Policy Evaluation

Wen Yu
Departamento de Control Automático
CINVESTAV-IPN

Calculation of Value Function

Value V: sum of future reward r under a particular policy π

$$V^{\pi} = \mathbb{E}_{\pi}\{G \mid s\} = \mathbb{E}_{\pi}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid s\right]$$
$$\approx \frac{1}{N}\sum\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots\right] \text{ for all } s$$

C1 C2 C3 Pass Sleep	*G1 = -2 + (-2)*1/2 + (-2)*1/4 + 10*1/8 + 0*1/16 = -2.25*
C1 FB FB C1 C2 Sleep	*G1 = -2 + (-1)*1/2 + (-1)*1/4 + (-2)*1/8 + (-2)*1/16 + 0*1/32 = -3.125*
C1 C2 C3 Pub C2 C3 Pass Sleep	*G1 = -2 + (-2)*1/2 + (-2)*1/4 + (1)*1/8 + (-2)*1/16 + ... = -3.41*
C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep	*G1 = -2 + (-1)*1/2 + (-1)*1/4 + (-2)*1/8 + (-2)*1/16 + (-2)*1/32 + ... = -3.20*

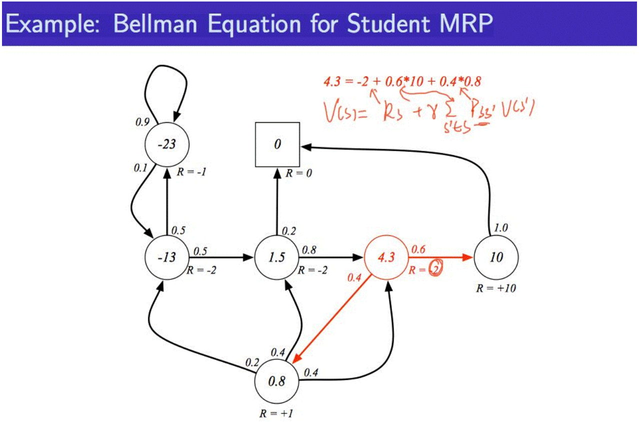
The value function needs "expected return" when evaluating the value of the state $C1$.

$$V(C1) = \frac{1}{4}\left[(-2.25 + (-3.125) + (-3.41) + (-3.20)]\right) = 2.996$$

Calculation of Value Function

Bellman equation is

$$V(s) = r(s) + \gamma \sum p(s' \mid s)\, V(s')$$



$$V(C3) = -2 + 0.6 * 10 + 0.4 * 0.8 = 4.3$$

Methods to Approximate Value Function

$$V(s) = r(s) + \gamma \sum p(s' \mid s)\, V(s')$$
$$V_k = r_k + \gamma \mathbb{E}\{V[x_{k+1}]\}$$

In reinforcement learning, there are three methods to calculate: $\mathbb{E}\{V[x_{k+1}]\}$ or $\gamma \sum p(s' \mid s)\, V(s')$

1. Dynamic programming
2. Monte Carlo
3. Temporal difference

Dynamic Programming

Calculate: $\mathbb{E}\left\{V\left[x_{k+1}\right]\right\}$ or $\gamma \sum p\left(s' \mid s\right) V\left(s'\right)$
The Markov decision process has the above two properties:

1. Bellman equation recursively solves the problem into sub-problems
2. The value function is equivalent to the solutions of some sub-problems, which can be saved and reused.

So we can use DP.
We need

- Dynamic /transient $P\left(s' \mid s, a\right)$
- The immediate reward (Reward model) $r\left(s, a\right)$
- Markov assumption (Belman equation)

Dynamic Programming

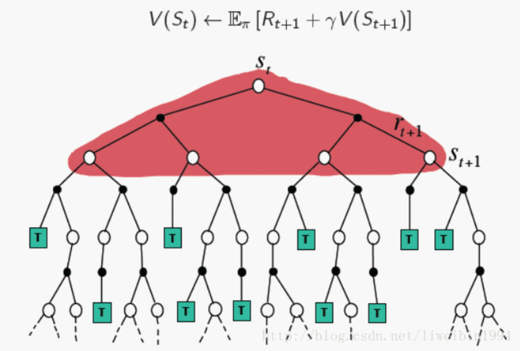
Initialize $V_0^\pi(s) = 0$ for all s

For $k = 1$ until convergence ($\left|V_k^\pi - V_{k-1}^\pi\right| < \varepsilon$)

For all s

$$V_{k+1}^\pi(s) = r_k(s, \pi(s)) + \gamma \sum_{s' \in S} P\left(s' \mid s, \pi(s)\right) V_k^\pi\left(s'\right)$$

end


$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1})\right]$
s_t
r_{t+1}
s_{t+1}
T
http://blog.csdn.net/liweibin1994

Monte Carlo (MC)

MP: value=mean return

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[G_t \mid s\right]$$

If trajectories are all finite, we can sample them and average returns:

- average over returns from a complete episode
- requires each episode to terminate

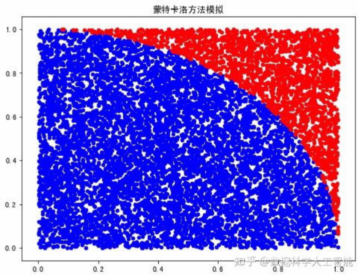
It does not require MDP dynamic, does not assume the state is Markov, no bootstrapping

$$average\left[r_k + \gamma V_k\left(s'\right)\right] \to V_k^{\pi}(s)$$

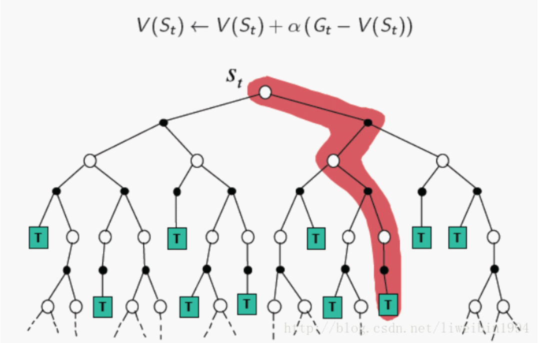
For repeated process

The area of blue color is $\frac{1}{4}\pi$, add points **randomly**

$$\pi = \lim_{n\to\infty} 4 \frac{\text{blue color points}}{\text{red color points+blue color points}} (1 \times 1)$$



Use the policy π to do many experiments and generate many serials of data (episode). Each episode starts from an arbitrary initial state until the end state,



Example

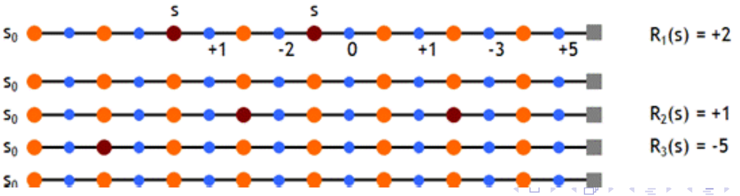
Average:When using the Monte Carlo method to find the value function at the state s, it can be divided into

- First-visit MC method: only consider the value when the state s is accessed for the first time in each episode.

$$v(s) = \frac{G_{11}(s) + G_{31}(s) + G_{41}(s)}{3}$$

- Every-visit MC method: consider all returns when the state s is accessed

$$v(s) = \frac{G_{11}(s) + G_{12}(s) + G_{31}(s) + G_{32}(s) + G_{41}(s)}{5}$$



Properties of first-visit Monte Carlo (On Policy Evaluation)

By the law of large numbers, the sequence of averages of these estimates converges to their expected value

$$V(s) \overset{n\rightarrow\infty}{\longrightarrow} V^{\pi}(s)$$

The standard deviation of its error falls as $\frac{1}{\sqrt{n}}$, where n is the number of returns average

$V(s)$ estimator $V^{\pi}(s) = \frac{G(s)}{N(s)}$ is an unbiased of the true $\mathbb{E}_{\pi}\left[G_t \mid s_t = s\right]$

By law of large numbers $N(s \rightarrow \infty$

$$V^{\pi}(s) \rightarrow \mathbb{E}_{\pi}\left[G_t \mid s_t = s\right]$$

First-visit Monte Carlo (On Policy Evaluation)

Initialize the counter $N(s) = 0,\ G(s) = 0,\ \forall s \in S$

Loop For :

Sample the episode i, generate $s_{i,1}, a_{i,1}, r_{i,1}, \cdots s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$,

calculate the return from t to all path of the ith episode

$$G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$$

For each state s visited in episode i

For the **first** time t that the state s is visited in episode i

$$\begin{aligned} N(s) &= N(s) + 1 \\ G(s) &= G(s) + G_{i,t} \\ V^{\pi}(s) &= \frac{G(s)}{N(s)} \end{aligned}$$

Bias, Variance, MSE

A statistical model is parameterized by θ and determines a probability distribution $P\left(x \mid \theta\right)$ over observed data x

If $\hat{\theta}$ is an estimate of θ

Bias of $\hat{\theta}$ is

$$Bias\left(\hat{\theta}\right) = E_{x|\theta}\left[\hat{\theta}\right] - \theta$$

Variance of $\hat{\theta}$ is

$$Var\left(\hat{\theta}\right) = E_{x|\theta}\left[\left(E_{x|\theta}\left[\hat{\theta}\right] - \hat{\theta}\right)^2\right]$$

Mean squared error (MSE) of $\hat{\theta}$ is

$$MSE\left(\hat{\theta}\right) = Var\left(\hat{\theta}\right) + Bias\left(\hat{\theta}\right)^2$$

Every-visit Monte Carlo (On Policy Evaluation)

Initialize the counter $N(s) = 0,\ G(s) = 0,\ \forall s \in S$

Loop For :

Sample the episode i, generate $s_{i,1}, a_{i,1}, r_{i,1}, \cdots s_{i,T_i}, a_{i,T_i}, r_{i,T_i}$,

calculate the return from t to all path of the ith episode

$$G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \cdots + \gamma^{T_i - 1} r_{i,T_i}$$

For each state s visited in episode i

For the **every** time t that the state s is visited in episode i

$$\begin{aligned} N(s) &= N(s) + 1 \\ G(s) &= G(s) + G_{i,t} \\ V^{\pi}(s) &= \tfrac{G(s)}{N(s)} \end{aligned}$$

Becasue

$$N_{t+1}(s) = N_t(s) + 1, G_{t+1}(s) = G_t(s) + G_{i,t}$$

and

$$V_t^{\pi}(s) = \frac{G_t(s)}{N_t(s)}$$

Incremental Monte Carlo (On Policy Evaluation)

After each episode i
For each s visited at time t in episode i,

$$
\begin{aligned}
V_{t+1}^{\pi}(s) &= \frac{G_t(s)}{N_{t+1}(s)} = \frac{G_t(s)+G_{i,t}}{N_{t+1}(s)} \\
&= \frac{G_{i,t}(s)}{N_{t+1}(s)} + \frac{N_t(s)V_t^{\pi}(s)}{N_{t+1}(s)} \\
&= \frac{G_{i,t}(s)}{N_{t+1}(s)} + \frac{(N_{t+1}(s)-1)V_t^{\pi}(s)}{N_{t+1}(s)} \\
&= V_t^{\pi}(s) + \frac{1}{N_{t+1}(s)}\left[G_{i,t}(s) - V_t^{\pi}(s)\right]
\end{aligned}
$$

So

$$
V^{\pi}(s) = V^{\pi}(s) + \alpha\left[G_{i,t}(s) - V^{\pi}(s)\right]
$$

If $\alpha = \frac{1}{N(s)}$, it is Every-visit Monte Carlo
If $\alpha > \frac{1}{N(s)}$, forget older data, helpfully for non-stationary domains

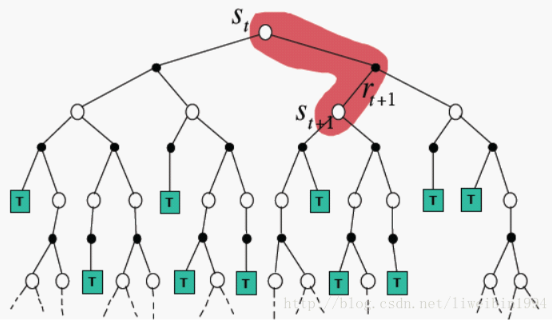
Monte Carlo Limitations

- High variance estimator; reducing it requires a lot of data
- Requires episode setting: Episode must end to update the value function. Alpha Go uses Monte Carlo

AlphaZero is not just playing better, it has discovered a new way to play !
Off-Line Training by Policy Iteration Using Self-Generated Data

Temporal Difference (TD)

$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$



Combination of MC (complete episode) and DP (all states): Bootstraps and samples

Model-free

Immediately updated estimate of V after each tuple

$$s, a, r, s'$$

Temporal Difference (TD)

Under policy π,

$$G_t = r_t + \gamma r_{t+1} + \cdots + \gamma^{T-1} r_T$$
$$V^{\pi}(s) \rightarrow \mathbb{E}_{\pi}\left[G_t \mid s_t = s\right]$$

Bellman operator (MDP model)

$$V^{\pi}(s) = r\left[s, \pi(s)\right] + \gamma \sum_{s' \in S} p\left(s' \mid s, \pi(s)\right) V\left(s'\right)$$

where immediate reward plus discounted sum of future rewards.
Incremental every-visit MC, use one sample of return

$$V^{\pi}_{t+1}(s) = V^{\pi}_{t}(s) + \alpha\left[G_{i,t}(s) - V^{\pi}_{t}(s)\right]$$

but $G_{i,t}(s)$ has to wait at the end of episode.
Why do not use our old estimator of and do not wait till the end of the episode.

$$V^{\pi}_{t+1}(s) = V^{\pi}_{t}(s) + \alpha\left[\left\{r_t + \gamma V^{\pi}_{t}\left(s_{t+1}\right)\right\} - V^{\pi}_{t}(s)\right]$$

Temporal Difference (TD)

Simplest TD learning, update it over time t

$$V_{t+1}^{\pi}(s) = V_t^{\pi}(s) + \alpha\left[\left\{r(s) + \gamma V_t^{\pi}(s') - V_t^{\pi}(s)\right\} - V_t^{\pi}(s)\right]$$
$$V^{\pi}(s) = V^{\pi}(s) + \alpha\left[\left\{r(s) + \gamma V_t^{\pi}(s') - V_t^{\pi}(s)\right\} - V^{\pi}(s)\right]$$

Or

$$V^{\pi}(s) = (1-\alpha)\, V^{\pi}(s) + \alpha\left[r(s) + \gamma V^{\pi}(s')\right]$$

where $r(s) + \gamma V_t^{\pi}(s')$ is TD target, the TD error is

$$\delta_t = r(s) + \gamma V_t^{\pi}(s') - V_t^{\pi}(s)$$

where $r(s) + \gamma V_t^{\pi}(s')$ is new estimation, r_t is immediate reward, $V^{\pi}(s)$ is value of the actual state, $V^{\pi}(s')$ is current estimate of current state (the expectations over s' value.

Temporal Difference (TD)

How different between the immediate reward+value of next state and current estimate of the value of current state

Can immediately update value after the tuple

$$s, a, r, s'$$

Do not need episode ending

Temporal Difference TD(0): Learning Algorithm

Input α
Initialize $V^{\pi}(s) = 0$
Loop

Sample tuple (s, a, r, s') at time t,

$$V^{\pi}_{t+1}(s) = V^{\pi}_{t}(s) + \alpha\left[\left\{r(s) + \gamma V^{\pi}_{t}(s')\right\} - V^{\pi}_{t}(s)\right]$$

$t = t + 1$

If $\alpha = 1$, there is only TD target $r_t + \gamma V^{\pi}(s_{t+1})$, it will oscillate, because the previous estimated is ignored.

Comparison

	DP	MC	TD
Use when no model of current domain		x	x
no episodic domain	x		x
Non-Markovian domain		x	
Converges to true value in limit 1	x	x	x
Unbiased estimate of value		x (first-visit)	