Start Reading Sutton and Barto!

24th Sept

MONTE-CARLO METHODS

(1) Prediction

Given π, Find v^π through interaction with MDP.

Control (Also wants to change and converge to a good policy)

$v^\pi(s)$ Episodic Task Assumption

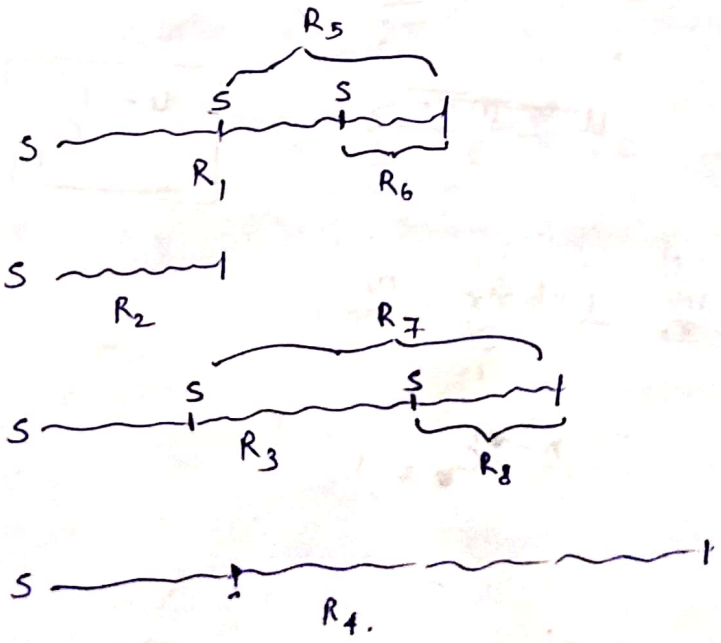
i : Run $s_i^0\ a_i^0\ r_i^0\ s_i^1\ a_i^1\ r_i^1\ s_i^2 \text{ ---- } s_i^{a_i}$

i^{th} "Return" : $r_i^0 + \gamma r_i^1 + \gamma^2 r_i^3 + \text{-----} \gamma^{a_i - 1} r_i^{a_i - 1}$

$R_1, R_2, R_3 \text{ --- } R_n$

Estimate $v^\pi \approx \frac{1}{n}(R_1 + R_2 + \text{---} R_n)$

$v^\pi(s)$



$v^\pi_1(s) = \frac{R_1 + R_2 + R_3 + R_4}{4}$

Instead, $v^\pi_2(s) = \frac{R_1 + R_2 + R_3 + R_4 + R_5 + R_6 + R_7 + R_8}{8}$

(2)

Is $\lim_{n\to\infty} V_1^{\pi}(s) = V^{\pi}(s)$? Yes → FIRST-VISIT MONTE CARLO

so $\lim_{n\to\infty} V_2^{\pi}(s) = V^{\pi}(s)$? Yes ! Proof different from 1st one.
↳ EVERY-VISIT MONTE CARLO.

$\forall n \; E[V_1^n(s)] = V^{\pi}(s)$ Unbiased Estimator for V^{π}

$E[V_2^n(s)] \neq V^{\pi}(s) \rightarrow = V^{\pi}(s) + [b(SATRY), n]$

As long as you don't look at the future, you're okay, but if you something like taking last or second-last occurrence of s is not correct.

Every-visit probably converges faster.

(2) Control

Assumption of exploring starts.

Monte Carlo's exploring states

```
Q[ ][ ] ← 8
π ← Arbitrary policy
visits[ ][ ] ← 0
For each episode: e = 1, 2, ---
    . Pick s∈S uniformly at random →
      Pick a∈A uniformly at random
      start at s, taken a, follow π, let return be R.
      Q[s][a] ← (Q[s][a] * visits[s][a] + R) / (visits[s][a] + 1)
      visits[s][a] ← +=1
      π ← greedy(Q)
        ↳ its argmax_{a∈A} Q(s,a).
```

(Suppose you pick s using arbit distribution on S.) → need not converge.

Is $\pi_{e\to\infty} = \pi^*$? → This will converge.

If you converge you'll converge to optimal policy

Problem: (Probably Approximately Correct (PAC) Formulation)

Continuing Tasks

Exploring starts (Restart occurs) (s, a)

Rewards b/w $-R_{max}$ & R_{max}.

Prediction: V^{π} Given π

Write an algorithm L

Required: Let L be output V_L as the ____

We want with probability at least $(1-\delta)$ that

$$\forall s \in S \quad |V_L(s) - V^{\pi}(s)| \leq \epsilon$$

(i) How many episodes?

(ii) Length of each episode?

$? > 1000 \cdot R_{max} < \epsilon$.

Try and work out yourselves.

Next: Bootstrapping.

1st Oct

① Estimators

Process with parameter $p \in [0, 1/2]$

$x \sim$ Bernoulli (p)

$y \sim$ Bernoulli $(2p)$

Run process $x = 1, y = 0$

What is p?

Lets take $\hat{p} = \dfrac{x + \frac{y}{2}}{2}$

* Least - squares estimator

$$\hat{p} = \underset{p \in [0,1/2]}{\operatorname{argmin}} \; (p-1)^2 + (2p-0)^2$$

$$= \underset{p \in [0,1/2]}{\operatorname{argmin}} \; p^2 - 2p + 1 + 4p^2$$

$p = 0.2$

* Maximum Likelihood Estimator

$$\hat{p} = \underset{p \in [0,1/2]}{\operatorname{argmax}} \; p(1-2p) = 0.25$$

$$p - 2p^2$$

② π V^{π} s

$s \sim\sim R_1$

$s \sim\sim R_2$

$\sim R_3$

$\vdots$

$\sim\sim R_n$.

$$V_n(s) = \frac{R_1 + R_2 + \cdots R_n}{n}$$

$$V_{n+1}(s) = \frac{R_1 + R_2 + \cdots R_{n+1}}{n+1}$$

$$= V_n(s)\left(1 - \frac{1}{n+1}\right) + R_{n+1}\left(\frac{1}{n+1}\right)$$

Result (Robbins and Monro, 1951)

Let $(\alpha_i)_{i=1}^{\infty}$ be a sequence such that $\sum_{i=1}^{\infty} \alpha_i = \infty$ and $\sum_{i=1}^{\infty} \alpha_i^2 < \infty$,

Consider updating $V_{n+1}(s) \leftarrow V_n(s)\{1 - \alpha_{n+1}\} + \alpha_{n+1} R_{n+1}$.

Learning Rate.

Then, $\lim_{n \to \infty} V_n(s) = V^{\pi}(s)$.

Stochastic Approximation – Robbins & Monro.

(3) $s = s_{n+1}^0\ a_{n+1}^0\ r_{n+1}^0\ s_{n+1}^1\ a_{n+1}^1\ r_{n+1}^1 \cdots$

Monte-carlo Method $\leftarrow V_{n+1}(s) \leftarrow V_n(s)\{1 - \alpha_{n+1}\} + \alpha_{n+1}\{r_{n+1}^0 + \gamma r_{n+1}^1 + \cdots \gamma^2 r_{n+1}^2 \cdots\}$

Bootstrapping $\leftarrow V_{n+1}(s) \leftarrow V_n(s)\{1 - \alpha_{n+1}\} + \alpha_{n+1}\{r_{n+1}^0 + \gamma V_n(s_{n+1}^1)\}$

TD(0)

↓

Form of Full Bootstrapping.

$$V_{n+1}^{TD(0)}(s) \leftarrow V_n^{TD(0)}(s) + \alpha_{n+1}\{\underbrace{r_{n+1}^0 + \gamma V_n^{TD(0)}(s_{n+1}^1)}_{\text{New Estimate}} - \underbrace{V_n^{TD(0)}(s)}_{\text{Old Estimate}}\}$$

Temporal Difference prediction error.

(4) TD(0) Algorithm

$V_0 \leftarrow$ Initial guess of V^{π}

Assume that agent is born in state s^0

For $t = 0, 1, 2 \cdots$

Take action $a^t = \pi(s^t)$

Obtain r^t, s^{t+1}

$$V_{t+1}(s^t) \leftarrow V_t(s^t) + \alpha_{t+1}\{r^t + \gamma V_t(s^{t+1}) - V_t(s^t)\}$$

$\lim_{t \to \infty} V_t = V^{\pi}$

(5) Batch MC and Batch TD(0)

TD(1)

s a r s a r s a r s

s a r s a r s

s a r s

TD(λ) $0 \le \lambda \le 1$

s a r s a r s a r s

Assume that episodes loop-back after end.

Least-Squared Error Estimate

V^{MC}_{Batch} (Dataset set of size N)

$V^{TD(0)}_{Batch}$ (Dataset of size N)

} Will they converge to the same thing or different thing?

Value fⁿ of the MDP to have most-likely generated this data. (MLE)

(1) Multi-Step Return s^t

4th Oct

$$G_{t:t+1} = r^t + \gamma V_t(s^{t+1})$$

$$G_{t:t+2} = r^t + \gamma r^{t+1} + \gamma^2 V_{t+1}(s^{t+2})$$

$$G_{t:t+n} = r^t + \gamma r^{t+1} + \cdots \gamma^n V_{t+n-1}(s^{t+n})$$

$$G_t = \lim_{n \to \infty} G_{t:t+n}$$

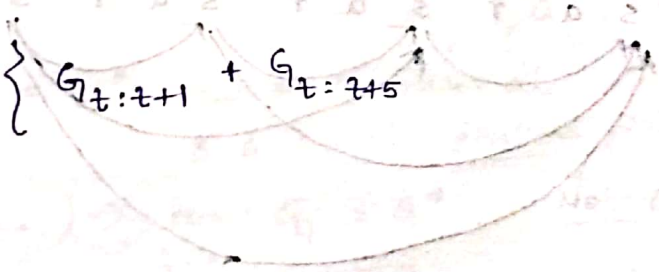
$$V_{t+1}(s^t) \leftarrow V_t(s^t) + \alpha_{t+1}\{G_{t:t+n} - V_t(s^t)\}.$$

$$V_{t+1}(s^t) \leftarrow V_t(s^t) + \alpha_{t+1}\left\{\frac{G_{t:t+1} + G_{t:t+5}}{2} - V_t(s^t)\right\}$$

$\leftarrow$ Need not work. $\leftarrow \times \{1.5\, G_{t:t+2} - 0.5\, G_{t:t+3}$

The minus sign might be a problem.

" " $\leftarrow \{G_{t:t+1} + G_{t:t+5}$



w_1	$G_{t:t+1}$
w_2	$G_{t:t+2}$
w_3	$G_{t:t+3}$
$\vdots$	$\vdots$
w_n	$G_{t:t+n}$

$\downarrow$

Any convex combination of weights would work, coefficients non-negative, sum to 1.

(2) λ – return

$\lambda \in [0, 1]$

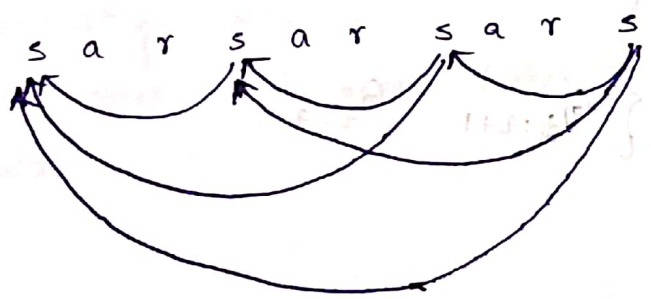
λ – return

$$G_t^{\lambda} = (1-\lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_t$$

where T is the last time-step in the episode.

$$V_{t+1}(s_t) \leftarrow V_t(s_t) + \alpha_{t+1} \left\{ G_{t:t+n} - V_t(s_t) \right\}$$

(3) TD(λ) Algorithm



Initialise V

Repeat for each episode

$e \leftarrow 0$ // $e : S \rightarrow \mathbb{R}$, eligibility trace vector

Be born in state s

Repeat for each step of episode

Take action $a = \pi(s)$, obtain r and s'

$\delta \leftarrow r + \gamma V(s') - V(s)$

$e(s) \leftarrow e(s) + 1$ ← Analogous to Every-Visit Monte-Carlo.

Backward View

For all $\bar{s}$: (Assuming constant α)

1. $V(\bar{s}) \leftarrow V(\bar{s}) + \alpha \delta e(\bar{s})$
2. $e(\bar{s}) \leftarrow \gamma \lambda e(\bar{s})$

If $e(s) \leftarrow 1$ Analogous First-Visit Monte-Carlo.

④ Control

1* $a^t = \pi_t(s^t)$

$$Q_{t+1}(s^t, a^t) \leftarrow Q_t(s^t, a^t) + \alpha_{t+1}\{r^t + \gamma Q_t(s^{t+1}, a^{t+1}) - Q_t(s^t, a^t)\}$$

$\pi_t = \pi \Rightarrow \lim_{t\to\infty} Q_t = Q^\pi$

$\pi_t = \varepsilon_t\text{-greedy}(Q_{t-1}) \Rightarrow \lim_{t\to\infty} Q_t = Q^*$ Sarsa. 1996. (Harder to analyze)

↓ on-policy

If you follow ~~on-p~~ fixed policy, you'll converge to Q^π

2* $$Q_{t+1}(s^t, a^t) \leftarrow Q_t(s^t, a^t) + \alpha_{t+1}\{r^t + \gamma \max_a Q_t(s^{t+1}, a) - Q_t(s^t, a^t)\}$$

$\pi_t = \pi \Rightarrow \lim_{t\to\infty} Q_t = Q^*$ assum π take all s, a with positive prob. ↓ Not 0.

$\pi_t = \varepsilon_t\text{-greedy}(Q_{t-1}) \Rightarrow \lim_{t\to\infty} Q_t = Q^*$

Q-Learning.

off-policy -

Q-Learning is the 1st Algorithm which has this guarentee.

Watkins, P.hD thesis.

Importance? Model-Free!

$\theta(\|S\|\|A\|)$ space.

3* Expected - Sarsa (Recent).

Only change.

$$Q_{t+1}(s^t, a^t) \leftarrow Q_t(s^t, a^t) + \alpha_{t+1}\left\{ r^t + \gamma \sum_a \pi_t(s^{t+1}, a)\, Q(s^{t+1}, a) - Q_t(s^t, a^t) \right\}$$

Some behaviours as Sarsa,
Less variance than Sarsa because of Expectation.

11th Oct

(0) TD(λ)

s // visited

$$e(s) \leftarrow \begin{cases} e(s)+1 & \text{// Accumulatory trace} \rightarrow \text{Every-visit MC} \\ 1 & \text{// Replacing trace} \rightarrow \text{First-visit MC} \end{cases}$$

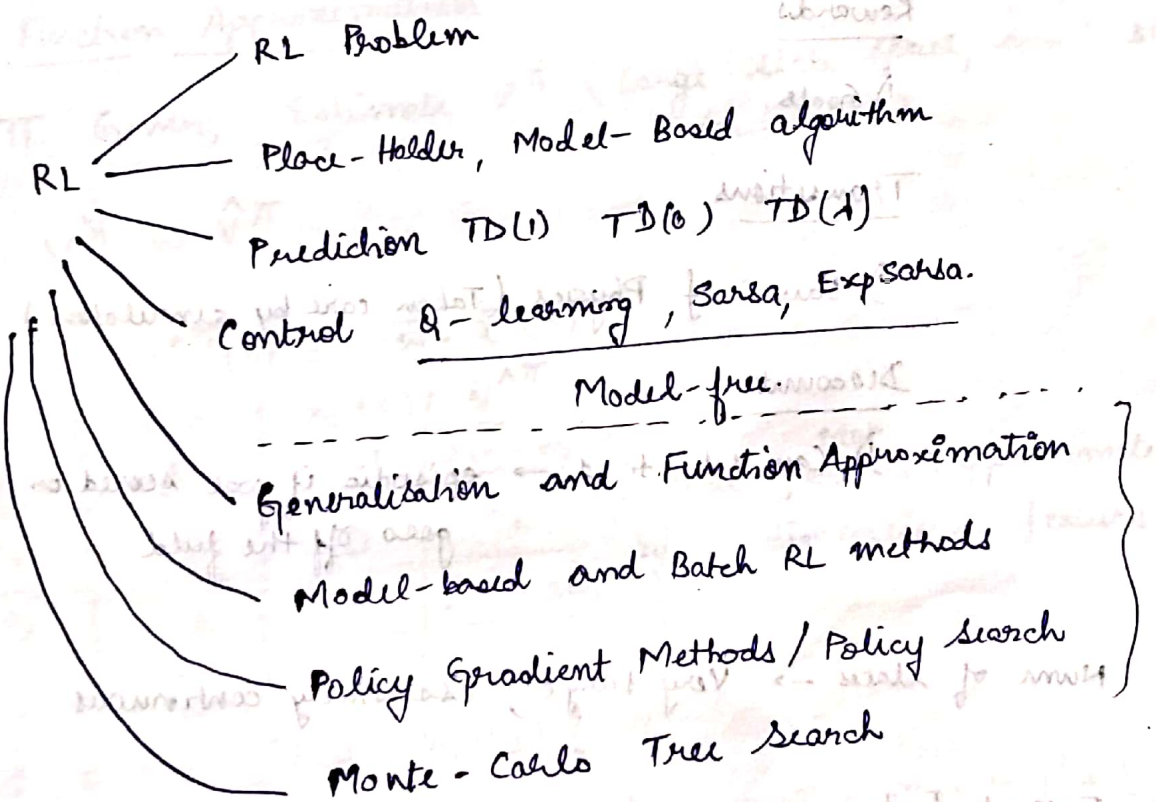
For $s' \in S$: // All states

$$V(s) \leftarrow V(s) + \alpha_t \delta e(s)$$

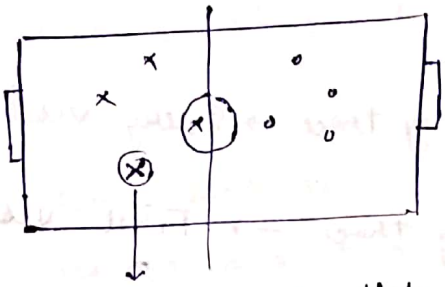
$$e(s) \leftarrow \gamma \lambda e(s)$$

valid for certain types of α.

(1) Overview



(2) Illustration : Soccer



Let us assume that one player is in our control.

State :

1) Positions of players and football
2) Velocities
3) Score ?
4) Stamina.

Rewards

1) Goals.

Transitions

1) Laws of Physics (Taken care by simulator).

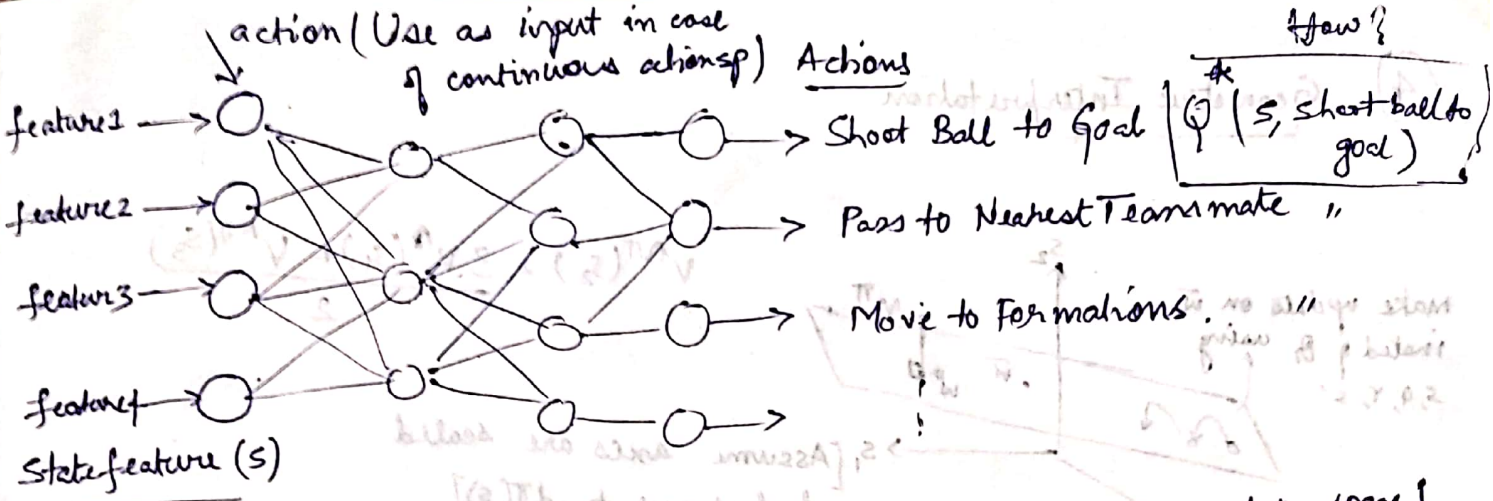
Discount

1) Can set to 1 → episodic if goal scored or ball goes off the field.

Num of states → Very large, essentially continuous.

⇒ Extract Features :

1) Which Team Attacking ?
2) Position of nearest teammates ?
3) Distance to goal ?
4) Angle to goal ?



Output ≠ Q^* exactly? Why? → 1) Not looking at entire state space!
Only looking at extracted features.

Optimality ✗
Convergence?

(3) Linear Function Approximation

π Given, Estimate v^π (Large state space, can't store in table).

$$v^\pi \approx \hat{v}^\pi$$

$$v^{\wedge\pi}(s) = \bar{w} \cdot \bar{x}(s)$$

v^π	s	$x_1(s)$	$x_2(s)$	$v^{\wedge\pi}$
7	s_1	2	-1	$2w_1 - w_2$
2	s_2	4	0	w_1
-4	s_3	2	3	$2w_1 + 3w_2$

(3-state MDP ~~features~~ v_π approximated by 2-dimensional features.)

Which is the "best" (w_1, w_2)?

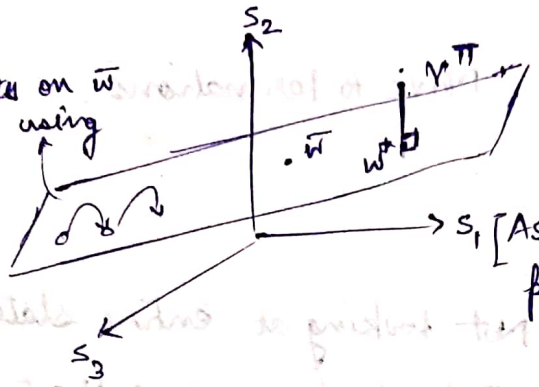
$$(w_1^*, w_2^*) = \underset{(w_1, w_2)}{\operatorname{argmin}} \left([7 - (2w_1 - w_2)]^2 + [2 - w_1]^2 + [-4 - 2w_1 - 3w_2]^2 \right)$$

$$\bar{w}^* = \underset{(\bar{w})}{\operatorname{argmin}} \, MSVE(\bar{w})$$

$$MSVE(\bar{w}) = \frac{1}{2} \sum_{s \in S} d^\pi(s) \left\{ v^\pi(s) - \frac{\hat{v}^\pi_{\bar{w}}(s)}{\bar{w} \cdot \bar{x}(s)} \right\}^2$$

(4) Geometric Interpretation

$$V^{\hat{\pi}}(s_2) = \frac{3V^{\hat{\pi}}(s_1) + V^{\hat{\pi}}(s_3)}{2}$$



Make updates on $\bar{w}$ instead of $\hat{q}$ using s, a, r, s'

[Assume axes are scaled proportional to $d\pi(s)$].

Oct - 14th

(1) Stochastic Gradient Descent

$\bar{w}_0 \leftarrow$ Initialisation

Follow π

For $t = 0, 1, 2, \ldots$

$$\bar{w}_{t+1} \leftarrow w_t - \alpha_{t+1} \nabla_{\bar{w}} \left\{ \frac{\left[\left(v^{\pi}(s^t) - \hat{v}^{\pi}\left(\bar{x}(s^t), \bar{w}_t \right) \right] \right]^2}{2} \right\}$$

$$\leftarrow w_t + \alpha_{t+1} \left\{ v^{\pi}(s^t) - \hat{v}^{\pi}\left(\bar{x}(s^t), \bar{w}_t \right) \right\} \times \nabla_{\bar{w}} \hat{v}\left(\bar{x}(s^t), \bar{w}_t \right)$$

If linear,

$$\leftarrow w_t + \alpha_{t+1} \left(v^{\pi}(s^t) - \bar{x}(s^t) \cdot \bar{w}_t \right) \cdot \bar{x}(s_t)$$

- Option - 1 : Use $r^t + \gamma r^{t+1} + \gamma^2 r^{t+2} + \cdots$

  True Gradient Descent

  Linear TD(1)

Option 2 : Use $r^t + \gamma \hat{v}^{\pi}\left(\bar{x}(s^{t+1}), \bar{w}_t \right)$

Linear TD(0), semi Gradient Descent.

Option - 3 : Use λ - returns

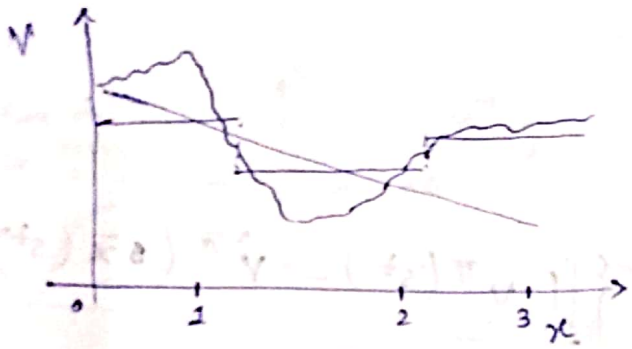
Linear TD(λ).

Eligibility Trace : One parameter for each element of $\bar{w}$

For TD(λ), $MSVE(\bar{w}_\infty) \leq \frac{1 - \gamma\lambda}{1 - \gamma} MSVE(\bar{w}^*)$

TD(1) → Convergence not proved, but of divergence also not proved

(2) Tile Coding



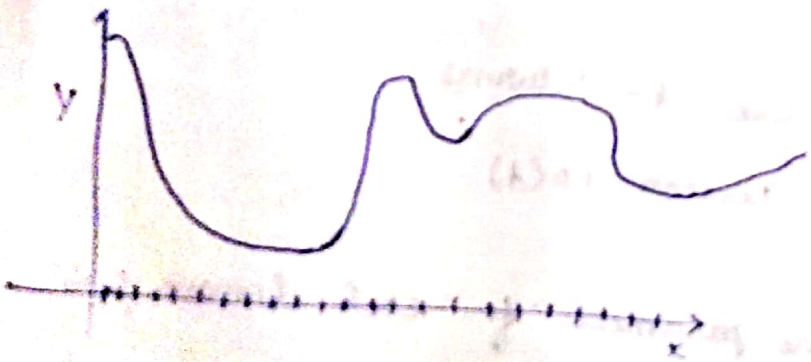
$$\hat{v} = mx + c$$

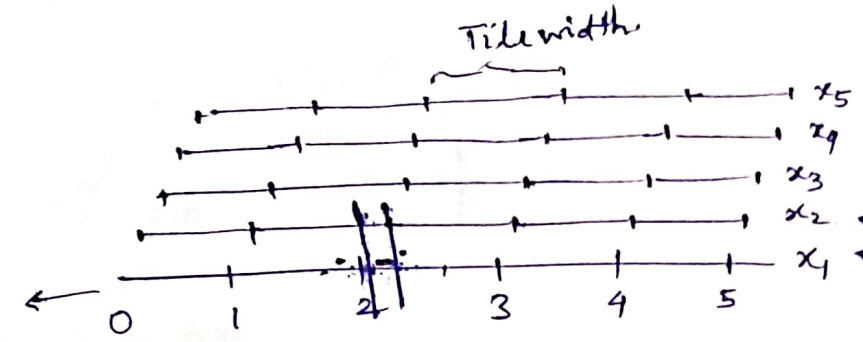
$$x_1 = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & o/w \end{cases}$$

$$x_2 = \begin{cases} 1 & 1 \leq x < 2 \\ 0 & o/w \end{cases}$$

$$x_3 = \begin{cases} 1 & 2 \leq x \\ 0 & o/w \end{cases}$$

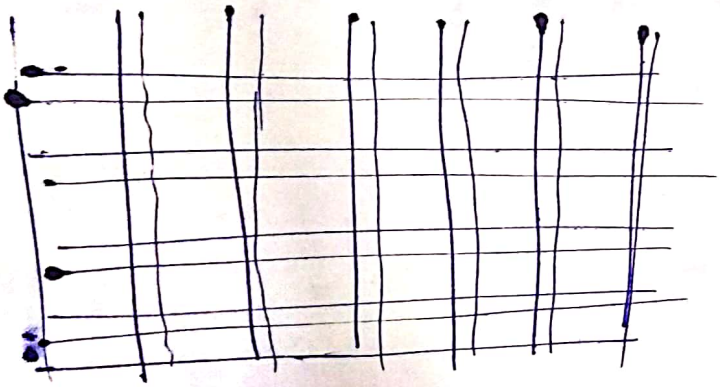
$$\hat{v}(x) = w_1 x_1 + w_2 x_2 + w_3 x_3$$





Til.

Resolution = $\frac{\text{Tile width}}{\text{Number of Tilings}}$ ✓



2-d tiling

→ π V^{π} TD(1) $\bar{w}$

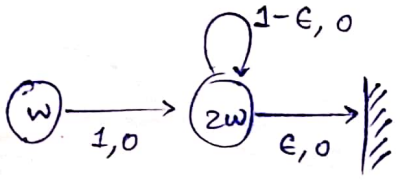
π^* // Control

s a r s' a' (o)

$$\bar{w}_{t+1}^{a} \leftarrow \bar{w}_t^{a} + \alpha_{t+1}\left\{ r^t + \bar{w}_t^{a'} \cdot \bar{x}(s^t) - \bar{w}_t^{a} \cdot \bar{x}(s) \right\} x$$

(1) Tsitsklis and Vankoy's Counter example

Oct - 18



$w_0 \to w_1 \to w_2 \cdots\cdots$

For each state, a "better" estimate of the value is $\mathbb{E}_\pi [r + \gamma \hat{V}^\pi (\bar{x}(s), w)]$

Set $w_{k+1} \leftarrow \underset{w \in \mathbb{R}}{\operatorname{argmin}} \sum_s \left\{ \mathbb{E}_\pi [r + \gamma \hat{V}^\pi (\bar{x}, s_{next}, w_k)] - \hat{V}^\pi (\bar{x}(s), w_0) \right\}^2$

Function Approx.

Boot Strapping

Off-policy update

↓

Root-Cause

Not visiting states according to $d_\pi(s)$

$$= \underset{w \in \mathbb{R}}{\operatorname{argmin}} \left\{ (2\gamma w_k - w_0)^2 + [(1-\epsilon) 2\gamma w_k + \epsilon \times 0 - 2 w_0]^2 \right\}$$

$$= \underset{w \in \mathbb{R}}{\operatorname{argmin}} \left\{ w^2 + 4\gamma^2 w_k^2 - 4\gamma w w_k + 4 w_0^2 + (1-\epsilon)^2 4\gamma^2 w_k^2 - 8\gamma w w_k (1-\epsilon) \right\}$$

$$= \underset{w \in \mathbb{R}}{\operatorname{argmin}} \left\{ 5w^2 + 4\gamma^2 w_k^2 (1 + (1-\epsilon)^2) + - 4\gamma w w_k \{ 1 + 2 - 2\epsilon \} \right\}$$

$$= \underset{w \in \mathbb{R}}{\operatorname{argmin}} \left\{ 5w^2 - 4\gamma w w_k (3 - 2\epsilon) \right\} + \text{const.}$$

$$\Rightarrow 10w = 4\gamma w_k (3 - 2\epsilon)$$

$$\Rightarrow w = \frac{2}{5} \gamma w_k (3 - 2\epsilon)$$

$$\Rightarrow \boxed{w_{k+1} = \frac{2}{5} \gamma w_k (3 - 2\epsilon)}$$

May Diverge

(2) Approximate Policy Improvement

$Q^\pi / Q^{*\pi}$	a_1	a_2
s_1	3/2.5	2/3
s_2	1/2	6/5.5

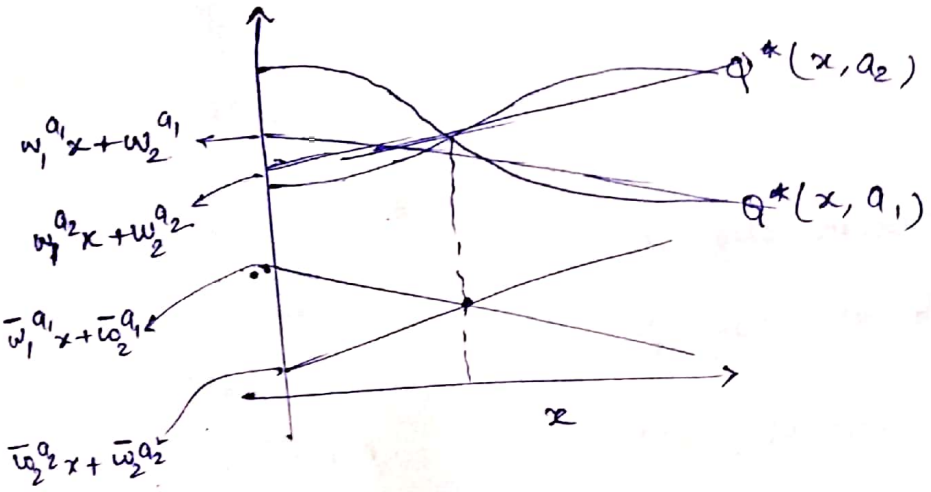
Greedy (Q^π) = (a_1, a_2)

Greedy $(Q^{*\pi})$ = (a_2, a_2)

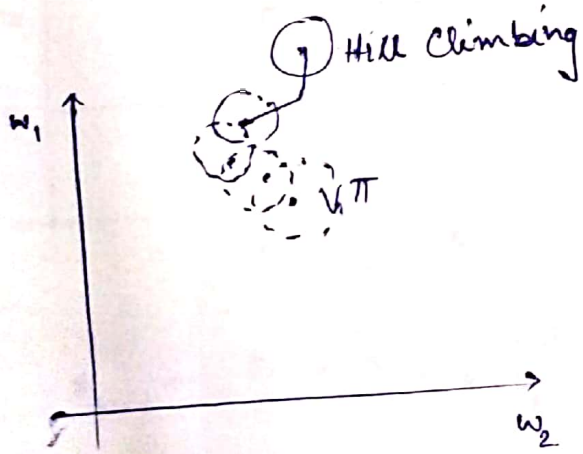
(3) Summary of results

Method	Tabular	Linear FA	Non-Linear FA
TD(0)	C, exact	C, bounded error.	NK
TD(λ)	C, exact	C, bounded error	NK
TD(1)	C, exact	C, best	C, Local Optimum
sarsa(0)	C, opt	Chattering.	NK
sarsa(λ)	NK	NK	NK
sarsa(1)	NK	NK	NK
Q-learning(0)	C, opt	NK	NK

(4) Policy search methods



$$q = w_1 x + w_2$$



: Evolutionary Algorithm

: Particle swarm optimisation

: Grid-search

① Introduction

π_θ

Policy

If $\theta < 6$:

Take a_1

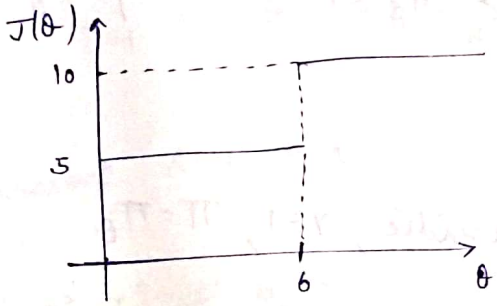
Else

Take a_2.

a_1 gets reward of 5

a_2 gets reward of 10

$J(\theta)$: reward achieved by π_θ



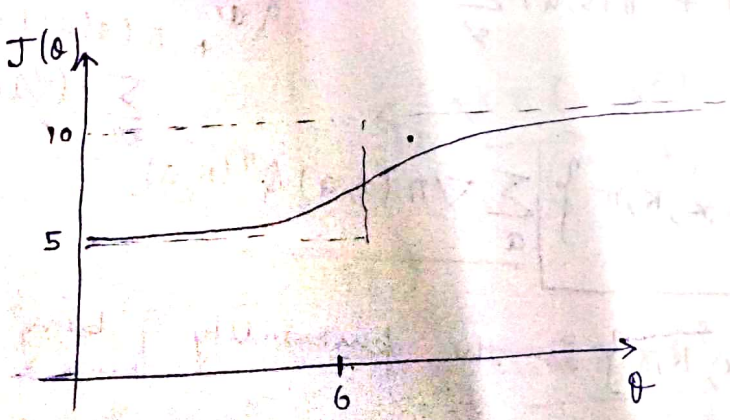
$$\theta_{t+1} \leftarrow \theta_t + \alpha_{t+1} \nabla_{\theta=\theta_t} J(\theta)$$

$$\pi(s, a_1) = \frac{1}{1 + e^{-(6-\theta)}}$$

$$\pi(s, a_2) = 1 - \pi(s, a_1)$$

} probabilities of taking action a_1, a_2.

$$J(\theta) = \pi(s, a_1) \times 5 + \pi(s, a_2) \times 10$$



Start state s_0

$$J(\theta) = V^{\pi_\theta}(s_0)$$

Common choice of π_θ

$$\pi_\theta(s,a) = \frac{e^{\bar{\theta}\cdot\bar{x}(s,a)}}{\sum_b e^{\bar{\theta}\cdot\bar{x}(s,b)}}$$

$$\nabla_\theta \pi_\theta(s,a) = \left\{ \bar{x}(s,a) - \sum_b \pi_{\bar{\theta}}(s,b)\, \bar{x}(s,b) \right\} \pi_{\bar{\theta}}(s,a)$$

(2) Policy Gradient Theorem

Assumptions : ∇ w.r.t $\hat{\theta}$, Episodic, $\gamma = 1$, $\pi = \pi_\theta$

$$\nabla V^\pi(s) = \nabla \left[\sum_a \pi(s,a)\, Q^\pi(s,a) \right]$$

$$= \sum_a \Big[\nabla \pi(s,a)\, Q^\pi(s,a) + \pi(s,a)\, \nabla \Big\{ \sum_{s'} T(s,a,s') \left(R(s,a,s') + V^\pi(s') \right) \Big\} \Big]$$

$$= \sum_a \Big[\nabla \pi(s,a)\, Q^\pi(s,a) + \pi(s,a) \Big(\sum_{a'} T(s,a,s')\, \nabla V^\pi(s') \Big) \Big]$$

$$= \sum_a \Big[\nabla \pi(s,a)\, Q^\pi(s,a) + \pi(s,a) \sum_{s'} T(s,a,s') \Big(\sum_{a'} \nabla \pi(s',a')\, Q^\pi(s',a') + \pi(s',a') \times \sum_{s''} T(s',a',s'')\, \nabla V^\pi(s'') \Big) \Big]$$

$$= \sum_{x \in S} \boxed{\sum_{K=0}^{\infty} \mathbb{P}\{s \to x, K, \pi\}} \sum_a \nabla \pi(x,a)\, Q^\pi(x,a),$$

where $\mathbb{P}\{s \to x, K, \pi\}$ is the probability of being in x after K steps following π starting at s.

$$\nabla V^{\pi}(s) \propto \sum_{x} \mu^{\pi,s}(x) \sum_{a} \nabla \pi(x,a)\, Q^{\pi}(x,a)$$

where $\mu^{\pi,s}(x)$ is the stationary of being in x starting from s and following π.

$$\nabla J(s_0) \propto \sum_{s \in S} \mu^{\pi,s_0}(s) \sum_{a \in A} \nabla \pi(s,a)\, Q^{\pi}(s,a)$$

Proportionality constant is expected length of the episode.

Policy Gradient Theorem

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t \nabla J(\theta)$$

(3) Policy Gradient Algorithm

π_θ : $s_0 = s^0\ a^0\ r^0\ s^1\ a^1\ r^1 \ldots\ s^t\ a^t\ r^t \ldots$

$$\nabla J(\theta) \propto \sum_{s} \mu^{\pi,s_0}(s) \sum_{a} \nabla \pi(s,a)\, Q^{\pi}(s,a)$$

$$= E_{\pi}\left[\sum_{a} \nabla \pi(s^t,a)\, Q^{\pi}(s^t,a) \right]$$

$$= E_{\pi}\left[\sum_{a} \pi(s^t,a) \left\{ Q^{\pi}(s^t,a) \frac{\nabla \pi(s^t,a)}{\pi(s^t,a)} \right\} \right]$$

$$= E_{\pi}\left[Q^{\pi}(s^t,a^t) \cdot \frac{\nabla \pi(s^t,a^t)}{\pi(s^t,a^t)} \right]$$

$$= E_{\pi}\left[G_t \frac{\nabla \pi(s^t,a^t)}{\pi(s^t,a^t)} \right]$$

REINFORCE (Williams, '92)

$$\theta_{t+1} \leftarrow \theta_t + \alpha_{t+1} G_t \frac{\nabla \pi(s^t,a^t)}{\pi(s^t,a^t)}$$

$$= \theta_t + \alpha_{t+1} G_t \nabla \log \pi(s^t,a^t)$$

(0) $\nabla J(\theta) = \nabla V^{\pi}(s) = \sum_{x \in S} \sum_{k=0}^{\infty} \mathbb{P}_{\pi}\{s_0 \to x, k, \pi\} \underbrace{\sum_a \nabla \pi(x,a) Q^{\pi}(x,a)}_{f(x)}$

Follow π, starting at s_0

$s_0 = s^0\ a^0\ r^0\ s^1\ a^1\ r^1 - \cdots - s^{T-1}$

Return Estimate $= f(s^0) + f(s^1) + \cdots$

$\mathbb{E}[\text{Estimate}] = \sum_{k=0}^{\infty} \mathbb{E}[f(s^k)]$

Loop for ever :

Generate episode $e = s^0\ a^0\ r^0 \cdots r^{T-1}\ s^T$, following π_θ

~~Loop~~ Loop for $t = 0, 1, \cdots T-1$

$G \leftarrow \sum_{k=t}^{T-1} \gamma^{k-t} r^k$

$\theta_{new} \leftarrow \theta_{new} + \alpha_e\ \gamma^t\ G\ \nabla \ln \pi_\theta(s^t, a^t)$

↓

Not θ_{new}

(as in SB)

(1) Baseline

Q^{π}

	a_1	a_2	a_3	$V^{\pi} = \sum_a \pi(s,a) Q^{\pi}(s,a)$
s_1	105	79	100	90
s_2	10	6	13	12
s_3	-50	-60	-50	-55

$\nabla \pi(s^t, a^t)\ Q^{\pi}(s^t, a^t)$

↳ Might be high. → cause high variance

(2) Ac

Critic

Actor

Let $b: S \to \mathbb{R}$ be any function of state

$$\nabla J(\theta) \propto \sum_s \mu^{\pi, s_0}(s) \sum_a \nabla \pi(s,a) Q^{\pi}(s,a)$$

$$= \sum_s \mu^{\pi, s_0}(s) \sum_a \nabla \pi(s,a) \{ Q^{\pi}(s,a) - b(s) \}$$

$$\sum_s \mu^{\pi, s_0}(s) \sum_a \nabla \pi(s,a)\, b(s)$$

$$= \sum_s \mu^{\pi, s_0}(s)\, b(s)\, \nabla \Big(\underbrace{\sum_a \pi(s,a)}_{1} \Big) = 0.$$

If you don't know something, what do you do in reinforcement learning? Stick in an estimator.

REINFORCE with baseline

$$\theta_{new} \leftarrow \theta_{new} + \alpha \sum_{t=0}^{\infty} \gamma^t \left(G_t - b(s^t) \right) \nabla \ln \pi_\theta (s^t, a^t)$$

$\hat{V}^{\pi_\theta}(s_t)$, which is itself estimated (empirically).

(2) Actor - Critic Methods

Critic Update w such that $\hat{V}_w$ is a good estimate of the value function of the current policy.

Actor $\theta_{new} \leftarrow \theta_{new} + \alpha \sum_{t=0}^{\infty} \gamma^t \left(r^t + \gamma \hat{V}_w(s^{t+1}) - \hat{V}_w(s^t) \right) \times \nabla \ln \pi_\theta (s^t, a^t)$.

$s\ a\ r\ s'\ a'\ r$ ⟶ $s\ a\ r$ Length of trajectory = Sample complexity

$$Q(s,a) \leftarrow Q(s,a) + \alpha \{ r + \gamma \max_a Q(s',a) - Q(s,a) \}$$

↑ Computational Complexity.

③ Batch RL

$Q \leftarrow 0$

Repeat for ever:

$\pi \leftarrow \epsilon$-greedy (Q)

Follow π for N- episodes, gather data.

$$D = (s_i, a_i, r_i, s_{i+1})^L$$

$Q \leftarrow$ Batch Update (D, Q) // Optional to use Q.

E.g.

Batch Update Experience Replay (D, Q)

Repeat M times

Pick $i \in L$ Uniformly at random

$$Q(s_i, a_i) \leftarrow Q(s_i, a_i) + \alpha \{ r + \gamma \max_a Q(s^{i+1}, a) - Q(s^i, a) \}$$

Return Q.

Batch Update Fitted Q Iteration (D)

$Q_0 \leftarrow 0, \quad i \leftarrow 0$

Assume predictor $\leftarrow$ Supervised Learning $\left((x_j, y_j)^L\right)$

Repeat H times:

for $j = 1$ to L

$\bar{x}_j$ = featureVector (s_j, a_j)

$y_j = r_j + \gamma \max_{a'} Q_i(s_{j+1}, a')$

$Q_{i+1} \leftarrow$ Supervised Learning (x_j, y_j)

$i \leftarrow i+1$.

Return Q.

Nov-1

① Models S, A, T, R, γ

$\hat{T}, \hat{R} \rightarrow$ Model

Distributional Models : $T(s,a,s')$ is known/given → D

↓

More powerful, Not always available.

Sampling Model : $s' \sim T(s,a)$

↓

sampling operator → S

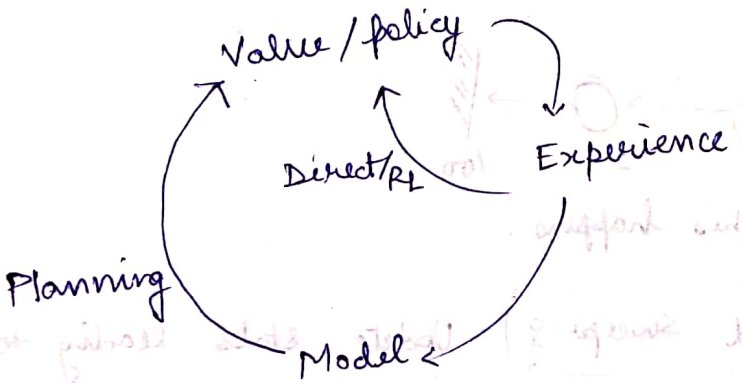
Systems where set of states too large, or complex dynamics to model the probability exactly.

Board games, which involve a roll of dice. Distribution of states very difficult as compared to just sampling a new state.

② Concept / Idea.

Experience → Value Function → Policy

Model → simulated experience → value function → policy.



Model ⇒

1) less experienced
2) More computation
3) Useful in a changing world.

Thinking : Fast and Slow

Dyna-Q

Initialise $Q(\cdot,\cdot)$, $Model(\cdot,\cdot)$

Loop for ever

Direct RL

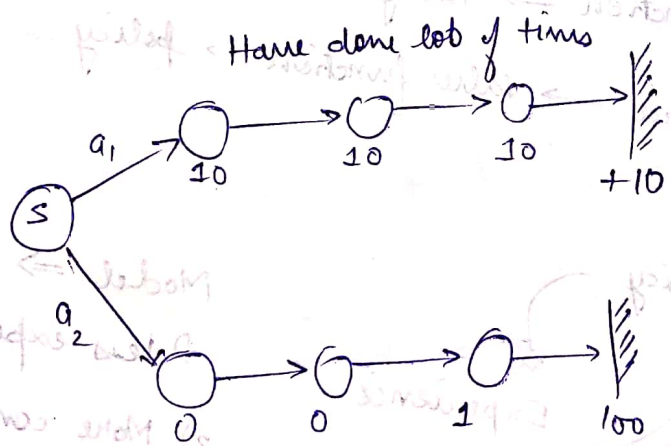
- $s \leftarrow$ current state
- $a \leftarrow \epsilon$-greedy(a,s)
- Take action a. Get next state s', reward r.
- $Q(s,a) \leftarrow Q(s,a) + \alpha\{r + \gamma \max_{a'} Q(s',a') - Q(s,a)\}$

→ Update Model (Model, s, a, r, s')

Simulation

Loop 'n' times:

- $s \leftarrow$ random previously observed state
- $a \leftarrow$ random previously taken action from s.
- $r, s' \leftarrow$ Model (s,a)
- $Q(s,a) \leftarrow Q(s,a) + \alpha\{r + \gamma \max_{a'} Q(s,a') - Q(s,a)\}$

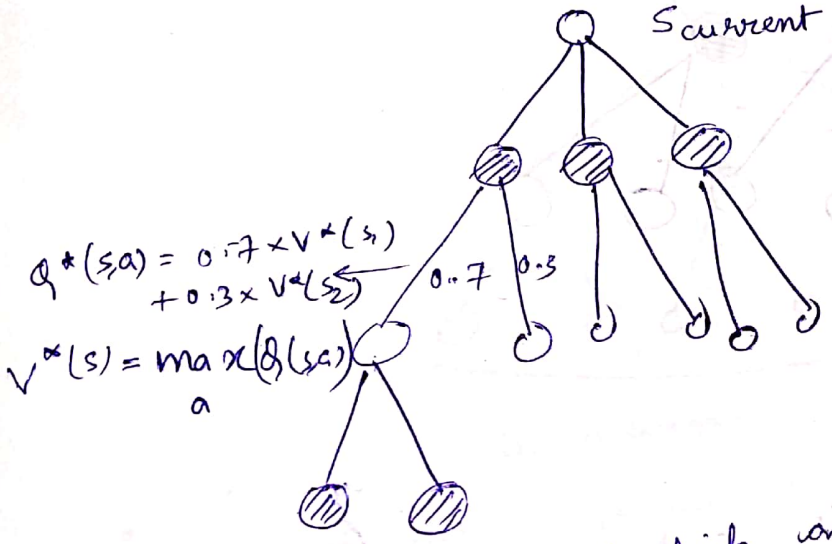


Then this happens.

Prioritised Sweeping Update stats leading to states whose values have changed the most.

④ Action Selection at run time

Forward Simulation / Tree Search

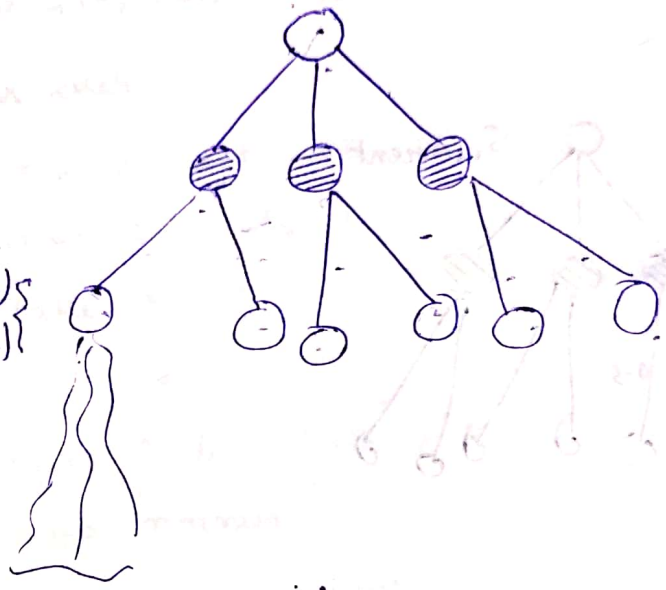


Suppose we want to pick an ϵ-optimal action we need a tree with depth $f\left(\frac{1}{\epsilon}, \frac{1}{1-\gamma}\right)$

b^d size of tree.

MCTS

(1) Roll-out Algorithms



Ideal to use $V^*(s)$
Practical to use $\sim V^{\bar{\pi}}(s)$

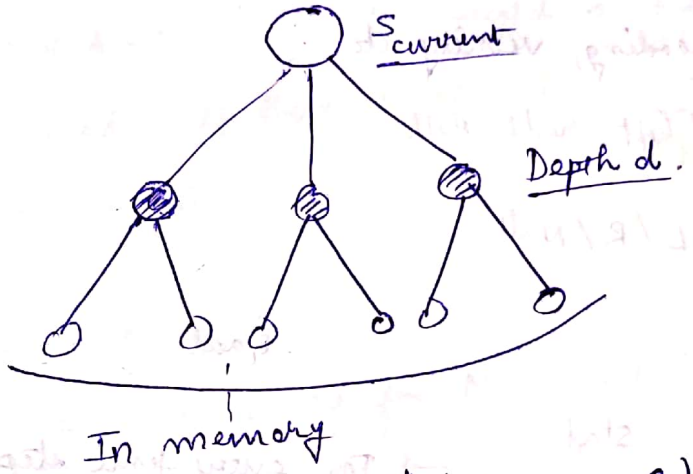
Approximate with $V^{\pi}(s)$, where π is a ~~good policy~~ random policy.

In practice, use rollout after expansion to a reasonable depth. Hope that a randomly good policy (π) is also able to distinguish b/w actions.

Problem: expanding lower Q values same no. of times as higher Q values.

② UCT (MCTS)

Happening at run time



- Aim is to find $\arg\max_a Q^*(s_{current}, a)$.
- ~~Store~~ Store (s,a) pairs reachable in d steps in table/memory.

For each pair maintain:

- $Q(s,a)$, the empirical average of cumulative reward based on max, expectation, rollouts.
- $UCB(s,a) = Q(s,a) + c_p \sqrt{\frac{\ln(t)}{u(s,a)}}$ → No. of times you have called your simulator.

  ↓ No. of visits to (s,a).

- When in state s, sample action $a = \arg\max_a UCB(s,a)$.
- When at leaf, follow roll-out policy.
- Update values along simulate path.

Non-stationary handled by large values of c_p.

For attention on "high-value" states.

③ Reward Shaping

Bicycle

State: Angle, heading, velocity etc.

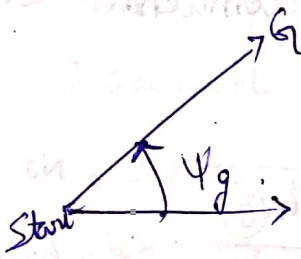
Action: Turn, Left, right, Null

Hand L/R/Null.

Goal

Start → For every time step.

$\frac{1}{R}$ at every time step.



$a - b\,\gamma_g^{\alpha}(s^t)$.

$M = SATR\gamma$
$M' = SATR'\gamma$
} Under what conditions will an ~~oppo~~ optimal policy of M also be optimal for M'?

Definition: $F: S \times A \times S \to R$ is called a potential-based functional if there exists $\phi: S \to R$ such that $\forall s \in S, a \in A$, $F(s, a, s') = \gamma \phi(s') - \phi(s)$

Theorem (Ng, M ___, Russel, 99).

If π^* is an optimal policy for $M = (S, A, T, R, \gamma)$, then for every potential based shaping f^N F, π^* is also an optimal policy for

$$M_F = (S \; A \; T \; R+F \; \gamma)$$

Necessity.

If F is not a potential-based shaping f^N, then there exists $N = SATR\gamma$, π^* such π^* is an optimal policy for M, π^* is not an optimal policy f^N.

$$M = (S \; A \; T \; R+F \; \gamma)$$