CS 188
Spring 2021

# Regular Discussion 9 Solutions

## Decision Networks

- **Chance nodes** - Chance nodes in a decision network behave identically to Bayes' nets. Each outcome in a chance node has an associated probability, which can be determined by running inference on the underlying Bayes' net it belongs to. We'll represent these with ovals.
- **Action nodes** - Action nodes are nodes that we have complete control over; they're nodes representing a choice between any of a number of actions which we have the power to choose from. We'll represent action nodes with rectangles.
- **Utility nodes** - Utility nodes are children of some combination of action and chance nodes. They output a utility based on the values taken on by their parents, and are represented as diamonds in our decision networks.

The **expected utility** of taking an action $A = a$ given evidence $E = e$ and $n$ chance nodes is computed with the following formula:

$$EU(A=a|E=e) = \sum_{x_1,...,x_n} P(X_1 = x_1, ..., X_n = x_n|E=e)U(A=a, X_1 = x_1, ..., X_n = x_n)$$

where each $x_i$ represents a value that the $i^{th}$ chance node can take on. The **maximum expected utility** is the expected utility of the action that has the highest expected utility:

$$MEU(E=e) = \max_a EU(A=a|E=e).$$

## Value of Perfect Information

**Value of perfect information** (VPI) quantifies the amount an agent's maximum expected utility is expected to increase if it were to observe some new evidence. Usually observing new evidence comes at a cost. If we observed some new evidence $E' = e'$ before acting, the maximum expected utility of our action at that point would become

$$MEU(E=e, E'=e') = \max_a \sum_x P(X=x|E=e, E'=e')U(X=x, A=a).$$

However, note that *we don't know what new evidence we'll get*. Because we don't know what what new evidence $e'$ we'll get, we must represent it as a random variable $E'$. We will compute the expected value of the maximum expected utility:

$$MEU(E=e, E') = \sum_{e'} P(E'=e'|E=e)MEU(E=e, E'=e').$$

Observing a new evidence variable yields a different MEU with probabilities corresponding to the probabilities of observing each value for the evidence variable, and so by computing $MEU(E=e, E')$ as above, we compute what we expect our new MEU will be if we choose to observe new evidence. The VPI is the expected maximum expected utility if we were to observe the new evidence, minus the maximum expected utility if we were not to observe the new evidence:

$$VPI(E'|E=e) = MEU(E=e, E') - MEU(E=e).$$

# Properties of VPI

The value of perfect information has several very important properties, namely:

- **Nonnegativity.** $\forall E', e \; VPI(E'|E = e) \geq 0$
  Observing new information always allows you to make a *more informed* decision, and so your maximum expected utility can only increase (or stay the same if the information is irrelevant for the decision you must make).

- **Nonadditivity.** $VPI(E_j, E_k|E = e) \neq VPI(E_j|E = e) + VPI(E_k|E = e)$ in general.
  This is probably the trickiest of the three properties to understand intuitively. It's true because generally observing some new evidence $E_j$ might change how much we care about $E_k$; therefore we can't simply add the VPI of observing $E_j$ to the VPI of observing $E_k$ to get the VPI of observing both of them. Rather, the VPI of observing two new evidence variables is equivalent to observing one, incorporating it into our current evidence, then observing the other. This is encapsulated by the order-independence property of VPI, described more below.

- **Order-independence.** $VPI(E_j, E_k|E = e) = VPI(E_j|E = e) + VPI(E_k|E = e, E_j) = VPI(E_k|E = e) + VPI(E_j|E = e, E_k)$
  Observing multiple new evidences yields the same gain in maximum expected utility regardless of the order of observation. This should be a fairly straightforward assumption - because we don't actually take any action until after observing any new evidence variables, it doesn't actually matter whether we observe the new evidence variables together or in some arbitrary sequential order.

# Markov Decision Processes

A Markov Decision Process is defined by several properties:

- A set of states $S$
- A set of actions $A$.
- A start state.
- Possibly one or more terminal states.
- Possibly a **discount factor** $\gamma$.
- A **transition function** $T(s, a, s')$.
- A **reward function** $R(s, a, s')$.

# The Bellman Equation

- $V^*(s)$ – the optimal value of $s$ is the expected value of the utility an optimally-behaving agent that starts in $s$ will receive, over the rest of the agent's lifetime.
- $Q^*(s, a)$ - the optimal value of $(s, a)$ is the expected value of the utility an agent receives after starting in $s$, taking $a$, and acting optimally henceforth.

Using these two new quantities and the other MDP quantities discussed earlier, the Bellman equation is defined as follows:

$$V^*(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V^*(s')]$$

We can also define he equation for the optimal value of a q-state (more commonly known as an optimal **q-value**):

$$Q^*(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$

which allows us to reexpress the Bellman equation as

$$V^*(s) = \max_a Q^*(s,a).$$

# Value Iteration

The **time-limited value** for a state $s$ with a time-limit of $k$ timesteps is denoted $V_k(s)$, and represents the maximum expected utility attainable from $s$ given that the Markov decision process under consideration terminates in $k$ timesteps. Equivalently, this is what a depth-$k$ expectimax run on the search tree for a MDP returns.

**Value iteration** is a **dynamic programming algorithm** that uses an iteratively longer time limit to compute time-limited values until convergence (that is, until the $V$ values are the same for each state as they were in the past iteration: $\forall s, V_{k+1}(s) = V_k(s)$). It operates as follows:

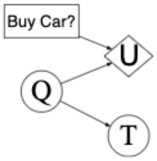
1. $\forall s \in S$, initialize $V_0(s) = 0$. This should be intuitive, since setting a time limit of 0 timesteps means no actions can be taken before termination, and so no rewards can be acquired.
2. Repeat the following update rule until convergence:

   $$\forall s \in S,\ \ V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V_k(s')]$$

   At iteration $k$ of value iteration, we use the time-limited values for with limit $k$ for each state to generate the time-limited values with limit $(k+1)$. In essence, we use computed solutions to subproblems (all the $V_k(s)$) to iteratively build up solutions to larger subproblems (all the $V_{k+1}(s)$); this is what makes value iteration a dynamic programming algorithm.

# 1 Decision Networks and VPI

A used car buyer can decide to carry out various tests with various costs (e.g., kick the tires, take the car to a qualified mechanic) and then, depending on the outcome of the tests, decide which car to buy. We will assume that the buyer is deciding whether to buy car $c$ and that there is time to carry out at most one test which costs \$50 and which can help to figure out the quality of the car. A car can be in good shape (of good quality $Q = +q$) or in bad shape (of bad quality Q=¬q), and the test might help to indicate what shape the car is in. There are only two outcomes for the test T: pass (T=pass) or fail (T=fail). Car $c$ costs \$1,500, and its market value is \$2,000 if it is in good shape; if not, \$700 in repairs will be needed to make it in good shape. The buyer's estimate is that $c$ has 70% chance of being in good shape. The Decision Network is shown below.



**(a)** Calculate the expected net gain from buying car c, given no test.

$$\begin{aligned} EU(\text{buy}) &= P(Q=+q)\cdot U(+q,\text{buy}) + P(Q=\neg q)\cdot U(\neg q, buy) \\ &= .7\cdot 500 + 0.3\cdot -200 = 290 \end{aligned}$$

**(b)** Tests can be described by the probability that the car will pass or fail the test given that the car is in good or bad shape. We have the following information:

$$P(T=\text{pass}|Q=+q) = 0.9$$
$$P(T=\text{pass}|Q=\neg q) = 0.2$$

Calculate the probability that the car will pass (or fail) its test, and then the probability that it is in good (or bad) shape given each possible test outcome.

$$\begin{aligned} P(T=\text{pass}) &= \sum_q P(T=\text{pass}, Q=q) \\ &= P(T=\text{pass}|Q=+q)P(Q=+q) + P(T=\text{pass}|Q=\neg q)P(Q=\neg q) \\ &= 0.69 \\ P(T=fail) &= 0.31 \\ P(Q=+q|T=\text{pass}) &= \frac{P(T=\text{pass}|Q=+q)P(Q=+q)}{P(T=\text{pass})} \\ &= \frac{0.9\cdot 0.7}{0.69} = \frac{21}{23} \approx 0.91 \\ P(Q=+q|T=\text{fail}) &= \frac{P(T=\text{fail}|Q=+q)P(Q=+q)}{P(T=\text{fail})} \\ &= \frac{0.1\cdot 0.7}{0.31} = \frac{7}{31} \approx 0.22 \end{aligned}$$

**(c)** Calculate the optimal decisions given either a pass or a fail, and their expected utilities.

$$\begin{aligned}
EU(\text{buy}|T=pass) &= P(Q=+q|T=\text{pass})U(+q,\text{buy}) + P(Q=\neg q|T=\text{pass})U(\neg q,\text{buy})\\
&\approx 0.91\cdot 500 + 0.09\cdot(-200) \approx 437\\
EU(\text{buy}|T=\text{fail}) &= P(Q=+q|T=\text{fail})U(+q,\text{buy}) + P(Q=\neg q|T=\text{fail})U(\neg q,\text{buy})\\
&\approx 0.22\cdot 500 + 0.78\cdot(-200) = -46\\
EU(\neg\text{buy}|T=pass) &= 0\\
EU(\neg\text{buy}|T=fail) &= 0
\end{aligned}$$

Therefore: $MEU(T=\text{pass}) = 437$ (with buy) and $MEU(T=\text{fail}) = 0$ (using $\neg$buy)

**(d)** Calculate the value of (perfect) information of the test. Should the buyer pay for a test?

$$\begin{aligned}
VPI(T) &= (\sum_t P(T=t)MEU(T=t)) - MEU(\phi)\\
&= 0.69\cdot 437 + 0.31\cdot 0 - 290 \approx 11.53
\end{aligned}$$

You shouldn't pay for it, since the cost is \$50.

# 2 MDPs: Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. If your total score is 6 or higher, the game ends, and you receive a utility of 0. When you Stop, your utility is equal to your total score (up to 5), and the game ends. When you Draw, you receive no utility. There is no discount ($\gamma = 1$). Let's formulate this problem as an MDP with the following states: $0, 2, 3, 4, 5$ and a *Done* state, for when the game ends.

**(a)** What is the transition function and the reward function for this MDP? The transition function is

$$\begin{aligned} T(s, Stop, Done) &= 1 \\ T(0, Draw, s') &= 1/3 \text{ for } s' \in \{2, 3, 4\} \\ T(2, Draw, s') &= 1/3 \text{ for } s' \in \{4, 5, Done\} \\ T(3, Draw, s') &= \begin{matrix} 1/3 \text{ if } s' = 5 \\ 2/3 \text{ if } s' = Done \end{matrix} \\ T(4, Draw, Done) &= 1 \\ T(5, Draw, Done) &= 1 \\ T(s, a, s') &= 0 \text{ otherwise} \end{aligned}$$

The reward function is

$$R(s, Stop, Done) = s, s \leq 5$$

$$R(s, a, s') = 0 \text{ otherwise}$$

**(b)** Fill in the following table of value iteration values for the first 4 iterations.

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $V_0$ | 0 | 0 | 0 | 0 | 0 |
| $V_1$ | 0 | 2 | 3 | 4 | 5 |
| $V_2$ | 3 | 3 | 3 | 4 | 5 |
| $V_3$ | 10/3 | 3 | 3 | 4 | 5 |
| $V_4$ | 10/3 | 3 | 3 | 4 | 5 |

**(c)** You should have noticed that value iteration converged above. What is the optimal policy for the MDP?

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi^*$ | Draw | Draw | Stop | Stop | Stop |

**(d)** Perform one iteration of policy iteration for one step of this MDP, starting from the fixed policy below:

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi_i$ | Draw | Stop | Draw | Stop | Draw |
| $V^{\pi_i}$ | 2 | 2 | 0 | 4 | 0 |
| $\pi_{i+1}$ | Draw | Stop | Stop | Stop | Stop |