Chapter 8 : value function approximation

- Introduction:

tabular representation

↓

function representation

1.

So far : State and action value are represented by tables. (离散)

- For example, action value: $q_\pi(S, a)$

	a_1	a_2	a_3	a_4	a_5
s_1	$q_\pi(s_1, a_1)$	$q_\pi(s_1, a_2)$	$q_\pi(s_1, a_3)$	$q_\pi(s_1, a_4)$	$q_\pi(s_1, a_5)$
⋮	⋮	⋮	⋮	⋮	⋮
s_9	$q_\pi(s_9, a_1)$	$q_\pi(s_9, a_2)$	$q_\pi(s_9, a_3)$	$q_\pi(s_9, a_4)$	$q_\pi(s_9, a_5)$

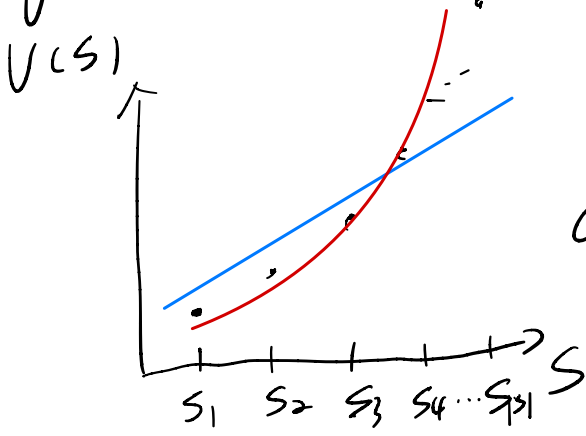
① Storage 存储能力.

② generalization ability 泛化能力

2. Consider Example.

One-dimensional state : $s_1, \dots s_{|S|}$

π is given: State value: $V_\pi(s_1), V_\pi(s_2) \cdots V_\pi(s_{|S|})$



hope use simple curve to approximate dot to save storage

Suppose straight: $V(s,w) = as+b = \underbrace{[s,1]}_{\phi^T(s)} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{w}$

$= \phi^T(s) \cdot w$

- w – parameter vector
- $\phi(s)$ – the feature vector of s
- $\hat{V}(s,w)$ is linear in w

广义线性模型

↓network

非线性系统.

$V_\pi(s) \approx \hat{V}(s,w)$ called value approximation

表格中. $S_1 \to S_2$ S_2 的 value 改变了
其余不变

函数中. $S_1 \to S_2$ S_2 的 value 改变了
对应 $V(S,w)$ 中 w 改变
会导致 S_1, S_3, value 都改变.

二. objective function

1. optimal w so that $V(S,w)$ can best approximate $V_\pi(S)$ for every S.

2. Objective function:

$$J(w) = E[(V_\pi(S) - \hat{V}(S,w))^2]$$

① uniform distribution.

every state not equal important

② Stationary distribution

long run behavior $\{d_\pi(S)\}_{S \in S}$

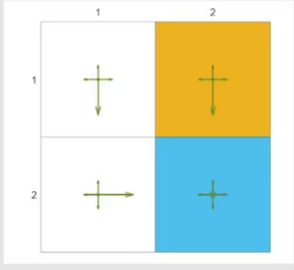
$$J(w) = \mathbb{E}\left[\left(V_\pi(S) - \hat{V}(S,w)\right)^2\right]$$

$$= \sum_{s \in \mathcal{S}} d_\pi(s)\left(V_\pi(s) - \hat{V}(s,w)\right)^2$$

Illustrative example:

- Given a policy shown in the figure.
- Let $n_\pi(s)$ denote the number of times that s has been visited in a very long episode generated by π.
- Then, $d_\pi(s)$ can be approximated by

$$d_\pi(s) \approx \frac{n_\pi(s)}{\sum_{s' \in \mathcal{S}} n_\pi(s')}$$



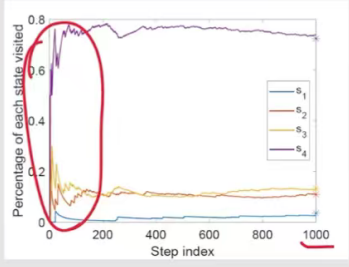


Figure: Long-run behavior of an ϵ-greedy policy with $\epsilon = 0.5$.

Zhao 18

$$d_\pi^T = d_\pi^T P_\pi$$

P_π: 状态转移矩阵. $P(s'|s)$

$$V_\pi = r_\pi + \gamma P_\pi V_\pi$$

根据π

$$P_\pi = \begin{bmatrix} 0.3 & 0.1 & 0.6 & 0 \\ 0.1 & 0.3 & 0 & 0.6 \\ 0.1 & 0 & 0.3 & 0.6 \\ 0 & 0.1 & 0.1 & 0.8 \end{bmatrix} \Rightarrow \text{代入 } d_\pi^T = d_\pi^T P_\pi$$

$$d_\pi^T = [0.0345, 0.1084, 0.1330, 0.2441]^T$$

3. Optimization algorithm:

gradient descent: $w_{k+1} = w_k - \alpha_k \nabla_w J(w_k)$

$$\nabla_w J(w_k) = \nabla_w E[(V_\pi(S) - \hat{V}(S,w))^2]$$

$$= E[\nabla_w (V_\pi(S) - \hat{V}(S,w))^2]$$

$$= 2E[(V_\pi(S) - \hat{V}_\pi(S,w))(-\nabla_w \hat{V}(S,w))]$$

Stochastic gradient → true

$$w_{t+1} = w_t + \alpha_t (\underline{V_t(S_t)} - \hat{V}(S_t, w_t)) \nabla_w \hat{V}(S_t, w_t)$$

replace $V_\pi(S_t)$

① Monte Carlo

$$w_{t+1} = w_t + \alpha_t (g_t - \hat{V}(S_t, w_t)) \nabla_w \hat{V}(S_t, w_t)$$

② TD learning:

$$w_{t+1} = w_t + \alpha_t [r_{t+1} + \gamma \hat{V}(S_{t+1}, w_t) - \hat{V}(S_t, w_t)] \nabla_w \hat{V}(S_t, w_t)$$

4. How to select function $\hat{v}(s,w)$

① $\hat{V}(s,w) = \underline{\phi}^T(s)\, w$

② network

5. In the linear case:

$$\hat{V}(s,w) = \underline{\phi}^T(s)\, w$$

We have $\nabla_w \hat{V}(s,w) = \underline{\phi}(s)$

$$w_{t+1} = w_t + \alpha_t [r_{t+1} + \gamma \hat{V}(s_{t+1}, w_t) - \hat{V}(s_t, w_t)] \nabla_w \hat{V}(s_t, w_t)$$

$$w_{t+1} = w_t + \alpha_t [r_{t+1} + \gamma \underline{\phi}^T(s_{t+1}) w_t - \underline{\phi}^T(s_t) w_t] \underline{\phi}(s_t)$$

TD-linear .

theoretical properties

6. illustrative example.

三. Sarsa with function approximation.

$$w_{t+1} = w_t + \alpha_t \left[r_{t+1} + \gamma \hat{q}(s_{t+1}, a_{t+1}, w_t) - \hat{q}(s_t, a_t, w_t) \right] \nabla_w \hat{q}(s_t, a_t, w_t)$$

四 Q-learning with function approximation

$$w_{t+1} = w_t + \alpha_t \left[r_{t+1} + \gamma \max_{a \in \mathcal{A}(s_{t+1})} \hat{q}(s_{t+1}, a, w_t) - \hat{q}(s_t, a_t, w_t) \right] \nabla_w \hat{q}(s_t, a_t, w_t)$$

五、Deep Q-learning

1. deep Q network.

$$w_{t+1} = w_t + \alpha_t [r_{t+1} + \gamma \max_{a \in A(s_{t+1})} \hat{q}(s_{t+1}, a, w_t) - \hat{q}(s_t, a_t, w_t)] \nabla_w \hat{q}(s_t, a_t, w_t)$$

$$J_w = E[(R + \gamma \max_{a \in A(S')} \hat{q}(S', a, w) - \hat{q}(S, A, w))^2]$$

2.

2 network:

① main network representing $\hat{q}(s, a, w)$

② target network $\hat{q}(s, a, w_T)$

$$J = E[(R + \gamma \max_{a \in A(S')} \hat{q}(S', a, w_T) - \hat{q}(S, A, w))^2]$$

3. Experience. reply; 经验回放.

$$B = \{(s, a, r, s')\}$$

随机一定是 uniform distribution. 再拿出来训练