Introduction to Machine Learning -

Machine learning in a general sense refers to the culmination of useful data obtained from given data to be able to process previously unseen cases of the same category. A good learner should be able to generalize data, and this is reffered to as inductive reasoning.

Great care must be taken to ensure that only meaningful data is being extracted, and not the properties which are favourable by chance, like in the pigeon superstition example.

Having prior knowledge regarding the property in question greatly helps to deduce whether the information is useful, or not. The bias obtained due to prior knowledge is called inductive bias. However, having a very rigid prior makes the learning to be less flexible.

* Supervised Learning :-

The training data provided has additional information than the testing data, Such as labels indicating what the given picture represents. This additional data "supervises" the learning process, hence the name.

* Unsupervised Learning :-

No difference between training and testing data. The algorithm needs to find classifications by itself. Clustering of data sets is a typical example.

* Reinforcement Learning

Training examples contain more information than testing examples. This involves the agent interacting with the environment in a Trail-and-error manner. A good example for this would be Q-Learning.

– Other Classifications include –

- Active and passive
  - Active learner refers to when the agent actively seeks for feedback and user input. Passive learners usually do neither
- Helpfulness of teacher
  - Humans trying to understand nature can be labelled as nature being a passive teacher, as nature simply doesn't care what humans do.
  - A security software learning by dealing with hackers would model the hackers adversial teachers, and they act as "worst-case" training scenario.
- Online vs Batch Learning
  - Batch Learning usually has large amounts of data for the agent to learn from before making the final call, wheras online learning requires the agent to give answers on the go, making mistakes as it learns.

Formal model – Nomenclature

* Domain Set –

  - A set $\mathcal{X}$ which contains all the objects we wish to Label. Each element in $\mathcal{X}$ is represented by a vector of its features.

* Label Set :-

  - A set $\mathcal{Y}$ which contains all possible labels that every element in $\mathcal{X}$ may be awarded.

* Training data :-

  - A finite sequence of pairs, where each pair $\in \mathcal{X} \times \mathcal{Y}$ (looking at supervised learning) which the agent has access to.

* Learner's Output :-

  - A function $h : \mathcal{X} \to \mathcal{Y}$ obtained from analyzing the training data.
    The function is also called as classifier, predictor, Hypothesis or a prediction rule.
  - Let the training data be given by S, the predictor obtained is represented as $A(S)$ where 'A' is the predictor.

* Data generation model :-

  - We first assume the instances in $\mathcal{X}$ are generated via some pdf D, and that there is a "correct" labelling function $f : \mathcal{X} \to \mathcal{Y}$. Both D, f are unknown to the agent. The training data is obtained by obtaining a sample $x_i \in \mathcal{X}$ and using f to get its correct Label.

* Measure of Success :-

The error of a classifier is the probability that the label given by the agent doesn't align with the "true label". It is represented by L.

$$L_{D,f}(h) = \underset{x \sim D}{P}[h(x) \neq f(x)] = D(\{x : f(x) \neq h(x)\})$$

Loss of estimator 'h'
D - distr. of x
f - "true function"

Prob. that
$h \neq f$

By LOTUS, write
interms of D.

L is also known as risk, generalization error, true error. etc

• However, the agent has no idea what the values of D, f are; so the true error cannot be calculated by the agent. We therefore define Emperical error using the traing data as follows :-

$$L_S(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

emperical risk of h

→ no. of x_i where h is wrong

We thus try to minimize the emperical risk. The process of coming up with an estimator to minimize emperical risk is Emperical Risk Minimization (ERM). However, this does have some problems.

Problem

Overfitting - "If it's too good to be true, it probably is."

• Consider the estimator
$$h(x) = \begin{cases} y_i & \text{if } \exists x_i \in S \text{ and } x = x_i \\ 0 & ; \text{ otherwise} \end{cases}$$

The estimator can be seen to have emperical error as zero, meanwhile its true error is 0.5, which is horrible. Such a phenomenon where the estimator performs amazingly well with train data, but poorly outside, is called as Overfitting.

* ERM using Inductive bias :-

- Suppose that you define a set of functions, H, using prior knowledge of the problem. Also, suppose that $\forall h \in H$, overfitting does not occur. From this bias, we can choose an estimator $h \in H$ such that $\forall h' \in H ; L_H(h') \geq L_H(h)$, i.e, the ERM for h is the least.
- Therefore, the problem now becomes choosing a proper set H, and proving that overfitting cannot occur for any element.
- Also understand that reducing the size of H protects us against overfitting at the cost of learning becoming stiff. (Trade-off!)