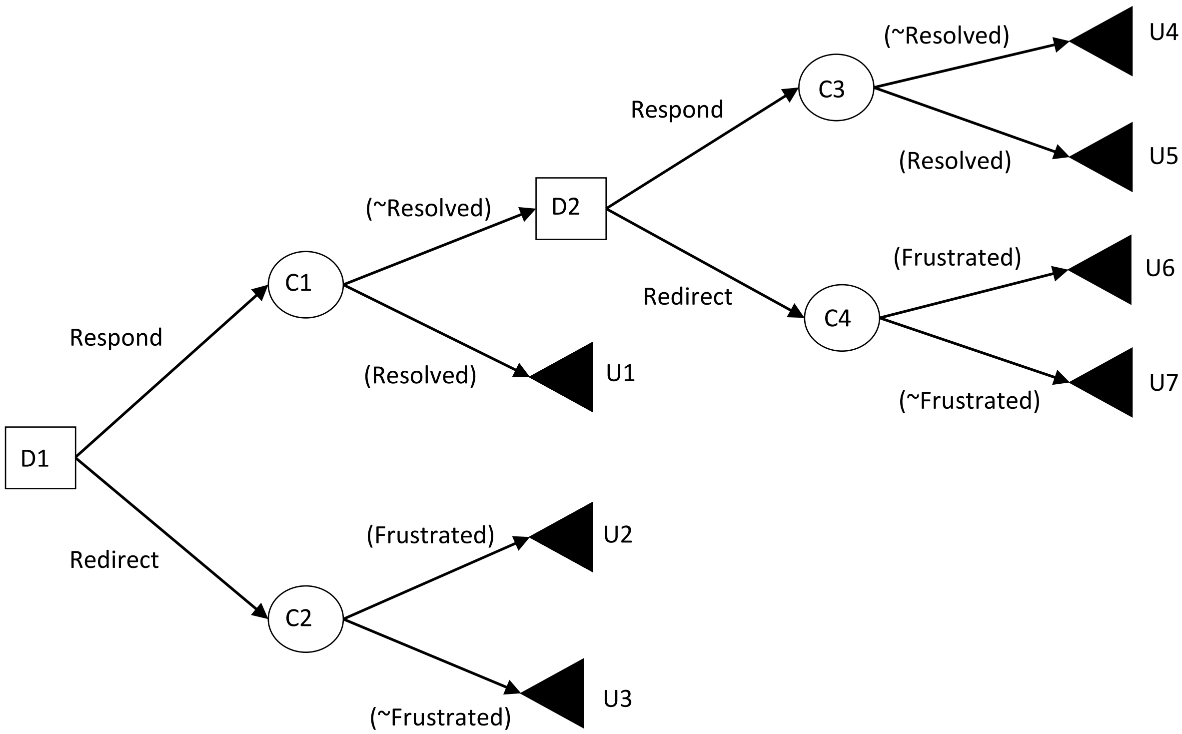
CS1571
Assignment 3

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:



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	10
U2	1.5
U3	-66.5
U4	0
U5	9
U6	2.7
U7	-56.7
C1	10
C2	-65
C3	9
C4	-54

Best course of action:

The best course of action is to respond C1.

2. (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.

Call function q2 with the maximum number of turn for the above tree q2(2). It should output a path ”Respond, Respond” to lead you to C1 and then C3. If you called q2(1) it’ll let the function know there is only 1 turn that needs to be found and will return the response “Respond”

3. (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).

Its best to redirect at turn 16. To solve this I used coding. I created a function q3(). It calls q2(30) and splits the returned string into an array and finds the length of the array. The length shows how many times the chatbox responded before it redirected.

4. (5 pts) Write a new utility function *U'* that prioritizes <u>shorter</u> 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.

> If we made the new utility value for when the chatbot get a response to equal 50 and the utility value for redirecting a user to a human if they are NOT frustrated equals -50+5*k*. It would cause the chatbox to redirect at turn 8 instead of 16, because the value to keep responding get lower a lot quicker while the value to redirect increases a lot sooner.

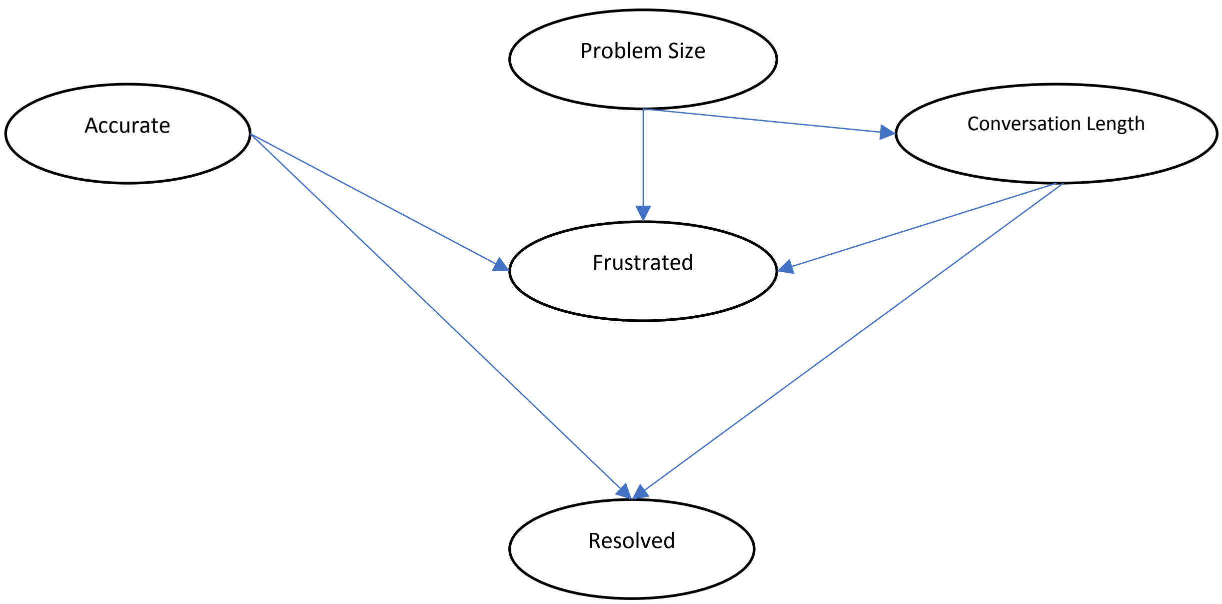
5. (5 pts) Assume the chatbot always has the option to ask a person if they'd like to be directed to a human prior to deciding to redirect them. You can assume if people say yes there is a higher probability they are frustrated, and if they say no, they are likely not frustrated – but sometimes people 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.

> I would add an extra Decision Node that is the user input of whether they want to redirect or not. Then from that node there will be two edges that leads to the next decision node of whether the robot redirects or not. Those nodes will lead to a similar tree above, but if the user said they wanted to redirect, but the chat box kept responding P(Frustrated)=0.7. This is an increase from the normal probability, because in the real world when people say they want to speak to a human, but get ignored they are a lot more likely to be frustrated. Likewise, if the user said they didn't want to redirect then P(Frustrated) = 0.2.

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

6. (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.

P(Frustrated|Accurate) = P(Frustrated|Resolved, Accurate)

P(Frustrated|Conversation Length) = P(Frustrated|Resolved, Conversation Length)

7. (10 pts) By hand, compute the probability of P(Resolved = True | Conversation_Length = Long, Problem_Size = Big, Accurate = True).

P(Resolved = True | Conversation_Length = Long, Problem_Size = Big, Accurate = True) =

P(Resolved = True | Conversation_Length = Long , Accurate = True)* P(Problem_Size = Big)

P(Resolved = True | Conversation_Length = Long , Accurate = True) = P(Resolved = True ^ Conversation_Length = Long ^ Accurate = True)/P(Conversation_Length = Long ^ Accurate = True)

P(Conversation_Length = Long ^ Accurate = True) = 0.5 * 0.9 = 0.45

P(Resolved = True ^ Conversation_Length = Long ^ Accurate = True) = .7

(0.7/0.45)*0. 1 = 0.16

8. (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.

The function is q8(A,B) I used aima probability functions for this.

To get P(Resolved = True | Conversation_Length = Long, Problem_Size = Big, Accurate = True)

In A enter 'Resolved' and B enter 'Conversation_Length:long, Problem_Size:F, Accurate:T'.

To look up the probability for any value for A enter either:

'Resolved', 'Problem_Size', 'Conversation_Length', 'Accurate', or 'Frustrated'

For B enter those same variables, but with their values as well for example 'Conversation_Length:short'. The values are T, F, short, medium, and long. For problem size T = small and F = big. The also must be a , and a space in-between each new variable.

This should return the probability of all the values of A

9. (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.

Call q9() and it will print a list of the full joint distribution of the Bayes Net

10. (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 in depth how you got your answer. If you used code to get your answer, submit your exact function call and output.

I called q8() with the different Conversation_Lengths and a Problem_Size of small.

For a short conversation P(Resolved) = .29 and P(Frustrated) = 0.22

For a medium conversation P(Resolved) = .48 and P(Frustrated) = 0.32

For a long conversation P(Resolved) = .7 and P(Frustrated) = 0.62

I think took that number and entered it into a modified version of q2(turns) called q10(turns)

The modified version will change the probability of Resolved and Frustrated dependent on the number of turns done. I started with the input of 10, so q10(10), and if the chatbox doesn't redirect the function will call itself but with q10(turns+10). This will keep going till it reaches a redirect value.

It returned 30, so at turn 30 it should redirect.

Part C –Markov Decision Processes (30 pts)

11. (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:
    a. You should consider the Part B probabilities related to Accurate, Frustrated, Resolved, and ConversationLength in your response. You can assume that the constraints in Question 10 of Part B apply, including the fact that Problem_Size is fixed at Small.
    b. You can design your own reward function, based on the states that you identify. The reward function should roughly preserve the relative values of different terminal outcomes as described in Part A (to the extent that resolving the problem is better than redirecting if the user is frustrated, which is better than redirecting if the user is not frustrated).

    While we are grading your solution, you may want to add a brief explanation of your solution so that we understand what you are trying to accomplish.

There are 6 states: Start, Short, Medium, Long, Resolved, and Redirect. With the initial state bring Start.

Action(Short, Medium, Long) = redirect, stop

Action(Start) = length

Resolved and Redirect to not get actions because once the chatbox get to those states it's done, so they are terminal states.

Transition Probabilities:

T(Start,length,Short)=0.4

T(Start, length, Medium) = 0.4

T(Start, length, Long) = 0.2

T(Short, stop, Resolved) = 0.33

T(Short, stop, Short) = 0.67

T(Medium, stop, Resolved) = 0.55

T(Medium, stop, Medium) = 0.45

T(Long, stop, Resolved) = 0.77

T(Long, stop, Long) = 0.23

T(Short, redirect, Redirect) = 0.22

T(Short, redirect, Short) = 0.78

T(Medium, redirect, Redirect) = 0.33

T(Medium, redirect, Medium) = 0.67

T(Long, redirect, Redirect) = 0.66

T(Long, redirect, Long) = 0.54

R(Short) = -0.1, R(Medium) = -0.2 R(Long) = -0.5

R(Resolved)= +10 with a decay of 0.01

R(Redirect)= +5

12. (5 pts) In what ways do you believe your implementation will function in a different way as the specification in Part A. Explain.

Part A doesn't consider the size of the Problem Size, the Conversation Length, or if the data is Accurate. Since my implementations include theses variables it will hopefully provide a more accurate answer on the probability of resolving the problems.

13. (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.

I would say a decision tree. MDP tend to ignore the past, but we need to know how long the chatbox has been trying to resolve the problem. This is because as more time goes by the more likely a user will get frustrated. The MDP won't be able to take that information into account.

Part D – Bonus (up to 30 points)

Implement your design in Part C.