Melvin Mokhtari
Email: melvin.mokhtari@ec.iut.ac.ir
Course: Artificial Intelligence
Instructor: Dr. Samaneh Hosseini

Assignment 4
ID: 9831143
Semester: Winter 2023
Date: June 27, 2023

Problem 1

Solution:

Part 1

Utility represents a measure of preference or satisfaction derived from a particular outcome or choice. In a decision-making problem, different alternatives are associated with different utilities, which indicate the desirability of each outcome for the decision-maker.

The utility function is used to quantify the preferences of the decision-maker and allows for a systematic comparison of different alternatives. By assigning a numerical value to each possible outcome, the decision-maker can identify the most preferred alternative and make a rational choice.

Part 2

Expected Utility is a central concept in decision-making under uncertainty. It combines the probabilities of different outcomes with their associated utilities to produce a single value that represents the overall desirability of a decision. The Expected Utility is calculated as follows:

$$\text{Expected Utility} = \sum(ProbabilityOfOutcome * UtilityOfOutcome) \tag{1}$$

Here, the sum is taken over all possible outcomes. By comparing the Expected Utilities of different decisions, the decision-maker can choose the option that maximizes their overall satisfaction, given the uncertainties involved.

Part 3

A policy in a decision network is a function that maps the current state of the environment to a specific action or decision. In other words, a policy provides a set of rules or guidelines that prescribe the actions that an agent should take in each possible state. An optimal policy is a policy that maximizes the Expected Utility for the agent over time, given the constraints and uncertainties in the environment.

In a decision network, the goal is often to find the optimal policy, which will guide the agent's actions to achieve the best possible outcome in the long run. Finding the optimal policy typically involves solving a dynamic programming problem, such as the Bellman equation for Markov Decision Processes (MDPs).

Part 4

The Discount factor, denoted as γ, is a crucial parameter in MDPs. It is a number between 0 and 1 that determines the relative importance of immediate versus future rewards. The Discount factor is used to weigh the rewards that an agent receives at different time steps, with future rewards being discounted by a factor of γ^t, where t is the number of time steps into the future.

The use of a Discount factor has several justifications:

- It models the fact that future rewards are often less certain or less valuable than immediate rewards, due to factors such as risk, inflation, or time preference.
- It helps ensure that the value function, which is used to evaluate states and actions in MDPs, converges to a unique solution.

- It allows for the comparison of different actions and policies on a common scale, by aggregating the discounted rewards over time.

The behavior of an agent in an MDP can be significantly influenced by the choice of the Discount factor. An agent with a high Discount factor (e.g., 0.9) places more value on future rewards and tends to be more far-sighted in its decision-making. This means the agent is more likely to make choices that lead to higher long-term rewards, even if they involve short-term sacrifices.

On the other hand, an agent with a low Discount factor (e.g., 0.6) is more focused on immediate rewards and tends to be more myopic in its decision-making. This agent is likely to prioritize actions that yield immediate benefits, even if they result in lower long-term rewards. The choice of the Discount factor can thus have a significant impact on the agent's behavior and the optimality of its policy.

Part 5

Value iteration is a widely used algorithm for solving MDPs and finding the optimal policy. It involves iteratively updating the value function until convergence. There are two main variants of value iteration: synchronous and asynchronous.

- Synchronous value iteration updates the value function for all states simultaneously at each iteration. This means that the new values for all states are computed based on the previous values, and the value function is updated only after all states have been processed.

- Asynchronous value iteration updates the value function for one or more states at a time, using the most recent values for the other states. This means that the value function is updated more frequently and in a more flexible manner.

Asynchronous value iteration is considered more efficient and often converges faster than synchronous value iteration. This is because asynchronous updates can take advantage of the latest information and propagate value changes more quickly through the state space. However, the choice between synchronous and asynchronous value iteration depends on the specific problem andthe computational resources available, as both methods can be used effectively to solve MDPs.

Problem 2

Solution:

Part 1

After eliminating Run by maximizing, we have the following factors on look and see:

Look	*See*	*Value*
true	true	23
true	false	56
false	true	28
false	false	22

Part 2

The optimal decision function for run is:

Look	*See*	*Run*
true	true	yes
true	false	no
false	true	yes
false	false	no

Part 3

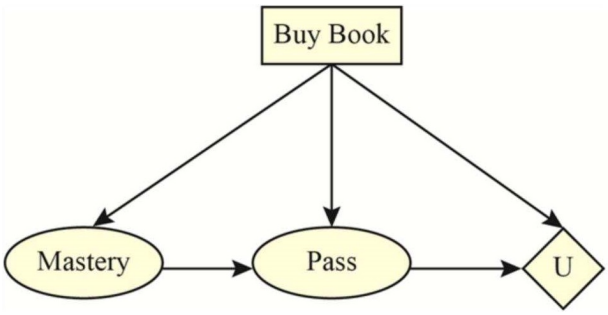
That is, if the agent sees, it should run.

Problem 3

Solution:

Part 1

The decision network for the given problem statement is given as follows:



It is a directed acyclic graph which has Buy Book as decision node, Mastery and Pass as chance node, and U as a utility node.

Part 2

The expected utility of buying the book is represented as EU[b] and not buying the book is represented by EU[$\neg b$]. They are computed as given below:

$$\begin{aligned} P(p \mid b) &= \sum_{m} P(p \mid b, m) P(m \mid b) \\ &= 0.9 \times 0.9 + 0.5 \times 0.1 \\ &= 0.86 \\ P(p \mid \neg b) &= \sum_{m} P(p \mid \neg b, m) P(m \mid \neg b) \\ &= 0.8 \times 0.7 + 0.3 \times 0.3 \\ &= 0.65 \end{aligned}$$

So, the expected utility of buying the book is computed as:

$$\begin{aligned} EU[b] &= \sum_{p} P(p \mid b) U(p, b) \\ &= 0.86(2000 - 100) + 0.14(-100) \\ &= 1620 \end{aligned}$$

The expected utility of not buying the book is computed as:

$$\begin{aligned} EU[\neg b] &= \sum_{p} P(p \mid \neg b) U(p, \neg b) \\ &= 0.65 \times 2000 + 0.14 \times 0 \\ &= 1300 \end{aligned}$$

Part 3

Since the expected utility of buying the book is less than not buying the book as the expected utility of buying the book is 1620 and the value 1620 is less than 1300 (expected utility of not buying the book), student should go for the option to purchase a book.

Problem 4

Solution:

Part 1

	1	2	3	4
1	Start 0	0	0	0
2	0	X	0	-1
3	0	0	0	+1

$V_{0(1)}$

Value Iteration

$V_1(S_{11}) \leq -0.05 + 0.9 * (\max_{a} (0.8 \times 0 + 0.1 \times 0 + 0.1 \times 0))$

$= -0.05$

* In a similar way, all other cells except S_3 will be valued -0.05; so I refrain to put them down again.

$V_1(S_{33}) = -0.05 + 0.9 * (0.8 \times 1 + 0.1 \times 0 + 0.1 \times 0) = 0.67$

The output table would be something like what is depicted below:

	1	2	3	4
1	Start	-0.05	-0.05	-0.05
2	-0.05	X	-0.05	-1
3	-0.05	-0.05	0.67	+1

$V_1(S)$

$V_2(S_{11}) = -0.05 + 0.9(0.8 \times -0.05 + 0.1 \times -0.05 + 0.1 \times -0.05) = -0.095$

* This value will be the same for all other cells except S_{23}, S_{32}, S_{33}.

$V_2(S_{23}) = -0.05 + 0.9(0.8 \times 0.67 + 0.1 \times -1 + 0.1 \times -0.05 = 0.3379$

$V_2(S_{32}) = -0.05 + 0.9(0.8 \times 0.67 + 0.1 \times 0.05 + 0.1 \times -0.05 = 0.4234$

$V_2(S_{33}) = 0.05 + 0.9(0.8 \times 1 + 0.1 \times (-0.05) + 0.1 \times (-0.05) = -0.84$

So, the final table of the second Iteration will be something like the table depicted below!

$V_2(S)$

	1	2	3	4
1	-0.095	-0.095	-0.095	-0.095
2	-0.095	X	0.3379	-1
3	-0.095	0.4234	-0.84	+1

Part 2

Policy Iteration

Iteration 0

	1	2	3	4
1	←	←	←	←
2	←	X	←	-1
3	←	←	←	+1

Iteration 1

	1	2	3	4
1	↓	↓	↓	↓
2	↓	X	↓	-1
3	↓	↓	↓	+1

Iteration 2

	1	2	3	4
1	↓	→	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

Iteration 3

	1	2	3	4
1	↓	←	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

Iteration 4

	1	2	3	4
1	↓	←	↓	←
2	↓	X	↓	-1
3	→	→	→	+1

* And this process continues until 9th Iteration. But the final resulted table is the same as the table above. So the same resulted values as the previously assumed Part will be applied to this section.

Part 3

	1	2	3	4
1	-0.05	-0.05	-0.05	+0.05
2	-0.05	X	-0.05	-1
3	-0.05	-0.05	-0.05	+1

-0.05 -0.05 -0.05 -0.05 +0.05 +1

S_{11} — S_{21} — S_{31} — S_{32} — S_{33} — S_{34} goal state

S_{33} : $(-0.05+1)*1$ = 0.95

S_{32} : $((-0.05)\times 2+1)*1$ = 0.9

S_{31} : $((-0.05)\times 3+1)*1$ = 0.85

S_{21} : $((-0.05)\times 4+1)*1$ = 0.8

S_{11} : $((-0.05)\times 5+1)*1$ = 0.75