Mini-Course 1, Module 4
Dynamic Programming

CMPUT 365
Fall 2021

Reminders: Oct 1, 2021

- Things you must submit over the coming days:
  - Next week Friday (8th) at noon: Graded quiz
  - Next week monday (i.e. the 4): Blackjack notebook (instead of practice quiz)
    - Its a 5 min exercise: open run the things, hit submit
- **There is NO Jupiter notebook assignment next week. A graded quiz instead!**

Reminders: Oct 1, 2021

- We imported your grades into eclass! Check them out email cmput365@ about any problems….like “I got zero for everything! What happened?”

- If you have questions about submitting assignments, platform bugs, or technical problems email cmput365@ualberta.ca

Value functions are functions

- $v_\pi(s)$ is a function
- Input: states
- Output: real numbers
- $v_\pi : \mathcal{S} \to \mathbb{R}$

Q functions are functions

- $q_\pi(s, a) = p(s', r \mid s, a)[r + \gamma v_\pi(s')]$ is a function
- Input: states and actions
- Output: real numbers
- $q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

Querying these functions

- We can query q_π with an input state s, and action a
- We can query v_π with an input state s
- We cannot query v_π with an input state s, and action a
  - The v_π function does not take an action as an input!!

Definitions & Bellman Equations

- $v_\pi(s) \doteq \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \ldots \mid S_t = s]$
  - This is the definition of the value function
- $$v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s', r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$
  - Its a property of the value function defined in line one
  - We start with the definition of v_\pi and use math to show that its true
  - That’s why the Bellman equation does not use ‘dot’ =

Dynamic Programming and functions

- Dynamic Programming methods show us how we can turn these recursive bellman Equations into compute programs

  - Turn these functions into python programs that compute value functions and policies

Outline for today

- Overview of mini essay due Oct 20th
- Worksheet questions

Mini essay instructions

- **Pick one** of the topics and write a **two paragraph** discussion of the topic. You are allowed **300 words maximum**. Submit via E-class.

- You will be marked on:
  - **Presentation:** Spelling, grammar, punctuation, formating :: **⅓ of the mark**

  - **Structure** (use of topic sentences, paragraphs about only one idea, use short sentences) :: **⅓ of the mark**

  - **Content** (did you think about the topic and illustrate clarity of thought, sensible comments, showed that you looked up the topic and read about it) :: **⅓ of the mark**

Mini essay instructions

- https://docs.google.com/document/d/1EOC5TQh-SKC5zYXrvvC44hR8wthwttrROKwyDPlUBz0

Writing is hard, assume the reader is barely following at all times

- **The reader cannot ask you questions as they read: this is your one shot to convey your ideas and messages**
- Never underestimate how people can misunderstand another's writing
- Never underestimate how two people can think one paragraph can mean totally different things

General advice

- **Writing is about structure**
- Write a topic sentence
- Make sure each paragraph has one idea
- Say important things first
- Be direct and say things as plainly as possible

Be sincere

- Be sincere about what you are trying to do in the paper
    - You have to care about what you are doing, and your writing will reveal when you don’t!
- Think about: what do I really want to communicate here?
- If its not clear in your mind what you want to say, then what you write down will not be clear
- Writing is also for you: it makes you question your work which makes the work better

Minimalist and Just-in-time

- Don’t talk about things that are not relevant to your topic, to your contributions, to you insights, and to your reader

- Tell the reader what they need to know, only when they need to know it

Be consistent, be boring

- At least at first
- Don’t use a different word or phrase for the same thing to spice things up
- Hunt for consistency issues in your document: e.g., interchanging “method”, “algorithm”, “agent”
- Don’t use flowery, over the top language: called **purple prose**
- Don’t use words like “very”, “extremely”, “interestingly” to make your prose more impactful. Improve the content instead

Edit, Edit, Edit

- You have to be willing to throw it all in the garbage
  - I often delete sentences, paragraphs and sections…multiple times
- Be your own reviewer
  - Question everything; anticipate questions the reader might have
  - Did this paragraph convey what I wanted? What was this paragraph or section even about?
  - Is this idea concisely explained? Remove extra words and phrases
  - Could I completely re-organize this to get it across better?

Small things

- Watch out for backward sentences: say the most important thing first
- Don’t define acronyms that you only use once
- Don’t use lists too much
- Don’t use meaningless or irrelevant words (“modern” RL algorithms, “popular” optimizer)

Small things

- Read your sentences and ask yourself: “is this true?”, often times its not—sloppy prose
- Wrong subject for verb: “Reinforcement learning tries to solve”, RL is a formalism, it cannot be trying something. This is literally not true!
- Ask yourself: could the opposite of this sentence also be true?
- Avoid long sentences. The reader forgets halfway through
    - Short punchy declarative sentences are easy to read

Small things

- Watch out for false parallelism in lists
  - “There are many possible approaches to exploration including (1) optimistic initial values, (2) upper confidence bound actions selection,…” all list items should be the same type
- Don’t use **bold** or **colours** to emphasize things
- Be consistent with British vs American spellings
- Avoid strong words like “must”, “requires”
  - Avoid strong statements…they are often false

It takes time …

- Find good writers and study how they craft intros and their general writing style
  - Learn from demonstration
- Practice, Practice, Practice
- Remember writing is hard for all of us, and many good writers don’t enjoy it!

Links to resources

- Strunk and White is the classic reference book
- Other stuff:
  - http://approximatelycorrect.com/2018/01/29/heuristics-technical-scientific-writing-machine-learning-perspective/
  - https://icml.cc/Conferences/2002/craft.html

Worksheet time

Q1

1. In iterative policy evaluation, we seek to find the value function for a policy π by applying the Bellman equation many times to generate a sequence of value functions v_k that will eventually converge to the true value function v_π. How can we modify the update below to generate a sequence of action value functions q_k?

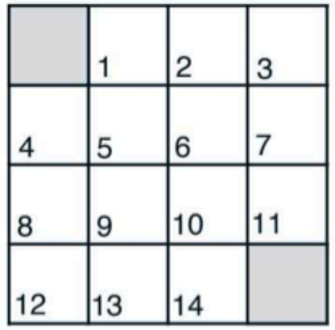
$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right]$$

Q2

2. A deterministic policy $\pi(s)$ outputs an action $a \in \mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ directly. More generally, a policy $\pi(\cdot|s)$ outputs the probabilities for all actions: $\pi(\cdot|s) = [\pi(a_1|s), \pi(a_2|s), \ldots, \pi(a_k|s)$. How can you write a deterministic policy in this form? Let $\pi(s) = a_i$ and define $\pi(\cdot|s)$.

Q3

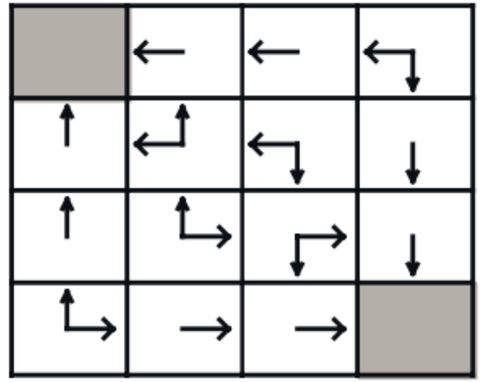
3. (*Exercise 4.1 S&B)* Consider the 4x4 gridworld below, where actions that would take the agent off the grid leave the state unchanged. The task is episodic with $\gamma = 1$ and the terminal states are the shaded blocks. Using the precomputed values for the equiprobable policy below, what is $q_\pi(11, \text{down})$? What is $q_\pi(7, \text{down})$?



$R_t = -1$
on all transitions

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



Challenge Q

5. **(Challenge Question)** A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips. If the coin comes up heads, she wins as many dollars as she has staked on that flip; if it is tails, she loses her stake. The game ends when the gambler wins by reaching her goal of \$100, or loses by running out of money. On each flip, the gambler must decide what portion of her capital to stake, in integer numbers of dollars. This problem can be formulated as an undiscounted, episodic, finite MDP. The state is the gambler's capital, $s \in \{1, 2, ..., 99\}$ and the actions are stakes, $a \in \{0, 1, ..., \min(s, 100 - s)\}$. The reward is $+1$ when reaching the goal of \$100 and zero on all other transitions. The probability of seeing heads is $p_h = 0.4$.

5. **(Challenge Question)** A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips. If the coin comes up heads, she wins as many dollars as she has staked on that flip; if it is tails, she loses her stake. The game ends when the gambler wins by reaching her goal of \$100, or loses by running out of money. On each flip, the gambler must decide what portion of her capital to stake, in integer numbers of dollars. This problem can be formulated as an undiscounted, episodic, finite MDP. The state is the gambler's capital, $s \in \{1, 2, ..., 99\}$ and the actions are stakes, $a \in \{0, 1, ..., \min(s, 100-s)\}$. The reward is $+1$ when reaching the goal of \$100 and zero on all other transitions. The probability of seeing heads is $p_h = 0.4$.

(a) What does the value of a state mean in this problem? For example, in a gridworld where the value of 1 per step, the value represents the expected number of steps to goal. What does the value of state mean in the gambler's problem? Think about the minimum and maximum possible values.

Gambler's problem

- What are the states?
  - gambler's capital, s ϵ {1,2, …, 99}
  - terminal states 0 and 100; Let $V_0 = 0$
- What are the rewards?
  - +1 win, zero otherwise
- What are the actions:
  - a = [1:min(s,100-s)]

Gambler's problem

- For each action there are two possible outcomes, heads and tails
- For each action, compute $p(s', r \mid s, a)[r + V(s')]$ for each outcome, each $r + V(s')$

5. **(Challenge Question)** A gambler has the opportunity to make bets on the outcomes of a sequence of coin flips. If the coin comes up heads, she wins as many dollars as she has staked on that flip; if it is tails, she loses her stake. The game ends when the gambler wins by reaching her goal of \$100, or loses by running out of money. On each flip, the gambler must decide what portion of her capital to stake, in integer numbers of dollars. This problem can be formulated as an undiscounted, episodic, finite MDP. The state is the gambler's capital, $s \in \{1, 2, ..., 99\}$ and the actions are stakes, $a \in \{0, 1, ..., \min(s, 100 - s)\}$. The reward is $+1$ when reaching the goal of \$100 and zero on all other transitions. The probability of seeing heads is $p_h = 0.4$.

   (b) Modify the pseudocode for value iteration to more efficiently solve this specific problem, by exploiting your knowledge of the dynamics. *Hint: Not all states transition to every other state. For example, can you transition from state 1 to state 99?*

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\textit{terminal}) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \text{argmax}_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$