Date: Subject:

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

discount → $\gamma \in [0, 1]$

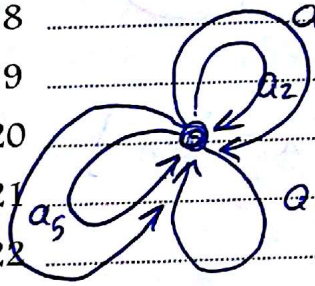
myoptic ← → far-sighted

optimal policy

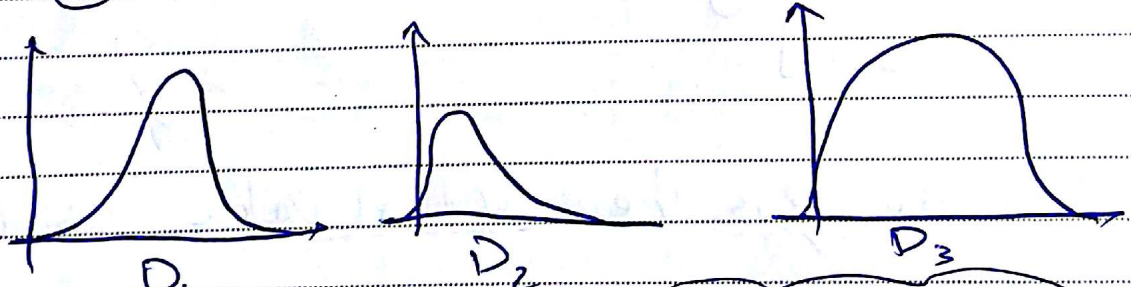
$\pi \geq \pi'$ if $V_\pi(s) \geq V_{\pi'}(s)$, $\forall s$

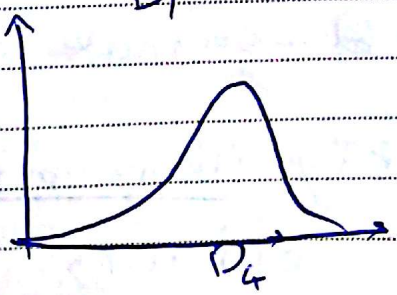
$V_\pi(s)$ → Value of a State (s) under π Policy

Single state problems



→ with every action the state doesn't change





→ بازوی (A)ای را انتخاب می‌کنیم که بیشترین صاحب بیشتری دارد

Date: Action Value Subject:

$$q_\pi(a) = R_s^a = E\{R_{t+1} \mid S_t = S, a_t = a\}$$

Action value estimation at time t

$$Q \doteq \frac{\text{Sum of all rewards}}{\text{sum of all actions}} \Rightarrow \text{Batch mode}$$

$$Q_{n+1} = Q_n + \frac{1}{n}[R_n - Q_n] \Rightarrow \text{online mode}$$

این در واقع همان میانگین گیری است

↳ انتظار داشتیم که مقدار ریوارد برابر Q_n باشد اما مقدار R_n را گرفتیم

The core of R.L $\Rightarrow Q_{n+1} = Q_n + \alpha[R_n - Q_n]$

Learning Rate

$[0, 1]$

Note that the q_π is the actual value which is not accessible. Q is the estimate of q_π

Q will converge to q_π if α satisfy robbins monro conditions

Kian

Date: Subject:

1. $\sum_{0}^{\infty} \alpha = \infty$

2. $\sum \alpha^2 < \infty$

Robbins monro condition

$\sum_{1}^{\infty} \frac{1}{n} = \infty$

$\sum_{1}^{\infty} \frac{1}{n^2} < \infty$

→ So $\frac{1}{n}$ Can be Learning rate

Learning

Explorate → Learning →

- Explorate (2)
- Exploite (8)

(10)

Weight

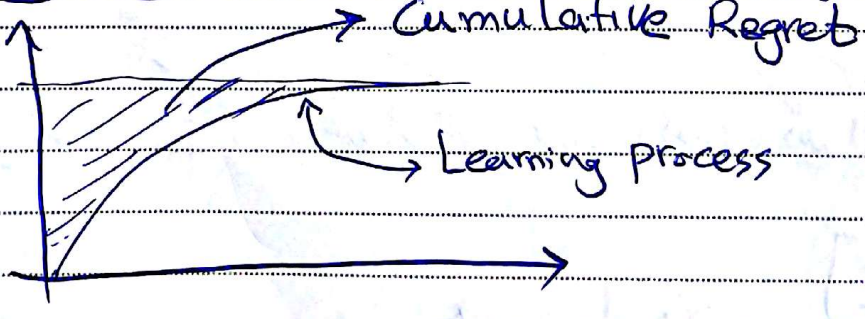
★★★

★★★★

★★★★

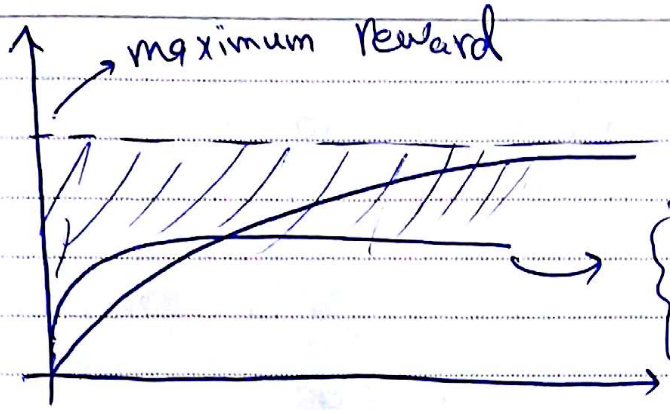
Don't Forget Exploring while Exploiting after being Learned

Regret minimization as a goal



Kian

Date: Subject:



Stuck at suboptimal ~~[illegible]~~ arm

↓

in this Case the ~~[illegible]~~ regret will be as

Exploration – Exploitation rate ⇒ β

ε_Greedy

action 1 → $\frac{\varepsilon}{m}$

action 2 → $\frac{\varepsilon}{m}$

action 3 → $\frac{\varepsilon}{m}$

اکشن بهینه ← action i → $\frac{\varepsilon}{m} + (1-\varepsilon)$

action m → $\frac{\varepsilon}{m}$

احتمال انتخاب شدن اکشن‌ها

ایراد من این است که به غیر از اکشن بهینه به بقیه اکشن‌ها احتمال برابر میدهیم

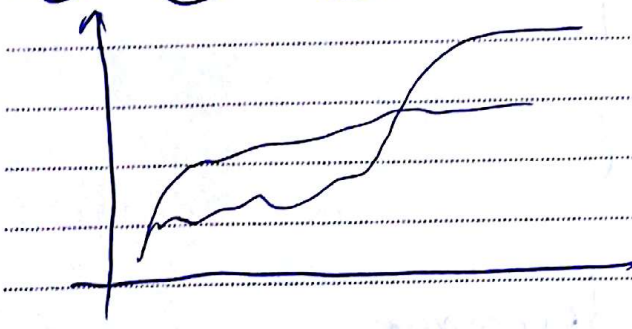
$\varepsilon \in [0, 1]$ → Randomized

↑ Greedy

Kian

Date: Subject:

with initialization of $Q_0(a)$'s we can gain more exploration power



Boltzman Soft max :

$$P(a_i) = \frac{e^{Q(a_i)\beta}}{\sum_{j=1}^{n} e^{\beta Q(a_j)}}$$

Reinforcement Comparison :

$$P_{t+1}(a_t) = P_t(a_t) + \beta(r_t - r'_t)$$

Preference

$$r'_{t+1} = r'_t + \alpha(r_t - r'_t)$$

$$P(a_t) = \frac{e^{P_t(a_t)}}{\sum e^{P_t(a_t)}}$$

Kian

Date: Subject:

Actor Critic

$$a_t^* = \arg\max\left(Q_{t-1}(a)\right)$$

$$\pi_t(a_t^*) = \pi_{t-1}(a_{t-1}^*) - \beta($$

$$\pi_t(a_t) = \pi_{t-1}(a_{t-1}) - \beta($$

Exploration - Exploitation dilemma

More on the Basics

→ Markov Decision Process

→ Policy Gradients

→ Q-Learning

→ DQNs

→ Bellman Equations

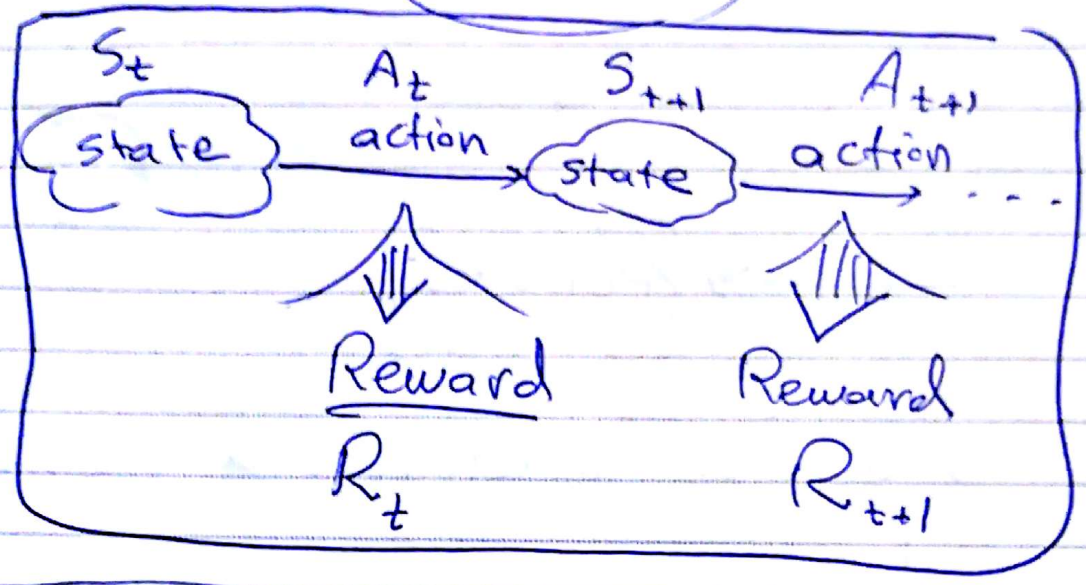
Kian

Date: Subject:

Markov Decision Processes (MDP)

Environment | Agent | States | Actions

Rewards



Return

$$G_t = R_{t+1} + R_{t+2} + \cdots + R_T \rightarrow \text{Final Time Step}$$

Types of Tasks

→ Episodic Tasks

→ Continuing Tasks

It is the agent's goal to maximize the expected discounted return of rewards

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

Kian

Date: Subject:

$$G_t = R_{t+1} + \gamma G_{t+1}$$

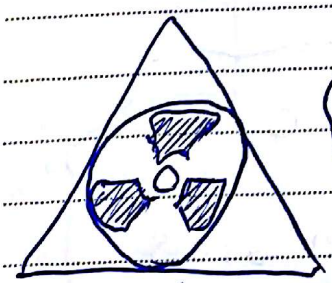
What is the probability that an agent will select a specific action from a specific state?!

→ Policies

How good is a specific action or a specific state for the agent?!

→ Value Function

$\pi(a|s)$ = the probability that $A_t = a$ if $S_t = s$



Important

Note that for each state $s \in S$, π is a probability distribution over $a \in A(s)$

Kian

Date: Subject:

State value Function $\Rightarrow V_{\pi}(s)$

$$V_{\pi} = E\{G_t \mid S_t = s\}$$
$$= E\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right\}$$

Action value Function $\Rightarrow q_{\pi}(s, a)$

$$q_{\pi}(s,a) = E\{G_t \mid S_t = s, A_t = a\}$$
$$= E\left\{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right\}$$

$$q_{\pi}(s,a) = E\{G_t \mid S_t = s, A_t = a\}$$

q_{π} → Q-Function

$E\{G_t \mid S_t = s, A_t = a\}$ → Q-value

optimal policy

$\pi \geq \pi'$ if and only if $V_{\pi}(s) \geq V_{\pi'}(s)$ for $\forall s \in S$

optimal state value function

$$V_*(s) = \max_{\pi} V_{\pi}(s)$$

Kian

Date: Subject:

Optimal action value function

$$q_*(s,a) = \max_{\pi} q_{\pi}(s,a)$$

Bellman optimality equation for q_*

$$q_*(s,a) = E\left\{R_{t+1} + \gamma \max_{a'} q_*(s',a')\right\}$$

(s': the state which the best possible next action a' can be taken at time t)

Q-Learning

→ we will face exploration_exploitation delimma

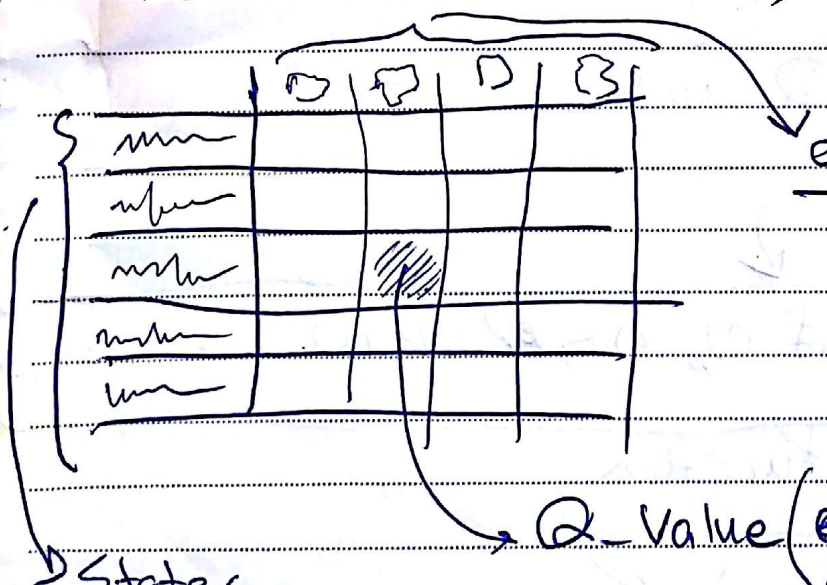
How To solve ?!

⇒ epsilon greedy

⇒ Boltzman Softmax

⇒ Actor critic

⇒ Reinforcement Comparison



Date: Subject:

Learning rate

$$Q^{new}(s,a) = (1-\alpha)\underbrace{Q_t(s,a)}_{\text{old value}} + \alpha\left(R_{t+1} + \gamma \max_{a'} Q_t(s',a')\right)$$

Date: Subject:

Second Session 6

$$G_t = R_{t+1} + \gamma G_{t+1}$$ → Recursive Return

→ Bellman Equation For $V(s)$

$$V(s) = E\{G_t \mid S_{(t)} = s\}$$
$$= E\{R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s\}$$

$$V(s) = R_s + \gamma \sum P_{ss'} \cdot V(s')$$

(اخر) (از) (به)

$$\begin{pmatrix} V(1) \\ V(2) \\ \vdots \\ V(n) \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{pmatrix} + \gamma \begin{pmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \vdots \\ P_{n1} & \cdots & P_{nn} \end{pmatrix} \begin{pmatrix} V(1) \\ V(2) \\ \vdots \\ V(n) \end{pmatrix}$$

$$q_\pi(s,a) = E\{G_t \mid S_{(t)} = s, A_{(t)} = a\}$$

$$V_\pi(s) = \sum_{a \in A} \pi(a|s)\, q_\pi(s,a)$$

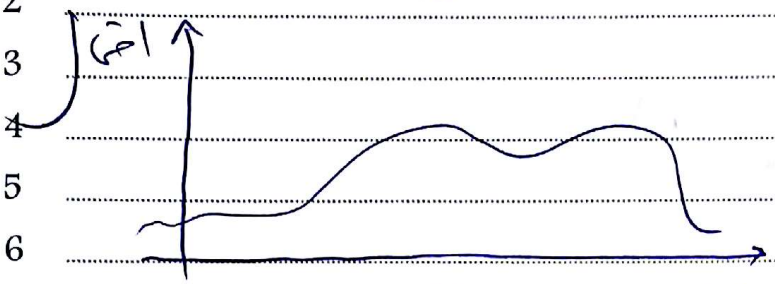
$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \cdot V_\pi(s')$$

(حتما بلدیم) (جمع‌ها) (نوشته شده)

Kian

Date: Subject:

اینکه هر عملی را به چه احتمالی انجام دهیم را با $\pi(s|a)$ تعیین می‌کنند.



actions in states

$$V_\pi(s) = \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s') \right)$$

$q_\pi(s,a)$

$$R_s^a = \sum_{a \in A} \pi(a|s) R_s^a$$

$V_\pi(s)$ رابطه بازگشتی

$$P_{ss'}^a = \sum_{a \in A} \pi(a|s) P_{ss'}^a$$

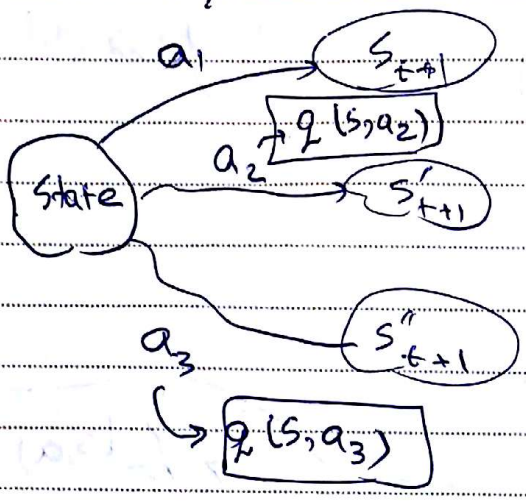
$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi(s')$$

$$= R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s',a')$$

$q_\pi(s,a)$ رابطه بازگشتی

Kian

Date: $q_*(s,a_i)$ Subject:



وقتی که q مشخص q داریم

که ماکسیمم است $\Leftarrow q_*(s,a_i)$

$$V_*(s) = \max_a q_*(s,a)$$

$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_*(s')$$

Dynamic Programming

Solve Complex problems by Breaking them to subproblems

I) iterative policy evaluation

assumption ⇒ we have fully defined environment

$$V_{k+1}(s) = \sum_{a \in A} \pi(a|A) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_k(s') \right)$$

Kian

Date: Subject:

Policy Evaluation

loop:

loop
- Policy evaluation
- Policy improvement

$q_\pi(s,a)$

$$V_1(6) = \sum_a \pi(a|6) \sum_{s',r} p(s',r|6,a)[r + \gamma V_0(s')]$$

1 -1 0 $\forall s'$

$\pi(a|6)$: $\frac{1}{4}\ \forall a$ → Random policy

$$= 0.25 \times \{-p(2|6,u) - p(10|6,d) - p(5|6,l) - p(7|6,r)\}$$

$$= \frac{1}{4} \times (-4) = -1$$

	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

این پرسش بایه ۱ یا ۱۶ پرسد.

همچنان ریوارد ۱- دارد.

0	-14	-20	-22
-14	-18	-20	-20
-20	-20	-18	-14
-22	-20	-14	0

after Convergence

$p(s',r|s,a)$ متغیر تصادفی

Kiarash

Date: Subject:

Policy improvment 8

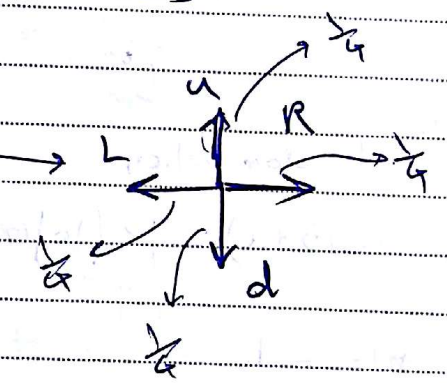
$$q_{\pi}(s, \pi'(s)) = \max_a q_{\pi}(s,a) \geq q_{\pi}(s, \pi(a)) = V(s)$$

$$\rightarrow \pi' \geq \pi$$

in a greedy manner we choose the action that has a better $q(s,a)$ $\pi(s) = \arg\max_a (q(s,a))$

$$q_{\pi}(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V(s')$$

0	-4	-20	-22
-4	-18	-20	-20
-20	-20	-18	-14



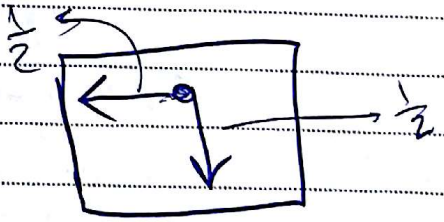
$$q(s,a) = \sum_{s',r} P(s',r|s,a)\left[r + \gamma V(s')\right]$$

$$\rightarrow q(7, R) = 1 \times (-10 - 20) = -21$$

$$q(7, U) = -21$$

$$q(7, D) = -19$$

$$q(7, L) = -19$$



Kian

Date: Subject:

$q_\pi(s,a)$ → اگر Policy π را در پیش گیریم، اگر در حالت s، عمل a را انجام دهیم (توصیه نمی‌شد) را انجام دهیم.

$q(s,\pi(s))$ → اگر Policy π را در پیش گیریم، اگر در حالت s، عملی را که Policy π توصیه می‌کند را انجام دهیم

Value Iteration

Loop:

$\Delta \leftarrow 0$

Loop For each $s \in S$

$v \leftarrow V(s)$

$$V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)[r+\gamma V(s')]$$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until $\Delta < \theta$

Output a deterministic policy $\pi \approx \pi_*$, such that

$$\pi(s) = \arg\max_a (q_\pi(s,a))$$

$$= \arg\max_a \left| p(s',r|s,a)[r+\gamma V(s')] \right.$$

Kian

Date: Subject:

a note on How the Equations and Formulas of $q_\pi(s,a)$ and $V(s)$ are related to each other &

The main Formula and definition

$$V(s) = E\{G_t \mid S_t = s\}$$

$$q_\pi(s,a) = E\{G_t \mid S_t = s \quad A_t = a\}$$

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

$$= R_{t+1} + \gamma G_{t+1} \Rightarrow \boxed{G_t = R_{t+1} + \gamma G_{t+1}}$$

$$V(S_{t+1}) = E\{G_{t+1} \mid S_{t+1} = s\}$$

$$\rightarrow E\{V(S_{t+1})\} = E\{G_{t+1} \mid S_{t+1} = s\}$$

$$\rightarrow V(S_t) = E\{R_{t+1} + \gamma G_{t+1} \mid S_t = s\}$$

$$= E\{R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s\}$$

$$= \sum_{s',r} p(s',r \mid s,r)\left(r + \gamma V_\pi(S_{t+1})\right)$$

~~$V(S_t) = \sum_{s',r} p(s',r \mid s,r)\left(r + \gamma V_\pi(S_{t+1})\right)$~~

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

Kian

Date: Subject:

Therefore we will have

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

Model Free Learning :

↳ unLike The perivious method (dynamic programming)

in this Case we don't have the model of env

→ the MDP is unknown

Monte Carlo Learning :

$$V_{\bar{\pi}}(s) = E\left[G_{\bar{\pi}} | S_t = s\right]$$

input : policy $\bar{\pi}$ to be evaluated

initialize :

$V(s) \in \mathbb{R}$, arbitrary, For all $s \in S$

returns(s) ← an empty list, For all $s \in S$

Loop Forever (For each episode):

Generate an episode Following $\bar{\pi}$: $S_0, A_0, S_1, A_1, S_2, A_2, \ldots$

$G \leftarrow 0$

Loop For each step of episode : $t = T-1, T-2, T-3, \ldots$

unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$

append G to Returns(S_t)

$V(S_t) \leftarrow$ average(Returns(S_t))

Kian

Date: Subject:

NL

$$V(S_t) = V(S_t) + \alpha\left(G_t - V(S_t)\right)$$

Temporal Difference :

$$V_\pi(s) = E_\pi[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s]$$

$$V_\pi(s) = E_\pi[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s]$$

$$V(S_t) \leftarrow V(S_t) + \alpha\left[G_t - V(S_t)\right]$$

$$G_t \rightarrow R_{t+1} + \gamma V(S_{t+1})$$

$$\Rightarrow V(S_t) \leftarrow V(S_t) + \alpha\left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right]$$

Monte Carlo $\Rightarrow$ unBiased – Higher variance

Temporal Difference $\Rightarrow$ Biased – lower variance

Kian

Date: / Subject:

در مونت کارلو تا انتهای روی دنباله ولیدیت را آپدیت میزنیم.

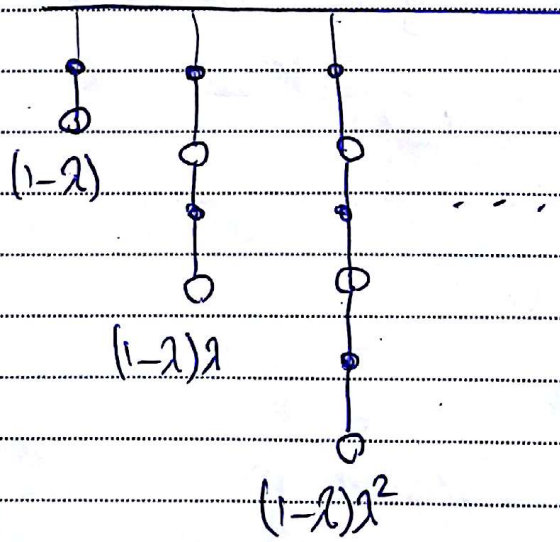
در تمپورال دیفرنس n قدم جلو میریم و بعدش ولیو را آپدیت میکنیم.

n-Step Bootstrapping

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n V(S_{t+n})$$

n-Step return

→ Averaging n-steps



Eligibility Returns

Frequency heuristic / Recency heuristic

Kian

Date: Subject:

Egibility Trace

Monte Carlo Control & (OFF policy)

Sudo Code :

Date: Subject:

Because the policy is Greedy, so Monte Carlo Control will not have any Exploration

↓

برای حل این مشکل، بیاد بیا تشل & ما، بهست ε-greedy عمل کنیم

off policy Monte Carlo Control

π' ⇒ Behavior policy ⇒ Soft

π ⇒ Estimation policy ⇒ Greedy

if $\pi(a|s) > 0 \Rightarrow \pi'(a|s) > 0$

$\pi(a|s)$

↓

احتمال انتخاب action a وقتی در s هستیم.

$P_{ss'}^{a}$

احتمال گذار از s به s' با انتخاب a

این را بهصورت greedy آپدیت π می‌کنیم

این را بهصورت soft آپدیت π می‌کنیم

Kian

Date: Subject:

How to Compute $V_\pi(s) = E_\pi[G_t | S_t = s]$

$$V_\pi(s) = E_\pi[G_t | S_t = s] = E_{\pi'}[?! | S_t = s]$$

$$E_{x\sim P}[F(x)] = \sum P(x) F(x) = \sum Q(x) \frac{P(x)}{Q(x)} F(x)$$

$$= E_{x\sim Q}\left[\frac{P(x)}{Q(x)} F(x)\right]$$

$$V_\pi(s) = E_\pi[G_t | S_t = s] = E_{\pi'}[w_t G_t | S_t = s]$$

$$= \frac{\sum w_t G_t}{\sum w_t}$$

$$P'(E_{st}) = \prod_{k=t}^{T-1} \pi'(S_k, a_k) P_{S_k, S_{k+1}}^{a_k}$$

$$P$$

$$w_t = \frac{P}{P'} = \frac{\prod_{k=t}^{T-1} \pi(S_k, a_k)}{\prod_{k=t}^{T-1} \pi'(S_k, a_k)}$$

Kian

Date: Subject:

$$V_{n+1}(s) = V_n(s) + \frac{W_n}{C_n}\left[G_n - V_n(s)\right]$$

$$C_{n+1} = C_n + W_{n+1}$$

updating V_n incrementally

Sudo code:

initialize For all $s \in S$, $a \in A(s)$

$C(s,a) \leftarrow 0$

$\pi(s) \leftarrow \arg\max_a Q(s,a)$

Loop For ever (For each episode):

$\pi' \leftarrow$ update using any soft policy update (like ε-greedy)

Generate an episode: π': $S_0, A_0, S_1, A_1, S_2, A_2, \ldots$

$G \leftarrow 0$

$W \leftarrow 1$

Loop For each step of episode: $T-1, T-2, \ldots$

$G \leftarrow \gamma G + R_{t+1}$

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}\left[G - Q(S_t, A_t)\right]$

$\pi(S_t) \leftarrow \arg\max(Q(S_t, a))$

iF $A_t \neq \pi(S_t)$ then exit the loop

$W \leftarrow W \times \frac{1}{\pi'(A_t, S_t)}$

Kian

Date: Subject:

How To use TD instead of MC in MC Control?!

Sarsa ⇒ on policy TD control

Q, V of Terminals are always Zero

Sudo Code:

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$.
initialize $Q(s,a)$, For all $s \in S^+$, $a \in A(s)$
$Q(\text{terminal}, \cdot) = 0$
Loop For each episode:
  initialize S
  choose A From S using policy derived From Q (e.g. ε-greedy)
  Loop For each step of episode:
    Take action A, observe R, S'
    Choose A' From S' using policy derived From Q (e.g. ε-greedy)
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
  until S is terminal

n-step Sarsa

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[q_t^{(n)} - Q(S_t, A_t)\right]$$

$$q_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n Q(S_{t+n}, A_{t+n}).$$

Sarsa λ:

$$q_t^{\lambda} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} q_t^{(n)}$$

Kian

Date: Subject:

Q-Learning : off policy TD Control

π' → Behavorial (Soft) (will be updated using ε-Greedy)

π → Estimational

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t)]$$

Sudo Code:

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$

initialize $Q(s,a)$, for all $s \in S^+$, $a \in A(s)$,

$Q(terminal, \cdot) = 0$

Loop For each step of episode:

choose A From S using policy derived From Q (e.g ε-Greedy)

Take action A, observe R, S'

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

$S \leftarrow S'$

until S is terminal

در واقع با هر بار زدن می‌ریم و یه بار Q رو آپدیت می‌کنیم

Kian