

11/30

## Reinforcement learning

\* What if  $P(s'/s, a)$  and  $R(s)$  are not known?

Can we learn  $\pi^*$  or  $V^*(s)$  from experience?

1) Model-based (indirect) approach

Explore world, estimate model  $\hat{P}(s'/s, a) \approx P(s'/s, a)$ , compute  $\hat{\pi}^*$  from  $\hat{P}(s'/s, a)$

\* Cons: to store  $P(s'/s, a)$  is  $O(n^2)$  for  $n$  states

Only care about  $\pi^*(s)$  or  $V^*(s)$  which are  $O(n)$ .

Is it really necessary to estimate a model?

\* Pro: model  $P(s'/s, a)$  useful for task transfer, where rewards  $R(s)$  or discount factor  $\gamma$  changes, but  $P(s'/s, a)$  stays the same.

2) Direct approach: learn  $\pi^*(s)$ ,  $V^*(s)$  w/o building model. How?

## Aside: Stochastic approximation theory

\* How to estimate mean of random variable  $X$  from samples  $X_0, X_1, \dots, X_T$ ?

1) obvious sample average

$$\mu = \frac{1}{T} (X_0 + X_1 + \dots + X_{T-1})$$

: estimate converges to mean  $\mu \rightarrow E[X]$  as  $T \rightarrow \infty$  by law of large numbers.

2) incremental update

initialize  $\mu_0 = 0$

update  $\mu_t = (1 - \alpha_t) \mu_{t-1} + \alpha_t X_t$  for  $0 < \alpha_t < 1$  **learning rule**

also write this as:  $\mu_t = \mu_{t-1} + \alpha_t (X_t - \mu_{t-1})$

**new estimate = old + td** temporal difference (TD)

known as TD learning algorithm.

Thm:  $\mu_t \rightarrow E[X]$  as  $t \rightarrow \infty$  with probability 1 if

(i)  $\sum_{t=1}^{\infty} \alpha_t = \infty$  (diverges)

(ii)  $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$  (converges)

Intuitively: (i)  $\alpha_t$  decays sufficiently slowly to incorporate large # samples.

(ii)  $\alpha_t$  decays sufficiently fast to allow for convergence (damp oscillations).

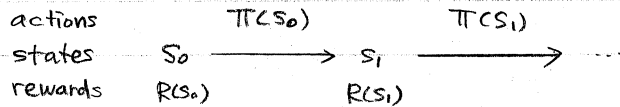
↓ Final

## Temporal difference (TD) prediction

\* How to evaluate policy without model?

How to compute  $V^\pi(s)$  without knowing  $P(s'|s, \pi(s))$ ?

\* Explore state space via policy  $\pi$



\* Recall Bellman equation:

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

\* TD learning algorithm

Initialize  $V_0(s) = 0$  for all  $s$  (at time  $t=0$ )

$$\text{Update: } V_{t+1}(s_t) = \underbrace{V_t(s_t)}_{\text{previous estimate}} + \underbrace{\alpha}_{\text{learning rate}} \left[ \underbrace{R(s_t) + \gamma V_t(s_{t+1})}_{\text{random sample}} - V_t(s_t) \right] \quad \text{known as TD}(\phi).$$

\* Features:

- update after each step of experience
- learns directly from experience w/o model.
- easy to implement.

\* Asymptotic convergence

$$\lim_{t \rightarrow \infty} V_t(s) \rightarrow V^\pi(s)?$$

Assume that each state of MDP is visited infinitely often by policy  $\pi$ .

Then,  $\text{TD}(\phi)$  converges:

- "with probability 1" if:

- each state  $s$  has its own learning rate  $\alpha_v(s)$  where  $v$  denotes # visits so far to state  $s$

- learning rates satisfy for all states  $s$ .

$$(i) \sum_{v=1}^{\infty} \alpha_v(s) = \infty$$

$$(ii) \sum_{v=1}^{\infty} \alpha_v^2(s) < \infty$$

Should agents in practice enforce (i) and (ii)?

- yes, for theoretical convergence guarantee
- no, for non-stationary worlds where MDP is just an approximation.
- "in mean" if step size  $\alpha$  is constant and sufficiently small.

## Q-learning

\* How to optimize policy  $\pi^*$  without model  $P(s'|s, a)$ ?

How to compute  $Q^*(s, a)$  without model?

\* Explore state-action space at random:

actions  $a_0$   $a_1$  ... Not following any particular policy!  
 states  $s_0 \rightarrow s_1 \rightarrow \dots$   
 rewards  $R(s_0)$   $R(s_1)$

\* Bellman optimality equation

$$Q^*(s, a) = R(s) + \gamma \sum_{s'} P(s'|s, a) V^*(s')$$

$$Q^*(s, a) = R(s) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} [Q^*(s', a')]$$

\* One-step Q-learning:

Initialize  $Q_0(s, a) = 0$  for all states  $s$  and actions  $a$ .

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha [R(s_t) + \gamma \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t)]$$

\* Features:

- simple, incremental
- model-free
- experience-based.

\* Asymptotic convergence:  $\lim_{t \rightarrow \infty} Q_t(s, a) \rightarrow Q^*(s, a)$  appropriately

Thm: if each state-action pair is visited infinitely often, and an  $\alpha$  decaying step size  $\alpha_t(s, a)$  is used for each state-action pair, then Q-learning converges with probability one.