UNIT -4

Machine Learning :

- Supervised and unsupervised learning
- Decision trees.
- Statistical learning models
- Learning with complete data - Naive Bayes models.
-

Learning with hidden data

- EM algorithm
- Reinforcement learning

ANUJ KHANNA
ASSITANT PROFESSOR (KIOT,Kanpur).

Short Questions & Answers

Ques 1. Name out three basic techniques of machine learning .

Ans : (a) Supervised Learning (b) Unsupervised Learning (c) Reinforcement Learning.

Ques 2. Write some applications of Supervised Learning.

Ans :

- Implementation of Perceptrons in AI.
- Implementation of Adaline network
- Application in Back propagation algorithms.
- Used in Hetero associative learning.

Ques 3. What is Boolean Decision Tree?

Ans : These are used in Decision Making learning technique. This consists of a vector of input attributes X, and a single Boolean output y. Example: Set of examples (X_1 , Y_1)……(X_6 , Y_6).

Positive examples are in which goal is true . Negative examples are in which goal is false.

Complete set is called Training Set.

Ques 4. Compare the Decision tree method with Naïve Baye's Learning.

Ans : (i) Naïve Baye's learns little less efficiently as compared to decision tree learning.

(ii) Naïve Baye's learning works well fro wide range of applications as compared to decision tree.

(iii) Naïve Baye's Scale well to very large problems. E.g : If n Boolean attributes , then 2n + 1 Parameters are required.

Ques 5. What is Reward Function in Re-enforcement learning ?

Ans : Reward function is used to define a goal. It maps each perceived state action pair of environment to a single number; i.e. a reward that indicates desirability of that state. A re-enforcement agent's only objective is to maximize total reward received in long run. Reward functions are stochastic/ random in nature.

Long Question & Answers

Ques 6. Explain Machine learning. Illustrate learning model? Mention some factors that affect the learning.

Ans : Machine learning is the sub field of AI in which we try to improve decision making power of intelligent agents. Agent has a performance element that decides what actions to take and a learning element that modifies the performance element so that it makes better decisions. Design of learning element is affected by following three major factors :

1) Which components of performance element are to be learned.
2) What feedback is available to learn these components.
3) What is representation method used for components**.**

Following are some ways of learning mostly used in machines:

(A) Logical learning (B) Inductive learning (C) Deductive learning.

(B)

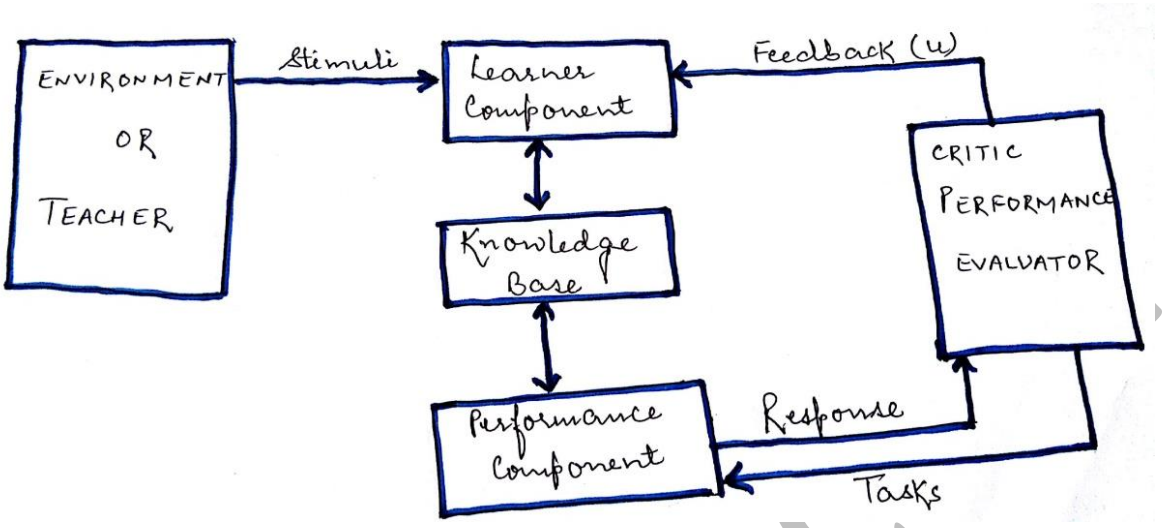
Logical Learning: In this process a new concept or solution through the use of similar known concepts. We use this type of learning when solving problems on an exam , where previously learned examples serve as a guide or when we learn to drive a truck using our knowledge of car driving.

Inductive Learning: This technique requires the use of inductive inference, a form of invalid but useful inference. We use inductive learning when we formulate a general concept after seeing a number of instances or examples of the concept. E.g : When we learn the concept of color or sweet taste after experiencing sensation associated with several objects.

Deductive Learning: This is performed through a sequence of deductive inference steps using known facts. From the known facts , new facts or relationships are logically delivered. E.g : If we have an information that weather is Hot and Humid then we can infer that it may Rain also. Another example may be , let

$P \rightarrow Q \ \& \ Q \rightarrow R$, *then we can infer that* $P \rightarrow R$

General Learning Model



Environment has been included as a part of the overall learning system. It produces random stimuli, which work as a organized training source such as a teacher which provides carefully selected training examples for learner component. A user working on a keyboard can also be an environment for some specific systems.

Inputs to the learning system may be physical ***stimuli, some sound , signal ,description of text , symbolic notations*** . Information is used to create and modify knowledge structures in the KB. Same knowledge is used by the performance component to carry out some tasks, such as solving a problem, playing a computer game .

Performance component produces a response/actions when a task is provided. The **Critic module** then evaluates this response relative to an optimal response. A **feedback** indicating whether or not the performance is acceptable. It is then forwarded by critic module to learner component for its subsequent use in modifying the structure in knowledge base.

Factors affecting the Machine Learning Process:

1) Types of training provided. E.g: Supervised technique , Unsupervised technique etc.
2) Form and extent of any initial background knowledge or past history.
3) The types of feedbacks provided.
4) Learning algorithms applied.

Ques7. Differentiate between Supervised Learning and Unsupervised Learning. Also mention some of the application areas of both.

Ans :

S.No	Supervised Learning	Unsupervised Learning
1.	Learning of a function can be done from its inputs and outputs,	Learning can be used to draw inference from some data set containing input data
2.	Classifies the data on the basis of training set available and uses that data for classifying new data.	Clusters the data on the basis of similarities according to the characteristics found in the data and grouping similar objects into clusters.
3.	Also known as Classification	Also known as Clustering
4.	The class labels on the training data is known in advance which further helps in data classification.	Class labels on the training data is not known in advance i.e. no predefined class.
5.	**Classification Methods:** Decision Trees, Bayesian Classification. Rule Based Classification Classification by back propagation, Associative Classification.	**Clustering Methods** : Hierarchical, Partitioning, Density Based. Grid Based, Model Based.

Issues in supervised learning

- Data Cleaning: In data cleaning, noise and missing values are handled.
- Feature Selection: Abundant an irrelevant attributes are removed while feature selection is done.
- Data Transformation: Data normalization and data generalization is included in data transformation.

Ques. 8 Write Short notes on the following: (a) Statistical Learning (b) Naïve Baye's Model

Ans : (a) Statistical Learning Technique: In this technique main idea is data and hypothesis. Here data is evidence i.e. instantiations of some or all random variables describing the domain. Bayesian learning calculates probabilities of each hypothesis given the data and makes prediction.

Let D: data set, with observed value d as an output. Then the probability of each hypothesis is obtained by Baye's Rule as: $\mathbf{P(h_i \mid d) = \alpha P(d \mid hi) P(hi)}$.

For prediction of an unknown quantity x , expression is given as below :

$$\mathbf{P(x \mid d) = \sum_i P(x \mid d, hi)\, P(hi \mid d) = \sum_i P(x \mid hi)\, P(hi \mid d).}$$

Prediction above is weighted averages over predictions of individual hypothesis. Hypothesis are intermediate values between raw data and predictions. ***A very common approximation which is generally used is to make predictions based on a single most probable hypothesis i.e. an hi that maximizes P (hi | d) is called Maximum a Posteriori.***

(b) Naïve Baye's Model: This is the most common Bayesian network model used in machine learning. In this model the class variable C (to be predicted) is the root and attribute Xi are leaves. Model is called Naïve because it assumes that attributes are conditionally independent of each other, given the class.

Once the model has been trained using maximum likelihood technique, it can be used to classify new examples for which the class variable C is unobserved. For the observed attributes $\mathbf{x_1, x_2, \ldots\ldots x_n}$, the Probability of each class is given as: $\mathbf{P(C \mid x_1, x_2 \ldots, X_n) = \alpha\, P(C) \prod_i P(Xi \mid C)}$.

Ques.9 What is learning with complete data? Explain Maximum Likelihood Parameter Learning with Discrete Model in detail.

Ans . Statistical learning methods are based on simple task parameter learning with complete data.

Parameter learning involves finding the numerical parameters for a probability model with a fix structure. E.g: In Bayesian network conditional probabilities are obtained for a given scenario. Data are complete when each point contains values for every variable in a specific learning model.

Maximum Likelihood Parameter Learning : Suppose we buy a bag of lime and cherry candy from a new manufacturer whose lime–cherry proportions are completely unknown—that is, the fraction could be anywhere between 0 and 1. Parameter θ is proportion of cherry candies.

Hypothesis is : $\mathbf{h\theta}$**, proportion of limes = 1 -** θ

If we assume that all proportions are equally ***likely a priori***, then a ***maximum-likelihood approach*** is reasonable. If we model the situation with a Bayesian network, we need just one random variable, **Flavor** (the flavor of a randomly chosen candy from the bag). It has values **cherry** and **lime**, where the probability of **cherry** is . Now suppose we unwrap N candies, of which c are cherries and l = N - c are limes

Likelihood of above data set is as given below:

$$P(\mathbf{d}|h_\theta) = \prod_{j=1}^{N} P(d_j|h_\theta) = \theta^c \cdot (1-\theta)^\ell$$

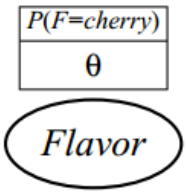
So maximum likelihood is value of θ that maximizes above equation .Computing log likelihood:

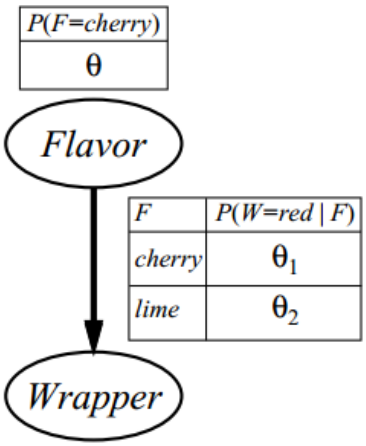
$$L(\mathbf{d}|h_\theta) = \log P(\mathbf{d}|h_\theta) = \sum_{j=1}^{N} \log P(d_j|h_\theta) = c\log\theta + \ell\log(1-\theta)$$

By taking logarithms, we reduce the product to a sum over the data, which is usually easier to maximize.) To find the maximum-likelihood value ofθ, we differentiate L with respect to θ and set the resulting expression to zero:

$$\frac{dL(\mathbf{d}|h_\theta)}{d\theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \qquad \Rightarrow \qquad \theta = \frac{c}{c+\ell} = \frac{c}{N} .$$

1. Write down an expression for the likelihood of the data as a function of the parameter(s).
2. Write down the derivative of the log likelihood with respect to each parameter.
3. Find the parameter values such that the derivatives are zer





when the data set is small enough that some events have not yet been observed—for instance, no cherry candies—the maximum likelihood hypothesis assigns zero probability to those events. Various tricks are used to avoid this problem, such as initializing the counts for each event to 1 instead of zero.

With complete data maximum likelihood parameter learning problem for a Bayesian Network Decomposes into the separate learning problems one for each parameter. Also parameter values for a

variable given its parents are just observed frequencies of variable values for each setting of parent values.

Let us look at another example: Suppose this new candy manufacturer wants to give a little hint to the consumer and uses ***candy wrappers colored red and green***. The Wrapper for each candy is selected ***probabilistically,*** according to some unknown conditional distribution, ***depending on the flavor***. The corresponding probability model has three parameters: θ, θ_1, and θ_2.

θ_1 : wrapper color of cherry candy. θ_2. : Wrapper color of lime candy.

Let us assume a case for Cherry Candy Wrapper, then using Joint probability distribution we can have following equation:

P (Flavor = Cherry, Wrapper = Green | h_θ , θ_1, θ_2).

$$= P(Flavor = cherry|h_{\theta,\theta_1,\theta_2})P(Wrapper = green|Flavor = cherry, h_{\theta,\theta_1,\theta_2})$$
$$= \theta \cdot (1 - \theta_1) .$$

Now let N candies are to be unwrapped: C : cherries , L = N – C : Lime

Let wrapper count is as given: **r_c : Cherries with red wrappers , g_c : Cherries with green wrappers**

r_l : Limes with red wrappers , g_l : Limes with green wrappers.

So the likelihood of data is given as below:

$$P(\mathbf{d}|h_{\theta,\theta_1,\theta_2}) = \theta^c(1-\theta)^\ell \cdot \theta_1^{r_c}(1-\theta_1)^{g_c} \cdot \theta_2^{r_\ell}(1-\theta_2)^{g_\ell} .$$

Now for Maximum Likelihood Estimation , simplify it by taking Log , to come up with addition form :

$$L = [c\log\theta + \ell\log(1-\theta)] + [r_c\log\theta_1 + g_c\log(1-\theta_1)] + [r_\ell\log\theta_2 + g_\ell\log(1-\theta_2)] .$$

Now compute I order partial derivatives w.r.t θ, θ_1, θ_2 , Equate it to zero , we will get values of parameters.

$$\frac{\partial L}{\partial\theta} = \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \quad \Rightarrow \quad \theta = \frac{c}{c+\ell}$$
$$\frac{\partial L}{\partial\theta_1} = \frac{r_c}{\theta_1} - \frac{g_c}{1-\theta_1} = 0 \quad \Rightarrow \quad \theta_1 = \frac{r_c}{r_c+g_c}$$
$$\frac{\partial L}{\partial\theta_2} = \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1-\theta_2} = 0 \quad \Rightarrow \quad \theta_2 = \frac{r_\ell}{r_\ell+g_\ell} .$$

Ques.10 Write short notes on

(a) Continuous model for Maximum likelihood Estimation

(b) Learning with Hidden Variables.

(c) EM Algorithm.

Ans : (a) Continuous model for Maximum likelihood Estimation : Continuous variables are very common in real-world applications, it is important to know how to learn continuous models from data. The principles for maximum-likelihood learning are identical to those of the discrete case. In learning the parameters of a Gaussian density function on a single variable. That is, the data are generated as follows:

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} .$$

The parameters of this model are the mean and the standard deviation. Let the observed values be **x1, X2 … xN .** Then the log likelihood is

$$L = \sum_{j=1}^{N} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_j-\mu)^2}{2\sigma^2}} = N(-\log\sqrt{2\pi} - \log\sigma) - \sum_{j=1}^{N} \frac{(x_j-\mu)^2}{2\sigma^2} .$$

:

Now setting theI order partial derivative equal to zero we obtain:

$$\frac{\partial L}{\partial \mu} = -\frac{1}{\sigma^2}\sum_{j=1}^{N}(x_j-\mu) = 0 \quad \Rightarrow \quad \mu = \frac{\sum_j x_j}{N}$$

$$\frac{\partial L}{\partial \sigma} = -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{j=1}^{N}(x_j-\mu)^2 = 0 \quad \Rightarrow \quad \sigma = \sqrt{\frac{\sum_j (x_j-\mu)^2}{N}} .$$

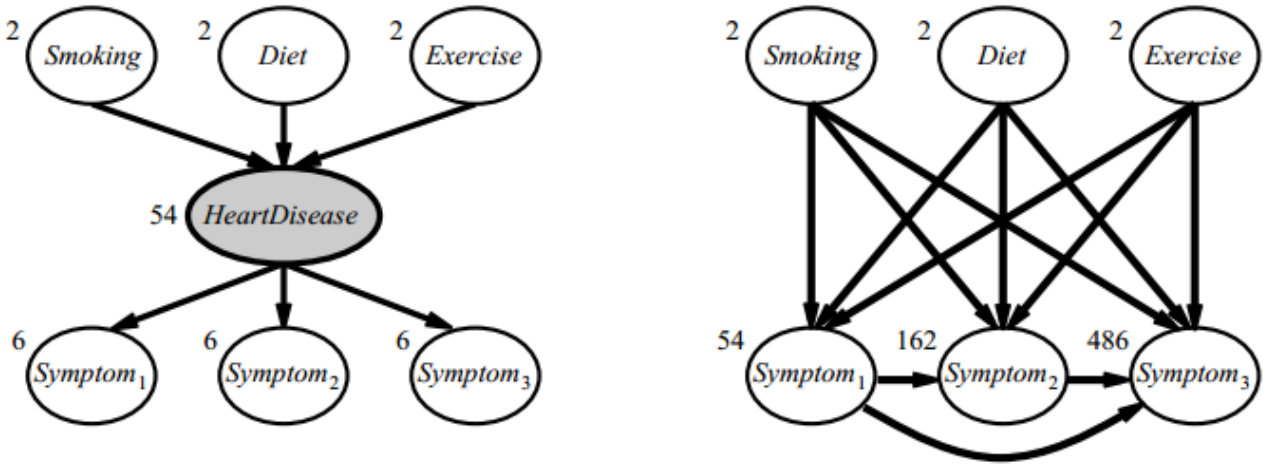
The maximum-likelihood value of the mean is the sample average and the maximum likelihood value of the standard deviation is the square root of the sample variance.

(b) Learning with Hidden Variables : Many real world problems have hidden variables (also called Latent Variables), which are not observable in given data set samples.

Example : (i) In medical diagnosis, records mostly consist of symptoms , treatment used and outcom of the treatment. But seldom have direct observation of disease itself.

(ii) A scenario of traffic congestion prediction at office hours (Hidden variables can be an unobservable " Rainy Day" causing very less traffic at peak hours.

Example : Let Bayesian Network for heart disease (a hidden variable) is as given in below figure :



. In figure (a): Each variable has three possible values and is labeled with the number of independent parameters in its conditional distribution. In figure (b): The equivalent network with Heart Disease removed. Note that the symptom variables are no longer conditionally independent given their parents. ***Therefore Latent Variables can dramatically reduce the number of parameters required to specify a Bayesian Network. This can reduce he amount of data needed to learn the parameters.***

(c) EM Algorithm(Expectation Maximization Algorithm) : This algorithm is used to solve the problems arised in Laerning with hidden variables. Basic idea is to pretend that we know the parameters of model and then infer the probability that each data point belongs to each component is fitted to entire data set with each point weighted by the probability that it belongs to that component.

- Expectation maximization the process that is used for clustering the data sample.
- EM for a given data, has the ability to predict feature values for each class on the basis of classification of examples by learning the theory that specifies it.
- It works on the concept of, starting with the random theory and randomly classified data along with the execution of below mentioned steps. Compute expected values of each hidden variables for each examples and then re-computing the parameters using the expected values as if they were observed values. Let X is the observed values in all examples. Z is the set of all hidden variables. θ is all parameters for probability model. $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

$$\boldsymbol{\theta}^{(i+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\mathbf{z}} P(\mathbf{Z}=\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^{(i)}) L(\mathbf{x}, \mathbf{Z}=\mathbf{z}|\boldsymbol{\theta})$$

- **E- Step:** In this computation of sum (i.e. expectation of Log likelihood of completed data w.r.t. P (Z = z | x , θ^i) , which is posteriori over hidden variables.

- **M – Step:** In this step we find new values of the parameters that maximize the Log Likelihood of data given the expected values of hidden indicator variables.

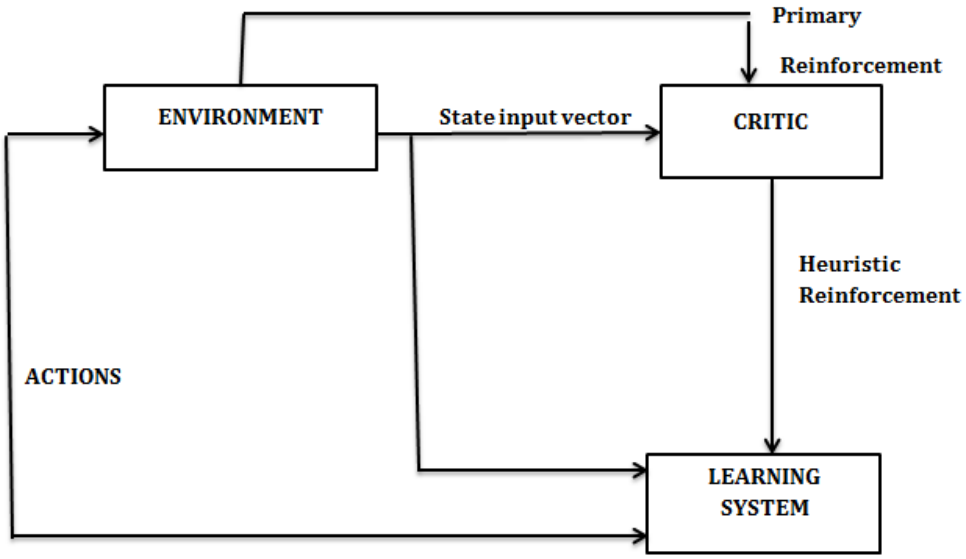
- EM algorithm increases the Log Likelihood of data at every iteration. Under certain conditions EM can be proven to reach a local maximum in likelihood. So EM is like ***Gradient Based Hill Climbing Algorithm.***

Ques. 11 Explain Re-inforcement learning technique in detail .Also Mention its applications in the field of Artificial intelligence.

Ans : Re-inforcement learning : This type of learning technique is used for agents learning when there is no teacher telling the agent what action to take in each circumstances.

Example 1 : Let a chess playing agent by supervised learning given examples of game situations along with the best moves for those situations. He can also try random moves , so agent can eventually build a predictive model of its environment. Issue is that "Without some feedback about what is good and bad , agent will have no grounds for deciding which move to select." ***Agent needs to know that something good has happened when it wins and that something bad has occurred. This kind of Feedback is called Reward or Re-inforcement .***

A General Learning Model of Reinforcement Learning:



- Reinforcement learning was developed in context to optimal control strategy.
- This method is useful in making sequential decisions
- Critic converts a primary reinforcement signal received from the environment into a higher quality signal (Heuristic Signal), both of which are scalar inputs.
- System is designed to learn delayed reinforcement (Temporal sequence of stimuli).

Example 2 : A mobile robot decides whether it should enter a new room in search of more trash to collect or start trying to find its way back to its battery recharging situation. It makes it decision based on how quickly and easily it has been able to find the recharger in past.

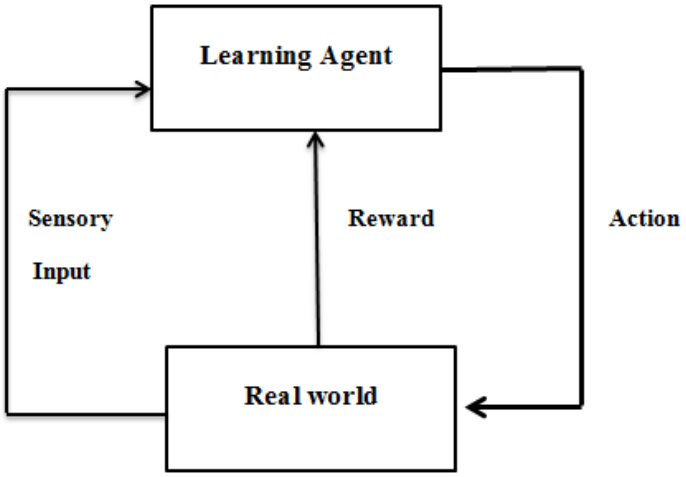
- Agent's actions are permitted to affect the future state of environment .E.g : Next chess position.
- This involves interaction between an active decision making agent and its environment, where goal is to be searched.

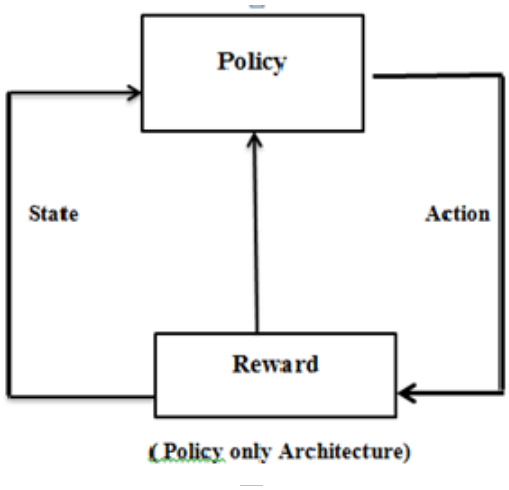
Markov Decision Process: Rewards serve to define optimal policies in MDP's. An optimal policies that maximizes expected total reward. Task of re-inforcement learning is to ise observed rewards to learn an optimal policy.

Elements of re-inforcement Learning:

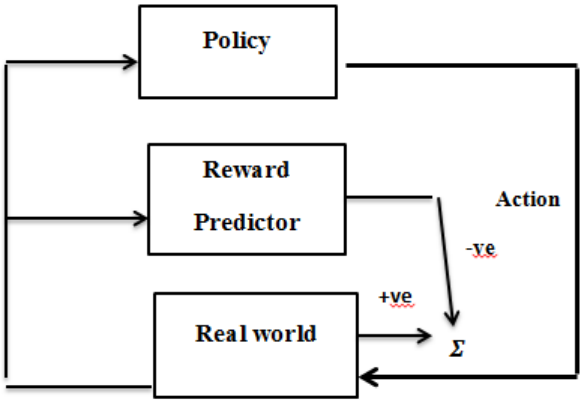
a). A policy b). A reward function c). A value function d). A model of environment

Architectures in Reinforcement Learning





(Policy only Architecture)



(Comparison - Architecture)

Policy: This defines learning agent's behavior at a particular time. It is a mapping from perceived states of environment to actions to be taken when present in those states. Policy can be a simple function , a look up table or a search process too.

Reward Function: This is used to define a goal. It maps each perceived state action pair of environment to a single number; a reward point that indicates desirability of that state. Objective is to maximize total reward function received in long run. Reward functions are stochastic/random.

Value function: Reward function indicates what is good in an immediate sense, a value function specifies what is good in the long run. Value of a state is total amount of reward an agent can expect to accumulate over the future.

Model: this represents behavior of the environment . Models are used for planning, i.e a way of deciding for a course of actions by considering future situations.

Application areas of Reinforcement learning are as mentioned below:

1) The most recent version of Deep Mind's AI system for playing Go) means interest in reinforcement learning (RL) is bound to increase.
2) RL requires a lot of data, and as such, it has often been associated with domains where simulated data is available (gameplay, robotics).
3) Automation of well-defined tasks, that would benefit from sequential decision-making that RL can help automate (or at least, where RL can augment a human expert).
4) Industrial automation is another promising area**.** ***It appears that RL technologies from DeepMind*** helped Google significantly reduce energy consumption (HVAC) in its own data centers.
5) The use of RL can lead to training systems that provide custom instruction and materials tuned to the needs of individual students**.** ***A group of researchers is developing RL algorithms and statistical methods*** that require less data for use in future tutoring systems.
6) Many RL applications in health care mostly pertain to finding ***optimal treatment policies*****.**
7) Companies collect a lot of text, and good tools that can help unlock unstructured text will find users.
8) A technique for automatically generating summaries from text based on content "abstracted" from some original text document).
9) ***A*** **Financial Times** ***article*** described an RL-based system for optimal trade execution. ***The system (dubbed "LOXM")*** is being used to execute trading orders at maximum speed and at the best possible price.
10) Many warehousing facilities used by E - Commerce sites and other supermarkets use these intelligent robots for sorting their millions of products every day and helping to deliver the right products to the right people. If you look at Tesla's factory, it comprises of more than 160 robots that do major part of work on its cars to reduce the risk of any defect.
11) Reinforcement learning algorithms can be built to reduce transit time for stocking as well as retrieving products in the warehouse for optimizing space utilization and warehouse operations.
12). Reinforcement Learning and optimization techniques are utilized to assess the security of the electric power systems and to enhance Microgrid performance. Adaptive learning methods are employed to develop control and protection schemes.

Ques 12. Discuss Various Types of Reinforcement Learning Techniques.

Ans : Reinforcement learning are of following three types :

(a). Passive Reinforcement (b) Temporal Difference Learning (c) Active Reinforcement learning.

Passive Reinforcement Learning: In this technique agent's policy is fixed and the task to learn the utilities of state action pairs. If policy is π and state is S , then agent always executes the action $\pi(S)$.

- Goal is to learn how good policy is i.e to learn th e utility function U^{π}(S). Passive learning agent is not aware of the transition model T (S , a , S') , which specifies probability of reaching state S' from state S after action a.
- Passive learning also not knows the Reward Function R (S).
- A **utility** is defined to be the expected sum of rewards obtained if policy π is followed.

$$\mathbf{U^{\pi}(S) = E [\sum_{t=0}^{\infty} \gamma^{t} R (S_t) : \pi , S_0 = S] , where \gamma is a discount factor.}$$

Temporal difference Learning: When a transition occurs from state S to state S', we update U^{π}(S) as following: $\mathbf{U^{\pi}(S) \leftarrow U^{\pi}(S) + \alpha (R (S) + \gamma U^{\pi}(S') - U^{\pi}(S)) .}$

$\boldsymbol{\alpha}$ **:** Learning rate parameter. This update rule uses the difference in utilities between successive states, it is often called **TEMPORAL DIFFERENCE** equation.

Active Reinforcement Learning: The compression achieved by a function approximator allows the learning agent to generalize from states it has visited to states it has not visited.

E.g : An evaluation function for CHESS that is represented as a weighted linear function of a set of features or a basis function $\mathbf{f_1 , f_2 , \ldots\ldots f_n}$.

$$\hat{U}_{\theta}(s) = \theta_1 f_1(s) + \theta_2 f_2(s) + \cdots + \theta_n f_n(s)$$

Where θ_i: is the coefficient we want to learn and

f_i: is feature extracted from state.

Ques 13. What is Decision Tree Learning? Why it is useful in AI applications?

Ans : Decision tree method is one of the most simplest and yet most successful forms of learning algorithm. It emphasis is towards the area of Inductive Learning. In inductive learning " a collection of examples of f is given , we return a function h that approximates f ", where example f is " A pair (x , f(x)) ", where x is input and f (x) is output of function applied to x.

- H is hypothesis. A good hypothesis will generalize well i.e will predict examples correctly.
- A decision tree takes input an object , with certain feature set and returns a decision of predicted output value. ***Output may be Discrete or Continuous***.
- ***Learning a discrete function is known as classification learning, wheras learning a continuous function is termed as Regression in decision tree.***
- Decision tree reaches itsdecision by performing a sequence of tests.
- Each internal node is a test value of one of the properties and branches from node are labeled with possible values of the test.
- Each leaf node consists of return value.
- Application of Decision Tree learning is in designing an expert System based on Decision Tree Architecture.
- Decision trees are completely expressive with the class of propositional logic.
- Various propositions are connected via logical OR operator(V).

Example : $\forall s\, F_1(s) \rightarrow (P_1(s) \vee P_2(s) \vee \ldots\ldots \vee P_n(s))$

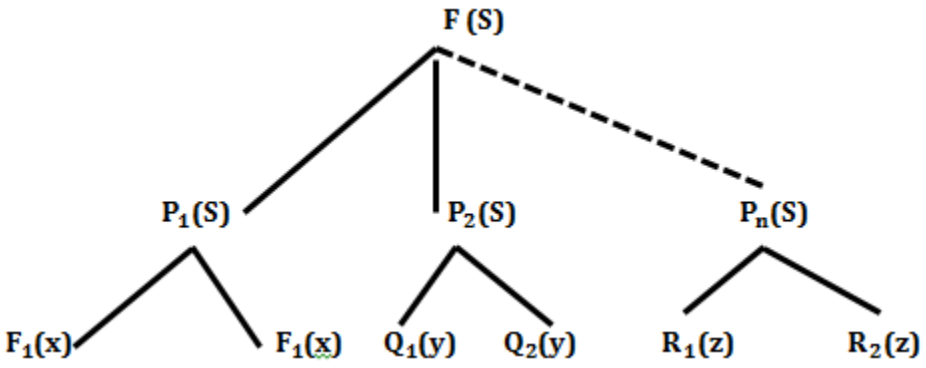
$\forall x\, P_1(x) \rightarrow (F_1(x) \vee F_2(x))$

$\forall y\, P_2(y) \rightarrow (Q_1(y) \vee Q_2(y))$

……so on for Pn (s).

$\forall z\, P_n(z) \rightarrow (R_1(z) \vee R_2(z))$

A general decision tree for above propositional formula can be as given below:



Boolean Decision Trees : This technique consists of a vector of input attributes , X and a single Boolean output Y.

E.g : Set of examples (X_1 , Y_1….., (X_6 , Y_6)).

- Positive examples are those in which goal is true.
- Negative examples are those in which goal is false.
- Complete set is known as a **TRAINING SET.**

a) In case of numeric attributes, decision trees can be geometrically interpreted as a collection of hyper planes , each orthogonal to one of the axes.

b) The tree complexity has a crucial effect on its accuracy. It is explicitly controlled by the stopping criteria used and the pruning method employed.

c) Usually the tree complexity is measured by one of the following metrics: ***the total number of nodes, total number of leaves, tree depth and number of attributes used.***

d) Decision tree induction is closely related to rule induction. Each path from the root of a decision tree to one of its leaves can be transformed into a rule simply by conjoining the tests along the path to form the antecedent part, and taking the leaf's class prediction as the class value.

Example:

- Given this classifier, the analyst can predict the response of a potential customer (by sorting it down the tree), and understand the behavioral characteristics of the entire potential customers population regarding direct mailing.
- Each node is labeled with the attribute it tests, and its branches are labeled with its corresponding values.
- For example, one of the paths in below figure can be converted into the rule ***:"If customer age is is less than or equal to or equal to 30, and the customer is "Male" – then the customer will respond to the mail".***

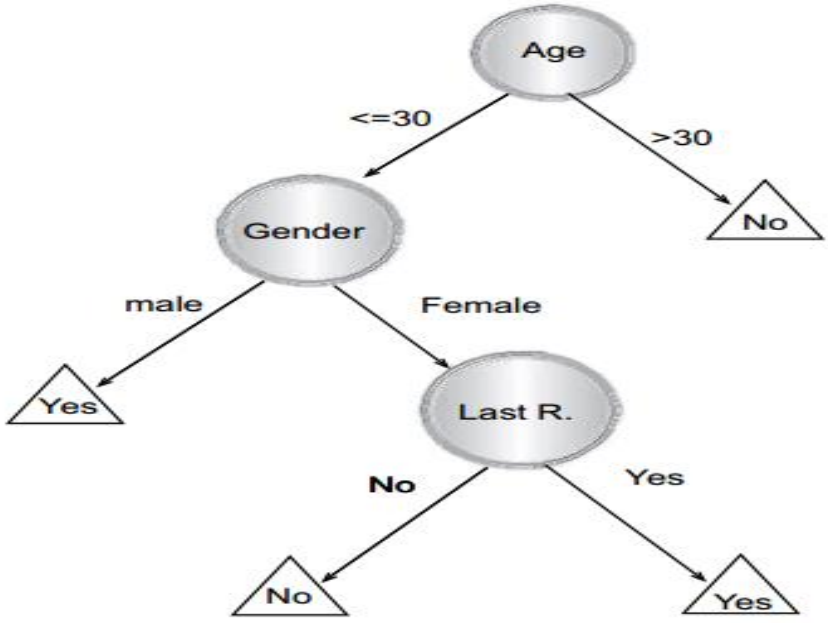


Figure 9.1. Decision Tree Presenting Response to Direct Mailing.

Application Areas of Decision Tree Learning

1) **Variable selection**: The number of variables that are routinely monitored in clinical settings has increased dramatically with the introduction of electronic data storage. Many of these variables are of marginal relevance and, thus, should probably not be included in data mining exercises.
2) **Handling of missing values:** A common - but incorrect - method of handling missing data is to exclude cases with missing values; this is both inefficient and runs the risk of introducing bias in the analysis. Decision tree analysis can deal with missing data in two ways: it can either classify missing values as a separate category that can be analyzed with the other categories or use a built decision tree model which set the variable with lots of missing value as a target variable to make prediction and replace these missing ones with the predicted value.
3) **Prediction**: This is one of the most important usages of decision tree models. Using the tree model derived from historical data, it's easy to predict the result for future records.
4) **Data manipulation**: Too many categories of one categorical variable or heavily skewed continuous data are common in medical research.

Ques 14 : Write Short Notes on the following : (A) Regression Trees

(B) Bayesian Parameter Learning.

Ans : Regression Trees : Regression trees are commonly used to solve the problems where target variable is numerical / continuous instead of discrete. Regression trees posses following properties :

a) Leaf nodes predict the average value of all instances.

b) Splitting criteria : Minimize the variance of the values in each subset $\mathbf{S_i}$

c) Standard Deviation Reduction : $\mathbf{SDR\ (A, S) = SD\ (S) - \sum_i \frac{|S_i|}{|S|} SD\ (S_i)}$

d) Termination Criteria: Lower bound on SD in a node and Lower bound on number of examples in a node.

e) Pruning criteria is Mean Squared Error.

Bayesian Parameter Learning: This learning technique works on parametric variables which are random having some prior distribution. An optimal learning classifier can be designed using "Class conditional densities", p (x | w_i). In a typical case we merely have some unclear knowledge about situations with given number of samples and training. Observation of samples converts this to a posteriori density, and true values of parameters are revised. In Bayesian learning sharpening of Posteriori Density Function is done, causing it to peak near the true values.

- We assume priors are known: **P (ω_i | *D*) = P(ω_i).**
- Also, assume functional independence :

$$p(\omega_i \mid \mathrm{x}, D) = \frac{p(\mathrm{x}|\omega_i, D)P(\omega_i)}{\sum_{j=1}^{c} p(\mathrm{x}|\omega_j, D)P(\omega_j)}$$

- Any information we have about about θ prior to collecting samples is contained in **p(*D*|θ).**
- Observation of samples converts this to a posterior, **p(θ|*D*),** which we hope is peaked around the true value of θ.
- ***Our goal is to estimate a parameter vector***:

$$p(\mathrm{x}|D) = \int p(\mathrm{x}, \theta|D)d\theta$$

- We can write the ***joint distribution*** as a product**:**

$$p(\mathrm{x}|D) = \int p(\mathrm{x}|\theta, D)p(\theta|D)d\theta$$
$$= \int p(\mathrm{x}|\theta)p(\theta|D)d\theta$$

[END OF 4th UNIT]