

MACHINE LEARNING Notes

Supervised Learning:

Applications in which the training data comprises of input vectors along with the corresponding target vectors.

Classification:

Applications in which the aim is to assign the input vector to one of a finite number of discrete categories.

For ex: Classification of digits in digit-recognition problem.

Regression:

Similar to Classification problem, but if the desired output consists of one or more continuous variables then the task is called Regression

For ex: If the desired o/p is 'yield' of a chemical manufacturing plant where the input vector is temperature, pressure, conc. of reactants etc.

Unsupervised Learning:

In other pattern recognition problems, the training data consists of a set of input vectors X without any corresponding target vectors. This is called Unsupervised Learning problem.

Clustering:

Unsupervised learning problems where the goal is to discover groups of similar examples within the data.

Density Estimation:

Unsupervised learning problem where the goal is to determine the distribution of data within the input space.

Reinforcement Learning:

It is concerned with the problem of finding a suitable actions to take in a given situation to maximize a reward.

Here the learning algorithm is not given the examples of optimal output, in contrast to Supervised Learning, but must instead discover them by Trial & Error.

Typically there is a sequence of states & actions in which the learning algorithm is interacting with its environment.

In many cases, the current action not only affects the immediate reward but also has an impact on the rewards at all subsequent time steps.

For eg: Using appropriate reinforcement learning techniques a Neural Network can learn to play a game of chessboard to a very high standard.

A general feature of reinforcement learning is a trade-off b/w

1. EXPLORATION: System tries out new kinds of actions to know how effective they are
2. EXPLOITATION: System uses known actions to yield a high reward

Linear Models:

Functions which are linear in unknown parameters are known as Linear Models.

Overfitting:

For a given model complexity, the overfitting problem becomes less severe as the size of data increases.

The larger the data-set, the more complex(in other words more flexible) the model that we can afford to fit the data.

One rough heuristics to choose the data set is:

The number of data points should be no less than some multiple of the no. of adaptive parameters in the model.

However, no. of parameters is not the only measure of model complexity.

Overfitting is an general problem of Maximum Likelihood Estimation(MLE) and it can be avoided by choosing the Bayesian approach.

1. Regularization
2. Ridge Regression
3. Weight Decay (in Neural Networks)

Expectation $E[*]$:

The average value of some function $f(x)$ under a probability distribution $p(x)$ is called the **Expectation** of $f(x)$ and is denoted by $E[f]$.

$$E[f] = \sum_x p(x) f(x)$$

(Average is weighted by relative probabilities of different values of x)

$E_x[f(x, y)]$ denotes average of function $f(x, y)$ w.r.t distribution of x . So, $E[f(x, y)]$ will be a function in y .

We can also define **Conditional Expectation** ($E[f|y]$) w.r.t. conditional distribution ($p(x|y)$) in similar manner.

$$E[f|y] = \sum_x p(x|y) f(x)$$

Variance `var[*]` & Covariance `cov[*]` :

Variance `var[*]` :

It provides a measure of how much variability there is in `f(x)` around its mean value (`E[f(x)]`).

$$\text{var}[f(x)] = E[(f(x) - E[f(x)])^2]$$

$$\text{var}[X] = E[(X)^2] - (E[X])^2$$

Covariance `cov[*]` :

For two random variables `x` and `y`, the covariance (`cov[x,y]`) is a measure of the extent to which `x` and `y` vary together.

$$\text{cov}[x,y] = E[xy] - E[x]E[y]$$

If `x` and `y` are mutually independent then their covariance is zero.

In case of two vectors of random variables X and Y , covariance is a Matrix.

Covariance of components of vector X with each other :

$$\text{cov}[X] = \text{cov}[X, X]$$

Likelihood function: $p(D|w)$

In ML literature, the negative log of Likelihood function is called Error function

$$\text{Error function} : -\log(p(D|w))$$

In MLE estimation, we try to **maximize** the *likelihood function* ($p(D|w)$) & b'coz $\log()$ is a monotonically increasing function thus, **maximizing likelihood minimizes error**

(remember: error is negative of log, so error is a monotonically decreasing function)

Frequentist v/s Bayesian viewpoints:

One common criticism of Bayesian viewpoint is that, the **prior distribution** is often selected on the basis of mathematical convenience rather than as a reflection of any prior beliefs.

Bayesian approach based on poor priors can give poor results with high confidence.

