Linear Regression

Statistical Machine Learning (ENGG*6600*08)

School of Engineering,
University of Guelph, ON, Canada

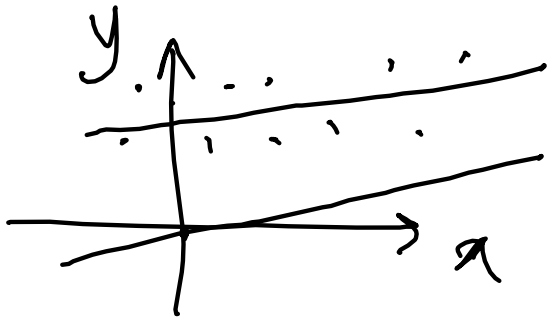
Course Instructor: Benyamin Ghojogh
Fall 2023

Linear Regression

Dataset

- Consider a dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$.
- We have some labels too: $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$ where $\boldsymbol{y}_i \in \mathbb{R}^p$. Usually $p = 1$ (but not always).
- The labels are not necessarily discrete but can be continuous.
- Example: data can be the data of weather temperature, longitude, latitude, etc, of the city. The label can be the pollution of the city.

Linear Regression



- We can consider a map $f(.)$ which maps data to the labels:

$$\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}, \tag{1}$$

  where $\boldsymbol{\epsilon} \in \mathbb{R}^p$ is noise.
- In regression, we want to estimate this map f.
- In linear regression, we want to estimate this map f by a line (or affine function).
- First, consider the case $p = 1$:

$$\mathbb{R} \ni f(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j \tag{2}$$

  where x_j is the j-th element of $\boldsymbol{x}$ and $\{\beta_j \in \mathbb{R}\}_{j=0}^{d}$ are the learnable parameters and $\beta_0 \in \mathbb{R}$ is specifically for learning the bias (intercept).

Linear Regression

- One way to do this estimation is to minimize the least squares error between the labels and the estimated model:

$$\underset{\{\beta_j\}_{j=0}^{d}}{\text{minimize}} \quad \sum_{i=1}^{n} \left(y_i - f(\boldsymbol{x}_i)\right)^2 \overset{(2)}{=} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij}\right)^2, \tag{3}$$

where x_{ij} is the j-th element of $\boldsymbol{x}_i$, i.e., $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{id}]^\top$.

- We can write it in matrix form. Let:

$$\mathbb{R}^{d+1} \ni \boldsymbol{\beta} := [\beta_0, \beta_1, \ldots, \beta_d]^\top, \qquad \mathbb{R}^{n\times(d+1)} \ni \boldsymbol{X} := \begin{bmatrix} 1, \boldsymbol{x}_1^\top \\ 1, \boldsymbol{x}_2^\top \\ \vdots \\ 1, \boldsymbol{x}_n^\top \end{bmatrix} = \begin{bmatrix} 1, x_{11}, \ldots, x_{1d} \\ 1, x_{21}, \ldots, x_{2d} \\ \vdots \\ 1, x_{n1}, \ldots, x_{nd} \end{bmatrix},$$

$$\mathbb{R}^{n} \ni \boldsymbol{y} := [y_1, \ldots, y_n]^\top.$$

The above optimization problem becomes:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2. \tag{4}$$

Linear Regression

- The cost function is simplified as:

$$\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2=(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})=(\boldsymbol{y}^\top-\boldsymbol{\beta}^\top\boldsymbol{X}^\top)(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$$
$$=\boldsymbol{y}^\top\boldsymbol{y}-\boldsymbol{y}^\top\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{\beta}^\top\boldsymbol{X}^\top\boldsymbol{y}+\boldsymbol{\beta}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta}.$$

- Taking derivative of the cost function and setting to zero:

$$\frac{\partial}{\partial\boldsymbol{\beta}}(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2)=-\boldsymbol{X}^\top\boldsymbol{y}-\boldsymbol{X}^\top\boldsymbol{y}+2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta}\overset{\text{set}}{=}\boldsymbol{0}$$
$$\implies 2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta}=2\boldsymbol{X}^\top\boldsymbol{y}$$
$$\implies \boldsymbol{\beta}=(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}. \tag{5}$$

- Test phase: The predicted labels for some data $\boldsymbol{X}$ is:

$$\boldsymbol{y}=\boldsymbol{X}\boldsymbol{\beta}=\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}. \tag{6}$$

Linear Regression

- In a more general form, if p is not necessarily one, we have:

$$\mathbb{R}^{(d+1)\times p} \ni \boldsymbol{B} := [\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_d]^\top, \qquad \mathbb{R}^{n\times p} \ni \boldsymbol{Y} := \begin{bmatrix} \boldsymbol{y}_1^\top \\ \boldsymbol{y}_2^\top \\ \vdots \\ \boldsymbol{y}_n^\top \end{bmatrix},$$

  where $\boldsymbol{\beta}_j \in \mathbb{R}^p$ and $\boldsymbol{y}_j \in \mathbb{R}^p$.

- In this case, the optimization problem becomes:

$$\underset{\boldsymbol{B}}{\text{minimize}} \quad \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2, \tag{7}$$

  where $\|.\|_F$ denotes the Frobenius norm.

Linear Regression

- The cost function is simplified as:

$$\begin{aligned}\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{B}\|_F^2 &= \mathbf{tr}\big((\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{B})^\top(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{B})\big) = \mathbf{tr}\big((\boldsymbol{Y}^\top-\boldsymbol{B}^\top\boldsymbol{X}^\top)(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{B})\big)\\ &= \mathbf{tr}(\boldsymbol{Y}^\top\boldsymbol{Y}) - \mathbf{tr}(\boldsymbol{Y}^\top\boldsymbol{X}\boldsymbol{B}) - \mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{X}^\top\boldsymbol{Y}) + \mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B}).\end{aligned}$$

- Taking derivative of the cost function and setting to zero:

$$\begin{aligned}&\frac{\partial}{\partial \boldsymbol{B}}(\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{B}\|_F^2) = -\boldsymbol{X}^\top\boldsymbol{Y} - \boldsymbol{X}^\top\boldsymbol{Y} + 2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B} \overset{\text{set}}{=} \mathbf{0}\\ &\Longrightarrow 2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B} = 2\boldsymbol{X}^\top\boldsymbol{Y}\\ &\Longrightarrow \widehat{\boldsymbol{B}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}\end{aligned} \tag{8}$$

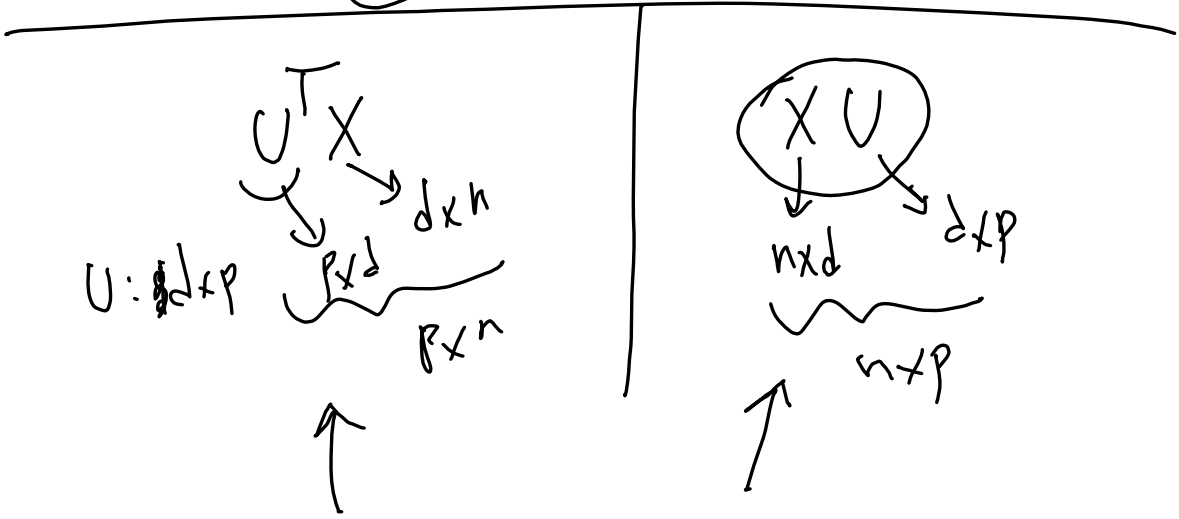
- Test phase: The predicted labels for some data $\boldsymbol{X}$ is:

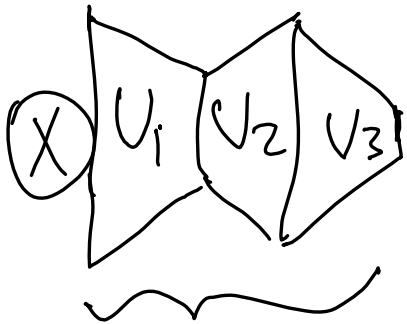
$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\widehat{\boldsymbol{B}} = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}. \tag{9}$$

$$B = (X^T X)^{-1} X^T Y$$

$$\hat{Y} = XB$$

XB: $n \times (d+1)$ $\cdot$ $(d+1) \times p$ $\longrightarrow$ $\hat{Y} \in \mathbb{R}^{n \times p}$

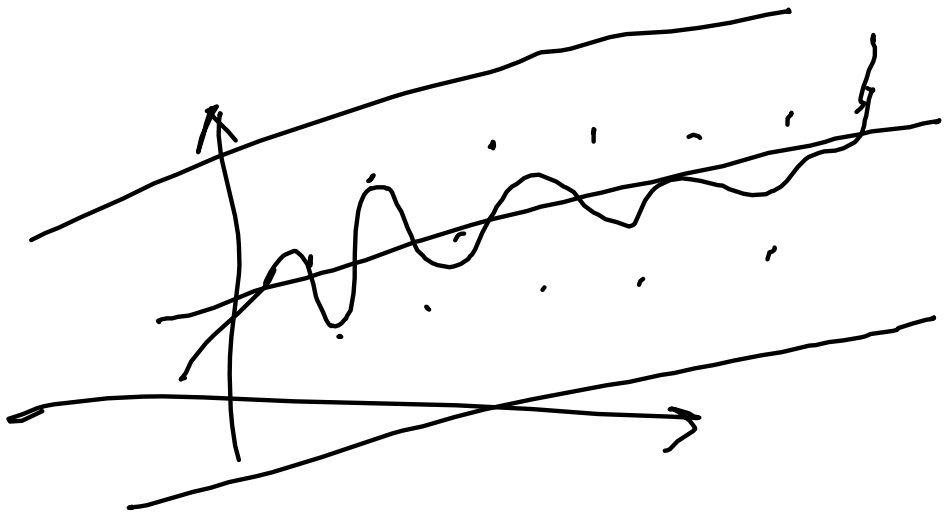




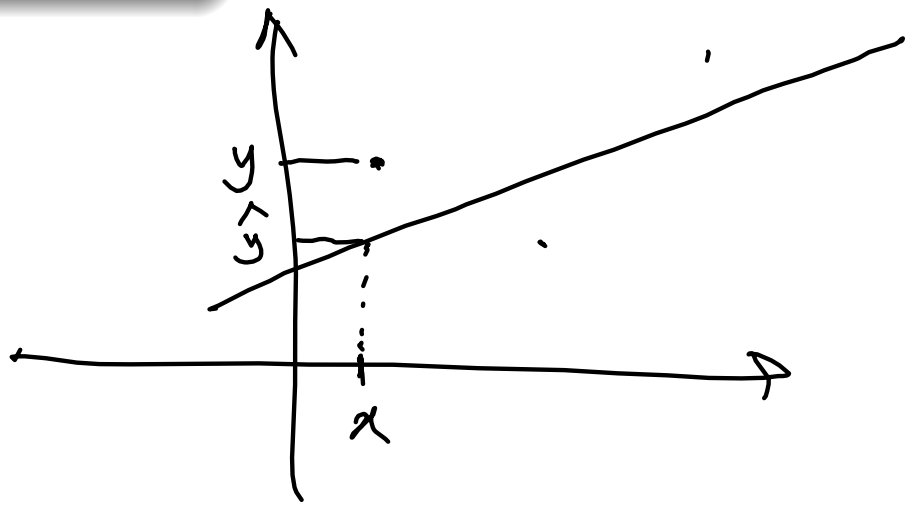
$\rightarrow$ loss $= \|\hat{Y} - Y\|_F^2$

$\hat{Y} = X \underbrace{U_1 U_2 U_3}_{U_4} = X U_4 = X\beta$

$X\, f_1(U_1)\, f_2(U_2)\, f_3(U_3)$



Ridge Linear Regression



Ridge Linear Regression

- We add ℓ_2 norm regularization term as a penalty on the size of the learnable parameters.
- Case $p = 1$:

$$\underset{\{\beta_j\}_{j=0}^{d}}{\textit{minimize}} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=0}^{d} \beta_j^2, \tag{10}$$

  where $\lambda > 0$ is the regularization parameter.
- We can write it in matrix form. The above optimization problem becomes:

$$\underset{\boldsymbol{\beta}}{\textit{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2. \tag{11}$$

Ridge Linear Regression

- The cost function is simplified as:

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 = \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{Y} + \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta}^\top \boldsymbol{\beta}.$$

- Taking derivative of the cost function and setting to zero:

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\beta}}(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2) = -\boldsymbol{X}^\top \boldsymbol{y} - \boldsymbol{X}^\top \boldsymbol{y} + 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta} \overset{\text{set}}{=} \boldsymbol{0} \\
&\implies 2\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta} = 2\boldsymbol{X}^\top \boldsymbol{y} \implies 2(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})\boldsymbol{\beta} = 2\boldsymbol{X}^\top \boldsymbol{y} \\
&\implies \boldsymbol{\beta} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^\top \boldsymbol{y}. \qquad (12)
\end{aligned}$$

- It is strengthening the main diagonal of $\boldsymbol{X}^\top \boldsymbol{X}$ so it makes $\boldsymbol{X}^\top \boldsymbol{X}$ full rank and non-singular. In other words, it makes $\boldsymbol{X}^\top \boldsymbol{X}$ invertible.
- Test phase: The predicted labels for some data $\boldsymbol{X}$ is:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^\top \boldsymbol{y}. \qquad (13)$$

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

Ridge Linear Regression

- More general case where p is not necessarily one:

$$\underset{\boldsymbol{B}}{\text{minimize}} \quad \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 + \lambda\|\boldsymbol{B}\|_F^2. \tag{14}$$

- The cost function is simplified as:

$$\begin{aligned}
&\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 + \lambda\|\boldsymbol{B}\|_F^2 = \mathbf{tr}\big((\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})\big) + \lambda\mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{B}) \\
&= \mathbf{tr}\big((\boldsymbol{Y}^\top - \boldsymbol{B}^\top\boldsymbol{X}^\top)(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B})\big) + \lambda\mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{B}) \\
&= \mathbf{tr}(\boldsymbol{Y}^\top\boldsymbol{Y}) - \mathbf{tr}(\boldsymbol{Y}^\top\boldsymbol{X}\boldsymbol{B}) - \mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{X}^\top\boldsymbol{Y}) + \mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B}) + \lambda\mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{B}).
\end{aligned}$$

- Taking derivative of the cost function and setting to zero:

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{B}}(\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}\|_F^2 + \lambda\mathbf{tr}(\boldsymbol{B}^\top\boldsymbol{B})) = -\boldsymbol{X}^\top\boldsymbol{Y} - \boldsymbol{X}^\top\boldsymbol{Y} + 2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B} + 2\lambda\boldsymbol{B} \overset{\text{set}}{=} \mathbf{0} \\
&\implies 2\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B} + 2\lambda\boldsymbol{B} = 2\boldsymbol{X}^\top\boldsymbol{Y} \implies 2(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})\boldsymbol{B} = 2\boldsymbol{X}^\top\boldsymbol{Y} \\
&\implies \boldsymbol{B} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}.
\end{aligned} \tag{15}$$

- Test phase: The predicted labels for some data $\boldsymbol{X}$ is:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}. \tag{16}$$

Lasso Linear Regression

ℓ_1 Norm Regularization

- As explained before, sparsity is very useful and effective. If $\boldsymbol{x} = [x_1, \ldots, x_d]^\top$, for having sparsity, we should use **subset selection** for the regularization:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \widetilde{J}(\boldsymbol{x}; \theta) := J(\boldsymbol{x}; \theta) + \alpha \, ||\boldsymbol{x}||_0, \tag{17}$$

  where:

$$||\boldsymbol{x}||_0 := \sum_{j=1}^{d} \mathbb{I}(x_j \neq 0) = \begin{cases} 0 & \text{if } x_j = 0, \\ 1 & \text{if } x_j \neq 0, \end{cases} \tag{18}$$

  is "ℓ_0" norm, which is not a norm (so we used "." for it) because it does not satisfy the norm properties [1]. The "ℓ_0" norm counts the number of non-zero elements so when we penalize it, it means that we want to have sparser solutions with many zero entries.
- According to [2], the convex relaxation of "ℓ_0" norm (subset selection) is ℓ_1 norm. Therefore, we write the regularized optimization as:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \widetilde{J}(\boldsymbol{x}; \theta) := J(\boldsymbol{x}; \theta) + \alpha \, ||\boldsymbol{x}||_1. \tag{19}$$

- The ℓ_1 regularization is also referred to as **lasso (least absolute shrinkage and selection operator)** regularization [3].

Lasso Linear Regression

- We add ℓ_1 norm regularization term as a penalty on the size of the learnable parameters.
- Case $p = 1$:

$$\underset{\{\beta_j\}_{j=0}^{d}}{\textit{minimize}} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=0}^{d} |\beta_j|, \tag{20}$$

where $\lambda > 0$ is the regularization parameter and $|.|$ denotes the absolute value function.

- We can write it in matrix form. The above optimization problem becomes:

$$\underset{\boldsymbol{\beta}}{\textit{minimize}} \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1. \tag{21}$$

- Let $\boldsymbol{x}'_k \in \mathbb{R}^n$ denote the k-th column of $\boldsymbol{X} \in \mathbb{R}^{n \times (d+1)}$, i.e., $\boldsymbol{X} = [\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{d+1}]$. The above cost function can be restated as:

$$\underset{\boldsymbol{\beta}=[\beta_1,\ldots,\beta_{d+1}]^\top}{\textit{minimize}} \quad \Big\|\boldsymbol{y} - \sum_{k=1}^{d+1} \beta_k \boldsymbol{x}'_k\Big\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1. \tag{22}$$

- This cost function can be further restated as below by taking out the j-th element from the summation:

$$\underset{\boldsymbol{\beta}=[\beta_1,\ldots,\beta_{d+1}]^\top}{\textit{minimize}} \quad \Big\|\boldsymbol{y} - \sum_{k=1,k\neq j}^{d+1} \beta_k \boldsymbol{x}'_k - \beta_j \boldsymbol{x}'_j\Big\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1. \tag{23}$$

$$\beta_1 x_1' + \beta_2 x_2' + \cdots + \beta_j x_j' + \cdots + \beta_{d+1} x_{d+1}'$$

$$= \left(\beta_1 x_1' + \cdots + \beta_{j-1} x_{j-1}' + \beta_{j+1} x_{j+1}' + \cdots + \beta_{d+1} x_{d+1}' \right)$$

$$+ \beta_j x_j'$$

Lasso Linear Regression

- We had:

$$\underset{\boldsymbol{\beta}=[\beta_1,\ldots,\beta_{d+1}]^\top}{\textit{minimize}} \quad \Big\|\boldsymbol{y} - \sum_{k=1,k\neq j}^{d+1} \beta_k \boldsymbol{x}'_k - \beta_j \boldsymbol{x}'_j\Big\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1.$$

- We define:

$$\mathbb{R}^n \ni \boldsymbol{z} := \boldsymbol{y} - \sum_{k=1,k\neq j}^{d+1} \beta_k \boldsymbol{x}'_k. \tag{24}$$

Therefore the cost function becomes:

$$\underset{\boldsymbol{\beta}=[\beta_1,\ldots,\beta_{d+1}]^\top}{\textit{minimize}} \quad \|\boldsymbol{z} - \beta_j \boldsymbol{x}'_j\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1.$$

- We use coordinate descent for optimization. Considering the j-th element of $\boldsymbol{\beta}$, we have:

$$\underset{\beta_j}{\textit{minimize}} \quad \|\boldsymbol{z} - \beta_j \boldsymbol{x}'_j\|_2^2 + \lambda|\beta_j|,$$

where $\boldsymbol{z}$ is a constant with respect to β_j.

Lasso Linear Regression

$a^T b = b^T a$

- We had:

$$\not\star \quad \underset{\beta_j}{\textit{minimize}} \quad \|\boldsymbol{z} - \beta_j \boldsymbol{x}'_j\|_2^2 + \lambda|\beta_j|.$$

- The cost is simplified as:

$$\|\boldsymbol{z} - \beta_j \boldsymbol{x}'_j\|_2^2 + \lambda|\beta_j| = (\boldsymbol{z} - \beta_j \boldsymbol{x}'_j)^\top (\boldsymbol{z} - \beta_j \boldsymbol{x}'_j) + \lambda|\beta_j|$$
$$= (\boldsymbol{z}^\top - \beta_j \boldsymbol{x}_j'^\top)(\boldsymbol{z} - \beta_j \boldsymbol{x}'_j) + \lambda|\beta_j| = \boldsymbol{z}^\top \boldsymbol{z} - \beta_j \boldsymbol{z}^\top \boldsymbol{x}'_j - \beta_j \boldsymbol{x}_j'^\top \boldsymbol{z} + \beta_j^2 \boldsymbol{x}_j'^\top \boldsymbol{x}'_j + \lambda|\beta_j|,$$

where it is noticed in calculations that β_j is a scalar.

- Taking derivative of the cost function with respect to β_j and setting to zero:

$$\frac{\partial}{\partial \beta_j}(\|\boldsymbol{z} - \beta_j \boldsymbol{x}'_j\|_2^2 + \lambda|\beta_j|) = -\boldsymbol{z}^\top \boldsymbol{x}'_j - \boldsymbol{x}_j'^\top \boldsymbol{z} + 2\beta_j \boldsymbol{x}_j'^\top \boldsymbol{x}'_j + \lambda\, \mathbf{sign}(\beta_j)$$
$$= -\boldsymbol{x}_j'^\top \boldsymbol{z} - \boldsymbol{x}_j'^\top \boldsymbol{z} + 2\beta_j \boldsymbol{x}_j'^\top \boldsymbol{x}'_j + \lambda\, \mathbf{sign}(\beta_j) = -2\boldsymbol{x}_j'^\top \boldsymbol{z} + 2\beta_j \boldsymbol{x}_j'^\top \boldsymbol{x}'_j + \lambda\, \mathbf{sign}(\beta_j) \overset{\text{set}}{=} 0$$

$$\implies \beta_j = \begin{cases} \frac{\boldsymbol{x}_j'^\top \boldsymbol{z}}{\boldsymbol{x}_j'^\top \boldsymbol{x}'_j} - \frac{\lambda}{2} & \text{if } \beta_j \geq 0 \\ \frac{\boldsymbol{x}_j'^\top \boldsymbol{z}}{\boldsymbol{x}_j'^\top \boldsymbol{x}'_j} + \frac{\lambda}{2} & \text{if } \beta_j < 0. \end{cases}$$

Lasso Linear Regression

- We found:

$$\beta_j = \begin{cases} \frac{\boldsymbol{x}_j'^{\top}\boldsymbol{z}}{\boldsymbol{x}_j'^{\top}\boldsymbol{x}_j'} - \frac{\lambda}{2} & \text{if } \beta_j \geq 0 \\ \frac{\boldsymbol{x}_j'^{\top}\boldsymbol{z}}{\boldsymbol{x}_j'^{\top}\boldsymbol{x}_j'} + \frac{\lambda}{2} & \text{if } \beta_j < 0. \end{cases} \tag{25}$$

- If the cost is $(1/2)\|\boldsymbol{z} - \beta_j \boldsymbol{x}_j'\|_2^2 + \lambda|\beta_j|$, then the $(1/2)$ multipliers of λ will go away (if you put $(1/2)$ multiplied by the norm and calculate the derivative as was done, you will see why):

$$\beta_j = \begin{cases} \frac{\boldsymbol{x}_j'^{\top}\boldsymbol{z}}{\boldsymbol{x}_j'^{\top}\boldsymbol{x}_j'} - \lambda & \text{if } \beta_j \geq 0 \\ \frac{\boldsymbol{x}_j'^{\top}\boldsymbol{z}}{\boldsymbol{x}_j'^{\top}\boldsymbol{x}_j'} + \lambda & \text{if } \beta_j < 0. \end{cases} \tag{26}$$

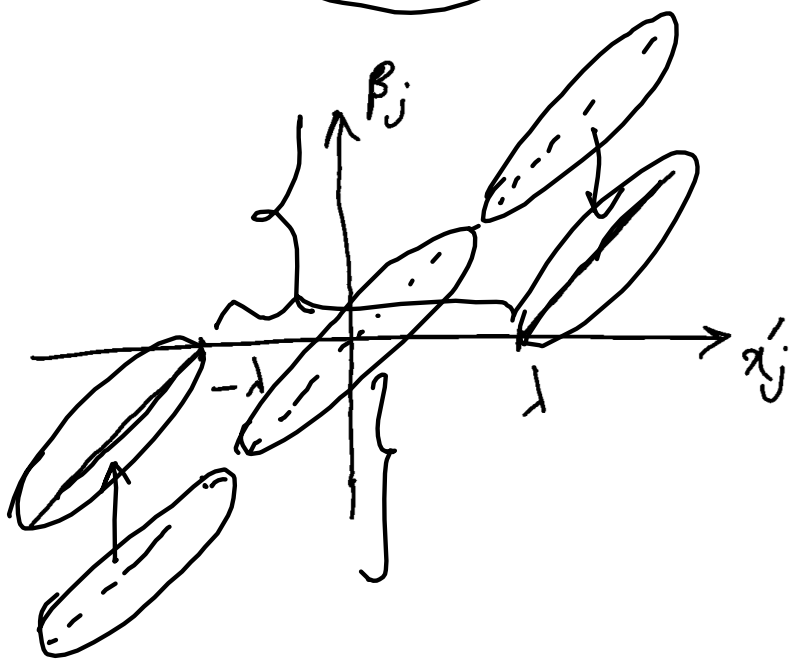
- If the columns of matrix $\boldsymbol{X}$ are normalized to have unit length, then $\boldsymbol{x}_j'^{\top}\boldsymbol{x}_j' = \|\boldsymbol{x}_j'\|_2^2 = 1$ and it would simplify the solution further to:

$$\beta_j = \begin{cases} \boldsymbol{x}_j'^{\top}\boldsymbol{z} - \lambda & \text{if } \beta_j \geq 0 \\ \boldsymbol{x}_j'^{\top}\boldsymbol{z} + \lambda & \text{if } \beta_j < 0. \end{cases} \tag{27}$$

Lasso Linear Regression

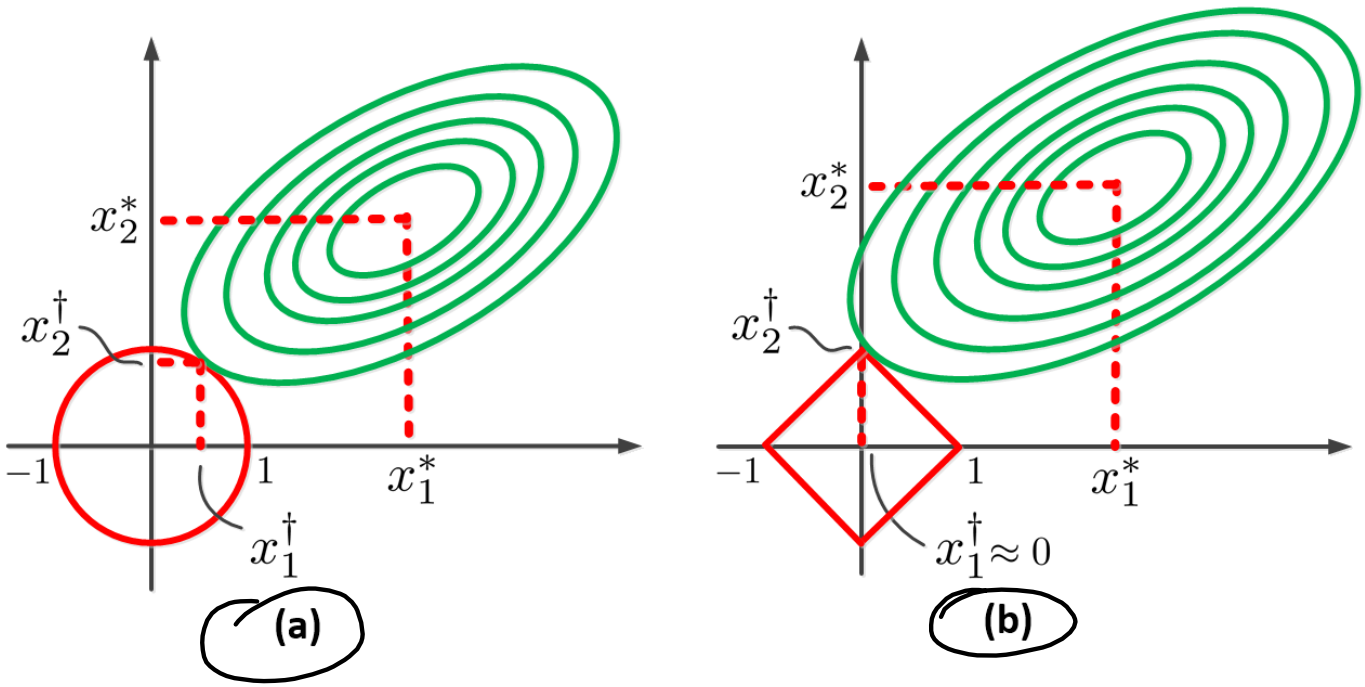
- This is a soft-thresholding function:

$$\beta_j = \begin{cases} \boldsymbol{x}_j'^{\top} \boldsymbol{z} - \lambda & \text{if } \beta_j \geq 0 \\ \boldsymbol{x}_j'^{\top} \boldsymbol{z} + \lambda & \text{if } \beta_j < 0. \end{cases}$$



Comparison of ℓ_2 and ℓ_1 Regularization

- An intuition for why the ℓ_1 norm regularization is sparse is illustrated below (credit if Tibshirani - 1996 [3]).
- The objective $J(\boldsymbol{x}; \theta)$ has some contour levels like a bowl (if it is convex). The regularization term is also a norm ball, which is a sphere bowl (cone) for ℓ_2 norm and a diamond bowl (cone) for ℓ_1 norm [1].
- For ℓ_2 norm regularization, the objective and the penalty term contact at a point where some of the coordinates might be small; however, for ℓ_1 norm, the contact point can be at some point where some variables are exactly zero. This again shows the reason of sparsity in ℓ_1 norm regularization.



Acknowledgment

- For more information on linear regression, see the book: Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, Jerome H. Friedman. "The elements of statistical learning: data mining, inference, and prediction". Vol. 2. New York: springer, 2009 [4].
- Another textbook suitable for sparsity in machine learning is: Robert Tibshirani, Martin Wainwright, Trevor Hastie, "Statistical learning with sparsity: the lasso and generalizations", Chapman and Hall/CRC, 2015 [5].
- Some slides of this slide deck are inspired by teachings of Prof. Mu Zhu at University of Waterloo, Department of Statistics and Prof. Hoda Mohammadzade at Sharif University of Technology, Department of Electrical Engineering.

References

[1] S. Boyd and L. Vandenberghe, *Convex optimization*.
Cambridge university press, 2004.

[2] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 6, pp. 797–829, 2006.

[3] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[4] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2.
Springer, 2009.

[5] R. Tibshirani, M. Wainwright, and T. Hastie, *Statistical learning with sparsity: the lasso and generalizations*.
Chapman and Hall/CRC, 2015.