TDT4171 Artificial Intelligence Methods
Assigment 4

Odd André Owren

March 2020

1 Utility

a) For the four different people in this problem, we have the following assessments:

- **Gabriel** is a risk-seeking person, as seen by the graph with the concave side facing upwards, thus describing a low utility compared to the amount of apples.
- **Gustav** is a risk-neutral individual, as this is a linear graph and the utility is constant.
- **Maria** is a risk-averse person, as seen by the graph with the concave side facing downwards, thus describing a high utility compared to the amount of apples.
- **Sonja** is a risk-neutral individual, but not in the same way as Gustav. Sonja has a utility that is linear with regards to the amount of apples, unlike Gustav who has a constant utility no matter what the amount of apples.

b) The graph of x^3 is shown in figure 1. From this graph, it is clear to see that when $x < 0$ then $x^3 < 0$ and the same holds for the opposite, i.e. $x > 0$ gives $x^3 > 0$. By looking at the graph, one can see that when $x < 0$ then we have risk-averse situation, and the opposite for $x > 0$; a risk-seeking situation.

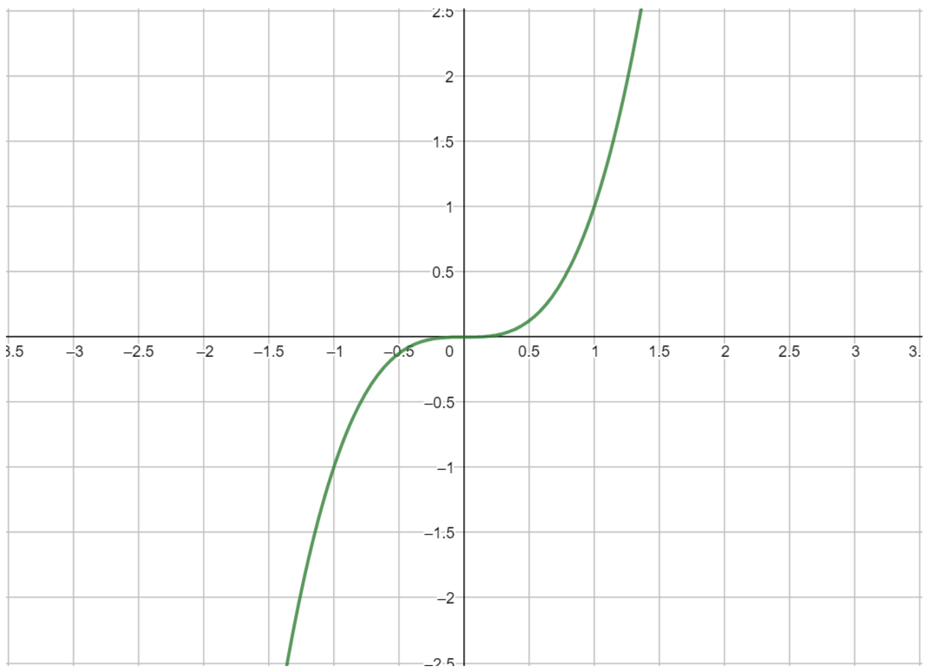

Figure 1: Graph of $f(x) = x^3$

By defining the lotteries from x^3 where x is the monetary value and x^3 defines the utility of the money, we get the following lotteries:
Risk-seeking: [(1,1),(0.3,2)] (Choice between getting 1 for sure or betting on getting 2 with a probability of 0.3, 0 else)
Risk-averse: [(1,-1), (0.3, -2)] (Choice between losing 1 for sure or betting on losing 2 with a chance of 0.3, 0 else)
The table then looks like this:

	Risk-seeking	Risk-averse
Lottery	[(1,1),(0.3,2)]	[(1,-1),(0.3,-2)]
Expected utility of lottery	[1, 2.4]	[-1, -2.4]
Utility of expected monetary value	[1, 0.216]	[-1, -0.216]

It is then pretty clear to see from the table above that the risk-seeking lottery will choose (0.3,2) with an expected utility of 2.4, although the utility of expected monetary value is lower. The opposite holds for the risk-averse lottery which will choose (1,-1) since this does not involve the possible risk of losing more money, which also gives a lower utility, even though the utility of expected monetary value is lower for this lottery.

2 Decision Network

a) The decision network in the problem modeled as a Bayesian network can be seen in figure 2.

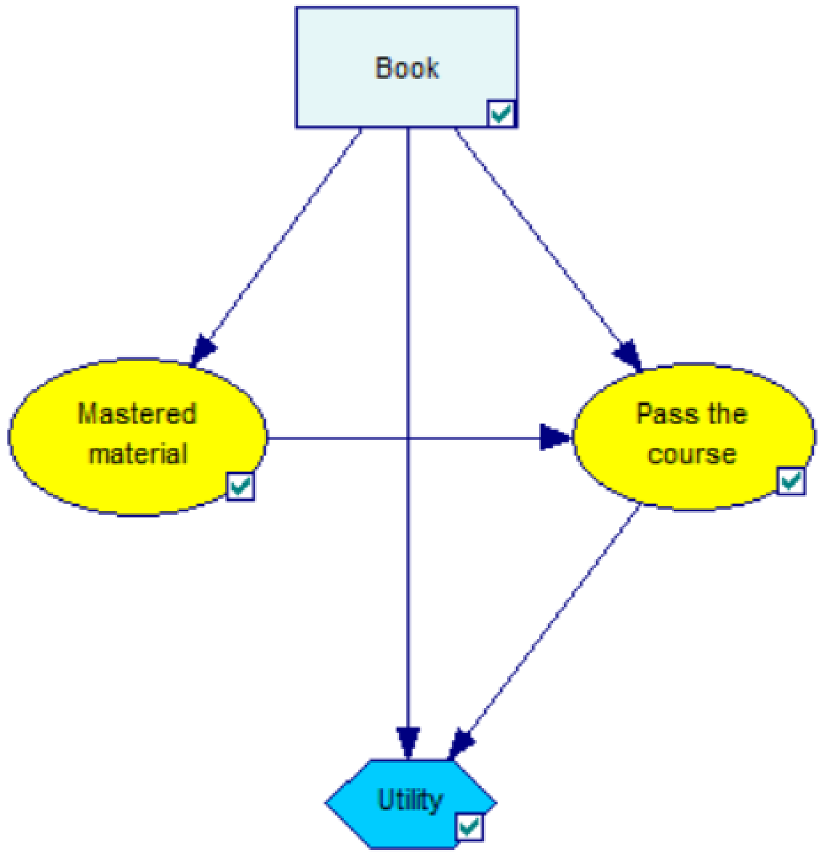

Figure 2: Decision network for whether or not to buy a book

b) The calculations of the utilities of buying the book and not buying the book, EU(b) and EU(¬b) respectively, are shown below.

$EU(b) = P(p|b) \cdot U(p, b) + P(\neg p|b) \cdot U(\neg p, b)$
where
$P(p|b) = P(p|b, m) \cdot P(m|b) + P(p|b, \neg m) \cdot P(\neg m|b) = 0.9 \cdot 0.9 + 0.1 \cdot 0.5 = 0.86$
and

$P(\neg p|b) = P(\neg p|b, m) \cdot P(m|b) + P(\neg p|b, \neg m) \cdot P(\neg m|b) = 0.1 \cdot 0.9 + 0.5 \cdot 0.1 = 0.14$
which gives
$EU(b) = 0.86 \cdot 2000 + 0.14 \cdot -100 = \underline{1620}$

$EU(\neg b) = P(p|\neg b) \cdot U(p, \neg b) + P(\neg p|\neg b) \cdot U(\neg p, \neg b)$
where
$P(p|\neg b) = P(p|\neg b, m) \cdot P(m|\neg b) + P(p|\neg b, \neg m) \cdot P(\neg m|\neg b) = 0.8 \cdot 0.7 + o.3 \cdot 0.3 = 0.65$
and
$P(\neg p|\neg b) = P(\neg p|\neg b, m) \cdot P(m|\neg b) + P(\neg p|\neg b, \neg m) \cdot P(\neg m|\neg b) = 0.2 \cdot 0.7 + 0.7 \cdot 0.3 = 0.35$
which gives
$EU(\neg b) = 0.65 \cdot 2000 + 0.35 \cdot 0 = \underline{1300}$

c) From the calculations done in the previous problem, it is pretty clear that the highest utility comes from buying the book. Of course there is a risk included in spending money on the book without a guaranteed payback, but Sam will with a pretty high probability get 2000 back for the cost of 100, and thus the expected payoff is higher when Sam buys the book. This leads to the conclusion that Sam should buy the book.

3 Markov Decision Process

a)

Initial	U[1]=0	U[2]=0	U[3]=0
Iteration 1	U[1]=0	U[2]=1	U[3]=0
Iteration 2	U[1]=0.375	U[2]=1	U[3]=0.375

On the first iteration, all the values are still 0, and only in state 2 is the reward anything else than 0. This gives the following calculations for U_2:

$U_2 = R_2 + \gamma max(P(Left|choiceLeft) \cdot U_1' + P(Right|choiceLeft) \cdot U_3', P(Left|choiceRight) \cdot U_1' + P(Right|choiceRight) \cdot U_3') = 1 + 0.5 \cdot (0.75 \cdot 0 + 0.25 \cdot 0) = 1$

where U' is the previous state.
On iteration 2 the value of state 2 is updated to 1, and thus we can start propagating this to state 1 and 3. This gives the following values:

$U_1 = R_1 + \gamma max(P(Left|choiceLeft) \cdot U_3' + P(Right|choiceLeft) \cdot U_2', P(Left|choiceRight) \cdot U_3' + P(Right|choiceRight) \cdot U_2') = 0 + 0.5 \cdot (0.75 \cdot 1 + 0.25 \cdot 0) = 0.375$

$U_2 = R_2 + \gamma max(P(Left|choiceLeft) \cdot U_1' + P(Right|choiceLeft) \cdot U_3', P(Left|choiceRight) \cdot U_1' + P(Right|choiceRight) \cdot U_3') = 1 + 0.5 \cdot (0.75 \cdot 0 + 0.25 \cdot 0) = 1$

$U_3 = R_3 + \gamma max(P(Left|choiceLeft) \cdot U_2' + P(Right|choiceLeft) \cdot U_1', P(Left|choiceRight) \cdot U_2' + P(Right|choiceRight) \cdot U_1') = 0 + 0.5 \cdot (0.75 \cdot 1 + 0.25 \cdot 0) = 0.375$

b) When picking a policy for state 1, we use an argmax-function on the actions in that state with converged utilities. This gives the following calculations:

$a_1 = argmax_{a \in actions}(\sum P(s'|1,a)U_{s'}) = argmax(\text{right: } 0.75 \cdot 1.25 + 0.25 \cdot 0.5, \text{left: } 0.25 \cdot 1.25 + 0.75 \cdot 0.5) = \text{right: } 1.0625$

I.e. the best action in state 1 with converged values is right with an expected utility of 1.0625.

4 Value-iteration

a) This is implemented in the code that is attached in the delivery.

b) See figure 3. It is pretty clear to see that after the iteration, the value of going into the hole is extremely negative. This can be seen in the context of the high discount-factor (future rewards more important) which means that it is more important to get to the goal at one point than to get there faster.

-0.970	-0.989	-0.996	-0.998
-0.964	-10.000	-2.371	-10.000
-0.917	-0.452	-0.655	-10.000
-10.000	-0.293	-0.175	0.000

Figure 3: Results after convergence of value iteration

c) See figure 4. As one may notice, some of the policies seem kind of weird (e.g. going up when in 1st column, 3rd row). This can be explained by looking at the transition table which gives us that the probability of going the opposite direction of what you choose is 0, and thus the safest option when standing next to a hole (or other highly negative states) is to go the complete opposite direction, and "hope" for an adjustment of the action (e.g. going right when choice is up).

down	up	up	up
left	left	down	left
up	down	left	left
left	right	right	left

Figure 4: Move policy after value iteration.