EE6550 Machine Learning

Lecture Ten – Reinforcement Learning

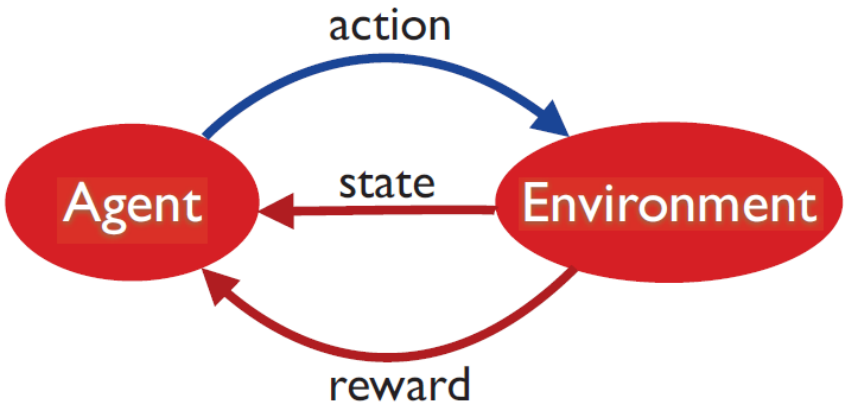
Chung-Chin Lu

Department of Electrical Engineering

National Tsing Hua University

May 8, 2017

Reinforcement Learning Problem



- Agent: a learner actively interacting and exploring the environment to achieve a certain goal.
  - The achievement of the agent's goal is typically measured by the reward he receives from the environment.
- Environment: the entity responding to the action taken by the agent and returning the agent's current state in the entity and the immediate reward to the agent.

- Problem: find action policy that maximizes cumulative reward over the course of interactions.

Key Features

- Contrast with supervised learning:
  - No explicitly labeled training data given passively. Information is collected through a course of actions by interacting with the environment.
  - Distribution determined by actions taken.
- Delayed rewards or penalties:
  - No future or long-term reward feedback is provided by the environment.
- Reinforcement learning trade-off:
  - Exploration (of unknown states and actions) to gain more reward information; vs.
  - Exploitation (of known information) to optimize reward.

Two Main Settings

- Planning problem: the environment model is known to the agent.
- Learning problem: the environment model is unknown to the agent.
  - The agent must learn from the state and reward information gathered to both gain information about the environment and determine the best action policy.

The Contents of This Lecture

- Markov decision processes (MDP) model of the environment
- Action policy
- Planning algorithms
- Learning algorithms

Markov Decision Process (MDP)

- $\{0, 1, \ldots, T-1\}$: a set of decision epochs;
- $\mathcal{S}$: a set of states, possibly infinite;
- $\mathcal{A}$: a set of actions, possibly infinite.
- $Pr[s'|s, a]$: the transition probability to the next state s' given the current state s and action a taken.
- $Pr[r'|s, a]$: the probability of reward returned given the current state s and action a taken.

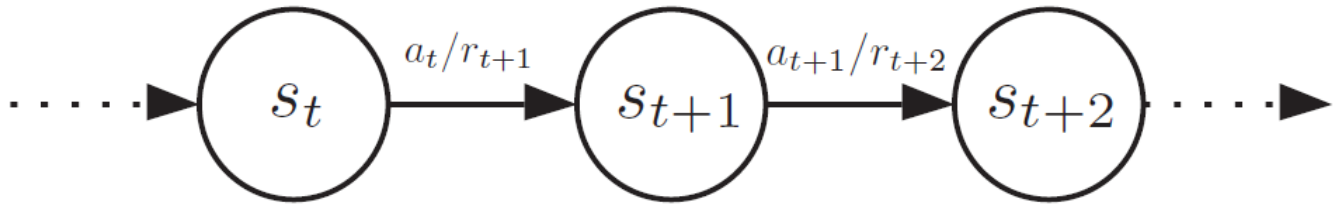
- Markov properties: for all $t \geq 0$, $s', s, s_{t-1} \ldots, s_0 \in \mathcal{S}$, $a, a_{t-1}, \ldots, a_0 \in \mathcal{A}$, $r', r_t, r_{t-1}, \ldots, r_1 \in \mathbb{R}$,

$$\begin{aligned} & Pr[S_{t+1} = s' | S_t = s, A_t = a, R_t = r_t, S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, \\ & \quad R_{t-1} = r_{t-1}, \ldots, S_1 = s_1, A_1 = a_1, R_1 = r_1, S_0 = s_0, A_0 = a_0] \\ = \quad & Pr[S_{t+1} = s' | S_t = s, A_t = a] = Pr[s' | s, a] \end{aligned}$$

and

$$\begin{aligned} & Pr[R_{t+1} = r' | S_t = s, A_t = a, R_t = r_t, S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, \\ & \quad R_{t-1} = r_{t-1}, \ldots, S_1 = s_1, A_1 = a_1, R_1 = r_1, S_0 = s_0, A_0 = a_0] \\ = \quad & Pr[R_{t+1} = r' | S_t = s, A_t = a] = Pr[r' | s, a]. \end{aligned}$$

Time Evolution Model of an MDP

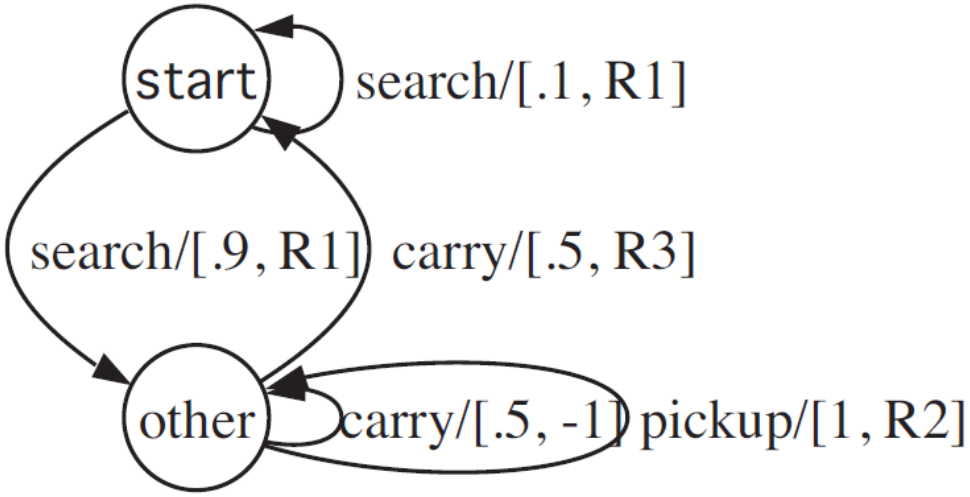


- $S_t \in \mathcal{S}$: state observed at time t;
- $A_t \in \mathcal{A}$: action taken at time t;
- $S_{t+1} \in \mathcal{S}$: state reached at time $t+1$;
- $R_{t+1} \in \mathbb{R}$: reward received at time $t+1$.

Properties of MDPs

- Finite or infinite MDPs: both $\mathcal{S}$ and $\mathcal{A}$ are finite sets or not.
- Finite or infinite horizon: $T < \infty$ or $T = \infty$.
- Deterministic or random reward: the immediate reward $r' = r(s, a)$ is deterministic or random with distribution $Pr[r'|s, a]$ given s and a.

Example: Robot Picking Up Balls



A simple MDP for a robot picking up balls on a tennis court.

- $\mathcal{S} = \{\text{start}, \text{other}\}$: the set of states;
- $\mathcal{A} = \{\text{search}, \text{carry}, \text{pickup}\}$: the set of actions;
- $R1, R2, R3$: possible reward values.

The Contents of This Lecture

- Markov decision processes (MDP) model of the environment
- Action policy
- Planning algorithms
- Learning algorithms

Policy

- Definition:
  - A stationary policy is a mapping $\pi : \mathcal{S} \to \mathcal{A}$.
    - Usually employed in the infinite horizon.
  - A non-stationary policy is a sequence of mappings $\pi_t : \mathcal{S} \to \mathcal{A}$ for $t = 0, 1, 2, T - 1$.
    - Usually employed in a finite horizon.
- Objective: find a policy $\{\pi_t\}$ maximizing expected (reward) return.

- The return along a realization of a sequence of states $S_{t_0}(\omega), \ldots, S_{T-1}(\omega)$ following a policy $\{\pi_t\}$:
  - Finite horizon return:
    $\sum_{\tau=0}^{T-1-t_0} r(S_{t_0+\tau}(\omega), \pi_{t_0+\tau}(S_{t_0+\tau}(\omega)))$.
  - Infinite horizon return:
    $\sum_{\tau=0}^{\infty} \gamma^\tau r(S_{t_0+\tau}(\omega), \pi_{t_0+\tau}(S_{t_0+\tau}(\omega)))$.
    * $\gamma \in (0, 1)$ is a constant factor to discount future rewards.
  - Return is a random variable.

Policy Value

- Definition: The value $V_{\{\pi_t\}}(s)$ of a policy $\{\pi_t\}$ at state $s \in \mathcal{S}$ is the expected (reward) return when starting at s and following policy $\{\pi_t\}$:
  - finite horizon:

$$V_{\{\pi_t\}}(s) \triangleq E\left[\sum_{\tau=0}^{T-1-t_0} r(S_{t_0+\tau}(\omega), \pi_{t_0+\tau}(S_{t_0+\tau}(\omega)))\Big| S_{t_0} = s\right]$$

  - infinite horizon with a discount factor $\gamma \in (0, 1)$:

$$V_{\{\pi_t\}}(s) \triangleq E\left[\sum_{\tau=0}^{\infty} \gamma^\tau r(S_{t_0+\tau}(\omega), \pi_{t_0+\tau}(S_{t_0+\tau}(\omega)))\Big| S_{t_0} = s\right]$$

- Problem: Find a policy $\{\pi_t\}$ with maximum value for all starting states.

Linear Bellman Equations for Policy Evaluation

Proposition 14.1: The values $V_\pi(s)$ of a stationary policy π at states $s \in \mathcal{S}$ for an infinite horizon MDP obey the following system of linear equations:

$$\forall\ s \in \mathcal{S},\ V_\pi(s) = E[r(s, \pi(s))] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s, \pi(s)] V_\pi(s').$$

Proof.

$$
\begin{aligned}
& V_\pi(s) \\
= \;& E\left[\sum_{\tau=0}^{\infty} \gamma^\tau r(S_{t_0+\tau}, \pi(S_{t_0+\tau}))\Big| S_{t_0} = s\right] \\
= \;& E\Big[E\Big[\sum_{\tau=0}^{\infty} \gamma^\tau r(S_{t_0+\tau}, \pi(S_{t_0+\tau}))\Big| S_{t_0+1}, S_{t_0} = s\Big]\Big| S_{t_0} = s\Big] \\
= \;& E\Big[E\Big[r(S_{t_0}, \pi(S_{t_0}))\Big| S_{t_0+1}, S_{t_0} = s\Big]\Big| S_{t_0} = s\Big] \\
& + E\Big[E\Big[\sum_{\tau=1}^{\infty} \gamma^\tau r(S_{t_0+\tau}, \pi(S_{t_0+\tau}))\Big| S_{t_0+1}, S_{t_0} = s\Big]\Big| S_{t_0} = s\Big] \\
= \;& E\Big[r(S_{t_0}, \pi(S_{t_0}))\Big| S_{t_0} = s\Big] \\
& + \gamma E\Big[E\Big[\sum_{\tau'=0}^{\infty} \gamma^{\tau'} r(S_{t_0+1+\tau'}, \pi(S_{t_0+1+\tau'}))\Big| S_{t_0+1}, S_{t_0} = s\Big]\Big| S_{t_0} = s\Big].
\end{aligned}
$$

Since

$$E\Big[\sum_{\tau'=0}^{\infty}\gamma^{\tau'}r(S_{t_0+1+\tau'},\pi(S_{t_0+1+\tau'}))\Big|S_{t_0+1},S_{t_0}=s\Big]$$

$$= \sum_{s'\in\mathcal{S}} Pr[S_{t_0+1}=s'|S_{t_0}=s]\cdot$$

$$E\Big[\sum_{\tau'=0}^{\infty}\gamma^{\tau'}r(S_{t_0+1+\tau'},\pi(S_{t_0+1+\tau'}))\Big|S_{t_0+1}=s',S_{t_0}=s\Big]$$

$$= \sum_{s'\in\mathcal{S}} Pr[S_{t_0+1}=s'|S_{t_0}=s,A_{t_0}=\pi(s)]\cdot$$

$$E\Big[\sum_{\tau'=0}^{\infty}\gamma^{\tau'}r(S_{t_0+1+\tau'},\pi(S_{t_0+1+\tau'}))\Big|S_{t_0+1}=s'\Big]$$

by the Markov property, we have

$$E\Big[E\Big[\sum_{\tau'=0}^{\infty}\gamma^{\tau'}r(S_{t_0+1+\tau'},\pi(S_{t_0+1+\tau'}))\Big|S_{t_0+1},S_{t_0}=s\Big]\Big|S_{t_0}=s\Big]$$

$$= \sum_{s'\in\mathcal{S}} Pr[S_{t_0+1}=s'|S_{t_0}=s,A_{t_0}=\pi(s)]\cdot$$

$$E\Big[E\Big[\sum_{\tau'=0}^{\infty}\gamma^{\tau'}r(S_{t_0+1+\tau'},\pi(S_{t_0+1+\tau'}))\Big|S_{t_0+1}=s'\Big]\Big|S_{t_0}=s\Big]$$

$$= \sum_{s'\in\mathcal{S}} Pr[s'|s,\pi(s)]E\Big[\sum_{\tau'=0}^{\infty}\gamma^{\tau'}r(S_{t_0+1+\tau'},\pi(S_{t_0+1+\tau'}))\Big|S_{t_0+1}=s'\Big]$$

$$= \sum_{s'\in\mathcal{S}} Pr[s'|s,\pi(s)]V_{\pi}(s').$$

Since $E\Big[r(S_{t_0}, \pi(S_{t_0}))\Big|S_{t_0} = s\Big] = E\Big[r(s, \pi(s))\Big]$, we have

$$V_\pi(s) \quad = \quad E\Big[r(s, \pi(s))\Big] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s, \pi(s)]\ V_\pi(s').$$

□

Linear Bellman Equations - Matrix Form

The linear Bellman equations can be written as

$$\boldsymbol{V}_\pi = \boldsymbol{R}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{V}_\pi,$$

where

- $\boldsymbol{V}_\pi$: the policy value column vector whose sth component is $V_\pi(s)$;
- $\boldsymbol{R}_\pi$: the policy-dependent reward column vector whose sth component is $E\Big[r(s, \pi(s))\Big]$;
- $\boldsymbol{P}_\pi$: the policy-dependent transition probability matrix whose (s, s')th component is $Pr[s'|s, \pi(s)]$.

Uniqueness and Existence

Theorem 14.1: For a finite MDP, linear Bellman equations admit a unique solution for any stationary policy π given by

$$\boldsymbol{V}_\pi = (I - \gamma \boldsymbol{P}_\pi)^{-1} \boldsymbol{R}_\pi.$$

Optimal Policy for the Infinite Horizon

- Definition: An optimal policy for the infinite horizon is a stationary policy π^* with maximal value for all states $s \in \mathcal{S}$ over all possible stationary policies π.
  - value of π^* (optimal value):

$$\forall\ s \in \mathcal{S},\ V_{\pi^*}(s) = \max_{\pi} V_{\pi}(s).$$

- $V^*(s) \triangleq V_{\pi^*}(s)$: the maximal cumulative reward the agent can expect to receive when starting at state s.

Optimal State-Action Value Function $Q^*(s,a)$

- Definition: The optimal state-action value function $Q^*(s,a)$ is the expected return for taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$ and then following an optimal policy,

$$Q^*(s,a) = E[r(s,a)] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,a] V^*(s').$$

- Property: The following equalities hold:

$$\forall\ s \in \mathcal{S},\ V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a).$$

  - The knowledge of the optimal policy value vector $\boldsymbol{V}^*$ is equivalent to the knowledge of the optimal state-action value function Q^*.

- Consequence: $\forall\ s \in \mathcal{S},\ \pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a)$.
  - The knowledge of the optimal state-action value function Q^* is sufficient for the agent to determine the optimal policy π^*, without any direct knowledge of the reward or transition probabilities.

Nonlinear Bellman Equations

The system of nonlinear equations for the optimal policy values $V^*(s)$ is:

$$\forall\ s \in \mathcal{S},\ V^*(s) = \max_{a \in \mathcal{A}} \left\{ E[r(s,a)] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,a] V^*(s') \right\}.$$

The Contents of This Lecture

- Markov decision processes (MDP) model of the environment
- Action policy
- Planning algorithms
- Learning algorithms

Known Model

- Setting: The environment model of an MDP is known.
- Problem: Find an optimal policy for the infinite horizon.
- Algorithms:
  - Value iteration.
  - Policy iteration.
  - Linear programming.

Value Iteration Algorithm

$\textsc{ValueIteration}(\boldsymbol{V}_0)$

1. $\boldsymbol{V} \leftarrow \boldsymbol{V}_0$ $\qquad \triangleright \boldsymbol{V}_0$ arbitrary value column vector
2. **while** $\|\boldsymbol{V} - \boldsymbol{\Phi}(\boldsymbol{V})\| \geq \frac{(1-\gamma)\epsilon}{\gamma}$ **do**
3. $\qquad \boldsymbol{V} \leftarrow \boldsymbol{\Phi}(\boldsymbol{V})$
4. **return** $\boldsymbol{\Phi}(\boldsymbol{V})$

Value Iteration Function

- Based on the nonlinear Bellman equations.
- $\boldsymbol{\Phi} : \mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{S}|}$: the iteration function which generates a new policy value column vector $\boldsymbol{\Phi}(\boldsymbol{V})$ from an old policy value column vector $\boldsymbol{V}$ by the nonlinear Bellman equations: $\forall\ s \in \mathcal{S}$,

$$\begin{aligned}
\boldsymbol{\Phi}(\boldsymbol{V})(s) &= \max_{a \in \mathcal{A}} \left\{ E[r(s,a)] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,a] V(s') \right\} \\
&= \max_{\pi} \left\{ E[r(s,\pi(s))] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,\pi(s)] V(s') \right\}.
\end{aligned}$$

- In vector form, $\boldsymbol{\Phi}(\boldsymbol{V}) = \max_{\pi} \{\boldsymbol{R}_{\pi} + \gamma \boldsymbol{P}_{\pi} \boldsymbol{V}\}$.

Value Iteration Algorithm - Convergence

Theorem 14.2: For any initial value $\boldsymbol{V}_0$, the sequence defined by $\boldsymbol{V}_{n+1} = \boldsymbol{\Phi}(\boldsymbol{V}_n)$ converges to V^*.

- Based on the γ-contraction property:

$$\|\boldsymbol{\Phi}(\boldsymbol{V}) - \boldsymbol{\Phi}(\boldsymbol{U})\|_\infty \leq \gamma \|\boldsymbol{V} - \boldsymbol{U}\|_\infty \; \forall \; \boldsymbol{V}, \boldsymbol{U} \in \mathbb{R}^{|\mathcal{S}|}.$$

- V^* is a fixed point of the iteration function $\boldsymbol{\Phi}$, i.e.,

$$\boldsymbol{\Phi}(V^*) = V^*.$$

ϵ-Optimality

- $\boldsymbol{V}_{n+1}$: the value column vector returned from the value iteration algorithm.
- $\|\boldsymbol{V}_{n+1} - \boldsymbol{V}_n\|_\infty < \frac{(1-\gamma)\epsilon}{\gamma}$.

Since

$$\begin{aligned}\|\boldsymbol{V}^* - \boldsymbol{V}_{n+1}\|_\infty &\leq \|\boldsymbol{\Phi}(\boldsymbol{V}^*) - \boldsymbol{\Phi}(\boldsymbol{V}_{n+1})\|_\infty + \|\boldsymbol{\Phi}(\boldsymbol{V}_{n+1}) - \boldsymbol{\Phi}(\boldsymbol{V}_n)\|_\infty \\ &\leq \gamma\|\boldsymbol{V}^* - \boldsymbol{V}_{n+1}\|_\infty + \gamma\|\boldsymbol{V}_{n+1} - \boldsymbol{V}_n\|_\infty,\end{aligned}$$

we have

$$(1-\gamma)\|\boldsymbol{V}^* - \boldsymbol{V}_{n+1}\|_\infty \leq \gamma\|\boldsymbol{V}_{n+1} - \boldsymbol{V}_n\|_\infty$$

so that

$$\|\boldsymbol{V}^* - \boldsymbol{V}_{n+1}\|_\infty \leq \frac{\gamma}{1-\gamma}\|\boldsymbol{V}_{n+1} - \boldsymbol{V}_n\|_\infty < \epsilon.$$

Complexity

Since

$$\|\boldsymbol{V}_{n+1} - \boldsymbol{V}_n\|_\infty \leq \gamma \|\boldsymbol{V}_n - \boldsymbol{V}_{n-1}\|_\infty \leq \gamma^n \|\boldsymbol{V}_1 - \boldsymbol{V}_0\|_\infty,$$

we have

$$\|\boldsymbol{V}_{n+1} - \boldsymbol{V}_n\|_\infty < \frac{(1-\gamma)\epsilon}{\gamma}$$

if

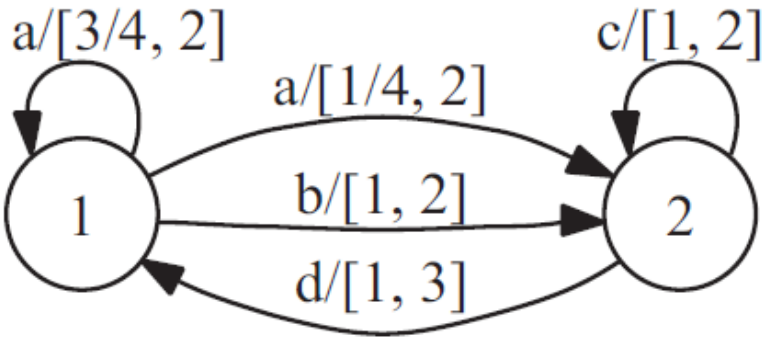
$$\gamma^n \|\boldsymbol{V}_1 - \boldsymbol{V}_0\|_\infty < \frac{(1-\gamma)\epsilon}{\gamma}$$

which implies that

$$n > \frac{\ln \frac{1}{\epsilon} + \ln(\frac{\gamma \|\boldsymbol{V}_1 - \boldsymbol{V}_0\|_\infty}{1-\gamma})}{\ln \frac{1}{\gamma}}.$$

- $n = O(\ln \frac{1}{\epsilon})$.

Value Iteration Algorithm - An Example



The environment model of an MDP has

- $\mathcal{S} = \{1, 2\}$;
- $\mathcal{A} = \{a, b, c, d\}$;
- transition probabilities: $p[1|1, a] = 3/4, p[2|1, a] = 1/4$, $p[2|1, b] = 1$, $p[2|2, c] = 1$, $p[1|2, d] = 1$ ($Pr[S_t = 1, A_t = c] = Pr[S_t = 1, A_t = d] = Pr[S_t = 2, A_t = a] = Pr[S_t = 2, A_t = b] = 0$);

- a deterministic reward function $r(s,a)$:
  $r(1,a)=2, r(1,b)=2, r(2,c)=2, r(2,d)=3$, and all others are zeros.

The iteration function for this MDP is

$$\begin{aligned}
V_{n+1}(1) &= \max\{r(1,a)+\gamma(p[1|1,a]V_n(1)+p[2|1,a]V_n(2)),\\
&\qquad r(1,b)+\gamma p[2|1,b]V_n(2)\}\\
&= \max\{2+\gamma((3/4)V_n(1)+(1/4)V_n(2)), 2+\gamma V_n(2)\},\\
V_{n+1}(2) &= \max\{r(2,c)+\gamma p[2|2,c]V_n(2), r(2,d)+\gamma p[1|2,d]V_n(1)\}\\
&= \max\{2+\gamma V_n(2), 3+\gamma V_n(1)\}.
\end{aligned}$$

- With $V_0(1)=V_0(2)=0$ and $\gamma=1/2$, we obtain $V_1(1)=2$, $V_1(2)=3$ and $V_1(1)=7/2$, $V_1(2)=4$.
- The algorithm quickly converges to the optimal values $V^*(1)=14/3$ and $V^*(2)=16/3$.

Policy Iteration Algorithm

PolicyIteration(π_0)

1. $\pi \leftarrow \pi_0$ $\quad \triangleright$ π_0 arbitrary policy
2. $\pi' \leftarrow$ NIL
3. **while** $\pi \neq \pi'$ **do**
4. $\quad \boldsymbol{V} \leftarrow \boldsymbol{V}_\pi$ $\quad \triangleright$ policy evaluation: solve $(\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)\boldsymbol{V} = \boldsymbol{R}_\pi$
5. $\quad \pi' \leftarrow \pi$
6. $\quad \pi \leftarrow \Pi(\boldsymbol{V}) = \arg\max_\pi \{\boldsymbol{R}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{V}\}$ $\quad \triangleright$ greedy policy improvement
7. **return** π

Policy Generation Function

- Based on the nonlinear Bellman equations.
- $\mathscr{P}$: the set of all possible policies $\pi : \mathcal{S} \to \mathcal{A}$.
- $\Pi : \mathbb{R}^{|\mathcal{S}|} \to \mathscr{P}$: the policy generation function which generates a new policy $\Pi(\boldsymbol{V})$ from an old policy value column vector $\boldsymbol{V}$ by the nonlinear Bellman equations:

$$\begin{aligned}\Pi(\boldsymbol{V}) &\triangleq \arg\max_{\pi} \left\{ E[r(s,\pi(s))] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,\pi(s)] V(s') \right\}_{s \in \mathcal{S}} \\ &= \arg\max_{\pi} \left\{ \boldsymbol{R}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{V} \right\}.\end{aligned}$$

Policy Iteration Algorithm - Convergence

Theorem 14.3: Let $\{\boldsymbol{V}_n\}_{n=0}^{\infty}$ be the sequence of policy value vectors generated by the policy iteration algorithm. Then for any $n \in \mathbb{N}_0$, the following inequalities hold:

$$\boldsymbol{V}_n \leq \boldsymbol{V}_{n+1} \leq \boldsymbol{V}^*.$$

Proof.

- Let π_n be the policy improvement in the nth iteration of the algorithm and $\boldsymbol{V}_n$ the corresponding policy value vector.
- We first show that $(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_n})^{-1}$ preserves ordering, that is, for any column matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ in $\mathbb{R}^{|\mathcal{S}|}$, if $\boldsymbol{X} - \boldsymbol{Y} \geq \boldsymbol{0}$, then $(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_n})^{-1}(\boldsymbol{X} - \boldsymbol{Y}) \geq \boldsymbol{0}$. Since $\|\gamma \boldsymbol{P}_{\pi_n}\|_\infty < 1$ as shown in Theorem 14.2, $(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_n})$ is invertible and its inverse is

$$(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_n})^{-1} = \sum_{j=0}^{\infty} \gamma^j \boldsymbol{P}_{\pi_n}^j .$$

  Since entries of $\boldsymbol{P}_{\pi_n}$ are nonnegative, entries of $(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_n})^{-1}$ are nonnegative so that entries of $(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_n})^{-1}(\boldsymbol{X} - \boldsymbol{Y})$ are nonnegative.

- By the definition of π_{n+1}, we have

$$\boldsymbol{R}_{\pi_{n+1}} + \gamma \boldsymbol{P}_{\pi_{n+1}} \boldsymbol{V}_n \geq \boldsymbol{R}_{\pi_n} + \gamma \boldsymbol{P}_{\pi_n} \boldsymbol{V}_n = \boldsymbol{V}_n,$$

  which shows that

$$\boldsymbol{R}_{\pi_{n+1}} \geq (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{n+1}}) \boldsymbol{V}_n.$$

- Since $(\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{n+1}})^{-1}$ preserves order, we have

$$\begin{aligned} \boldsymbol{V}_{n+1} &= (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{n+1}})^{-1} \boldsymbol{R}_{\pi_{n+1}} \\ &\geq (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{n+1}})^{-1} (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{n+1}}) \boldsymbol{V}_n \\ &= \boldsymbol{V}_n. \end{aligned}$$

□

Relation Between Value Iteration and Policy Iteration Algorithms

Theorem 14.4:

- $\{\boldsymbol{U}_n\}_{n=0}^{\infty}$: the sequence of policy value vectors generated by the value iteration algorithm;
- $\{\boldsymbol{V}_n\}_{n=0}^{\infty}$: the sequence of policy value vectors generated by the policy iteration algorithm.

If $\boldsymbol{U}_0 = \boldsymbol{V}_0$, then

$$\boldsymbol{U}_n \leq \boldsymbol{V}_n \leq \boldsymbol{V}^* \ \forall \ n \geq 1.$$

Proof.

- Let π_0 be the initial policy of the policy iteration algorithm and $\boldsymbol{V}_0$ is the corresponding policy value vector.
- Let $\boldsymbol{U}_0 = \boldsymbol{V}_0$ be the initial policy value vector of the value iteration algorithm.
- We first show that the iteration function $\boldsymbol{\Phi}$ is monotonic. Let $\boldsymbol{U}$ and $\boldsymbol{V}$ in $\mathbb{R}^{|\mathcal{S}|}$ be such that $\boldsymbol{U} \leq \boldsymbol{V}$, i.e., $\boldsymbol{U} - \boldsymbol{V} \leq 0$. Since

$$\begin{aligned}
\boldsymbol{\Phi}(\boldsymbol{U}) &= \max_{\pi}\{\boldsymbol{R}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{U}\} \\
&= \boldsymbol{R}_{\pi'} + \gamma \boldsymbol{P}_{\pi'} \boldsymbol{U} \text{ for some policy } \pi' \\
&\leq \boldsymbol{R}_{\pi'} + \gamma \boldsymbol{P}_{\pi'} \boldsymbol{V} \text{ since } \boldsymbol{P}_{\pi'}(\boldsymbol{U} - \boldsymbol{V}) \leq \boldsymbol{0} \\
&= \max_{\pi}\{\boldsymbol{R}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{V}\} \\
&= \boldsymbol{\Phi}(\boldsymbol{V}).
\end{aligned}$$

- We complete the proof by induction on n. We have assumed that $\boldsymbol{U}_0 \leq \boldsymbol{V}_0$. Assume that $\boldsymbol{U}_n \leq \boldsymbol{V}_n$ for some $n \geq 0$. Then we have

$$
\begin{aligned}
\boldsymbol{U}_{n+1} &= \boldsymbol{\Phi}(\boldsymbol{U}_n) \text{ by the value iteration algorithm} \\
&\leq \boldsymbol{\Phi}(\boldsymbol{V}_n) \text{ by the monotonicity of } \boldsymbol{\Phi} \\
&= \max_{\pi}\{\boldsymbol{R}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{V}_n\} \\
&= \boldsymbol{R}_{\pi_{n+1}} + \gamma \boldsymbol{P}_{\pi_{n+1}} \boldsymbol{V}_n \text{ by the definition of } \pi_{n+1} \\
&\leq \boldsymbol{R}_{\pi_{n+1}} + \gamma \boldsymbol{P}_{\pi_{n+1}} \boldsymbol{V}_{n+1} \text{ since } \boldsymbol{P}_{\pi_{n+1}}(\boldsymbol{V}_n - \boldsymbol{V}_{n+1}) \leq \boldsymbol{0} \\
&\quad \text{by Theorem 14.3} \\
&= \boldsymbol{V}_{n+1} \text{ by linear Bellman equations.}
\end{aligned}
$$

□

Complexity of Policy Iteration Algorithm

- Theorem 14.4 shows that the policy iteration algorithm converges in a smaller number of iterations than the value iteration algorithm due to the optimal policy.
- But, each iteration of the policy iteration algorithm requires computing a policy value, that is, solving a system of linear equations, which is more expensive than the computation in an iteration of the value iteration algorithm.

Reformulation of the System of Nonlinear Bellman Equations as a Linear Programming Problem

The system of nonlinear Bellman equations:

$$\forall\ s \in \mathcal{S},\ V^*(s) = \max_{a \in \mathcal{A}} \left\{ E[r(s,a)] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,a] V^*(s') \right\}.$$

is equivalent to the following primal optimization problem:

$$\begin{aligned} \text{Minimize} \quad & F(\boldsymbol{V}) = \textstyle\sum_{s \in \mathcal{S}} \alpha(s) V(s) \\ \text{Subject to} \quad & E[r(s,a)] + \gamma \textstyle\sum_{s' \in \mathcal{S}} Pr[s'|s,a] V(s') - V(s) \leq 0, \\ & \qquad\qquad \forall\ s \in \mathcal{S},\ \forall\ a \in \mathcal{A}, \\ & \boldsymbol{V} \in \mathbb{R}^{|\mathcal{S}|}, \end{aligned}$$

where $\alpha(s), s \in \mathcal{S}$, are any fixed positive values.

Comments

- This primal optimization problem is a linear programming (LP) problem.
- There exist a variety of different methods for solving relative large LPs in practice, using the simplex method, interior-point methods, or a variety of special-purpose solutions.
- The number of rows of the constraint matrix inequality in this LP is $|\mathcal{S}||\mathcal{A}|$ and its number of columns $|\mathcal{S}|$.
- The complexity of the solution techniques for LPs is typically more favorable in terms of the number of rows than the number of columns.
- This motivate a solution based on the equivalent dual problem of this LP.

Lagrangian Dual Function

- Lagrangian function: for all $\boldsymbol{V} \in \mathbb{R}^{|\mathcal{S}|}$ and $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$,

$$
\begin{aligned}
& L(\boldsymbol{V}, \boldsymbol{\lambda}) \\
= \; & F(\boldsymbol{V}) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \lambda(s,a) g_{s,a}(\boldsymbol{V}) \\
= \; & \sum_{s \in \mathcal{S}} \alpha(s) V(s) + \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \lambda(s,a) \Big(E[r(s,a)] + \\
& \qquad \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,a] V(s') - V(s) \Big) \\
= \; & \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \lambda(s,a) E[r(s,a)] + \sum_{s \in \mathcal{S}} \Big(\alpha(s) + \\
& \gamma \sum_{s' \in \mathcal{S}, a \in \mathcal{A}} Pr[s|s',a] \lambda(s',a) - \sum_{a \in \mathcal{A}} \lambda(s,a) \Big) V(s).
\end{aligned}
$$

- Lagrangian dual function: for all $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$,

$$\begin{aligned} & \theta(\boldsymbol{\lambda}) \\ \triangleq & \inf_{\boldsymbol{V} \in \mathbb{R}^{|\mathcal{S}|}} L(\boldsymbol{V}, \boldsymbol{\lambda}) \\ = & \begin{cases} \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \lambda(s,a) E[r(s,a)], \\ \text{if } \alpha(s) + \gamma \sum_{s' \in \mathcal{S}, a \in \mathcal{A}} Pr[s|s',a] \lambda(s',a) - \sum_{a \in \mathcal{A}} \lambda(s,a) = 0, \\ -\infty, \text{ otherwise.} \end{cases} \end{aligned}$$

Lagrangian Dual Problem

$$\begin{aligned}
\text{Maximize} \quad & \theta(\boldsymbol{\lambda}) = \textstyle\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} E[r(s,a)]\lambda(s,a) \\
\text{Subject to} \quad & \lambda(s,a) \geq 0 \;\forall\; s \in \mathcal{S},\; \forall\; a \in \mathcal{A}, \\
& \textstyle\sum_{a\in\mathcal{A}} \lambda(s,a) - \alpha(s) - \gamma \sum_{s'\in\mathcal{S},a\in\mathcal{A}} Pr[s|s',a]\lambda(s',a) = 0, \\
& \qquad\qquad\qquad\qquad\qquad\qquad s \in \mathcal{S}, \\
& \boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}.
\end{aligned}$$

Comments

- By setting $\sum_{s \in \mathcal{S}} \alpha(s) = 1$, $\alpha(s)$ can be interpreted as probabilities.
- The Lagrange variables $\lambda(s, a)$ in the dual problem can be interpreted as the probability of being in state s and taking action a.

The Contents of This Lecture

- Markov decision processes (MDP) model of the environment
- Action policy
- Planning algorithms
- Learning algorithms

Problem

- Unknown environment model:
  - Transition and reward probabilities not known.
  - Realistic scenario in many practical problems.
- Training information: a sequence of immediate rewards based on actions taken.
- Learning approaches:
  - Model-free: learn policy directly.
  - Model-based: learn model, use it to learn policy.
- How do we estimate reward and transition probabilities?
  - Use equations derived for policy value and Q-functions.
  - But those equations are given in terms of some expectations.
- An instance of a stochastic approximation problem.

Stochastic Approximation

- Problem: Find a solution of $\boldsymbol{x} = \boldsymbol{H}(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathbb{R}^N$ while
  - $\boldsymbol{H}$ cannot be computed, e.g., not accessible;
  - A sequence of random noisy observations $\boldsymbol{H}(\boldsymbol{X}_t) + \boldsymbol{W}_{t+1}$, $t \geq 0$, is available with $E[\boldsymbol{W}_{t+1}] = \boldsymbol{0}$ for all $t \geq 0$.
- Idea: an algorithm based on iterative technique:

$$\begin{aligned} \boldsymbol{X}_{t+1} &= (1 - \boldsymbol{\alpha}_t) \odot \boldsymbol{X}_t + \boldsymbol{\alpha}_t \odot (\boldsymbol{H}(\boldsymbol{X}_t) + \boldsymbol{W}_{t+1}) \\ &= \boldsymbol{X}_t + \boldsymbol{\alpha}_t \odot (\boldsymbol{H}(\boldsymbol{X}_t) - \boldsymbol{X}_t + \boldsymbol{W}_{t+1}), \end{aligned}$$

  more generally,

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \boldsymbol{\alpha}_t \odot \boldsymbol{D}(\boldsymbol{X}_t, \boldsymbol{W}_{t+1}).$$

  - $[a_1, a_2, \ldots, a_N]^T \odot [b_1, b_2, \ldots, b_N]^T \triangleq [a_1 b_1, a_2 b_2, \ldots, a_N b_N]^T$.

Strong Law of Large Numbers

Let $X_1, X_2, \ldots$ be an infinite sequence of i.i.d. random variables with finite mean $E[X]$. Then we have

$$\lim_{n\to\infty} \frac{X_1 + X_2 + \cdots + X_n}{n} = E[X] \text{ w.p.1.}$$

- The arithmetic mean $\frac{X_1+X_2+\cdots+X_n}{n}$ can be rewritten as

$$\begin{aligned} & \frac{X_1 + X_2 + \cdots + X_n}{n} \\ = \quad & \frac{n-1}{n} \frac{X_1 + X_2 + \cdots + X_{n-1}}{n-1} + \frac{X_n}{n} \\ = \quad & \left(1 - \frac{1}{n}\right) \frac{X_1 + X_2 + \cdots + X_{n-1}}{n-1} + \frac{1}{n} X_n. \end{aligned}$$

- Let $\mu_0 \triangleq 0$ and $\mu_n \triangleq \frac{X_1+X_2+\cdots+X_n}{n},\ n \geq 1$, be the arithmetic mean of the first n r.v.s $X_1, X_2, \ldots, X_n$ and let $\alpha_{n-1} = 1/n$.

Then we have

$$\mu_n = (1 - \alpha_{n-1})\mu_{n-1} + \alpha_{n-1} X_n$$

and

$$\lim_{n \to \infty} \mu_n = E[X] \text{ w.p.1.}$$

- Note that $\alpha_n \in [0, 1]$ for all $n \geq 0$, $\sum_{n=0} \alpha_n = \infty$, and $\sum_{n=0} \alpha_n^2 < \infty$.
- The next theorem is a generalization of the strong law of large numbers for bounded random variables.

Mean Estimation

Theorem 14.5: Let

- $X_1, X_2, \ldots$: an infinite sequence of i.i.d. random variables taking value in $[a, b]$, $-\infty < a < b < \infty$.
- $\alpha_0, \alpha_1, \ldots$: an infinite sequence of real numbers in $[0, 1]$ such that $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

Define an infinite sequence $\mu_0, \mu_1, \ldots$ of random variables such that $\mu_0 \triangleq 0$ and

$$\mu_k \triangleq (1 - \alpha_{k-1})\mu_{k-1} + \alpha_{k-1} X_k, \ \forall\ k \geq 1.$$

Then we have

$$\lim_{k \to \infty} \mu_k = E[X] \text{ w.p.1.}$$

Proof. This is a special case of Theorem 14.7. □

Supermartingale Convergence Theorem

Theorem 14.6: Let $\{X_t,\ t \geq 0\}, \{Y_t,\ t \geq 0\}, \{Z_t,\ t \geq 0\}$ be three sequences of nonnegative random variables such that each of them is adapted to a filtration $\{\mathcal{F}_t,\ t \geq 0\}$, i.e., $\mathcal{F}(X_t) \subseteq \mathcal{F}_t,\ \forall\ t \geq 0$, $\mathcal{F}(Y_t) \subseteq \mathcal{F}_t,\ \forall\ t \geq 0$, and $\mathcal{F}(Z_t) \subseteq \mathcal{F}_t,\ \forall\ t \geq 0$. Assume that $\sum_{t=0}^{\infty} Y_t < \infty$. Then if

$$E[X_{t+1} \mid \mathcal{F}_t] \leq X_t + Y_t - Z_t \ \forall\ t \geq 0,$$

the following holds:

- As t goes to infinity, X_t converges to a limit w.p.1.
- $\sum_{t=0}^{\infty} Z_t < \infty$ w.p.1.

Comments

- A filtration $\{\mathcal{F}_t,\ t \geq 0\}$ is a non-decreasing sequence of σ-algebras on a sample space Ω, i.e. $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$.
- It clear that $\mathcal{F}_t$ contains the entire history of the three processes $\{X_t,\ t \geq 0\}, \{Y_t,\ t \geq 0\}, \{Z_t,\ t \geq 0\}$ up to time t for all $t \geq 0$.

Convergence of Stochastic Approximation I

Theorem 14.7: Let

- $\boldsymbol{D} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$: a measurable mapping.
- $\{\boldsymbol{W}_t,\ t \geq 1\}$: a sequence of random vectors in $\mathbb{R}^N$.
- $\{\boldsymbol{\alpha}_t,\ t \geq 0\}$: a sequences of random vectors in $\mathbb{R}^N$ such that

$$\sum_{t=0}^{\infty} \|\boldsymbol{\alpha}_t\|_2 = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \|\boldsymbol{\alpha}_t\|_2^2 < \infty.$$

- $\{\boldsymbol{X}_t,\ t \geq 0\}$: a sequence of random vectors in $\mathbb{R}^N$ generated by an initial random vector $\boldsymbol{X}_0$ and the two processes $\{\boldsymbol{W}_t,\ t \geq 1\}$ and $\{\boldsymbol{\alpha}_t,\ t \geq 0\}$ with a recursive formula:

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \boldsymbol{\alpha}_t \odot \boldsymbol{D}(\boldsymbol{X}_t, \boldsymbol{W}_{t+1}) \ \forall\ t \geq 0.$$

- $\mathcal{F}_t = \mathcal{F}(\boldsymbol{X}_0, \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_i, \boldsymbol{W}_i, 1 \leq i \leq t)$: the entire history of the stochastic dynamic system up to time t, $\forall\ t \geq 0$.
- $\Psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2$ for all $\boldsymbol{x} \in \mathbb{R}^N$ and for some fixed $\boldsymbol{x}^* \in \mathbb{R}^N$.

Assume that

- $\exists\ K_1 \geq 0, K_2 \in \mathbb{R}$:

$$E[\|\boldsymbol{\alpha}_t \odot \boldsymbol{D}(\boldsymbol{X}_t, \boldsymbol{W}_{t+1})\|_2^2 \mid \mathcal{F}_t] \leq \|\boldsymbol{\alpha}_t\|_2^2(K_1 + K_2\Psi(\boldsymbol{X}_t));$$

- $\exists\, c > 0 : \nabla\Phi(\boldsymbol{X}_t)^T E[\boldsymbol{\alpha}_t \odot \boldsymbol{D}(\boldsymbol{X}_t, \boldsymbol{W}_{t+1}) \mid \mathcal{F}_t] \leq -c\|\boldsymbol{\alpha}_t\|_2\Psi(\boldsymbol{X}_t)$.

Then the random sequence $\{\boldsymbol{X}_t, t \geq 0\}$ converges almost surely to $\boldsymbol{x}^*$, i.e.,

$$\lim_{t\to\infty} \boldsymbol{X}_t = \boldsymbol{x}^* \quad \text{w.p.1.}$$

Proof. Since the function $\Psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2$ is quadratic, a Taylor expansion of $\Psi(\boldsymbol{x})$ at $\boldsymbol{x} = \boldsymbol{X}_t$ gives

$$\begin{aligned}\Psi(\boldsymbol{X}_{t+1}) &= \Psi(\boldsymbol{X}_t) + \nabla\Phi(\boldsymbol{X}_t)^T(\boldsymbol{X}_{t+1} - \boldsymbol{X}_t) \\ &\quad + \frac{1}{2}(\boldsymbol{X}_{t+1} - \boldsymbol{X}_t)^T\nabla^2\Phi(\boldsymbol{X}_t)(\boldsymbol{X}_{t+1} - \boldsymbol{X}_t).\end{aligned}$$

Since $\nabla^2\Phi(\boldsymbol{x}) = \boldsymbol{I}$, the identity matrix, $\forall\ \boldsymbol{x} \in \mathbb{R}^N$, we have

$$\begin{aligned}E[\Psi(\boldsymbol{X}_{t+1}) \mid \mathcal{F}_t] &= \Psi(\boldsymbol{X}_t) + \nabla\Phi(\boldsymbol{X}_t)^T E[(\boldsymbol{X}_{t+1} - \boldsymbol{X}_t) \mid \mathcal{F}_t] \\ &\quad + \frac{1}{2}E[\|\boldsymbol{X}_{t+1} - \boldsymbol{X}_t\|_2^2 \mid \mathcal{F}_t]\end{aligned}$$

Since

$$\boldsymbol{X}_{t+1} - \boldsymbol{X}_t = \boldsymbol{\alpha}_t \odot \boldsymbol{D}(\boldsymbol{X}_t, \boldsymbol{W}_{t+1}),$$

we have

$$\begin{aligned} & E[\Psi(\boldsymbol{X}_{t+1}) \mid \mathcal{F}_t] \\ \leq \quad & \Psi(\boldsymbol{X}_t) - c\|\boldsymbol{\alpha}_t\|_2 \Psi(\boldsymbol{X}_t) + \frac{\|\boldsymbol{\alpha}_t\|_2^2}{2}(K_1 + K_2 \Psi(\boldsymbol{X}_t)) \\ = \quad & \Psi(\boldsymbol{X}_t) + \frac{\|\boldsymbol{\alpha}_t\|_2^2 K_1}{2} - \left(c - \frac{\|\boldsymbol{\alpha}_t\|_2 K_2}{2}\right) \|\boldsymbol{\alpha}_t\|_2 \Psi(\boldsymbol{X}_t). \end{aligned}$$

- $\{\Psi(\boldsymbol{X}_t),\ t \geq 0\}$ is a sequence of nonnegative random variables;
- $\{\frac{\|\boldsymbol{\alpha}_t\|_2^2 K_1}{2},\ t \geq 0\}$ is a sequence of nonnegative random variables such that $\sum_{t=0}^{\infty} \frac{\|\boldsymbol{\alpha}_t\|_2^2 K_1}{2} = \frac{K_1}{2} \sum_{t=0}^{\infty} \|\boldsymbol{\alpha}_t\|_2^2 < \infty$;
- Since $\sum_{t=0}^{\infty} \|\boldsymbol{\alpha}_t\|_2^2 < \infty$, we have $\lim_{t\to\infty} \|\boldsymbol{\alpha}_t\|_2^2 = 0$ and then $\lim_{t\to\infty} \|\boldsymbol{\alpha}_t\|_2 = 0$ so that $c - \frac{\|\boldsymbol{\alpha}_t\|_2 K_2}{2} > 0\ \forall\ t \geq t_0$ for a sufficiently large t_0. Thus, $\{(c - \frac{\|\boldsymbol{\alpha}_t\|_2 K_2}{2})\|\boldsymbol{\alpha}_t\|_2 \Psi(\boldsymbol{X}_t),\ t \geq t_0\}$ is a sequence of nonnegative random variables.

It is clear that the three random sequences $\{\Psi(\boldsymbol{X}_t),\ t \geq 0\}$, $\{\frac{\|\boldsymbol{\alpha}_t\|_2^2 K_1}{2},\ t \geq 0\}$, and $\{(c - \frac{\|\boldsymbol{\alpha}_t\|_2 K_2}{2})\|\boldsymbol{\alpha}_t\|_2 \Psi(\boldsymbol{X}_t),\ t \geq 0\}$ are adapted to the filtration $\{\mathcal{F}_t,\ t \geq 0\}$. By the supermartingale convergence theorem, $\Psi(\boldsymbol{X}_t)$ converges to a limit as $t \to \infty$ w.p.1 and

$$\sum_{t=0}^{\infty}(c - \frac{\|\boldsymbol{\alpha}_t\|_2 K_2}{2})\|\boldsymbol{\alpha}_t\|_2 \Psi(\boldsymbol{X}_t) < \infty \text{ w.p.1.}$$

Since $\Psi(\boldsymbol{X}_t)$ converges to a limit w.p.1, it is bounded w.p.1 and then $\sum_{t=0}^{\infty} \frac{\|\boldsymbol{\alpha}_t\|_2^2 K_2}{2} \Psi(\boldsymbol{X}_t) < \infty$. Since $\sum_{t=0}^{\infty} \|\boldsymbol{\alpha}_t\|_2 = \infty$, if $\Psi(\boldsymbol{X}_t)$ does not converge to 0, then the series $\sum_{t=0}^{\infty} \|\boldsymbol{\alpha}_t\|_2 \Psi(\boldsymbol{X}_t)$ will diverge to ∞, a contradiction. Thus we must have $\lim_{t\to\infty} \Psi(\boldsymbol{X}_t) = \lim_{t\to\infty} \frac{1}{2}\|\boldsymbol{X}_t - \boldsymbol{x}^*\|_2^2 = 0$ w.p.1 and then $\lim_{t\to\infty} \boldsymbol{X}_t = \boldsymbol{x}^*$ w.p.1. □

Convergence of Stochastic Approximation II

Theorem 14.8: Let

- $\boldsymbol{H} : \mathbb{R}^N \to \mathbb{R}^N$: a $\|\cdot\|_\infty$-contraction with the fixed point $\boldsymbol{x}^*$.
- $\{\boldsymbol{W}_t, t \geq 1\}$: a sequence of random vectors in $\mathbb{R}^N$.
- $\{\boldsymbol{\alpha}_t, t \geq 0\}$: a sequences of nonnegative random vectors such that

$$\sum_{t=0}^{\infty} \boldsymbol{\alpha}_t(s) = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \boldsymbol{\alpha}_t^2(s) < \infty \ \forall\ s \in [1, N].$$

- $\{\boldsymbol{X}_t, t \geq 0\}$: a sequence of random vectors in $\mathbb{R}^N$ generated by an initial random vector $\boldsymbol{X}_0$ and the two processes $\{\boldsymbol{W}_t, t \geq 1\}$ and $\{\boldsymbol{\alpha}_t, t \geq 0\}$ with a recursive formula:

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \boldsymbol{\alpha}_t \odot [\boldsymbol{H}(\boldsymbol{X}_t) - \boldsymbol{X}_t + \boldsymbol{W}_{t+1}] \ \forall\ t \geq 0.$$

- $\mathcal{F}_t = \mathcal{F}(\boldsymbol{X}_0, \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_i, \boldsymbol{W}_i, 1 \leq i \leq t),\ t \geq 0$: the entire history of the stochastic dynamic system up to time t.

Assume that

- $\exists\ K_1 > 0, K_2 \in \mathbb{R} : E[\boldsymbol{W}_{t+1}(s)^2 \mid \mathcal{F}_t] \leq K_1 + K_2 \|\boldsymbol{X}_t\|^2$ for some norm $\|\cdot\|$ for all $s \in [1, N]$;
- $E[\boldsymbol{W}_{t+1} \mid \mathcal{F}_t] = \mathbf{0}$;

Then the random sequence $\{\boldsymbol{X}_t, t \in \mathbb{N}\}$ converges almost surely to $\boldsymbol{x}^*$, i.e.,

$$\lim_{t \to \infty} \boldsymbol{X}_t = \boldsymbol{x}^* \quad \text{w.p.1.}$$

TD(0) Algorithm

TD(0)()

1. $\boldsymbol{V} \leftarrow \boldsymbol{V}_0$ $\quad \triangleright \boldsymbol{V}_0$ arbitrary initial value column vector
2. **for** $t \leftarrow 0$ **to** $T - 1$ **do**
3. $\quad s \leftarrow \textsc{SelectState}()$
4. $\quad$ **for** each step of epoch t **do**
5. $\quad\quad r' \leftarrow \textsc{Reward}(s, \pi(s))$
6. $\quad\quad s' \leftarrow \textsc{NextState}(\pi, s)$
7. $\quad\quad V(s) \leftarrow (1 - \alpha)V(s) + \alpha(r' + \gamma V(s'))$
8. $\quad\quad s \leftarrow s'$
9. **return** $\boldsymbol{V}$

Q-Learning Algorithm

Q-Learning()

1. $Q \leftarrow Q_0$ ▷ initialization, e.g., $Q_0 = 0$
2. **for** $t \leftarrow 0$ **to** $T - 1$ **do**
3. $s \leftarrow$ SelectState()
4. **for** each step of epoch t **do**
5. $a \leftarrow$ SelectAction(π, s) ▷ policy π derived from Q,
6. $r' \leftarrow$ Reward(s, a) e.g., ϵ-greedy
7. $s' \leftarrow$ NextState(s, a)
8. $Q(s, a) \leftarrow Q(s, a) + \alpha(r' + \gamma \max_{a'} Q(s', a') - Q(s, a))$
9. $s \leftarrow s'$
10. **return** Q

Optimal State-Action Value Function Q^* is a Fixed Point

Recall

- $Q^*(s,a) = E[r(s,a)] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,a] V^*(s')$ $\forall$ $(s,a) \in \mathcal{S} \times \mathcal{A}$;
- $V^*(s) = \max_{a \in \mathcal{A}} Q^*(s,a)$ $\forall$ $s \in \mathcal{S}$;

Thus we have

$$\begin{aligned} Q^*(s,a) &= E[r(s,a)] + \gamma \sum_{s' \in \mathcal{S}} Pr[s'|s,a] \max_{a' \in \mathcal{A}} Q^*(s',a') \\ &= \underset{Pr[r'|s,a]}{E}\Big[r'\Big] + \gamma \underset{Pr[s'|s,a]}{E}\Big[\max_{a' \in \mathcal{A}} Q^*(s',a')\Big]. \end{aligned}$$

- $\{Q^*(s,a), (s,a) \in \mathcal{S} \times \mathcal{A}\}$ is a fixed point of a mapping $\boldsymbol{H} : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \rightarrow \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ which is determined by the unknown state transition probabilities $Pr[s'|s,a]$ and the unknown reward probabilities $Pr[s'|s,a]$.

- Unknown environment model: Both transition probabilities $Pr[s'|s,a]$ and reward probabilities $Pr[r'|s,a]$ are unknown.
- Using stochastic approximation to learn (i.e. estimate) the optimal state-action value function $Q^*(s,a)$ in the case of an unknown environment model.

Formulation of Stochastic Approximation

- $\{S_t, t \geq 0\}$: the sequence of states returned by the environment.
- $\{A_t, t \geq 0\}$: the sequence of actions taken by the agent.
- $\{R_t, t \geq 1\}$: the sequence of reward returned by the environment.

The update rule is

$$Q_{t+1}(s,a) = (1-\alpha_t(s,a))Q_t(s,a)+\alpha_t(s,a)[R_{t+1}+\gamma \max_{a' \in \mathcal{A}} Q_t(S_{t+1}, a')]$$

Convergence of Q-Learning Algorithm

Theorem 14.9: Consider a finite MDP. Assume that the reward function $r(s,a)$ is deterministic and for any state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, $\alpha_t(s,a) \in [0,1]\ \forall\ t \geq 1$ and $\sum_{t=1}^{\infty} \alpha_t(s,a) = \infty$ and $\sum_{t=1}^{\infty} \alpha_t^2(s,a) < \infty$. Then the Q-learning algorithm converges to the optimal value Q^* with probability one.

- The conditions $\sum_{t=1}^{\infty} \alpha_t(s,a) = \infty$ on $\alpha_t(s,a)$ impose that each state-action pair (s,a) is visited infinitely many times.

Proof.

- For each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, let $\{Q_t(s, a), t \geq 1\}$ be the sequence of state-action value functions at the state-action pair (s, a) generated by the Q-learning algorithm.
- Given the (current) state $S_t = s$ and (current) action $A_t = a$ (taken), the returned (next) state S_{t+1} and reward $R_{t+1} = r(s, a)$ are used to update the old $Q_t(s, a)$ to the new $Q_{t+1}(s, a)$ as follows:

$$\begin{aligned} & Q_{t+1}(s, a) \\ = \quad & Q_t(s, a) + \alpha[r(s, a) + \gamma \max_{a' \in \mathcal{A}} Q_t(S_{t+1}, a') - Q_t(s, a)]. \end{aligned}$$

- With $S_{t+1} = s'$, the update rule can be rewritten as follows: for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$\begin{aligned} & Q_{t+1}(s,a) \\ = \; & Q_t(s,a) + \alpha_t(s,a)\Big[r(s,a) + \gamma \underset{Pr[s'|s,a]}{E}\Big[\max_{a' \in \mathcal{A}} Q^*(s',a')\Big] \\ & -Q_t(s,a) + \gamma\Big(\max_{a' \in \mathcal{A}} Q_t(S_{t+1},a') - \underset{Pr[s'|s,a]}{E}\Big[\max_{a' \in \mathcal{A}} Q^*(s',a')\Big]\Big)\Big], \end{aligned}$$

  where we define $\alpha_t(s,a) = 0$ if $(S_t, A_t) \neq (s,a)$ and $\alpha_t(s,a)$ otherwise.

- $\boldsymbol{Q}_t$: the vector with components $Q_t(s,a)$.

- $\boldsymbol{H}(\boldsymbol{Q}_t)$: the vector with components $\boldsymbol{H}(\boldsymbol{Q}_t)(s,a)$ defined as

$$\boldsymbol{H}(\boldsymbol{Q}_t)(s,a) = r(s,a) + \gamma \underset{Pr[s'|s,a]}{E}\Big[\max_{a' \in \mathcal{A}} Q^*(s',a')\Big].$$

- $\boldsymbol{W}_{t+1}$: the vector with component

□

Policy Used in the Q-Learning Algorithm

- The choice of the policy $\{\pi_t, t \geq 0\}$ in the Q-learning algorithm according to which an action a is selected (line 5) when the state at time T is s is not specified and, as already indicated, the theorem guarantees the convergence of the algorithm for an arbitrary policy so long as it ensures that every pair (s, a) is visited infinitely many times.
- Greedy policy: The action taken for a state s at time t is derived from the state-action value function Q_t at time t, i.e.,

$$\pi_t(s; Q_t) = \arg\max_{a \in \mathcal{A}} Q_t(s, a).$$

  - Greedy policy typically does not guarantee that all state-action pairs are visited for infinitely many times.

- ϵ-greedy policy: Take the greedy selection of an action at time t based on Q_t with probability $(1-\epsilon)$ and the random selection of an action from $\mathcal{A}$ with probability ϵ.
  - ϵ-greedy policy guarantees that all state-action pairs are visited for infinitely many times and thus is a standard choice in reinforcement learning.
- Boltzmann exploration: An action a is taken when the state at time t is s and the state-action value function is Q_t with probability

$$p_t(a \mid s, Q_t) = \frac{e^{\frac{Q_t(s,a)}{\tau_t}}}{\sum_{a' \in \mathcal{A}} e^{\frac{Q_t(s,a')}{\tau_t}}}.$$

  - τ_t is called the temperature at time t.
  - τ_t must be defined so that $\tau \to 0$ as $t \to \infty$, which ensures that for large values of t, the greedy action based on Q_t is selected.

- On the other hand, τ_t must be chosen so that it does not tend to 0 too fast to ensure that all actions are visited infinitely often.
  - It can be chosen, for instance, as $1/\log(n_t(s))$, where $n_t(s)$ is the number of times s has been visited up to time t.

State-Action-Reward-State-Action (SARSA) Algorithm

SARSA(π)

1. $Q \leftarrow Q_0$ ▷ initialization, e.g., $Q_0 = 0$
2. **for** $t \leftarrow 0$ **to** $T - 1$ **do**
3. $s \leftarrow \textsc{SelectState}()$
4. $a \leftarrow \textsc{SelectAction}(\pi(Q), s)$ ▷ policy π derived from Q, e.g., ϵ-greedy
5. **for** each step of epoch t **do**
6. $r' \leftarrow \textsc{Reward}(s, a)$
7. $s' \leftarrow \textsc{NextState}(s, a)$
8. $a' \leftarrow \textsc{SelectAction}(\pi(Q), s')$ ▷ policy π derived from Q, e.g., ϵ-greedy

9. $Q(s,a) \leftarrow Q(s,a) + \alpha_t(s,a)(r' + \gamma Q(s',a') - Q(s,a))$

10. $s \leftarrow s'$

11. $a \leftarrow a'$

12. **return** Q

TD(λ) Algorithm

TD(λ)()

$\boldsymbol{V} \leftarrow \boldsymbol{V}_0$ $\quad \triangleright \boldsymbol{V}_0$ arbitrary initial value column vector

$\boldsymbol{e} \leftarrow \mathbf{0}$

for $t \leftarrow 0$ **to** $T - 1$ **do**

    $s \leftarrow$ SelectState()

    for each step of epoch t **do**

        $s' \leftarrow$ NextState(π, s)

        $\delta \leftarrow r(s, \pi(s)) + \lambda V(s') - V(s)$

        $e(s) \leftarrow \lambda e(s) + 1$

9. **for** $u \in \mathcal{S}$ **do**
10. **if** $u \neq s$ **then**
11. $e(u) \leftarrow \gamma\lambda e(u)$
12. $V(u) \leftarrow V(u) + \alpha\delta e(u)$
13. $s \leftarrow s'$
14. **return** $\boldsymbol{V}$