CS 188 Discussion 7:

MDP Review

Kenny Wang (kwkw@berkeley.edu)
Wed Oct 11, 2023

Slides based on Sashrika Pandey’s

Administrivia

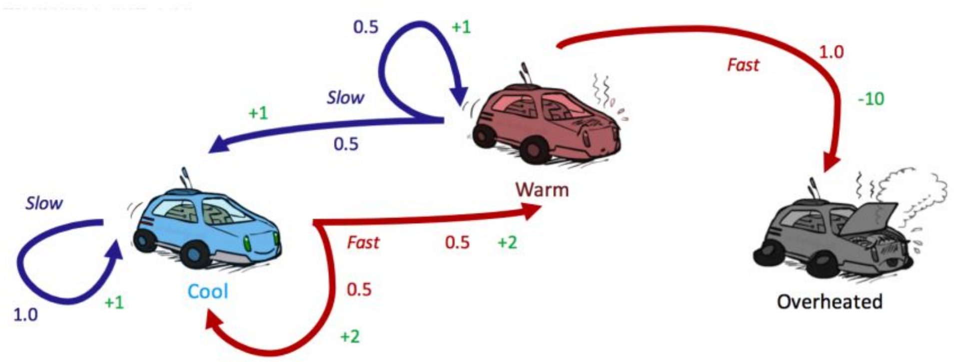
- **Midterm** on next Monday, October 16, 7-9 PM PT
- Exam requests form closes **today** Wednesday, October 11, 11:59 PM PT
  - DSP, alternate exam, remote exam, left handed seat, etc?
- Homework 5 is due yesterday–extensions capped to 3 days!
  - This is so we can release solutions before the exam! If you really need a longer extension, we have a makeup assignment. See Ed for details.
- We have office hours pretty much all day every weekday (12-7), come to Soda 341B!
- Discussion slides are on Ed

Today’s Topics

- MDPs (Markov Decision Processes)
  - Bellman Equation(s)
  - Value Iteration
  - Policy Iteration

What is an MDP?

- Set of states **S**
- Set of actions **A**
- Start state
- Terminal state(s)
- Discount factor **γ** [gamma]
    - Rewards decay as time passes, so we prefer sooner rewards
- Transition function **T(s, a, s')**
    - Probability of ending up in state s' by starting in s and taking action a
- Reward function **R(s, a, s')**



Example:

- S = {cool, warm, overheated}
- A = {slow, fast}
- T(warm, slow, cool) = 0.5
- R(warm, slow, cool) = 1

Incorporating Discount γ [gamma]

- **Additive utility**

$$U([s_0, a_0, s_1, a_1, s_2, \ldots]) = R(s_0, a_0, s_1) + R(s_1, a_1, s_2) + R(s_2, a_2, s_3) + \ldots$$

- **Discounted utility** incorporates **discount factor γ**
    - Reward of $\gamma^t R(s_t, a_t, s_{t+1})$ instead of $R(s_t, a_t, s_{t+1})$

$$U([s_0, a_0, s_1, a_1, s_2, \ldots]) = R(s_0, a_0, s_1) + \gamma R(s_1, a_1, s_2) + \gamma^2 R(s_2, a_2, s_3) + \ldots$$

Bellman Equations

- **Q*(s,a): the optimal value of (s, a)** [state, action pair]
  - The expected value of the utility an agent receives after starting in s, taking a, and acting optimally

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

- **V*(s) aka U*(s): the optimal value of state s**
  - The expected value of the utility that an agent starting in s and acting optimally will receive

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

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

In general, the * means optimal

Value Iteration

Value Iteration

- A dynamic programming algorithm we use to compute values until convergence ($\forall s, V_{k+1}(s) = V_k(s) = V^*(s)$ [optimal])
- Algorithm
  - $\forall s \in S$, initialize $U_0(s) = 0$ [initial value estimate is 0]
  - Repeat rule until convergence (U-values stop changing)

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

- Convergence is when
  - $\forall s \in S, U_k(s) = U_{k+1}(s) = U^*(s)$

Very similar to Bellman equation

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

Policy Iteration

Policy Iteration

- Issues with value iteration
  - $O(|S|^2|A|)$ runtime
  - Overcomputes, policy tends to converge faster than values
- Policy iteration: preserve the optimality from value iteration but with better performance by iterating until only the *policy* converges instead of the U-values

Policy Iteration

- Algorithm
  - Define initial policy $\boldsymbol{\pi_0}$ (can be arbitrary)
  - Repeat until convergence:
    - **Policy evaluation:** compute expected utility of starting in state s when following policy π, for all states s

$$U^{\pi}(s) = \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma U^{\pi}(s')]$$

    - **Policy improvement:** generate a better policy

$$\pi_{i+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma U^{\pi_i}(s')]$$

  - Convergence when $\boldsymbol{\pi_{i+1} = \pi_i}$ (policy stops changing)

Summary

- **Value Iteration**
  - ∀s ∈ S, initialize $U_0(s) = 0$ [initial value estimate is 0]
  - Repeat rule until convergence (U-values stop changing)

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

  - Convergence is when ∀s ∈ S, $U_k(s) = U_{k+1}(s) = U^*(s)$
- **Policy Iteration**
  - Define initial policy $\boldsymbol{\pi_0}$ (can be arbitrary)
  - Repeat until convergence:
    - **Policy evaluation:** compute expected utility of starting in state s when following policy π, for all states s

$$U^\pi(s) = \sum_{s'} T(s,\pi(s),s')[R(s,\pi(s),s') + \gamma U^\pi(s')]$$

    - **Policy improvement:** generate a better policy

$$\pi_{i+1}(s) = \operatorname{argmax}_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U^{\pi_i}(s')]$$

  - Convergence when $\boldsymbol{\pi_{i+1} = \pi_i}$ (policy stops changing)
- **Q*(s,a): the optimal value of (s, a)** [state, action pair]

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

- **U*(s): the optimal value of state s** $U^*(s) = \max_a Q^*(s,a)$

Good luck on your exam!

Thank you for attending!

Attendance link:

- https://tinyurl.com/cs188fa23

Discussion No: 7

Remember my name is Kenny

My email: kwkw@berkeley.edu