Notes while understanding Maximum Likelihood Estimates

by

Harsh Nilesh Pathak

,

Abstract

Notes while understanding Maximum Likelihood Estimates

by

Harsh Nilesh Pathak

,

Understanding some basic concepts in organised fashion. This article is a compilation of many online articles, and purposed for self-study only and not for research. This article does not claim original content, we have just organised some content so that its easier to draw relations between various machine learning concepts.

Contents

Chapter 1

Maximum Likelihood Estimates

1.1 Bayes Theorem

Definition:

$$p(\theta|data) = \frac{p(data|\theta) \cdot p(\theta)}{p(data)} \tag{1.1}$$

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}$$

In Bayesian inference, we're concerned with the posterior - the probability of the parameters given the data. Put in another way, we're looking to estimate the probability distribution of the parameters (θ) given the data we have observed [10].

1.2 Introduction

In statistical inference we estimate the probability of parameters given a **parametric model**[1] and observed data drawn from it. MLE provides answer for the following question:

"For which parameter value does the observed data have the biggest probability?" [12]

$$L(\theta \mid data) = p(data \mid \theta) \tag{1.2}$$

Definition: Given data the maximum likelihood estimate (MLE) for the parameter (θ) is the value of $\hat{\theta}$ that maximizes the likelihood $p(data \mid \theta)$, here $\theta \in R^N$. In other words, the MLE is the value of θ for which the data is most likely.

$$p(data \mid \theta) \overset{i.i.d}{=} \prod_{i=1} p(d_i \mid \theta) \tag{1.3}$$

Defination of MLE:

$$\hat{\theta} = \underset{\theta}{argmax} \prod_{i=1} p(d_i \mid \theta) \tag{1.4}$$

Some simple examples are explained that can help it understanding further are here [12].

Assumption: Independent Identical distribution
What i.i.d. assumption states is that random variables are independent and identically distributed. Informally it says that all the variables provide the same kind of information independently of each other [2, 12, 4].

From the abstract ideas let's jump for a moment to concrete example: in most cases your data can be stored in a matrix, with observations row-wise and variables column-wise. If you assume your data to be i.i.d., then it means for you that you need to bother only about relations between columns and do not have to bother about relations between rows. If you bothered about both then you would model dependence of columns on columns and rows on rows, i.e. everything on everything. It is very hard to make simplifications and build a statistical model of everything depending on everything. [4]

1.3 Relation with Optimization

Maximum Likelihood Estimation is an optimization problem.

$$\begin{array}{ll} maximize & \prod_{i=1} p(d_i \mid \theta) \\ s.t. & \theta \in C \end{array} \tag{1.5}$$

if the density function is concave then this is a convex optimization problem such as Logistic Regression. Next, yes logarithms are easy to compute and also prevent underflow problem when dealing with joint probability distribution of larger datasets. Hence, equation 1.4 can be written as following:

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1} \log p(d_i \mid \theta) \tag{1.6}$$

And, now the optimization problem becomes

$$\begin{array}{ll} maximize & \sum_{i=1} \log p(d_i \mid \theta) \\ s.t. & \theta \in C \end{array} \tag{1.7}$$

1.4 Relation with Machine Learning

1.4.1 Supervised Learning

Recall MLE :

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1} \log p(d_i \mid \theta) \tag{1.8}$$

In supervised learning we have observed data in pairs i.e features - $\mathbf{x}$ and labels - $\mathbf{y}$. So, continuing further MLE for supervised learning can be seen as:

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1} \log p((y_i, x_i) \mid \theta) \tag{1.9}$$

Also, popularly known - Minimizing Negative Log Loss (NLL) is equivalent to MLE.

$$L = -\sum_{i=1} \log p((y_i, x_i) \mid \theta) \tag{1.10}$$

1.4.2 Unsupervised (Self-Supervised) Learning

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1} \log p(d_i \mid \theta) \tag{1.11}$$

Again recall MLE, Maximum Likelihood Estimation is a probabilistic framework for solving the problem of density estimation [5]. For example, given a sample of observation $\mathbf{x}$ from a domain, where each observation is drawn independently from the domain with the same probability distribution (so-called independent and identically distributed, i.i.d., or close to it) [5]. Density estimation is self-supervised learning where we solve MLE to find the set of parameters that best describes the distribution from where the observed data is coming from.

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1} \log p(x_i \mid \theta) \tag{1.12}$$

Fascinating talk on this from NIPS 2018 by Alex Graves [3].

Chapter 2

Maximum a posteriori (MAP) estimate

2.1 MLE a special case of MAP

MAP usually comes up in Bayesian setting. Because, as the name suggests, it works on a posterior distribution, not only the likelihood [8].

Recall Bayes Theorem:

$$p(\theta|data) = \frac{p(data|\theta) \cdot p(\theta)}{p(data)} \tag{2.1}$$

Definition:

$$\begin{aligned}
\hat{\theta}_{\text{MAP}} &= \arg\max_{\theta} p(\theta|data) \\
&= \arg\max_{\theta} \frac{p(data|\theta)p(\theta)}{p(data)} \\
&= \arg\max_{\theta} p(data|\theta)p(\theta) \\
&= \arg\max_{\theta} \log(p(data|\theta)p(\theta)) \\
&= \arg\max_{\theta} \log p(data|\theta) + \log p(\theta) \\
&= \hat{\theta}_{\text{MLE}} + \log p(\theta)
\end{aligned} \tag{2.2}$$

2.2 Uniform Distribution priors

If the prior distribution is uniform i.e values assigned to $\theta = \frac{1}{N}$ everywhere in the distribution. Here we show that MLE is a special case of MAP, where the prior is uniform [8].

$$\begin{aligned}
\hat{\theta}_{\text{MAP}} &= \arg\max_{\theta} p(data|\theta)p(\theta) \\
&= \arg\max_{\theta} \log(p(data|\theta)p(\theta)) \\
&= \arg\max_{\theta} \log p(data|\theta) + \log p(\theta) \\
&= \hat{\theta}_{\text{MLE}} + constant \\
&= \hat{\theta}_{\text{MLE}}
\end{aligned} \tag{2.3}$$

2.3 Normal Distribution priors or L2 regularization

Now let's take the case of Normal distribution Recall Normal Distribution:

$$Normal(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{2.4}$$

L2 regularization

$$\begin{aligned}
\hat{\theta}_{\text{MAP}} &= \arg\max_{\beta} \Big[\log \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-(\beta_0+\beta_1 x_{i,1}+\ldots+\beta_p x_{i,p}))^2}{2\sigma^2}} + \log \prod_{j=0}^{p} \frac{1}{\tau\sqrt{2\pi}} e^{-\frac{\beta_j^2}{2\tau^2}} \Big] \\
&= \arg\max_{\beta} \Big[-\sum_{i=1}^{n} \frac{(y_i - (\beta_0 + \beta_1 x_{i,1} + \ldots + \beta_p x_{i,p}))^2}{2\sigma^2} - \sum_{j=0}^{p} \frac{\beta_j^2}{2\tau^2} \Big] \\
&= \arg\min_{\beta} \frac{1}{2\sigma^2} \Big[\sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i,1} + \ldots + \beta_p x_{i,p}))^2 + \frac{\sigma^2}{\tau^2} \sum_{j=0}^{p} \beta_j^2 \Big] \\
&= \arg\min_{\beta} \Big[\sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i,1} + \ldots + \beta_p x_{i,p}))^2 + \lambda \sum_{j=0}^{p} \beta_j^2 \Big]
\end{aligned} \tag{2.5}$$

2.4 Laplacean priors or L1 regularization

Laplacean distribution:

$$Laplace(x \mid \mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}} \tag{2.6}$$

L1 regularization

$$
\begin{aligned}
\hat{\theta}_{\text{MAP}} &= \arg\max_{\beta} \Big[\log \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-(\beta_0+\beta_1 x_{i,1}+\ldots+\beta_p x_{i,p}))^2}{2\sigma^2}} + \log \prod_{j=0}^{p} \frac{1}{2b} e^{-\frac{|\beta_j|}{2b}} \Big] \\
&= \arg\max_{\beta} \Big[-\sum_{i=1}^{n} \frac{(y_i-(\beta_0+\beta_1 x_{i,1}+\ldots+\beta_p x_{i,p}))^2}{2\sigma^2} - \sum_{j=0}^{p} \frac{|\beta_j|}{2b} \Big] \\
&= \arg\min_{\beta} \frac{1}{2\sigma^2} \big[\sum_{i=1}^{n} (y_i-(\beta_0+\beta_1 x_{i,1}+\ldots+\beta_p x_{i,p}))^2 + \frac{\sigma^2}{b} \sum_{j=0}^{p} |\beta_j| \big] \\
&= \arg\min_{\beta} \big[\sum_{i=1}^{n} (y_i-(\beta_0+\beta_1 x_{i,1}+\ldots+\beta_p x_{i,p}))^2 + \lambda \sum_{j=0}^{p} |\beta_j| \big]
\end{aligned}
\tag{2.7}
$$

Chapter 3

Entropy

3.1 Information

How much useful information is being communicated?

Measure of information is given by:

$$I(x_i) = -\log(p(x_i)) \tag{3.1}$$

Example : If the events are equally likely (flip of coin or 50% chance that it will rain tomorrow, only $(-ln(2) = 1)$ 1 Bit of useful information is communicated.
If they are not equally likely then we need to measure what is the average message length that was communicated. Say you have 2 events that are not equally likely in that case

$$H(p(x)) = -p(x_i)\log(p(x_i)) - (1 - p(x_i))\log(1 - p(x_i)) \tag{3.2}$$

Now you can generalize it for multiple events

$$H(p(x)) = -\sum_i p(x_i)\log p(x_i) \tag{3.3}$$

$$H(p(x)) = \mathbb{E}_{x\sim p}\log p(x) \tag{3.4}$$

3.2 Cross Entropy

Cross entropy is the Avg msg length.

$$H(p(x), q(x)) = -\sum_i p(x_i)\log q(x_i) \tag{3.5}$$

Entropy is also the average message length if the events are equally likely. But you may have variable messge length per event and thus you can reduce the number of bits that can be communicated.

$$H(p,q) = -\sum_i p \log q \tag{3.6}$$

But now you have to predict the distribution, why predict? -BCZ you don't know the distribution of weather every day. Hence for prediction you have to do Maximum likelihood estimation

3.3 KL Divergence

Cross Entropy will be larger than entropy by KL divergence.

$$H(p,q) = H(p) + D_{KL}(p||q) \tag{3.7}$$

$$\begin{aligned} D_{KL}(p||q) &= \sum_{i=1}^{N} p(x_i) \log \frac{p(x_i)}{q(x_i)} \\ &= \sum_{i=1}^{N} p(x)[\log p(x_i) - \log q(x_i)] \end{aligned} \tag{3.8}$$

3.4 Entropy relation with MLE

Recall MLE

$$\hat{\theta} = \underset{\theta}{argmax} \sum_{i=1} \log p(d_i \mid \theta) \tag{3.9}$$

NLL and minimizing cross entropy is equivalent:

$$\begin{aligned} \hat{\theta} &= \arg\min_{\theta} -\sum_{i=1}^{N} \log q(x_i|\theta) \\ &= \arg\min_{\theta} -\sum_{x \in X} p(x) \log q(x|\theta) \\ &= \arg\min_{\theta} H(p,q) \end{aligned} \tag{3.10}$$

You can add p(x) as a coefficient as it does not changes the minima of optimization problem.

Chapter 4

Solving MLE

4.1 Taylor Series Approximation

It all starts from Taylor Series approximation which tells the geometry of any function at a given point (say a). If the Taylor series is centered at zero (a=0), then that series is also called a Maclaurin series.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(x-a)}{3!}x^3 + \cdots = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k \tag{4.1}$$

4.1.1 Why Least Square Error?

Any dataset that you are trying to fit is assumed to have some noise. The noise can be independent of our data, we call this noise additive. It is generally introduced by human errors when labelling and/or sensor inaccuracy [9].

$$Y = X\beta + Z \tag{4.2}$$

Z = Additive Gaussian Noise.

$$Normal(0 \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(z)^2/2\sigma^2} \tag{4.3}$$

Hence our objective is to minimize this noise using MLE.

$$\arg\max_{\beta} \sum_{i=1} \log p((y_i, x_i) \mid \beta) = \arg\min_{\beta} \left[\log \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-(\beta_0+\beta_1 x_{i,1}+\ldots+\beta_p x_{i,p}))^2}{2\sigma^2}} \right] \tag{4.4}$$

4.1.2 Method

Method could be simply explained with an example of Linear regression. We know if we assume the distribution of parameters is Normal then we MLE can be written as Mean Squared Error (MSE). The loss function can be shown by the following expression.

$$\sum_{i=1}^{N} \left(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j \right)^2 = 0 \tag{4.5}$$

Step 1: We then want to take partial derrivatives with respect to each component or parameter.

$$\frac{\partial}{\partial \beta_k} \sum_{i=1}^{N} \left(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j \right)^2 = 0 \tag{4.6}$$

Step 2: Now you have p (number of parameters = p) of these equations, one for each beta and equate it to zero. This is a simple application of the chain rule [7]:

$$-2 \sum_{i=1}^{N} X_{ik} \left(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j \right) = 0 \tag{4.7}$$

Now we can re-write the sum inside the bracket as $\sum_{j=1}^{p} X_{ij}\beta_j = x_i^T \boldsymbol{\beta}$ So you get:

$$\sum_{i=1}^{N} X_{ik} Y_i - \sum_{i=1}^{N} X_{ik} x_i^T \boldsymbol{\beta} = 0 \tag{4.8}$$

Now we have p of these equations, and we will "stack them" in a column vector. Notice how X_{ik} is the only term which depends on k, so we can stack this into the vector x_i and we get:

$$\sum_{i=1}^{N} x_i Y_i = \sum_{i=1}^{N} x_i x_i^T \boldsymbol{\beta} \tag{4.9}$$

Now we can take the beta outside the sum (but must stay on RHS of sum), and then take the inverse:

$$\left(\sum_{i=1}^{N} x_i x_i^T \right)^{-1} \sum_{i=1}^{N} x_i Y_i = \boldsymbol{\beta} \tag{4.10}$$

Why do we need iterative methods?, when analytical solution exists[6]

Even in the case of, say, linear models, where you have an analytical solution, it may still be best to use such an iterative solver. As an example, if we consider linear regression, the explicit solution requires inverting a matrix which has complexity $O(N^3)$. This becomes prohibitive in the context of big data. Also, a lot of problems in machine learning are convex, so using gradients ensure that we will get to the extrema. However, there are still relevant non-convex problems, like neural networks, where gradient methods (backpropagation) provide an efficient solver. Again this is specially relevant for the case of deep learning [6].

4.2 Maxima, Minima and Saddle point

Now we know how to formulate MLE and now we need to compute its maximum or minimum. Depending on the complexity of the parameter there could be one or more critical points. Critical points bascially could be either a maximum, minimum or saddle point.

Bibliography

[1] URL: https://en.wikipedia.org/wiki/Parametric_model.

[2] URL: https://blog.metaflow.fr/ml-notes-why-the-log-likelihood-24f7b6c40f83.

[3] URL: https://blog.metaflow.fr/ml-notes-why-the-log-likelihood-24f7b6c40f83.

[4] URL: https://stats.stackexchange.com/questions/213464/on-the-importance-of-the-i-i-d-assumption-in-statistical-learning.

[5] URL: https://machinelearningmastery.com/what-is-maximum-likelihood-estimation-in-machine-learning/.

[6] URL: https://stats.stackexchange.com/questions/212619/why-is-gradient-descent-required.

[7] URL: https://stats.stackexchange.com/questions/9801/analytical-solution-to-linear-regression-coefficient-estimates.

[8] URL: https://wiseodd.github.io/techblog/2017/01/01/mle-vs-map/.

[9] URL: https://towardsdatascience.com/ml-notes-why-the-least-square-error-bf27fdd9a721.

[10] Brian Keng. 2016. URL: http://bjlkeng.github.io/posts/probabilistic-interpretation-of-regularization/.

[11] Lj Miranda. 2017. URL: https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood/.

[12] Jeremy Orloff and Jonathan Bloom. 2014. URL: https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/.

[13] Reza Zadeh. 2016. URL: https://www.oreilly.com/ideas/the-hard-thing-about-deep-learning.