CS 188 Spring 2021 — Introduction to Artificial Intelligence

# Exam Prep 9 Solutions

## Q1. VPI

You are the latest contestant on Monty Hall's game show, which has undergone a few changes over the years.
In the game, there are $n$ closed doors: behind one door is a car ($U(car) = 1000$), while the other $n-1$ doors each have a goat behind them ($U(goat) = 10$). You are permitted to open exactly one door and claim the prize behind it.

You begin by choosing a door uniformly at random.

**(a)** What is your expected utility?

Answer: $(1000 * \frac{1}{n} + 10 * \frac{n-1}{n})$ or $(10 + 990 * \frac{1}{n})$

We can calculate the expected utility via the usual formula of expectation, or we can note that there is a guaranteed utility of 10, with a small probability of a bonus utility. The latter is a bit simpler, so the answers to the following parts use this form.

**(b)** After you choose a door but before you open it, Monty offers to open $k$ other doors, each of which are guaranteed to have a goat behind it.
If you accept this offer, should you keep your original choice of a door, or switch to a new door?

$EU(keep)$: $10 + 990 * \frac{1}{n}$

$EU(switch)$: $10 + 990 * \frac{(n-1)}{n*(n-k-1)}$

Action that achieves $MEU$: switch

The expected utility if we keep must be the same as the answer from the previous part: the probability that we have a winning door has not changed at all, since we have gotten no meaningful information.
In order to win a car by switching, we must have chosen a goat door previously (probability $\frac{n-1}{n}$) and then switch to the car door (probability $\frac{1}{n-k-1}$).
Since $n-1 > n-k-1$ for positive $k$, switching gets a larger expected utility.

**(c)** What is the value of the information that Monty is offering you?

Answer: $990 * \frac{1}{n} * \frac{k}{n-k-1}$

The formula for VPI is $MEU(e) - MEU(\emptyset)$. Thus, we want the difference between $EU(switch)$ (the optimal action if Monty opens the doors) and our expected utility from part (a).
(It is true that $EU(keep)$ happens to have the same numeric expression as in part (a), but this fact is not meaningful in answering this part.)

**(d)** Monty is changing his offer!
After you choose your initial door, you are given the offer to choose any other door and open this second door. If you do, after you see what is inside the other door, you may switch your initial choice (to the newly opened door) or keep your initial choice.
What is the value of this new offer?

Answer: $\boxed{\frac{990}{n}}$

Intuitively, if we take this offer, it is as if we just chose two doors in the beginning, and we win if either door has the car behind it. Unlike in the previous parts, if the new door has a goat behind it, it is not more optimal to switch doors.
Mathematically, letting $D_i$ be the event that door $i$ has the car, we can calculate this as $P(D_2 \cup D_1) = P(D_1) + P(D_2) = 1/n + 1/n = 2/n$, to see that $MEU(\text{offer}) = 10 + 990 * \frac{2}{n}$. Subtracting the expected utility without taking the offer, we are left with $990 * \frac{1}{n}$.

**(e)** Monty is generalizing his offer: you can pay $\$d^3$ to open $d$ doors as in the previous part. (Assume that $U(\$x) = x$.) You may now switch your choice to any of the open doors (or keep your initial choice). What is the largest value of $d$ for which it would be rational to accept the offer?

Answer: $\boxed{d = \sqrt{\frac{990}{n}}}$

It is a key insight (whether intuitive or determined mathematically) that the answer to the previous part is constant for each successive door we open. Thus, the value of opening $d$ doors is just $d * 990 * \frac{1}{n}$. Setting this equal to $d^3$, we can solve for $d$.

# Q2. How do you Value It(eration)?

**(a)** Fill out the following True/False questions.

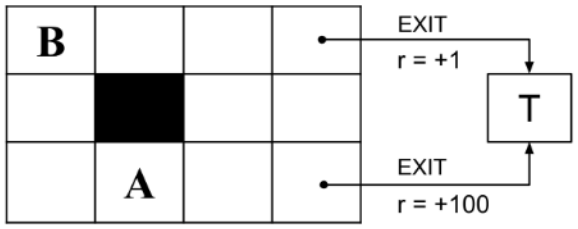
**(i)** [✓ True] ○ False: Let $A$ be the set of all actions and $S$ the set of states for some MDP. Assuming that $|A| \ll |S|$, one iteration of value iteration is generally faster than one iteration of policy iteration that solves a linear system during policy evaluation. One iteration of value iteration is $O(|S|^2|A|)$, whereas one iteration of policy iteration is $O(|S|^3)$, so value iteration is generally faster when $|A| \ll |S|$

**(ii)** ○ True [✓ False]: For any MDP, changing the discount factor does not affect the optimal policy for the MDP. Consider an infinite horizon setting where we have 2 states $A$, $B$, where we can alternate between $A$ and $B$ forever, gaining a reward of 1 each transition, or exit from $B$ with a reward of 100. In the case that $\gamma = 1$, the optimal policy is to forever oscillate between $A$ and $B$. If $\gamma = \frac{1}{2}$, then it is optimal to exit.

The following problem will take place in various instances of a grid world MDP. Shaded cells represent walls. In all states, the agent has available actions ↑, ↓, ←, →. Performing an action that would transition to an invalid state (outside the grid or into a wall) results in the agent remaining in its original state. In states with an arrow coming out, the agent has an *additional* action $EXIT$. In the event that the $EXIT$ action is taken, the agent receives the labeled reward and ends the game in the terminal state $T$. Unless otherwise stated, all other transitions receive no reward, and all transitions are deterministic.

For all parts of the problem, assume that value iteration begins with all states initialized to zero, i.e., $V_0(s) = 0\ \forall s$. **Let the discount factor be $\gamma = \frac{1}{2}$ for all following parts.**

**(b)** Suppose that we are performing value iteration on the grid world MDP below.



**(i)** Fill in the optimal values for A and B in the given boxes.

$V^*(A)$ : $\boxed{25}$ $V^*(B)$ : $\boxed{\frac{25}{8}}$

**(ii)** After how many iterations $k$ will we have $V_k(s) = V^*(s)$ for all states $s$? If it never occurs, write "never". Write your answer in the given box.

$\boxed{6}$

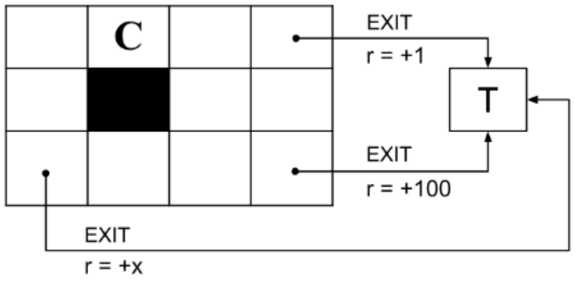
**(iii)** Suppose that we wanted to re-design the reward function. For which of the following new reward functions would the optimal policy **remain unchanged**? Let $R(s, a, s')$ be the original reward function.

- ■ $R_1(s, a, s') = 10R(s, a, s')$
- ■ $R_2(s, a, s') = 1 + R(s, a, s')$
- ■ $R_3(s, a, s') = R(s, a, s')^2$
- ☐ $R_4(s, a, s') = -1$
- ☐ None

$R_1$: Scaling the reward function does not affect the optimal policy, as it scales all Q-values by 10, which retains ordering
$R_2$: Since reward is discounted, the agent would get more reward exiting then infinitely cycling between states

$R_3$: The only positive reward remains to be from exiting state $+100$ and $+1$, so the optimal policy doesn't change
$R_4$: With negative reward at every step, the agent would want to exit as soon as possible, which means the agent would not always exit at the bottom-right square.

**(c)** For the following problem, we add a new state in which we can take the $EXIT$ action with a reward of $+x$.



**(i)** For what values of $x$ is it *guaranteed* that our optimal policy $\pi^*$ has $\pi^*(C) = \leftarrow$? Write $\infty$ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

$\boxed{50} < x < \boxed{\infty}$

We go left if $Q(C, \leftarrow) > Q(C, \rightarrow)$. $Q(C, \leftarrow) = \frac{1}{8}x$, and $Q(C, \rightarrow) = \frac{100}{16}$. Solving for $x$, we get $x > 50$.

**(ii)** For what values of $x$ does value iteration take the **minimum** number of iterations $k$ to converge to $V^*$ for all states? Write $\infty$ and $-\infty$ if there is no upper or lower bound, respectively. Write the upper and lower bounds in each respective box.

$\boxed{50} \leq x \leq \boxed{200}$

The two states that will take the longest for value iteration to become non-zero from either $+x$ or $+100$, are states $C$, and $D$, where $D$ is defined as the state to the right of $C$. $C$ will become nonzero at iteration 4 from $+x$, and $D$ will become nonzero at iteration 4 from $+100$. We must bound $x$ so that the optimal policy at $D$ does not choose to go to $+x$, or else value iteration will take 5 iterations. Similar reasoning for $D$ and $+x$. Then our inequalities are $\frac{1}{8}x \geq \frac{100}{16}$ and $\frac{1}{16}x \leq \frac{100}{8}$. Simplifying, we get the following bound on $x$: $50 \leq x \leq 200$

**(iii)** Fill the box with value $k$, the **minimum** number of iterations until $V_k$ has converged to $V^*$ for all states.

$\boxed{4}$

See the explanation for the part above