Least squares and maximum likelihood estimation

Jnaneshwar Das, jdas5@asu.edu

January 2026

1 Introduction

Autonomous exploration systems produce, consume, and process information corrupted by noise, and this information is typically represented as vectors and matrices. In the following two sections, we will explore least squares estimation and maximum likelihood estimation in the context of fitting a linear model to data corrupted by Gaussian noise. Through your assignments, you will learn how vectors and matrices are represented and manipulated using Python. When you work in simulation in ROS and Gazebo, and as you build and deploy your robots, the concepts discussed in this, and following material on linear dynamical systems will often be revisited.

2 Linear regression

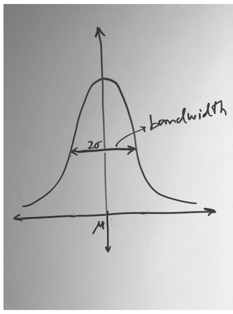

Figure 1: The normal or Gaussian distribution with mean of μ and variance of σ^2.

Linear regression is the process of fitting a line (or hyperplane) to data points. Data points are observations or measurements from a process, corrupted by noise from the environment. The normal or Gaussian distribution is a common choice to model the observation noise. It is given by,

$$\mathcal{N}(\mu, \sigma^2) \sim \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

where μ is the mean, σ^2 is the variance. These parameters are representative of noise in measurements, for example, a GPS sensor will typically have zero mean and a standard deviation σ of a few meters. These parameters are often assumed to be constant for a sensor, in reality however, they may drift with time, or change with operating temperature or other factors.

The equation of a line in 2-D is given by,

$$y = mx + c \tag{2}$$

where m is the slope of the line, and c is the intercept. The original input dimensionality is 1, however, we will rearrange equation 2 into a form with two input dimensions instead of one, with the first variable set to a constant we will call the bias term, representing the intercept. This step converts the equation of the line into a linear combination, a form that allows us to use linear algebra to estimate the slope and the intercept.

$$y = w_0 1 + w_1 x \tag{3}$$

We will set D=1 to keep the dimensionality of the original 1-D input unambiguous, and define two $D + 1 \times 1$ matrices representing the input and weight parameter vectors.

$$\mathbf{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} c \\ m \end{bmatrix} \tag{4}$$

.

Using the definitions above and with the notations $\mathbf{x} = [x_0 \quad x_1]^T$ and $\mathbf{w} = [w_0 \quad w_1]^T$, where $w_0 = c$ and $w_1 = m$, and $D = 1$, the equation of the line is written as a linear combination of inputs weighted by the weight vector parameters.

$$y = \sum_{i=0}^{D} w_i x_i \tag{5}$$

$$y = \mathbf{w}^T \mathbf{x} \tag{6}$$

When we observe from this line, the data points are corrupted by noise, hence the above equation can be written as.

$$y = \mathbf{w}^T \mathbf{x} + \epsilon \tag{7}$$

where ϵ is noise drawn from a Gaussian distribution defined in equation 1.

2.1 Fitting a line to measured data

Equipped with equation 7 defining observations from a line with noise, let us explore how to fit a line to the observed data. Figure 2 shows examples of data points drawn from the linear model represented by the line.

Data is acquired as observations drawn from a process that is represented by the line, for example, the relationship between electrical signal and environmental temperature for a temperature sensor. These measurements constitute

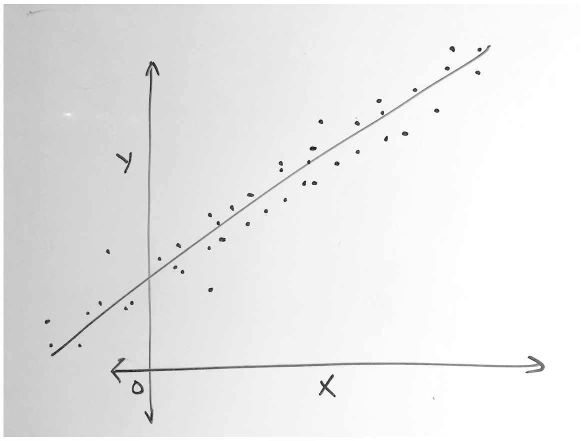

Figure 2: Data points drawn from a line $y = \mathbf{w}^T\mathbf{x}$, corrupted by Gaussian noise with mean of μ and variance of σ^2.

our training dataset consisting of an $N \times (D+1)$ input matrix X, and the $N \times 1$ output or observation matrix Y.

$$X = \begin{bmatrix} x_0^1 & x_1^1 \\ x_0^2 & x_1^2 \\ \vdots & \vdots \\ x_0^N & x_1^N \end{bmatrix} = \begin{bmatrix} 1 & x^1 \\ 1 & x^2 \\ \vdots & \vdots \\ 1 & x^N \end{bmatrix} \tag{8}$$

$$Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix} \tag{9}$$

A common choice to measure error when considering candidate values of slope and intercept (we will call these weights henceforth), is sum of square error. It is given by the loss (or error) L,

$$L = \Sigma_{i=1}^N (y^i - \mathbf{w}^T\mathbf{x}^i)^2 \tag{10}$$

The goal is to estimate the weight vector that minimizes the sum of squares loss.

$$\arg\min_{\mathbf{w}} \quad \sum_{i=1}^{N} (y^i - \mathbf{w}^T\mathbf{x}^i)^2 \tag{11}$$

The solution to this optimization problem to minimize the sum of squares loss function is given by the least squares method.

$$\mathbf{w} = (X^TX)^{-1}X^TY \tag{12}$$

Here, the operation $(X^TX)^{-1}X^T$ is called the Moore-Penrose inverse or pseudo-inverse of the matrix X, and the solution minimizes the sum of squares error defined in equation 10.

2.2 Probabilistic interpretation of linear regression

Let us revisit our training dataset composed of matrices X, Y, with the goal of estimating the weight vector w that minimizes the sum of squares error. We will consider a different perspective on the problem of estimating the best fit for the line defined in equation 7.

We start with the assumption that the data points are drawn i.i.d., or are independent and identically distributed. Next we define a metric *likelihood* of an observed datapoint y^i, for a choice of weight vector $\mathbf{w}^T$. This given by the conditional probability $p(y^i|\mathbf{x}^i, \mathbf{w}, \sigma^2)$, which is a Gaussian distribution $\sim \mathcal{N}(\mathbf{w}^T\mathbf{x}^i, \sigma^2)$. The likelihood of all observed data, represented using training matrices X and Y is computed by taking the product of the likelihood of the individual data points for a candidate weight vector.

$$\begin{aligned}\mathcal{L}(\mathbf{w}; X, Y) &= \prod_{i=1}^{N} p(y^i|\mathbf{x}^i, \mathbf{w}, \sigma^2) && (13)\\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-\frac{(y^i-\mathbf{w}^T\mathbf{x}^i)^2}{2\sigma^2}} && (14)\end{aligned}$$

The goal is to find the weight vector that maximizes the log likelihood. Without loss of generality, the likelihood of observations for a choice of weight parameters can be represented by taking the natural logarithm of the likelihood in equation 14.

$$\mathcal{L}_{log}(\mathbf{w}; X, Y) = \ln \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-\frac{(y^i-\mathbf{w}^T\mathbf{x}^i)^2}{2\sigma^2}} \tag{15}$$

Recall that ln a.b = ln a + ln b. Hence, the above equation reduces to the following.

$$\begin{aligned}\mathcal{L}_{log}(\mathbf{w}; X, Y) &= \sum_{i=1}^{N} \big(\ln \frac{1}{\sqrt{2\pi\sigma^2}} + \ln exp^{-\frac{(y^i-\mathbf{w}^T\mathbf{x}^i)^2}{2\sigma^2}} \big) && (16)\\ &= \sum_{i=1}^{N} \ln \frac{1}{\sqrt{2\pi\sigma^2}} + \sum_{i=1}^{N} \ln exp^{-\frac{(y^i-\mathbf{w}^T\mathbf{x}^i)^2}{2\sigma^2}} \big) && (17)\end{aligned}$$

The first term in equation 18 is a constant, we will call it C. The second term reduces to only the power of the exponential term. Upon rearranging we get.

$$\mathcal{L}_{log}(\mathbf{w}; X, Y) = C - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y^i - \mathbf{w}^T\mathbf{x}^i)^2 \tag{18}$$

The best estimate of weight vector, $\mathbf{w}^*$ is given by,

$$\begin{aligned}\mathbf{w}^* &= \arg\max_{\mathbf{w}} \quad \mathcal{L}_{log}(\mathbf{w}; X, Y) && (19)\\ &= \arg\max_{\mathbf{w}} \quad -\sum_{i=1}^{N}(y^i - \mathbf{w}^T\mathbf{x}^i)^2 && (20)\end{aligned}$$

The above optimization is equivalent to minimizing the negative log likelihood,

$$\arg\min_{\mathbf{w}} \quad \sum_{i=1}^{N}(y^i - \mathbf{w}^T\mathbf{x}^i)^2 \tag{21}$$

Note the similarity in the form of minimizing the negative log likelihood of observed data in the above equation, and the sum of squares error used earlier in equation 11.