National University of Singapore
School of Computing

Semester 1, AY2023-24

CS4246/CS5446 AI Planning and Decision Making

Tutorial Week 7: DA and MDP

Guidelines

You may discuss the content of the questions with your classmates. But everyone should work on and be ready to present ALL the solutions.

Problem 1: VPI

Mr. Huges has bought a new suit that he wants to wear to office. However, it looks like it might rain. Mr. Huges has a long walk to the MRT. If it rains his suit will be ruined. If it doesn't rain, no harm done. His umbrella will protect his suit, but he hates the inconvenience of carying it around all day. If it rains and he has the umbrella the suit is saved. There is a 40% chance that it will rain. His decision problem is represented as the following decision tree.

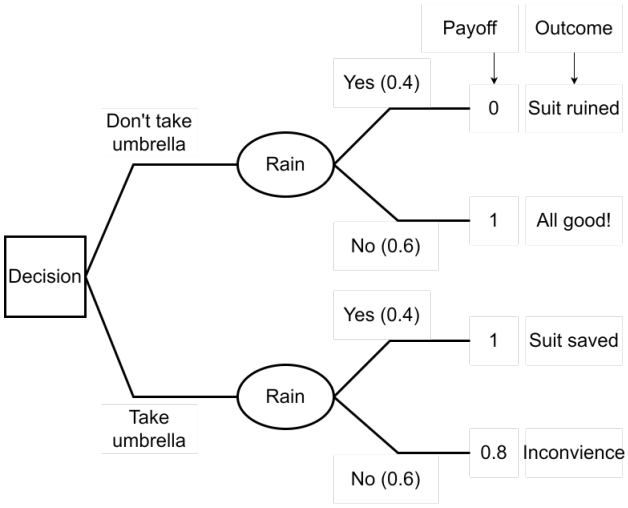
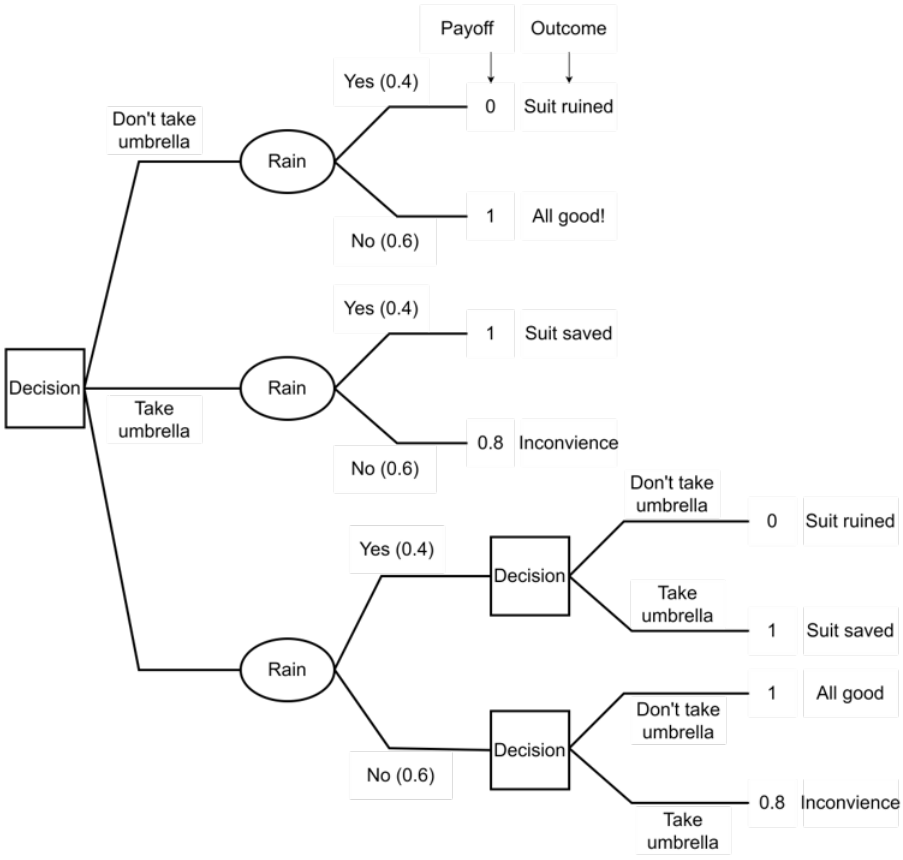


Figure 1: Mr. Huges' decision tree

He can call AccurateWeather, a perfect information service that tells if it rains or not perfectly.

1. Draw the decision tree to include this perfect information source in the decision making.
Solution:



2. What is the "value" (in terms of the payoff) that Mr. Huges should pay to AccurateWeather to gather information about the rain?

Solution:
Let's call the situation of not asking AccurateWeather as 'RiskyDecision'.
$EU(\neg Umbrella) = 0.4 + 1 \times 0.6 = 0.6$
$EU(Umbrella) = 0.4 \times 1 + 0.6 \times 0.8 = 0.88$
$EU(Ask) = 0.4 \times 1 + 0.6 \times 1 = 1$
$EVPI = EU(Ask) - EU(RiskyDecision) = 1 - 0.88 = 0.12.$
Hence, Mr. Huges should pay ≤ 0.12 "value" in terms of the payoff.

Problem 2: Formulating Markov Decision Processes

Specify the following problems as a Markov decision process, *i.e.* specify the state space, the actions, the transition functions, and the reward function. What is the (approximate) size of the state space and the action space?

- The traveling salesman problem. A salesman must visit every city in a graph and minimize travel time and is constrained not to visit any city twice.

  Solution:
  Assume that there are N cities. The state can be specified by a pair (V, c) where V is a subset of cities that have already been visited and c is the city that the salesman is currently at. V can take 2^N possible values and c can take N values, so the state space size is $O(N2^N)$. The actions are to move from the current city to another city that is connected to the current city. There is a special initial state where the salesman starts with none of the cities visited, and a special terminal state where all the cities have been visited and the salesman is back at the starting city. The action space size is N. Transition is deterministic: moving from c to c' changes (V, c) to $(V \cup c, c')$. At the terminal state, all actions self-loop with zero reward. Rewards for actions at other states is the negative of the edge weight of the edge (c, c') for moving from c to c' if c' has not been visited and $-\infty$ (or a very large negative number) otherwise. This is a finite horizon MDP with horizon N. It is also deterministic, so can be solved using deterministic planning methods (although can also be represented as a MDP).

- Inventory control. The company has space to store N items. At the end of each day, the company will make an order to increase the number of items up to $M \leq N$. Placing an order cost c for each time an order is made. If there is not enough items in the inventory to meet the orders for the day, a back order has to be made at the cost of b per unit back ordered (up to a known maximum of B units). There is a holding cost of 1 for each item in the inventory at the end of the day.

  Solution:
  The state space is the set of integers ranging from $-B$ to N indicating how many items is in stock where a negative number indicates the number that needs to be backordered. The state space size is $N + B + 1$. The actions are to order items to increase the number to M for $M = 0, \ldots, N$, with $N + 1$ actions. The transition is deterministic and transitions the system to state M. This is followed by subtracting the day's demand d, where d is drawn from the demand probability $\Pr(d)$. For state $s \geq 0$, the action to increase the number to M has reward $-c - s$ unless no order is required giving reward -s. For state $s < 0$, the action to increase the number to M has reward $sb - c$ (s is negative so sb is negative), unless no order is required giving reward sb.

Problem 3: Value Iteration

Consider the following 2 state, 2 action MDP with discount factor 0.9.

$P(s_1\|s_1,a_1)$	$P(s_2\|s_1,a_1)$	$P(s_1\|s_2,a_1)$	$P(s_2\|s_2,a_1)$
0.9	0.1	0	1

$P(s_1\|s_1,a_2)$	$P(s_2\|s_1,a_2)$	$P(s_1\|s_2,a_2)$	$P(s_2\|s_2,a_2)$
0.1	0.9	0	1

$R(s_1,a_1)$	$R(s_1,a_2)$	$R(s_2,a_1)$	$R(s_2,a_2)$
1	0	3	3

1. Assume a finite horizon problem with horizon 1 (only 1 action is to be taken). What is the utility or value function and the optimal action in each state?

   Solution:
   $U_1(s_1) = 1$, $U_1(s_2) = 3$, $a^*(s_1) = a_1$, $a^*(s_2) = a_1$ or a_2.

2. Assume a finite horizon problem with horizon 2 (2 actions to be taken). What is the utility or value function and the optimal action in each state?

   Solution:
   Use

   $$U_2(s_i) = \max_a (R(s_i,a) + \gamma \sum_{j=1}^{2} P(s_j|s_i,a)U_1(s_j)).$$

   For state 1 action 1

   $$value(utility) = 1 + 0.9(0.9 * 1 + 0.1 * 3) = 2.08.$$

   For state 1 action 2

   $$value = 0 + 0.9(0.9 * 3 + 0.1 * 1) = 2.52.$$

   Taking the max, we get $V_2(s_1) = 2.52$ with action a_2. For state 2, the system will self loop regardless of action with

   $$value = 3 + 0.9 * 3 = 5.7.$$

   Notice that this policy is different from the policy for horizon 1. Finite horizon problems have non-stationary (time dependent) policies.

3. What is the optimal infinite horizon policy?

   Solution:
   At state s_2, the system self-loops with reward 3 regardless of the action taken, so the infinite horizon utility or value at state 2 is $\frac{3}{1-\gamma} = \frac{3}{1-0.9} = 30$.
   Once the action in state s_2 is fixed, there are two possible policies corresponding to action

a_1 and a_2 in state s_1.
If action a_1 is taken, the value of the policy must satisfy

$$U(s_1) = 1 + 0.9(0.9U(s_1) + 0.1 * 30)$$

giving $U(s_1) = 19.47$.
If action a_2 is taken, the value of the policy must satisfy

$$U(s_1) = 0 + 0.9(0.9 * 30 + 0.1U(s_1))$$

giving $U(s_1) = 26.7$.
Hence action a_2 should be taken in state s_1.
