Linear Regression

Tim Widmoser

HWR Berlin, Fachbereich 2

December 16, 2025

Abstract

Useful Notes of Linear Regression's Basic Math Concepts

***Keywords*.** Machine Learning · Mathematics · Regression · Vector

1 Linear Regression

This work heavily relies on [1]. The goal is to find a function f such that $x \in \mathbb{R}^d$ get mapped to $f(x) \in \mathbb{R}$ given that the training input x_n is slightly noisy meaning $y_n = f(x_n) + \varepsilon$. Since noise is random, it makes sense to work with likelihood function. Usually it is denoted:

$$p(y|x) = \mathcal{N}(y|f(x), \sigma^2) \tag{1.1}$$

Note that $y = f(x) + \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. We are looking for a function that is as close as possible to the **unknown f(x)** that generated the data which is not over or underfit.

2 Parameters

Let's assume that σ^2 is known. A parametrized model can look along the lines of:

$$p(y|x, \theta) = \mathcal{N}(y|x^T\theta, \sigma^2) \Longleftrightarrow y = x^T\theta + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

Note that $\theta \in \mathbb{R}^D$ and that the likelihood is gaussian due to the noise being gaussian.These parameters can be guessed but should especially in higher dimensions mathematically estimated. Gradient Descent or Ascent will come in handy. For now, let $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}, y \in \mathbb{R}$ be a training set. Also note that y_i, y_j are conditionally independent since x_i, x_j are their respective inputs. Of course we assume $i \neq j$ here. Thus we can define the likelihood function as follows:

$$\begin{aligned} p(\mathcal{Y}|\mathcal{X}, \theta) &= p(y_1, ..., y_N|x_1, ..., x_N, \theta) \\ &= \prod_{n=1}^{N} p(y_n|x_n, \theta) \\ &= \prod_{n=1}^{N} \mathcal{N}(y_n|x_n^T\theta, \sigma^2) \end{aligned}$$

The last step holds due to (1.1). Here we defined $\mathcal{X} := \{x_1, ..., x_N\}$ and $\mathcal{Y} = \{y_1, ..., y_N\}$ as training sets with inputs and target. We define an optimal parameter as $\theta^* \in \mathbb{R}^D$. The goal is that for an arbitrary x_*, the distribution of y_* is

$$p(y_*|x_*, \theta) = \mathcal{N}(y_*|x_*^T, \sigma^2)$$

The parameter can be estimated by maximizing the likelihood.

2.1 Maximum Likelihood Estimation

Note that in the following, the function is not a probability distribution in θ it is a function with parameters θ.

$$\theta_{\mathrm{ML}} \in \arg\max_{\theta} p(\mathcal{Y}|\mathcal{X}, \theta)$$

We can use the log-likelihood to account for the fact of precision loss when mulitplying a lot of small values for example. Since the log has great properties of handling multiplications via additions.

$$\begin{aligned} -\log p(\mathcal{Y}|\mathcal{X}, \theta) &= -\log \prod_{n=1}^{N} p(y_n|x_n, \theta) \\ &= -\sum_{n=1}^{N} \log p(y_n|x_n, \theta) \end{aligned} \tag{2.1}$$

Remember the Gaussian $\mathcal{N}(x|\mu, \sigma^2)$:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x-\mu^2}{2\sigma^2}}$$

From

$$p(y|x, \theta) = \mathcal{N}\left(\underbrace{y}_{x} | \underbrace{x^T\theta}_{\mu}, \sigma^2\right) \iff y = x^T\theta + \varepsilon$$

we can derive:

$$\begin{aligned} \log p(y_n|x_n, \theta) &= \log \mathcal{N}(y_n|x_n^T\theta, \sigma^2) \\ &= \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_n - x_n^T\theta\right)^2}{2\sigma^2}}\right) \\ &= \underbrace{\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)}_{\text{const. } \delta} \cancel{\log}\left(\cancel{e}^{-\frac{\left(y_n - x_n^T\theta\right)^2}{2\sigma^2}}\right) \\ &= -\frac{\left(y_n - x_n^T\theta\right)^2}{2\sigma^2} + \delta \\ &= -\frac{1}{2\sigma^2}\left(y_n - x_n^T\theta\right)^2 + \delta \qquad \square \end{aligned}$$

Plugging this result (ignoring δ) into (2.1) gives:

$$\begin{aligned} \mathcal{L}(\theta) &= \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(y_n - x_n^T\theta\right)^2 \\ &= \frac{1}{2\sigma^2}(y - X\theta)^T(y - X\theta) \\ &= \frac{1}{2\sigma^2}\|y - X\theta\|^2 \end{aligned} \tag{2.2}$$

with $X := [x_1, ..., x_N]^T \in \mathbb{R}^{N \times D}$ and $y := [y_1, ..., y_N]^T \in \mathbb{R}^N$. The n-th row of the matrix corresponds to the training input of x_n. Since we want to optimize for θ and its a square order function we can compute the gradient of $\mathcal{L}$ and solve for θ setting the gradient 0.

Computing further gives:

$$\begin{aligned}
\frac{d\mathcal{L}}{d\theta} &= \frac{d}{d\theta}\left(\frac{1}{2\sigma^2}(y - X\theta)^T(y - X\theta)\right) \\
&= \frac{1}{2\sigma^2}\frac{d}{d\theta}((y - X\theta)^T(y - X\theta)) \\
&= \frac{1}{2\sigma^2}\frac{d}{d\theta}\left(\underbrace{y^T y}_{\to 0} - 2y^T X \underbrace{\theta}_{\to 1} + \theta^T X^T X\theta\right) \\
&\text{Note that } \theta^T \underbrace{X^T X}_{\text{symmetric}} \theta = 2\theta^T X^T X \\
&= \frac{1}{2\sigma^2}(-2y^T X + 2\theta^T X^T X) \\
&= \frac{1}{\sigma^2}(-y^T X + \theta^T X^T X) \in \mathbb{R}^{1 \times D} \\
&\text{We now set } \frac{d\mathcal{L}}{d\theta} = 0 \text{ to obtain } \theta_{\mathrm{ML}} \\
&\frac{d\mathcal{L}}{d\theta} = 0^T \Longleftrightarrow \\
&0^T = -y^T X + \theta_{\mathrm{ML}}^T X^T X \\
&\Leftrightarrow \theta_{\mathrm{ML}}^T X^T X = y^T X \\
&\Leftrightarrow \theta_{\mathrm{ML}}^T = y^T X (X^T X)^{-1} \\
&\Leftrightarrow \theta_{\mathrm{ML}}^T = (X^T X)^{-1} X^T y
\end{aligned}$$

We will obtain a global minimum since $\nabla_\theta^2 \mathcal{L}(\theta) = X^T X \in \mathbb{R}^{D \times D}$ is positive definite. □

This procedure is similar to solving a System of Linear Equations:

$$\theta_{\mathrm{ML}}^T = \underbrace{(X^T X)^{-1}}_{A^{-1}} \underbrace{X^T y}_{b}$$

This resembles $A\theta = b$.

Linear Regression must not only be used to find optimal linear functions. One can also transform x nonlinearly to $\Phi^T(x)$. This gives:

$$\begin{aligned}
p(x|y, \theta) &= \mathcal{N}(y|\Phi^T(x)\theta, \sigma^2) \\
&\Longleftrightarrow y = \Phi^T(x)\Theta + \varepsilon \\
&= \sum_{k=0}^{K-1} \theta_k \Phi_k(x) + \varepsilon
\end{aligned}$$

Here $\Phi : \mathbb{R}^D \to \mathbb{R}^K$ and $\Phi_k : \mathbb{R}^D \to \mathbb{R}$. Polynomial Regression will be discussed another time.

Bibliography

[1] C. S. O. Marc Deisenroth Aldo Faisal, “Mathematics for Machine Learning,” 2024, [Online]. Available: https://mml-book.github.io/book/mml-book.pdf