**CS1571  
Assignment 3**

**Simran Gidwani**

**Due:** Friday, Dec. 6

**Part A – Decision Trees (60 points)**

Say you are building a customer service spoken dialogue chatbot. Each time the chatbot responds to the user is denoted by a *conversational turn.* In each turn, the chatbot has two simple choices: respond to the user’s query (**Respond**), or direct the user to a human customer service representative (**Redirect**). Consider the following decision tree:

10

-60

K=2

10

19

-65

K=1

19

Respond

Redirect

Respond

Redirect

(~Resolved)-.9

(Resolved)-.10

(Frustrated)-.30

(~Frustrated)-.7

(~Resolved)-.9

(Frustrated)-.30

(~Frustrated)-.7

C1

C2

C3

C4

D2

D1

U1: 100

U2: 5(1)

U3: -100+5(1)

U4:0

U5:100

U6: 5(2)

U7: -100+5(2)

You’ll notice there are two different types of chance nodes. At this point, assume probabilities are the same at every chance node.

* *Resolved* represents the probability that a problem will be resolved after a given turn. P(Resolved) = .10
* *Frustrated* represents the probability that the user was frustrated with the dialogue system on the previous turn. P(Frustrated) = .30

Assume a utility function *U,* and that the number of turns taken so far is represented by *k*. At the first decision made by the chatbot, *k*=1*.* The utility value for redirecting a user to a human if they are NOT frustrated is -100+5*k*. The utility value for redirecting a user to a human if they are frustrated is 5*k*. The utility value for resolving the user’s problem is 100. If your tree has a set number of turns (above the maximum number of terms is 2), assume that the utility value of continuing to respond without resolving the problem or redirecting the user is 0. Note, if the problem is resolved or the user is redirected to a human, no further turns are taken by the chatbot.

1. (10 pts) Given the decision tree above, fill out the expected value of each chance node in the below table, the utility value for each of the outcome nodes. Then, describe the best course of action for the chatbot on the first turn.

|  |  |
| --- | --- |
| Node | Value |
| U1 | 100 |
| U2 | 5 |
| U3 | -95 |
| U4 | 0 |
| U5 | 100 |
| U6 | 10 |
| U7 | -90 |
| C1 | 19 |
| C2 | -65 |
| C3 | 10 |
| C4 | -60 |

Best course of action:

Best course of action would be for the chatbot to respond with the expected value of 19 vs -65 and if getting to turn two, should decide to again respond leading to an expected value of 10 rather than -60.

1. (30 pts) Implement Expectimax for this particular scenario and test it on these two turns. Does the action returned by the function match your answer? **Submit your code and output according to the instructions in the rubric document.** In the box below, as described in the rubric document, write the exact function call to use with the exact parameters, and the exact output we are supposed to get when we call your function.

Action returned does match the results of my hand computation

expectiMax(iterations): where iterations == 2

output:

C1-19.0

D2-10.0

C3-10.0

U5-100

1. (10 pts) Assume the tree is cut off at 30 levels rather than 2. Given the current reward function and probabilities, at which turn is it better to stop the conversation and redirect the user to a human? You can derive your answer through reasoning or code. If using reasoning, show below how you got your answer. If using code, submit the exact function call and the output of your function call (as described in the A3 rubric).

If the tree was cut off at 30 levels rather than 2, given the previous utility function, at turn 16 is when it is better to stop the conversation and redirect the user to a human. Starting at turn 17, the utility value for redirecting becomes higher than the utility value for continuing to respond. I was able to derive this answer using the code and examining at what turn value produced a change in the decisions made at each decision node in the future.

(.7)(-100+5k) + (.3)5k > 100\*.10 + 0\*.90

-70 + 5k > 10

5k > 80

K > 16

1. **(**5 pts) Write a new utility function *U’* that prioritizes shorter conversations than the current function (redirecting the user to a human sooner). That is, the function should redirect the user to a human sooner in the conversation than the current function. Explain why the utility function achieves the requirements and is reasonable.

U’ = -100 + 15k for Not Frustrated

= 5k for Frustrated

= 100 for Resolved

= 0 for ~Resolved

Changes it to redirect at turn 7 rather than turn 17

1. (5 pts) Assume the chatbot always has the option to ask a person if they’d like to be directed to a human prior to deciding to redirect them. You can assume if people say yes there is a higher probability they are frustrated, and if they say no, they are likely not frustrated – but sometimes people will say yes even if they are not frustrated or no if they are frustrated. Name two different changes you would make to the decision tree, utility function, or probabilities given that action, and justify why your changes make sense based on how these interactions are likely to unfold in the real world.

One thing I would change about the Decision Tree if the chatbox had the option to ask whether they would like to be redirected would be adding a branch after ~Resolved that has two branches for yes and no (user responses to being frustrated or not) and from both of those branches would consist the same branches after the D2 node but one for if the user responds yes and one for if the user responds no. If the user responds yes, the I would change the probability of Frustrated to .70 and the probability of ~Frustrated to .30. If the answer was no than I would leave the probabilities the same.

Adding this change would give a more accurate representation of whether a human user is frustrated during a call with a chatbox. When being asked whether they would like to redirect or not there is a higher chance that if the user says yes that they are frustrated with the chatbot so therefor increasing the probability that they are frustrated to .7 from .3 would likely prioritize redirecting rather than continuing to respond. However, in most cases when a human is asked whether they would like to be redirected they are extremely unlikely to answer ‘No’ so that could also play a part in the implantation of adding that question to the decision tree.

**Part B – Conditional Probabilities & Bayes Nets (60 points)**

It is possible to compute the above relevant probabilities in real-time based on new information, rather than fixing them ahead of time. The following is a Bayes Net representing the following random variables:

* Accurate. The accuracy of the speech recognition. Domain: {True,False}
* ProblemSize. The size of the problem the user is having. Domain: {Big, Small}
* ConversationLength. The length of the conversation so far. Domain: {Short, Medium, Long}
* Frustrated. Whether the user was frustrated after the last conversational turn. Domain: {True, False}
* Resolved. Whether this conversational turn will resolve the problem. Domain: {True, False}

Conditional Probability Tables:

|  |
| --- |
| P(Accurate=True) |
| 0.90 |

|  |
| --- |
| P(ProblemSize=Small) |
| 0.90 |

|  |  |  |
| --- | --- | --- |
| Problem\_Size | P(ConversationLength=Short) | P(ConversationLength)=Medium |
| Small | .40 | .40 |
| Big | .20 | .30 |

|  |  |  |
| --- | --- | --- |
| Conversation\_Length | Accurate | P(Resolved) |
| Short | True | .30 |
| Short | False | .20 |
| Medium | True | .50 |
| Medium | False | .30 |
| Long | True | .70 |
| Long | False | .40 |

|  |  |  |  |
| --- | --- | --- | --- |
| Problem\_Size | Conversation\_Length | Accurate | P(Frustrated) |
| Small | Short | True | .20 |
| Small | Short | False | .40 |
| Small | Medium | True | .30 |
| Small | Medium | False | .50 |
| Small | Long | True | .60 |
| Small | Long | False | .80 |
| Big | Short | True | .30 |
| Big | Short | False | .50 |
| Big | Medium | True | .60 |
| Big | Medium | False | .80 |
| Big | Long | True | .70 |
| Big | Long | False | .90 |

1. (10 pts) Write out two different conditional independence relationships within the above Bayes Nets. That is, enumerate two relationships in which two nodes are conditionally independent, given at least one other node.

Problem Size is independent of Resolved given Conversation Length

Resolved is independent of Frustrated given Accurate

1. (**10 pts) By hand, compute the probability of P(Resolved = True | Conversation\_Length = Long, Problem\_Size = Big, Accurate = True).**

P(Resolved= True | CL = Long, PS = Big, Accurate = True):

P(Resolved | CL = Long, Accurate=True) = .70

Because in the above question I was able to state that Problem Size is independent of Resolved given Conversation Length, then you can determine the value of P(Resolved=True | CL=Long, PS=Big, Accurate=True) by simply calculating the P(Resolved | CL=Long, Accurate=True) which from the charts you can see is .70.

1. (15 pts) Write a function or call one of the aima probably functions (e.g., enumerate\_ask) to solve the above problem. As per the requirements in the assignment rubric document, paste your exact function call and expected output below. Your function can be specific to this Bayes Net, but should be generalizable enough to respond to different queries on this Bayes Net.

Function call -- computeBayesNet():

Output -- {F: 0.30000000000000004, T: 0.7}

Calling the computeBayesNet() function, creates the generalized Bayes Net problem and then calls

enumeration\_ask(Resolved, {ConversationLength: ‘long’, ProblemSize: T, Accurate: T}, bayes\_net) on line 290

The above call to enumeration\_ask then returns the solution which is the output shown above. All this computation is done within the computeBayesNet() function.

1. (15 points) Write a program or call one of the aima functions to print out the full joint distribution from the above Bayes Net. Submit your exact function call and expected output below.

Line 293: joint\_distribution(bayes\_net):

This is the call I used to print out the full joint distribution.

Again, like above, the call is made within the function computeBayesNet() so making a call to that function will print out the output to the above solution and also print out the full joint distribution table as the output shown below.

{Accurate, ProblemSize(T=Small, F=Big), ConversationLength, Resolved, Frustrated}

{(F, F, 'medium', F, F): 0.00041999999999999964, (F, F, 'medium', F, T): 0.001679999999999999,

(F, F, 'medium', T, F): 0.00017999999999999985, (F, F, 'medium', T, T): 0.0007199999999999996,

(F, F, 'long', F, F): 0.00029999999999999976, (F, F, 'long', F, T): 0.0026999999999999984,

(F, F, 'long', T, F): 0.00019999999999999987, (F, F, 'long', T, T): 0.0017999999999999993,

(F, F, 'short', F, F): 0.0007999999999999997, (F, F, 'short', F, T): 0.0007999999999999997,

(F, F, 'short', T, F): 0.00019999999999999993, (F, F, 'short', T, T): 0.00019999999999999993,

(F, T, 'medium', F, F): 0.012599999999999998, (F, T, 'medium', F, T): 0.012599999999999998,

(F, T, 'medium', T, F): 0.005399999999999999, (F, T, 'medium', T, T): 0.005399999999999999,

(F, T, 'long', F, F): 0.002159999999999999, (F, T, 'long', F, T): 0.00864,

(F, T, 'long', T, F): 0.0014399999999999997, (F, T, 'long', T, T): 0.00576,

(F, T, 'short', F, F): 0.01728, (F, T, 'short', F, T): 0.01152,

(F, T, 'short', T, F): 0.00432, (F, T, 'short', T, T): 0.00288,

(T, F, 'medium', F, F): 0.0053999999999999986, (T, F, 'medium', F, T): 0.008099999999999998,

(T, F, 'medium', T, F): 0.0053999999999999986, (T, F, 'medium', T, T): 0.008099999999999998,

(T, F, 'long', F, F): 0.004050000000000001, (T, F, 'long', F, T): 0.00945,

(T, F, 'long', T, F): 0.00945, (T, F, 'long', T, T): 0.022049999999999993,

(T, F, 'short', F, F): 0.008819999999999998, (T, F, 'short', F, T): 0.0037799999999999995,

(T, F, 'short', T, F): 0.0037799999999999995, (T, F, 'short', T, T): 0.0016199999999999997,

(T, T, 'medium', F, F): 0.11340000000000001, (T, T, 'medium', F, T): 0.04860000000000001,

(T, T, 'medium', T, F): 0.11340000000000001, (T, T, 'medium', T, T): 0.04860000000000001,

(T, T, 'long', F, F): 0.01944000000000001, (T, T, 'long', F, T): 0.02916000000000001,

(T, T, 'long', T, F): 0.04536000000000001, (T, T, 'long', T, T): 0.06804,

(T, T, 'short', F, F): 0.18144000000000005, (T, T, 'short', F, T): 0.04536000000000001,

(T, T, 'short', T, F): 0.07776000000000002, (T, T, 'short', T, T): 0.019440000000000006}

1. **(10 points)** Assume for a specific user-agent interaction, P(Accurate=True) = .90 for every conversational turn and Problem\_Size = Small. If *k* <= 5 turns, Conversation\_Length=Short; if 6 <= *k* <= 10, Conversation\_Length = Medium, otherwise Conversation\_Length=Long. At what point should the chatbot redirect the conversation to a human in the decision tree above. Explain how you got your answer. If you used code to get your answer, submit your exact function call and output.

To get my answer, I first gathered from the chart the following probabilities:

P(Frustrated| Accurate, Problem\_Size=Small, CL = Short) = .20

P(Resolved | Accurate, Problem\_Size=Small, CL= Short) = .30

P(Frustrated | Accurate, Problem\_Size =Small, CL= Medium) = .30

P(Resolved | Accurate, Problem\_Size=Small, CL= Medium) = .50

P(Frustrated | Accurate, Problem\_Size=Small, CL= Long) = .60

P(Resolved | Accurate, Problem\_Size=Small, CL= Long) = .70

From there, I used those probabilities to calculate the value of K in every situation. If the conversation length is short and Accurate = True and Problem\_Size = Small, at turn 18 the chatbox would redirect the conversation to a human. If the conversation length is medium, Accurate = True and ProblemSize = Small, then after turn 16 the chatbox would redirect the user to a human. And finally if the Conversation Length = long and Accurate= True and problemSize = Small, then at turn 14, the chatbox would redirect the user to a human. These probabilities were dependent on the expected utility value based on the probability that the user is frustrated. Another value I looked at was the probability that the probability was resolved given Accurate = True, ProblemSize = Small and then CL = short, medium, and long.

Even from these probabilities, it shows that although the problem has a higher chance of being resolved given the ConversationLength = Long, the probability that the user is frustrated is also the highest.

For that reason, I would suggest that the chatbox redirecting after turn 16 is the most efficient because there is still a 50 percent chance the problem is resolved and only a 30 percent chance the use is frustrated.

**Part C –Markov Decision Processes (30 pts)** (20 pts) Explain how you would implement the problem of when to redirect a user to a human operator as a Markov Decision Process by outlining states, actions, transition probabilities, and rewards. The following are the constraints of your implementation:

States: {‘D1’, ‘D2’}

Terminal\_states: {U1, U2, U3, U4, U5, U6, U7}

Actions: {‘Respond-Resolved’, ‘Respond-notResolved’, ‘Redirect-Frustrated’, ‘Redirect-notFrustrated’}

Transition\_matrix: {

‘D1’: T(‘Respond-Resolved’, (PS=Small, Accurate, CL=Short), ‘U1’): .30

T(‘Respond-notResolved’, (PS=Small, Accurate, CL=Short), ‘D2’) = .70

T(‘Redirect-Frustrated’, (PS=Small, Accurate, CL=Short), ‘U2‘)= .20

T(‘Redirect-notFrustrated’, (PS=Small, Accurate, CL=Short), ‘U3’) = .80

T(‘Respond-Resolved’, (PS=Small, Accurate, CL=Medium), ‘U1’): .50

T(‘Respond-notResolved’, (PS=Small, Accurate, CL=Medium), ‘D2’) = .50

T(‘Redirect-Frustrated, (PS=Small, Accurate, CL=Medium), ‘U2’) = .30

T(‘Redirect-notFrustrated, (PS=Small, Accurate, CL=Medium), ‘U3’) = .70

T(‘Respond-Resolved’, (PS=Small, Accurate, CL=Long), ‘U1’): .70

T(‘Respond-notResolved’, (PS=Small, Accurate, CL=Long), ‘D2’): .30

T(‘Redirect-Frustrated’, (PS=Small, Accurate, CL=Long), ‘U2’) : .60

T(‘Redirect-notFrustrated’, (PS=Small, Accurate, CL=Long), ‘U3’): .40

‘D2’: T(‘Respond-Resolved’, (PS=Small, Accurate, CL=Short), ‘U5’): .30

T(‘Respond-notResolved’, (PS=Small, Accurate, CL=Short), ‘U4’) = .70

T(‘Redirect-Frustrated’, (PS=Small, Accurate, CL=Short), ‘U6‘)= .20

T(‘Redirect-notFrustrated’, (PS=Small, Accurate, CL=Short), ‘U7’) = .80

T(‘Respond-Resolved’, (PS=Small, Accurate, CL=Medium), ‘U5’): .50

T(‘Respond-notResolved’, (PS=Small, Accurate, CL=Medium), ‘U4’) = .50

T(‘Redirect-Frustrated, (PS=Small, Accurate, CL=Medium), ‘U6’) = .30

T(‘Redirect-notFrustrated, (PS=Small, Accurate, CL=Medium), ‘U7’) = .70

T(‘Respond-Resolved’, (PS=Small, Accurate, CL=Long), ‘U5’): .70

T(‘Respond-notResolved’, (PS=Small, Accurate, CL=Long), ‘U4’): .30

T(‘Redirect-Frustrated’, (PS=Small, Accurate, CL=Long), ‘U6’) : .60

T(‘Redirect-notFrustrated’, (PS=Small, Accurate, CL=Long), ‘U7’): .40

}

Rewards: {‘U1’: 5, ‘U2’: -1, ‘U3’: 5, ‘U4’: -3, ‘U5’: 5, ‘U6’:-1, ‘U7’: 5, ‘D1’: 0, ‘D2’:0}

My basic idea here was to incorporate all the probabilities of the determinable parameters like conversation length and using those numbers in the transition model depending on whether the conversation is short medium or long. Since you can only make decisions from two states, I made D1 and D2 the only states and then set the rewards at the terminal states which are the nodes U1-U7. I had made the reward values based on whether the problem was resolved and whether the customer was left frustrated.

1. (5 pts) In what ways do you believe your implementation will function in a different way as the specification in Part A. Explain.

For one, because of the fact that each Utility value (U1-U7) can be given a specific reward function, will make it so that the algorithm will try to stray from the terminal states that have a negative reward. For the purpose of this problem, I assigned negative reward states to U2, U4 and U6 due to the fact that they were states where the customer was left Frustrated, or the problem was unresolved through the responding/redirecting of the chat bot. For that reason, the algorithm will have a policy that is more conservative, meaning will always try to get to the terminal states with the higher reward outcomes. Because you would use value iteration to solve this problem, the utility values are also all starting at 0 and at each iteration are updated so I think the optimal solution for this problem may be different than that of the implementation in part A.

1. (5 pts) Is the general problem of determining when to redirect the user to a human or respond to the user more suitable to implement as a MDP (Part C) or as a decision tree (as in Part A). Explain your answer.

Standard decision trees have limitations in their ability to model complex situations especially when outcomes or events occur/reoccur over time. As a result of this, they are usually replaced with the use of Markov decision processes.

However, if there is a large number of embedded decision nodes in the branches of the decision tree, since each iteration of a standard Markov process can evaluate only one set of decision rules at a time, the standard Markov process can be computationally impractical if decisions occur repetitively over time.

Since there is only two different actions you can take from each decision node, the number of decision nodes that are created at each iteration is relatively small.

For that reason, this problem is more suitable implemented as an MDP.

**Part D – Bonus (up to 30 points)**

Implement your design in Part C. – implemented:

**Method call: mdpProblem(conversationLength)**

**-conversationLength either ‘short’, ‘medium’, ‘long’**

**Prints the value iteration model for this problem**

**{'U5': 5.0, 'U4': -3.0, 'U6': -1.0, 'U2': -1.0, 'D2': 3.6, 'U7': 5.0, 'U1': 5.0, 'U3': 5.0, 'D1': 3.6}**