DSEB K61, MFE, NEU

Machine Learning Assignment #4

Tô Đức Anh | 11196328

September 2021

1 Problems

1. Prove $X^T X$ is invertible when X is full rank.
2. Re-transform linear regression to LaTeX, prove that t = y(x,w) + noise -> w = $(X^T X)^{-1} X^T t$.
3. Using numpy, find linear regression model for house's price predicting problems, using this dataset.
4. Plot prediction model (straight line) and data (point - scatter).
5. Predict price of houses that have the area of 50, 100, 150 accordingly.

2 Answers

2.1 Answer 1

Let's say we have:

$$A_{nxm} = \left[\vec{a_1}, \vec{a_2}, \vec{a_3}, ..., \vec{a_n}\right]$$

So if matrix A is full rank, all the vectors that lie in matrix A will be linear independent.
We have 3 cases:
n = m
n > m
In the first case, we can see that if $n = m$, if matrix A fullrank, all vectors that made up A will be linear independent, so that it could transform to Identity Matrix, so that it is invertible.
In the second case, it requires us to have full row rank, which means all the rows are linear independent.
Suppose $X^T v = 0$. Then, of course, $XX^T v = 0$ too.
Conversely, suppose that $XX^T v = 0$. Then $v^T XX^T v = 0$, so that $(X^T v)^T (X^T v) = 0$. This implies $X^T v = 0$.
Hence, we have proved that $X^T v = 0$ if and only if v is in the nullspace of $XX^T v$. But $X^T v = 0$. and $v \neq 0$ if and only if X has linearly dependent rows. Thus, XX^T has nullspace 0 (i.e. XX^T is invertible) if and only if X has linearly independent rows.

2.2 Answer 2

We have the formula for a linear equation:

$$y = ax + b$$

But to make it easier to work with, we transform the formula into:

$$y = w_1 * x + b$$

And it can be extended to:

$$y = w_0 + w_1 * x_1 + w_2 * x_2 + ... + w_i * x_i$$

We have a data set of observations $x = (x_1, x_2, ..., x_N)^T$, representing N observations of the scalar variable x and their corresponding target values $t = (t_1, t_2, ..., t_N)^T \rightarrow$ make predictions for some new values of the input variable x.

Suppose that the observations are drawn independently from a Gaussian distribution. Data points that are drawn independently from the same distribution are said to be independent and identically distributed (i.i.d)
We have:

$$t = y(x, w) + \epsilon$$

With t is the real value in the dataset, $y(x, w)$ as the linear equation and ,ϵ as error or noise.
Since ϵ follows the normal distribution, so that ϵ can be written as:

$$\epsilon = \mathcal{N}(\mu, \sigma^2)$$

But we can adjust the w_0 in the linear equation, so that we can set the μ to 0, so that ϵ can be normalized as:

$$\epsilon = \mathcal{N}(0, \sigma^2)$$

t can now be re-written as:

$$t = y(x, w) + \mathcal{N}(0, \sigma^2)$$

We can see that when adding a value into a normal distribution would only shifts the μ, so that after transforming t is:

$$t = y(x, w) + \mathcal{N}(0, \sigma^2) \equiv \mathcal{N}(y(x, w), \sigma^2)$$

With Precision parameter:

$$\beta = \frac{1}{\sigma^{-1}}$$

t can be re-written again as:

$$p(t|x, w, \beta) = \mathcal{N}(t|y(x, w), \beta^{-1})$$

We now use the training data x, t to determine the values of the unknown parameters w and by maximum likelihood. If the data are assumed to be drawn independently from the distribution then the likelihood function:

$$p(t|x, w, \beta) = \prod_{n=1}^{n} \mathcal{N}(t_n|y(x_n, w), \beta^{-1})$$

It is convenient to maximize the logarithm of the likelihood function:

$$\begin{aligned}\log_e p(t|x, w, \beta) &= \sum_{n=1}^{n} \frac{1}{\sqrt{2\pi\beta^{-1}}} e^{-(t_n - y(x_n, w)^2 * \frac{\beta}{2})} \\ &= \sum_{n=1}^{n} \big(- \frac{1}{2} \log_e 2\pi\beta^{-1}\big) - \Big(t_n - y\big(x_n, w\big)\Big)^2 - \frac{\beta}{2} \qquad (1) \\ &= -\sum_{n=1}^{n} (t_n - y(x_n, w))^2)\end{aligned}$$

We put away $-\frac{1}{2}\log_e 2\pi\beta^{-1}$ and $\frac{\beta}{2}$ since they are not the element we want to optimize, they are just number anyways, we are focusing on optimizing $\sum_{n=1}^{n}(t_n - y(x_n, w))^2)$
Since we are maximizing $\log_e p(t|x, w, \beta)$ so that we have to minimize $\frac{1}{2}\sum_{n=1}^{n}(t_n - y(x_n, w))^2)$ since we are subtracting it from the equation.
We have:

$$t_n - y(x_n, w))^2 = w_1 + x_n + w_0$$

And:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix}, t = \begin{bmatrix} t_1 \\ t_2 \\ ... \\ t_n \end{bmatrix}, w = \begin{bmatrix} w_1 \\ w_0 \end{bmatrix}$$

y also equal:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_0 \\ w_1x_2 + w_0 \\ ... \\ w_1x_n + w_0 \end{bmatrix}$$

Which equal to: xW But to do matrix calculation, we have to add a column contains 1 to the vector x, so that x will become:

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ ... & ... \\ 1 & x_n \end{bmatrix}$$

And y = XW
We have the distance between our prediction and the real value is $t - y$:

$$\begin{bmatrix} t_1 - y_1 \\ t_2 - y_2 \\ ... \\ t_n - y_n \end{bmatrix} = \|(t - y)\|_2^2 = \sum_{1}^{n}(t - y)^2$$

So that we have our loss function:

$$L = \|(t - y)\|_2^2 = \|(t - XW)\|_2^2$$

To minimize the Loss function, we let its derivative = 0

$$\frac{dL}{dw} = 0$$

Using Vector Calculus, We can see that, $\frac{dL}{dw}$ is:

$$\frac{dL}{dW} = \frac{\begin{bmatrix} t_1 - y_1 \\ t_2 - y_2 \\ ... \\ t_n - y_n \end{bmatrix}}{dW} = \begin{bmatrix} \frac{t_1 - y_1}{dW} \\ \frac{t_2 - y_2}{dW} \\ ... \\ \frac{t_n - y_3}{dW} \end{bmatrix} = \begin{bmatrix} \frac{t_1 - x_1 w_1}{dW} \\ \frac{t_2 - x_2 w_2}{dW} \\ ... \\ \frac{t_n - x_n w_n}{dW} \end{bmatrix} = \begin{bmatrix} 2x_1 \frac{t_1 - x_1 w_1}{dW} \\ 2x_n \frac{t_2 - x_2 w_2}{dW} \\ ... \\ 2x_n \frac{t_n - x_n w_n}{dW} \end{bmatrix} = 2X^T(t - XW)$$

We have that $2X^T(t - XW) = 0$, so that:

$$X^T t - X^T X w = 0$$

$$X^T t = X^T X w$$

$$w = (X^T X)^{-1} X^T t$$