Lecture-21st

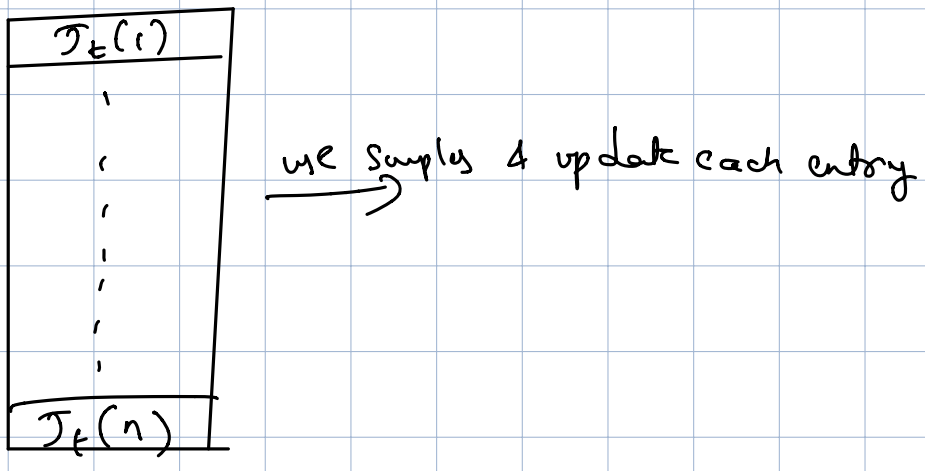
TD-learning, Q-learning with full state representations

ref: Chapter 5 of NDP book

Full state-representations?

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

A full-state algorithm aka tabular case, maintains an estimate $J_t(i)$, $\forall i \in X$ & 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

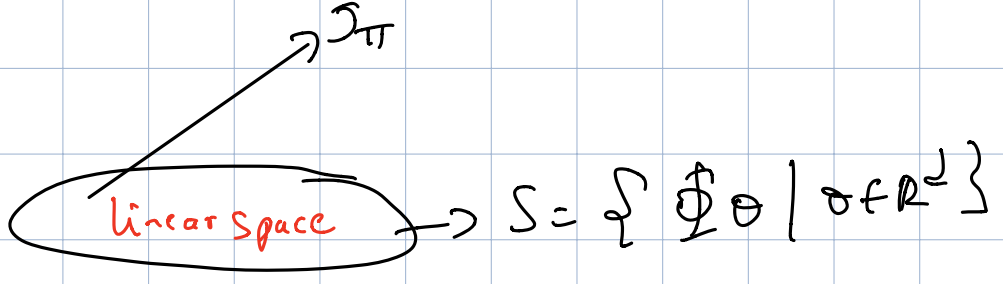
or game Go has 10^{170} states (rough approx)

On such problem, we require parametric approximation of J_π.

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

θ: parameter vector; $\phi(i)$: feature vector

linear function approximation



Recall: Mean estimation

Want to estimate $\mu = EX$

$$r_{m+1} = r_m + \beta_m (X_{m+1} - r_m)$$

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

Assume a SSP context.

Fixed point equation for J_π:

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

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

↑ expectation over next state $\bar{i}$

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

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

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

sample of the r.v. in expectation highlighted above

This is the TD(0) algorithm = temporal difference

NOTE: $\bar{i} \sim p_{ij}(\pi(i))$

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

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

This is the noise factor $w_m(i)$ from the general sto. it. alg.

Monte Carlo Policy Evaluation

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

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

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

$\vdots$

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

N is random

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

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

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

for $k = 1, \ldots l$

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

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

$$\tilde{J}(i) = \frac{1}{l} \sum_{k=1}^{l} \hat{J}(k)$$ ← Sample average.

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

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

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

Rewrite trajectories:

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

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

$$J_{mel}(i_k) = J_m(i_k) + \beta_m \left(g(i_k, i_{k+1}) + \cdots + g(i_{N-1}, i_N) - J_m(i_k) \right)$$

J_{mel}

This would estimate J_{TT} with start state i_k.

Lecture-22

Monte Carlo policy evaluation & its relation to temporal differences.

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

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

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

Rewrite $(*)$ as

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

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

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

where

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

(**) can be done incrementally as

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

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

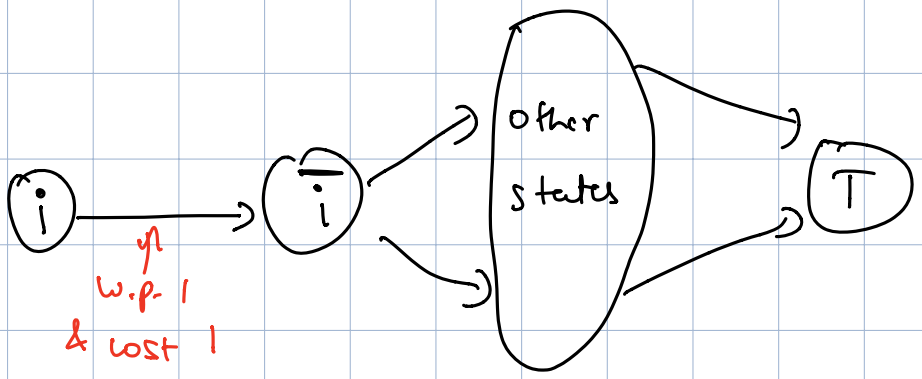
→ do this for $l = k, \ldots, N-1$

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 $\tilde{J}(\bar{i})$.

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) → biased estimate

MCPE → 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.

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

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

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

Using the above, we can have the following iterative algorithm

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

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

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

$(\ell+1)$-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?

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

ℓ-step fixed point target.

$(1-\lambda)$: normalize (since $\sum_\ell \lambda^\ell = \frac{1}{1-\lambda}$)

λ^ℓ: 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.

Lecture-23*

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

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

l-step fixed point target.

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

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

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

Simplifying the term (A):

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

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

Simplifying the term (B):

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

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

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

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

Combining ① & ②, we obtain

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

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

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

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

→ TD(λ) update iteration.

Remarks:

① If $\lambda = 1$, then ④ becomes

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

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

↑ MCPE scheme ⟺ TD(1)

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

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

(Note: $i_0 = i$)

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

TD(0) update iteration

Note to typesetters: See if you can make i in $J_\pi(i)$ / $J_t(i)$ as i_0.

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).

TD(λ) variations:

I Every visit vs. first visit

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

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

$(i_0, i_1, \ldots, i_k, \ldots, i_N)$

both are $= i$

Every visit: $(i_1, \dots, i_N)$ & $(i_k, \dots i_N)$ to estimate $J_\pi(i)$

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

Formally, suppose i is visited M times in $(i_0, \dots i_N)$, and $(m_1, \dots m_M)$ 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:

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

Online TD(λ): (Incremental update)

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

→ using constant stepsize for simplicity

On (i_1, i_2):

$$J_2(i_0) = J_1(i_0) + \beta \lambda d_1$$

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

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.

Lecture-26*

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

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)

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

$$= J_0(1) + \beta\Big(\big(g(1,2) + J_0(2) - J_0(1)\big) + \lambda\big(g(2,1) + J_0(1) - J_0(2)\big) + \lambda^2\big(g(1,T) - J_0(1)\big) + g(1,T) - J_0(1)\Big)$$

from (1,2,1,T)

from sub-trajectory (1,T)

$$J_f(2) = J_0(2) + \beta\Big(\big(g(2,1) + J_0(1) - J_0(2)\big) + \lambda\big(g(1,T) - J_0(1)\big)\Big)$$

→ from (2,1,T)

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

On (1,2):

$$J_1(1) = J_0(1) + \beta\big(g(1,2) + J_0(2) - J_0(1)\big)$$

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

P.T.O.

On $(2,1)$:

$$V_2(1) = V_1(1) + \beta\lambda\left(g(2,1) + V_1(1) - V_1(2)\right)$$

$$V_2(2) = V_1(2) + \beta\left(g(2,1) + V_1(1) - V_1(2)\right)$$

On $(1,T)$:

$$V_3(1) = V_2(1) + \beta\left(\lambda^2\left(g(1,T) - V_2(1)\right) + \left(g(1,T) - V_2(1)\right)\right)$$

$$V_3(2) = V_2(2) + \beta\lambda\left(g(1,T) - V_2(1)\right)$$

$(V_3(1), V_3(2)) \rightarrow$ Online TD(λ) estimates.

Compare this with $(V_f(1), V_f(2))$:

If we replace V_1 & V_2 by V_0 in online TD update,

then $(V_3(1), V_3(2)) = (V_f(1), V_f(2))$

V_1 & V_0 difference is $O(\beta)$

V_2 & V_0 difference is $O(\beta^2)$

V_3 & V_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.

Convergence of TD: A high-level sketch

Recall $$J_{t+1} = J_t + \beta_t (H J_t + w_t - J_t)$$ — (*)

If (i) H is a contraction,

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

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

then $J_t \to J^*$, where $HJ^* = J^*$ — (1)

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

Condition (ii) 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.

Recall the fixed point equation underlying TD(λ):

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

(l-step fixed point target.)

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

$$J_\pi = G_1 + (1-\lambda) \sum_{l=0}^{\infty} \lambda^l P^{l+1} J_\pi$$

P: transition probability matrix underlying the policy considered.

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

G_1 is $(1-\lambda) \sum_{l=0}^{\infty} \lambda^l E(g(-,-))$

For a proper policy, we showed earlier that there exists a positive vector ξ and some $e \in (0,1)$ s.t.

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

Or, equivalently

$$\|PJ\|_\xi \le e\, \|J\|_\xi, \quad \text{where } \|\cdot\|_\xi \text{ is the weighted max-norm.}$$

Define $HJ = G_1 + (1-\lambda) \sum_{l=0}^{\infty} \lambda^l P^{l+1} J$

Want $\|HJ - HJ'\|_\xi \le e\, \|J - J'\|_\xi$ — (xx)

($0 < e < 1$)

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

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

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

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

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

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

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

So, we have

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

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

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

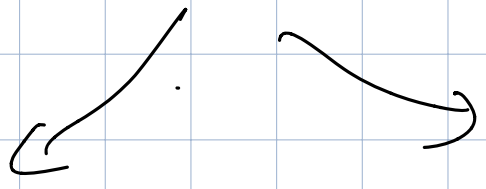
$$J_t \to J^* \quad \text{a.s. as } t \to \infty$$

where J_t is the TD(λ) iterate.

Lecture-25

TD(λ) for discounted MDPs

Policy evaluation using TD(λ)



Approach I

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

Approach II

Do something else

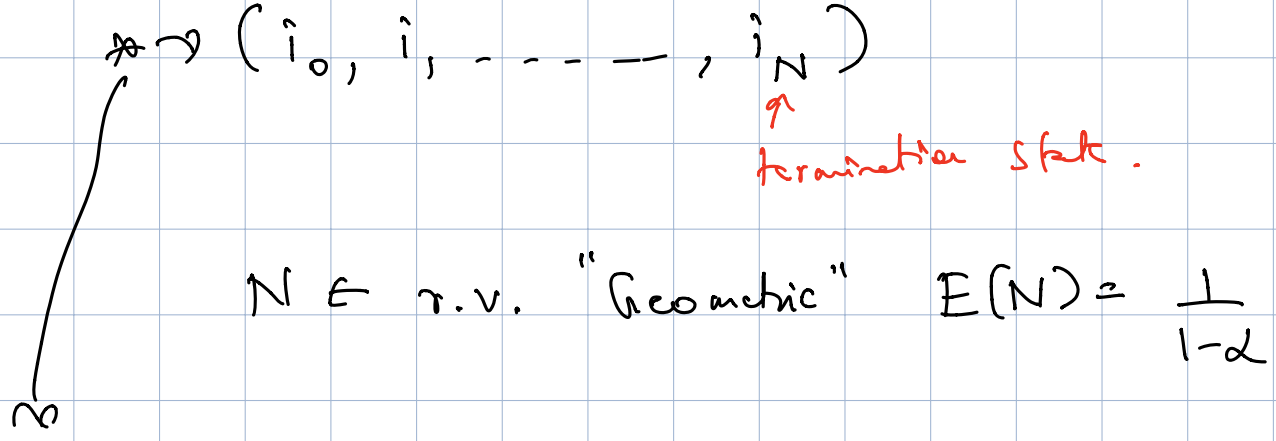
Approach I: Discounted MDP, fix some policy π.

An MDP trajectory has no end (∵ 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.



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

Use this trajectory to do TD(λ) update.

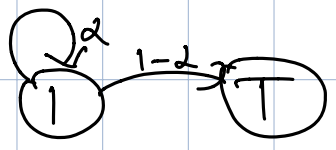
Drawback of this approach: High variance

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"

$J_\pi(1) = 0$



From a trajectory we get single stage costs

$g_1, g_2, \dots g_N$ (N is a r.v.)

Variance of this total cost sample

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

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

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

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

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

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

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

$(\ell+1)$-step fixed point equation

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

Repeating all the steps from the SSP TD(λ) derivation

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

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.

Variance calculation for the single state MDP

Note: We aren't adding the termination state.

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

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

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

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

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

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

$$g_1 + 2g_2 + \cdots\cdots + \alpha^{\tau-1} g_\tau + \alpha^\tau J(i_\tau)$$

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 $\mathbb{E}\left(\sum_{m=\tau+1}^{\infty} \alpha^m g_m\right)$ to the total cost is negligible & $\mathbb{E}\left(\sum_{m=0}^{\tau} \alpha^m g_m\right)$ is a good enough approximation.

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

Lecture-26*

Q-learning

Recall Q-factors:

optimal cost starting in j.

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

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":

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

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

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

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

Thm, $Q = Q^*$.

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 $\Rightarrow$

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

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

↑ is the operator underlying Q-Bellman equation

Value iteration (VI) using Q-factors:

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

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

A variation to VI is

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

VI requires knowledge of these transition probabilities

Sto-iter-algo version of the above:

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

($\bar{i} \leftarrow$ 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!)

Convergence analysis of Q-learning:

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

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

$\bar{i}$: sampled from $p_{ij}(a)$; b: action in next state $\bar{i}$

Assumptions:

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

(A2) All policies are proper

Recall from previous chapter:

Suppose Q-learning update in compact notation is

$$Q_{t+1} = (1-\beta_t) Q_t + \beta_t (HQ_t + \omega_t)$$

Thm, if we show ($\mathcal{F}_t = \sigma(Q_0, \ldots, Q_t, \omega_0, \ldots, \omega_{t-1})$)

(B1) H is a contraction

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

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

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

Main proof (Theorem comes later):

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

$\forall i, a$.

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

So, Q-learning update iteration is equivalent to

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

(**)

Verifying conditions on noise:

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

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

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

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

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

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

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

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

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

So, we have verified (B2)

(B3) is satisfied if we chose β_t carefully.

Onto (B1): H is a contraction.

All policies proper $\Rightarrow$ $\exists$ a positive vector ξ & scalar c, such that

Fact ↑ $\sum_j P_{ij}(a)\,\xi(j) \le e\,\xi(i)$ ← from SSP chapter.

Define $\|Q\|_\xi = \max_{i,a} \frac{|Q(i,a)|}{\xi(i)}$

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

If this holds, then (B1) is satisfied.

Pf of (need to show):

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

$|(HQ)(i,a) - (HQ')(i,a)|$

$\le \sum_j P_{ij}(a)\,|\min_b Q(j,b) - \min_b Q'(j,b)|$ ← D^e inequ.

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

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

↓ take max over j as well.

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

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

So, we got

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

$\Rightarrow$

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

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

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

$\Rightarrow$ H 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.

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

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

A variation when (A1) is not satisfied.

Instead, we have "∃ a proper policy" & "improper policies have infinite cost".

Even here Q^* is the unique solution to the Q-Bellman equation.

Question: Does Q-learning converge in this case?

(A1') ∃ 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 :-

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

c = vector of all ones.

Theorem: (Q-learning convergence under monotonicity)

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

"bounded style stage cost" +

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

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

where $HQ^* = Q^*$.

} 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:

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

(α: discount; $\min_b Q^*(j,b)$: (*))

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:

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

($\bar{j}$: 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.

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

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

$\uparrow$ max-norm

So, Under (A1'), (A2'), we have

$Q_t \to Q^*$ w.p.1 as $t \to \infty$

Issue of exploration:

MCPE: $E\left(\sum_{k=0}^{\infty} \alpha^k g(i, \pi(i), j)\right)$

$\downarrow$

trajectories to do policy evaluation

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

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

All pairs "(i,a)" are visited frequently.

issue of exploration

Lecture - 28*

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

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

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

Or, why not use $J^* = TJ^*$ & 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: $Q^*(i,a) = \sum_j p_{ij}(a)\left(g(i,a,j) + \alpha \min_b Q^*(j,b)\right)$ $\hookrightarrow (2)$

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

Take samples of X & do an iterative update.

Eq (2) is in the form $Q^*(i,a) = \underset{j}{E}\left(g(i,a,j) + \alpha \min_b Q^*(j,b)\right)$

↓

Can sample this & do an iterative update.

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

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

How to do exploration:

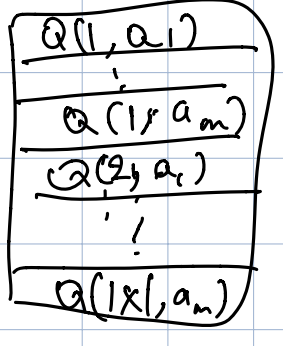
For Q-learning to converge, we require all state-action pairs to be visited frequently.

Recall Q-learning update:

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

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-table

→ Each entry of this table has to be updated 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.

$$\frac{\exp\left(-Q_t(i,a)/\tau\right)}{\sum_b \exp\left(-Q_t(i,b)/\tau\right)}$$

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".