Worksheet Bandits

CMPUT 397
January 18, 2021

1. Suppose a game where you choose to flip one of two (possibly unfair) coins. You win $1 if your chosen coin shows heads and lose $1 if it shows tails.

(a) Model this as a K-armed bandit problem: define the action set.

解: Action set : two actions: a_1: flip coin 1
a_2: flip coin 2.

Worksheet Bandits

(b) Is the reward a deterministic or stochastic function of your action? => The coin is un-fair.

解: Stochastic because there is probabilities for winning or losing, and you don't know the probabilities.

例子: P_i = Probability of observing Heads with coin i.
q_i = flip coin i.

coin 1: H win \$1 P_1; T lose \$1 $(1-P_1)$

coin 2: H win \$1 P_2; T lose \$1 $(1-P_2)$

The true value $q_*(a_i) = \mathbb{E}[R \mid A = A_i]$
$= (+1)\cdot P_i + (-1)\cdot(1-P_i)$
$= 2P_i - 1$

注: 如果是 fair coin, 则 $q_*(a_i) = (1)\cdot 0.5 + (-1)\cdot 0.5$
$= 0$
$= \phi$.

Mark Holmstrom to Everyone 1:39 PM
MH: it's written as being a function of your action, not a function of your belief

I think if p = 1 it would be deterministic

Deepak Ranganatha Sastry Mamillapalli 1:39 PM to Everyone
DR: yes

but if you don't know if p=1, then its stochastic

Harsh Shah to Everyone 1:41 PM

I think if the action is already chosen and we know one of the trial has coin 2 chosen then it should be deterministic regardless of the probability I think

Daniel Asimiakwini to Everyone 1:44 PM
D: A question: If we do not know pi, is it inherently stochastic? => Yes!! (the probability)

Mohamed Ahmed to Everyone 1:45 PM
MA: I missed what was said about deterministic. when would it be deterministic again? => when we choose coin 2.

Nicholas Rebstock to Everyone 1:47 PM
N: We basically never know true probabilities. If you define stochasticity that way, everything is stochastic maybe => yes.

Cameron Jen to Everyone 1:51 PM
CJ: The only situation in which we could say it is deterministic would be 1/0 ? Yes!!

Cameron Jen to Everyone 1:52 PM
CJ: So in all other cases it would be stochastic, regardless of our knowledge of the dist or not => 不管我们是不是知道分布 (distributions), 它都是随机的.

Worksheet Bandits

(c) You do not know the coin flip probabilities. Instead, you are able to view 6 sample flips for each coin respectively: (T,H,H,T,T,T) and (H,T,H,H,H,T). Use the sample average formula (equation 2.1 in the book) to compute the estimates of the value of each action.

Sample Average formula: (formula 2.1) $Q_t(a) = \frac{\text{Sum of rewards when } a \text{ taken prior to } t}{\#\text{ of times } a \text{ taken prior to } t} \Rightarrow$ t^{th} estimate of the value of action a.

Actions: A_i = flipping the i^{th} coin

If observe Heads: win \$1 $P_i \rightarrow$ probability of flipping H by coin i.
observe Tails: lose \$1 $(1-P_i) \rightarrow$ probability of flipping T by coin i.

$Q_7(\text{flip coin 1}) = \frac{(-1)\cdot 4 + (1)\cdot 2}{6} \Rightarrow$ all of the reward of all of the action we flip coin 1.
(do the first action)
$\therefore -0.33$

$Q_7(\text{flip coin 2}) = \frac{(1)\cdot 4 + (-1)\cdot 2}{6}$
$\therefore 0.33$

(d) Decide on which coin to flip next! Assume it's an exploit step.

exploit step: 有开始也有终止，并且每次开始 step 时重置一切状态.

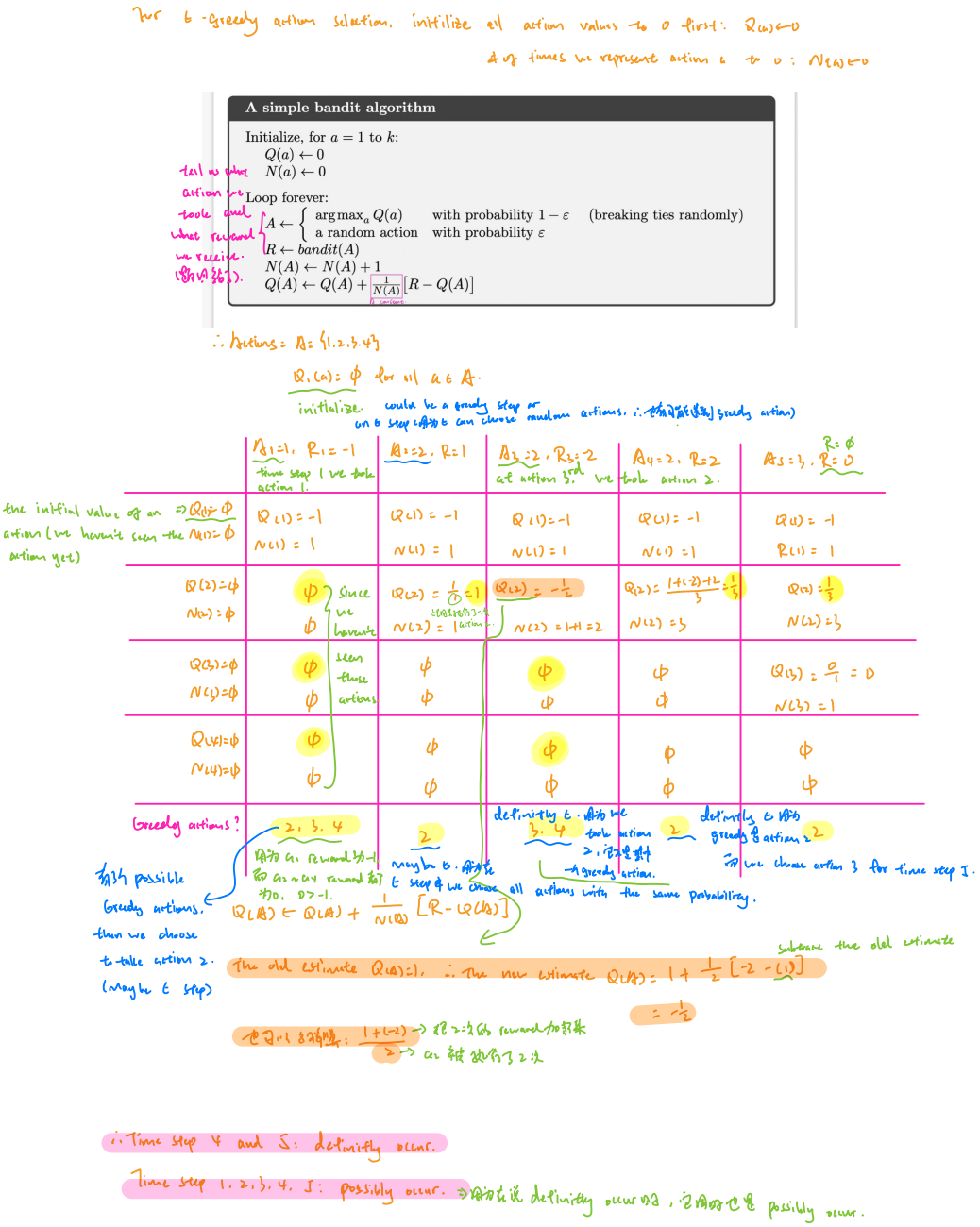
∵ Q (flip coin 1) = −0.33, Q (flip coin 2) = 0.33

∴ coin 2 has the higher estimate value.

∴ we choose to flip coin 2 next.

Worksheet Bandits

2. (Exercise 2.2 from S&B 2nd edition) Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ϵ-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 0$. On some of these time steps the ϵ case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?



Worksheet Bandits

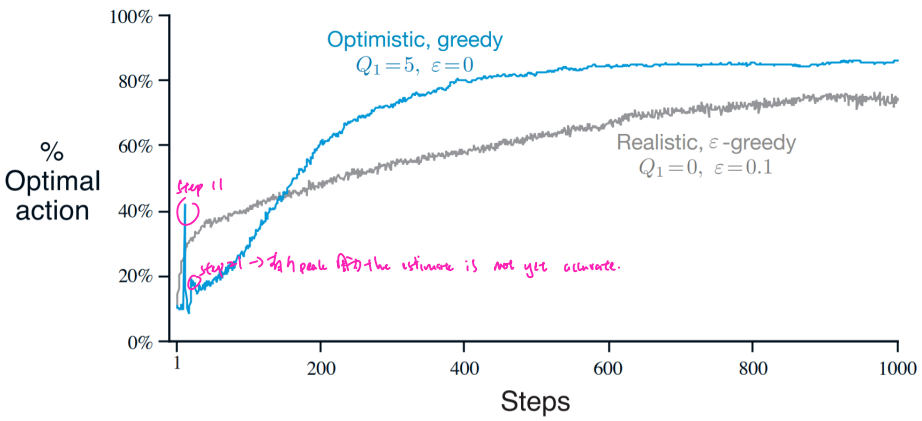


Figure 2.3: The effect of optimistic initial action-value estimates on the 10-armed testbed. Both methods used a constant step-size parameter, $\alpha = 0.1$.

4. **Challenge Problem:** (Exercise 2.6 from S&B 2nd edition) The results shown in Figure 2.3 should be quite reliable because they are averages over 2000 individual, randomly chosen 10-armed bandit tasks. Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps?

Steps 1~10: Sample all 10 actions at least once disappointing reward decrease Q(a).
optimal action in first 10 steps seen 10% of the time.

the peak.
steps 11: first time that we select among all 10 actions.
more probable that we select optimal action.

step 12: step 11 reward disappointing us, so select second best action...
⋮
step >1: once again select among all 10 actions. but our values are less optimistic (as the first time and are not yet accurate of the true value)
less probable to select the optimal action.
representing

Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps?

那么，为什么乐观的方法在曲线的早期部分会出现振荡和尖峰？换句话说，是什么原因使这种方法在特定的早期步骤上平均表现得特别好或特别差？

Worksheet MDPs

CMPUT 397
January 20, 2021

1. Suppose $\gamma = 0.9$ and the reward sequence is $R_1 = 2, R_2 = -2, R_3 = 0$ followed by an infinite sequence of 7s. What are G_1 and G_0?

Reward sequences: R_1 2, R_2 -2, R_3 0, R_4 7, R_5 7, R_6 7, ... 后面全是7.

$G_t = R_{t+1} + \gamma G_{t+1} \quad \forall t \geq 0$

$= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} \cdots$

$\therefore G_3 = R_4 + \gamma R_5 + \gamma^2 R_6 + \cdots$

$= 7 + \gamma 7 + \gamma^2 7 + \cdots$

$= \frac{7}{1-\gamma}$

$= \frac{7}{1-0.9}$

$= 70.$

$\therefore G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 R_4 + \cdots$ ($R_4 + \cdots = G_3$)

$= R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 (G_3)$

$= 2 + \gamma(-2) + 0 + \gamma^3 \times 70$

$= 51.2$

=> To compute G_0, we have to compute G_1 first.

G_1 以此类推.

G_2

$G_0 = R_1 + \gamma G_1$

$G_1 = R_2 + \gamma G_2$

$G_2 = R_3 + \gamma G_3$

$G_3 = R_4 + \gamma G_4$

$\vdots$

Worksheet MDPs

3. Prove that the discounted sum of rewards is always finite, if the rewards are bounded: $|R_{t+1}| \leq R_{\max}$ for all t for some finite $R_{\max} > 0$.

$$\left| \sum_{i=0}^{\infty} \gamma^i R_{t+1+i} \right| < \infty \qquad \text{for } \gamma \in [0, 1)$$

Hint: Recall that $|a + b| < |a| + |b|$.

$\because |R_{t+1}| \leq R_{max}$

$|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}| = \sum_{i=0}^{\infty} \gamma^i R_{t+1+i}$

$= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$

$\leq R_{max} + \gamma R_{max} + \gamma^2 R_{max} + \cdots$

$= R_{max}(1 + \gamma + \gamma^2 + \gamma^3 + \cdots)$ 记做系数

$= \frac{R_{max}}{1-\gamma}$

$< \infty$

$\therefore |\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}| < \infty$

Worksheet 3: Markov Decision Processes

CMPUT 397
January 21, 2022

6. Write $r(s,a)$ in terms of $r(s,a,s')$.

$$r(s,a) = \sum_{s'} \sum_{r} r\, p(s', r \mid s, a)$$

$$= \mathbb{E}\left[r \mid S_{t-1}=s,\ A_{t-1}=a \right]$$

$$r(s,a,s') = \sum_{r} r\, p(r \mid s, a, s')$$

$$= \mathbb{E}\left[r \mid S_{t-1}=s,\ A_{t-1}=a,\ S_t=s' \right]$$

$$\Rightarrow r(s,a) = \sum_{s'} \sum_{r} r\, p(s', r \mid s, a)$$

$$= \sum_{s'} \sum_{r} r\, p(r \mid s, a, s')\, p(s' \mid s, a)$$

$$= \sum_{s'} p(s' \mid s, a)\, r(s,a,s')$$

$$= \mathbb{E}\left[r(s, a, S_t) \right]$$

$r(s,a,S_t)$ is a r.v. which makes the whole function a r.v.

$$r(s,a) = \mathbb{E}\left[r(s,a,S_t) \right]$$

r.v.

$$\text{R.H.S.} = \sum_{s'} p(s' \mid s, a)\, r(s,a,s')$$

$$= \sum_{x} x\, P\left(r(s,a,S_t) = r(s,a,s') \right)$$

注: $\mathbb{E}[f(X)] = \sum_{x} f(x)\, p(f(X)=f(x))$

a r.v. because the input X is a r.v.

$$= \sum_{x} f(x)\, P(X=x)$$

$\mathbb{E}[X] = \sum_{x} x\, P(X=x)$

Replace X with $r(s,a,S_t)$, 就可得出上面的结果.

注: $P(A,B) = P(A \mid B)\, P(B)$

probability of A happen given B

probability of B happened.

$$\therefore r(s,a) = \sum_{s'} \sum_{r} r\, p(s', r \mid s, a)$$

$$= \sum_{s'} \sum_{r} r\, p(r \mid s, a, s')\, p(s' \mid s, a)$$

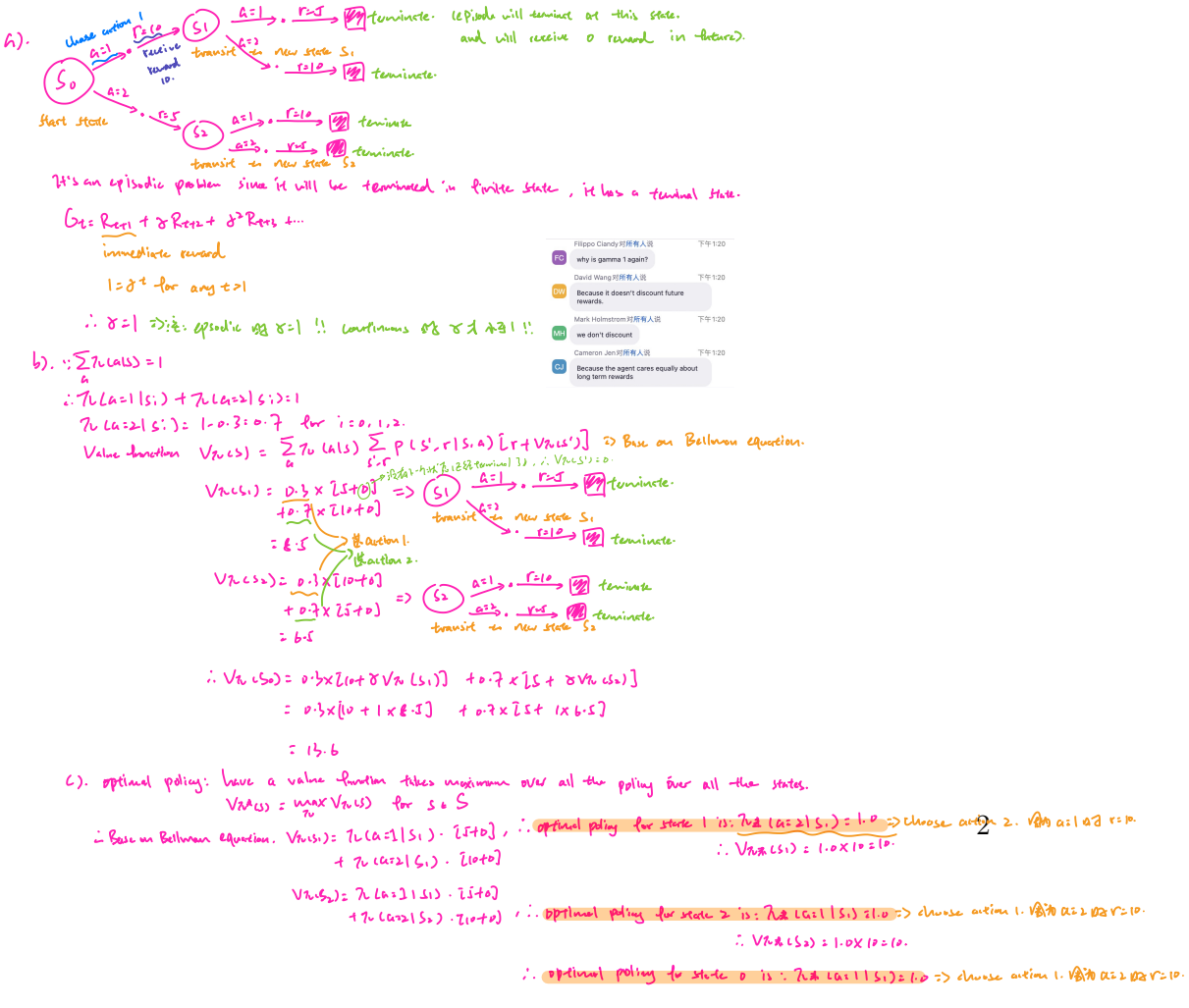
Worksheet 4

2. In this question, you will take a word specification of an MDP, and write the formal terms and determine the optimal policy. Suppose you have a problem with two actions. The agent always starts in the same state, s_0. From this state, if it takes action 1 it transitions to a new state s_1 and receives reward 10; if it takes action 2 it transitions to a new state s_2 and receives reward 5. From s_1 if it takes action 1 it receives a reward of 5 and terminates; if it takes action 2 it receives a reward of 10 and terminates. From s_2 if it takes action 1 it receives a reward of 10 and terminates; if it takes action 2 it receives a reward of 5 and terminates. Assume the agent cares equally about long term reward as about immediate reward.

(a) Draw the MDP for this problem. Is it an episodic or continuing problem? What is γ?

(b) Assume the policy is $\pi(a = 1|s_i) = 0.3$ for all $s_i \in \{s_0, s_1, s_2\}$. What is $\pi(a = 2|s_i)$? And what is the value function for this policy? In other words, find $v_\pi(s)$ for all three states.

(c) What is the optimal policy in this environment?



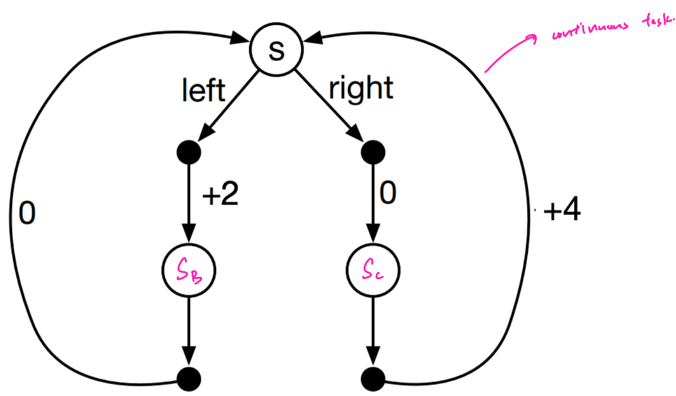
4. (*Exercise 3.22 in* 2^{nd} *ed.*) Consider the continuing MDP shown on the bottom. The only decision to be made is that in the top state, where two actions are available, left and right. The numbers show the rewards that are received deterministically after each action.

(a) List and describe all the possible policies in this MDP.

(b) Is the following policy valid for this MDP (i.e. does if fit our definition of a policy): Choose *left* for five steps, then *right* for five steps, then *left* for five steps, and so on? Explain your answer.

(c) What policy is optimal if $\gamma = 0$? If $\gamma = 0.9$? If $\gamma = 0.5$?

(d) For each possible policy, what is the value of state s? Write down the numeric value to two decimal places. *Hint*: write down the return under each policy starting in state s (don't forget γ). Simplify the infinite sum, using the fact that many rewards are zero. Then plug in the rewards and γ and compute the number.



a). $\pi(a=right \mid S) \in [0,1]$ a real number between 0 and 1.
$\pi(a=left \mid S) = 1 - \pi(a=right \mid S)$
$\pi(a \mid S_B) = 1.0$
$\pi(a \mid S_C) = 1.0$ } 因为 S_B 与 S_C 的时候只有一个action可选择.

b). No. The policy is not Markov. 它需要了它的上一步来做决定 (need history information). 不符合马尔可夫性. (只需要当前状态去预测未来, 决定下一步).
Markov property: the current state include all information to make a decision.

c). For $\gamma=0$: 只考虑下一步利益, 不考虑长远.
$\therefore \pi_*(a=left \mid S) = 1.0$
$\pi_*(a \mid S_B) = 1.0$
$\pi_*(a \mid S_C) = 1.0$
For $\gamma=0.9$: 考虑长远利益.

在 state S 选 action = left 或 action = right 得到的回报.
left: $q_\pi(S, left) = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$
$= 2 + 0.9\times0 + 0.9^2\times2 + \cdots = 2 + \gamma^2\times2 + \gamma^4\times2 + \cdots = 2\cdot\frac{1}{1-\gamma^2} = 10.52$
right: $q_\pi(S, right) = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots$
$= 0 + 0.9\times4 + 0 + 0.9^3\times4 + \cdots = \gamma\cdot4 + \gamma^3\times4 + \gamma^5\times4 + \cdots = 4\cdot\frac{\gamma}{1-\gamma^2} = 18.95$

David Wang对所有人说 下午1:40
DW This policy is not Markov. Its actions depends on its trajectory.

Worksheet 5: Dynamic Programming

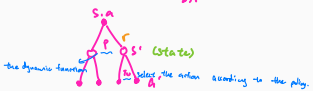
CMPUT 397
February 2, 2022

1. In iterative policy evaluation, we seek to find the value function v_π for an arbitrary policy π. Using the Bellman equation as an update rule, we can generate a sequence of approximate value functions $\{v_k\}$ that will eventually converge to the value function v_π. The expected update is written as:

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

Write the expected update to generate a sequence of approximate action-value functions $\{q_k\}$?

$q_{\pi}(s,a) = \sum_{s',r} p(s',r|s,a)\left[r + \gamma \sum_{a'} \pi(a'|s')\, q_{\pi}(s',a')\right]$



∴ approximate action-value functions $\{q_k\}$ is: $q_{k+1}(s,a) = \sum_{s',r} p(s',r|s,a)\left[r + \gamma \sum_{a'} \pi(a'|s')\, q_k(s',a')\right]$

Worksheet 5: Dynamic Programming

CMPUT 397
February 4, 2022

2. A deterministic policy $\pi(s)$ outputs an action $a \in \mathcal{A}(s) = \{a_0, a_1, \ldots, a_k\}$. More generally, a policy $\pi(\cdot|s)$ outputs the probabilities for all actions $a \in \mathcal{A}(s)$: $\pi(\cdot|s) = [\pi(a_0|s), \pi(a_1|s), \ldots, \pi(a_k|s)]$. How can you write a deterministic policy in this form? Let $\pi(s) = a_i$ and define $\pi(\cdot|s)$.

$\pi(s) = a_i$

meaning: With probability 1, we take action a_i in state s. $\Rightarrow \pi(a_i|s) = 1$.

What about the other actions?

$a \in \mathcal{A}(s) \setminus \{a_i\}$ $\quad \pi(a|s) = 0$ $\quad \pi(a_j|s) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$

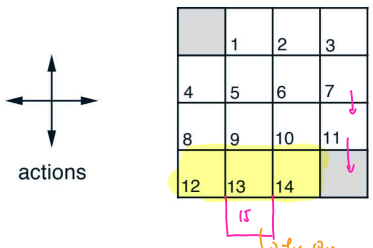
$\{a \mid a \in \mathcal{A}(s) \text{ and } a \neq a_i\}$

$\therefore \pi(\cdot|s) = [\pi(a_0|s), \pi(a_1|s), \ldots, \pi(a_m|s)]$

0th element (subscript is 0)

$\pi(\cdot|s)[j] = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases}$

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 $q_\pi(7, \text{down})$? T = terminal state.



$R_t = -1$ on all transitions

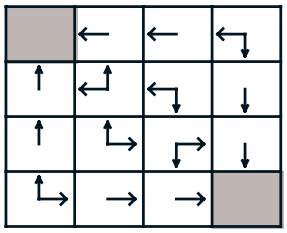
$q_\pi(s,a) = \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$

$\therefore q_\pi(11, \text{down}) = -1 + (1)(0)$ moving from 11 to the terminal state. v_π of terminal state 7.

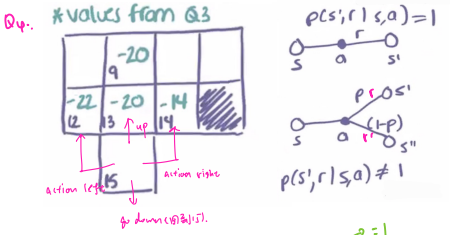
$q_\pi(7, \text{down}) = -1 + (1)(-14) = -15$.

$k = \infty$

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



4. (*Exercise 4.1 from S&B*) Suppose in the above gridworld where a new state 15 is added to the gridworld just below state 13, and its actions, left, up, right, and down, take the agent to the states 12, 13, 14, and 15, respectively. Assume that the transitions from the original states are unchanged. What, then is, $v_\pi(15)$ for the equiprobable random policy? Now suppose the dynamics of state 13 are also changed, such that action down from state 13 takes the agent to the new state 15. What is $v_\pi(15)$ for the equiprobable random policy in this case?



$\therefore v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$ ⇒ Bellman equation

$r = -1$ for all transitions

$\therefore v_\pi(15) = 0.25[(-1 + 1 \cdot v_\pi(12)) + (0.25)(-1 + 1 \cdot (-22)) + (0.25)(-1 + (-14)) + (0.25)(-1 + v_\pi(15))]$

go left · go up · go right · go down

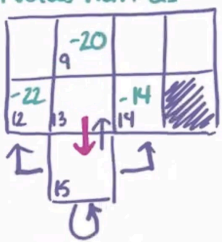
$v_\pi(15) = -15 + 0.25 v_\pi(15)$

$0.75 v_\pi(15) = -15$

$v_\pi(15) = -20$.

4. (*Exercise 4.2 S&B*) In Question 3, suppose a new state 15 is added to the gridworld just below state 13, and its actions, left, up, right, and down, take the agent to states 12, 13, 14, and 15, respectively. Assume that the transitions from the original states are unchanged. What, then, is $v_\pi(15)$ for the equiprobable random policy? Now suppose the dynamics of state 13 are also changed, such that action down from state 13 takes the agent to the new state 15. What is $v_\pi(15)$ for the equiprobable random policy in this case?

* values from Q3



$v_\pi(13) = 0.25\,[-1 + 1\cdot(-22) + (-1 + v_\pi(14)) + (-1 + (-14)) + (-1 + v_\pi(15))]$

$v_\pi(13) = -15 + 0.25\, v_\pi(15)$

$v_\pi(15) = (0.25)\,[(-1 + (-22)) + (-1 + v_\pi(13)) + (-1 + (-14)) + (-1 + v_\pi(15))]$

$= -10 + 0.25\, v_\pi(13) + 0.25\, v_\pi(15)$

$= -10 + 0.25\,[-15 + 0.25\, v_\pi(15)] + 0.25\, v_\pi(15)$

$= -13.75 + 0.1875\, v_\pi(15)$

$v_\pi(15) = -20$

$\therefore v_\pi(13) = -15 + 0.25(-20) = -20$

6. **(Challenge Question)** (*Exercise 4.4 S&B)* The policy iteration algorithm on page 80 has a subtle bug in that it may never terminate if the policy continually switches between two or more policies that are equally good. This is ok for pedagogy, but not for actual use. Modify the pseudocode so that convergence is guaranteed. Note that there is more than one approach to solve this problem.

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
      $\Delta \leftarrow 0$
      Loop for each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))\left[r + \gamma V(s')\right]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      old-action $\leftarrow \pi(s)$
      $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma V(s')\right]$
      If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Worksheet 6: Monte Carlo

CMPUT 397
February 8, 2022

5. Let $\rho_t = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$. (target policy / behavior policy)

(a) Verify that $\mathbb{E}_b[\rho_t|S_t = s] = 1$.

(b) Verify that $\mathbb{E}_b[\rho_t R_{t+1}|S_t = s] = \mathbb{E}_\pi[R_{t+1}|S_t = s]$.

(c) What is the variance of the importance corrected one-step reward, $\mathbb{V}(\rho_t R_{t+1}|S_t = s)$? When would this variance be large?

$\rho_t = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

a). $\mathbb{E}_b[\rho_t | S_t = s]$ based on behavior policy

$= \mathbb{E}_{A\sim b}\left[\frac{\pi(A_t|S_t)}{b(A_t|S_t)} \,\middle|\, S_t = s\right]$ All actions from the behavior policy b.

$= \sum_{a\in\mathcal{A}} \frac{\pi(a|s)}{b(a|s)} b(a|s)$ Actions drawn from the whole actions set. by LoTUS.

$E[g(X)] = \sum_{x\in X} g(x)p(x)$; the law of the unconscious statistician (LOTUS).

$= \sum_a \pi(a|s)$

$= 1$ by definition.

b). $\mathbb{E}_b[\rho_t R_{t+1} | S_t = s] = \mathbb{E}_\pi[R_{t+1} | S_t = s]$ behavior policy / true target policy.

LHS: $\mathbb{E}_b[\rho_t R_{t+1} | S_t = s]$

$= \mathbb{E}_b\left[\mathbb{E}_b[\rho_t R_{t+1} | S_t = s, A_t] \,|\, S_t = s\right]$ LOTE

$= \sum_{a\in\mathcal{A}} b(a|s)\, \mathbb{E}[\rho_t R_{t+1} | S_t = s, A_t = a]$ by LoTUS. (probability of chose action a. 以policy π的概率采样action a.)

$= \sum_a b(a|s) \sum_{s',r} p(s',r|s,a) \frac{\pi(a|s)}{b(a|s)} \cdot r$

$= \sum_a b(a|s) \frac{\pi(a|s)}{b(a|s)} \sum_{s',r} p(s',r|s,a)\, r$

$= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\, r$

$= \sum_a \pi(a|s)\, \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$

$= \mathbb{E}_\pi[\mathbb{E}[R_{t+1} | S_t = s, A_t] \,|\, S_t = s]$

$= \mathbb{E}_\pi[R_{t+1} | S_t = s]$

ρ_t: constant because St and At are given. 根据给的 $\rho_t = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

$\therefore \frac{\pi(a|s)}{b(a|s)}$ constants, since we know s, a. ∴ 可以pull the out.

$V(S_t) \leftarrow V(S_t) + \alpha (R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$ ⇒ when the same variable as an array was taken as an input.

$V_{t+1}(S_t) \doteq V_t(S_t) + \alpha (R_{t+1} + \gamma V_t(S_{t+1}) - V_t(S_t))$

$V_{t+1}(s) \doteq V_t(s) \quad \forall s \in \mathcal{S} - \{S_t\}, \; A_t \sim \pi$

Worksheet 7: TD Learning Methods for Prediction

CMPUT 397

May 2, 2021

1. (*Exercise 6.1 S&B*) If V changes during the episode, then

$$G_t - V(S_t) = \sum_{k=t}^{T-1} \gamma^{k-t} \delta_k$$

only holds approximately; what would the difference be between the two sides? Let V_t denote the array of state values used at time t in the TD error and in the TD update. Redo the derivation to determine the additional amount that must be added to the sum of TD errors in order to equal the Monte Carlo error.

$\because \delta_t = R_{t+1} + \gamma V(S_{t+1}) - V_t(S_t)$

$\therefore G_t - V_t(S_t) = R_{t+1} + \gamma (G_{t+1} - V_t(S_t)) + \gamma V_t(S_{t+1}) - \gamma V_t(S_{t+1})$

(the MC error.)

$= \delta_t + \gamma (G_{t+1} - V_t(S_{t+1}))$

$X_t \doteq G_t - V_t(S_t), \quad X_{t+1} \doteq G_{t+1} - V_{t+1}(S_{t+1})$

$= \delta_t + \gamma (G_{t+1} - V_t(S_{t+1}) + V_{t+1}(S_{t+1}) - V_{t+1}(S_{t+1}))$

$X_t = \delta_t + \gamma (V_{t+1}(S_{t+1}) - V_t(S_{t+1})) + \gamma X_{t+1}$

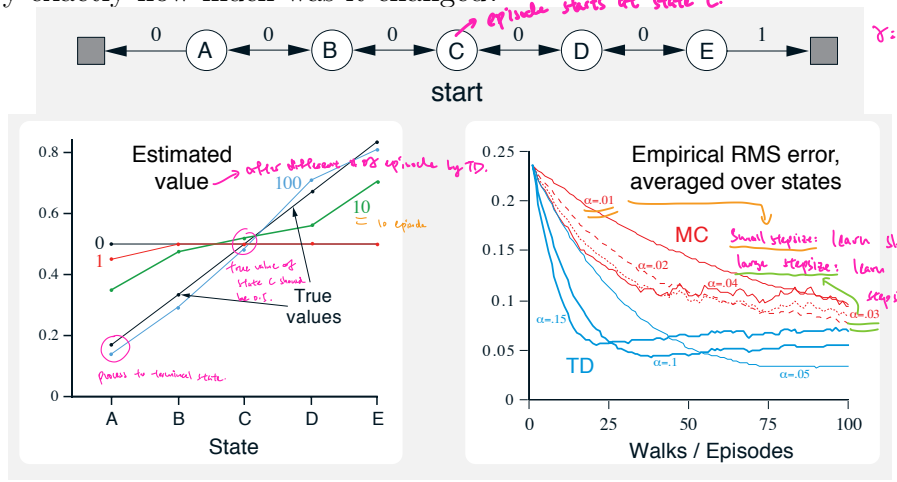
$= \sum_{k=t}^{T-1} \gamma^{k-t} \delta_k + \sum_{k=t}^{T-1} \gamma^{k-t+1} (V_{k+1}(S_{k+1}) - V_k(S_k))$ ← If stepsize is small, V_{k+1} 会 $= V_k$. 这样 value 就不变化了. ∴ 用黄色圈起来的一项 $= 0$.

$X_t = \delta_t + \gamma (V_{t+1}(S_{t+1}) - V_t(S_{t+1})) + \gamma (\delta_{t+1} + \gamma (V_{t+2}(S_{t+2}) - V_{t+1}(S_{t+2})) + \gamma X_{t+2})$

Worksheet 7:
TD Learning Methods for Prediction

CMPUT 397
May 2, 2021

2. (*Exercise 6.3 S&B*) From the results shown in the left graph of the random walk example it appears that the first episode results in a change in only $V(A)$. What does this tell you about what happened on the first episode? Why was only the estimate for this one state changed? By exactly how much was it changed?



MC reward process: transition stochastic.

by randomness of policy: once the action is chosen, the transition is deterministic.

∴ If ends at state A, only the state value of state A will change.

$V(A) \leftarrow V(A) + \alpha[0 + V(L) - V(A)]$

$\leftarrow 0.5 + 0.1[0 + 0 - 0.5]$

$\leftarrow 0.45$

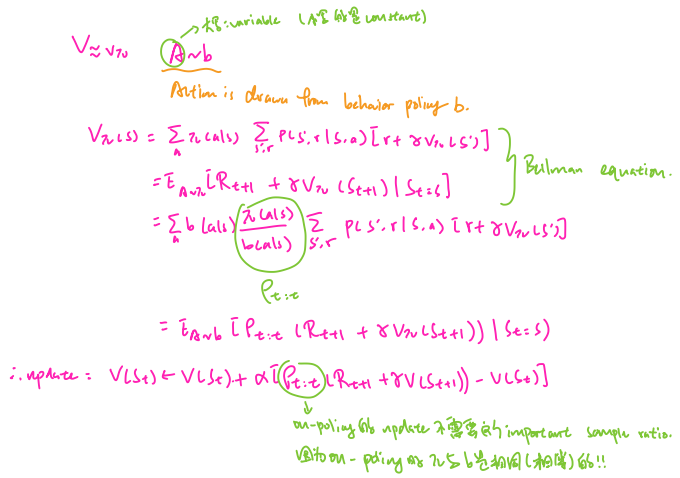
$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$

$\leftarrow V(S_t) + 0.1(R_{t+1} + V(S_{t+1}) - V(S_t))$

Worksheet 7:
TD Learning Methods for Prediction

CMPUT 397
May 2, 2021

5. (*Exercise 6.7 S&B*) Design an off-policy version of the TD(0) update that can be used with arbitrary target policy π and covering behavior policy b, using at each step t the importance sampling ratio $\rho_{t:t}$ (5.3).



Worksheet 8:
TD Learning Methods for Control

CMPUT 397
May 2, 2021

1. Consider the following MDP, with three states B, C and D ($\mathcal{S} = \{B, C, D\}$), and 2 actions ($\mathcal{A} = \{1, 2\}$), with $\gamma = 1.0$. Assume the action values are initialized $Q(s, a) = 0 \ \forall \ s \in \mathcal{S}$ and $a \in \mathcal{A}$. The agent takes actions according to an ϵ-greedy with $\epsilon = 0.1$.

(a) What is the optimal policy for this MDP and what are the action-values corresponding to the optimal policy: $q^*(s, a)$?

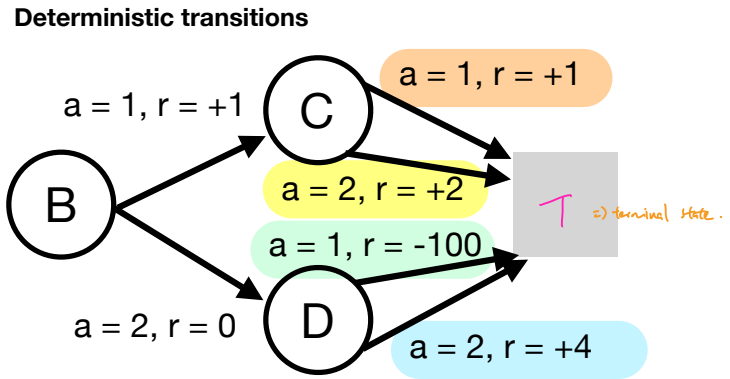
(b) Imagine the agent experienced a single episode, and the following experience: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 2, R_2 = 4$. What are the Sarsa updates during this episode, assuming $\alpha = 0.1$? Start with state B, and perform the Sarsa update, then update the value of state D.

(c) Using the sample episode above, compute the updates Q-learning would make, with $\alpha = 0.1$. Again start with state B, and then state D.

(d) Let's consider one more episode: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 1, R_2 = -100$. What would the Sarsa updates be? And what would the Q-learning updates be?

(e) Assume you see one more episode, and it's the same on as in 1d. Once more update the action values, for Sarsa and Q-learning. What do you notice?

(f) What policy does Q-learning converge to? What policy does Sarsa converge to?



Worksheet 8:
TD Learning Methods for Control

CMPUT 397
May 2, 2021

1. Consider the following MDP, with three states B, C and D ($\mathcal{S} = \{B, C, D\}$), and 2 actions ($\mathcal{A} = \{1, 2\}$), with $\gamma = 1.0$. Assume the action values are initialized $Q(s, a) = 0 \; \forall \; s \in \mathcal{S}$ and $a \in \mathcal{A}$. The agent takes actions according to an ϵ-greedy with $\epsilon = 0.1$.

(a) What is the optimal policy for this MDP and what are the action-values corresponding to the optimal policy: $q^*(s, a)$?

(b) Imagine the agent experienced a single episode, and the following experience: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 2, R_2 = 4$. What are the Sarsa updates during this episode, assuming $\alpha = 0.1$? Start with state B, and perform the Sarsa update, then update the value of state D.

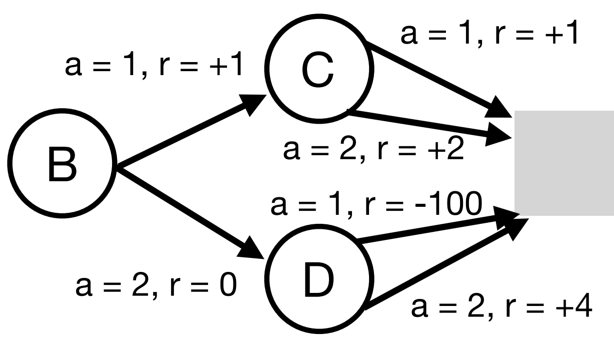
(c) Using the sample episode above, compute the updates Q-learning would make, with $\alpha = 0.1$. Again start with state B, and then state D.

(d) Let's consider one more episode: $S_0 = B, A_0 = 2, R_1 = 0, S_1 = D, A_1 = 1, R_2 = -100$. What would the Sarsa updates be? And what would the Q-learning updates be?

(e) Assume you see one more episode, and it's the same on as in 1d. Once more update the action values, for Sarsa and Q-learning. What do you notice?

(f) What policy does Q-learning converge to? What policy does Sarsa converge to?

Deterministic transitions



c). $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max_{a'} Q(S', A') - Q(S,A)]$ => By Q-learning update

$Q(B,2) = Q(B,2) + 0.1[0 + \gamma \max_{a'} Q(D, a') - Q(B,2)]$

从 state B 开始的 optimal action 为下一步走 state D, 共 action 2, a'=2.

$= 0 + 0.1[0 + 0 - 0] = 0.$

$Q(D,2) = Q(D,2) + 0.1[4 + \gamma \max_{a'} Q(T, a') - Q(D,2)]$

$= 0 + 0.1[4 + 0 - 0]$

$= 0.4$

because the episode generated for sarsa is the greedy policy anyway

d). $S_0 = B$, $A_0 = 2$, $R_1 = 0$
$S_1 = D$ (S'), $A_1 = 1$ (A'), $R_2 = -100$

$Q(B,2) = 0$, $Q(D,2) = 0.4$

[illegible]

By SARSA update.

$Q(B,2) = Q(B,2) + 0.1[0 + \gamma Q(D,1) - Q(B,2)]$

$= 0 + 0.1[0 + 0 - 0] = 0$

$Q(D,1) = Q(D,1) + 0.1[-100 + \gamma Q(T,\cdot) - Q(D,1)]$

$= 0 + 0.1[-100 + 0 - 0]$

$= -10.$

Deepak Ranganatha Sastry Mamillapalli 下午1:51
对所有人说
q learning converges to optimal policy and sarsa converges to epsilon greedy policy? =>
so for q1(f)
what is the answer?

Q-learning

$Q(B,2) = Q(B,2) + 0.1[0 + \gamma \max_{a'} Q(D, a') - Q(B,2)]$ ($= Q(D,2) = 0.4$ (见 b)).

$= 0 + 0.1[0 + 0.4 - 0]$

$= 0.04$

$Q(D,1) = -10$

Worksheet 9:
Planning, Learning & Acting

CMPUT 397
March 19, 2021

1. An agent observes the following two episodes from an MDP,

$$\text{ep 1: } S_0 = 0, A_0 = 1, R_1 = 1, S_1 = 1, A_1 = 1, R_2 = 1.$$

$$\text{ep 2: } S_0 = 0, A_0 = 0, R_1 = 0, S_1 = 0, A_1 = 1, R_2 = 1, S_2 = 1, A_2 = 1, R_3 = 1.$$

and updates its deterministic model accordingly. What would the model output for the following queries:

(a) Model$(S = 0, A = 0)$:

(b) Model$(S = 0, A = 1)$:

(c) Model$(S = 1, A = 0)$:

(d) Model$(S = 1, A = 1)$:

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S,Q)
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
- (e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$

Worksheet 9: Planning, Learning & Acting

CMPUT 397
March 19, 2021

2. An agent is in a 4-state MDP, $\mathcal{S} = \{1, 2, 3, 4\}$, where each state has two actions $\mathcal{A} = \{1, 2\}$. Assume the agent saw the following trajectory,

$$
\begin{aligned}
&S_0 = 1, A_0 = 2, R_1 = -1, \\
&S_1 = 1, A_1 = 1, R_2 = 1, \\
&S_2 = 2, A_2 = 2, R_3 = -1, \\
&S_3 = 2, A_3 = 1, R_4 = 1, \\
&S_4 = 3, A_4 = 1, R_5 = 100, \\
&S_5 = 4
\end{aligned}
$$

and uses Tabular Dyna-Q with 5 planning steps for each interaction with the environment.

(a) Once the agent sees S_5, how many Q-learning updates has it done with **real experience**? How many updates has it done with **simulated experience**?

(b) Which of the following are possible (or not possible) simulated transitions $\{S, A, R, S'\}$ given the above observed trajectory with a deterministic model and random search control?

i. $\{S = 1, A = 1, R = 1, S' = 2\}$

ii. $\{S = 2, A = 1, R = -1, S' = 3\}$

iii. $\{S = 2, A = 2, R = -1, S' = 2\}$

iv. $\{S = 1, A = 2, R = -1, S' = 1\}$

v. $\{S = 3, A = 1, R = 100, S' = 5\}$

Tabular Dyna-Q

Initialize $Q(s, a)$ and $Model(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S, Q)
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
- (e) $Model(S, A) \leftarrow R, S'$ (assuming deterministic environment)
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S, A)$
  - $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$

a).

b). i). iii). iv).

For each update we do based on the real experience, we do n many updates using the simulated experience.

The question shows that there are 5 time steps of state-action pairs and their subsequent next state and reward that we observed from the real experience, and the question also says that we will do 5 planning steps, so n=5

So 5 updates based on real experience, and for each of these updates we do 5 planning steps

-> 5x5 = 25

对于我们基于真实经验所做的每一次更新，我们使用模拟经验做n次更新。

问题显示，有5个时间步骤的状态-动作对及其随后的下一个状态和奖励是我们从真实经验中观察到的，问题还说我们将做5个计划步骤，所以n=5

所以有5个基于真实经验的更新，对于每个更新，我们做5个规划步骤

-> 5x5 = 25

Worksheet 9:
Planning, Learning & Acting

CMPUT 397
March 19, 2021

3. Modify the Tabular Dyna-Q algorithm so that it uses Expected Sarsa instead of Q-learning. Assume that the target policy is ϵ-greedy. What should we call this algorithm?

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$

Loop forever:

(a) $S \leftarrow$ current (nonterminal) state

(b) $A \leftarrow \varepsilon$-greedy(S,Q)

(c) Take action A; observe resultant reward, R, and state, S'

(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)

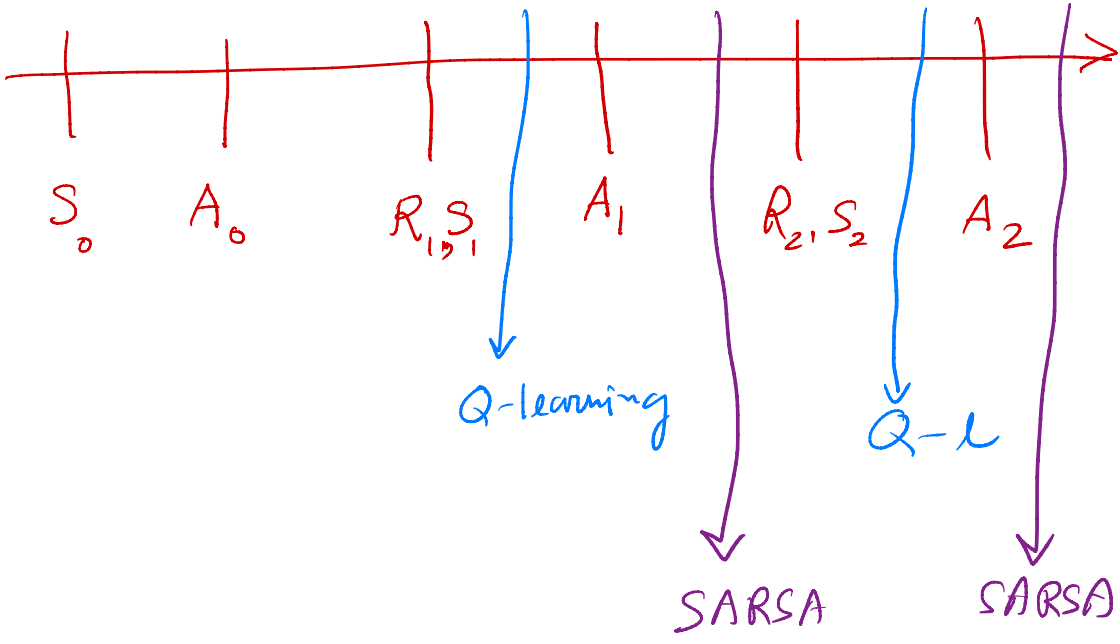
(f) Loop repeat n times:

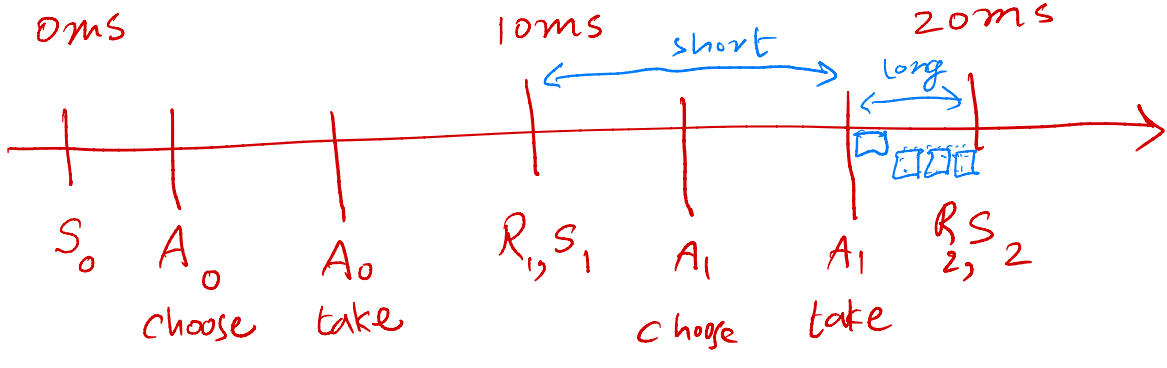
$S \leftarrow$ random previously observed state

$A \leftarrow$ random action previously taken in S

$R, S' \leftarrow Model(S,A)$

$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$


S_0
A_0
R_1, S_1
A_1
R_2, S_2
A_2
Q-learning
Q-l
SARSA
SARSA


0ms
10ms
20ms
short
long
S_0
A_0
choose
A_0
take
R_1, S_1
A_1
choose
A_1
take
R_2, S_2

Worksheet 10: On-policy Prediction with Approximation

CMPUT 397
March 19, 2021

2. Find the gradient of f (function)

(a) if $f(x, y, z) = \frac{y^z}{x}$ (scalars.)

(b) if $f(x) = e^{x^2+5}$ ⇒ 直接对x求导即可.

(c) if $f(\mathbf{x}) = \mathbf{x}^T\beta$ where $\mathbf{x}$ is a vector in $\mathbb{R}^N$ and β is a vector of constants in $\mathbb{R}^N$

(d) if $f(\mathbf{x}) = (\mathbf{x}^T\beta - y)^2$ where $\mathbf{x}$ is a vector in $\mathbb{R}^N$, β is a vector of constants in $\mathbb{R}^N$, and y is a scalar in $\mathbb{R}$

partial derivative (相当于 function f, 只求微分里那一部分的导数).

gradient of f: $\nabla f(x,y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}$ w.r.t x, y ⇒ 比如求 function 中 x 的 derivative.

a). $\nabla f(x,y,z) = \begin{bmatrix} \frac{\partial f(x,y,z)}{\partial x} \\ \vdots \end{bmatrix}$

$\therefore \frac{\partial f(x,y,z)}{\partial x} = y^z \frac{\partial(\frac{1}{x})}{\partial x}$ ⇒ x^{-1} ⇒ $-x^{-2}$

$= \frac{-y^z}{x^2}$

$\frac{\partial f(x,y,z)}{\partial y} = \frac{1}{x} \frac{\partial y^z}{\partial y}$ 只对y的 derivative.

$= \frac{z y^{z-1}}{x}$

$\frac{\partial f(x,y,z)}{\partial z} = \frac{1}{x} \cdot \frac{\partial e^{\ln y^z}}{\partial z}$ ⇒ $e^{z\ln y} = y^z$

$= \frac{1}{x} \cdot \frac{\partial e^{\ln y}}{\partial (z \ln y)} \cdot \frac{\partial (z \ln y)}{\partial z}$

$= \frac{y^z \ln y}{x}$

$\therefore \nabla f(x,y,z) = \begin{bmatrix} \frac{-y^z}{x^2} \\ \frac{z y^{z-1}}{x} \\ \frac{y^z \ln y}{x} \end{bmatrix}$

c). $f(\vec{x}) = \vec{x}^T\vec{\beta}$ $\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$, $\vec{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$

$= \sum_{i=1}^{N} x_i \beta_i$ 因为是2个vector相乘, ∴为 dot product, 如 $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = x_1 \cdot y_1 + x_2 \cdot y_2$.

$\frac{\partial f(\vec{x})}{\partial x_j} = \frac{\partial \sum_{i=1}^{N} x_i \beta_i}{\partial x_j}$

$= \frac{\partial x_j \beta_j}{\partial x_j} + \frac{\partial \sum_{i \neq j} x_i \beta_i}{\partial x_j}$ → doesn't contain any x_j. ∴对x_j求偏导时值为0.

$= \beta_j$

$\therefore \nabla f(\vec{x}) = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix} = \vec{\beta}$

d). $f(\vec{x}) = (\vec{x}^T\vec{\beta} - y)^2$ $\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$, $\vec{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix}$

$\frac{\partial f(\vec{x})}{\partial x_j} = \frac{\partial (\vec{x}^T\vec{\beta} - y)^2}{\partial (\vec{x}^T\vec{\beta} - y)} \cdot \frac{\partial (\vec{x}^T\vec{\beta} - y)}{\partial x_j}$

$= 2(\vec{x}^T\vec{\beta} - y) \cdot \beta_j$ 由c)得到.

$\therefore \nabla f(\vec{x}) = 2 \cdot (\vec{x}^T\vec{\beta} - y) \cdot \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_N \end{bmatrix} = 2 \cdot (\vec{x}^T\vec{\beta} - y) \cdot \vec{\beta}$

Worksheet 10: On-policy Prediction with Approximation

CMPUT 397
March 19, 2021

3. (*Exercise 9.1 S&B*) Show that tabular methods such as presented in Course 2 of the MOOC (and Part I of the book) are a special case of linear function approximation. What would the feature vectors (对 w.) be?

$S = \{s_1, s_2, \dots, s_N\}$ State Space

Tabular case: have N elements in the table: $V(s_1), V(s_2), \dots, V(s_N)$ to be updated: $V(s_i) \leftarrow V(s_i) + \alpha[G - V(s_i)]$

linear function approximation: $\vec{w} = [w_1, w_2, \dots, w_m]^T$

$\vec{x}(s_k) = [x_1(s_k), x_2(s_k), \dots, x_m(s_k)]^T$

$\hat{v}(s_k, \vec{w}) = \sum_{i=1}^{m} w_i x_i(s_k)$

for each $i \in \{1, \dots, m\}$, $w_i \leftarrow w_i + \alpha[G - \hat{v}(s_k, \vec{w})] \cdot \nabla \hat{v}(s_k, \vec{w})$ ⇒ derive MC method from sample average.

(MC error; $= \frac{\partial \hat{v}(s_k, \vec{w})}{\partial w_i}$ ⇒ the feature of $x_i(s_k)$.)

for each $s_k \in S$, assign w_k to approximate its value.

∵ one-hot state feature $x_i(s_k) = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}$

one hot vector: vector 与 state space 一样大，但只有你在的那个 state 为1，其他都为0。如 $[0, 0, \dots, 0, 1, 0, 0, 0]^T$ ($s_1, s_2, \dots, s_x, \dots, s_N$) → 你在的 state ($s_x = k$)

$\hat{v}(s_k, \vec{w}) = w_k$

∴ $\begin{cases} w_i \leftarrow w_i + \alpha[G - w_k] & \text{if } i = k \\ w_i \leftarrow w_i + \alpha[G - w_k] \cdot 0 & \text{if } i \neq k \end{cases}$ ⇒ $w_k \leftarrow w_k + \alpha[G - w_k]$ for the tabular case.

Worksheet 10: On-policy Prediction with Approximation

CMPUT 397
March 19, 2021

4. Consider a random walk with three states $\mathcal{S} = \{1, 2, 3\}$ and a discount rate $\gamma = 0.9$. Suppose $\hat{v}(s; w) = w \cdot s$, where w and s are scalars with w initialized to 0 and $s \in \mathcal{S}$. You observed the following episode:

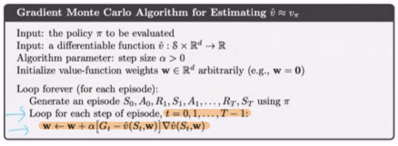
$$S_0 = 1, R_1 = -7, S_1 = 3, R_2 = 5, S_2 = 1, R_3 = 10.$$

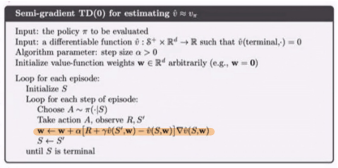
Then if S_3 is the terminal state and $\alpha = 0.1$, what is

(a) the gradient Monte-Carlo estimate for $\hat{v}(1; w)$?

(b) the semi-gradient Linear TD estimate for $\hat{v}(1; w)$?

(c) the estimate for both gradient Monte-Carlo and semi-gradient Linear TD of $\hat{v}(1; w)$ after seeing the same episode again?





a). $w \leftarrow w + \alpha[G_t - S_t w]\, S_t$ → the gradient of $S_t w$.

$\hat{v}(S_t, \vec{w}) = w \cdot s$

$G_2 = R_3 = 10$

$G_1 = R_2 + \gamma G_2 = 14$

$G_0 = R_1 + \gamma G_1 = 3.6$

when $t=0$, $w \leftarrow w + \alpha[G_0 - S_0 w] \cdot S_0$

$= 0 + 0.1[3.6 - 1 \times 0] \cdot 1$

$= 1.12$

$t=1$ $w \leftarrow w + \alpha[G_1 - S_1 w] \cdot S_1$

$= 1.12 + 0.1[14 - 3 \times 1.12] \times 3$

$= 4.31$

$t=2$ $w \leftarrow w + \alpha[G_2 - S_2 w] \cdot S_2$

$= 4.31 + 0.1[10 - 1 \times 4.31] \times 1$

$= 4.88$

b). $w \leftarrow w + \alpha[R + \gamma S' w - S w] S$

∴ when $t=0$, $w \leftarrow w + \alpha[R_1 + \gamma S_1 w - S_0 w] \cdot S_0$

$t=0$

$= 0 + 0.1 \times [-7 + 0.9 \times 3 \times 0 - 1 \times 0] \times 1$

$= -0.7$

$t+1$

$t=1$, $w \leftarrow w + \alpha[R_2 + \gamma S_2 w - S_1 w] \cdot S_1$

$= -0.7 + 0.1[5 + 0.9 \times 1 \times (-0.7) - 3 \times (-0.7)] \times 3$

$= 1.29$

$t+2$

$t=2$, $w \leftarrow w + \alpha[R_3 - S_2 w) \cdot S_2$ (S_2 is terminal state. ∴ don't have $\gamma S_2 w$)

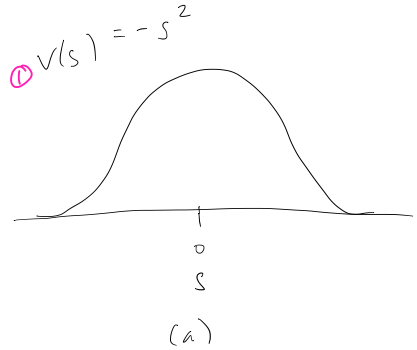
$= 1.29 + 0.1 \times [10 - 1 \times 1.29] \times 1.0$

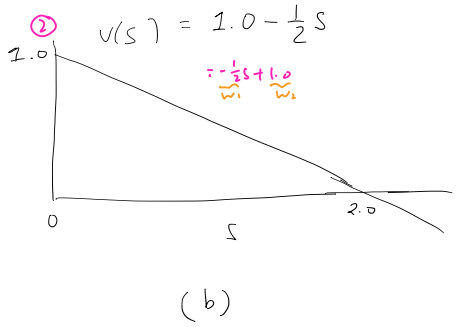
$= 2.116$

Worksheet 11:
Constructing Features for Prediction

CMPUT 397
May 1, 2021

1. Consider the following two functions.





(a) Design features for each function, to approximate them as a linear function of these features. Can you design features to make the approximation exact?

(b) Can you design one set of features, that allows you to represent both functions?

a). ① $f(s) = s^2$

$\hat{v}(s) = w f(s) = w s^2$

$\therefore w^* = -1$

② we have $s, 1$:

$\therefore \hat{v}(s) = w_1 s + w_2 1 = 1 - \frac{1}{2}s$

$\therefore w_1^* = -\frac{1}{2}, w_2^* = 1$

b). $(1, s, s^2) \Rightarrow$ ① & ② 的特征.

$\therefore$ ①: $\hat{v}(s) = w_1 s^2 + w_2 s + w_3 1$

$w_1^* = -1, w_2^* = 0, w_3^* = 0.$

②: $\hat{v}(s) = w_1 s^2 + w_2 s + w_3 1$

$w_1^* = 0, w_2^* = -\frac{1}{2}, w_3^* = 1.$

Worksheet 11: Constructing Features for Prediction

CMPUT 397
May 1, 2021

2. Consider the following neural network $\hat{v}$ with one-hidden layer, relu activation function g, with weights $\mathbf{W}^{[0]}, \mathbf{W}^{[1]}, \mathbf{b}^{[0]}, \mathbf{b}^{[1]}$,

$$\hat{v}(\mathbf{x}; \mathbf{W}^{[0]}, \mathbf{W}^{[1]}, \mathbf{b}^{[0]}, \mathbf{b}^{[1]}) = \mathbf{W}^{[1]} g(\mathbf{W}^{[0]}\mathbf{x} + \mathbf{b}^{[0]}) + \mathbf{b}^{[1]}.$$

output g is also a feature vector.

Recall that the following gradients ⇒ input is a vector. ∴ output should also be a vector.

$$\frac{\partial \hat{v}}{\partial \mathbf{W}^{[1]}_{ij}} = g(\mathbf{W}^{[0]}\mathbf{x} + \mathbf{b}^{[0]})_j$$ ⇒ partial derivative of $w^{[1]}_{ij}$

ij ⇒ i is the feature, j is the input

$$\frac{\partial \hat{v}}{\partial \mathbf{b}^{[1]}_j} = 1$$

$$\frac{\partial \hat{v}}{\partial \mathbf{b}^{[0]}_i} = \sum_k \mathbf{W}^{[1]}_{ki} \frac{\partial g(\mathbf{W}^{[0]}\mathbf{x} + \mathbf{b}^{[0]})_i}{\partial \mathbf{b}^{[0]}_i}$$

feature g.

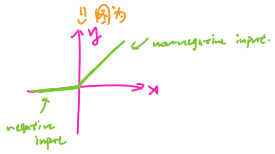
$$\frac{\partial \hat{v}}{\partial \mathbf{W}^{[0]}_{ij}} = \sum_k \mathbf{W}^{[1]}_{ki} \frac{\partial g(\mathbf{W}^{[0]}\mathbf{x} + \mathbf{b}^{[0]})_i}{\partial \mathbf{W}^{[0]}_{ij}}$$

input $w^{[0]}$

(a) What are the derivatives specifically for the relu activation g?

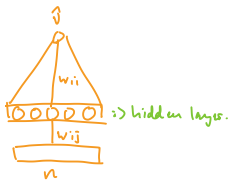
(b) We talked about carefully initializing the weights for the NN. For example, each weight can be sampled from a Gaussian distribution. Imagine instead you decided to initialize all the weights to zero. Why would this be a problem? Hint: Consider the derivatives in (a).

a). $f(a) = \begin{cases} 0 & a < 0 \Rightarrow \text{relu function output is } 0 \text{ if input is negative.} \\ a & a \geq 0 \Rightarrow \text{relu function output is the input itself if the input is non-negative} \end{cases}$



∴ derivative of relu function g would be: $g'(a) = \begin{cases} 0 & a < 0 \\ 1 & a > 0 \end{cases}$

$$\frac{\partial \hat{v}}{\partial w^{[0]}_{ij}} = g(\underline{w}^{[0]}\underline{x} + b^{[0]})_j = \begin{cases} 0 & \text{if } (w^{[0]}x + b^{[0]})_j < 0 \\ (w^{[0]}x + b^{[0]})_j & \text{if } (w^{[0]}x + b^{[0]})_j \geq 0 \end{cases}$$



$$\frac{\partial \hat{v}}{\partial w^{[0]}_{ij}} = w^{[1]}_{ii} \frac{\partial g(w^{[0]}x + b^{[0]})}{\partial w^{[0]}_{ij}} = w^{[1]}_{ii} g'(w^{[0]}x + b^{[0]})_i \cdot x_j$$

$$= \begin{cases} 0 & \text{if } (w^{[0]}x + b^{[0]})_i < 0 \\ w^{[1]}_{ii} x_j & \text{if } (w^{[0]}x + b^{[0]})_i \geq 0 \end{cases}$$

b). If all weights are 0, then no matter where the input is (< 0 or ≥ 0), the output will be 0. (因为 $w \times 0 = 0$)

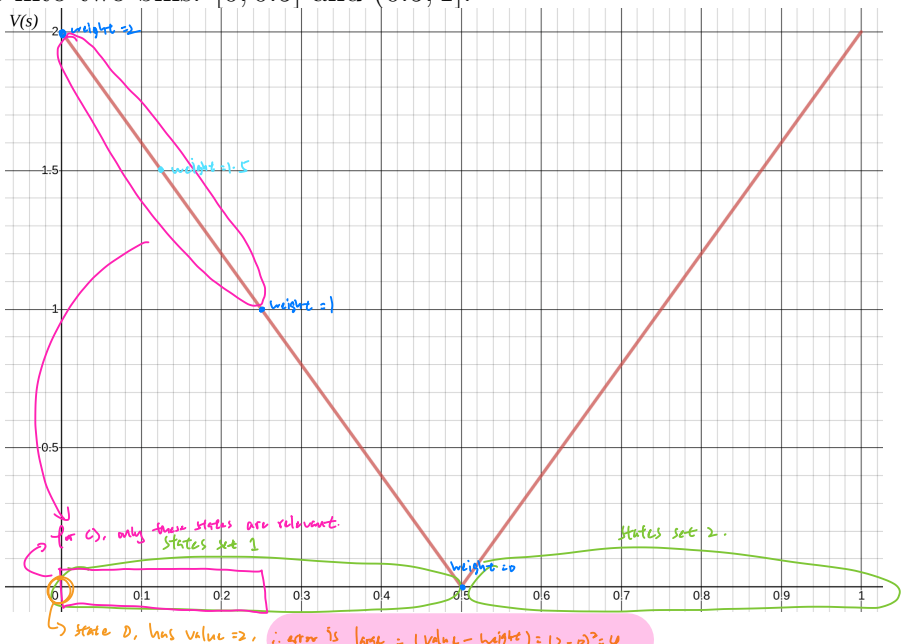
Worksheet 11: Constructing Features for Prediction

CMPUT 397
May 1, 2021

3. Consider a problem with the state space, $\mathcal{S} = \{0, 0.01, 0.02, \cdots, 1\}$. Assume the true value function is

$$v_\pi(s) = 4|s - 0.5|$$

which is visualized below. We decide to create features with state aggregation, and choose to aggregate into two bins: $[0, 0.5]$ and $(0.5, 1]$.



(a) What are the possible feature vectors for this state aggregation?

(b) Imagine you minimize the $\overline{\text{VE}}(\mathbf{w}) = \sum_{s \in \mathcal{S}} d(s)(v_\pi(s) - \hat{v}(s, \mathbf{w}))^2$ with a uniform weighting $d(s) = \frac{1}{101}$ for all $s \in \mathcal{S}$. What vector $\mathbf{w}$ is found?

(c) Now, if the agent puts all of the weighting on the range $[0, 0.25]$, (i.e. $d(s) = 0$ for all $s \in (0.25, 1]$), then what vector $\mathbf{w}$ is found by minimizing $\overline{\text{VE}}$?

a). $[1, 0]^T$
$[0, 1]^T$ 对于 state 0
b). state 1, 对应 x=0 对应 y=2, ∴ error = $(2-0)^2 = 4$
state 2: 对应 x=0.5 对应 y=0 the value is 0, ∴ error = $(0-0)^2 = 0$
对于 state 0.5
if we choose weight => : 对于 state 0
state 1: error = $(2-2)^2 = 0$
state 2: error = $(0-2)^2 = 4$.
对于 state 0.5
if we chose the weight in the middle: 对于 weight = 1: the largest error we can get = $(2-1)^2 = 1$
c). states 对应区间 $[0, 0.5]$, ∴ 为使 error 最小, weight 应为 2, 1.5, 1
∴ we should chose weight $w = [1.5, 1.5]^T$ to give the minimum error.
we want to find $w = [w_1, w_2]^T$ => we chose weight = 0 here, ∴ smaller weight 会 get larger error in smaller states.
error 最大.
∴ the weight should be $w = [1, 1]^T$

Worksheet 12:
Control with Approximation

CMPUT 397
April 7, 2022

3. (*Exercise 10.5 S&B*) What equations are needed, in addition to the equation below,

$$\delta_t = R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t)$$

to specify the differential version of $TD(0)$?

∴ You have to write the TD update that is missing:

∴ $w_{t+1} \doteq w_t + \alpha \delta_t \nabla \hat{v}(S_t, w_t)$ => TD update.

Specify that: { Sample generated by following policy π and $\hat{v}(S_t, w_t) \approx v_\pi(S)$. (on policy learning; 만약 off-policy, 곱해줘야 함 importance sampling ratio!)

$\bar{R}_{t+1} \doteq \bar{R}_t + \beta(R_{t+1} - \bar{R}_t)$.

The update of $\bar{R}$

Worksheet 12:
Control with Approximation

CMPUT 397
April 7, 2022

4. (*Exercise 10.6 S&B*) Suppose there is an MDP that under any policy produces the deterministic sequence of rewards $+1, 0, +1, 0, +1, 0, \ldots$ going on forever. Technically, this is not allowed because it violates ergodicity; there is no stationary limiting distribution μ_π and the limit

$$\lim_{t\to\infty} \mathbb{E}[R_t | S_0, A_{0:(t-1)} \sim \pi]$$

does not exist. Nevertheless, the average reward,

$$\lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[R_t | S_0, A_{0:(t-1)} \sim \pi]$$

is well defined; What is it? Now consider two states in this MDP. From A, the reward sequence is exactly as described above, starting with a $+1$, whereas, from B, the reward sequence starts with a 0 and then continues with $+1, 0, +1, 0, \ldots$ The differential return

$$G_t = R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + R_{t+3} - r(\pi) + \ldots$$

A: 1 0 1 0 1 0 ...
B: 0 1 0 1 0 1 ...

is not well defined for this case as the limit does not exist. To repair this, one could alternately define the value of a state as

$$v_\pi(s) = \lim_{\gamma\to 1} \lim_{h\to\infty} \sum_{t=0}^{h} \gamma^t \left(\mathbb{E}_\pi[R_{t+1} | S_0 = s] - r(\pi)\right)$$

Under this definition, what are the values of states A and B?

Markov reward process: $C_1 = \frac{1}{1} = 1$

$C_2 = \frac{1+0}{2} = \frac{1}{2}$

$C_3 = \frac{1+0+1}{3} = \frac{2}{3}$

$C_4 = \frac{1+0+1+0}{4} = \frac{1}{2}$

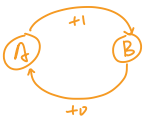
$C_5 = \frac{1+0+1+0+1}{5} = \frac{3}{5}$

$C_6 = \frac{1+0+1+0+1+0}{6} = \frac{1}{2}$

$C_7 = \frac{4}{7}$

$\therefore \lim_{n\to\infty} C_n = \frac{1}{2}$ is the average reward.

=> ∴ even number: all half
odd number: reducing. (converging to $\frac{1}{2}$, ∴ never going below 0.5 !!)



Here, we have a period of 2: ∴ $V_\pi(A) = \frac{1}{2}(2 + (R_1 - r(\pi)) + (R_2 - r(\pi)))$

Starting from A: A 다음은 (1, 0, 1, 0, 1, ...) average reward

$= \frac{1}{2}(2 + (1 - \frac{1}{2}) + (0 - \frac{1}{2}))$

$= \frac{1}{2}(1 - \frac{1}{2})$

$= \frac{1}{4}$

Starting from B: $V_\pi(B) = \frac{1}{2}(2 + (R_1 - r(\pi)) + (R_2 - r(\pi)))$

B 다음은 0이라서, ∴ $R_1 = 0$, $R_2 = 1$. average reward.

$= \frac{1}{2}(2 + (0 - \frac{1}{2}) + (1 - \frac{1}{2}))$

$= \frac{1}{4}$

Confirming accuracy through Bellman equation:

$V_\pi(A) = \frac{1}{4}$

$V_\pi(B) = -\frac{1}{4}$

$\therefore V_\pi(A) = 1 - r(\pi) + V_\pi(B)$

$= 1 - \frac{1}{2} - \frac{1}{4}$

$= \frac{1}{2} - \frac{1}{4}$

$= \frac{1}{4}$ => correct !!

$V_\pi(B) = 0 - r(\pi) + V_\pi(A)$

$= -\frac{1}{2} + \frac{1}{4}$

$= -\frac{1}{4}$ => correct !!

using Abel sum: $V_\pi(B) = \lim_{\gamma\to 1} ((R_1 - r(\pi)) + \gamma(R_2 - r(\pi)) + \gamma^2(R_3 - r(\pi)) + \ldots$

$= \lim_{\gamma\to 1} (1 - \frac{1}{2} + \gamma(0 - \frac{1}{2}) + \gamma^2(1 - \frac{1}{2}) + \ldots)$

$= \lim_{\gamma\to 1} (\frac{1}{2} - \frac{\gamma}{2} + \frac{\gamma^2}{2} \ldots)$

$= \frac{1}{2} \lim_{\gamma\to 1} (1 - \gamma + \gamma^2 - \gamma^3 + \ldots)$

$= \frac{1}{2} \lim_{\gamma\to 1} \frac{1}{1+\gamma}$

$= \frac{1}{2} \cdot \frac{1}{2}$

$= \frac{1}{4}$

Worksheet 12:
Control with Approximation

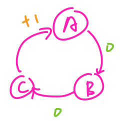
CMPUT 397
April 7, 2022

5. (*Exercise 10.7 S&B*) Consider a Markov reward process consisting of a ring of three states A, B, and C, with state transitions going deterministically around the ring. A reward of +1 is received upon arrival in A and otherwise the reward is 0. What are the differential values of the three states using

$$v_\pi = \lim_{\gamma \to 1} \lim_{h \to \infty} \sum_{t=0}^{h} \gamma^t \left(\mathbb{E}_\pi[R_{t+1} | S_0 = s] - r(\pi) \right)$$

⇒ 这里说 you have to find the answer equal this way!! 不是说你需用自己求的 answer!!

Markov reward process:



∴ Average reward $r(\pi) = \frac{1+0+0}{3} = \frac{1}{3}$

Solve by Bellman equation:

$V_\pi(C) = (1 - \frac{1}{3}) + V_\pi(A)$ (1: 这一步得到的 reward; $\frac{1}{3}$: Average reward; $V_\pi(A)$: 这一步到达的 state.)

$V_\pi(B) = (0 - \frac{1}{3}) + V_\pi(C)$

$V_\pi(A) = (0 - \frac{1}{3}) + V_\pi(B)$

$V_\pi(A) = \frac{1}{3}(3(R_1 - r(\pi)) + 2(R_2 - r(\pi)) + R_3 - r(\pi))$

$= \frac{1}{3}(3(0 - \frac{1}{3}) + 2(0 - \frac{1}{3}) + 1(1 - \frac{1}{3}))$

从 A 开始走，reward 依次为：0 → 0 → 1

$= \frac{1}{3}(-1 - \frac{2}{3} + \frac{2}{3})$

$= -\frac{1}{3}$

∴ can find $V_\pi(B)$ and $V_\pi(C)$ by Bellman equation. 但因为用了 Bellman equation 来 find answer, ∴ 必须用另一种方式验算.

$V_\pi(C) = 1 - \frac{1}{3} + V_\pi(A) = \frac{2}{3} - \frac{1}{3} = \frac{1}{3}$.

$V_\pi(B) = 0 - \frac{1}{3} + V_\pi(C) = -\frac{1}{3} + \frac{1}{3} = 0$.

验证：因为从 A 开始，后 2 个 reward 都为 0，∴ A 的 valuable 程度最低，∴ $V_\pi(A)$ 为负.

B 　 2 个 reward 分别为 0, 1. ∴ B 比 A 高，∴ $V_\pi(B) = 0$.

C 　 下一个 reward 就是 1. ∴ C 最高，∴ $V_\pi(C) = 1$. 最大.

∴ 对.