Element of Statistical Learning Note

Zichong Wang

December 31, 2025

Contents

1 Chap 3: Linear Methods for Regression

1.1 Confidence region for $\hat{\beta}$

We know that $\hat{\beta} \sim \mathcal{N}(\beta, \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1})$, thus $\hat{\beta} - \beta \sim \mathcal{N}(0, \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1})$.

Since for $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, we have $\boldsymbol{z}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{z} \sim \chi^2_k$, where $k = \text{rank}(\boldsymbol{\Sigma})$, we have

$$(\hat{\beta} - \beta)^T(\sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1})^{-1}(\hat{\beta} - \beta) \sim \chi^2_{p+1}$$

Thus,

$$\frac{1}{\sigma^2}(\hat{\beta} - \beta)^T\boldsymbol{X}^T\boldsymbol{X}(\hat{\beta} - \beta) \sim \chi^2_{p+1}$$

And have approx confidence region

$$(\hat{\beta} - \beta)^T\boldsymbol{X}^T\boldsymbol{X}(\hat{\beta} - \beta) \leq \hat{\sigma}^2\chi^2_{p+1,1-\alpha}$$

Actually, $\hat{\sigma}^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N-p-1}\text{RSS}$, and $\text{RSS}/\sigma^2 \sim \chi^2_{N-p-1}$,

$$\frac{(\hat{\beta} - \beta)^T\boldsymbol{X}^T\boldsymbol{X}(\hat{\beta} - \beta)/(p+1)}{\hat{\sigma}^2} \sim F_{p+1,N-p-1}$$

1.2 What is Linear

In the context of **linear model**, we are talking about linearity in parameters, meaning that the prediction $\hat{y}$ is a linear combination of the parameters β_j. The $\boldsymbol{X}$ itself can be non-linear transformations of the original features, e.g., polynomial terms, interaction terms, etc. $y = 1/(\beta_0 + \beta_1 x)$ and $y = \beta_0 e^{\beta_1 x}$ are not linear models, since they're not linear in parameters.

In the context of **Linear estimators**, we are talking about the estimator $\hat{\theta}$ (e.g. $\hat{\beta}$) can be written as a linear combination of the observed response values y_i, i.e. $\hat{\theta} = \boldsymbol{c}^T\boldsymbol{y}$. The weight $\boldsymbol{c}$ depends only on $\boldsymbol{X}$, not on $\boldsymbol{y}$. A linear estimator **can be** a prediction at a new point, or the estimated coefficients $\hat{\beta}$ themselves.

1.3 Gauss-Markov Theorem

Why assume only know $\boldsymbol{X}$, but not $\boldsymbol{y}$?

> Note that though y_i as sample responses, are observable, the following statements and arguments including assumptions, proofs and the others assume under the only condition of knowing $\boldsymbol{X}_{i,j}$ but not y_i. — [1]

We have a *challenger* linear estimator $\tilde{\beta} = \boldsymbol{C}\boldsymbol{y}$, where $\boldsymbol{C} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T + \boldsymbol{D}$, a modification

of OLS estimator. Ensure it's unbiased:

$$\begin{aligned}\mathbb{E}(\tilde{\beta}) &= \mathbb{E}(\boldsymbol{C}\boldsymbol{y}) \\ &= \boldsymbol{C}\mathbb{E}(\boldsymbol{y}) = ((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T + \boldsymbol{D})\boldsymbol{X}\beta \\ &= \beta + \boldsymbol{D}\boldsymbol{X}\beta \\ &= \beta\end{aligned}$$

Meaning $\boldsymbol{D}\boldsymbol{X} = 0$.

Now, compute the variance:

$$\begin{aligned}\mathrm{Var}(\tilde{\beta}) &= \mathrm{Var}(\boldsymbol{C}\boldsymbol{y}) \\ &= \boldsymbol{C}\mathrm{Var}(\boldsymbol{y})\boldsymbol{C}^T \\ &= \sigma^2\boldsymbol{C}\boldsymbol{C}^T \\ &= \sigma^2((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T + \boldsymbol{D})((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T + \boldsymbol{D})^T \\ &= \sigma^2((\boldsymbol{X}^T\boldsymbol{X})^{-1} + \boldsymbol{D}\boldsymbol{D}^T) \\ &= \mathrm{Var}(\hat{\beta}) + \sigma^2\boldsymbol{D}\boldsymbol{D}^T \\ &\geq \mathrm{Var}(\hat{\beta})\end{aligned}$$

1.4 QR decomposition

Any real squared matrix $\boldsymbol{A}$ can be decomposed as $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q}$ is orthogonal matrix ($\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$), and $\boldsymbol{R}$ is upper triangular matrix.

Any rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ ($m \geq n$), we can decompose it as $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q}$ is $m \times m$ orthogonal matrix, and $\boldsymbol{R}$ is $m \times n$ upper triangular matrix (the last $m - n$ rows are all zero). It can be regarded as $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R} = [\boldsymbol{Q}_1 \quad \boldsymbol{Q}_2]\begin{bmatrix}\boldsymbol{R}_1 \\ \boldsymbol{0}\end{bmatrix} = \boldsymbol{Q}_1\boldsymbol{R}_1$, where $\boldsymbol{Q}_1 \in \mathbb{R}^{m\times n}, \boldsymbol{Q}_2 \in \mathbb{R}^{m\times(m-n)}, \boldsymbol{R}_1 \in \mathbb{R}^{n\times n}$ which is upper triangular.

If $\boldsymbol{A}$ have k linearly independent columns, then first k columns of $\boldsymbol{Q}$ form an orthonormal basis of the column space of $\boldsymbol{A}$. The fact that any column k of $\boldsymbol{A}$ only depends on the first k columns of $\boldsymbol{Q}$ corresponds to the triangular form of $\boldsymbol{R}$.

QR decomposition can be calculated using Gram-Schmidt process, or using Householder reflections. In practice, Householder reflections are more stable and efficient.

1.4.1 Application to Least Squared

In linear least squares problems, we aim to find a vector $\boldsymbol{x}$ that minimizes the Euclidean norm of the residual for an overdetermined system $\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $m \geq n$. The goal is to solve:

$$\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$$

Using the Full QR Decomposition, we substitute $\boldsymbol{A} = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}_1 \\ \boldsymbol{0}\end{bmatrix}$:

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \left\|\boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}_1 \\ \boldsymbol{0}\end{bmatrix}\boldsymbol{x} - \boldsymbol{b}\right\|_2$$

Since multiplying by an orthogonal matrix $\boldsymbol{Q}$ preserves the Euclidean norm, we can left-multiply the entire expression by $\boldsymbol{Q}^T$:

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \left\| \begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{x} - \boldsymbol{Q}^T \boldsymbol{b} \right\|_2$$

If we partition $\boldsymbol{Q}^T\boldsymbol{b}$ into two components—$\boldsymbol{c}_1 \in \mathbb{R}^n$ and $\boldsymbol{c}_2 \in \mathbb{R}^{m-n}$:

$$\left\| \begin{bmatrix} \boldsymbol{R}_1\boldsymbol{x} - \boldsymbol{c}_1 \\ -\boldsymbol{c}_2 \end{bmatrix} \right\|_2^2 = \|\boldsymbol{R}_1\boldsymbol{x} - \boldsymbol{c}_1\|_2^2 + \|\boldsymbol{c}_2\|_2^2$$

To minimize the total error, we must make the first term zero. The least squares solution is found by solving the square, upper-triangular system:

$$\boldsymbol{R}_1\boldsymbol{x} = \boldsymbol{c}_1$$

Since $\boldsymbol{R}_1$ is upper triangular, we can efficiently solve this system using back substitution, **that's how we solve linear equations manually in algebra class**! The remaining term $\|\boldsymbol{c}_2\|_2$ represents the minimum residual norm (the "error" of the fit).

1.4.2 About orthogonal matrix

Let $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ be an orthogonal matrix. Write $\boldsymbol{Q}$ in terms of its column vectors:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_n \end{bmatrix}, \qquad \boldsymbol{q}_i^\top \boldsymbol{q}_j = \delta_{ij}.$$

Then

$$\boldsymbol{Q}^\top = \begin{bmatrix} \boldsymbol{q}_1^\top \\ \boldsymbol{q}_2^\top \\ \vdots \\ \boldsymbol{q}_n^\top \end{bmatrix}.$$

Computation of $\boldsymbol{Q}^\top\boldsymbol{Q}$.

$$\boldsymbol{Q}^\top\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1^\top \\ \boldsymbol{q}_2^\top \\ \vdots \\ \boldsymbol{q}_n^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1^\top\boldsymbol{q}_1 & \boldsymbol{q}_1^\top\boldsymbol{q}_2 & \cdots & \boldsymbol{q}_1^\top\boldsymbol{q}_n \\ \boldsymbol{q}_2^\top\boldsymbol{q}_1 & \boldsymbol{q}_2^\top\boldsymbol{q}_2 & \cdots & \boldsymbol{q}_2^\top\boldsymbol{q}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{q}_n^\top\boldsymbol{q}_1 & \boldsymbol{q}_n^\top\boldsymbol{q}_2 & \cdots & \boldsymbol{q}_n^\top\boldsymbol{q}_n \end{bmatrix}.$$

Using $\boldsymbol{q}_i^\top\boldsymbol{q}_j = \delta_{ij}$,

$$\boldsymbol{Q}^\top\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \boldsymbol{I}.$$

Computation of $\boldsymbol{Q}\boldsymbol{Q}^\top$.

$$\boldsymbol{Q}\boldsymbol{Q}^\top = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_1^\top \\ \boldsymbol{q}_2^\top \\ \vdots \\ \boldsymbol{q}_n^\top \end{bmatrix} = \boldsymbol{q}_1\boldsymbol{q}_1^\top + \boldsymbol{q}_2\boldsymbol{q}_2^\top + \cdots + \boldsymbol{q}_n\boldsymbol{q}_n^\top = \sum_{k=1}^{n} \boldsymbol{q}_k\boldsymbol{q}_k^\top.$$

For any $\boldsymbol{x} \in \mathbb{R}^n$,

$$(\boldsymbol{Q}\boldsymbol{Q}^\top)\boldsymbol{x} = \sum_{k=1}^{n} \boldsymbol{q}_k(\boldsymbol{q}_k^\top \boldsymbol{x}) = \boldsymbol{x},$$

since $\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$ is an orthonormal basis of $\mathbb{R}^n$. Hence

$$\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}.$$

Therefore, for an orthogonal matrix $\boldsymbol{Q}$,

$$\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}.$$

1.4.3 SVD and orthogonal matrix

There are two kinds of SVD: full SVD and reduced SVD. Let $\boldsymbol{X} \in \mathbb{R}^{N\times p}$, and nomally $N > p$.

- **Full SVD** in standard linear algebra.

$$\boldsymbol{X} = \boldsymbol{U}_{\text{full}}\boldsymbol{D}_{\text{full}}\boldsymbol{V}^T$$

  - $\boldsymbol{U}_{\text{full}} \in \mathbb{R}^{N\times N}$ is orthogonal matrix, whose columns are eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$.
  - $\boldsymbol{D}_{\text{full}} \in \mathbb{R}^{N\times p}$ is a rectangular diagonal matrix, and the last $N-p$ rows are all zero.
  - $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ is orthogonal matrix, whose columns are eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$, and are basis of row space of $\boldsymbol{X}$.

- **Reduced SVD** in linear regression. We simply ignore the zero parts of $\boldsymbol{U}$ and $\boldsymbol{D}$.

  $\boldsymbol{D}$ and $\boldsymbol{U}$ can be seen as

$$\boldsymbol{D}_{\text{full}} = \begin{bmatrix} d_1 & 0 & \ldots \\ 0 & \ddots & \\ \ldots & & d_p \\ \hline 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} p \text{ (Non-zero part)} \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} N-p \text{ (All zero part)} \end{matrix}$$

$$\mathbf{U}_{\text{full}} = \begin{bmatrix} \mathbf{U}_1 & | & \mathbf{U}_2 \end{bmatrix}$$

  Thus,

$$\begin{aligned} \mathbf{U}_{full} \cdot \mathbf{D}_{full} &= \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{D}_p \\ \mathbf{0} \end{bmatrix} \\ &= \mathbf{U}_1 \cdot \mathbf{D}_p + \mathbf{U}_2 \cdot \mathbf{0} \\ &= \mathbf{U}_1 \cdot \mathbf{D}_p \end{aligned}$$

  That's the reduced SVD:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$$

  where $\boldsymbol{U} \in \mathbb{R}^{N\times p}$ with orthonormal columns, $\boldsymbol{D} \in \mathbb{R}^{p\times p}$ diagonal with positive entries, and $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ orthogonal.

In regression, we usually use reduced SVD, and $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{H} \neq \boldsymbol{I}$, but $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$. For $\boldsymbol{y} \in \mathbb{R}^N$, $\boldsymbol{U}^T\boldsymbol{y}$ map $\boldsymbol{y}$ from $\mathbb{R}^N$ to $\mathbb{R}^p$, which is the coefficients in the basis of columns of $\boldsymbol{U}$. And $\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{y}$ project $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}$.

1.5 Multiple testing in forward selection

ESL page 60:

> Other more traditional packages base the selection on F -statistics, adding "significant" terms, and dropping "non-significant" terms. These are out of fashion, since they do not take proper account of the multiple testing issues.

Assume we have p candidate features, and already selected k features. When considering adding a new feature, we are actually performing $p-k$ hypothesis tests (each test for one feature). Even the rest $p-k$ features are all noise, with significance level α, we still have a probability of $1-(1-\alpha)^{p-k}$ to incorrectly add at least one noise feature.

1.6 Ridge regression

Answer questions: why two forms are equivalent? Why not equivariant under scaling of the inputs? What is a good practice for it? df of ridge? In the case of orthogonal inputs, why $\hat{\beta}^{\text{ridge}} = \hat{\beta}/(1+\lambda)$?

Ridge regression shrinks the regression coefficients by imposing a penalty on their size:

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

and is equivalent to

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \le t$$

And there is a one-to-one correspondence between λ and t.

1.6.1 How to use ridge regression

Apparently, ridge regression is not equivariant under scaling of the inputs. For example, using OLS, measuring x in meters or in centimeters will not change the predictions, since the latter coefficients will just 100 times of the first. However, in ridge regression, the penalty term $\lambda \sum_{j=1}^{p} \beta_j^2$ will be affected by the scale of x_j. Which means, using centimeters instead of meters will make the penalty on β_j 10000 times larger, leading to different solutions. Thus, it's important to standardize the features (zero mean and unit variance) before applying ridge.

Usually, we calculate μ and σ from training set, and use them to standardize both training and test sets. Scaler can be regarded as part of the model, not data cleaning!

1.6.2 Equivalence of two forms

First, take a review of **KKT conditions**.

Condider a minimization problem with both equality and inequality constraints:

$$\begin{aligned} &\min_{\boldsymbol{x}} f(\boldsymbol{x}) \\ \text{subject to } &g_i(\boldsymbol{x}) \leq 0, && i = 1, \ldots, m \\ &h_j(\boldsymbol{x}) = 0, && j = 1, \ldots, l \end{aligned}$$

And the lagrangian function is:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \sum_{j=1}^{l} \mu_j h_j(\boldsymbol{x})$$

If $\boldsymbol{x}^*$ is a local minimum, then there exist multipliers $\lambda_i^* \geq 0$ and μ_j^* such that the following conditions hold:

- **Stationary**
$$\nabla_x L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = 0$$
- **Primal feasibility**
$$g_i(\boldsymbol{x}^*) \leq 0, \quad i = 1, \ldots, m$$
$$h_j(\boldsymbol{x}^*) = 0, \quad j = 1, \ldots, l$$
The gradient can be regarded as the force to push a particle, primal stationary means the force of $\partial f(\boldsymbol{x}^*)$ is balanced by a linear sum of forces from constraints.
- **Dual feasibility**
$$\lambda_i^* \geq 0, \quad i = 1, \ldots, m$$
All the $\partial g_i(\boldsymbol{x}^*)$ forces must be one-sided, pointing inwards into the feasible set for $\boldsymbol{x}$.
- **Complementary slackness**
$$\lambda_i^* g_i(\boldsymbol{x}^*) = 0, \quad i = 1, \ldots, m$$
The force only activated when the particle is on the boundary of feasible set.

In ridge regression, we have:

$$\mathcal{L} = \|\boldsymbol{Y} - \boldsymbol{X}\beta\|_2^2 + \alpha(\|\beta\|_2^2 - t)$$

According to stationary condition:

$$\nabla_\beta \mathcal{L} = -2\boldsymbol{X}^T(\boldsymbol{Y} - \boldsymbol{X}\beta) + 2\alpha\beta = 0$$

On the other hand, solving the unconstrained form is

$$\nabla_\beta \left(\|\boldsymbol{Y} - \boldsymbol{X}\beta\|_2^2 + \lambda\|\beta\|_2^2 \right) = -2\boldsymbol{X}^T(\boldsymbol{Y} - \boldsymbol{X}\beta) + 2\lambda\beta = 0$$

Thus, if we set $\lambda = \alpha$, the two forms are equivalent.

One step further, according to complementary slackness:

$$\alpha(\|\beta\|_2^2 - t) = 0$$

From unconstrained form, given λ, we can solve β, denote $\beta(\lambda)$, and define $t(\lambda) = \|\beta(\lambda)\|_2^2$. Then, $\beta(\lambda)$ is also the solution of constrained form with $t = t(\lambda)$ (apparently it's on the boundary, and the coefficient $\alpha = \lambda$).

From the constrained form, given t, we can also solve β, denote $\beta(t)$.

- If $\|\beta(t)\|_2^2 < t$, then according to complementary slackness, $\alpha = 0$, which means no penalty, $\lambda = 0$, back to OLS.
- If $\|\beta(t)\|_2^2 = t$, the boundary is effective, correspond to some $\alpha > 0$, and $\lambda = \alpha$.

1.6.3 Bayesian view

Assume prior $\boldsymbol{\beta} \sim \mathcal{N}(0, \tau^2 \boldsymbol{I})$, and likelihood $\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{X}\beta, \sigma^2 \boldsymbol{I})$. Then the posterior is:

$$\begin{aligned} p(\beta|\boldsymbol{Y}) &\propto p(\boldsymbol{Y}|\beta)p(\beta) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{Y} - \boldsymbol{X}\beta\|_2^2\right) \exp\left(-\frac{1}{2\tau^2}\|\beta\|_2^2\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{1}{\sigma^2}\|\boldsymbol{Y} - \boldsymbol{X}\beta\|_2^2 + \frac{1}{\tau^2}\|\beta\|_2^2\right)\right) \end{aligned}$$

Maximizing the posterior is equivalent to minimizing the negative log-posterior. The object is equivalent to ridge regression with $\lambda = \sigma^2/\tau^2$.

However, it seems that from bayesian view, we don't have a hard constrain on the size of β, just a prior that β is likely to be small. But the constrained form of ridge regression directly enforce $\|\beta\|_2^2 \leq t$. Is it contradictory?

No. Actually, the posterior distribution of β still have infinite support, meaning that β can still be large with small probability. Our MAP estimate of β is the mode of the posterior, it's just a **point estimate**. This point estimator can still satisfy the hard constraint $\|\beta\|_2^2 \leq t$ for some t.

1.6.4 SVD in ridge

The solution of ridge regression is:

$$\hat{\beta}^{\text{ridge}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

When $\boldsymbol{X}^T\boldsymbol{X}$ is singular (not full rank), OLS estimator is not defined, and adding $\lambda\boldsymbol{I}$ ensures that $\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}$ is positive definite and invertible.

Take reduced SVD of $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, $r = \text{Rank}(\boldsymbol{X})$, where $\boldsymbol{U} \in \mathbb{R}^{N\times r}, \boldsymbol{D} \in \mathbb{R}^{r\times r}, \boldsymbol{V} \in \mathbb{R}^{p\times r}$. Then,

$$\begin{aligned} \boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I} &= \boldsymbol{V}\boldsymbol{D}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T + \lambda\boldsymbol{I} \\ &= \boldsymbol{V}\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{V}^T + \lambda\boldsymbol{V}\boldsymbol{V}^T \\ &= \boldsymbol{V}(\boldsymbol{D}^2 + \lambda\boldsymbol{I})\boldsymbol{V}^T \end{aligned}$$

must be full rank and invertible.

It's also easy to see that in the case of orthogonal inputs (i.e., $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}$), the ridge estimates are just a scaled version of the least squares estimates, that is

$$\hat{\beta}^{\text{ridge}} = \frac{1}{1+\lambda}\hat{\beta}$$

Further more,

$$\begin{aligned} \boldsymbol{X}\beta^{\text{OLS}} &= \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T(\boldsymbol{V}\boldsymbol{D}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T)^{-1}\boldsymbol{V}\boldsymbol{D}^T\boldsymbol{U}^T\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{D}\boldsymbol{D}^{-2}\boldsymbol{D}^T\boldsymbol{U}^T\boldsymbol{Y} \\ &= \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{Y} \end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{X}\beta^{\text{ridge}} &= \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \\
&= \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T(\boldsymbol{V}(\boldsymbol{D}^2 + \lambda\boldsymbol{I})\boldsymbol{V}^T)^{-1}\boldsymbol{V}\boldsymbol{D}^T\boldsymbol{U}^T\boldsymbol{Y} \\
&= \boldsymbol{U}\boldsymbol{D}(\boldsymbol{D}^2 + \lambda\boldsymbol{I})^{-1}\boldsymbol{D}^T\boldsymbol{U}^T\boldsymbol{Y} \\
&= \boldsymbol{U}\text{diag}\left(\frac{d_1^2}{d_1^2+\lambda}, \frac{d_2^2}{d_2^2+\lambda}, \ldots, \frac{d_r^2}{d_r^2+\lambda}\right)\boldsymbol{U}^T\boldsymbol{Y}
\end{aligned}$$

In both way, $\boldsymbol{U}^T\boldsymbol{Y}$ gets the coordinates of $\boldsymbol{Y}$ in the basis of columns of $\boldsymbol{U}$ (the principal components of $\boldsymbol{X}$), and then send back to original space using $\boldsymbol{U}$. But ridge regression shrink the coordinates.

References

[1] Wikipedia contributors, "Gauss–Markov theorem," *Wikipedia, The Free Encyclopedia*, https://en.wikipedia.org/wiki/Gauss%E2%80%93Markov_theorem (accessed Dec 29, 2025).