

### Maximum Likelihood Estimation (MLE) and Maximum a Posteriori (MAP)

Suppose you are given a dataset  $\mathbb{D} = \{X, Y\}$ , where

$$\begin{aligned} X &= [x^1 x^2 \dots x^N] \in \mathbb{R}^N \\ Y &= [y^1 y^2 \dots y^N] \in \mathbb{R}^N \end{aligned}$$

and  $x^i \in \mathbb{R} \forall i = 1, \dots, N$ ,  $y^i \in \mathbb{R}$ ,  $\forall i = 1, \dots, N$ . Moreover, suppose that:

$$y^i = \theta_1 + \theta_2 x^i + \dots + \theta_K (x^i)^{K-1} + e^i \quad \forall i = 1, \dots, N \quad (1)$$

Where  $e^i \sim \mathcal{N}(0; \sigma^2)$  is random Gaussian Noise and  $\theta = (\theta_1, \dots, \theta_n)^T$ . It is known that the Maximum Likelihood Estimation (MLE) approach works by defining the conditional probability of  $y$  given  $x$ ,  $p_\theta(y|x)$  and then optimizes the parameters  $\theta$  to maximize this probability distribution over  $\mathbb{D}$ . Moreover, it is also known that this approach can be made equivalent to the deterministic approach to solve such problems (the Least Square method) by taking the negative-log of  $p_\theta(y|x)$ . Indeed, by assuming that the noise  $e^i$  is Gaussian for each  $i = 1, \dots, N$ , we have

$$p_\theta(y|x) = \mathcal{N}(f_\theta(x^i), \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(f_\theta(x^i) - y)^2}$$

Thus

$$p_\theta(Y|X) = \prod_{i=1}^N p_\theta(y^i|x^i) \implies$$

$$\theta_{MLE} = \arg \max_{\theta \in \mathbb{R}^s} p_\theta(Y|X) = \arg \min_{\theta \in \mathbb{R}^s} -\log p_\theta(Y|X) = \arg \min_{\theta \in \mathbb{R}^s} -\log \prod_{i=1}^N p_\theta(y^i|x^i) = \arg \min_{\theta \in \mathbb{R}^s} \sum_{i=1}^N -\log p_\theta(y^i|x^i) =$$

$$\arg \min_{\theta \in \mathbb{R}^s} \sum_{i=1}^N \frac{1}{2\sigma^2} (f_\theta(x^i) - y^i)^2 = \arg \min_{\theta \in \mathbb{R}^s} \frac{1}{2\sigma^2} \|\mathbf{f}_\theta - \mathbf{y}\|_2^2$$

If  $f_\theta(x) = \theta_1 + \theta_2 x^i + \dots + \theta_K (x^i)^{K-1} = \sum_{j=1}^K \phi_j(x^i) \theta_j$ , with  $\phi_j(x) = x^{j-1}$ , we already shown that the problem above becomes

$$\theta_{MLE} = \arg \min_{\theta \in \mathbb{R}^s} \frac{1}{2\sigma^2} (\Psi(X)\theta - \mathbf{y})^2$$

where  $\Psi(X)$  is the  $N \times K$  matrix that contains the powers from 0 to  $K-1$  of the input datapoints  $x^i$  as rows (the Vandermonde matrix associated with  $(x^1, \dots, x^N)^T$ ),  $\theta = (\theta_1, \dots, \theta_n)^T$  and  $Y = (y^1, \dots, y^N)^T$ .

Note that the above equation is equivalent to the function you optimized in the Exercise 3 of Lab3 with GD, with  $A := \Phi(X)$ ,  $x := \theta$  and  $b := y$ .

When it is unclear how to set the parameter  $K$  and it is impossible to use the error plot, it is required to use the Maximum A Posterior (MAP) approach.

To show how it works, suppose that we know that the parameters are normally distributed  $\theta \sim \mathcal{N}(0; \sigma_\theta^2 I)$ . Then we can use the Bayes Theorem to express the A Posteriori probability on  $y$  given  $x$  and  $\theta$  as

$$p(\theta|X, Y) = \frac{p(Y|X, \theta)p(\theta)}{p(Y|X)}$$

The MAP solution searches for a set of parameters  $\theta$  that maximizes  $p(\theta|X, Y)$ . Following the same reasoning as before,

$$\theta_{MAP} = \arg \max_{\theta} p(\theta|X, Y) = \arg \min_{\theta} \sum_{i=1}^N -\log p(\theta|x^i, y^i) = \arg \min_{\theta} \sum_{i=1}^N -\log p(y^i|x^i, \theta) - \log p(\theta)$$

Given the two optimization problem above, you are required to implement a program that compare the two solutions, that we will refer to as  $\theta_{MLE}$  and  $\theta_{MAP}$ .

1. Define a test problem in the following way:

- Let the user fix a positive integer  $K > 0$ , and define  $\theta_{true} = (1, 1, \dots, 1)^T$  (you can also consider different  $\theta_{true}$ );
- Define an input dataset

$$X = [x^1 x^2 \dots x^N] \in \mathbb{R}^N$$

where the  $x^i$  are  $N$  uniformly distributed datapoints in the interval  $[a, b]$ , where  $a < b$  are value that the user can select;

- Given a set of functions  $\{\phi_1, \phi_2, \dots, \phi_K\}$ , define the Generalized Vandermonde matrix  $\Phi(X) \in \mathbb{R}^{N \times K}$ , whose element in position  $i, j$  is  $\phi_j(x^i)$ . In particular, write a function defining the classical Vandermonde matrix where  $\phi_j(x) = x^{j-1}$ ;
- Given a variance  $\sigma^2 > 0$  defined by the user, compute

$$Y = \Phi(X)\theta_{true} + e$$

where  $e \sim \mathcal{N}(0, \sigma^2 I)$  is Gaussian distributed noise with variance  $\sigma^2$ . Note that the test problem defined in this way is very similar to what we did to define a test problem in the first Lab.

2. We now built a dataset  $\mathbb{D} = \{(X, Y)\}$  such that  $\theta_{true} = (1, 1, \dots, 1)^T \in \mathbb{R}^K$  is the **best** solution to the least squares problem

$$\Phi(X)\theta \approx Y$$

3. Pretend not to know the correct value of  $K$ . The first task is to try to guess it and use it to approximate the true solution  $\theta_{true}$  by MLE and MAP. To do that:

- Write a function that takes as input the training data  $\mathbb{D} = (X, Y)$  and  $K$  and returns the MLE solution (with Gaussian assumption)  $\theta_{MLE} \in \mathbb{R}^K$  for that problem. Note that the loss function can be optimized by GD, SGD or Normal Equations.
- Write a function that takes as input a set of  $K$ -dimensional parameter vector  $\theta$  and a test set  $\mathcal{TE} = \{(X_{test}, Y_{test})\}$  and returns the average absolute error of the polynomial regressor  $f_\theta(x)$  over  $X_{test}$ , computed as:

$$\frac{1}{N_{test}} \|f_\theta(X_{test}) - Y_{test}\|_2^2$$

- For different values of  $K$ , plot the training datapoints and the test datapoints with different colors, and visualize (as a continuous line) the learnt regression model  $f_{\theta_{MLE}}(x)$ . Comment the results.
- For increasing values of  $K$ , use the functions defined above to compute the training and test error, where the test set is generated by sampling  $N_{test}$  new points on the same interval  $[a, b]$  of the training set and generating the corresponding  $Y_{test}$  with the same procedure of the training set. Plot the two errors with respect to  $K$ . Comment the results.
- Write a function that takes as input the training data  $\mathbb{D} = (X, Y)$ ,  $K$  and  $\lambda > 0$  and returns the MAP solution (with Gaussian assumption)  $\theta_{MAP} \in \mathbb{R}^K$  for that problem. Note that the loss function can be optimized by GD, SGD or Normal Equations.
- For  $K$  lower, equal and greater than the correct degree of the test polynomial, plot the training datapoints and the test datapoints with different colors, and visualize (as a continuous line) the learnt regression model  $f_{\theta_{MAP}}(x)$  with different values of  $\lambda$ . Comment the results.
- For  $K$  being way greater than the correct degree of the polynomial, compute the MLE and MAP solution. Compare the test error of the two, for different values of  $\lambda$  (in the case of MAP).
- For  $K$  greater than the true degree of the polynomial, define  $Err(\theta) = \frac{\|\theta - \theta_{true}\|_2}{\|\theta_{true}\|_2}$  where  $\theta_{true}$  has been padded with zeros to match the shape of  $\theta$ . Compute  $Err(\theta_{MLE})$  and  $Err(\theta_{MAP})$  for increasing values of  $K$  and different values of  $\lambda$ .
- Compare the results obtained by increasing the number  $N$  of datapoints.
- Compare the results obtained by the three algorithms GD, SGD and Normal Equations.

4. Repeat the first point of the homework by setting

$$y^i \sim Poi(y|\theta_{true,1} + \theta_{true,2}x^i + \dots + \theta_{true,K}(x^i)^{K-1}) \quad \forall i = 1, \dots, N \quad (2)$$

Where  $Poi(y|\lambda)$  is the Poisson distribution with parameter  $\lambda$  (you can find the Python function to do that), such that

$$Poi(y|\lambda) = \frac{\lambda^y e^{-\lambda}}{y!}$$

And consequently,

$$p_{\theta}(y|x) = Poi(y|\theta_1 + \theta_2 x^i + \dots + \theta_K (x^i)^{K-1})$$

5. Compute the MLE and MAP losses for this choice of  $p_{\theta}(y|x)$  and repeat the experiments in the previous points.