Thursday
18-6-2020

AI Assignment - 5

UNIT-5

N.V.S.K. kalyani
17U41A0584
3rd B.Tech CSE-C

2 marks

1A) Types of learning mechanisms:-

There are 3 forms of learning. Those are:

i, Supervised learning: It can take what it has learned in the past & apply that to new data using labelled examples to predict future pattern and events.

ii, Unsupervised learning:

unsupervised learning tasks find patterns where we don't. This may be because the "right answers" are unobservable.

iii, Reinforcement learning:- It is a type of dynamic program that trains algorithms using a system of reward and punishment.

2A) Inductive learning:- System tries to Induce a "general rule" from a set of observed instances. Supervised learning algorithm is given the correct value of the functions for particular Inputs, and changes its representation of the function to try to match the Information provided by the feedback

Ex:- Is a pair(x, f(x)), where x is the Input & f(x) is the output of the function applied to x.

for each t such that $N_{s'|sa}[t,s,a]$ is nonzero d

$p(t|s,a] \leftarrow N_{s'|sa}[t,s,a] / N_{sa}[s,a]$

$U \leftarrow$ POLICY-EVALUATION (π, U, mdp)

if s'.TERMINAL? then s, a $\leftarrow$ null else s, a $\leftarrow$ s', π[s

return a

5A) List statistical learning:-

1. parameter learning:- task involves finding the numerical parameters for a probability model whose structure is fixed.

2. Naive Bayes learning:- the model is 'naive' because it assumes that the attributes are conditionally Independent of each other given the class.

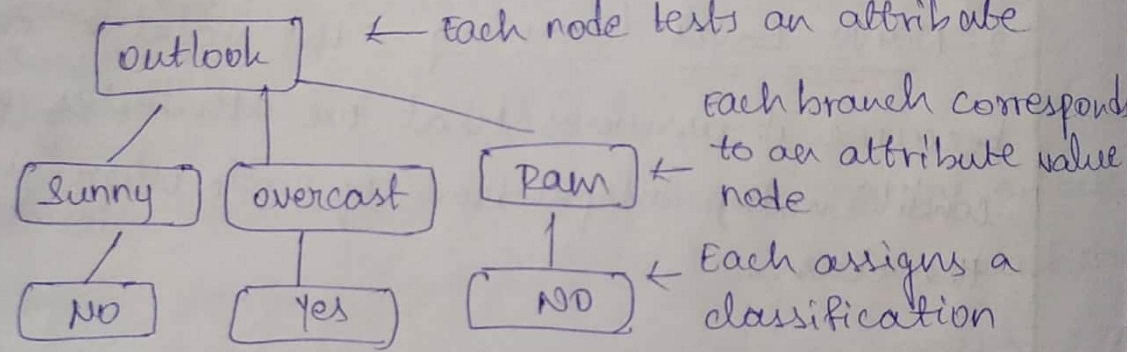
10 marks

1A) Decision trees:-

→ It represents a function that takes as i/p a vector of attribute values and returns a "decision" a single o/p value.

→ Decision tree algorithm falls under the category of supervisor learning. They can be used to solve both regression & classification problems.

→ A decision tree reaches its decision by performing a sequence of tests.



Algorithm:-

~~ID3 (Iterative Dichotomiser 3)~~

function DECISION-Tree-LEARNING (examples, attributes, parent_examples) return a tree

If examples is empty then return PLURALITY_value (parent_examples)

else if all examples have the same classification then return classification

else if attributes is empty then return PLURALITY-value (ex)

else

$A \leftarrow \underset{a \in attributes}{argmax}$ Importance(a, examples) tree ← a new decision tree with root test A

for each value V_k of A do

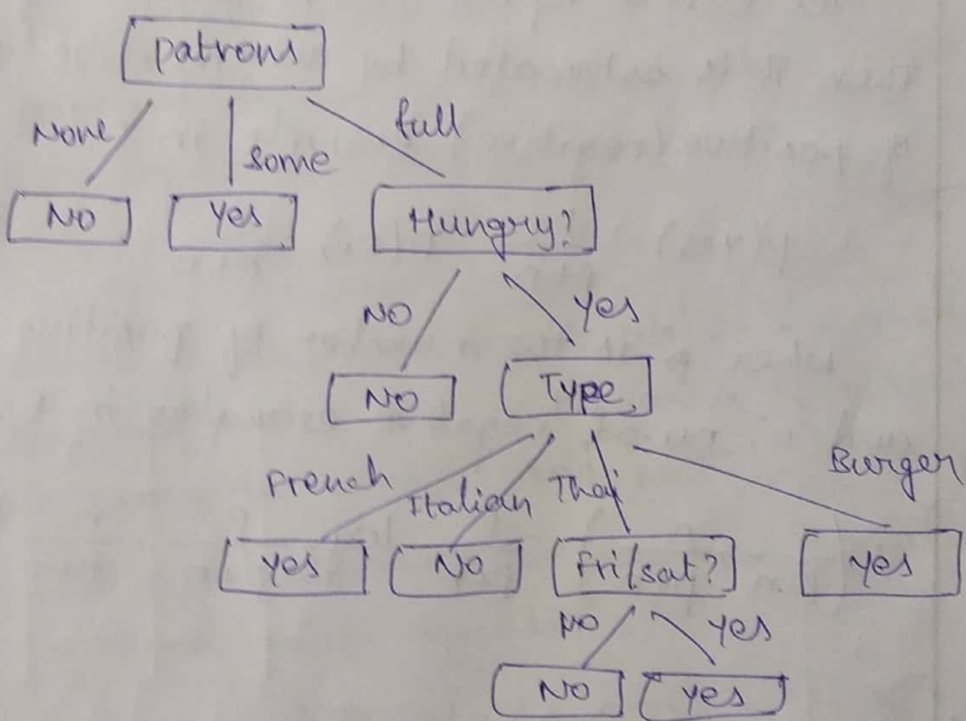
exs ← {e : e∈ examples and e.A = V_k}

subtree ← DECISION-TREE-LEARNING (exs, attributes - A, examples)

add a branch to tree with label (A = V_k) and subtree subtree

return tree

<u>output of the learning algorithm</u>



<u>choosing attributes set</u>:-

* The scheme used in decision tree learning for selecting attributes is designed to minimize the depth of final tree.
* A perfect attribute divide the examples into sets that are all positive an all negative
* In general, if the possible an V_i have probability

17UN1A0524

$P(v_i)$ then the Information content I of the actu answer is given by

$$I(P(v_1), \ldots P(v_n)) = \sum_{i=1}^{n} -P(v_i) \log_2 P(v_i)$$

Information content of a Decision tree :-

For decision tree, the event is question that is whether the tree will answer yes or m to a given input example.

let E is a representative sample of the domain Then it is estimated by the relative frequency of positive (negative) example in E.

$$p(yes) = \frac{p}{p+n} \quad p(no) = \frac{n}{p+n}$$

wher 'p' is the number of positive examples and 'n' no.of negative examples in E.

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

2A] Passive learning:-

1. It is observed in fully observable environment
2. In passive learning, the agents policy 'π' is fixed in state s, it always executes the action - $\pi(s)$.
3. It is goal is simple to learn how good the policy is that is, to learn utility function $U^{\pi}(s)$.

Scanned with CamScanner

3	→	→	→	+1
2	↑		↑	-1
1	↑	←	←	←
	1	2	3	4

3	0.412	0.868	0.418	+1
2	0.762		0.660	-1
1	0.705	0.855	0.611	0.388
	1	2	3	4

The equation is

$$U^{\pi}(s) = E\left[\sum_{i=n}^{\infty} \gamma^t R(s_t)\right]$$

$R(s)$ = Reward for a state

s_t = state reached at time t.

A discount factor 'γ' in all equations, but for the 4x3 world it is set as $\gamma = 1$.

Direct utility Estimation :-

* A simple method for direct utility estimation was invested in the late in the area of adaptive control theory by window and not^{-1}
* The utility values obey the Bellman equations for a fixed policy equations.

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) \, U^{\pi}(s')$$

Adaptive Dynamic Equation :-

ADP agent works by learning along the transition model of the environment as it goes along and solving the corresponding makel decision process using a dynamic programming method.

Temporal difference learning:

But in, TD learning, doesn't requires the agent to learn the transition model

$$U^{\pi}(s) \leftarrow U^{\pi}(s) + \alpha\left(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s)\right)$$

α is the learning rate parameter.

Active reinforcement learning:-

* Active RL is nothing but ADP with explanation function.
* An active agent must decide what actions to take. Bellman equation as given below

$$U(s) = R(s) + \gamma \max_{a} \sum p(s' \mid s,a)\, U(s')$$