Markov Decision Process

A Markov decision process (MDP) is a **stochastic decision-making process** that is used to model the decision-making of a dynamic system.

It represents an environment for reinforcement learning which satisfies markov property. Here, we assume here that the environment is fully observable, which means that we have all the information we need to make a decision given the current state.

MDP is composed of following 5 components
S: set of states
A: set of actions
R: reward function
P: transition probability function
γ: discount for future reward

Markov property states that given the present state, the future is independent of the past states.
A state S_t is Markov if and only if

$$P[S_{(t+1)} \mid S_t] = P[S\_(t+1) \mid S\_1, \ldots, S\_t]$$

It means that the present state captures all relevant information from history.
Here, P represents state transition probability to future state S_(t+1) from the current state S_t.

The transition probabilities describe the likelihood of transitioning from one state to another when a particular action is taken.
With each transition a reward is associated.

The main objective of an MDP is to find a policy, or a set of rules for selecting actions in each state, that maximizes the expected cumulative reward obtained.

MDPs are widely used in artificial intelligence and decision-making problems, such as robot navigation.

Probabilistic Reasoning

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge.

In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.

In the real world, there are a lot of scenarios where there is uncertainty like - “Will it rain today”. In such situations we need to use probabilistic reasoning.

It is based on Bayes Theorem.
According to Bayes theorem,

$$P(A \mid B) = \{ P(B \mid A) \cdot P(A) \} / P(B)$$ [see notes]

The key concept of probabilistic reasoning is the use of probabilities to represent uncertainty. Probabilities are assigned to events or outcomes based on their likelihood of occurring which can be used to make predictions or decisions, taking into account the level of uncertainty involved.

Probabilistic reasoning is useful for analyzing and making decisions in situations which are complex and uncertain.

Unsupervised Learning - K means Clustering

Unsupervised learning is a type of machine learning in which the model is not provided with a labeled dataset, and instead, the model is expected to identify patterns and structures in the data on its own.

One popular unsupervised learning technique is K-means clustering.

K-means clustering is a type of clustering algorithm that aims to partition a set of data points into K (selected by user) clusters based on their similarity.

Algorithm:-

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points and make them centroids.

Step-3: Assign each data point to their closest centroid.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means re-assign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to Step-7

Step-7: The model is ready.

Diagram -:

Advantages:-

1. Easy to implement.
2. Fast and efficient.

Disadvantage:-

1. Selection of appropriate value of K - initial clusters
2. Sensitive to outliers

SVM

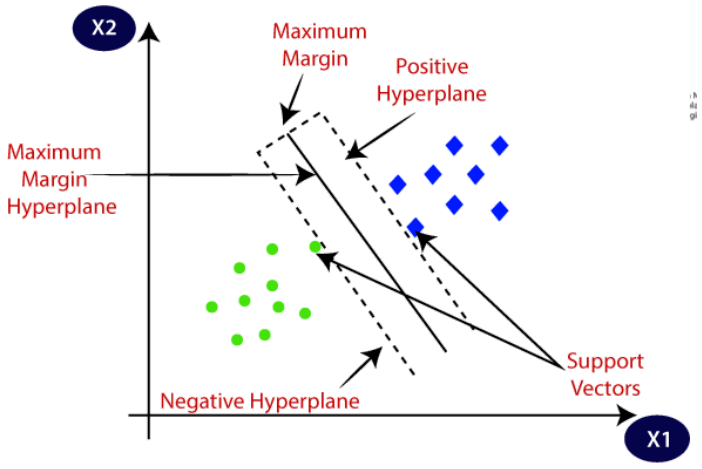
SVM can be of **two types**:

- **Linear SVM**: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM**: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Kernelization - used to transform non-linear dataset to linear dataset
Kernel functions used :-

1. RBF - Radial Basis Function
2. Sigmoid
3. Polynomial
4. Linear

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme points are called support vectors, and hence algorithm is termed as Support Vector Machine.



SVM algorithm can be used for Face detection, image classification, etc

AO * Algorithm

AO* search is an informed search algorithm that combines the strengths of A* search and iterative deepening. Like A* search, AO* search uses heuristics to guide the search towards the goal state. However, unlike A* search, which expands all nodes up to a certain depth, AO* search incrementally expands nodes in order of their estimated evaluation function value.

Algorithm:-

Diagram:-

Advantage -
One of the advantages of AO* search over A* search is that it can handle situations where the heuristic function is not admissible, meaning it overestimates the evaluation function value,

Disadvantage -

1. It doesn't always give the optimal solution
2. It doesn't explore all solution paths once it gets the solution.

Cross-validation

Cross-validation is a technique for validating the model efficiency by training it on the subset of input data and testing on previously unseen subset of the input data. We can also say that it is a technique to check how a statistical model generalizes to an independent dataset.

The basic steps of cross-validations are:

- Reserve a subset of the dataset as a validation set.
- Provide the training to the model using the training dataset.
- Now, evaluate model performance using the validation set. If the model performs well with the validation set, perform the further step, else check for the issues.

Types of cross-validation
1. K-fold cross-validation 2. Hold-out cross-validation 3. Stratified k-fold cross-validation 4. Leave-p-out cross-validation 5. Leave-one-out cross-validation 6. Monte Carlo (shuffle-split)

K-fold cross-validation

In this technique, the whole dataset is partitioned in k parts of equal size and each partition is called a fold. It's known as k-fold since there are k parts where k can be any integer - 3,4,5, etc.
One fold is used for validation and other K-1 folds are used for training the model.
To use every fold as a validation set and other left-outs as a training set, this technique is repeated k times until each fold is used once.

Regularization

Regularization is a technique in machine learning that is used to prevent overfitting of the model. Overfitting occurs when the model learns the noise in the training data, which can result in poor performance on the test data. Regularization methods add a penalty term to the cost function of the model, which helps to reduce the complexity of the model and prevent Overfitting.

L1 Regularization (Lasso Regression): L1 regularization adds a penalty term proportional to the absolute value of the coefficients to the cost function of the model. L1 regularization is useful when the number of features is large, and only a few of them are relevant to the prediction.

L2 Regularization (Ridge Regression): L2 regularization adds a penalty term proportional to the square of the coefficients to the cost function of the model. L2 regularization is useful when all features are potentially relevant to the prediction, and the model needs to be regularized to prevent overfitting.

Elastic Net Regularization: Elastic Net regularization is a combination of L1 and L2 regularization. It adds both penalty terms to the cost function of the model, which helps to overcome the limitations of each individual regularization method.