Extra: how to perform training

# Why Invertible In Ridge Regression?

$$\beta^* = (X^T X + \lambda I)^{-1} X^T \bar{y}$$

(**NOT AN EASY PROOF from SVD angle**), many concepts, SVD, PCA, Eigenvalues, relation to singular

- NOT AN EASY PROOF If through SVD
  - [https://www.quora.com/When-is-the-matrix-frac{1}{n}X^T X + \lambda I\\_d \text{ invertible}](https://www.quora.com/When-is-the-matrix-frac{1}{n}X^T X + \lambda I_d \text{ invertible})

- The determinant of  $A$  is equal to the product of its eigenvalues,

$$|A| = \prod_{i=1}^n \lambda_i.$$

- The rank of  $A$  is equal to the number of non-zero eigenvalues of  $A$ .

# Why Invertible In Ridge Regression?

symmetric, positive semi-definite *square* Gram matrix  $K = A^T A$  — which can be naturally formed even when  $A$  is not square. Perhaps the eigenvalues of  $K$  might play a comparably important role for general matrices. Since they are not easily related to the eigenvalues of  $A$  — which, in the non-square case, don't even exist — we shall endow them with a new name.

**Definition 6.27.** The *singular values*  $\sigma_1, \dots, \sigma_r$  of an  $m \times n$  matrix  $A$  are the positive square roots,  $\sigma_i = \sqrt{\lambda_i} > 0$ , of the nonzero eigenvalues of the associated Gram matrix  $K = A^T A$ . The corresponding eigenvectors of  $K$  are known as the *singular vectors* of  $A$ .

Since  $K$  is necessarily positive semi-definite, its eigenvalues are always non-negative,  $\lambda_i \geq 0$ , which justifies the positivity of the singular values of  $A$  — independently of whether  $A$  itself has positive, negative, or even complex eigenvalues — or is rectangular and has no eigenvalues at all. The standard convention is to label the singular values in decreasing order, so that  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ . Thus,  $\sigma_1$  will always denote the largest or

# Extra: Intercept Term is usually not shrinked

- If the data is not centered, there exists bias term
  - <http://stats.stackexchange.com/questions/86991/reason-for-not-shrinking-the-bias-intercept-term-in-regression>
- We normally assume we centered x and y. If this is true, no need to have bias term, e.g., for lasso,

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

*For ridge, in implementation,  
just set the bias  
corresponding entry  
as 0 in the I-  
matrix*

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1 \|\beta\|_1$$

*for ridge  
 $+ \lambda \|\beta\|_2^2$*

# Extra Recap

- ❑ More about LR Model with Regularizations

- ❑ Ridge Regression

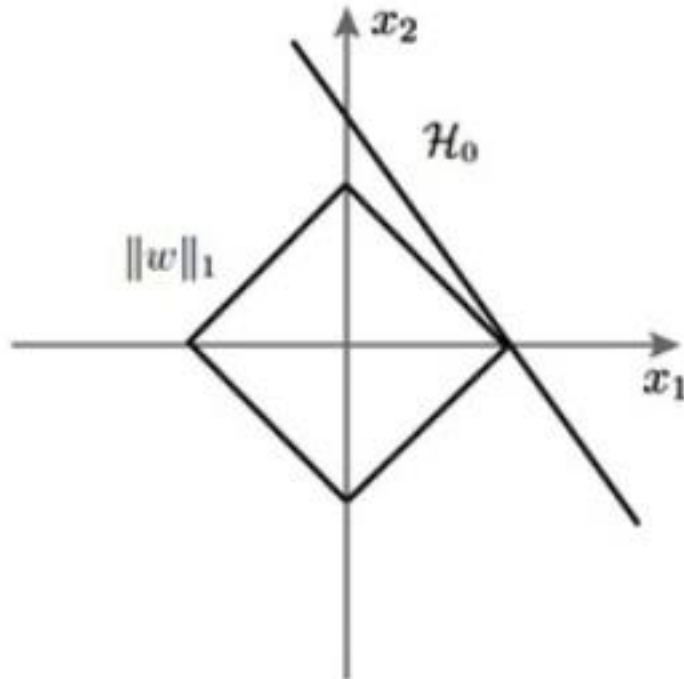
- ❑ Lasso Regression

- ❑ Extra: how to perform training

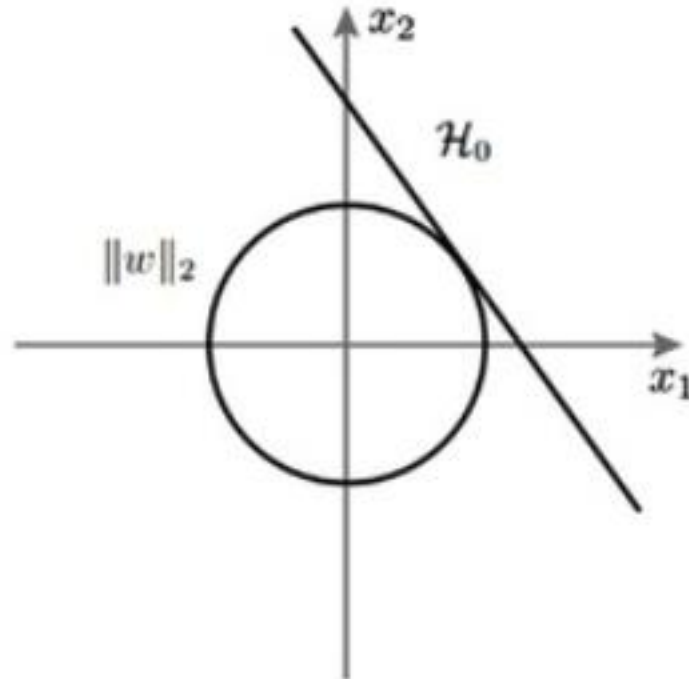
- ❑ Elastic net

- ❑ Extra: how to perform training

**A** L1 regularization



**B** L2 regularization



due to the nature of L<sub>1</sub> norm, the viable solutions are limited to corners, **which are on a few axis only**  
- **in the above case x1. Value of x2 = 0.** This means that the solution has eliminated the role of x2, leading to sparsity

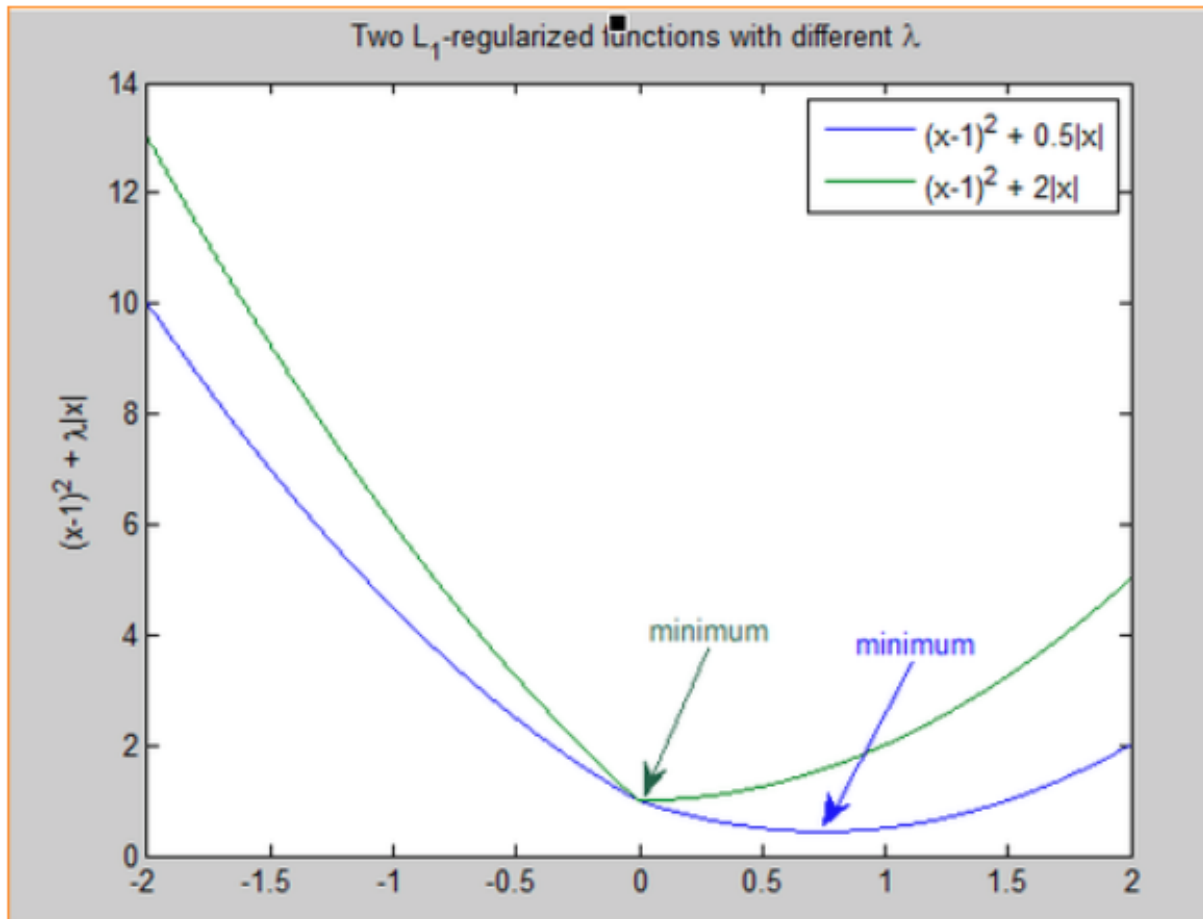
$L_2$ -regularized loss function  $F(x) = f(x) + \lambda \|x\|_2^2$  is smooth. This means that the optimum is the stationary point (0-derivative point). The stationary point of  $F$  can get very small when you increase  $\lambda$ , but still won't be 0 unless  $f'(0) = 0$ .

$L_1$ -regularized loss function  $F(x) = f(x) + \lambda \|x\|_1$  is non-smooth. It's not differentiable at 0. Optimization theory says that the optimum of a function is either the point with 0-derivative or one of the irregularities (corners, kinks, etc.). So, it's possible that the optimal point of  $F$  is 0 even if 0 isn't the stationary point of  $f$ . In fact, it would be 0 if  $\lambda$  is large enough (stronger regularization effect). Below is a graphical illustration.

In multi-dimensional settings: if a feature is not important, the loss contributed by it is small and hence the (non-differentiable) regularization effect would turn it off.

$L_1$ -regularized loss function  $F(x) = f(x) + \lambda\|x\|_1$  is non-smooth. It's not differentiable at 0. Optimization theory says that the optimum of a function is either the point with 0-derivative or one of the irregularities (corners, kinks, etc.). So, it's possible that the optimal point of  $F$  is 0 even if 0 isn't the stationary point of  $f$ . In fact, it would be 0 if  $\lambda$  is large enough (stronger regularization effect). Below is a graphical illustration.

<http://www.quora.com/What-is-the-difference-between-L1-and-L2-regularization>



In mathematics, particularly in calculus, a stationary point or critical point of a differentiable function of one variable is a point of the domain of the function where the derivative is zero (equivalently, the slope of the graph at that point is zero).

## How to train Parameter for Lasso

$$\hat{\beta}^{lasso} = \arg \min (y - X\beta)^T (y - X\beta)$$

subject to  $\square |\beta_j| \leq s$

- **$\ell_1$ -norm is non differentiable!**

- cannot compute the gradient of the absolute value

$\Rightarrow$  **Directional derivatives** (or subgradient)

Here assume  $x$  and  $y$  have been centered (normally), therefore no bias term needed in above !

$$\begin{aligned}
\underline{\text{RSS loss}}(\lambda) &= (y - X\beta)^T (y - X\beta) + \lambda \sum_{j=1}^p |\beta_j| \\
&= \sum_{i=1}^n (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| \\
&= \left[ \sum_{i=1}^n (y_i - \underbrace{x_{ij} \beta_j}_{\text{green}} - \underbrace{x_{i(-j)}^T \beta_{(-j)}}_{\text{green}}) \right]^2 + \underbrace{\lambda \sum_{j=1}^p |\beta_j|}_{\text{green}}
\end{aligned}$$

if  $\beta = (\beta_1, \beta_2, \beta_3)$

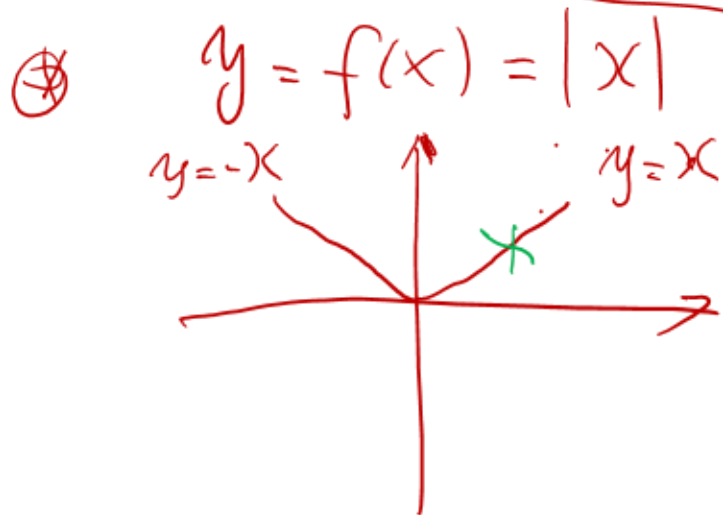
$\Rightarrow \beta_{-2} = (\beta_1, \beta_3)$

$$\begin{aligned}
\Rightarrow \frac{\partial l}{\partial \beta_j} &= \sum_{i=1}^n \underbrace{2(y_i - x_{ij} \beta_j - x_{i(-j)}^T \beta_{(-j)})}_{\text{green}} \underbrace{(-x_{ij})}_{\text{green}} \\
&\quad + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|
\end{aligned}$$

$$= \underbrace{2 \sum_{i=1}^n x_{ij}^2 \beta_j}_{a_j} - \underbrace{2 \sum_{i=1}^n (y_i - x_i^T \beta) x_{ij}}_{C_j} + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|$$

$$= a_j \beta_j - C_j + \lambda \frac{\partial}{\partial \beta_j} |\beta_j| \quad \underline{\underline{\text{Set to } 0}}$$

convex  $\Rightarrow$  unique



$$\partial f(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \end{cases}$$

$$\frac{\partial \mathcal{L}}{\partial \beta_j} = \begin{cases} a_j \beta_j - c_j - \lambda & , \text{ if } \beta_j < 0 \\ a_j \beta_j - c_j + \lambda & , \text{ if } \beta_j > 0 \\ [a_j \beta_j - c_j - \lambda, a_j \beta_j - c_j + \lambda] & , \text{ if } \beta_j = 0 \end{cases}$$

Set to 0

$$\hat{\beta}_j = \begin{cases} \frac{c_j + \lambda}{a_j} & , \text{ if } c_j + \lambda < 0 \Rightarrow c_j < -\lambda \\ \frac{c_j - \lambda}{a_j} & , \text{ if } c_j > \lambda \\ 0 & , \text{ if } -\lambda \leq c_j \leq \lambda \end{cases}$$

Soft thresholding

We just need 0 in the region  $[-c_j - \lambda, -c_j + \lambda]$  (subgradient calculus)

$$\begin{aligned}
 RSS(\lambda) &= (y - x\beta)^T (y - x\beta) + \lambda \sum_{j=1}^p |\beta_j| \\
 &= \sum_{i=1}^n (y_i - \underbrace{x_i^T \beta}_{\hat{y}}) + \lambda \sum_{j=1}^p |\beta_j| \\
 &= \left[ \sum_{i=1}^n (y_i - x_{ij}\beta_j - x_{i-j}^T \beta_{-j})^2 \right] + \left[ \lambda \sum_{j=1}^p |\beta_j| \right]
 \end{aligned}$$

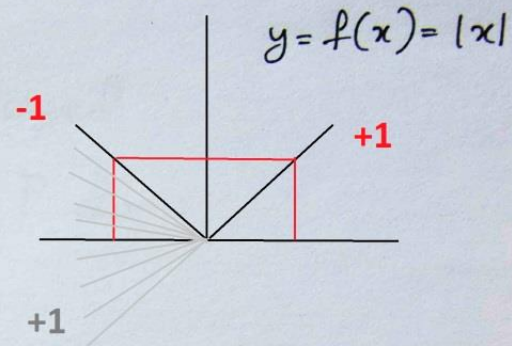
Here, if  $\beta = (\beta_1, \beta_2, \beta_3) \Rightarrow \beta_{-2} = (\beta_1, \beta_3)$

$$\begin{aligned}
 \frac{\partial}{\partial \beta_j} RSS(\lambda) &= \sum_{i=1}^n 2(y_i - x_{ij}\beta_j - x_{i-j}^T \beta_{-j})(-x_{ij}) \\
 &\quad + \lambda \frac{\partial}{\partial \beta_j} (|\beta_1| + |\beta_2| + \dots + |\beta_p|) \\
 &= \underbrace{2 \sum_{i=1}^n x_{ij}^2}_{a_j} \beta_j - \underbrace{2 \sum_{i=1}^n (y_i - x_{i-j}^T \beta_{-j}) x_{ij}}_{c_j} + \lambda \frac{\partial}{\partial \beta_j} |\beta_j| \\
 &= a_j \beta_j - c_j + \lambda \frac{\partial}{\partial \beta_j} |\beta_j|
 \end{aligned}$$

# Sub differentials

$$f(x) = |x|$$

$$\partial f(x) = \begin{cases} \{-1\} & \text{if } x < 0 \\ \{-1, 1\} & \text{if } x = 0 \\ \{+1\} & \text{if } x > 0 \end{cases}$$



$$\begin{aligned} \frac{\partial}{\partial \beta_j} \text{RSS}(\lambda) &= a_j \beta_j + e_j + \lambda \frac{\partial}{\partial \beta_j} |\beta_j| \\ &= \begin{cases} \{a_j \beta_j - e_j - \lambda\} & \text{if } \beta_j < 0 \\ \{-e_j - \lambda, -e_j + \lambda\} & \text{if } \beta_j = 0 \\ \{a_j \beta_j - e_j + \lambda\} & \text{if } \beta_j > 0 \end{cases} \end{aligned}$$

$$\hat{\beta}_j = \begin{cases} (e_j + \lambda) / a_j & \text{if } e_j < -\lambda \\ 0 & \text{if } e_j \in [-\lambda, \lambda] \\ (e_j - \lambda) / a_j & \text{if } e_j > \lambda \end{cases} \left| \begin{array}{l} \text{When } \beta_j < 0 \\ a_j \beta_j - e_j - \lambda = 0 \\ \beta_j = \frac{e_j + \lambda}{a_j} \\ \text{When } \beta_j > 0 \\ a_j \beta_j - e_j + \lambda = 0 \\ \beta_j = \frac{e_j - \lambda}{a_j} \end{array} \right.$$

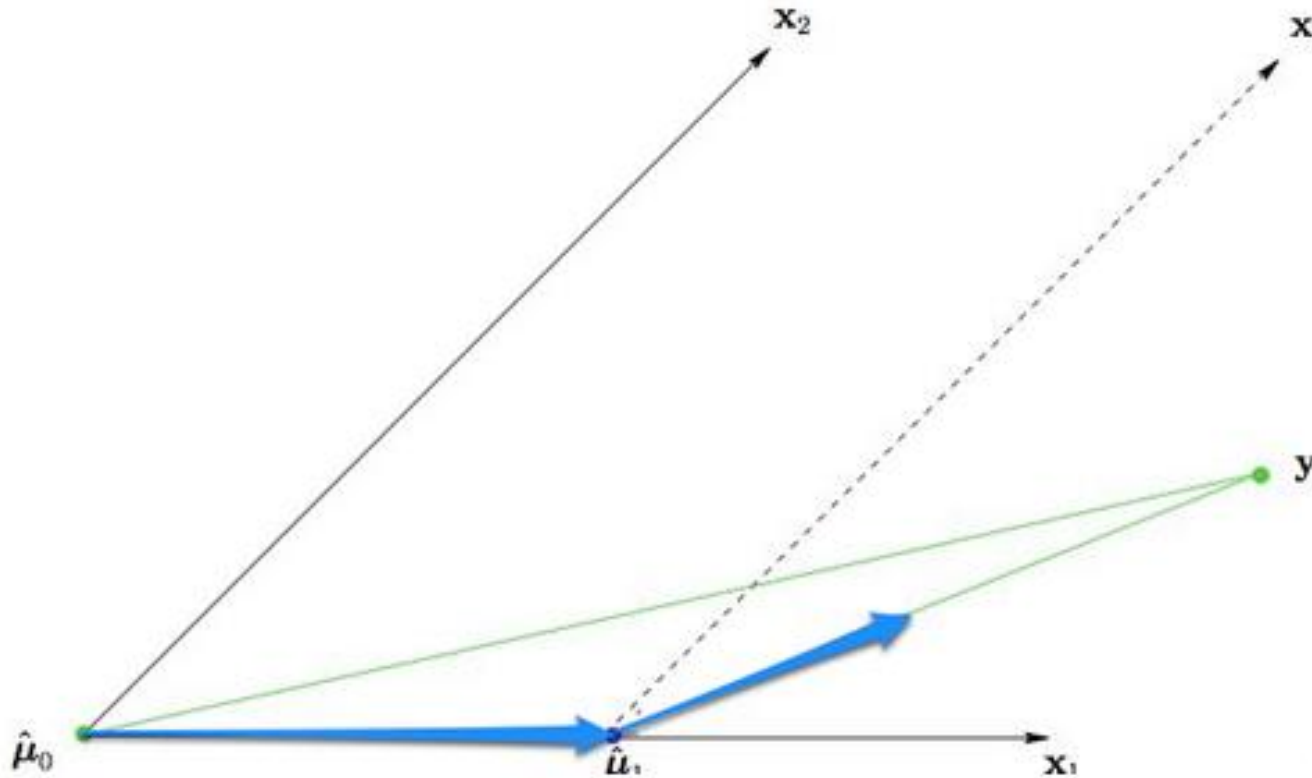
# Coordinate descent based Learning of Lasso

1. Initialize  $\beta$
2. Repeat until converged
3. For  $j = 1, 2, \dots, P$  do
  - $a_j = 2 \sum_{i=1}^n x_{ij}^2$
  - $e_j = 2 \sum_{i=1}^n x_{ij} (y_i - x_i^T \beta + x_{ij} \beta_j)$
  - if  $e_j < -\lambda$ 
$$\beta_j = (e_j + \lambda) / a_j$$
  - else if,  $e_j > \lambda$ 
$$\beta_j = (e_j - \lambda) / a_j$$
  - else, soft-thresholding
$$\beta_j = 0$$

Coordinate descent (WIKI) → one does line search along one coordinate direction at the current point in each iteration.

One uses different coordinate directions cyclically throughout the procedure.

# Least Angle Regression (LARS) (State-of-the-art LASSO solver)



<http://statweb.stanford.edu/~tibs/ftp/lars.pdf>

# LARS: Least Angle Regression

- Starts like classic Forward Selection
  - Find predictor  $x_{j_1}$  most correlated with the current residual
  - Make a step (epsilon) large enough until another predictor  $x_{j_2}$  has as much correlation with the current residual
  - LARS – now step in the direction equiangular between two predictors until  $x_{j_3}$  earns its way into the “correlated set”

Correlation: 
$$c(\mu) = X'(y - \mu)$$

# Extra Recap

- ❑ More about LR Model with Regularizations

- ❑ Ridge Regression

- ❑ Lasso Regression

- ❑ Extra: how to perform training

- ❑ Elastic net

- ❑ Extra: how to perform training



# Connecting LASSO and Naïve Elastic net

- Lemma: Given  $(\lambda_1, \lambda_2)$ , define an artificial data set  $(y^*, X^*)$

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

Let  $\gamma = \lambda_1 / \sqrt{(1 + \lambda_2)}$  and  $\beta^* = \sqrt{(1 + \lambda_2)} \beta$ . Then the naïve elastic net criterion can be written as

$$L(\gamma, \beta) = L(\gamma, \beta^*) = |\mathbf{y}^* - \mathbf{X}^* \beta^*|^2 + \gamma |\beta^*|_1.$$

- Let,

naive 
$$\hat{\beta}^* = \arg \min_{\beta^*} L\{(\gamma, \beta^*)\};$$

- Then

$$\hat{\beta} = \frac{1}{\sqrt{(1 + \lambda_2)}} \hat{\beta}^*.$$

# Connecting LASSO and Naïve Elastic net

- Lemma: Given  $(\lambda_1, \lambda_2)$ , define an artificial data set  $(y^*, X^*)$

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

Let  $\gamma = \lambda_1 / \sqrt{1 + \lambda_2}$  and  $\beta^* = \sqrt{1 + \lambda_2} \beta$ . Then the naïve elastic net criterion can be written as

$$L(\gamma, \beta) = L(\gamma, \beta^*) = \left[ |y^* - X^* \beta^*|^2 + \gamma |\beta^*|_1 \right] \Rightarrow \beta^*$$

- Let,

$$\text{naive } \hat{\beta}^* = \arg \min_{\beta^*} L\{(\gamma, \beta^*)\};$$

- Then

elastic  $\hat{\beta} = \frac{1}{\sqrt{1 + \lambda_2}} \hat{\beta}^*$  *lasso augmented*

$$\begin{matrix} \cancel{X} \beta = y \\ \begin{matrix} n \times p & p \times 1 & n \times 1 \\ (n+p) \times p & p \times 1 & (n+p) \times 1 \end{matrix} \end{matrix}$$

$$Loss = |y^* - X^* \beta^*|^2 + r |\beta^*|_1$$

$$= \left| \begin{bmatrix} y \\ 0 \end{bmatrix} - \underbrace{\begin{pmatrix} X \\ \sqrt{\lambda_2} I \end{pmatrix}}_{(n+p) \times p} \beta^* \right|^2 + r |\beta^*|_1$$

$$= \left| \begin{pmatrix} y - X \beta^* \\ -\sqrt{\lambda_2} \beta^* \end{pmatrix} \right|^2 + r |\beta^*|_1$$

$$= (y - X \beta^*)^2 + \lambda_2 \beta^{*2} + r |\beta^*|_1$$

$$= (y - X \beta^*)^2 + \lambda_2 (1 + \lambda_2) \beta^2 + \frac{\lambda_1}{\sqrt{1 + \lambda_2}} \left| \frac{\lambda_1}{\sqrt{1 + \lambda_2}} \beta \right|_1$$

# Advantage of Elastic net $p \gg n$

- Native Elastic set can be converted to lasso with augmented data

$$\Rightarrow X_{n \times p} \text{ (when } n \ll p \text{)}$$

- In the augmented formulation,  $\Rightarrow X^*$ 
  - sample size  $n+p$  and  $X^*$  has rank  $p$
  - $\Rightarrow$  can potentially select all the predictors

$$(n+p) \times p$$

- Naïve elastic net can perform automatic variable selection like lasso

# Grouping Effect

Qualitatively speaking, a regression method exhibits the grouping effect if the regression coefficients of a group of highly correlated variables tend to be equal (up to a change of sign if negatively correlated). In particular, in the extreme situation where some variables are exactly identical, the regression method should assign identical coefficients to the identical variables.

If there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected.

uations. We illustrate our points by considering the gene selection problem in microarray data analysis. A typical microarray data set has many thousands of predictors (genes) and often fewer than 100 samples. For those genes sharing the same biological 'pathway', the correlations between them can be high (Segal and Conklin, 2003). We think of those genes as forming a group. The ideal gene selection method should be able to do two things: eliminate the trivial genes and automatically include whole groups into the model once one gene among them is selected ('grouped selection'). For this kind of  $p \gg n$  and grouped variables situation, the lasso is not the ideal method, because it can only select at most  $n$  variables out of  $p$  candidates (Efron *et al.*, 2004), and it lacks the ability to reveal the grouping information. As for prediction per-

# Grouping Effect of Naïve Elastic net

$$\hat{\beta} = \arg \min_{\beta} |y - X\beta|^2 + \lambda J(\beta)$$

- Consider the following penalized regression model: Where  $J(\cdot)$  positive for  $\beta \neq 0$ .

*Lemma 2.* Assume that  $\mathbf{x}_i = \mathbf{x}_j$ ,  $i, j \in \{1, \dots, p\}$ .

- (a) If  $J(\cdot)$  is strictly convex, then  $\hat{\beta}_i = \hat{\beta}_j$ ,  $\forall \lambda > 0$ .
- (b) If  $J(\beta) = |\beta|_1$ , then  $\hat{\beta}_i \hat{\beta}_j \geq 0$  and  $\hat{\beta}^*$  is another minimizer of equation (7), where

$$\hat{\beta}_k^* = \begin{cases} \hat{\beta}_k & \text{if } k \neq i \text{ and } k \neq j, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot s & \text{if } k = i, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (1 - s) & \text{if } k = j, \end{cases}$$

for any  $s \in [0, 1]$ .

Lemma 2 shows a clear distinction between *strictly* convex penalty functions and the lasso penalty. Strict convexity guarantees the grouping effect in the extreme situation with identical predictors. In contrast the lasso does not even have a unique solution. The elastic net penalty with  $\lambda_2 > 0$  is strictly convex, thus enjoying the property in assertion (1).

# Grouping Effect of Naïve Elastic net

$$\hat{\beta} = \arg \min_{\beta} |y - X\beta|^2 + \lambda J(\beta)$$

- Consider the following penalized regression model: Where  $J(\cdot)$  positive for  $\beta \neq 0$ .

*Lemma 2.* Assume that  $\mathbf{x}_i = \mathbf{x}_j$ ,  $i, j \in \{1, \dots, p\}$ .

- (a) If  $J(\cdot)$  is strictly convex, then  $\hat{\beta}_i = \hat{\beta}_j$ ,  $\forall \lambda > 0$ .
- (b) If  $J(\beta) = |\beta|_1$ , then  $\hat{\beta}_i \hat{\beta}_j \geq 0$  and  $\hat{\beta}^*$  is another minimizer of equation (7), where

$$\hat{\beta}_k^* = \begin{cases} \hat{\beta}_k & \text{if } k \neq i \text{ and } k \neq j, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot s & \text{if } k = i, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (1 - s) & \text{if } k = j, \end{cases}$$

for any  $s \in [0, 1]$ .

Lasso does not provide a unique solution

Lemma 2 shows a clear distinction between *strictly* convex penalty functions and the lasso penalty. Strict convexity guarantees the grouping effect in the extreme situation with identical predictors. In contrast the lasso does not even have a unique solution. The elastic net penalty with  $\lambda_2 > 0$  is strictly convex, thus enjoying the property in assertion (a).

# Grouping Effect of Naïve Elastic net

*Theorem 1.* Given data  $(\mathbf{y}, \mathbf{X})$  and parameters  $(\lambda_1, \lambda_2)$ , the response  $\mathbf{y}$  is centred and the predictors  $\mathbf{X}$  are standardized. Let  $\hat{\beta}(\lambda_1, \lambda_2)$  be the naïve elastic net estimate. Suppose that  $\hat{\beta}_i(\lambda_1, \lambda_2) \hat{\beta}_j(\lambda_1, \lambda_2) > 0$ . Define

$$D_{\lambda_1, \lambda_2}(i, j) = \frac{1}{\|\mathbf{y}\|_1} |\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)|;$$

then

$$D_{\lambda_1, \lambda_2}(i, j) \leq \frac{1}{\lambda_2} \sqrt{2(1 - \rho)},$$

where  $\rho = \mathbf{x}_i^T \mathbf{x}_j$ , the sample correlation.

- $D$  is the difference between the coefficient paths of predictors  $i$  and  $j$ .
- If  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are high correlated  $\rho=1$ , this theorem provides a quantitative description for the grouping effect of Naive Elastic Net.

# Grouping Effect of Naïve Elastic net

*Theorem 1.* Given data  $(\mathbf{y}, \mathbf{X})$  and parameters  $(\lambda_1, \lambda_2)$ , the response  $\mathbf{y}$  is centred and the predictors  $\mathbf{X}$  are standardized. Let  $\hat{\beta}(\lambda_1, \lambda_2)$  be the naïve elastic net estimate. Suppose that  $\hat{\beta}_i(\lambda_1, \lambda_2) \hat{\beta}_j(\lambda_1, \lambda_2) > 0$ . Define

$$D_{\lambda_1, \lambda_2}(i, j) = \frac{1}{\|\mathbf{y}\|_1} |\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)|;$$

then

$$D_{\lambda_1, \lambda_2}(i, j) \leq \frac{1}{\lambda_2} \sqrt{2(1 - \rho)},$$

where  $\rho = \mathbf{x}_i^T \mathbf{x}_j$ , the sample correlation.

$$\rightarrow \frac{1}{\lambda_2} \sqrt{2(1 - \rho_{ij})}$$

$\mathbf{x}_i \quad \mathbf{x}_j$

- $D$  is the difference between the coefficient paths of predictors  $i$  and  $j$ .
- If  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are high correlated  $\rho=1$ , this theorem provides a quantitative description for the grouping effect of Naive Elastic Net.

# Elastic Net

In the regression prediction setting, an accurate penalization method achieves good prediction performance through the bias–variance trade-off. The naïve elastic net estimator is a two-stage procedure: for each fixed  $\lambda_2$  we first find the ridge regression coefficients, and then we do the lasso-type shrinkage along the lasso coefficient solution paths. It appears to incur a double amount of shrinkage. Double shrinkage does not help to reduce the variances much and introduces unnecessary extra bias, compared with pure lasso or ridge shrinkage. In the next section we improve the prediction performance of the naïve elastic net by correcting this double shrinkage.

- **Deficiency of the Naive Elastic Net:** Empirical evidence shows the Naive Elastic Net does not perform satisfactorily. The reason is that there are two shrinkage procedures (Ridge and LASSO) in it. Double shrinkage introduces unnecessary bias.
- Re-scaling of Naive Elastic Net gives better performance, yielding the Elastic Net solution:
- Reason: Undo shrinkage. 
$$\hat{\beta}(\text{ENet}) = (1 + \lambda_2) \cdot \hat{\beta}(\text{Naive ENet})$$

# Elastic Net

## 3.2. The elastic net estimate

We follow the notation in Section 2.2. Given data  $(\mathbf{y}, \mathbf{X})$ , penalty parameter  $(\lambda_1, \lambda_2)$  and augmented data  $(\mathbf{y}^*, \mathbf{X}^*)$ , the naïve elastic net solves a lasso-type problem

$$\hat{\beta}^* = \arg \min_{\beta^*} |\mathbf{y}^* - \mathbf{X}^* \beta^*|^2 + \frac{\lambda_1}{\sqrt{(1 + \lambda_2)}} |\beta^*|_1. \quad (10)$$

The elastic net (corrected) estimates  $\hat{\beta}$  are defined by

$$\hat{\beta}(\text{elastic net}) = \sqrt{(1 + \lambda_2)} \hat{\beta}^*. \quad (11)$$

Recall that  $\hat{\beta}(\text{naïve elastic net}) = \{1/\sqrt{(1 + \lambda_2)}\} \hat{\beta}^*$ ; thus

$$\hat{\beta}(\text{elastic net}) = (1 + \lambda_2) \hat{\beta}(\text{naïve elastic net}). \quad (12)$$

Hence the elastic net coefficient is a rescaled naïve elastic net coefficient.

Such a scaling transformation preserves the variable selection property of the naïve elastic net and is the simplest way to undo shrinkage. Hence all the good properties of the naïve elastic

# Computation of elastic net

- First solve the Naive Elastic Net problem, then rescale it.
- For fixed  $\lambda_2$ , the Naive Elastic Net problem is equivalent to a LASSO problem, with a huge data matrix if  $p \gg n$
- LASSO already has an efficient solver called LARS (Least Angle Regression).
- → LARS-EN algorithm.

# Elastic Net interpreted as a stabilized Lasso

*Theorem 2.* Given data  $(\mathbf{y}, \mathbf{X})$  and  $(\lambda_1, \lambda_2)$ , then the elastic net estimates  $\hat{\beta}$  are given by

$$\hat{\beta} = \arg \min_{\beta} \beta^T \left( \frac{\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{I}}{1 + \lambda_2} \right) \beta - 2\mathbf{y}^T \mathbf{X} \beta + \lambda_1 |\beta|_1. \quad (14)$$

It is easy to see that

$$\hat{\beta}(\text{lasso}) = \arg \min_{\beta} \beta^T (\mathbf{X}^T \mathbf{X}) \beta - 2\mathbf{y}^T \mathbf{X} \beta + \lambda_1 |\beta|_1. \quad (15)$$

Hence theorem 2 interprets the elastic net as a stabilized version of the lasso. Note that  $\hat{\Sigma} = \mathbf{X}^T \mathbf{X}$  is a sample version of the correlation matrix  $\Sigma$  and

$$\frac{\mathbf{X}^T \mathbf{X} + \lambda_2 \mathbf{I}}{1 + \lambda_2} = (1 - \gamma) \hat{\Sigma} + \gamma \mathbf{I}$$

with  $\gamma = \lambda_2 / (1 + \lambda_2)$  shrinks  $\hat{\Sigma}$  towards the identity matrix. Together equations (14) and (15) say that rescaling after the elastic net penalization is mathematically equivalent to replacing  $\hat{\Sigma}$  with its shrunken version in the lasso. In linear discriminant analysis, the prediction accuracy can often be improved by replacing  $\hat{\Sigma}$  by a shrunken estimate (Friedman, 1989; Hastie *et al.*, 2001). Likewise we improve the lasso by regularizing  $\hat{\Sigma}$  in equation (15).

# Extra Recap

- ❑ More about LR Model with Regularizations
  - ❑ Ridge Regression
  - ❑ Lasso Regression
    - ❑ Extra: how to perform training
  - ❑ Elastic net
    - ❑ Extra: how to perform training