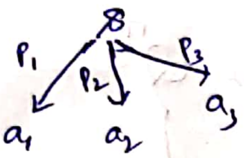
# Actor-Critic to Policy iteration

$$\sum_{\substack{\text{minibatch}\\ \text{of } (s,a)}} \nabla_\theta \log \pi_\theta(a/s)\, A^\pi(a,s)$$

Let's consider 'one' particular state & see what Actor-critic does!

Say, 3 actions possible & $\underline{p_i = \pi_\theta(a_i/s)}$

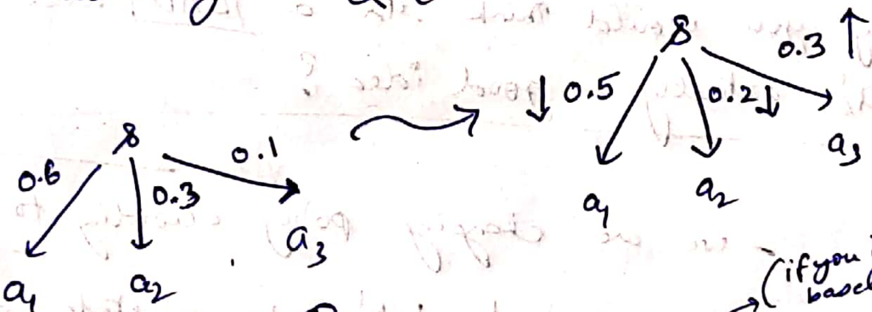



Since $A^\pi(a_i,s) = Q^\pi(a_i,s) - V^\pi(s)$

If we remove the baseline, (since it's same here as we consider only one state)

Then, $p_i$ is increased (or) decreased according to $Q^\pi(a_i,s)$

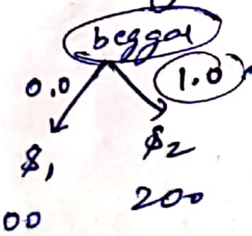
That is,




$Q_1$ $Q_2$ $Q_3$

Say 100, 200, $\boxed{300}$ → then "relatively" (if you include baseline (or) otherwise), the one of the highest Q's probability is increased, & lowest Q's probability decreased

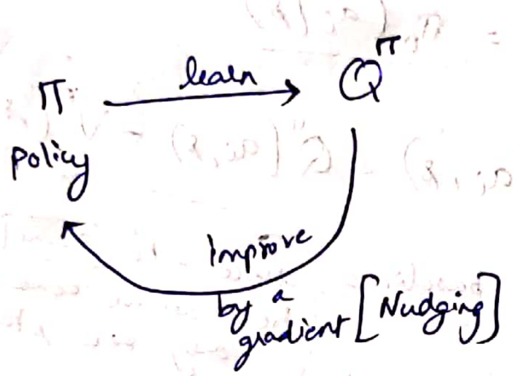
If you consider the same as a "beggar" scenario choosing between streets!



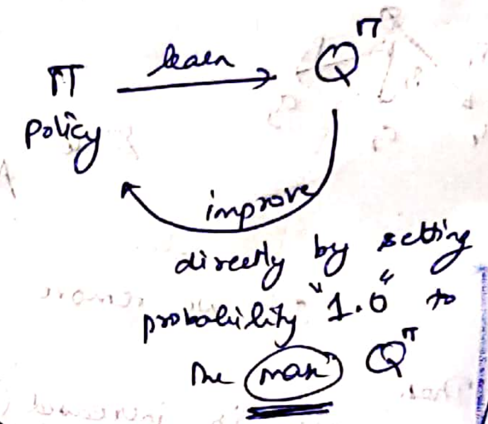

→ we simply keep choosing the street of the highest expected reward! $\boxed{\text{AND}}$ not choose the other!!

So, if that's the case, then ~~it's~~ instead of $\boxed{\text{nudging}}$ the probabilities only a little bit, we can completely change it $\boxed{\text{like}}$ in policy iteration! which is the only difference between the algorithms!

Actor-Critic



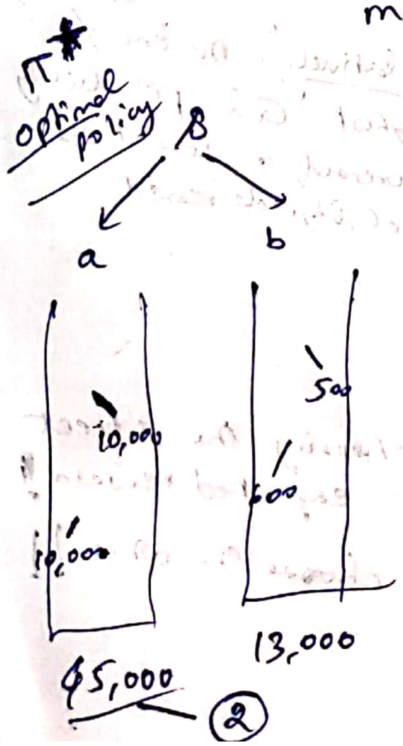

Policy iteration

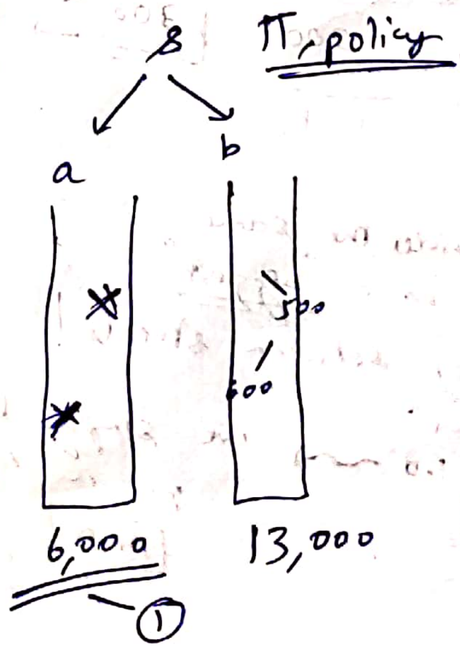



* Why you would think it's a $\boxed{\text{BAD}}$ idea when it's actually a good idea?

- we are changing policy $\boxed{\text{according to } Q^\pi \text{, not } Q^*}$

meaning!, Let's take a state, s, & 2 actions



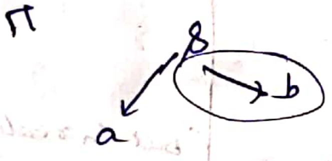
So, under optimal policy all, actions under all states "are" optimal! So, $Q^*(a,s) = 45,000$

But since $\pi$, is some policy, it has been configured of actions sub-optimal under 'a' but the same as optimal policy under 'b'

So, $Q^\pi(a,s) = 6,000$

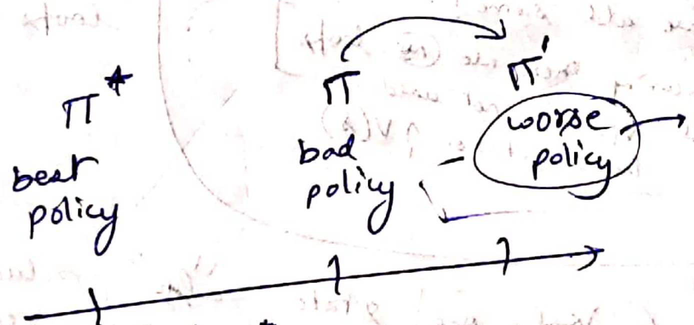
So, Because $\pi$ is a $\boxed{\text{BAD}}$ policy, it's Q-value estimate from 'a' looks bad!

As, a result, update to $\pi$ [only in one state] under policy iteration changed the policy to $\pi'$




which $\boxed{\text{prefers}}$ 'b' instead of a!

So,




since it's the same $\pi$, of the policy changed for 'only' one state, where only the sub-optimal was selected!

So, if you "rate" the policies by the $\boxed{\text{number}}$ of sub-optimal choices the policy makes,

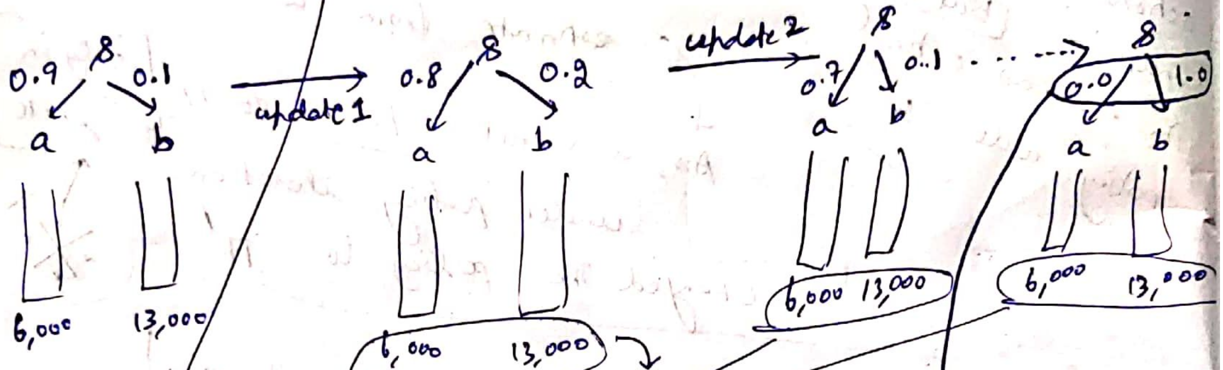
Then $\pi'$ does not look like an improvement to $\pi$ but "like" a deprovement!

Disclaimer: All this reasoning is assuming, we are applying policy iteration update to $\boxed{\text{only one}}$ state, not every one

But policy iteration is $\boxed{\text{proved}}$ to improve policy, so, what is happening?

Proof via Actor-Critic

Say, you apply actor-critic update also to $\boxed{\text{only one}}$ state, then,




remains the "same",
Since in the new policy,
only s is changed & others ~~are all same!~~

[Assuming, there are no loops & $V(s)$ is not used in the computation of $V(s)$]

(but in reality loops exist)

So, Actor-Critic @ $\boxed{\text{only one}}$ state "also" produces same result

So, how is our intuition that it won't improve is wrong?

This is because we are evaluating the policies $\pi^*, \pi, \pi'$ on the metrics of $\boxed{\text{number}}$ of optimal decisions?

in which case, $\pi'$ ranks $\boxed{\text{LAST}}$!

But in terms of expected reward, as our metric, $\pi'$ is not LAST!

This is because, not all sub-optimal choices are EQUAL

meaning - not all mistakes cost EQUAL

You can make a 100 small mistakes, still receive 1000 points

but by making only a few mistakes say 5, you can receive 100 points

Example → death ↳ 1 mistake, but costliest !

↳ but skipping lunch oneday → small cost !!
1 mistake but

So, in metrics of expected reward :

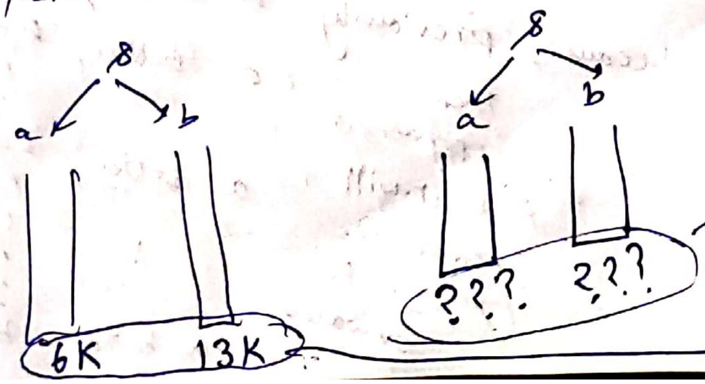
$\pi^*$ best policy | $\pi'$ improved policy | $\pi$ BAD policy

This is the rank !

★★ why you would think it's a bad idea when it's actually a good idea ?

Say, you are doing policy iteration on all states !

Then → since the policy is changed everywhere




we don't know what the estimates will be in this new policy

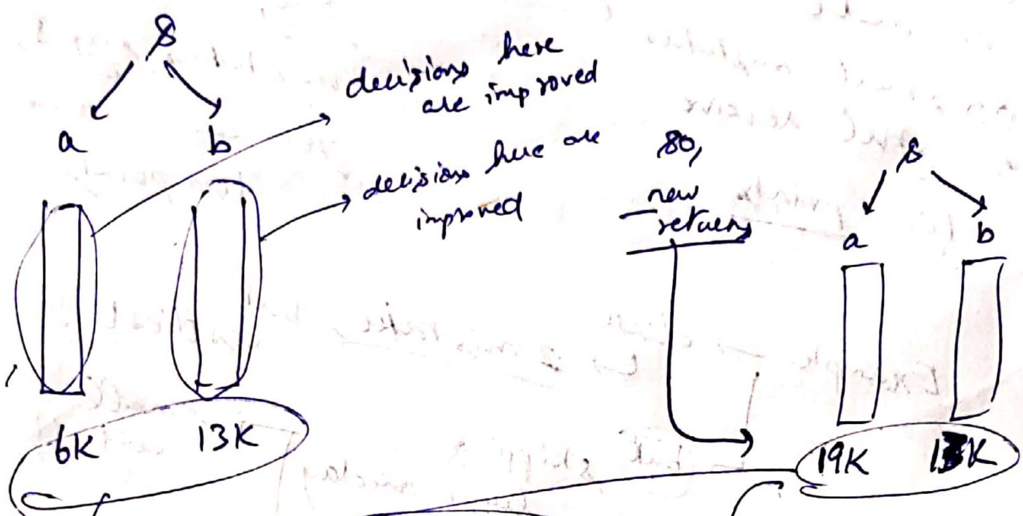
But we update s using the invalid old policy !

Not only are we using incorrect values to update (s) !

↳ there will ~~also~~ also be $\boxed{\text{OSCILLATION}}$

↳ Since we have proved that the policy improves,

So,




Obviously this would improve, in fact it will improve till 45K !

So, now a seems better, previously b seemed better,

So, the policy keeps on oscillating !

~~But point is oscillates only once~~ !

↳ But point is it oscillates, only a $\boxed{\text{Fixed}}$ number of times!

— the oscillations are bounded and $\boxed{\text{NOT}}$ infinite!

But why does the oscillation $\boxed{\text{STOP}}$ ??

This is because, previously a worse, b better

then improvement a better, b worse

And it will remain the same here onwards a better, b worse

a better b worse

In other words,

The way, we calculate $\underline{Q^{\pi}}$ is using

$\boxed{Q^{\pi}_{t} \text{ \& discounting}}$

where $Q^{\pi}_{t}$ is the Q-value under $\pi$, if we had 't' timesteps to live

And discounting makes

$$\boxed{Q^{\pi}_{n-1} = Q^{\pi}_{n} = Q^{\pi}_{n+1}}$$

when $n$ is large!

where in policy iteration, first $Q^{\pi}_{1}$ is improved, rest all $Q^{\pi}_{2 \dots n}$ taking garbage values!

if $Q^{\pi}_{1}$ settled then $Q^{\pi}_{2}$ settles

$Q^{\pi}_{3 \dots n}$ oscillates!

So, slowly

$Q^{\pi}_{1 \dots k}$ are settled & $Q^{\pi}_{k+1 \dots n}$ oscillate

until $Q^{\pi}_{n}$ settles

which you could do directly

Evaluate only $Q^{\pi}_{0}$, then $Q^{\pi}_{1}$, then $Q^{\pi}_{2}$ .... until you get

$\underline{Q^{\pi}_{n}}$

which is called as Q-learning!

Hence the transition from

Actor-Critic to Policy Iteration to Q-learning (ends)

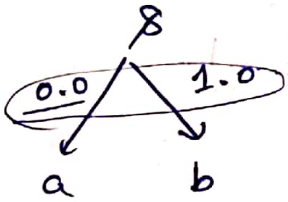
But important thing to note

ON-POLICY

1-step OFF-POLICY, n-1 step ON-POLICY

Completely OFF-POLICY

The ~~reason~~ reason is, definition of Q




A policy can be anything, even '0' probability for a,

But to calculate $Q(a, s)$, we still need to sample 'a' action, 1,000 times to find expected return!

In Actor-Critic, there is always some probability for all actions!

So, all $Q(a_i, s_j)$ are calculated!

In policy iteration, it is deterministic policy!

So, for 'one' step, i.e for only the 1st step $(a, s)$, we need to sample 1,000 times, but off-policy

Then on, we need to only sample according to policy to calculate $Q^{\pi}(a, s)$

Last of all

$Q^*(a,s)$ is completely off-policy

To summarize,

Actor-critic's policy has off-policy built in, by having "some" probability for all actions in the policy, so, purely on-policy!

Policy iteration evaluates $Q^{\pi}(a,s)$, where,

a is off-policy

Rest all, on-policy!

$Q^*(a,s)$ is strictly off-policy!

off-policy

$$Q^*(a,s) = \mathop{\mathbb{E}}_{s'} \max_{a'} Q^*(a',s')$$

max takes care of using the optimal policy $\boxed{\text{always}}$!

DONE