Decision Networks

Decision networks differ from Bayesian networks in two primary ways—their ability to contain actions and utilities. They contain **Chance Nodes (ovals), Decision nodes (rectangles), Utility nodes (diamonds).**

Chance Nodes – standard random variables in Bayes nets.

Decision Nodes – represent points where the decision-maker has a choice of actions. For example, if we were choosing whether we would take a bet or not based on the utility and probability associated with each outcome, this would be represented as a decision node.

Utility Nodes – which represent the agent's utility function.

An intuitive method for building Decision Diagrams is representing each variable as a function. For example, if we say the utility of driving on the freeway is dependent on the traffic (T), if there is an accident (A), and the day of the week (DW). Additionally, let's say that whether we drive or not is the action D. We can simply write out our utility as $U(T, A, DW, D)$. Thus, we should draw an arrow from each of the variables which predict influence the function to our diamond representing the utility. This heuristic, can be used also to draw Bayesian networks.

Question 1

A bored student is deciding between going to the movies (A=M), working out at the gym (A=G), programming their CS188 project (A=P), or spending time on Facebook (A=F).

If the student watches the movie peacefully (p=0.8), he will lower his cortisol levels by 10 points (higher levels are correlated with more stress, negatively correlated with happiness). However, with (p=0.2), he will see his ex-girlfriend and his cortisol levels will rise 25 points. He has the option of calling his friend and awkwardly asking her if his ex-girlfriend is at the movies, but this increases his cortisol level by 5 points.

If the student goes to the gym without pulling a muscle (p=0.95), he will lower his cortisol by 20 points. If he pulls a muscle, he will increase his cortisol by 100 points.

If the student works on his programming assignment and completes it (p=0.6), he will lower his cortisol by 40 points. However, he may run into a bug (p=0.4) that he will not be able to fix that raises his cortisol 50 points. He has the option of going to office hours, which guarantee that he will complete his assignment, but will raise his cortisol by 10 points.

The student may also surf facebook (p=1.00) and his cortisol levels will stay the same.

a) Draw the decision network that represents this problem.

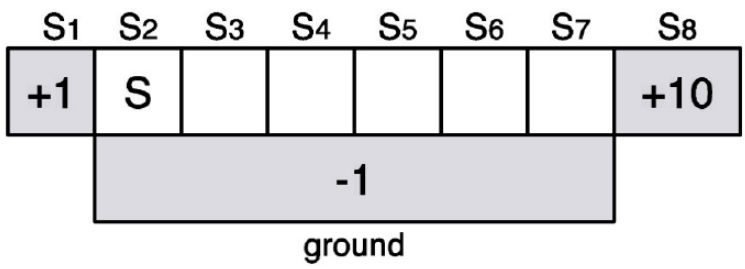
b) Calculate the expected cortisol for each of the four options without calling a friend or seeing a TA. Then calculate the expected cortisol after seeing a TA and calling your friend.

c) Suppose that your TA is too busy to help you with probability 0.4, and available to help you with probability 0.6. If he helps you, you are able to fix your bug. If he can't help you, you are unable to fix your bug. Should you seek your TA for help?

d) Calculate the optimal decision given all four possible activities, include the TA's availability as a random variable to this question.

Question 2

Value iteration: $U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U_i(s')$



Consider the above MDP, representing a robot on a balance beam. Each grid square is a state and the available actions are right and left. The agent starts in state s2 , and all states have reward 0 aside from the ends of the grid s1 and s8 and the ground state, which have the rewards shown. Moving left or right results in a move left or right (respectively) with probability p. With probability 1 − p, the robot falls off the beam (transitions to ground, and receives a reward of -1). Falling off, or reaching either endpoint, result in the end of the episode (i.e., they are terminal states). Note that terminal states receive no future reward.

a. For what values of p is the optimal action from s_2 to move right if the discount γ is 1?

b. For what values of γ is the optimal action from s_2 to move right if p = 1?

c. Given initial value estimates of zero, show the results of one, then two rounds of value iteration.