Machine Learning Cheatsheet

Edoardo Riggio

June 4, 2022

Machine Learning - SA. 2022
Computer Science
Università della Svizzera Italiana, Lugano

Contents

1 Introduction

What does it mean "to learn"? We have to different definitions, one from a **statistical perspective**, and one from a **computer science perspective**.

- **Statistical Perspective**

  Vast amounts of data are being generated in many fields. The statistician's job is to make sense of all of this data, extract meaningful patterns and trends, and understand "what the data says". This approach is also known as **learning from data**.

- **Computer Science Perspective**

  The field of machine learning is concerned with how to construct computer programs that automatically improve with experience.

1.1 Mitchell's Formalisation

A computer program is said to learn from **experience** E – concerning some class of **task** T, and **performance measurement** P – if its performance at task T, as measured by P, improves with experience E.

1.2 Models

1.2.1 White Box Model

In this case, both physical laws and structural parameters of the problem are known. A family equation can be derived.

1.2.2 Grey Box Model

The physical laws are known in this case, and at least one parameter is unknown. A family of equations can be derived, but the parameters need to be identified.

1.2.3 Black Box Model

In this case, the physical laws are unknown. A family of equations cannot be derived.

1.3 Measures and Measurements

The operation of measuring an unknown quantity x_0 can be modeled as taking an instance – i.e., a **measurement** – x_i at time i, with an ad-hoc sensor S.

Although S has been suitably designed and realized, the physical elements that compose it are far from ideal and introduce uncertainties in the measurement process. As a result, x_i only represents an estimate of x_0.

1.4 Types of Models

1.4.1 Additive Model

The measurement process can be modeled as:

$$x = x_0 + \eta \qquad \text{where } \eta = f_n(0, \sigma_\eta^2)$$

Where η is an independent and identically distributed random variable, the model assumes that the i.i.d. noise does not depend on the working point x_0.

1.5 Multiplicative Model

The measurement process can be modeled as:

$$x = x_0(1 + \eta) \qquad \text{where } \eta = f_n(0, \sigma_\eta^2)$$

Where η is an independent and identically distributed random variable, the noise, in this case, depends on the working point x_0. In absolute terms, the impact of the noise on the signal is $x_0\eta$, but the relative contribution is η – which does not depend on x_0.

1.6 Supervised Learning

In a supervised learning framework, we have the following elements: a **concept to learn**, a **teacher**, and a **student**.

1.6.1 Regression

The goal of regression is to determine the function that explains the given instances – **measuremets**. The student proposes a family of models $f(\theta, x)$, and after a learning procedure, the "best" model $f(\hat{\theta}, x)$ is found.

1.6.2 Classification

The goal of classification is to determine the function – **model** – that partitions the input – **measurements** – into classes. The student proposes a family of models $f(\theta, x)$, and after a learning procedure, the "best" model $f(\hat{\theta}, x)$ is found.

1.6.3 Prediction

The goal of prediction is to tell us which data – **measurements** – will come next, possibly along with a confidence level. The student proposes a family of models $f(\theta, x)$, and after a learning procedure, the "best" model $f(\hat{\theta}, x)$ is found.

1.7 Features

We might want to extract features from the measurements to ease the learning task. The features must:

- Provide a compact representation of inputs
- Be particularly advantageous if we have prior information to take advantage of
- Be reduced to a minimal set before processing them for task solving

1.8 Unsupervised Learning

The goal of unsupervised learning is to build a representation of data. During its operational life, given an input, the machine provides information that can be used for decision-making.

2 Linear Regression

To prepare a linear regression model, several techniques can be used. The most popular being the **Least Mean Square (LMS)**, and the **Gradient Descent**.

Linear models are good when we have the following conditions:

- The data is generated by a linear model
- The dataset is small
- The data is sparse
- The uncertainty is high

2.1 Muliple Linear Regression

We are given a set of points:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \qquad \text{where } x \in \mathbb{R}^d, y \in \mathbb{R}$$

This is known as the **training set**. We now assume that the unknown function that generates the data is linear. Moreover, we assume that there is a gaussian uncertainty affecting the measurements. The model is given as:

$$y(x) = \theta_1^0 + \theta_2^0 z_2 + \cdots + \theta_d^0 z_d \quad \text{where } \theta^0 \in \mathbb{R}, \eta = N(0, \sigma_\eta^2)$$

Which can be written in its canonical form:

$$y(x) = x^T \theta^0 + \eta \quad \text{where } x^T = \begin{bmatrix} 1 & z_2 & \cdots & z_d \end{bmatrix}, {\theta^0}^T = \begin{bmatrix} \theta_1^0 & \theta_2^0 & \cdots & \theta_d^0 \end{bmatrix}$$

Both the optimal parameters and the variance of the noise are unknown. Since we know that the system model is linear, the family of models which best fits the data is:

$$\hat{y}(x) = f(\theta, x) = x^T\theta$$

In order to determine the best parameters, we estimate them:

$$f(\hat{\theta}, x) = x^T\hat{\theta}$$

2.1.1 Least Mean Square

The main idea of this procedure is to select the linear function that minimizes the average distance between the given points and the linear function.

The **performance function** of this method is the following:

$$V_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y(x_i) - f(\theta, x_i))^2$$

Therefore, the parameter vector $\hat{\theta}$ minimizing the performance function is the following:

$$\hat{\theta} = \arg\min_{\theta\in\Theta} V_n(\theta)$$

2.1.2 Parameter Estimation

By grouping the data into vectors X – of all x values – and Y – of all y values, we can rewrite the formula above in its canonical form:

$$\begin{aligned} V_n^*(\theta) &= \sum_{i=1}^{n}(y(x_i) - x_i^T\theta)^2 \\ &= (Y - X\theta)^T(Y - X\theta) \end{aligned}$$

Stationary points are those for which:

$$\frac{\partial V_n^*(\theta)}{\partial\theta} = -2X^TY + 2X^TX\theta = 0$$

Therefore, the parameter vector $\hat{\theta}$ minimizing the performance function is the following:

$$\hat{\theta} = (X_TX)^{-1}X^TY$$

And the best approximating model is:

$$f(\hat{\theta}x) = x^T\hat{\theta}$$

2.2 Performance at Task

In order to test the model, we need to use another set of unseen data. This is because the **performance at task**:

$$V_n(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(y(x_i) - f(\theta, x_i))^2$$

It is biased to the training set. For this reason, we consider another set of data – different from the training set – and call it **test set**:

$$\{(\bar{x}_1, \bar{y}_1), (\bar{x}_2, \bar{y}_2), \ldots, (\bar{x}_l, \bar{y}_l)\}$$

And evaluate the performance at task on it:

$$V_l(\hat{\theta}) = \frac{1}{l}\sum_{i=1}^{l}(\bar{y}_i - \bar{x}^T\hat{\theta})^2$$

2.3 Cross-Validation

Cross-validation provides a means to assess the performance of a model. This method can also be considered for model selection – with some care. In the latter case, we consider the model that minimizes

$$V_l(\hat{\theta}) = \frac{1}{l}\sum_{i=1}^{l}(\bar{y}_i - \bar{x}^T\hat{\theta})^2$$

2.4 Properties

Under the linear framework presented earlier, it can be proved that both implications hold true:

$$\lim_{n\to\infty} \hat{\theta} = \theta^0$$

$$\lim_{l\to\infty} V_l(\hat{\theta}) = \sigma_\eta^2$$

In addition, assuming that n training couples are given, then it can be proved that:

$$Var(\hat{\theta}) = (X^TX)^{-1}\sigma_\eta^2 \qquad \text{with } \hat{\sigma}_\eta^2 = \frac{1}{n-d}\sum_{i=1}^{n}(y(x_i) - f(\hat{\theta}, x_i))^2$$

If a parameter is smaller than twice its standard deviation, it must be set to 0. After which, we re-evaluate the performance and decide whether to keep it. This is known as **Occam's razor strategy**.

2.5 Ridge Regression

Ridge regression aims at pushing as many parameters as possible towards zero. This is done by adding a shrinking penalty to the loss function. The **MSE** training performance mesure

$$V_n(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\theta)^2$$

Now becomes

$$V_{Ridge}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\theta)^2 + \lambda\sum_{i=2}^{d}\theta_i^2$$

Where λ is a hyperparameter weighting the two contributions. A smaller λ gives more importance to accuracy, while a high λ priviledges a smaller number of parameters in the model. A tradeoff can be obtained by estimating and appropriate λ thanks to the **validation set**.

2.6 Lasso Regression

Lasso regression also aims at pushing as many parameters as possible towards zero. In this case, however, we penalize the parameter itself rather than its square. The **MSE** now is:

$$V_{Lasso}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\theta)^2 + \lambda\sum_{i=2}^{d}|\theta_i|$$

The optimization problem is not convex anymore, and the loss function is not differentiable. This makes the problem computationally complex and solvable only via optimization techniques such as quadratic programming.

2.7 Final Prediction Error

The expected prediction error is computed as follows:

$$FPE = \frac{n+d}{n-d}\sum_{i=1}^{n}(y(x_i) - f(\hat{\theta}, x_i))^2$$

This measurement can be used for both model selection – only on hierarchical family of models, and for assessing the performance of unseen data.