03 Bayesian Linear Regression

June 25, 2021

Information: *Basic concepts and simple examples of Bayesian linear regression*

Written by: *Zihao Xu*

Last update date: *06.25.2021*

1 Maximum Likelihood Estimation

1.1 Motivation

In the chapter talking about ***Generalization and Regularization***, the concepts of parameter estimation, bias and variance are used to formally characterize notions of generalization, underfitting and overfitting. Here are some important remarks.

- View the parameter estimator $\hat{\boldsymbol{\theta}}$ as a **function** of the sampled training dataset

$$\hat{\boldsymbol{\theta}} = g\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(m)}\right)$$

- The datasets (training, testing and probably validation) are generated by a **i.i.d.** probability distribution over datasets called the **data-generating process** (i.i.d. assumptions can be applied to almost all the common tasks)
- Assume that the true parameter value $\boldsymbol{\theta}$ is fixed but unknown
- Since the **data** is drawn from a **random process**, any function of the data is random, which means the parameter estimator $\hat{\boldsymbol{\theta}}$ is a **random variable**

The concepts of **bias** and **variance** are used to measure the performance of a parameter estimator. However, **for obtaining a good estimator**, it's not a good idea to guess that some function might make a good estimator and then to analyze its bias and variance. This motivated some principles from which specific functions that are good estimators for different models can be derived.

1.2 Definition

- Consider a set of m examples $\mathcal{D} = \left\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(m)}\right\}$ drawn independently from the true but unknown data-generating distribution $p_{\text{data}}(\mathbf{x})$. Let $p_{\text{model}}(\mathbf{x}|\boldsymbol{\theta})$ be a parametric family of probability distributions over the same space indexed by $\boldsymbol{\theta}$
  - That is to say, p_{model} maps any configuration $\mathbf{x}$ to a real number estimating the true probability $p_{\text{data}}(\mathbf{x})$

- Particularly, focus on the **likelihood** which is first introduced in the prerequisite chapter ***Probability Theory***

$$\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(m)} \mid \boldsymbol{\theta} \sim p_{\text{model}}\left(\mathbf{x}^{(1:m)} \mid \boldsymbol{\theta}\right)$$

  As a fast review, the likelihood tells us how *plausible* it is to observe $\mathbf{x}^{(1:m)}$ if we know the model parameters are θ

- Since the examples are assumed to be drawn **independently**, the likelihood can be factorized

$$p_{\text{model}}\left(\mathbf{x}^{(1:m)} \mid \boldsymbol{\theta}\right) = \prod_{i=1}^{m} p_{\text{model}}\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)$$

  Then **maximum likelihood** estimator for $\boldsymbol{\theta}$ is then defined as

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)$$

- While this simple production may lead to a lot of inconveniences such as **numerical underflow**, taking the **logarithm** of the likelihood does not change the location for maximum ($\arg\max_{\boldsymbol{\theta}}$) but does conveniently transform a product into a sum

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\text{model}}\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)$$

- Obviously, rescaling the likelihood does not change the location for maximum ($\arg\max_{\boldsymbol{\theta}}$), we can divide by m to obtain a version of the criterion that is expressed as an expectation with respect to the empirical distribution $\hat{p}_{\text{data}}$ defined by the training data

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} \left[\log p_{\text{model}}\left(\mathbf{x} \mid \boldsymbol{\theta}\right)\right]$$

1.3 KL divergence

- Maximum likelihood estimation can be viewed as minimizing the dissimilarity between the empirical distribution $\hat{p}_{\text{data}}$, defined by the training set and the model distribution, with the degree of dissimilarity between the two measure by the **KL divergence**

$$D_{\text{KL}}\left(\hat{p}_{\text{data}} \| p_{\text{model}}\right) = \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} \left[\log \hat{p}_{\text{data}}\left(\mathbf{x}\right) - \log p_{\text{model}}\left(\mathbf{x} \mid \boldsymbol{\theta}\right)\right]$$

  The term on the left is a function only of the data-generating process, not the model. This means when we train the model to minimize the KL divergence, we need only minimize

$$-\mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} \left[\log p_{\text{model}}\left(\mathbf{x} \mid \boldsymbol{\theta}\right)\right]$$

- Minimizing this KL divergence corresponds exactly to minimizing the cross-entropy between the distributions. By definition, any loss consisting a negative log-likelihood is a **cross-entropy** between the **empirical distribution** defined by the training set ($\hat{p}_{\text{data}}$), and the **probability distribution** defined by the model (p_{model})

1.4 Least squares

- Least squares minimizing the mean square error is **equal** to maximum likelihood estimation when the likelihood is assigned to be **Gaussian**
- Assume the model is $\hat{y} = f(\mathbf{x}; \boldsymbol{\theta})$ with the dataset

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \cdots & \mathbf{x}^{(m)} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y^{(1)} & y^{(2)} & \cdots & y^{(m)} \end{bmatrix}$$

  The solution for $\boldsymbol{\theta}$ via least squares would be

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - f(\mathbf{X}; \boldsymbol{\theta})\|_2^2$$

1.5 Supervised Learning

- The mathematical representations for applying the concept of maximum likelihood estimation to supervised learning tasks are quite similar but worth listing. Consider a set of m examples $\mathcal{D} = \left\{ \left(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}\right), \left(\mathbf{x}^{(2)}, \mathbf{y}^{(2)}\right), \cdots, \left(\mathbf{x}^{(m)}, \mathbf{y}^{(m)}\right) \right\}$ drawn independently from the true but unknown data-generating distribution $p_{\text{data}}(\mathbf{x}, \mathbf{y})$
- In supervised learning, factorize the data-generating process

$$p_{\text{data}}(\mathbf{x}, \mathbf{y}) = p_{\text{data}}(\mathbf{y} \mid \mathbf{x}) \, p_{\text{data}}(\mathbf{x})$$

  Let $p_{\text{model}}(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta})$ be a parametric family of probability distributions over the same space indexed by $\boldsymbol{\theta}$. It also can be factorized

$$p_{\text{model}}(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) = p_{\text{model}}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) \, p_{\text{data}}(\mathbf{x})$$

- Notice that the later part $p_{\text{data}}(\mathbf{x})$ is fixed and not controlled by the parameter $\boldsymbol{\theta}$, the maximum likelihood estimation is going to focus on

$$p_{\text{model}}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$$

  The maximum likelihood estimator is

$$\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}\left(\mathbf{y}^{(i)} \,\middle|\, \mathbf{x}^{(i)}, \boldsymbol{\theta}\right)$$

- Similarly, this optimization problem is usually converted into a minimization problem by the **negative logarithm** operation