[AIMLBD] MACHINE LEARNING, BIG DATA, ARTIFICIAL INTELLIGENCE per medicina e chirurgia high tech

L03: Maximum Likelihood Estimation

Dott. Giorgio De Magistris

demagistris@diag.uniroma1.it

Corso di Laurea in Medicina e Chirurgia High Tech

Sapienza Università di Roma

I3S Facoltà di Ingegneria dell'Informazione, Informatica e Statistica

DIAG Dipartimento di Ingegneria Informatica, Automatica e Gestionale

Loss Functions

- Previously we saw that in general a ML model is trained to minimize an error function (also called **loss** function)
- But how do these loss functions came from? How can we derive the right loss function for the specific problem?
- We can answer by giving a probabilistic interpretation to the question

Likelihood

- Given the training data $X = \{x_1, \dots, x_m\}$ sampled from the data probability distribution $x_i \sim p_{data}(x)$

- Given a parametric model $p_{model}(x ; \theta)$

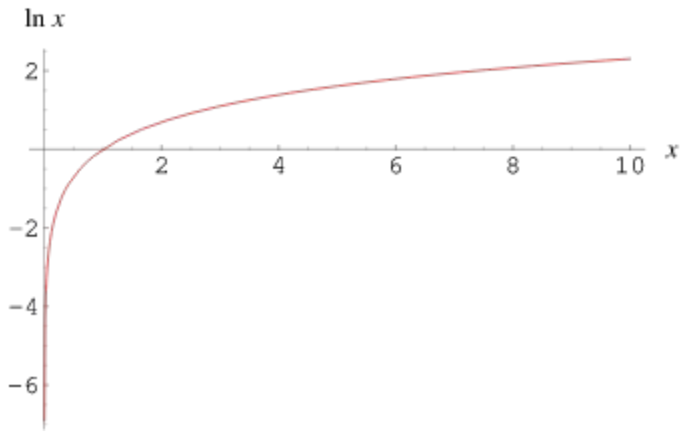
- What are the values of the parameters that maximize p_{model} when the x values come from the distribution p_{data} ?

$$\Theta_{ML} = \underset{\Theta}{\text{argmax}}\, p_{\text{model}}(X;\Theta)$$

- That, tanks to the i.i.d. assumption (the samples in the training set are independent) can be reformulated as

$$\Theta_{ML} = \underset{\Theta}{\text{argmax}} \prod_{i=1}^{m} p_{\text{model}}(x_i;\Theta)$$

Maximum Likelihood Estimation



- The values of the parameters that maximize the likelihood θML are called Maximum Likelihood Estimation

$$\Theta_{ML} = \underset{\Theta}{\text{argmax}} \prod_{i=1}^{m} p_{\text{model}}(x_i; \Theta)$$

- Sometimes we use to maximize the log (logarithm with base e) likelihood function instead of the likelihood

$$\Theta_{ML} = \underset{\Theta}{\text{argmax}} \sum_{i=1}^{m} log(p_{\text{model}}(x_i; \Theta))$$

- The logarithm is a monotonically increasing function so the argmax does not change (even though the max changes)

Remember

log(ab) = log(a) + log(b)
log(a/b) = log(a) - log(b)
log(a^c) = c log(a)

MLE for Supervised Learning

- We saw how to estimate the maximum likelihood parameters when a generic parametric probability distribution p_{model} tries to approximate the real data distribution p_{data}

- We can specialize MLE for different probability distributions, for example in supervised learning we often want to predict the targets Y given the inputs X

- In this case the MLE becomes

$$\Theta_{ML} = \underset{\Theta}{\operatorname{argmax}} \, p_{model}(Y|X;\Theta)$$

MLE for Regression

- Assuming that p_{model} is a parametric normal distribution:

$$p_{model}(y_i|x_i;\Theta) = N(\mu = f(x_i,\Theta), \sigma^2 = \text{const})$$

- We obtain the following Maximum Likelihood Estimation

$$\Theta_{ML} = \underset{\Theta}{argmax} \sum_{i=1}^{m} log(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(y_i - f(x_i,\Theta))^2}{\sigma^2}})$$

- And after further simplifications and by ignoring the terms that do not depend on θ, we obtain the well known **Mean Squared Error**

$$\Theta_{ML} = \underset{\Theta}{argmin} \frac{1}{m} \sum_{i=1}^{m} (y_i - f(x_i,\Theta))^2$$

MLE for binary classification

- For binary classification we assume that the distribution p_{model} is a parametric Bernoulli distribution

$$p_{model}(y_i|x_i;\Theta) = \begin{cases} p_i & if \quad y_i = 1 \\ 1 - p_i & if \quad y_i = 0 \end{cases} = p_i^{y_i}(1-p_i)^{(1-y_i)} \qquad p_i = \frac{1}{1+e^{-f(x_i,\Theta)}} = \sigma(f(x_i,\Theta))$$

- We obtain the following the following MLE:

$$\Theta_{ML} = \underset{\Theta}{argmax} \sum_{i=1}^{m} log(\sigma(f(x_i,\Theta))^{y_i}(1-\sigma(f(x_i,\Theta))^{(1-y_i)})$$

- That, after switching the sign and turning the argmax into an argmin, becomes the well known binary cross entropy loss

$$\Theta_{ML} = \underset{\Theta}{argmin} -\frac{1}{m}\sum_{i=1}^{m} y_i log(\sigma(f(x_i,\Theta))) + (1-y_i)log(1-\sigma(f(x_i,\Theta)))$$

MLE for multi-class classification

- Similarly for a k classes classification problem we assume that the distribution p_{model} is a multinoulli distribution (also known as categorical distribution)

$$p_{model}(y_i|x_i;\Theta_i) = \begin{cases} p_1 & if \quad y_i = 1 \\ \dots & \\ p_k & if \quad y_i = k \end{cases} = \prod_{j=1}^{k} p_j^{\delta_j(y_i)} \qquad p_j = \frac{e^{f(x_j;\Theta)}}{\sum_{i=1}^{k} e^{f(x_i;\Theta)}} = \hat{y}_j$$

with $\delta_j(y_i) = \begin{cases} 1 & \text{if} \quad y_i = j \\ 0 & \text{otherwise} \end{cases}$

- And by maximizing the log likelihood we obtain the **categorical cross entropy loss**

$$\Theta_{ML} = \underset{\Theta}{\text{argmax}} \sum_{i=1}^{m} log(\prod_{j=1}^{k} p_j^{\delta_j(y_i)}) = \underset{\Theta}{\text{argmax}} \sum_{i=1}^{m} \sum_{j=1}^{k} \delta_j(y_i) log(p_j)$$

$$= \underset{\Theta}{\text{argmin}} - \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} \delta_j(y_i) log(\hat{y}_j)$$