LINEAR REGRESSION

Chao Lan

1. Linear Regression Model

For regression task, a basic prediction model f is linear regression. It has the form

$$f(x) = \beta_0 + \beta_1 x._1 + \beta_2 x._2 + \ldots + \beta_p x.p, \tag{1}$$

where x is an instance described by p features $x._1, \ldots, x._p$, and $\beta_1, \ldots, \beta_p$ are regression coefficients and β_0 is bias.

[Discussion] Why is it called 'linear'?

[Discussion] Geometric interpretation of linear regression. (prediction, bias, slope)

We often write $f(x)$ in a matrix form. Let $x = \begin{bmatrix} 1 \\ x._1 \\ \vdots \\ x._p \end{bmatrix}$ and $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$ be two vectors, then

$$f(x) = \beta_0 \cdot 1 + \beta_1 x._1 + \beta_2 x._2 + \ldots + \beta_p x.p = [x._0, x._1, \ldots, x._p] \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} = x^T \beta. \tag{2}$$

The parameter of f is β, which we can learn from a set of n training instances: $(x_1, y_1), \ldots, (x_n, y_n)$.

2. Overview of Three Learning Methods

There are three common learning methods: least square, ridge regression and Lasso. They all aim to minimize training error, measured as mean-squared error on the training set

$$\hat{L}(f) = \sum_{i=1}^{n} (f(x_i) - y_i)^2 = \sum_{i=1}^{n} (x_i^T \beta - y_i)^2. \tag{3}$$

LS is okay with arbitrary models, whereas ridge and Lasso both favor simpler models but in different ways – ridge prefers models with smaller parameter domains, and Lasso prefers models constructed with fewer parameters.

[Discussion] Overview diagram of the three learning methods.

[Discussion] What are the pros and cons of ridge and Lasso, compared with LS?

[Discussion] Why work with training error while what we really want is to minimize the generalization error over the population, i.e., $L(f) = \mathrm{E}(f(x) - y)^2$? Is there any relation between the two errors?

[Discussion] What is the relation between generalization error, model bias and model variance?

Generalization error admits a bias-variance decomposition. Let (x, y) be a fixed instance but observed as a noisy instance (x, y') where $y' = y + \epsilon$ with noise ϵ satisfying $E[\epsilon] = 0$. Let f be a random model (as it is learned from a random training set). Assume $f(x)$ and y' are independent. We have

$$E(f(x) - y')^2 = (E(f(x)) - y)^2 + Var(f(x)) + Var(\epsilon), \tag{4}$$

where the first term is model bias, the second is model variance and the last is irreducible error depending on data.

[Discussion] Derive (4).

[Discussion] How to interpret $Var(\epsilon)$?

2.a. Least Square

Least square aims to find a β that minimizes MSE on training data

$$J(\beta) = \sum_{i=1}^{n} (f(x_i) - y_i)^2 = \sum_{i=1}^{n} (x_i^T \beta - y_i)^2. \tag{5}$$

In the above, we also call that sum the loss function. There are many types of loss function.

[Discussion] Geometric interpretation of minimizing $J(\beta)$.

For easier optimization, we often write $J(\beta)$ in a matrix form.[1]

$$J(\beta) = ||X\beta - Y||^2, \tag{6}$$

where $X \in \mathbb{R}^{n \times (p+1)}$ is the sample matrix with each row being an (augmented) instance and $Y \in \mathbb{R}^n$ is the label vector.

[Discussion] Derive (6).

Because $J(\beta)$ is a quadratic function of β, we can apply the "critical point method" to find its minimum point.

[Discussion] Why is $J(\beta)$ quadratic?

[Discussion] How do we know the critical point is a minimum point instead of maximum point?

Setting $J'(\beta) = 0$ and solving for β, we have

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \tag{7}$$

[Discussion] Derive (7). (check dimensions)

[Discussion] Are we making any (invertible) assumption when deriving $\hat{\beta}$?

[Discussion] When is $X^T X$ guaranteed to be not invertible? What is its connection to overfitting?

2.b. Ridge Regression

Ridge regression aims to find a β that not only minimizes MSE on training data but also has low model complexity – it achieves so by properly shrinking the parameter domains. Together, RR minimizes the following objective

$$J(\beta) = \sum_{i=1}^{n} (x_i^T \beta - y_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2, \tag{8}$$

where the second term is ℓ_2-regularization, and λ is the regularization coefficient that controls the degree of shrinkage. Note that β_0 is not included in the regularization term.

[Discussion] How does the regularization term help to shrink parameter domains? (intuitively)

[Discussion] How does λ control the degree of shrinkage?

[Discussion] What is the relation between $\min_\beta J(\beta)$ and the following constrained optimization problem? (Lagrange)

$$\min_{\beta} \sum_{i=1}^{n} (x_i^T \beta - y_i)^2, \quad \text{s.t. } \sum_{j=1}^{p} \beta_j^2 \leq \epsilon. \tag{9}$$

[Discussion] Contour of the constrained optimization problem.

We often write $J(\beta)$ in a matrix form.

$$J(\beta) = ||X\beta - Y||^2 + \lambda \beta^T I_0 \beta, \tag{10}$$

where $I_0 \in \mathbb{R}^{(p+1) \times (p+1)}$ is an 'almost identity' matrix except $I_0(0, 0) = 0$.

[Discussion] Verify (10).

Because $J(\beta)$ is a quadratic function of β, we can apply the critical point method to find a solution.

$$\hat{\beta} = (X^T X + \lambda I_0)^{-1} X^T Y. \tag{11}$$

[Discussion] Why is $J(\beta)$ quadratic?

[Discussion] Derive (11).

[Discussion] Compared the solutions of least square and ridge regression. How does λ make a difference?

[1] We will always use $|| \cdot ||$ to represent F-norm of matrix or, equivalently, ℓ_2-norm of vector.

2.c. Lasso

Like ridge, Lasso[2] aims to find a β that not only minimizes MSE on training data but also has low model complexity – but unlike ridge, it achieves simplicity by automatically selecting a subset of features to construct the model.

$$J(\beta) = \sum_{i=1}^{n}(x_i^T\beta - y_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{12}$$

where the second term is ℓ_1-regularization, and λ is the regularization coefficient that controls the degree of selection.

Minimizing $J(\beta)$ will automatically set certain regression coefficients to zero. Say, if $\beta_j = 0$, then feature $x_{\cdot j}$ will not contribute to the model prediction because $x_{\cdot j}\beta_j = 0$. This is as if feature $x_{\cdot j}$ is not used to construct f. For this reason, Lasso is also known to realize automatic feature selection.

[Discussion] How does the ℓ_1-regularization help to implement feature selection ? (contour)

[Discussion] How does λ affect the degree of feature selection?

[Discussion] Can ridge regression also implement feature selection (ℓ_2-regularization)?

We can write $J(\beta)$ in a matrix form

$$J(\beta) = ||X\beta - Y||^2 + \lambda||\beta_{[-0]}||_1, \tag{13}$$

where $\beta_{[-0]}$ is β except $\beta_0 = 0$.

[Discussion] Can we apply the critical point method to minimize $J(\beta)$?

We can minimize $J(\beta)$ using coordinate descent – its basic idea is to iteratively optimize β, one random element at a time (with other elements fixed), until some convergence criterion is met. Details are in Algorithm 1.

[Discussion] Geometric interpretation of coordinate descent.

[Discussion] Go over the sketch of the algorithm first.

[Discussion] How does Lasso implement automatic feature selection in Algorithm 1?

[Discussion] How does λ affect the degree of selection?

[Discussion] How to interpret the condition for β_j to be set zero?

Let's see how to derive the update rules (15) and (14).

For (14), we can optimize β_0 using the critical point method. This is because β_0 is not included in the ℓ_1 regularization and thus J is a quadratic function of it. Solving the following equation for β_0 gives (14).

$$\frac{\partial J}{\partial \beta_0} = \sum_{i=1}^{n} 2(x_i^T\beta - y_i) = 0, \tag{17}$$

[Discussion] Derive (17) and (14).

For (15), we can first rewrite $J(\beta)$ as a function of β_j to facilitate analysis.

$$\begin{aligned} J(\beta) = ||X\beta - Y||^2 + \lambda||\beta_{[-0]}||_1 &= ||\sum_{k=0}^{p} X_{:k}\beta_k - Y||^2 + \lambda \sum_{k=1}^{p}|\beta_k| \\ &= ||X_{:j}\beta_j + \sum_{k\neq j} X_{:k}\beta_k - Y||^2 + \lambda|\beta_j| + \lambda\sum_{k\neq j}|\beta_k| \\ &= ||X_{:j}\beta_j + A^{(j)}||^2 + \lambda|\beta_j| + B^{(j)} \\ &= \sum_{i=1}^{n}(X_{ij}\beta_j + A_i^{(j)})^2 + \lambda|\beta_j| + B^{(j)}, \end{aligned} \tag{18}$$

where $B^{(j)} = \lambda\sum_{k\neq j}|\beta_k|$ and $A^{(j)} = \sum_{k\neq j} X_{:k}\beta_k - Y = X\beta_{[-j]} - Y$.

Now, we want to get rid of the absolute value on β_j. We can do so by separately discussing three cases of β_j.

[2] Lasso stands for Least Absolute Shrinkage and Selection Operator.

Algorithm 1 Coordinate Descent for Lasso

0: Randomly initialize $\beta \in \mathcal{R}^{(p+1)}$.

while not converge **do**

1: Randomly pick an element in β, say β_j.

2: If $j = 0$, update β_j by

$$\beta_0 = -\frac{1}{n}\sum_{i=1}^{n}(x_i^T\beta_{[-0]} - y_i). \tag{14}$$

where $\beta_{[-0]}$ is β except $\beta_0 = 0$. Note that $\sum_{i=1}^{n}(x_i^T\beta_{[-0]} - y_i)$ is the sum of all elements in $X\beta_{[-0]} - Y$.

3: If $j \neq 0$, update β_j by

$$\beta_j = \begin{cases} \frac{-\lambda - 2X_{:j}^T A^{(j)}}{2||X_{:j}||^2} & \text{if } 2X_{:j}^T A^{(j)} < -\lambda \\ \frac{\lambda - 2X_{:j}^T A^{(j)}}{2||X_{:j}||^2} & \text{if } 2X_{:j}^T A^{(j)} > \lambda \\ 0 & \text{if } |2X_{:j}^T A^{(j)}| \leq \lambda \end{cases} \tag{15}$$

where $X_{:j}$ is the j_{th} column of X and

$$A^{(j)} = X\beta_{[-j]} - Y, \tag{16}$$

where $\beta_{[-j]}$ is β except $\beta_j = 0$.

end while

Case 1. If $\beta_j > 0$, then $|\beta_j| = \beta_j$. We have

$$\frac{\partial J(\beta)}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}\sum_{i=1}^{n}(X_{ij}\beta_j + A_i^{(j)})^2 + \lambda\,\beta_j + B^{(j)} = \sum_{i=1}^{n} 2X_{ij}^2\beta_j + \sum_{i=1}^{n} 2X_{ij}A_i^{(j)} + \lambda. \tag{19}$$

Solving $\frac{\partial J(\beta)}{\partial \beta_j} = 0$ for β_j gives

$$\beta_j = \frac{-\lambda - \sum_{i=1}^{n} 2X_{ij}A_i^{(j)}}{\sum_{i=1}^{n} 2X_{ij}^2} = \frac{-\lambda - 2X_{:j}^T A^{(j)}}{2||X_{:j}||^2}. \tag{20}$$

Note the solution only exists if $\beta_j > 0$, which means $-\lambda - 2X_{:j}^T A^{(j)} > 0$. Thus the condition of this solution is

$$\lambda < -2X_{:j}^T A^{(j)}. \tag{21}$$

Case 2. If $\beta_j < 0$, then $|\beta_j| = -\beta_j$. Following the same analysis in Case 1, we have

$$\beta_j = \frac{\lambda - 2X_{:j}^T A^{(j)}}{2||X_{:j}||^2}, \tag{22}$$

with the condition

$$\lambda < 2X_{:j}^T A^{(j)}. \tag{23}$$

[Discussion] Derive (22).

Case 3. If none of the two conditions (21) and (23) are satisfied, we have $\beta_j = 0$.