JUNAID. GIRKAR
60004190057
TE COMPS A-4

MACHINE LEARNING

ASSIGNMENT-1

Q1 Explain radial basis function in detail.

ANS. Radial basis kernel is a kernel function that is used in machine learning to find non-linear classifier or regression line.

Kernel function is used to convert or transform n-dimensional to m-dimensional input where m is much higher than n. then find the dot product in higher dimentional efficiently. The main idea to use kernel is - A linear classifier or regression curve in higher dimensions becomes a non-linear classifier or regression curve in lower dimension.

Mathematical defination of Radial Basis Kernel :-

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

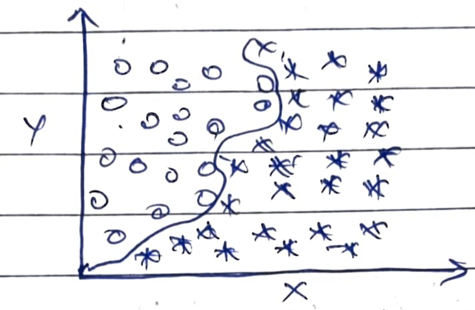
where x and x' is vector point in any fixed dimensional space. But if we expand the above exponential expression, it will go upto infinite power of x and x', as expansion of e^x contains infinite terms upto infinite power of x hence it involves terms upto infinite

powers in infinite dimensions.
If we apply any of the algorithm like perceptron algorithm or linear regression on this kernel, actually we would be applying our algorithm to new infinite-dimentional data point we have created. Hence it will give the hyperplane of infinite dimension; which will give very strong non-linear classifier or regression curve after returning to our original dimension.

$$a_1 x^{inf} + a_2 x^{inf-1} + a_3 x^{inf-2} + \dots + a_{inf} x + c.$$

So although we are applying linear classification/regression, it will give a non-linear classifier or regression line, that will help to be a polynomial or infinite power. And being a polynomial or infinite power, radial basis kernel is a very powerful kernel, which can give curve fitting to any complex dataset.

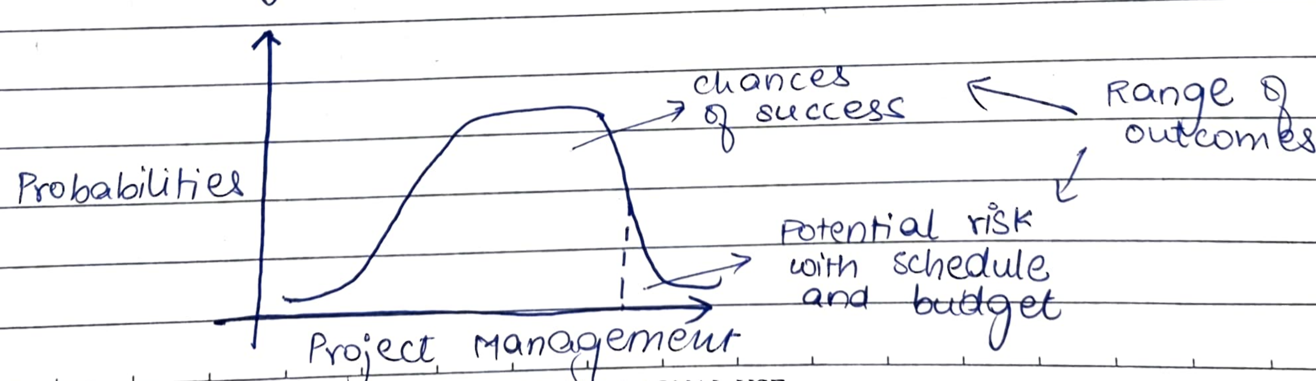
The main motive of this kernel is to do calculations in any d-dimentional space where $d > 1$, so we can get a quadratic, cubic or polynomial equation of large degree for our classification/regression line. Since radial bias kernel uses exponent as we know the exponents of e^x gives a polynomial equation of infinite power, so using this kernel, we can make classification line infinitely powerful too.



Q2 Enplain Monte Carlo with suitable enample ?

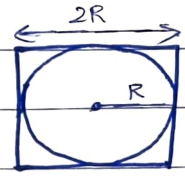
ANS Monte Carlo Simulation is a mathematical way for calculating odds of multiple possible outcomes occuring in uncertain processes through repeated random sampling. This computational algorithm makes risks associated with a particular process convinient, thereby enabling better decision making.

It is a statistical technique which uses the randomness to solve probabilistic problems. Also it can perform sensitive analysis and correlation between input variables. It finds its application in prediction and forecasting models in business supply chain, project management, etc.



Consider the following examples :

If a circle of radius R is inscribed inside a square with side length 2R, then the area of circle will be πR^2 and the area of square will be $(2R)^2$. So ratio of circle area to area of square will be $\pi/4$



It also means that if we pick a random point (x,y) both x and y are between $(1,-1)$, probability of this random point lies inside the unit circle is given as proposition between area of unit circle & square.

$$P(x^2 + y^2 < 1) = \frac{\text{Area (circle)}}{\text{Area (square)}} = \frac{\pi}{4}$$

So if ~~we~~ you pick N points at random inside square approximately $N\pi/4$ of these points should fall inside a circle.

$$M \text{ (no. of points inside the circle)} = \frac{N\pi}{4}$$

So supposing, we pick 'N' random points, out of which 'M' of those fall inside the circle, we can then calculate π by :-

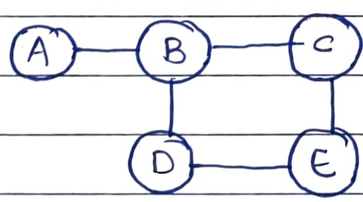
$$\pi = \frac{4M}{N}.$$

Q3 Write a short note on :

1) Markov Random fields :-

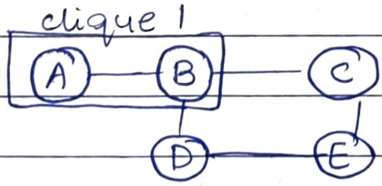
Markov Random model is a model which uses the unidirected graph. Unidirected graphical models represent edges represent the potential between two variables, syntatically, factorization distribution probabilities between variable. In each individual variable connected with edges represent a certain clique in graph ; means probability distribution of graph can factorize an individual clique potential function.

Just as we had CPD's for Bayesian networks, we have to incorporate relationship between nodes in Markov networks. However there are 2 crucial difference between the tables and CPD's.



clique in graph theory it is a subset of vertices of an unidirected graph.

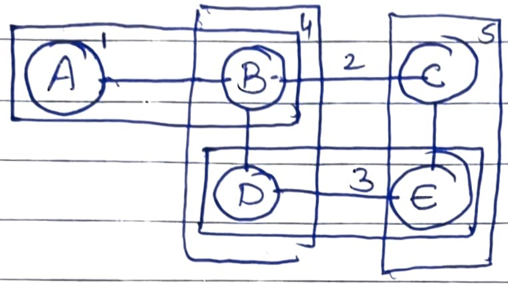
$P(A,B,C,D,E) \propto \phi(A,B)\ \phi(B,C)\ \phi(B,D)\ \phi(C,E)\ \phi(D,E)$

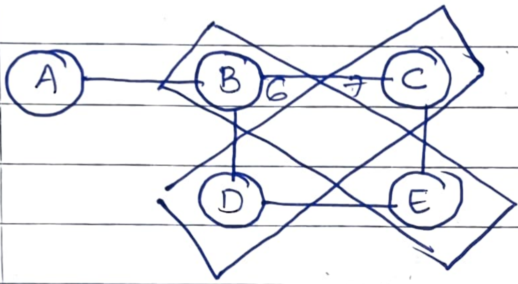


$P(x) = \frac{1}{Z} \prod_{clique} \phi_c(x_c)$

(Potential function)

Such that it includes subgraph in every vertices in a clique is adjacent. So clique in this graph adjust adjacenty one by one.

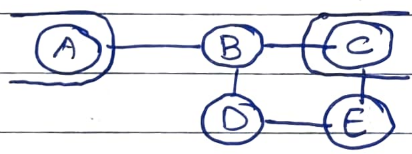




There is some difference if we join D,C & B,E clique over here, then it changes its probability.

$$P(A,B,C,D,E) \propto \phi(A,B)\ \phi(B,C,D)\ \phi(C,D,E)$$

Some undirected graphic model has Markov Random field. In MRF, certain paths between A and C



$A \rightarrow B \rightarrow C$

$A \rightarrow B \rightarrow D \rightarrow E \rightarrow C$

Unlike the Bayesian model, Markov networks do not need to be acyclic.

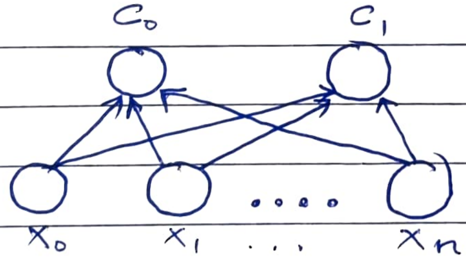
Independence properties such as Markov properties:-

- Any two subsets of variables are conditionally independent given a separating subset.
- If we take 'A' as a subset and 'C' as one subset then there is a wole between them. So there is no way to go between 'A' and 'C' without getting through the subset. So we are

using (A, B) then B, C, D, E. Therefore A and C are separating subset.

2 Self organizing Maps :- (SOM)

It is a type of Artificial Neural Network which is inspired by biological models of neural systems from 1970's. It follows an unsupervised learning approach and has trained its ~~algorithm~~ network through a competitive learning algorithm. SOM is used for clustering and mapping techniques to map multidimensional which allows people to reduce complex problems for easy interpretation. SOM has 2 layers : input layer and output layer. The architecture of SOM with 2 clusters and n input feature of any sample is given below.



ALGORITHM :

1) Weight Initialization
2) For 1 to n number of epochs
3) Select training example.

4) Compute winning vector
5) Update the winning vector
6) Repeat steps 3,4,5 for all training example.
7) Clustering the test sample.

Lets say an input data of size (m, n) where 'm' is number of training examples and 'n' is number of features in each example. First it initializes the weights of size (n, c) where 'c' is the number of clusters. Then iterating over the input data, for each training example, it updates the winning vector.

Weight updation rule is given by :-

$$w_{ij} = w_{ij}(old) + \alpha(t) * (x_i^k - w_{ij}(old))$$