Linear Regression

A unification of Maximum Likelihood Estimation and minimizing the sum of squares.

William Fleshman Feb 11, 2019 · 7 min read ★

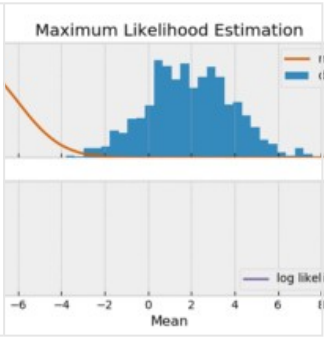
Introduction

I recently wrote about maximum likelihood estimation in my ongoing series on the fundamentals of machine learning:

Maximum Likelihood Estimation

Fundamentals of Machine Learning (Part 2)

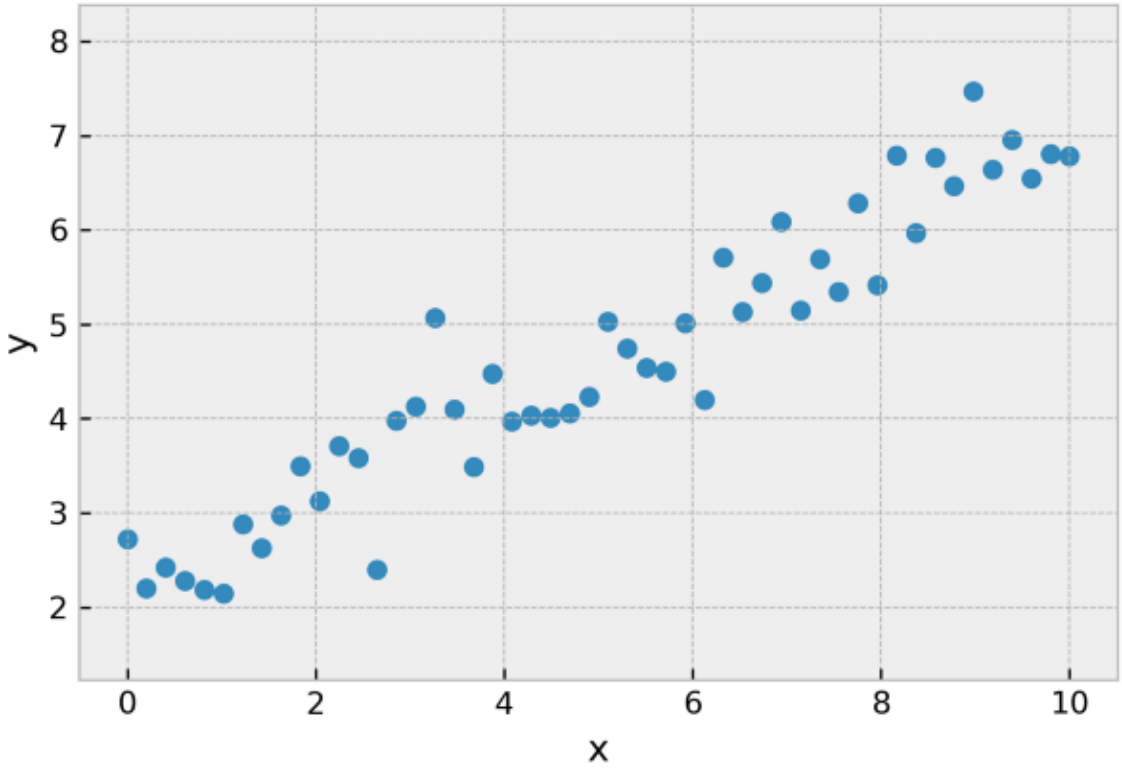
towardsdatascience.com



In that post, we learned what it means to "model" data, and then how to use MLE to find the parameters of our model. In this post, we're going to dive into linear regression, one of the most important models in statistics, and learn how to frame it in terms of MLE. The solution is a beautiful piece of mathematics, which like most MLE models, is rich with intuition. I'll be assuming you've got a handle on the vocabulary I've covered in the other series (probability densities, conditional probabilities, likelihood function, iid data, etc.). If you see something here that you're uncomfortable with, check out the Probability and MLE posts from that series for clarity.

The Model

We use linear regression when our data has a linear relationship between the independent variables (our features) and the dependent variable (our target). In the MLE post, we saw some data that looked similar too this:



We observed that there appears to be a linear relationship between x and y, but it's not perfect. We think of these imperfections as coming from some error or noise process. Imagine drawing a line right through the cloud of points. The error for each point would be the distance from the point to our line. We'd like to explicitly include those errors in our model. One method of doing this, is to assume the errors are distributed from a Gaussian distribution with a mean of 0 and some unknown variance σ^2. The Gaussian seems like a good choice, because our errors look like they're symmetric about were the line would be, and that small errors are more likely than large errors. We write our linear model with Gaussian noise like this:

$$\epsilon \sim N(0, \sigma^2)$$

Linear model with Gaussian noise term.

The error term is draw from our Gaussian, and then our observed y is then calculated by adding the error to the output of the linear equation. This model has three parameters: the slope and intercept of our line and the variance of the noise distribution. Our main goal is to find the best parameters for the slope and intercept of our line.

. . .

Likelihood Function

To apply maximum likelihood, we first need to derive the likelihood function. First, let's rewrite our model from above as a single conditional distribution given x:

$$y \sim N(\theta_1 x + \theta_0, \sigma^2)$$

Given x, y is drawn from a Gaussian centered on our line.

This is equivalent to pushing our x through the equation of the line and then adding noise from the 0 mean Gaussian.

Now, we can write the conditional distribution of y given x in terms of this Gaussian. This is just the equation of a Gaussian distribution's probability density function, with our linear equation in place of the mean:

$$f(y|x;\theta_0,\theta_1,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-(\theta_1 x+\theta_0))^2}{2\sigma^2}}$$

PDF of y given x and our linear model.

Each point is independent and identically distributed (iid), so we can write the likelihood function with respect to all of our observed points as the product of each individual probability density. Since σ^2 is the same for each data point, we can factor out the term of the Gaussian which doesn't include x or y from the product:

$$L_X(\theta_0, \theta_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \prod_{(x,y)\in X} e^{\frac{-(y-(\theta_1 x+\theta_0))^2}{2\sigma^2}}$$

Likelihood for our collection of data X.

(Note: Thanks to Tongxin for catching an error here. The term before the product should be raised to the number of data points because I factored it from the product.)

Log-Likelihood:

The next step in MLE, is to find the parameters which maximize this function. To make our equation simpler, let's take the log of our likelihood. Recall, that maximizing the log-likelihood is the same as maximizing the likelihood since the log is monotonic. The natural log cancels out with the exponential, turns products into sums of logs, and division into subtraction of logs; so our log-likelihood looks much simpler:

$$l_X(\theta_0, \theta_1, \sigma^2) = \log[\frac{1}{\sqrt{2\pi\sigma^2}} \prod_{(x,y)\in X} e^{\frac{-(y-(\theta_1 x+\theta_0))^2}{2\sigma^2}}]$$

$$= \log(\frac{1}{\sqrt{2\pi\sigma^2}}) + \sum_{(x,y)\in X} \log(e^{\frac{-(y-(\theta_1 x+\theta_0))^2}{2\sigma^2}})$$

$$= \log(\frac{1}{\sqrt{2\pi\sigma^2}}) + \sum_{(x,y)\in X} \frac{-(y-(\theta_1 x+\theta_0))^2}{2\sigma^2}$$

$$= \log(1) - \log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2} \sum_{(x,y)\in X} [y - (\theta_1 x + \theta_0)]^2$$

$$= -\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2} \sum_{(x,y)\in X} [y - (\theta_1 x + \theta_0)]^2$$

Sum of Squared Errors:

To clean things up a bit more, let's write the output of our line as a single value:

$$\hat{y} = \theta_1 x + \theta_0$$

Estimate of y from our line.

Now our log-likelihood can be written as:

$$l_X(\theta_0, \theta_1, \sigma^2) = -\log(\sqrt{2\pi\sigma^2}) - \frac{1}{2\sigma^2}\sum(y - \hat{y})^2$$

Simplified log-likelihood equation.

To remove the negative signs, let's recall that maximizing a number is the same thing as minimizing the negative of the number. So instead of maximizing the likelihood, let's minimize the negative log-likelihood:

$$-l_X(\theta_0, \theta_1, \sigma^2) = \log(\sqrt{2\pi\sigma^2}) + \frac{1}{2\sigma^2}\sum(y - \hat{y})^2$$

Minimize the negative log-likelihood.

Our ultimate goal is to find the parameters of our line. To minimize the negative log-likelihood with respect to the linear parameters (the θs), we can imagine that our variance term is a fixed constant.

Removing any constant's which don't include our θs won't alter the solution. Therefore, we can throw out any constant terms and elegantly write what we're trying to minimize as:

$$\sum(y - \hat{y})^2$$

The maximum likelihood estimate for our linear model is the line which minimizes the sum of squared errors! This is a beautiful result, and you'll see that minimizing squared errors crops up everywhere in machine learning and statistics.

. . .

Solving for Parameters

We've concluded that the maximum likelihood estimates for our slope and intercept can be found by minimizing the sum of squared errors. Let's expand out our minimization objective and use i as our index over our n data points:

$$\begin{aligned} \text{SSE} &= \sum_i^n (y_i - \hat{y}_i)^2 \\ &= \sum_i^n [y_i - (\theta_1 x_i + \theta_0)]^2 \\ &= \sum_i^n (y_i - \theta_1 x_i - \theta_0)^2 \end{aligned}$$

The square in the SSE formula makes it quadratic with a single minimum. The minimum can be found by taking the derivative with respect to each of the parameters, setting it equal to 0, and solving for the parameters in turn.

The Intercept:

Let's start by solving for the intercept. Taking the partial derivative with respect to the intercept and working through gives us:

$$\begin{aligned} \frac{\partial}{\partial \theta_0} \text{SSE} &= -2 \sum_i^n (y_i - \theta_1 x_i - \theta_0) \\ &= -2 \sum_i^n y_i + 2\theta_1 \sum_i^n x_i + 2n\theta_0 \end{aligned}$$

$$= -2n\bar{y} + 2\theta_1 n\bar{x} + 2n\theta_0$$

Derivative of SSE with respect to the intercept of our line.

The horizontal bars over the variables indicate the mean of those variables. We used the fact that the sum over the values of a variable is equal to the mean of those values multiplied by how many values we have. Setting the derivative equal to 0 and solving for the intercept gives us:

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

MLE for intercept.

This is a pretty neat result. It's the equation of the line, with the means of the x's and y's in place of those variables. The intercept still depends on the slope, so we'll need to find that next.

The Slope:

We start by taking the partial derivative of the SSE with respect to our slope. We plug in our solution for the intercept and use algebra to isolate the slope term:

$$\frac{\partial}{\partial \theta_1} \text{SSE} = -2 \sum_i^n (y_i - \theta_1 x_i - \theta_0) x_i$$

$$= -2 \sum_i^n x_i y_i + 2\theta_1 \sum_i^n x_i^2 + 2\theta_0 \sum_i^n x_i$$

$$= -2 \sum_i^n x_i y_i + 2\theta_1 \sum_i^n x_i^2 + 2(\bar{y} - \theta_1 \bar{x}) \sum_i^n x_i$$

$$= -2 \sum_i^n x_i y_i + 2\theta_1 \sum_i^n x_i^2 + 2\bar{y} \sum_i^n x_i - 2\theta_1 \bar{x} \sum_i^n x_i$$

$$= 2\theta_1 \sum_i^n x_i^2 - 2\theta_1 \bar{x} \sum_i^n x_i + 2\bar{y} \sum_i^n x_i - 2 \sum_i^n x_i y_i$$

$$= 2\theta_1 (\sum_i^n x_i^2 - \bar{x} \sum_i^n x_i) + 2\bar{y} \sum_i^n x_i - 2 \sum_i^n x_i y_i$$

Setting this equal to 0 and solving for the slope gives us:

$$\theta_1 = \frac{\sum_i^n x_i y_i - n\bar{y}\bar{x}}{\sum_i^n x_i^2 - n\bar{x}^2}$$

While we're technically done, we can use some fancier algebra to rewrite this without having to use the n:

$$\theta_1 = \frac{\sum_i^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_i^n (x_i - \bar{x})^2}$$

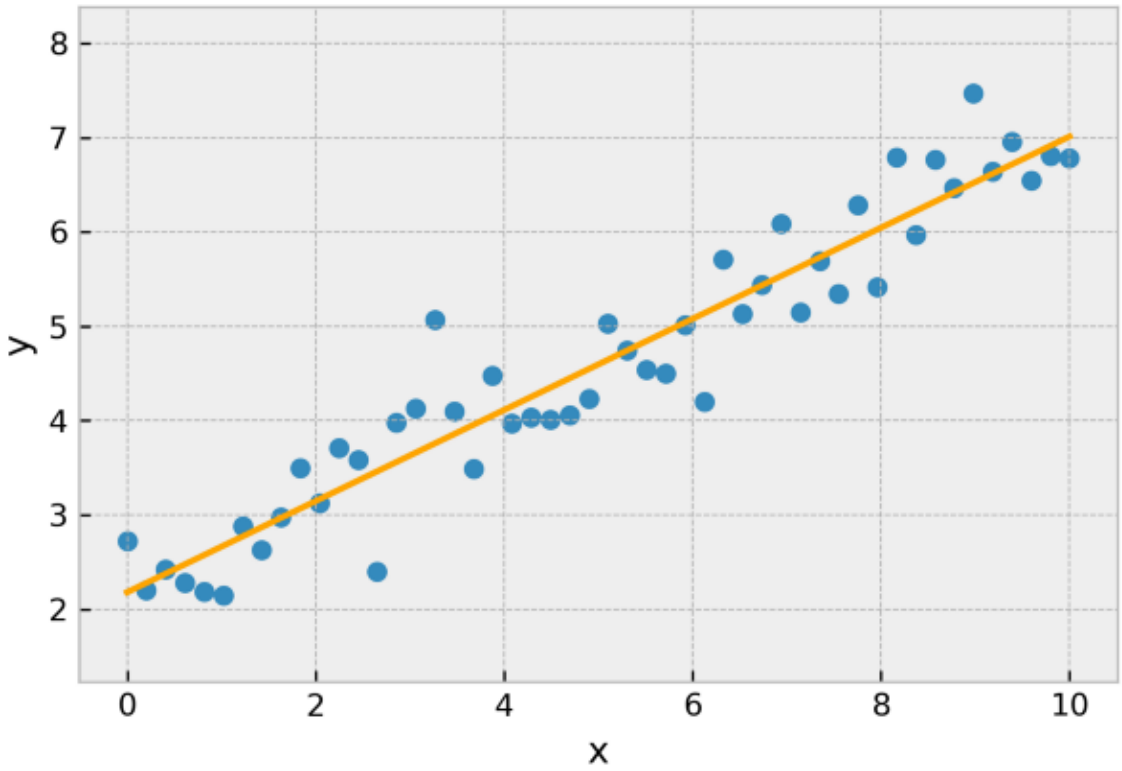
MLE estimate of the slope.

Putting it all together:

We can use these derived equations to write a simple function in python for solving the parameters for any line given at least two points:

```
def find_line(xs, ys):
    """Calculates the slope and intercept"""

    # number of points
    n = len(xs)

    # calculate means
    x_bar = sum(xs)/n
    y_bar = sum(ys)/n

    # calculate slope
    num = 0
    denom = 0
    for i in range(n):
        num += (xs[i]-x_bar)*(ys[i]-y_bar)
        denom += (xs[i]-x_bar)**2
    slope = num/denom

    # calculate intercept
    intercept = y_bar - slope*x_bar
```

Using this code, we can fit a line to our original data (see below). This is the maximum likelihood estimator for our data. The line minimizes the sum of squared errors, which is why this method of linear regression is often called ordinary least squares.



MLE solution to our Linear Regression model.

• • •

Final Thoughts

I wanted to write this post mostly to highlight the link between minimizing the sum of squared error and the maximum likelihood estimation approach to linear regression. Most people first learn to solve linear regression by minimizing the squared error, but it's not generally understood that this is derived from a probabilistic model with baked in assumptions (like Gaussian distributed errors).

There is a more elegant solution for finding the parameters of this model, but it requires linear algebra. I plan to revisit linear regression after covering linear algebra in

See you next time!

Machine Learning Statistics Maximum Likelihood Linear Regression William Fleshman