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 numer of discrete categories.

For ex: Classification of digits in digit-recognition prolem.
```

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 trainig 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)]`).
>
> $$var[f(x)] = E[(f(x) - E[f(x)])^2]$$
>
> $$var[X] = \sigma^2 = E[(X)^2] - (E[X])^2$$

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.
>
> $$cov[x, y] = E[xy] - E[x]E[y]$$
>
> 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 :
>
> $$cov[X] = cov[X, X]$$

Likelihood function: $p(D|w)$

> In ML literature, `the negative log` of Likelihood function is called `Error function`
>
> $$Error function : -log(p(D|w))$$
>
> 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)

Frequestist 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.

The Gaussian Distribution

For a single `real-valued` varaiable `x`, the Gaussian Distribution is defined as:

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{ -\frac{1}{2\sigma^2}(x-\mu)^2 \right\}$$

`Mean` = μ

`Variance` = σ^2

`Standard Deviation` = σ

`Precision` (β) = $1/\sigma^2$

> The `maximum` of a distribution is called `Mode`
> For a Gaussian Distribution, `Mode` coincides with `Mean` (μ)

$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{+\infty} \mathcal{N}(x|\mu,\sigma^2)\,dx = 1$$

Expectation or Mean :

$$E[x] = \int_{-\infty}^{+\infty} \mathcal{N}(x|\mu,\sigma^2)\,x\,dx = \mu$$

and

$$E[x^2] = \int_{-\infty}^{+\infty} \mathcal{N}(x|\mu,\sigma^2)\,x^2\,dx = (\mu^2+\sigma^2)$$

So **variance** `var[x]` is,

$$\sigma^2 = E[x^2] - E[x]$$

The Multivariate Gaussian Distribution

The Gaussian Distribution defined over a **`D-dimensional`** vector $\mathbf{x}$ of continuous variables is :

$$\mathcal{N}(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu) \right\}$$

where,

$\mathbf{x} = (x_1, x_2, x_3 ..... x_D)^T =$ `D-dimensional` **vector** of continuous variables

$\mu =$ `Mean` $=$ `D-dimensional` **vector**

$\mathbf{\Sigma} =$ `Covariance` $=$ `DxD` **Matrix**

1. `i.i.d` = `independent and identically distributed` : Data points that are drawn independentally from the same distribution.
2. `Joint Probability` of `independent` events is given by the product of the `marginal probabilities` of each event separately.

So, suppose we have a `data-set` of N observations of `single-valued variable` `x` :

$$\mathbf{X} = (x_1, x_2, x_3.....x_N)^T$$

Then, the probability of the `data-set` $\mathbf{X}$ or the `likelihood function` of the Gaussian is given by:

$$p(\mathbf{X}|\mu, \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu, \sigma^2)$$

> Note: ($\mathbf{X}$ is not a Vector like $\mathbf{x}$, it is a collection of N observations of single valued variable x.

1. `Binomial and Multinomial Distributions` for `Discrete random varables`
2. `Gaussian distribution` for `Continuous random variables`
3. **`Parametric Distributions`**: Distributions that are governed by small no. of `adaptive parameters` (like μ and σ for Gaussian Distributions).

   One limitation of the parametreic approach is that it assumes a specific functional form of distribution which may turn out to be inappropriate for a particular application.

4. **`Density Estimation`**: Modelling the *probability distribution* $p(\mathbf{x})$ of a random variable $\mathbf{x}$ given a finite set $\{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3} \ldots\ldots \mathbf{x_N}\}$ of `observations`.
5. The `conjugate prior` for the parameters of `multinomial distribution` is called **`Diritchlet Distributions`**
6. While, the `conjugate prior` for the Gaussian is another Gaussian.
7. `Exponential family of Distributions`.

8. **`Non-parametric Density Estimation`**: Here, Distributions typically depend on the size of the data-set. Such models still have `parameters`, but these control the `model complexity` rather than the Distribution.

It is a general property of **`Bayesian Learning`** that as we observe more and more data D the `uncertainity` represented by the `posterior distribution` $p(\theta|D)$ decreases steadily. Below is the explanation:

For eg: Consider a general `Bayesian` inference problem for a `parameter` θ for which we have observed a `data-set` D, described by the `Joint Distribution` $p(\theta, D)$. Then it can be deived that:

$$E_\theta[\theta] = E_D[E_\theta[\theta]]$$

The above result says that the **posterior mean of** θ averaged over the distribution generating the data D is equal to the **prior mean of** θ, which is proven in below derivation:

$$E_\theta[\theta] = \int_\theta \theta \; p(\theta) \; d\theta$$

$$E_D[E_\theta[\theta|D]] = \int_D \left\{ \int_\theta \theta \; p(\theta|D) \; d\theta \right\} p(D) \; dD$$

$$E_D[E_\theta[\theta|D]] = \int_\theta \left\{ \int_D \Big(p(\theta|D) \; p(D) \Big) \; dD \right\} \theta \; d\theta$$

$$E_D[E_\theta[\theta|D]] = \int_\theta \left\{ \int_D p(\theta, D) \; dD \right\} \theta \; d\theta$$

$$E_D[E_\theta[\theta|D]] = \int_\theta p(\theta) \; \theta \; d\theta = E_\theta[\theta]$$

So,

$$E_\theta[\theta] = E_D[E_\theta[\theta]]$$