
Assignment-8 : Supervised Learning

Harshvardhan Patidar

Department of Artificial Intelligence
Indian Institute of Technology Hyderabad
ai24btech11015@iith.ac.in

Chapter 19 : Learning from Examples

This chapter talks about how we can implement a sense of learning in the machines to ease our work and make them improve over time. Machine learning improves the functionality of different components of an agent program which depends upon:

1. Which component is to be improved.
2. What prior knowledge does the model have?
3. What data and feedback is available.

These components are agent function, sensors, information about how the world works and evolves over time, goal, utility function and the problem generator and critic which help the program to improve.

Machine learning has become a standard part of software development. Whether or not you call a software AI, but using machine learning can improve your software many-folds. Machine Learning had reduced the cost of cooling and maintaining high performance compute power that would have been required without it.

There are three main types of learning, supervised, unsupervised and reinforcement learning. In supervised learning, we use a large number of input and output, called as a label to train the model. In unsupervised learning, we instead try to make a cluster of similar input from a large input data and try to label them, while in reinforcement learning, agents get feedback in form of rewards and punishments and try to improve on the basis of the feedback.

Supervised Learning

The objective of supervised learning is to find a function h which is an approximation of the true function f which maps the given input-output pairs. The function h is called a hypothesis function. This hypothesis is drawn from a hypothesis space \mathcal{H} of possible functions. The \mathcal{H} may consist of various types of function such as boolean, polynomial, logical etc. For every data set, different hypothesis spaces might be appropriate. To find the best hypothesis space, we can perform exploratory data analysis, in other words we can try multiple hypothesis spaces and choose the best fitting one. A good hypothesis should fit well with unseen data also, as it determines the accuracy of the model. To fit well here means to accurately predict the output for some unseen input.

To discuss the accuracy of a hypothesis, we need to use the words bias and variance. Bias refers to the tendency of a hypothesis h to deviate from the true function f . If it has high bias, then this hypothesis may be underfitting, which means the hypothesis is unable to find a pattern in the given data. The other word Variance tells us about how much the hypothesis varies when the training data is changed. If it has a high variance, then it might struggle with overfitting, which means the

hypothesis may try to find useless patterns in the data rather than trying to find actual pattern. Since bias and variance don't go hand in hand, there is often a need to make a tradeoff between them, i.e. a choice between more complex, but low bias hypothesis, that might have high variance and may suffer overfitting, or a simpler model, which has lower variance, but is prone to under-fitting. Einstein suggests choosing the simplest model which goes well with the data which is identical to the Ockham's Razor principle.

Some analysts use the probability method to find the most probable hypothesis h^*

$$h^* = \arg \max_{h \in \mathcal{H}} P(h \mid \text{data})$$
$$h^* = \arg \max_{h \in \mathcal{H}} P(\text{data} \mid h)P(h)$$

Example Problem

We can consider an example problem, the aim of which is to decide whether to wait for a table at a restaurant or not. The input $x \in \mathbb{R}^{10}$ will have ten attributes, like is there any other restaurant nearby, is it raining or not, whether it has a comfortable area to wait or not, how hungry we're right now etc. You can always consider more or less attributes as per your choice. All these inputs will have an output $y \in \{True, False\}$ which tells whether or not one should wait at the restaurant. A model while learning from data relevant to this example can learn how significant different attributes are and what are their priority orders, and be able to produce the best possible output for some other test condition.