Statistical methods for machine learning

Francesco Tomaselli

March 1, 2021

Contents

1 Introduction

This first section introduces the concept of a machine learning task, with a particular focus on supervised classification.

1.1 Machine learning tasks and paradigms

Machine learning task Typically a machine learning task falls in the three categories below.

1. *Clustering*: group data according to similarity, e.g. group customers by shopping habits
2. *Classification*: predict semantic labels associated with data points, for instance document classification in relation to topics
3. *Planning*: decide which set of actions to be performed to achieve a certain goal, e.g. self driving cars

When it comes to machine learning there are mainly two learning paradigms.

Supervised learning This type of learning relies on semantic tagging of data. This usually solves classification tasks, as we can assign a label to each data point and learn patterns to classify new data.

Unsupervised learning There is no semantic tagging associated with data, this can for instance solve a clustering problem, as the algorithm will consider a form of similarity between data points to cluster them. Similarity can be interpreted as a semantic feature of data, but no explicit label is given.

1.2 Supervised classification

The main goal of this learning task is to learn a rule that maps data points to a certain label.

1.2.1 Labels

Along with the definition of a learning problem, there is a label set Y that collects possible labels of data.
For instance, when considering classification of documents, a label set could be defined as follows:

$$Y = \{sport, politics, business, \dots\}$$

Or in the case of stock predictions:

$$Y \subseteq \mathbb{R}$$

Labels definition The definition of labels changes the flavour of the task, in fact:

- if Y contains a finite number of symbols the task is called *classification/categorization*
- if Y is formed by real numbers, the task is called *regression*

Error measures The computation of prediction error differs between classification and regression: in the first we can consider an error if the prediction differs from the label, in the latter we can compute the difference between the prediction and the label.

1.2.2 Loss functions

When learning a map from data to labels, we need a way to tell the machine how good the mapping is, so a loss function is defined on the pair of true label and assigned label.

Classification loss examples In case of classification, we can define:

$$l(y, \hat{y}) = \begin{cases} 0 \ \mathit{if} \ y = \hat{y} \\ 1 \ \mathit{if} \ y \neq \hat{y} \end{cases}$$

It is possible to be a little more precise in the definition of a loss function, for instance:

$$Y = \{spam, notspam\}$$

$$l(y, \hat{y}) = \begin{cases} 2 \ \mathit{if} \ y = notspam \wedge \hat{y} = spam \\ 1 \ \mathit{if} \ y = spam \wedge \hat{y} = notspam \\ 0 \ otherwise \end{cases}$$

The idea is to penalize false positive mistakes, giving them a 2 contribution, and to count false negative mistakes.

Regression loss examples An example of a loss function in the case of regression are the *absolute loss*:

$$l(y, \hat{y}) = |y - \hat{y}|$$

ot the *square loss* that has some better properties:

$$l(y, \hat{y}) = (y - \hat{y})^2$$

Another example, in the case of whether forecast prediction:

$$Y = \{rain, sun\}, Z = [0, 1]$$

We want to define a regression task that outputs the probability of a whether condition. By using the absolute error loss function, we assume a linearity in the error, whereas we can assume that predicting sun while it rains can be a shame.
This means it is preferred to use something like a square loss, to keep track of such a wanted behavior.