CMPSCI 683 Artificial Intelligence
Questions & Answers

1. General Learning

Consider the following modification to the restaurant example described in class, which includes missing and partially specified attributes:

⇒ The outcomes for X1 and X7 are reversed.

⇒ X3 has the missing attribute value for "Pat".

⇒ X5 has the missing attribute value for "Hun".

⇒ X10 has the attribute for “TYPE” which could be either ITALIAN or FRENCH.

Define an algorithm for dealing with missing attributes and partially specified attributes,which includes the modified calculation for information gain use to make splitting decisions.

Generate a decision tree for this example using your new algorithm.

Answer

There are a lot of ways of answering this question. One algorithm is as follows:

For a training instance with multi-valued attributes, I will duplicate that instance by the number of values of that attribute. But each duplicated instance will be weighted down by the number of times I have seen each value in other training examples.

For example, in the restaurant example, X10 will now become X10’ and X10’’. X10’ will have a value of French, with a weight of 2/3 (note this is 2/3 because there are only 3 examples with either French or Italian of which 2 are French). X10’’ will have a weight of 1/3 when learning in my decision tree.

For a missing attribute, I will treat it like a multi-valued attribute, using all possible values of the missing attribute.

For example, X3 will become X3’, X3’’ and X3’’’. X3’ will have the value None with a weight of 2/11. X3’’ will have the value Some for Pat, with a weight of 3/11. X3’’’ will have the value Full for Pat, with a weight of 6/11. Note, that these weights are independent of each other. So, if X10 also had the value of Pat missing, I would have to generate 6 new training instances. X10’ would be French for Type and None for Pat, with a weight of 2/3 * 2/11.

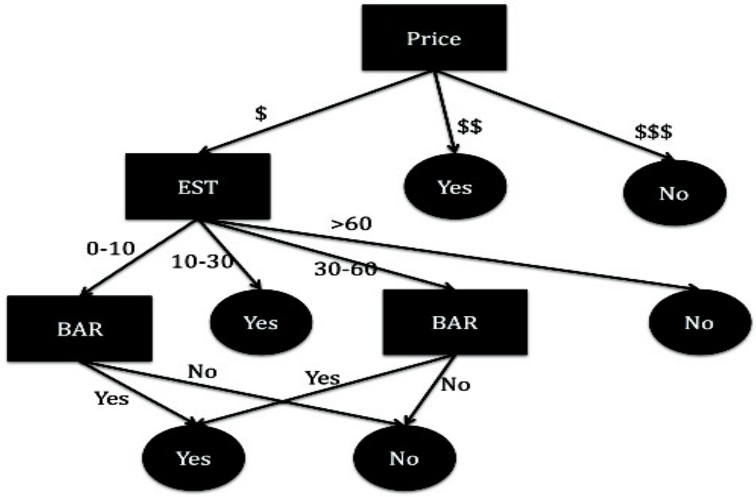
We can now use this modification of the algorithm given in class to compute our decision tree by calculating the information gain. I will show the formulas for some of the important ones here

FRI = 5/12 * I(2/5,3/5) + 7/12 * I(4/7,3/7)

HUN = (7 + 7/11)/12 * I[4/(7+7/11),(3 + 7/11)/(7+7/11)] + (4 + 4/11)/12 * I[2/(4+4/11),(2+4/11)/(4+4/11)]

TYPE = 4/12 * I(2/4,2/4) + 4/12 * I(3/4,1/4) + (2+2/3)/12 * I(1,0) + (1+1/3)/12 * I[1/(1+1/3),1/3/(1+1/3)]

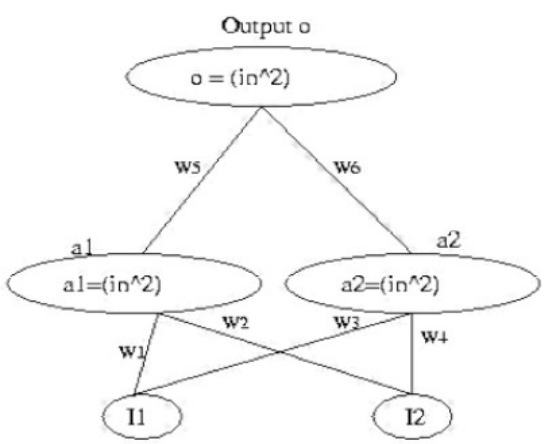
And so on. The final decision tree will look something like this



2. Neural Networks

Apply the back-propagation algorithm to the following network.
Your training example is:
E1: I1 = 1, I2 = 3, o=0;

All weights are set to value 1 initially; learning rate is set to 0.2.
Show the updated weight value after the training of the above two examples.
(In the figure below, "in^2" means the square of the sum of the inputs.)



Answer

g = in^2
g' = 2 * in
To answer this question, I will normalize my inputs.
I1 and I2 will become 0.25 and 0.75 respectively. A1 and a2 are both equal to 1 then.
I will normalize them again. A1 and A2 will become 0.5 and 0.5 respectively. O = 1.
In my training example, o was 0. I will compute the new weights using the back propagation method
Δo = (Training Output – Calculated output) * g'(input)
= (0 – 1) * 2 * 1 = -2

w5 = w5 + 0.2 * input a1 * Δo = 1 – 0.2 * 0.5 * 2 = 1-0.2 = 0.8
Similarly w6 is also 0.8
Δa1 = g'(input) * w5 * Δo = 2 * 1 * 0.8 * -2 = -3.2
w1 = w1 + 0.2 * I1 * Δa1 = 1 – 0.2 * 0.25 * 3.2 = 0.84
Similarly you can calculate the rest of the weights.

3. Reinforcement Learning

Consider the Markov Decision Process below. Actions have non-deterministic effects, i.e., taking an action in a state returns different states with some probabilities. There are two actions out of each state: D for development and R for research.

Consider the following deterministic ultimately-care-only-about-money reward for any transition resulting at state:

State	S1	S2	S3	S4
Reward	0	100	25	50

Assume you start with state S1 and perform the following actions:

- Action: R; New State: S3
- Action: D; New State: S2
- Action: R; New State: S1
- Action: R; New State: S3
- Action: R; New State: S4
- Action: D; New State: S2

a) Assume V(S) for all S = S1, S2, S3 and S4 is initialized to 0. Update V(S) for each of the states using the Temporal Difference Algorithm.

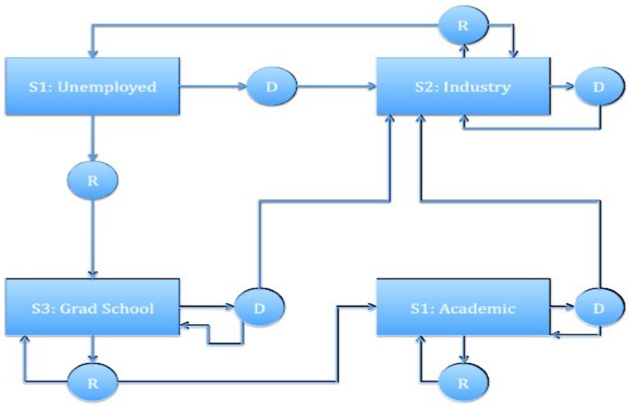
Answer

Here you will use the following algorithm
V(s) = (1 - "learning rate")V(s) + "learning rate" * [R(S) + "discount factor" * V(s')]
Here s' is the "New State".
An example calculation is as follows:
V(S1) = (1 – 0.9) * 0 + 0.9 * [25 + 0.9 * 0] = 0.9 * 25 = 22.5

b) What would your policy be at the end of training?

Answer

A policy of all Ds would for the most part be appropriate. It will bring you to the Industry state (S2), which is the state with the most reward for staying in. The figure of the policy is as follows.



4. Utility Theory

In class we discussed an example of decision trees that involved buying a car with tests that could be performed to assess the necessity of repair.

What would be the value of information for performing each of the tests in the context of the existence of the other test? What would be the value of information associated with the two tests?

Answer

First, we need to calculate some probabilities using Bayes rule as we can't use the given conditional probabilities directly.

Given:

$P(c1 = good) = 0.7$

$P(c2 = good) = 0.8$

$P(T1 = pass|c1 = good) = 0.8$

$P(T1 = pass|c1 = bad) = 0.35$

$P(T2 = pass|c2 = good) = 0.75$

$P(T2 = pass|c2 = bad) = 0.3$

Transformed using Bayes rule:

$P(c1 = good|T1 = fail) = 0.418$

$P(c1 = bad|T1 = fail) = 0.582$

$P(c1 = good|T1 = pass) = 0.842$

$P(c1 = bad|T1 = pass) = 0.158$

$P(c2 = good|T2 = fail) = 0.588$

$P(c2 = bad|T2 = fail) = 0.412$

$P(c2 = good|T2 = pass) = 0.909$

$P(c2 = bad|T2 = pass) = 0.091$

$P(T1 = fail) = 0.335$

$P(T1 = pass) = 0.665$

$P(T2 = fail) = 0.340$

P(T2 = pass) = 0.660

To determine the value of the tests I deducted the price whichever test we perform first from the profits, since we would have paid that already.

The case where we have already done test 1 is shown in Fig. 1. We can see that the expected profit is the same no matter if we do test 2 or not, so the value of information of test 2 given that we have already done test 1 is 0. As test 2 is not free, we would never do test 2 after test 1.

The case where we have already done test 2 is shown in Fig. 2. The expected profit with test 2 only is 270, if we add test 1 it is 312.6., so the value of information of test 1 is 42.6. Unfortunately, test 1 in not free either but comes at a price of 50, so we would never do test 1 after test 2 either.

Without any of the two tests, the expected profit is max {500 * 0.7 – 200 * 0.3 = 290, 250 * 0.8 + 100 * 0.2 = 220} = 290.

With both tests (assuming they are both free) it’s 332.6, so the value of information of both tests is 42.6.

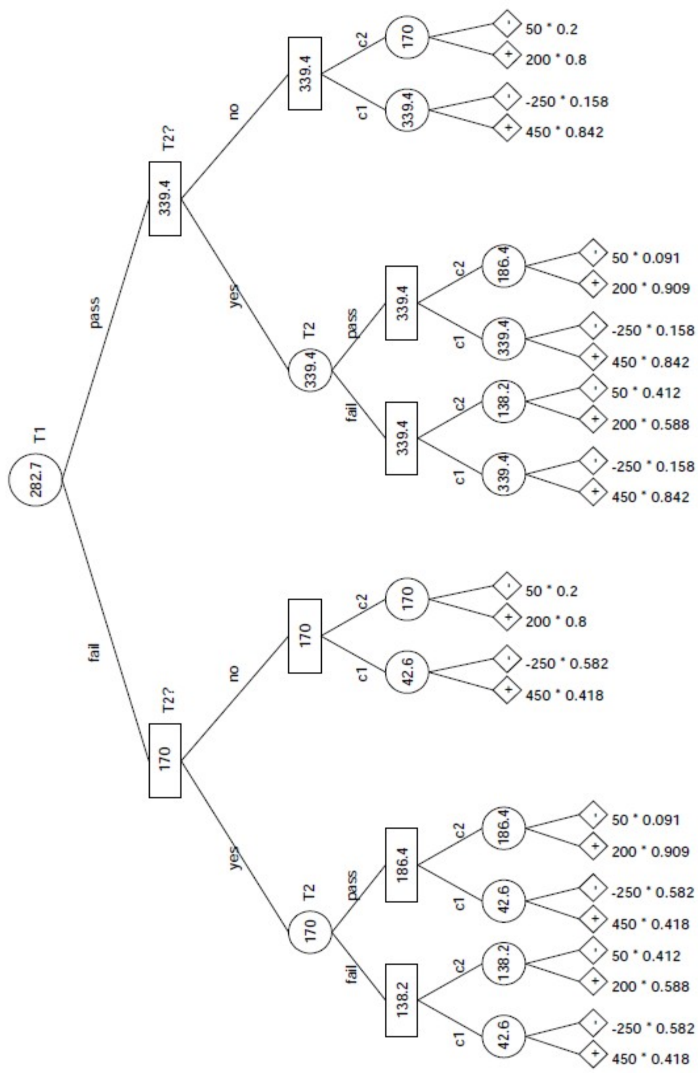


Figure 1: Value of test 2 given that we have already done test 1

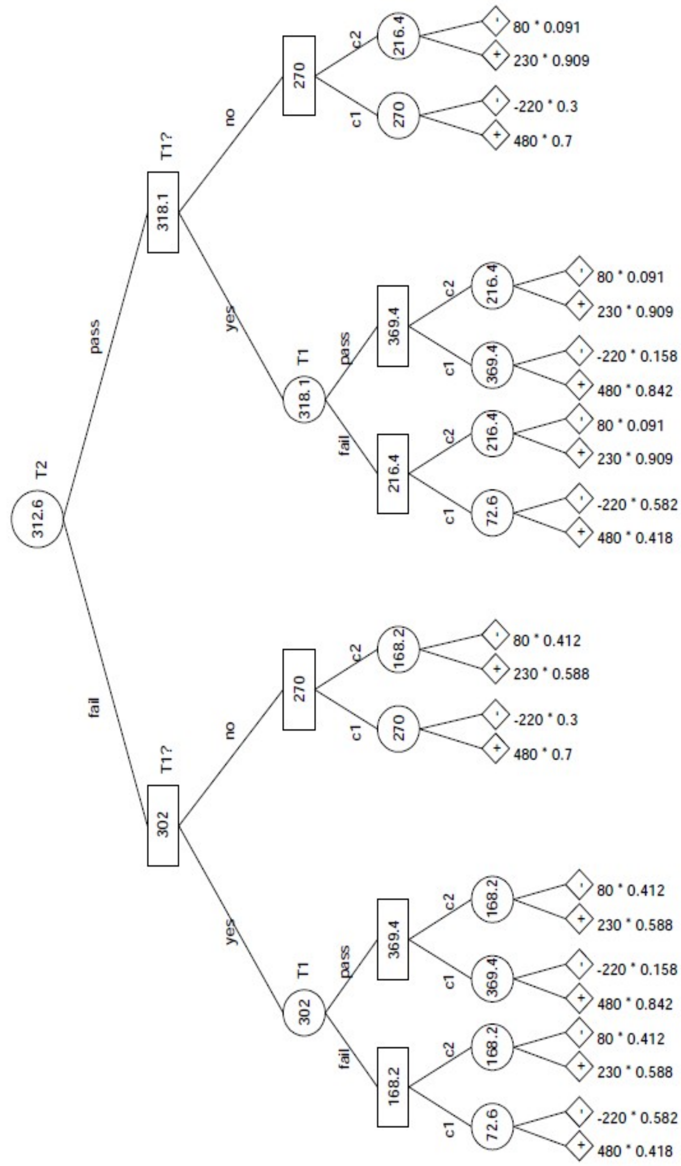


Figure 2: Value of test 1 given that we have already done test 2

5. Decision Trees

The following is a modification of one of the examples presented in class.

There are two candidate cars C1 and C2. Each can either be of good quality or badquality.

There are two possible tests. T1 on C1 costs \$100. T2 on C2 costs \$50.

C1 costs \$1500, which is \$500 below market value. If C1 is of bad quality, the repair cost is \$700

C2 costs \$1150, which is \$250 below market value. If C2 is of bad quality, the repair costs \$150.

The chance that car C1 is of good quality is 70%.

The chance that car C2 is of good quality is 80%.

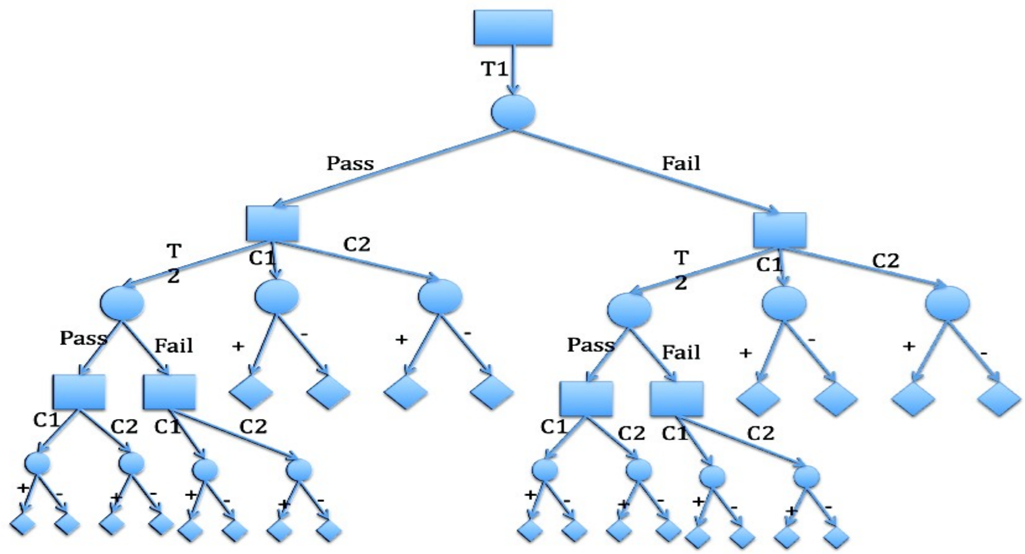
Test T1 on C1 will confirm good quality with probability 80% and bad quality with probability 65%.

Test T2 on C2 will confirm good quality with probability 75% and bad quality with probability 70%.

The buyer must buy one of the two cars and can perform T1, T2, T1 and T2 simultaneously, T1 followed by T2, or T2 followed by T1. The total cost of performing the two tests simultaneously is \$125 (this includes the cost of the T1 and T2) and the total cost of performing one test followed by the other is \$175 (again including the cost of performing T1 and T2).

1) Build a decision tree for this problem. Only draw the branch that corresponds to T1 being performed first. Note: You do not have to repeatedly draw similar branches. Just draw a representative branch and explain what the other branches might be that you did not end up drawing.

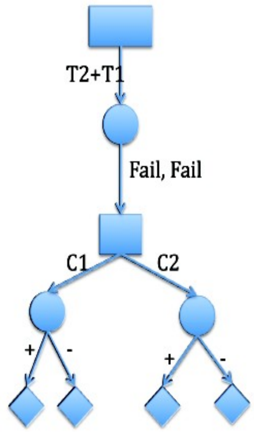
Answer



2) Consider an example branch from my decision tree. Determine the utility of each node in this branch. This is one of the branches for performing the test T1 and T2 simultaneously.

Answer

The utility of the + branch of chance node C1
= 500 – 125 = 375
The utility of the – branch of chance node C1
= 500 – 700 – 125 = -325
The utility of the chance node C1
= P(C1|~T1) * 375 + P(~C1|~T1) * (-325)
Note here that C1 is independent of C2 or T2.
P(C1|~T1) = P(~T1|C1) * P(C1) / P(~T1)
P(~T1) = P(~T1|C1) * P(C1) + P(~T1|~C1) * P(~C1)
= 0.2 * 0.7 + 0.65 * 0.3
= 0.335
P(C1|~T1) = 0.2 * 0.7 / 0.335 = 0.418
The utility of the chance node is -32.4
The utility of the + branch of chance node C2
= 250 – 125 = 125
The utility of the – branch of chance node C1
= 250 – 150 – 125 = -25



The utility of the chance node C1
= P(C2|~T2) * 125 + P(~C2|~T2) * (-25)
P(C2|~T2) = P(~T2|C2) * P(C2) / P(~T2)
P(~T2) = P(~T2|C2) * P(C2) + P(~T2|~C2) * P(~C2)
= 0.25 * 0.8 + 0.70 * 0.2
= 0.34
P(C1|~T1) = 0.25 * 0.8 / 0.34 = 0.588
The utility of the chance node is 63.2
This 63.2 value will then be propagated all the way to the top.

6. Probabilistic Inference

(a) Prove the conditionalized version of the general product rule:

P(A,B|E) = P(A|B,E)P(B|E)

Answer

$$\begin{aligned}\mathbf{P}(A,B|E) &= \frac{\mathbf{P}(A,B,E)}{\mathbf{P}(E)} \\ &= \frac{\mathbf{P}(A|B,E)\mathbf{P}(B,E)}{\mathbf{P}(E)} \\ &= \frac{\mathbf{P}(A|B,E)\mathbf{P}(B|E)\mathbf{P}(E)}{\mathbf{P}(E)} \\ &= \mathbf{P}(A|B,E)\mathbf{P}(B|E)\end{aligned}$$

(b) Prove the conditionalized version of Bayes' rule:

Answer

$$\mathbf{P}(A|B,C) = \frac{\mathbf{P}(B|A,C)\mathbf{P}(A|C)}{\mathbf{P}(B|C)}$$

$$\begin{aligned}\mathbf{P}(A|B,C) &= \frac{\mathbf{P}(B,C|A)\mathbf{P}(A)}{\mathbf{P}(B,C)} \\ &= \frac{\mathbf{P}(B,C|A)\mathbf{P}(A)\mathbf{P}(C)}{\mathbf{P}(B|C)} \\ &= \frac{\mathbf{P}(A,B,C)\mathbf{P}(A)\mathbf{P}(C)}{\mathbf{P}(A)\mathbf{P}(B|C)} \\ &= \frac{\mathbf{P}(B|A,C)\mathbf{P}(A,C)\mathbf{P}(C)}{\mathbf{P}(B|C)} \\ &= \frac{\mathbf{P}(B|A,C)\mathbf{P}(C)\mathbf{P}(A|C)}{\mathbf{P}(B|C)\mathbf{P}(C)} \\ &= \frac{\mathbf{P}(B|A,C)\mathbf{P}(A|C)}{\mathbf{P}(B|C)}\end{aligned}$$

Belief Networks

Orville, the robot juggler, drops balls quite often when its battery is low. In previous tests, it has been determined that the probability that it will drop a ball when its battery is low is 0.9. Whereas when a battery is not low, the probability that it drops a ball is only 0.05. The battery was recharged not so long ago, and our best guess (before looking at Orville's latest juggling record) is that the odds that the battery is low are 9 to 1 against.

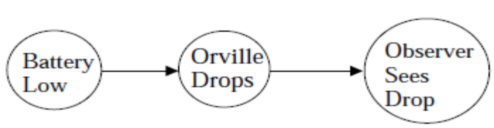
A robot observer, with a somewhat unreliable vision system, reports that Orville dropped a ball. The reliability of the observer is given by the following probabilities:

P(observer says that Orville drops j Orville drops) = 0.8

P(observer says that Orville drops j Orville does not drop) = 0.2

Draw the Bayes network, and calculate the probability that the battery is low given the observer's report.

Answer



Let BL be battery low, OD be Orville drops ball, and OSD be that the observer sees Orville drop the ball. Some probabilities that will be used in the calculations below: P(:B) = 0:1,

$$\begin{aligned}
\mathbf{P}(BL|OSD) &= \frac{\mathbf{P}(OSD|BL)\mathbf{P}(BL)}{\mathbf{P}(OSD)} \\
&= \alpha \mathbf{P}(OSD|BL)\mathbf{P}(BL) \\
&= \alpha \sum_{OD_i} \mathbf{P}(OSD|OD_i)\mathbf{P}(OD_i|BL)\mathbf{P}(BL) \\
&= \alpha \left[\mathbf{P}(OSD|OD)\mathbf{P}(OD|BL)\mathbf{P}(BL) + \mathbf{P}(OSD|\neg OD)\mathbf{P}(\neg OD|BL)\mathbf{P}(BL)\right] \\
&= \alpha \left[0.8 * 0.9 * 0.1 + 0.2 * 0.1 * 0.1\right] = 0.074\alpha
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}(\neg BL|OSD) &= \frac{\mathbf{P}(OSD|\neg BL)\mathbf{P}(\neg BL)}{\mathbf{P}(OSD)} \\
&= \alpha \mathbf{P}(OSD|\neg BL)\mathbf{P}(\neg BL) \\
&= \alpha \sum_{OD_i} \mathbf{P}(OSD|OD_i)\mathbf{P}(OD_i|\neg BL)\mathbf{P}(\neg BL) \\
&= \alpha \left[\mathbf{P}(OSD|OD)\mathbf{P}(OD|\neg BL)\mathbf{P}(\neg BL) + \mathbf{P}(OSD|\neg OD)\mathbf{P}(\neg OD|\neg BL)\mathbf{P}(\neg BL)\right] \\
&= \alpha \left[0.8 * 0.05 * 0.9 + 0.2 * 0.95 * 0.9\right] = 0.207\alpha
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}(BL|OSD) + \mathbf{P}(\neg BL|OSD) &= 1 \\
0.074\alpha + 0.207\alpha &= 1 \\
\alpha &= 3.5587
\end{aligned}$$

$\mathrm{P}(BL|OSD) = 0.074\alpha = 0.074 * 3.5587 = 0.263.$

The probability that Orville's battery is low given that the observer saw Orville drop the ball, is 0.263.