Lecture-21st

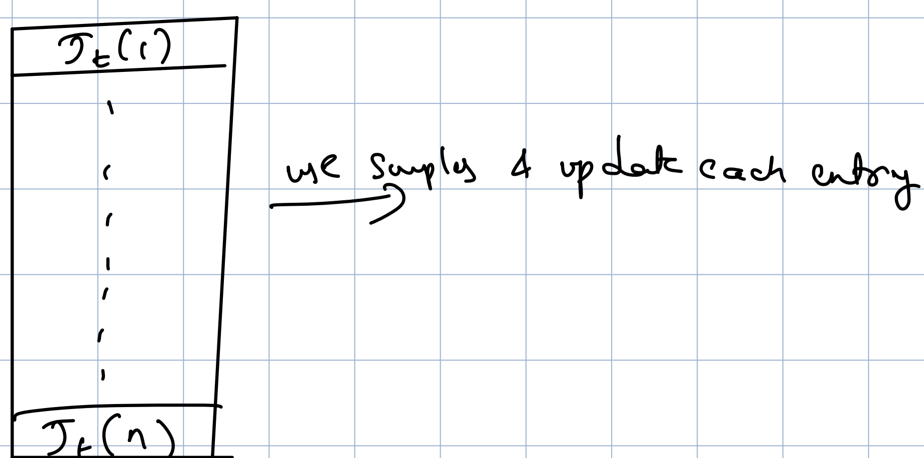
TD-learning, Q-learning with full state representations

ref: Chapter 5 of NDP book

Full state-representation?

Suppose we want to estimate $J_\pi(i)$, for a given policy π, & for any $i \in \mathcal{X}$.

A full-state algorithm aka tabular case, maintains an estimate $J_t(i)$, $\forall i \in \mathcal{X}$ & updates this estimate incrementally using samples.



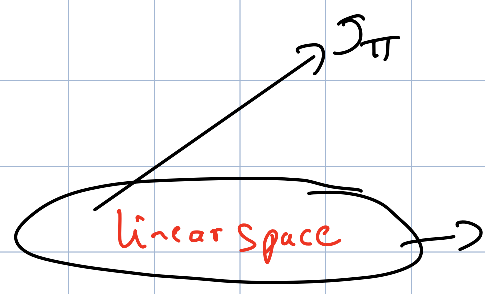
Problem with this approach: It doesn't scale. e.g. on large state spaced MDPs it may be computationally infeasible. e.g. consider a MDP with 10^{30} states or game Go has 10^{170} states (rough approx)

On such problem, we require parametric approximation of J_π.

e.g. $J_\pi(i) \approx \theta^T \phi(i)$ $\theta \in \mathbb{R}^d$, $d \ll n$.

parameter feature vector

linear function approximation



$S = \{ \Phi\theta \mid \theta \in \mathbb{R}^d \}$

Recall: Mean estimation

Want to estimate $\mu = EX$ & you keep getting samples $X_1, X_2, \ldots$

$$r_{m+1} = r_m + \beta_m \underbrace{(X_{m+1} - r_m)}_{\text{increment}}$$

r_{m+1}: estimate of μ from $\{X_1, \ldots X_m\}$

β_m: stepsize e.g. $\frac{1}{m}$

X_{m+1}: sample from distribution of X

r_m: prev. estimate of μ

Question of policy evaluation!

Want to estimate! $J_\pi(i) = E\left(\underbrace{\sum_{m=0}^{\infty} g(i_m, i_{m+1})}_{\text{"Total cost" r.v}} \middle| i_0 = i \right)$

Assume a SSP context.

Fixed point equation for J_π:

For simplicity, we have removed the dependence on action in single stage cost

$$J_\pi = T_\pi J_\pi$$

$$J_\pi(i) = \underset{\uparrow}{E}\left(g(i,\bar{i}) + J_\pi(\bar{i})\right)$$

expectation over next state $\bar{i}$

The action $\pi(i)$ is implicit.

or $\quad E\left(g(i,\bar{i}) + J_\pi(\bar{i})\right) - J_\pi(i) = 0$

Idea! sample "$g(i,\bar{i}) + J_\pi(\bar{i})$" & update incrementally

"Temporal difference"

$$(*) \quad J_{m+1}(i) = J_m(i) + \beta_m\left(\underbrace{g(i,\bar{i}) + J_m(\bar{i}) - J_m(i)}\right)$$

sample of the r.v. in expectation highlighted above

This is the TD(0) algorithm
temporal difference

NOTE: $\bar{i} \sim P_{i\bar{j}}(\pi(i))$

So, "$g(i,\bar{i}) + J_m(\bar{i})$" is a proxy for

$$E\left(g(i,\bar{i}) + J_m(\bar{i})\right) = (T_\pi J_m)(i)$$

$(*)$ is equivalent to

This is the mapping H in sto. iter. algo (1) from prev chapter

$$J_{m+1}(i) = J_m(i) + \beta_m\Big((T_\pi J_m)(i) - J_m(i)$$

$$+ \underbrace{g(i,\bar{i}) + J_m(\bar{i}) - (T_\pi J_m)(i)}\Big)$$

This is the noise factor $w_m(i)$ from the general sto. iter. algo.

Monte Carlo Policy Evaluation

SSP $\mathcal{S} = \{1, \ldots, n, T\}$

Fix proper policy π. Simulate the SSP to generate ℓ-trajectories

all start state are i

$(i_0^1, \ldots, i_N^1)$ Trajectory #1

$\vdots$

$(i_0^\ell, \ldots, i_N^\ell)$ Trajectory #ℓ

N is random

$i_N^1 = \cdots = i_N^\ell = T$

Actions using π.

Want to estimate: $J_\pi(i) = E\left(\sum_{m=0}^{\infty} g(i_m, i_{m+1}) \,\middle|\, i_0 = i\right)$

This is the r.v. whose expectation we estimate.

Let

$$\hat{J}_k(i_0^k) = g(i_0^k, i_1^k) + \cdots + g(i_{N-1}^k, i_N^k)$$

for $k = 1, \ldots, \ell$

$\hat{J} \to$ total cost in kth trajectory.

Assume $i_0^k = i \quad \forall k$.

trajectories used for averaging $\to \ell$

$$\tilde{J}_\ell(i) = \frac{1}{\ell} \sum_{k=1}^{\ell} \hat{J}_k(i) \quad \Leftarrow \text{Sample average.}$$

$\tilde{J}$ can be incrementally computed by

$$J_{m+1}(i) = J_m(i) + \beta_m \left(\hat{J}_m(i) - J_m(i) \right)$$

with initial condition $J_0(i) = 0$.

$\beta_m \rightarrow$ could be a general step size
$\left(\sum \beta_m = \infty, \sum \beta_m^2 < \infty \right)$

eg. $\beta_m = \frac{1}{m}$.

Reusing trajectories:

$(i_0, \dots, i_N) \rightarrow$ a trajectory

$(i_k, \dots, i_N) \rightarrow$ a sub-trajectory

Suppose e.g. $i_0 = i_k = i$

$\mathrm{I}(i, i_1, \dots\dots T)$ $\hat{J}(i,1)$

k-th state was i

$\mathrm{II}(i, i_{k+1}, \dots T)$ $\hat{J}(i,k)$

II is a sub-trajectory of I

$\hat{J}(i,1)$ & $\hat{J}(i,k)$ are dependent. But, we could still use them both to estimate $J_\pi(i)$

$$J_{m+1}(i_k) = J_m(i_k) + \beta_m \big(g(i_k, i_{k+1}) + \dots$$
$$\dots + g(i_{N-1}, i_N) - J_m(i_k) \big)$$

(**)

This would estimate J_π with start state i_k.

Suppose you obtain a trajectory

$$(i_0, i_1, \dots, i_k, \quad , i_N)$$

$i \qquad\qquad\qquad i$

For estimating $J_\pi(i)$, we could use entire trajectory

Terminal

to form $\hat{J}(i,1) = g(i_0, i_1) + g(i_1, i_2) + \dots + g(i_{N-1}, i_N)$

I could take the sub-trajectory starting at i_k

$$(i_k, i_{k+1}, \dots \; i_N)$$

state "i"

$$\hat{J}(i, k) = g(i_k, i_{k+1}) + \dots + g(i_{N-1}, i_N)$$

To estimate $J_\pi(i)$

"First-visit" variant! will use $\hat{J}(i,1)$

while "Every-visit" variant will use

both $\hat{J}(i,1)$ & $\hat{J}(i,k)$

are not independent

Lecture-22*

Monte Carlo policy evaluation & its relation to temporal differences.

Given a MDP trajectory $(i_k, \ldots, i_N)$ simulated using policy π.

One sample of "$\sum_{m=0}^{\infty} g(i_m, i_{m+1})$" r.v.

$$(*) \rightarrow J_{m+1}(i_k) = J_m(i_k) + \beta_m \Big(g(i_k, i_{k+1}) + \cdots + g(i_{N-1}, i_N) - J_m(i_k) \Big)$$

[Aside: If $\beta_m = \frac{1}{m}$, then $J_m \rightarrow$ is a sample mean]

Rewrite $(*)$ as

$$J_{m+1}(i_k) = J_m(i_k) + \beta_m \Big[\big(g(i_k, i_{k+1}) + J_m(i_{k+1}) - J_m(i_k) \big)$$
$$+ \big(g(i_{k+1}, i_{k+2}) + J_m(i_{k+2}) - J_m(i_{k+1}) \big)$$
$$\vdots$$
$$+ \big(g(i_{N-1}, i_N) + \cancelto{0}{J_m(i_N)} - J_m(i_{N-1}) \big) \Big]$$

Note: $J_m(i_N) = 0$ (why? $i_N = T$)

TD(0) uses only d_k to update.

$$J_{m+1}(i_k) = J_m(i_k) + \beta_m \underbrace{(d_k + d_{k+1} + \cdots + d_{N-1})}_{(**)},$$

where

$$d_\ell = \underbrace{g(i_\ell, i_{\ell+1}) + J_m(i_{\ell+1})}_{\text{Estimate of } J_\pi(i_\ell) \text{ based on a simulated transition}} - \underbrace{J_m(i_\ell)}_{\text{current estimate of } J_\pi(i_\ell)}, \quad \ell = k, \ldots, N-1$$

(**) can be done incrementally as

[Note: Going from $i_\ell \to i_{\ell+1}$ makes d_ℓ available]

$$\boxed{J_{m+1}(i_k) = J_m(i_k) + \beta_m d_\ell}$$

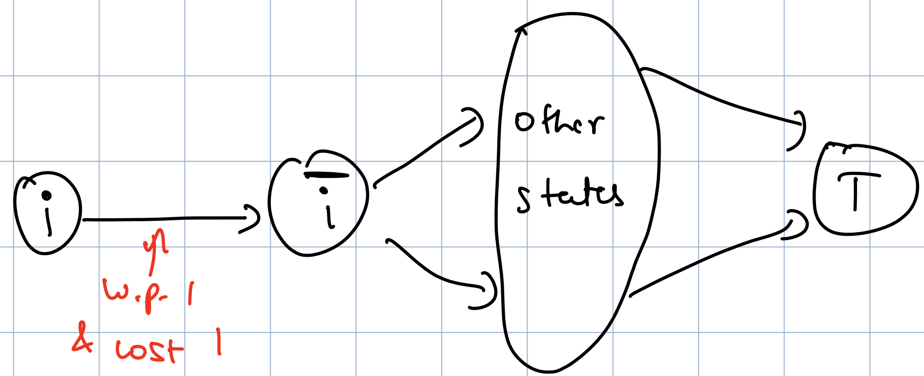
do this for $\ell = k, \ldots, N-1$

Remark: TD(0) updates as

$$J_{m+1}(i_k) = J_m(i_k) + \beta_m d_k$$

TD(0) does not use all the temporal differences ($d_k, d_{k+1}, \ldots, d_{N-1}$) to update its estimate $J_m(i_k)$, & instead relies on a 1-step fixed-point equation (i.e., uses d_k)

TD(0) vs. MCPE : A bias-variance tradeoff perspective



Fix some policy π.

From i, MDP always transitions to $\bar{i}$ under π, & $g(i,\bar{i})=1$.

Want to estimate: $J_\pi(i)$

MCPE: Simulate trajectories starting in i & collect the total cost samples, say $\{\hat{J}(m)\}_{m=1}^{N}$

$$\tilde{J}(i) = \frac{1}{N}\sum_{m=1}^{N} \hat{J}(m)$$

Is this an unbiased estimate of $J_\pi(i)$? Yes.

TD(0): Suppose through some other route (an independent simulation or some other approximate route), we have an estimate $J(\bar{i})$.

Then, we can estimate $J_\pi(i)$ by

$$J(i) = J(\bar{i}) + 1$$

MCPE would never use $J(\bar{i})$ & instead rely on sample trajectories, even at $\bar{i}$.

With TD(0), we have a biased estimate $J(i)$ of $J_\pi(i)$.

MCPE estimation may suffer from high variance (esp. if N is small), while a biased estimate $J(i)$ may do better.

Bottomline: TD(0) $\rightarrow$ biased estimate

MCPE $\rightarrow$ unbiased, but possibly high variance.

Is there a middle path?

Yes, TD(λ), $\lambda \in [0,1]$.

TD(0) used the 1-step fixed point equation to arrive at its update rule.

$$J_\pi(i) = E_{\bar{i}}\left(g(i,\bar{i}) + J_\pi(\bar{i}) \right) \Leftrightarrow J_\pi = T_\pi J_\pi$$

Why 1-step? We could also go 2 steps.

2-step fixed point equation

$$J_\pi(i) = \underset{\bar{i},\bar{\bar{i}}}{E}\left(g(i,\bar{i}) + g(\bar{i},\bar{\bar{i}}) + J_\pi(\bar{\bar{i}})\right) \quad (2)$$

$$J_\pi = T_\pi^2 J_\pi$$

Using the above, we can have the following iterative algorithm

$$J_{m+1}(i_k) = J_m(i_k) + \beta_m\Big(g(i_k, i_{k+1}) + g(i_{k+1}, i_{k+2}) + J_m(i_{k+2}) - J_m(i_k)\Big)$$

Can extend this to $(l+1)$-steps, i.e.,

$$\underbrace{J_\pi(i) = \underset{i_1,\dots,i_{l+1}}{E}\left(\sum_{m=0}^{l} g(i_m, i_{m+1}) + J_\pi(i_{l+1}) \,\Big|\, i_0 = i\right)}_{(l+1)\text{-step fixed point equation.}}$$

"Choice of l is arbitrary".

TD(λ) idea: Form a weighted average of the fixed point equations for different l.

How do we combine?

$$J_\pi(i) = (1-\lambda)\, E\left[\sum_{l=0}^{\infty} \lambda^l \Big(\overbrace{\sum_{m=0}^{l} g(i_m, i_{m+1}) + J_\pi(i_{l+1})}^{l\text{-step fixed point target.}}\Big)\right]$$

normalize (since $\sum \lambda^l = \frac{1}{1-\lambda}$)

weight for a particular l

$\lambda \in [0,1]$

We use the above fixed point equation to arrive at the TD(λ) update rule on Wednesday.

The fixed point equation that serves as a basis for TD(λ):

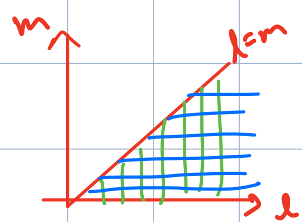
$$J_\pi(i) = (1-\lambda)\, E\left[\sum_{l=0}^{\infty} \lambda^l \overbrace{\left(\sum_{m=0}^{l} g(i_m, i_{m+1}) + J_\pi(i_{l+1})\right)}^{l\text{-step fixed point target.}}\right] \qquad \lambda \in [0,1]$$

normalize (since $\sum \lambda^l = \frac{1}{1-\lambda}$)   weight for a particular l

Trivial

$$= (1-\lambda)\, E\left[\sum_{l=0}^{\infty} \lambda^l \sum_{m=0}^{l} g(i_m, i_{m+1})\right] + (1-\lambda)\, E\left(\sum_{l=0}^{\infty} \lambda^l J_\pi(i_{l+1})\right)$$

$$= \underbrace{(1-\lambda)\, E\left[\sum_{m=0}^{\infty} \sum_{l=m}^{\infty} \lambda^l g(i_m, i_{m+1})\right]}_{(A)} + \underbrace{E\left(\sum_{l=0}^{\infty} (\lambda^l - \lambda^{l+1}) J_\pi(i_{l+1})\right)}_{(B)}$$



Simplifying the term (A):

$$(A) = (1-\lambda)\, E\left[\sum_{m=0}^{\infty} g(i_m, i_{m+1}) \sum_{l=m}^{\infty} \lambda^l\right] = E\left[\sum_{m=0}^{\infty} \lambda^m g(i_m, i_{m+1})\right] \quad (1)$$

since $(1-\lambda) \sum_{l=m}^{\infty} \lambda^l = \lambda^m$.

Simplifying the term (B):

$$E\left(\sum_{l=0}^{\infty} (\lambda^{l} - \lambda^{l+1}) J_{\pi}(i_{l+1}) \right)$$

$$= E\Big[(1-\lambda) J_{\pi}(i_1) + (\lambda - \lambda^2) J_{\pi}(i_2) + (\lambda^2 - \lambda^3) J_{\pi}(i_3) + \cdots \Big]$$

Added & subtracted this

$$= E\Big[(J_{\pi}(i_1) - J_{\pi}(i)) + \lambda (J_{\pi}(i_2) - J_{\pi}(i_1)) + \lambda^2 (J_{\pi}(i_3) - J_{\pi}(i_2)) + \cdots \Big] + J_{\pi}(i)$$

$$= E\left(\sum_{m=0}^{\infty} \lambda^{m} \left(J_{\pi}(i_{m+1}) - J_{\pi}(i_m) \right) \right) + J_{\pi}(i) \quad \text{— (2)}$$

Combining (1) & (2), we obtain

$$(**) \to J_{\pi}(i) = E\left(\sum_{m=0}^{\infty} \lambda^{m} \left(g(i_m, i_{m+1}) + J_{\pi}(i_{m+1}) - J_{\pi}(i_m) \right) \right) + J_{\pi}(i)$$

Recall $d_m = g(i_m, i_{m+1}) + J_{\pi}(i_{m+1}) - J_{\pi}(i_m)$

So, $(**) \Leftrightarrow$ $0 = E\left(\sum_{m=0}^{\infty} \lambda^{m} d_m \right)$ — (3)

This really is a finite sum with a random # of terms because $\forall m \geq N$, $d_m = 0$ since $i_N = T$.

Eq ③ is valid because $E(d_m) = 0 \quad \forall m$.

How to turn ③ into an iterative update rule?

NOTE to typesetters! Change variable m to something else

$$J_{t+1}(i) = J_t(i) + \beta_t \sum_{m=0}^{\infty} \lambda^m d_m \qquad (4)$$

→ TD(λ) update iteration.

A short detour into sto-approx update

$$h(x) = E(H(x))$$

$$x \to \boxed{\text{Box}} \to H(x)$$

Want to find x^* s.t. $h(x^*) = 0$

$$x_{t+1} = x_t + \beta_t (H(x_t))$$

→ Stochastic root-finding "Robbins-Monro" algorithm (1951)

$$= x_t + \beta_t \big(h(x_t) + \underbrace{(H(x_t) - h(x_t))}_{\text{noise}} \big)$$

Under regularity conditions (like A1-A3 before)

$x_t \to x^*$ a.s. as $t \to \infty$

where x^* satisfies $h(x^*) = 0$

Remarks:

① If $\lambda = 1$, then ④ becomes

$$J_{t+1}(i) = J_t(i) + \beta_t \sum_{m=0}^{\infty} d_m$$

$$J_{t+1}(i) = J_t(i) + \beta_t (d_0 + d_1 - - - + d_{N-1})$$

Assuming $i_N = T$

MCPE scheme $\Leftrightarrow$ TD(1)

② If $(\lambda = 0)$, then ④ becomes (using $0^0 = 1$)

Note: $(i_0 = i)$

$$J_{t+1}(i) = J_t(i) + \beta_t d_0$$

$$J_{t+1}(i) = J_t(i) + \beta_t \left(g(i_0, i_1) + J_t(i_1) - J_t(i_0) \right)$$

TD(0) update iteration

$i_0, - - - , i_{100}$

This TD term $\rightarrow g(i_{100}, i_{101}) + J_t(i_{101}) - J_t(i_{100})$
should have (in average) on $J_\pi(i_0) = J_\pi(i_1)$

③ For any $0 < \lambda < 1$, the temporal difference d_m is weighted by λ^m, making a future temporal difference less important while updating the estimate of the current state i, i.e., the effect of d_0 is more pronounced in TD(λ) update as compared to d_m, $m > 1$.

Note: λ weighs temporal differences. Not to be confused with discount (α).

P.T.O.

Lecture-23

TD(λ) variations:

I Every visit vs. first visit

Suppose we have a trajectory $(i_0, \dots, i_N)$

In this trajectory, a given state, say "i", may be visited more than once.

$$(i_0, i_m, \dots, i_k, \dots, i_N)$$

Terminal state

both are $= i$

Every visit: $(i_1, \ldots, i_N)$ & $(i_k, \ldots, i_N)$

to estimate $J_\pi(i)$

First visit: $(i_1, \ldots, i_N)$ to estimate $J_\pi(i)$

Formally, suppose i is visited M times in $(i_0, \ldots, i_N)$, and $(m_1, \ldots, m_M)$ are the time instants when state i is visited.

Then, TD(λ) would update as follows

$$J_{t+1}(i) = J_t(i) + \beta_t \sum_{j=1}^{M} \sum_{m=m_j}^{\infty} \lambda^{m-m_j} d_m$$

Why this?

In TD(λ) derivation, if we consider some state i_k instead of i_0, then the fixed point equation becomes

$$J_\pi(i_k) = E\left(\sum_{m=k}^{\infty} \lambda^{m-k} d_m\right) + J_\pi(i_k)$$

First visit TD(λ) would update as follows:

$$J_{t+1}(i) = J_t(i) + \beta_t \sum_{m=m_1}^{\infty} \lambda^{m-m_1} d_m$$

Question: Are these two variants equivalent?

No.

But, both variants can be shown to converge. (For $\lambda=1$, see Section 5.2 of NDP book)

Idea: SLLN

II Off-line TD(λ) vs Online TD(λ)

Offline: Simulate entire trajectory $(i_0, \ldots i_N)$ and update in the end, i.e., after all temporal differences $d_0, \ldots d_{N-1}$ are available.

Online: Update after each transition, i.e., after a single temporal difference term is available.

Offline TD-λ: (Every visit variant)

$$J_{t+1}(i) = J_t(i) + \beta_t \sum_{j=1}^{M} \sum_{m=m_j}^{\infty} \lambda^{m-m_j} d_m$$

One transition & "d_0" became available

$(i_0, i_1, \ldots i_N)$

Online TD(λ): (Incremental update)

initial estimate

On (i_0, i_1): $J_1(i_0) = J_0(i_0) + \beta\, d_0$

using constant stepsize for simplicity

$i_0 \xrightarrow{d_0} i_1 \xrightarrow{d_1} i_2$

On (i_1, i_2): $\quad J_2(i_0) = J_1(i_0) + \beta \lambda d_1$

Use d_1 to update estimates of value fn for i_0 & i_1

$\quad J_2(i_1) = J_1(i_1) + \beta d_1$

and so on.

In general, on (i_k, i_{k+1})

$$J_{t+1}(i_0) = J_t(i_0) + \beta \lambda^k d_k$$

$$J_{t+1}(i_1) = J_t(i_1) + \beta \lambda^{k-1} d_k$$

$$\vdots$$

$$J_{t+1}(i_k) = J_t(i_k) + \beta d_k$$

Remark! If a state "i" is visited multiple times, then offline and online TD(λ) result in different estimates.

Example to illustrate the updates of offline & online TD(λ):-

TD-terms → d_0, d_1, d_2

Suppose we have a trajectory $(1, 2, 1, T)$

Let $J_0(1)$ & $J_0(2)$ be initial values. $(J_0(T) = 0)$

Offline TD(λ): Denote estimates by $J_f(1)$ & $J_f(2)$

State 1's update: (Every visit style)

$(1,2,1,T)$ & $(1,T)$

$$J_f(1) = J_0(1) + \beta\left(d_0 + \lambda d_1 + \lambda^2 d_2 + d_2 \right)$$

$$\left.\begin{aligned} = J_0(1) + \beta\Big(&\big(g(1,2) + J_0(2) - J_0(1)\big) \\ &+ \lambda\big(g(2,1) + J_0(1) - J_0(2)\big) \\ &+ \lambda^2\big(g(1,T) - J_0(1)\big) \end{aligned}\right\} \text{from } (1,2,1,T)$$

$$+ \underbrace{g(1,T) - J_0(1)}_{\text{from sub-trajectory } (1,T)} \Big)$$

$$J_f(2) = J_0(2) + \beta\Big(\big(g(2,1) + J_0(1) - J_0(2)\big) + \lambda\big(g(1,T) - J_0(1)\big)\Big) \Big\} \to \text{from } (2,1,T)$$

On-line TD(λ) update: (Every-visit style)

On $(1,2)$:

$$J_1(1) = J_0(1) + \beta\big(\underbrace{g(1,2) + J_0(2) - J_0(1)}_{\text{process TD-term "}d_0\text{"}}\big)$$

$$J_1(2) = J_0(2)$$

P.T.O.

On $(2,1)$:

$$J_2(1) = J_1(1) + \beta\lambda\big(g(2,1) + J_1(1) - J_1(2)\big)$$
$$J_2(2) = J_1(2) + \beta\big(g(2,1) + J_1(1) - J_1(2)\big)$$

On $(1,T)$:

$$J_3(1) = J_2(1) + \beta\Big(\overset{(1,2,1,\cancel{T})}{\lambda^2\big(g(1,T) - J_2(1)\big)} + \overset{(1,T)}{\big(g(1,T) - J_2(1)\big)}\Big)$$
$$J_3(2) = J_2(2) + \beta\lambda\overset{(2,1,T)}{\big(g(1,T) - J_2(1)\big)}$$

$\big(J_3(1), J_3(2)\big) \rightarrow$ Online TD(λ) estimates.

Compare this with $\big(J_f(1), J_f(2)\big)$:

If we replace J_1 & J_2 by J_0 in online TD update, then $\big(J_3(1), J_3(2)\big) = \big(J_f(1), J_f(2)\big)$

J_1 & J_0 difference is $O(\beta)$

J_2 & J_0 difference is $O(\beta^2)$

J_3 & J_0 —— " —— $O(\beta^3)$

If we take the step-size $\beta \to 0$ as we update, (or $0<\beta<1$) then offline & online TD update will get close.

Lecture - 26*

Convergence of TD: A high-level sketch

Recall $J_{t+1}(i) = J_t(i) + \beta_t \left(H J_t(i) + w_t(i) - J_t(i) \right)$, $i = 1 \ldots n$ — (*)

If (i) H is a contraction,

(ii) $E(w_t(i) \mid \mathcal{F}_t) = 0$, $E(w_t^2(i) \mid \mathcal{F}_t) \leq A + B \|J_t\|^2$

(iii) $\sum \beta_t = \infty$, $\sum \beta_t^2 < \infty$ ← standard stochastic approximation conditions

then $J_t \xrightarrow[t \to \infty]{} J^*$ a.s., where $HJ^* = J^*$ — (1)

Want to show TD(λ) update is like the sto-iter-algo (*)

Condition (iii) requires no effort. e.g. $\beta_t = \frac{1}{t}$,

Condition (i): We will dig deeper into TD(λ)'s underlying mapping.

Note: We are doing policy evaluation using TD(λ) for a proper policy.

Side note! $\mathcal{X} = \{1, \ldots n\}$ $P = \begin{bmatrix} p_{11} & & \\ & \ddots & \\ & & p_{nn} \end{bmatrix}$ $p_{ij} = P(X_1 = j \mid X_0 = i)$

$$P(X_2 = k \mid X_0 = i) = \sum_j P(X_2 = k, X_1 = j \mid X_0 = i)$$

split & use Markov property

So, squaring matrix P, i.e., "P^2" would give two-step transition probabilities

Recall the fixed point equation underlying TD(λ):

$$J_\pi(i) = (1-\lambda)\, E\left[\sum_{\ell=0}^{\infty} \lambda^{\ell} \Big(\overbrace{\sum_{m=0}^{\ell} g(i_m, i_{m+1}) + J_\pi(i_{\ell+1})}^{\ell\text{-step fixed point target}} \Big) \Big|\, i_0 = i \right] \quad — ⓪$$

normalize (since $\sum \lambda^{\ell} = \frac{1}{1-\lambda}$) weight for a particular ℓ $\lambda \in [0,1]$

This is the TD(λ) operator

$$J_\pi = \boxed{G + (1-\lambda) \sum_{\ell=0}^{\infty} \lambda^{\ell} P^{\ell+1}}\, J_\pi \quad — ①$$

transition probability matrix underlying the policy considered.

$E\, J_\pi(i_{\ell+1})$ is like applying P $(\ell+1)$ times

G is $(1-\lambda) \sum_{\ell=0}^{\infty} \lambda^{\ell} E(g(-,-))$

What is the expectation over in eq ⓪

$$\underset{=i}{i_0} \xrightarrow{\pi(i_0)} i_1 \xrightarrow{\pi(i_1)} i_2 \text{ - - - - - - } \longrightarrow i_N = T$$

Sample trajectory $\{ i_0, i_1, i_2 \text{ - - - } i_N \}$ ($i_0 = i$, $i_N = T$)

Inside expectation is terms like

(i) $g(i_0, i_1) + J_\pi(i_1)$

(ii) $g(i_0, i_1) + g(i_1, i_2) + J_\pi(i_2)$ & so on.

each of these terms get a weight i.e., $\lambda^{\text{"something"}}$

Eq ① is like $J_\pi = H J_\pi$, where

$$H = G + (1-\lambda) \sum_{\ell=0}^{\infty} \lambda^{\ell} P^{\ell+1}$$

TD(λ) fixed point equation

In the all policies proper case, we showed earlier that there exists a positive vector ξ and some $\rho \in (0,1)$ s.t.

$$\sum_{j=1}^{n} p_{ij}(\pi(i))\,\xi(j) \le \xi(i) - 1 \le \rho\,\xi(i),$$

Or, equivalently

$$\|PJ\|_\xi \le \rho\,\|J\|_\xi,$$ where $\|\cdot\|_\xi$ is the weighted max-norm

Define $$HJ = G + (1-\lambda)\sum_{l=0}^{\infty} \lambda^l P^{l+1} J$$

Want $$\|HJ - HJ'\|_\xi \le \underset{0<\rho<1}{\rho}\,\|J - J'\|_\xi . \quad (**)$$

$\Rightarrow$ H is a ρ-contraction

& assuming the noise conditions $E(w_t(i)|\mathcal{F}_t) = 0$ & $E(w_t^2(i)|\mathcal{F}_t) \le A + B\|z_t\|^2$ are met,

one can infer TD(λ) iterate converges asymptotically.

If (**) holds, then the mapping underlying TD(λ) is a contraction & we can claim convergence using 1

$$\| HJ - HJ' \|_\xi$$

$$= \left\| \left(\cancel{g} + (1-\lambda) \sum_{l=0}^{\infty} \lambda^l P^{l+1} J \right) - \left(\cancel{g} + (1-\lambda) \sum_{l=0}^{\infty} \lambda^l P^{l+1} J' \right) \right\|_\xi$$

triangle inequality

$$\le (1-\lambda) \sum_{l=0}^{\infty} \lambda^l \| P^{l+1} (J - J') \|_\xi$$

$$\le (1-\lambda) \sum_{l=0}^{\infty} \lambda^l e \| J - J' \|_\xi$$

$$= (1-\lambda) e \| J - J' \|_\xi \sum_{l=0}^{\infty} \lambda^l$$

$$= e \| J - J' \|_\xi$$

Since

$$\| P^{l+1} (J - J') \|_\xi \le e^{l+1} \| J - J \|_\xi \le e \| J - J' \|_\xi$$

since $e \in (0,1)$

So, we have

$$\| HJ - HJ' \|_\xi \le e \| J - J' \|_\xi$$

So, the operator H underlying TD(λ) update is contractive.

Assuming condition (iii) leading up to Eq ① above hold, we can claim

$$J_t \to J^* \quad \text{a.s. as } t \to \infty$$

where J_t is the TD(λ) iterate.

TD(λ) for discounted MDPs

Policy evaluation using TD(λ)

Approach I

Since there is no termination state, to reuse the TD(λ) idea for SSPs, convert a discounted MDP to its equivalent SSP.

Approach II

Do something else

Approach I: Discounted MDP, fix some policy π.

An MDP trajectory has no end ($\because$ no termination state)

Add a "termination state" & from each state add a prob $(1-\alpha)$ transition ($\alpha \in (0,1)$ discount factor)

Simulation: Toss a coin with bias α in each time instant

If heads, continue simulation.
Else, end the trajectory & do TD(λ) update.

$\rightarrow (i_0, i_1, \ldots\ldots, i_N)$ → episode & aka trajectory aka sample path

i_N: termination state.

$N \in$ r.v. "Geometric" $E(N) = \dfrac{1}{1-\alpha}$

Use this trajectory to do TD(λ) update.

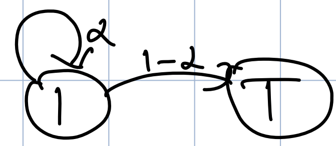
Drawback of this approach: High variance

Example: Just one state, say 1

Twist: single stage cost is a r.v. with mean 0 & variance σ^2 (note: cost does not depend on state).

Approach 1 adds termination state "T"

$J_\pi(1) = 0$



equivalent SSP

From a trajectory we got single stage costs

single stage cost samples $\to \{g_1, g_2, \ldots, g_N\}$ (N is a r.v.)

$\hat{J} = \sum_{i=1}^{N} g_i$

Variance of this total cost sample

$$= E\left((g_1 + g_2 + \cdots + g_N)^2\right)$$

conditional expectation works

$$= \sum_{k=1}^{\infty} E\left((g_1 + g_2 + \cdots g_N)^2 \mid N=k\right) P(N=k)$$

$$= \sum_{k=1}^{\infty} E\left((g_1 + g_2 + \cdots g_k)^2\right) P(N=k)$$

$$= \sum_{k=1}^{\infty} k\sigma^2 \, P(N=k)$$

$$= \sigma^2 \left(\sum_{k=1}^{\infty} k \, P(N=k)\right)$$

$$= \sigma^2 \, EN = \frac{\sigma^2}{1-\alpha}$$

Variance of estimate $\hat{J}$

$g_i \sim$ iid with mean 0 variance σ^2

$Z = X_1 + \cdots + X_k$

$X_i \sim \mathcal{N}(0, \sigma^2)$

$\{X_1, \ldots, X_k\}$ indep

$E\, Z^2 = k\sigma^2$

Note: α very close to 1 leads to "big" variance.

Approach 2:

1-Step fixed point equation $J_\pi(i) = E\left[g(i,\bar{i}) + \alpha J_\pi(\bar{i})\right]$

2-Step — " — $J_\pi(i) = E\left[g(i,\bar{i}) + \alpha g(\bar{i},\bar{\bar{i}}) + \alpha^2 J_\pi(\bar{\bar{i}})\right]$

TD(λ) fixed point equation for discounted case:

$$J_\pi(i_0) = (1-\lambda)\, E\left[\sum_{\ell=0}^{\infty} \lambda^\ell \left(\sum_{m=0}^{\ell} \alpha^m g(i_m, i_{m+1}) + \alpha^{\ell+1} J_\pi(i_{\ell+1})\right)\right]$$

Repeating all the steps from the SSP TD(λ) derivation, we obtain the following fixed point relation for discounted case:

(***)
$$J_\pi(i_0) = E\left(\sum_{m=0}^{\infty} (\alpha\lambda)^m d_m\right) + J_\pi(i_0),$$

where $d_m = g(i_m, i_{m+1}) + \alpha J(i_{m+1}) - J(i_m)$

Can do this update from an intermediate state i_k in the trajectory.

$$J_\pi(i_k) = E\left(\sum_{m=k}^{\infty} (\alpha\lambda)^{m-k} d_m\right) + J_\pi(i_k)$$

The TD(λ) update would be

$$J_{t+1}(i_k) = J_t(i_k) + \beta \sum_{m=k}^{\infty} (\alpha\lambda)^{m-k} d_m$$

in comparison to SSP case, we have the α factor here.

Ques: When to end trajectories? To be answered.

Lecture-25*

Variance calculation for the single state MDP

Note: We aren't adding the termination state.

Trajectory : $(i_0, i_1, i_2, \ldots\ldots)$

Total Cost sample $= g_1 + \alpha g_2 + \alpha^2 g_3 + \ldots\ldots$

$\{g_k\}$ iid
mean zero
Variance σ^2

$$\text{Variance} = E\left[\left(\sum_{k=0}^{\infty} \alpha^k g_{k+1}\right)^2\right]$$

$$= \sigma^2 \sum_{k=0}^{\infty} \alpha^{2k} = \frac{\sigma^2}{1-\alpha^2}$$

$$= \frac{\sigma^2}{(1-\alpha)(1+\alpha)}$$

With SSP formulation, we had a variance of $\frac{\sigma^2}{1-\alpha}$

which is $>$ than $\frac{\sigma^2}{1-\alpha^2}$

Approach 2 → Offline TD(λ) : cut the trajectory at some random time τ

" Simulate for a finite (τ) # of steps & approximate the discounted cost by

$$g_1 + \alpha g_2 + \cdots\cdots + \alpha^{\tau-1} g_\tau + \alpha^\tau J(i_\tau)$$

e.g. $\tau = 200$. $\alpha = 0.8$. Then, after 200 steps, the contribution of costs to J_π is negligible.

→ **Online TD(λ)**

update value function estimates on every sample transition.

Update rule ← very similar to the SSP case.

Just patch in "α".

Estimate: $$E\left(\sum_{m=0}^{\infty} \alpha^m g_m\right)$$

Tweak: $$E\left(\sum_{m=0}^{\tau} \alpha^m g_m\right)$$

Take samples of this expectation with a truncated trajectory.

If τ is large enough, then intuitively the contribution of $E\left(\sum_{m=\tau+1}^{\infty} \alpha^m g_m\right)$ to the total cost is negligible & $E\left(\sum_{m=0}^{\tau} \alpha^m \delta_m\right)$ is a good enough approximation.

Convergence analysis of TD(λ): skipped. Check Prop 5.1 of NDP book for details.

Why Q-factors in a learning scenario:

Policy evaluation: $J_\pi(i) = E_{\bar{i}}\left(g(i,\pi(i),\bar{i}) + J_\pi(\bar{i})\right)$

To learn J_π, sample from the r.v. inside expectation & do a sto-iter-algo.

Control: $J^*(i) = \min_a E\left(g(i,a,\bar{i}) + J^*(\bar{i})\right)$

Can I turn this fixed point relation into a sto-iter-algo?

Q-learning

Recall Q-factors:

optimal cost starting in j.

Def: $Q^*(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + J^*(j)\right)$

Take action a in state i & then follow the optimal policy from state j onwards.

Bellman equation $J^* = TJ^*$ is equivalent to

$$J^*(i) = \min_a Q^*(i,a)$$

Combining the two equations leads to the following "Q-Bellman equation".

$$Q^*(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \min_b Q^*(j,b)\right) \quad (\star)$$

optimal Q-values

Note: Q^* is the unique solution of $\star$.

Suppose Q is a solution of $\star$, i.e.,

$$Q(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \min_b Q(j,b)\right) \quad (1)$$

Thus, $Q = Q^*$.

This can be seen using the fact that J^* is the unique solution of $J^* = TJ^*$.

Q solves (1), so

$\left(\min_a Q(i,a)\right)$ solves the Bellman equation $J^* = TJ^*$

i.e., $J^*(i) = \min_a Q(i,a)$

J^* is unique $\Rightarrow$

$$\boxed{\min_a Q(i,a) = \min_a Q^*(i,a)}$$

Using $\min_b Q(j,b) = \min_b Q^*(j,b)$ in (1), we obtain $Q^* = Q$.

So, Q^* is the unique solution.

☆ We did not require contraction in this argument.

So, if $J^* = TJ^*$ has a unique solution, then $Q^* = HQ^*$ also has a unique solution

$\uparrow$ is the operator underlying Q-Bellman equation

LSPI
Policy gradient

Value iteration (VI) using Q-factors:

Q-value iteration

$$Q_{t+1}(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \min_b Q_t(j,b)\right)$$

(This is like $Q_{t+1} = HQ_t$, starting with some Q_0)

where $HQ(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \min_b Q(j,b)\right)$

A variation to VI is

$$Q_{t+1}(i,a) = (1-\beta_t)\, Q_t(i,a) + \beta_t \sum_j p_{ij}(a)\left(g(i,a,j) + \min_b Q_t(j,b)\right)$$

VI requires knowledge of these transition probabilities

Sto-iter-algo version of the above:

$$Q_{t+1}(i,a) = (1-\beta_t)\, Q_t(i,a) + \beta_t\left(g(i,a,\bar{i}) + \min_b Q_t(\bar{i},b)\right)$$

($\bar{i} \leftarrow$ sampled from $p_{ij}(a)$)

This is the Q-learning algorithm.

Note: The step-size could be iteration dependent i.e., β_t. Need $\sum \beta_t = \infty$ & $\sum \beta_t^2 < \infty$.

Remark: ① In principle, Q-learning is similar to TD(0). Both are based on VI & replace an expectation by its sample.

② There is no straightforward variation of Q-learning that is in the spirit of TD(λ). No $(l+1)$-step Q-Bellman equation. (Think about this!)

Lecture-26*

Convergence analysis of Q-learning:

Note: Convergence of Q-learning ⇒ Convergence of TD(0) (just consider a special MDP with only feasible action $\pi(i)$ in state i. Then Q-BE ⇔ ($J^\pi = T^\pi J^\pi$))

$$Q_{t+1}(i,a) = (1-\beta_t)\,Q_t(i,a) + \beta_t\Big(g(i,a,\bar{i}) + \min_b Q_t(\bar{i},b)\Big)$$

$\bar{i}$: Sampled from $P_{ij}(a)$

b: action in next state $\bar{i}$

Assumptions:

(A1) $\sum \beta_t = \infty$, $\sum \beta_t^2 < \infty$ ⟵ easy to satisfy e.g. $\beta_t = \frac{1}{t}$

(A2) All policies are proper in the underlying SSP.

Recall from previous chapter:

Suppose Q-learning update in compact notation is

$$Q_{t+1} = (1-\beta_t)\, Q_t + \beta_t \left(H Q_t + w_t \right)$$

Then, if we show $\left(\mathcal{F}_t = \sigma\left(Q_0, \dots, Q_t, w_0, \dots, w_{t-1} \right) \right)$

(B1) H is a contraction

(B2) $E(w_t \mid \mathcal{F}_t) = 0 \qquad E(w_t^2 \mid \mathcal{F}_t) \le A + B\|Q_t\|^2$

(B3) $\sum \beta_t = \infty, \qquad \sum \beta_t^2 < \infty$

Then, $Q_t \to Q^*$ (which is the fixed point of H) a.s. as $t \to \infty$.

Main proof (Theorem comes later):

Define $$\boxed{(HQ)(i,a) = \sum_j P_{ij}(a) \left(g(i,a,j) + \min_b Q(j,b) \right)}$$ $\forall i, a$.

Let $w_t(i,a) = g(i,a,\bar{i}) + \min_b Q_t(\bar{i}, b) - (HQ_t)(i,a)$

So, Q-learning update iteration is equivalent to

operator underlying Q-learning update — noise

$$Q_{t+1}(i,a) = (1-\beta_t)\, Q_t(i,a) + \beta_t \left((HQ_t)(i,a) + w_t(i,a) \right) \quad (**)$$

Verifying conditions on noise:

$w_t(i,a) = Y_t - E Y_t$, where

$Y = g(i,a,\bar{i}) + \min_b Q_t(\bar{i},b)$

$$E(w_t(i,a) \mid \mathcal{F}_t) = 0$$

$\Rightarrow$ w_t is conditionally (given Q_t) zero-mean

$$E(w_t^2(i,a) \mid \mathcal{F}_t) = E((Y_t - EY_t)^2 \mid \mathcal{F}_t)$$

$$\leq E(Y_t^2 \mid \mathcal{F}_t)$$

$Var(X) = E(X-EX)^2$
$= EX^2 - (EX)^2$
$\leq EX^2$

Assuming single stage cost $g(\cdot,\cdot,\cdot)$ is bounded, we have

$$E(Y^2 \mid \mathcal{F}_t) = E\left(\left(g(i,a,\bar{i}) + \min_b Q_t(\bar{i},b)\right)^2 \mid \mathcal{F}_t\right)$$

$(a+b)^2 \leq 2a^2 + 2b^2$

"$|g(i,a,j)| \leq$ const $\forall i,a,j$" & state-action spaces are both finite

$$\leq K\left(1 + \max_{j,b} Q_t^2(j,b)\right) \quad (*)$$

$$\Rightarrow \boxed{E(w_t^2(i,a) \mid \mathcal{F}_t) \leq K\left(1 + \max_{j,b} Q_t^2(j,b)\right)}$$

So, we have verified (B2)

(B3) is satisfied if we chose β_t carefully.

i.e., $\sum \beta_t = \infty$, $\sum \beta_t^2 < \infty$

Onto (B1): H is a contraction.

All policies proper $\Rightarrow$ $\exists$ a positive vector ξ & scalar ρ,

such that

"Fact A"

$$\sum_j p_{ij}(a)\,\xi(j) \le \rho\,\xi(i)$$ ← from SSP chapter.

Define $$\|Q\|_\xi = \max_{i,a} \frac{|Q(i,a)|}{\xi(i)}$$ ← weighted max-norm for Q-values.

ξ is given by the claim in SSP chapter.

Need to show: $\|HQ - HQ'\|_\xi \le \rho\,\|Q - Q'\|_\xi$ for some $\rho \in (0,1)$.

If this holds, then (B1) is satisfied.

Pf of (need to show):

Recall $$(HQ)(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \min_b Q(j,b)\right)$$

$$|(HQ)(i,a) - (HQ')(i,a)| = \left|\sum_j p_{ij}(a)\left(\min_b Q(j,b) - \min_b Q'(j,b)\right)\right|$$

$$\overset{\Delta^{le}\text{ ineq.}}{\le} \sum_j p_{ij}(a)\left|\min_b Q(j,b) - \min_b Q'(j,b)\right|$$

$$\le \sum_j p_{ij}(a) \max_b |Q(j,b) - Q'(j,b)|$$

$$= \sum_j p_{ij}(a)\left(\max_b \frac{|Q(j,b) - Q'(j,b)|}{\xi(j)}\right)\xi(j)$$

$$\le \sum_j p_{ij}(a)\left(\max_{j,b} \frac{|Q(j,b) - Q'(j,b)|}{\xi(j)}\right)\xi(j)$$

"add max over j"

$$\leq \sum_j P_{ij}(a) \|Q - Q'\|_\xi \, \xi(j) = \|Q-Q'\|_\xi \sum_j P_{ij}(a)\xi(j)$$

Using Fact A $\leq \|Q - Q'\|_\xi \, e \, \xi(i)$

So, we got

$$\boxed{|(HQ)(i,a) - (HQ')(i,a)| \leq \|Q - Q'\|_\xi \, e \, \xi(i)} \quad (\ast\ast)$$

$\Rightarrow$

$$\frac{|(HQ)(i,a) - (HQ')(i,a)|}{\xi(i)} \leq e\|Q - Q'\|_\xi$$

$$\max_{i,a} \frac{|(HQ)(i,a) - (HQ')(i,a)|}{\xi(i)} \leq e\|Q - Q'\|_\xi$$

$$\Rightarrow \quad \|HQ - HQ'\|_\xi \leq e \, \|Q - Q'\|_\xi$$

$\boxed{\Rightarrow \ H \text{ is a contraction wrt } \|\cdot\|_\xi}$

Thus, we have

Theorem (Q-learning convergence)

(A1) All policies proper (A2) $\sum \beta_t = \infty$, $\sum \beta_t^2 < \infty$

(A3) Single stage cost g is bounded, i.e., $\sup_{i,a,j} |g(i,a,j)| \leq M < \infty$

Under (A1)-(A3), the Q-learning algorithm converges a.s. i.e.,

$$Q_t \rightarrow Q^* \quad \text{a.s. as } t \rightarrow \infty.$$

A variation when (A1) is not satisfied.
Instead, we have "$\exists$ a proper policy" & "Improper policies have infinite cost".

Even here Q^* is the unique solution to the Q-Bellman equation (This didn't require H to be contractive. Instead, we only used $J^* = TJ^*$ & J^* is unique)

Question: Does Q-learning converge in this case?

(A1') $\exists$ a proper policy & all improper policies have infinite cost.

Under (A1'), H is a monotone mapping

i.e., $Q \leq Q' \Rightarrow HQ \leq HQ'$

"Check this using definition of H"
H.W.

Also check $\forall r$

$$Hr - \delta e \leq H(r - \delta e) \leq H(r + \delta e) \leq Hr + \delta e$$

e = vector of all ones.

Theorem: (Qlearning convergence under monotonicity)

$(A1') + $ " $\sum \beta_t = \infty, \ \sum \beta_t^2 < \infty$ " $+$

" bounded single stage cost "

$+$ " bounded iterate i.e., $\sup_{t,i,a} |Q_t(i,a)| < \infty$ "

$\Rightarrow \quad Q_t \to Q^*$ a.s. as $t \to \infty$

where $\quad HQ^* = Q^*$.

using Theorem 2 from previous chapter

A sufficient condition was stated after Theorem 2 in previous chapter.

This condition ensures boundedness of the iterates $\{Q_t\}$ & is satisfied for Q-learning.

For details, see Prop 5.6. of NDP book.

So, the final claim is Q-learning converges under $(A1')$ + step size condition. The boundedness of iterates is implied.

Lecture - 27

Q-learning for discounted MDPs:

Q-Bellman equation in a discounted setting:

$$Q^*(i,a) = \sum_j P_{ij}(a)\left(g(i,a,j) + \alpha \min_b Q^*(j,b)\right) \quad (\star)$$

discount

VI:

$$Q_{t+1}(i,a) = \sum_j P_{ij}(a)\left(g(i,a,j) + \alpha \min_b Q_t(j,b)\right)$$

Q-learning:

$$\boxed{Q_{t+1}(i,a) = (1-\beta_t)\, Q_t(i,a) + \beta_t\left(g(i,a,\bar{j}) + \alpha \min_b Q_t(\bar{j},b)\right)}$$

sampled from $P_{ij}(a)$

Convergence of Q-learning:

(A1') Assume $|g(\cdot,\cdot,\cdot)| \le M < \infty$

(A2') $\sum \beta_t = \infty, \quad \sum \beta_t^2 < \infty$

Then, following the proof in the "all policies are proper" case of SSP-Q-learning, one can infer that the underlying operator HQ is a α-contraction.

$$(HQ)(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \alpha \min_b Q(j,b)\right)$$

$$\|HQ - HQ'\|_\infty \le \alpha \|Q - Q'\|_\infty$$

max-norm $\quad \|Q\|_\infty = \max_{i=1 \ldots n} |Q(i)|$

So, Under (A1'), (A2'), we have
$Q_t \to Q^*$ w.p.1 as $t \to \infty$

Issue of exploration:

MCPE: $\quad E\Big(\underbrace{\sum_{k=0}^{\infty} \alpha^k g(i, \pi(i), j)}_{}\Big)$

trajectories to do policy evaluation

$\hat{J}(i) \to$ estimate of the value function/expected cost,

then $\quad \hat{J}(i) \longrightarrow J_\pi(i) \quad$ if

you see enough trajectories starting with "i".

$\hat{J}(i)$: sample average $\xrightarrow{\text{converge}}$ $J_\pi(i)$ if we see state i infinitely often in the trajectories.

Same logic applies to TD and Q-learning

With TD, for $J_t(i) \longrightarrow J_\pi(i)$ as $t \to \infty$

we need to visit i "i.o" in the trajectories.

But, sampling is using a fixed policy π.

Compare with Q-learning:

$$Q_t(i,a) \xrightarrow[t\to\infty]{} Q^*(i,a)$$

Here, the algorithm is free to choose the action "a" in each state.

The requirement for convergence:

$$\underbrace{\text{All pairs "}(i,a)\text{" are visited frequently.}}_{\text{issue of exploration}}$$

Q-learning: $\quad (i) \xrightarrow{a} (\bar{i}) \longrightarrow$

update the Q-table

which action to choose

If $Q_t \approx Q^*$

(i,a_1) … (i,a_n) Q-table

then select $a' = \arg\min Q(\bar{i}, \tilde{a})$

Why do we need to take the route using Q-factors for finding an optimal policy?

Policy evaluation: we need J which is a function of state variable.

So, why $Q(i,a)$ & Q-Bellman equation?

Or, why not use $\boxed{J^* = T J^*}$ & make a sto-iterative algo to find J^*?

Normal Bellman equation:
$$J^*(i) = \min_a \sum_j p_{ij}(a)\left(g(i,a,j) + \alpha J^*(j)\right) \rightarrow (1)$$

Q-Bellman equation:
$$Q^*(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \alpha \min_b Q^*(j,b)\right) \rightarrow (2)$$

General sto. iter-algo: Want to estimate $\mu = E(X)$

Take samples of X & do an iterative update.

Eq (2) is in the form
$$Q^*(i,a) = E_j\big(\underbrace{g(i,a,j) + \alpha \min_b Q^*(j,b)}_{\text{Can sample this}}\big)$$

& do an iterative update.

Eq (1) is of the form
$$J^*(i) = \min_a E(\_\_)$$

Since min is outside, an iterative algo is not possible by just replacing the expectation above with a sample.

How to do exploration:

For Q-learning to converge, we require all state-action pairs to be visited frequently.

Recall Q-Learning update:

$$Q_{t+1}(i,a) = (1-\beta_t)\, Q_t(i,a) + \left(g(i,a,j) + \alpha \min_b Q(j,b) \right)$$

Question: How to choose actions?

I Greedy: In state i, choose $\arg\min_a Q_t(i,a)$, at time instant t. If $Q_t \approx Q^*$, then this choice makes perfect sense. However, if Q_t isn't close to Q^*, then "we need to explore".

$Q(1,a_1)$
$\vdots$
$Q(1, a_m)$
$Q(2, a_1)$
$\vdots$
$Q(\lvert X \rvert, a_m)$

Q-table

→ Each entry of this table has to be update a good # of times for Q-learning to converge

II ϵ-greedy:

Fix $\epsilon > 0$, usually a small number.

At time instant t, pick the greedy action w.p. $(1-\epsilon)$

& pick an action unif. at random w.p. ϵ.

Alternative: Make ϵ a function of iteration t, say ϵ_t, & take $\epsilon_t \to 0$ as $t \to \infty$, i.e., reduce exploration as algorithm updates.

III

In state i, at time instant t,

action a is chosen w.p.

temperature parameter

$$\xi(i,a) = \underbrace{\frac{\exp\left(-Q_t(i,a)/\tau\right)}{\sum_{b \in A(i)} \exp\left(-Q_t(i,b)/\tau\right)}}$$

$\underbrace{\xi(i) = \left[\xi(i,a_1) \ldots \xi(i,a_m)\right]}$

Prob distribution over actions in state i

This is the prob. of choosing action a

$\tau \to$ "temperature" $\to$ controls the exploration.

Note: If τ is very small, the choice is "greedy".

Policy evaluation! $J_{\pi_k} = T_{\pi_k} J_{\pi_k}$

exact

no noise

MPI: $J_{k+1} = T_{\pi_k}^{m_k} J_k$

Compare this with TD(0):

$$J_{t+1}(i) = J_t(i) + \beta_t \left(T_{\pi_k} J_t(i) + w_t(i) - J_t(i) \right)$$

noise

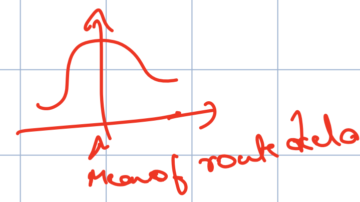
Running TD(0) for some # of steps, say M,
" J_M " J_M is very close to J_{π_k}

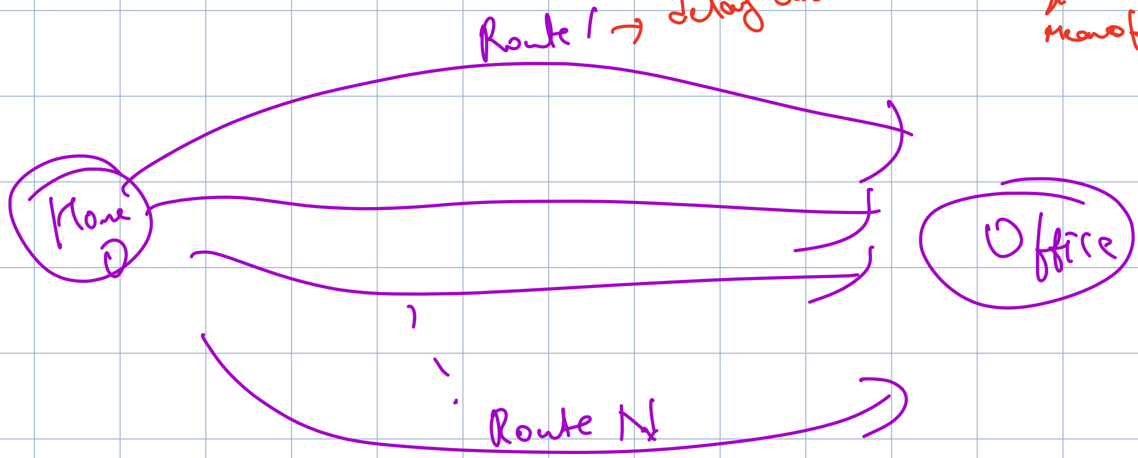
& using J_M for policy improvement,
there is no guarantee of improvement.

"Konda-Borkar
"Actor-critic algos"
1999, SIAM. J. Control. Opt

Bandit angle to exploration:

Naive setting





Routes: 1, – – N

Means: $\mu_1, - - \mu_N$

$$\mu^* = \min_{i=1-N} \mu_i$$

Opt-route $\quad i^* = \arg\min_{i=1-N} \mu_i$

for day = 1, 2, 3, – –

{

Pick a route $\text{Route}_{\text{day}}$

Observe a sample delay from distribution P_{day}

}

On some day $\longrightarrow$ sample average $\hat{r}_1$

$\vdots$

$\hat{r}_N$

How to pick route on following day?

exploitation	exploration
Pick the route with best sample avg	pick a random route

route to pick $\rightarrow$ $\arg\max \hat{r}_i$

In Q-learning

exploitation	exploration
$\arg\min Q_t(i,a)$	picking a random action

ϵ-greedy

exploitation	exploration
w.p. $(1-\epsilon)$	w.p. ϵ