Machine Learning, Spring 2019 Chao Lan
EN 2105 EN 4084A
MWF 9-9:50am clan@uwyo.edu

Linear Methods for Regression

A basic model for the regression task is linear regression. It *assumes* label is a linear combination of features[1] and has the form

$$f(x) = \beta_0 + \beta_1 x_{\cdot 1} + \beta_2 x_{\cdot 2} + \ldots + \beta_p x_{\cdot p}, \tag{1}$$

where $\beta_1, \ldots, \beta_p$ are regression coefficients and β_0 is a bias term. The β's are model parameters and will be learned from data.

[*Discussion*] What is the geometric interpretation of $f(x)$? What if we remove β_0?

We often write $f(x)$ in a matrix form to facilitate analysis. Let's augment x with a constant feature 1 so that

$$\tilde{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \in \mathbb{R}^{p+1}. \tag{2}$$

Then

$$f(x) = \beta_0 \tilde{x}_{\cdot 0} + \beta_1 \tilde{x}_{\cdot 1} + \ldots + \beta_p \tilde{x}_{\cdot p} = \sum_{i=0}^{p} \tilde{x}_{\cdot i} \beta_i = \tilde{x}^T \tilde{\beta}, \tag{3}$$

where $\beta = [\beta_0, \beta_1, \ldots, \beta_p]^T$ is a $p+1$-dimensional vector (to be learned from data). From now on, we always assume data are augmented and thus omit all tilde notations in (3).

There are many methods to learn β, using different objective functions with different purposes. We will go over (ordinary) least square, ridge regression and Lasso.[2]

Least square minimizes the following objective function

$$J(f) = \sum_{i=1}^{n} (f(x_i) - y_i)^2. \tag{4}$$

It simply finds a model that best fits observed sample by minimizing the squared loss on it.

[*Discussion*] What is the geometric interpretation of prediction loss?

The above least square problem has an analytic solution. Its $J(f)$ can be written as

$$\begin{aligned} J(\beta) = \sum_{i=1}^{n} (x_i^T \beta - y_i)^2 &= (X\beta - Y)^T (X\beta - Y) = ||X\beta - Y||_2^2 \\ &= \beta^T X^T X \beta - 2\beta^T X^T Y + Y^T Y, \end{aligned} \tag{5}$$

[1] There are debates on whether the function is actually 'linear' or 'affine' (since it has a bias term β_0). We will ignore this and just call the model linear (because we can augment data with a constant feature 1).

[2] We will briefly mention weighted least square and you will derive it yourself in assignment.

where $X \in \mathbb{R}^{n\times(p+1)}$ is a data matrix with x_i^T being the i_{th} row, and $Y \in \mathbb{R}^n$ is a label vector with y_i being its i_{th} element.

[*Exercise*] Derive (5).

From (5) we see $J(\beta)$ is a quadratic function, so we can apply the critical point method to find its minimum point. We can follow three steps:

1. compute derivative of $J(\beta)$ w.r.t. β, i.e.,
$$J'(\beta) = 2X^T X\beta - 2X^T Y \tag{6}$$
2. set $J'(\beta) = 0$, i.e.,
$$2X^T X\beta - 2X^T Y = 0 \tag{7}$$
3. solve $J'(\beta) = 0$ for β (assuming X is full-rank[3]), i.e.,
$$\hat{\beta} = (X^T X)^{-1} X^T Y. \tag{8}$$

Least square has no control on the model complexity, because β can take arbitrary value. Thus it is likely to overfit when sample size n is small.

[*Discussion*] When is $X^T X$ likely to be rank-deficient?

Weighted Least square minimizes the following objective function

$$J(\beta) = \sum_{i=1}^{n} w_i \cdot (x_i^T \beta - y_i)^2. \tag{9}$$

It assigns different weights w_i to different examples. The model will focus on learning examples with higher weights (and therefore achieving higher accuracy on these examples).

[*Discussion*] When do we want the model to focus on fitting some particular examples?

Ridge regression minimizes the following objective function

$$J(\beta) = \sum_{i=1}^{n} (x_i^T \beta - y_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2, \tag{10}$$

where the second term is often referred as L2-regularization. It helps to reduce model complexity by shrinking regression coefficients (but not β_0).

[*Discussion*] Why doesn't ridge regression shrink the bias term?

The ridge regression problem also has an analytic solution. Its $J(f)$ can be written as

$$J(\beta) = \sum_{i=1}^{n} (x_i^T \beta - y_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = ||X\beta - Y||_2^2 + \lambda \beta^T I_0 \beta, \tag{11}$$

[3] If X is not full rank, we can use ridge regression.

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

[*Exercise*] Derive (11). Why is $I_0(0,0) = 0$?

From (11) we see $J(\beta)$ is also a quadratic function. We can apply the critical point method:

1. compute derivative of $J(\beta)$ w.r.t. β, i.e.,
$$J'(\beta) = 2X^T X\beta - 2X^T Y + 2\lambda I_0\beta = 2(X^T X + \lambda I_0)\beta - 2X^T Y. \tag{12}$$
2. set $J'(\beta) = 0$ and solve for β, i.e.,
$$\hat{\beta} = (X^T X + \lambda I_0)^{-1} X^T Y. \tag{13}$$

Note $X^T X + \lambda I_0$ is less likely to be singular. Ridge regression can control model complexity but cannot select features.

[*Exercise*] How would β change as λ increases?

Lasso minimizes the following objective function

$$J(\beta) = \sum\nolimits_{i=1}^{n} (x_i^T \beta - y_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{14}$$

where the second term is L1-regularization. It helps to select features by automatically shrinking the regression coefficients of some features to zero (so they are removed from the model). The ideal constraint for feature selection is actually L0-regularization, but optimizing with this constraint is NP-hard and people find that L1 is a good approximation of L0.

[*Discussion*] How can L1-regularization shrink some coefficients to zero?

We often see $J(\beta)$ in a matrix form, e.g.,

$$J(\beta) = \sum\nolimits_{i=1}^{n} (x_i^T \beta - y_i)^2 + \lambda \sum\nolimits_{j=1}^{p} |\beta_j| = ||X\beta - Y||_2^2 + \lambda ||\tilde{\beta}||_1, \tag{15}$$

where $\tilde{\beta}$ is β with $\beta_0 = 0$. Lasso has no analytic solution as $||a||_1$ is not differentiable at $a = 0$.

[*Discussion*] Why is $||a||_1$ not differentiable at $a = 0$?

To solve Lasso, we can solve β_0 and the rest β_j's in different ways. We can optimize β_0 using the critical point method (as it is not included in the L1-constraint).

$$\frac{\partial J}{\partial \beta_0} = \sum\nolimits_{i=1}^{n} 2(x_i^T \beta - y_i). \tag{16}$$

Solving $\frac{\partial J}{\partial \beta_0} = 0$ gives

$$\beta_0 = -\frac{1}{n} \sum\nolimits_{i=1}^{n} (\tilde{x}_i^T \tilde{\beta} - y_i), \tag{17}$$

where $\tilde{x}_i \in \mathbb{R}^p$ is x_i without the augmented feature 1, and $\tilde{\beta} \in \mathbb{R}^p$ is β *without* β_0.

Algorithm 1 The Coordinate Descent Algorithm

0: (randomly) initialize $\beta \in \mathcal{R}^{(p+1)}$.
while β is not converged **do**
  1: (randomly) pick up a feature index $j \in \{1, 2, ..., p\}$
  2(a): if $j \neq 0$, update β_j using (25)
  2(b): if $j = 0$, update β_j using (17)
end while

For the rest β_j's, we can apply the a numerical method called coordinate descent. It iteratively updates β; in each iteration it optimizes a random β_j while fixing the rest.

[*Discussion*] What is the geometric interpretation of coordinate descent?

In each iteration, there is a closed-form solution for β_j. Recall $j \neq 0$. Rewrite $J(\beta)$ as

$$\begin{aligned} J(\beta) &= ||X\beta - Y||_2^2 + \lambda||\tilde{\beta}||_1 \\ &= ||\sum\nolimits_{k=0}^{p} X_{:k}\beta_k - Y||_2^2 + \lambda\sum\nolimits_{k=1}^{p}|\beta_k| \\ &= ||X_{:j}\beta_j + \sum\nolimits_{k\neq j} X_{:k}\beta_k - Y||_2^2 + \lambda|\beta_j| + \lambda\sum\nolimits_{k\neq j}|\beta_k| \\ &= ||X_{:j}\beta_j + A^{(j)}||_2^2 + \lambda|\beta_j| + B^{(j)} \\ &= \sum\nolimits_{i=1}^{n}(X_{ij}\beta_j + A_i^{(j)})^2 + \lambda|\beta_j| + B^{(j)} = J(\beta_j) \end{aligned} \tag{18}$$

where $X_{:k}$ is k_{th} column of X, and $B^{(j)} = \lambda\sum_{k\neq j}|\beta_k|$ and

$$A^{(j)} = \sum\nolimits_{k\neq j} X_{:k}\beta_k - Y = X\beta_{[-j]} - Y \in \mathbb{R}^n, \tag{19}$$

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

We want to find a β_j that minimizes $J(\beta_j)$. Critical point method does not apply as $|\beta_j|$ is not differentiable. To overcome this, we can remove the absolute value by case-studying β_j.

Case 1: $\beta_j > 0$.

$$\begin{aligned} \frac{\partial}{\partial\beta_j}J(\beta) &= \frac{\partial}{\partial\beta_j}\sum\nolimits_{i=1}^{n}(X_{ij}\beta_j + A_i^{(j)})^2 + \lambda\cdot\beta_j + B^{(j)} \\ &= \sum\nolimits_{i=1}^{n} 2X_{ij}(X_{ij}\beta_j + A_i^{(j)}) + \lambda \\ &= \sum\nolimits_{i=1}^{n} 2X_{ij}^2\cdot\beta_j + \sum\nolimits_{i=1}^{n} 2X_{ij}A_i^{(j)} + \lambda \end{aligned} \tag{20}$$

Setting the ride-hand-side (RHS) to zero and solving for β_j, we have

$$\beta_j = \frac{-\lambda - \sum_{i=1}^{n} 2X_{ij}A_i^{(j)}}{\sum_{i=1}^{n} 2X_{ij}^2} = \frac{-\lambda - 2X_{:j}^T A^{(j)}}{||X_{:j}||_2^2}, \tag{21}$$

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

The RHS must be positive for the solution to be valid (i.e., $\beta_j > 0$). Thus the condition is

$$\lambda < -2X_{:j}^T A^{(j)}. \tag{22}$$

Case 2: $\beta_j < 0$. Similar to Case 1, we have

$$\beta_j = \frac{\lambda - 2X_{:j}^T A^{(j)}}{||X_{:j}||_2^2}. \tag{23}$$

The RHS must be negative for $\beta_j < 0$. Thus the condition is

$$\lambda < 2X_{:j}^T A^{(j)}. \tag{24}$$

[*Exercise*] Derive (23).

Case 3: $\beta_j = 0$.

If neither condition (22) or (24) is satisfied, then the only solution is $\beta_j = 0$.

Wrap Up. Summarizing the three case, we have

$$\beta_j = \begin{cases} \frac{-\lambda - 2X_{:j}^T A^{(j)}}{||X_{:j}||_2^2} & \text{if } 2X_{:j}^T A^{(j)} < -\lambda \\ \frac{\lambda - 2X_{:j}^T A^{(j)}}{||X_{:j}||_2^2} & \text{if } 2X_{:j}^T A^{(j)} > \lambda \\ 0 & \text{if } |2X_{:j}^T A^{(j)}| \leq \lambda \end{cases} \tag{25}$$

We see β_j will be zero if λ is big enough to satisfy $|2X_{:j}^T A^{(j)}| \leq \lambda$. We also see increasing λ will set more β_j's to zero, resulting in a sparser linear model.

Coordinate descent is guaranteed to find local minimum point. For Lasso, it is guaranteed to find global minimum point because J is jointly convex over all β_j.

[*Exercise*] What's the strategy of optimizing L1 constraint? What makes coordinate descent a good optimizer?