RL Lecture 9 and 10 notes

Tuesday, October 3, 2017 2:25 PM

Audio 1

Audio recording 2017

Started 3 mins
DP 30
Why fixed poi
Asynchronous
Policy improv

Before:

① $V^*(s) = \max_{\pi} V^{\pi}(s)$

② Goals & MDPs.

Pop Quiz discussion

$V^{\pi}(s)$ = value of a state (single state)

$V^{\pi}(s_0)$ = value of a state-value distribution. (S_0 : is r.v.)

↓

r.v. over dist. of start states.

This would be $J(\pi)$

POLICY EVALUATION

Given a policy π, find V^{π}.

assume P & R are known. Its essentially a planning problem.

Method:

① Solve system of equations produced by the Bellman equations.

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} P(s,a,s')\left(R(s,a,s') + \gamma V^{\pi}(s')\right)$$

Bellman eqn.

② Dynamic Programming

DYNAMIC PROGRAMMING

Sequence of approximations of V^{π}.

$\hat{V}_0^{\pi}, \hat{V}_1^{\pi}, \hat{V}_2^{\pi} \ldots$

↓

- initial guess of Value function for every state
- is a vector
- chosen arbitrarily except for $V_0^{\pi}(s) = 0$ for terminal states (S_∞). Terminal state

$$\hat{V}_0^{\pi} = \begin{bmatrix} \hat{V}_0^{\pi}(s_1) \\ \hat{V}_0^{\pi}(s_2) \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} \vdots \\ \hat{V}_0^{\pi}(s_n) \end{bmatrix}$$

$$\hat{V}_{k+1}^{\pi}(s) = \sum_a \pi(s,a) \sum_{s'} P(s,a,s')\left[R(s,a,s') + \gamma \hat{V}_k^{\pi}(s')\right]$$

Properties:

1. $\hat{V}_k^{\pi} = V^{\pi}$ is a fixed point. $f(x) = x$.
2. $\hat{V}_k^{\pi} \longrightarrow V^{\pi}$ as $k \to \infty$ for finite MDPs with bounded rewards
3. A full pass over the state space is called a "full backup". A si state update is called a backup.

Example:

↓	↓	↓	↓
↓	↓	↓	↓
↓	↓	↓	↓
→	→	→	→

s^{∞}

$R_t = -1$ always

π is shown

$\gamma = 1$

$\hat{V}_0^{\pi} =$

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$\hat{V}_1^{\pi} =$

-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	0

$V(s) =$ $\uparrow$, $\pi(s,a) = 0$

$\rightarrow$ $\pi(s,a) = 0$

$\downarrow$ $\vdots$ $\longrightarrow$ $= 0$ Reward at $K+1$

This value never changes as terminal state

$\hat{V}_2^{\pi} =$

-2	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-1
-2	-2	-1	0

$\hat{V}_3^{\pi}$ =

-3	-3	-3	-3
-3	-3	-3	-2
-3	-3	-2	-1
-3	-2	-1	0

$\hat{V}_4^{\pi}$ =

-4	4	-4	-3
-4	-4	-3	-2
-4	-3	-2	-1
-3	-2	-1	0

$\hat{V}_5^{\pi}$ =

-5	-5	-4	-3
-5	-4	-3	-2
-4	-3	-2	-1
-3	-2	-1	0

$\hat{V}_6^{\pi}$ =

-6	-5	-4	-3
-5	-4	-3	-2
-4	-3	-2	-1
-3	-2	-1	0

now it converges.

In-place implementation

- When update $\hat{V}_k^{\pi}(s)$ store it back in the same table rather than
- Update states in any order.
- Asynchronous updates
  → update a state many times before updating another state.
- Guaranteed to converge to V^{π} if no states are starved of updat

→ Order doesn't change final convergence but it will ch

Policy Improvement

If we have estimated q^{π}, how can we improve π

Policy Improvement Theorem:

Let π and π' be deterministic policies.

such that for all $s \in S$. :

$$q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s) = q^{\pi}(s, \pi(s))$$

$\hookrightarrow q^{\pi}(s, 2) \longrightarrow$ say

Then, $\pi' \geq \pi$

$\downarrow$

$\forall s \in S, \; V^{\pi'}(s) \geq V^{\pi}(s)$

LECTURE-10 starts here:

Proof:

$$V^{\pi}(s) \leq Q^{\pi}(s, \pi'(s))$$

$$= \mathbb{E}[\underbrace{R_t}_{\text{from } \pi} + \underbrace{\gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} \ldots\ldots}_{\text{from } \pi'} \mid S_t = s, a_t =$$

$\downarrow$

actio

$$= \mathbb{E}[R_t + \gamma V^{\pi}(S_{t+1}) \mid S_t = s, \pi']$$

π' influences i

follow π thereafter (S_t+1) onwards

S_{t+1} is a r.v.

we do care about how we got here.

$$\leq \mathbb{E}[R_t + \gamma q^{\pi}(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s, \pi']$$

$$= \mathbb{E}[R_t + \gamma \mathbb{E}[R_{t+1} + \gamma V^{\pi}(S_{t+2}) \mid S_t = s, \pi']$$

Why no S_{t+1}?

Because S_{t+}

The dist of S_{t}

$$= \mathbb{E}\left[R_t + \gamma R_{t+1} + \gamma^2 V^{\pi}(S_{t+2}) \mid S_t = s, \pi'\right]$$
$$\leq \mathbb{E}\left[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 V^{\pi}(S_{t+3}) \mid S_t = s, \pi'\right]$$
$$\leq \mathbb{E}\left[\right.$$
$$\leq V^{\pi}$$

POLICY IMPROVEMENT

- For all states, select an action a that maximises $q^{\pi}(s,a)$
- This policy is called "greedy with respect to q^{π}."

$$\pi'(s) \in \underset{a}{\operatorname{argmax}} \sum_{s'} P(s,a,s')\left[R(s,a,s') + \gamma V^{\pi}(s')\right]$$

- what if $\pi' = \pi$? (fixed pt.)

So, $V^{\pi'}(s) = \max\limits_{a} \sum\limits_{s'} P(s,a,s')\left[R(s,a,s') + \gamma V^{\pi}(s')\right]$

Policy Iterations:

$$\pi_0 \xrightarrow{\text{evaluate}} V^{\pi_0} \xrightarrow{\text{improve}} \pi_1 \xrightarrow{\text{eval}} V^{\pi_1} \xrightarrow{\text{improve}} \pi_2 \longrightarrow \ldots$$

- Will terminate after finite # iterations.
- Exists an optimal policy that is deterministic.

Problem is evaluation?

→ The hard way is to run a policy fully & evaluate

But too expensive.

So, we will take $\underline{\underline{\text{some}}}$ backups and approximate V^{π}.

not all

VALUE Iteration

→ one full backup during policy evaluation in policy iteration.

V_0 : arbitrarily

π_0 : arbitrarily (but deterministic)

$$V_1 : \quad \forall s \quad V_1(s) = \sum_{s'} P(s, \underline{\underline{\pi_0(s)}}, s') \; R(s', \pi_0(s), s') + \gamma V_0(s')$$

no sum over actions as
π_0 is deterministic

$$\pi_1 : \quad \forall s \quad \pi_1 s \in \operatorname{argmax}_a \sum_{s'} P(s,a,s')\left[R(s,a,s') + \gamma V_1(s')\right]$$

$$V_2 : \quad \forall s \quad V_2(s) = \sum_{s'} P(s, \pi_1(s), s')\left[R(s, \pi(s), s') + \gamma V_1(s')\right]$$

$$\pi_2 : \quad \forall s \quad \pi_2(s) \in \operatorname{argmax}_a \sum_{s'} P(s,a,s')\left[R(s,a,s') + \gamma V_2(s')\right]$$

$$V_{k+1}(s) = \max_a \sum_{s'} P(s,a,s')\left[R(s,a,s') + \gamma V_k(s')\right]$$

This looks like Bellman Optimality equation
We can say that we have taken the BOE c

PROPERTIES:

1) $V_k \longrightarrow V^*$

$\boxed{\text{Asynchronous: You update some state 3 times before updating some other state.}}$

Value function converges. [How? wait for proof]

$\boxed{\text{BELLMAN OPERATOR}}$

[To prove Value Iteration converges]

View state value functions as vectors in $\mathbb{R}^{|S|}$

$$V_k = \begin{bmatrix} V_k(s_1) \\ V_k(s_2) \\ \vdots \\ V_k(s_{|S|}) \end{bmatrix}$$

Bellman Operator : $\mathbb{R}^{|S|} \Longrightarrow \mathbb{R}^{|S|}$

$$T(V_K) \triangleq V_{K+1}$$

$$\boxed{(T V_K)}(s) = V_{K+1}(s)$$

$$= \max_a \sum_{s'} P(s,a,s')\left[R(s,a,s') + \gamma \underline{\underline{V_K(s)}}\right]$$

This whole thing is a vector

s^{th} entry of vector.

Contraction Mapping

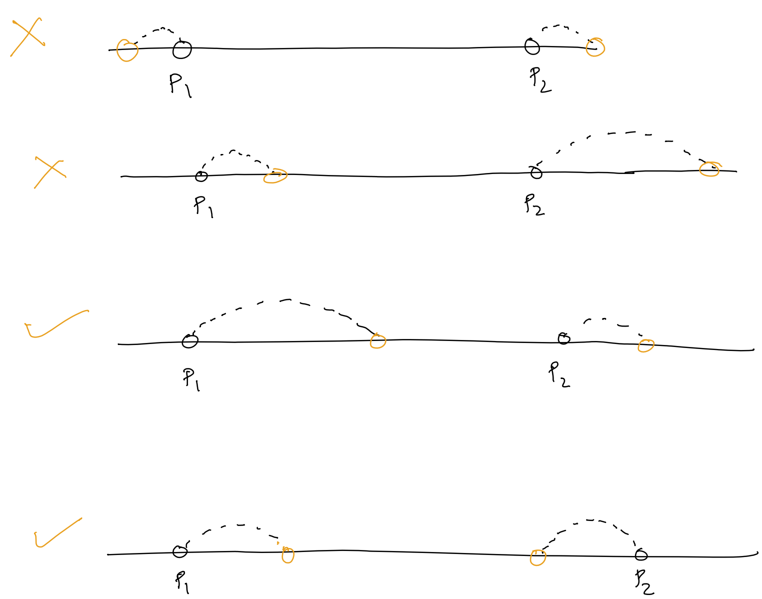
$f : \mathcal{X} \to \mathcal{X}$ is a contraction mapping if there exists a $\lambda \in [0,1)$ such that $\forall x \in \mathcal{X}$ and $\forall y \in \mathcal{X}$

$d\left(f(x), f(y)\right) \leq \lambda\, d(x,y)$ where $\underline{\underline{d}}$ is a distance function.

distance to self $= 0$

$\triangle$ inequality

$d \geq 0$



Banach fixed point theorem

If f is a contraction mapping on a non-empty complete normed vector space, then f has a unique fixed pt. and the sequence defined by $x_{k+1} = f(x_k)$

BELLMAN OPERATOR IS CONTRACTION MAPPING

$f \leftarrow T \quad \mathcal{X} \leftarrow \mathbb{R}^{|S|}, \; d(v, v') \triangleq \max |V(s) - V'(s)|$

$$\| Tv - Tv' \| \leq \lambda \| v - v' \|$$

where

$$\| Tv - Tv' \| \triangleq \max_s \left| (Tv - Tv')(s) \right|$$

$$\text{or} \quad \max_s \left| Tv(s) - Tv'(s) \right|$$

$$= \max_s \left| \max_a \sum_{s'} P(s,a,s') \left[R(s,a,s') + \gamma V(s') \right] - \right.$$

$$\leq \max_s \max_a \left| \sum_{s'} P(s,a,s') \left[R(s,a,s') + \gamma V(s' \right. \right.$$

Claim: $\left| \max_x f(x) - \max_x g(x) \right| \leq \max_x \left| f(x) - g(x) \right|$

$$\forall x \quad f(x) - g(x) \leq |f(x) - g(x)|$$

$$f(x) \leq |f(x) - g(x)| + g(x)$$

$$\max_x f(x) \leq \max_x |f(x) - g(x)| + g(x)$$

$$\leq \max_x |f(x) - g(x)| + \underbrace{\max_x g(x)}_{\text{move to LHS}}$$

$$\max_x f(x) - \max_x g(x) \leq \max_x |f(x) - g(x)|$$