Reinforcement Learning

Lokesh Mohanty

April 25, 2023

Contents

1 References:

1.1 Reinforcement Learning: An Introduction (Sutton, Barto)

- Introduction to Reinforcement Learning
- Multi-armed bandits

1.2 Neuro-Dynamic Programming (Bertsekas, Tsitsiklis)

- Finite Horizon Problem
- Stochastic Shortest Path Problems (Study)

1.3 Dynamic Programming and Optimal Control (Bertsekas)

- Stochastic Shortest Path Problems (Practice problems)

2 Doubts

- How does exploration happen in greedy multi-armed bandits
- Upper confidence bound

3 Temporal difference Algorithm (TD(λ))

Consider the (l+1) step Bellman equation

$$J_\pi(i_k) = E_\pi \left[\sum_{n=0}^{l} g(i_k, i_{k+1}) + J_\pi(i_{k+l+1}) \right], \text{ (assuming } \lambda = 1)$$

Since l is arbitrary, we form a weighted average of these Bellman equations

Let $0 \leq \lambda < 1$, Since $(1-\lambda)\sum_{l=0}^{\infty} \lambda^l = 1$, we rewrite the above to obtain a weighted Bellman equation

$$\begin{aligned}
J_\pi(i_k) &= (1-\lambda) E \left[\sum_{l=0}^{\infty} \lambda^l \left(\sum_{m=0}^{l} g(i_{k+m}, i_{k+m+1}) + J_\pi(i_{k+l+1}) \right) \right] \\
&= (1-\lambda) E \left[\sum_{l=0}^{\infty} \lambda^l \sum_{m=0}^{l} g(i_{k+m}, i_{k+m+1}) \right] + (1-\lambda) E \left[\sum_{l=0}^{\infty} \lambda^l J_\pi(i_{k+l+1}) \right]
\end{aligned}$$

Expanding the 1st part

$$\begin{aligned}
(1-\lambda) E \left[\sum_{l=0}^{\infty} \lambda^l \sum_{m=0}^{l} g(i_{k+m}, i_{k+m+1}) \right] &= (1-\lambda) E \left[\sum_{m=0}^{\infty} \sum_{l=m}^{l} \lambda^l g(i_{k+m}, i_{k+m+1}) \right] \\
&= (1-\lambda) \frac{E\left[\sum_{m=0}^{\infty} \lambda^m g(i_{k+m}, i_{k+m+1})\right]}{(1-\lambda)} \\
&= E \left[\sum_{m=0}^{\infty} \lambda^m g(i_{k+m}, i_{k+m+1}) \right]
\end{aligned}$$

Expanding the 2nd part

$$\begin{aligned}
(1-\lambda) E \left[\sum_{l=0}^{\infty} \lambda^l J_\pi(i_{k+l+1}) \right] &= E \left[\sum_{l=0}^{\infty} (\lambda^l - \lambda^{l+1}) J_\pi(i_{k+l+1}) \right] \\
&= E \left[(1-\lambda) J_\pi(i_{k+1}) + (\lambda - \lambda^2) J_\pi(i_{k+2}) + ... \right] \\
&= E \left[J_\pi(i_{k+1}) - J_\pi(i_k) + \lambda(J_\pi(i_{k+2}) - J_\pi(i_{k+1})) + \lambda^2(J_\pi(i_{k+3}) - J_\pi(i_{k+2} \right. \\
&= E \left[\sum_{m=0}^{\infty} \lambda^m (J_\pi(i_{k+m+1}) - J_\pi(i_{k+m})) \right] + J_\pi(i_k)
\end{aligned}$$

Combining the 2 parts, we get

$$J_\pi(i_k) = E\left[\sum_{m=0}^{\infty} \lambda^m \left(g(i_{k+m}, i_{k+m+1}) + J_\pi(i_{k+m+1}) - J_\pi(i_{k+m})\right)\right] + J_\pi(i_k)$$

Since we are in the setting of SSPP, there is a time N with $N < \infty$ such that $i_N = 0$ (terminal state). Further, $v_\pi(i_N) = 0$, $g(i_{N+m}, i_{N+m}) = 0\ \forall m \geq 0$.

Let $d_m = g(i_m, i_{m+1}) + J_\pi(i_{m+1}) - J_\pi(i_m)$ (temporal difference term)

Then,

$$\begin{aligned} J_\pi(i_k) &= E\left[\sum_{m=0}^{\infty} \lambda^m d_{m+k}\right] + J_\pi(i_k) \\ &= E\left[\sum_{m=k}^{\infty} \lambda^{m-k} d_m\right] + J_\pi(i_k) \\ E\left[\sum_{m=k}^{\infty} \lambda^{m-k} d_m\right] &= 0, \text{ (true since } E_\pi[d_m] = 0, \forall m) \end{aligned}$$

3.1 Robbins Monro Algorithm (for the above)

$$J(i_k) := J(i_k) + Y \sum_{m=k}^{\infty} \lambda^{m-k} \overline{d}_m$$
$$\text{where } \overline{d}_m = g(i_m, i_{m+1}) + J(i_{m+1}) - J(i_m)$$

Here, Y is the step-size parameter As the number of iterates tends to ∞,

$$J(i_k) \to J_\pi(i_k)$$

3.2 Special Cases

1. $\lambda = 0$ (TD(0) algorithm)

$$\begin{aligned} J(i_k) &:= J(i_k) + Y\overline{d}_k \\ &= J(i_k) + Y(g(i_k, i_{k+1}) + J(i_{k+1}) - J(i_k)) \end{aligned}$$

1. $\lambda = 1$ (Monte-Carlo or TD(1) algorithm)

$$
\begin{aligned}
J(i_k) :&= J(i_k) + Y \sum_{m=k}^{N-1} \overline{d}_k \\
&= J(i_k) + Y(\overline{d}_k + \overline{d}_{k+1} + ... + \overline{d}_{N-1}) \\
&= J(i_k) + Y(g(i_k, i_{k+1}) + g(i_{k+1}, i_{k+2}) + ... + g(i_{N-1}, i_N) + J(i_{k+1}) - J(i_k)) \\
\implies J(i_k) :&= J(i_k) + Y(g(i_k, i_{k+1}) + g(i_{k+1}, i_{k+2}) + ... + g(i_{N-1}, i_N) + J(i_{k+1}) - J(i_k))
\end{aligned}
$$

3.3 Q-learning

Recall now the Bellman equation for optimality

$$J^*(i) = \min_{\mu \in A(i)} \sum_{j \in S} p_{ij}(\mu)(g(i, \mu, j) + J^*(j)),\ i \in S \text{ (SSPP setting)}$$

Let $Q^*(i, \mu) = \sum_{j \in S} p_{ij}(\mu)(g(i, \mu, j) + J^*(j))$, $i \in S$, (these are called Q-values)

Then,

$$J^*(i) = \min_{\mu \in A(i)} Q^*(i, \mu),\ \forall i \in S$$

Thus, (Q-Bellman Equation in the state-action tuples (i, μ))

$$Q^*(i, \mu) = \sum_{j \in S} p_{ij}(\mu)(g(i, \mu, j) + \min_{\mu \in A(j)} Q^*(j, \mu) = E\left[g(i, \mu, n) + \min_{\mu \in A(n)} Q^*(n, \mu)\right]$$

Numerical procedure for solving Q-Bellman Equation
Q-value iteration:

$$Q_{m+1}(i, \mu) = \sum_{j \in S} p_{ij}(\mu)(g(i, \mu, j) + \min_{\mu \in A(j)} Q_m(j, \mu)),\ m = 0, 1, 2, ...$$

In case we don't know $p_{ij}(\mu)$, we resent to data driven (model-free) scheme. (update full Q-table at each instant)

$$Q_{m+1}(i, \mu) = Q_{m(i,\mu))} + Y(g(i, \mu, j) + \min_{\mu \in A(j)} Q_m(j, \mu) - Q_m(i, \mu))$$

Key problem: When Q-estimates are not properly developed, there is significant bias in algorithm. This algorithm requires one to explore Q-values sufficiently for the various actions.

Consider asynchronous version of the algorithm

$$Q_{m+1}(i_m, \mu_m) = Q_m(i_m, \mu_m) + Y(i_m, \mu_m)(g(i_m, \mu_m, i_{m+1}) + Q_m(i_{m+1}, \mu_{m+1}) - Q_m(i_m, \mu_m))$$

Here $Y(i_m, \mu_m) = \frac{1}{m}$ if i_m is the state visited at m and μ_m

Note: if μ_m is selected according to some policy π(fixed) in i_m, then TD(1) is simply TD(0)

3.3.1 Recall the Q-learning algorithm

$$Q_{t+1}(i_t, \mu_t) = Q_t(i_t, \mu_t) + \gamma(g(i_t, \mu_t, i_{t+1}) + Q_t(i_{t+1}, \mu_{t+1}) - Q_t(i_t, \mu_t))$$

Q) How to select μ_t in state i_t ... μ_{t+1} in state i_{t+1}

Possibility 1 (SARSA) (State Action Reward State Action) (on-policy)

$$\mu_t = \begin{cases} \arg\min_\mu Q_t(i_t, \mu) \text{ with p } 1-\epsilon \\ \text{random action with p } 1-\epsilon \end{cases}$$

$$\mu_{t+1} = \begin{cases} \arg\min_\mu Q_t(i_{t+1}, \mu) \text{ with p } 1-\epsilon \\ \text{random action with p } 1-\epsilon \end{cases}$$

Possibility 2 (Q-learning) (off-policy)

$$\mu_t = \begin{cases} \arg\min_\mu Q_t(i_t, \mu) \text{ with p } 1-\epsilon \\ \text{random action with p } 1-\epsilon \end{cases}$$

$$\mu_{t+1} = \arg\min_\mu Q_t(i_{t+1}, \mu)$$

target: greedy behaviour: epsilon greedy

4 On-policy vs off-policy methods (02/03/2023) (Chapter 5 of Sutton-Barto)

On-policy: data available from the policy for which we wish to find the value function Off-policy: data from a given policy is to be used to find value function of another policy (policy is hardwired)

Eg: Traffic signal control

Phase : A set of signals that go green together Q) Can we dynammically allocate green time to the phases? cost = sum of queue lengths at all junctions

4.1 Problem:

- Data is available from a behaviour policy (b)
- We want to estimate value function of another policy ($v_\pi(s)$) -> target policy (π)

Importance Sampling: Consider

$$\begin{aligned} P(A_t, S_{t+1}, A_{t+1}, ..., S_T | S_t, A_{t=T-1} \sim \pi) &= P(S_T | S_{T-1}, A_{T-1}, ..., S_{t+1}, A_t, S_t, A_{t=T-1} \sim \pi) \\ &\times P(S_{T-1}, A_{T-1}, ..., S_{t+1}, A_t | S_t, A_{t=T-1} \sim \pi) \\ &= P(S_T | S_{T-1}, A_{T-1}) \pi(A_{T-1} | S_{T-1}) P(S_{T-1} | S_{T-2}, A_{T-2}) \pi(A_{T-} \\ &= \Pi_{k=t}^{T-1} \pi(A_k | S_k) p(S_{k+1} | S_k, A_k) \end{aligned}$$

Similarly,

$$P(A_t, S_{t+1}, A_{t+1}, ..., S_T | S_t, A_{t=T-1} \sim b) = \Pi_{k=t}^{T-1} b(A_k | S_k) p(S_{k+1} | S_k, A_k)$$

Define the importance sampling ratio as

$$\begin{aligned} P_{t=T-1} &= \frac{P(A_t, S_{t+1}, A_{t+1}, S_{t+2}, ..., S_T | S_t, A_{t=T-1} \sim \pi)}{P(A_t, S_{t+1}, A_{t+1}, S_{t+2}, ..., S_T | S_t, A_{t=T-1} \sim b)} \\ &= \frac{\Pi_{k=t}^{T-1} \pi(A_k | S_k) \cancel{p(S_{k+1} | S_k, A_k)}}{\Pi_{k=t}^{T-1} b(A_k | S_k) \cancel{p(S_{k+1} | S_k, A_k)}} = \Pi_{k=t}^{T-1} \frac{\pi(A_k | S_k)}{b(A_k | S_k)} \end{aligned}$$

Note, we may estimate $v_b(s) = \mathbb{E}[G_t | S_t = s, b]$, $G_t = g(S_t, S_{t+1}) + \gamma g(S_{t+1}, S_{t+2}) + ... + \gamma^{T-t-1} g(S_{T-1}, S_T)$ Consider

$$\mathbb{E}[P_{t=T-1} G_t | S_t = s, b] = \mathbb{E}\left[\left(\Pi_{k=t}^{T-1} \frac{\pi(A_k | S_k)}{b(A_k | S_k)} \right) G_t \middle| S_t = s, b \right]$$

This expectation is w.r.t. dist $P(A_t, S_{t+1}, ..., S_T | S_t, A_{t=T-1} \sim b)$. Thus $\mathbb{E}[P_{t=T-1} G_t | S_t = s, b] = v_\pi(s)$

4.2 Monte-Carlo algorithm (estimates $v_\pi(s)$ from data coming according to b)

Let \$τ(s) = \$ set of all time steps in which state s is visited. (every visit method) $T(t) =$ first time after t that termination happens

$\{G_t\}_{t \in \tau(s)}$ are the returns pertaining to state S and $\{P_{t=T(t)-1}\}_{t \in \tau(s)}$ are the corresponding IS ratios.

4.3 Regular Monte-Carlo estimate:

$$v(s) = \frac{\sum_{t \in \tau(s)} p_{t=T(t)-1} G_t}{|\tau(s)|}$$

4.4 Low variance estimate

4.5 Incremental Implementation

Let $W_i = p_{t_i:T(t_i)-1}$, where t_i = ith time that state i is visited on the concateneted trajectory

$$\begin{aligned}
V_{n+1} &= \frac{\sum_{k=1}^{n+1} W_k G_k}{\sum_{k=1}^{n+1} W_k} = \frac{\sum_{k=1}^{n} W_k G_k + W_{n+1} G_{n+1}}{\sum_{k=1}^{n+1} W_k} \\
&= \left(\frac{\sum_{k=1}^{n} W_k}{\sum_{k=1}^{n+1} W_k} \right) \frac{\sum_{k=1}^{n} W_k G_k}{\sum_{k=1}^{n} W_k} + \frac{W_{n+1} G_{n+1}}{\sum_{k=1}^{n+1} W_k} \\
&= \left(\frac{\sum_{k=1}^{n} W_k}{\sum_{k=1}^{n+1} W_k} \right) V_n + \frac{W_{n+1} G_{n+1}}{\sum_{k=1}^{n+1} W_k} \\
&= V_n + \frac{W_{n+1}}{\sum_{k=1}^{n+1} W_k} (G_{n+1} - V_n)
\end{aligned}$$

Let $C_n = \sum_{k=1}^{n} W_k$ (Cumulative sum of weights for 1st n returns) and $C_0 = 0$
Then $C_{n+1} = C_n + W_{n+1}$ and $V_{n+1} = V_n + \frac{W_{n+1}}{C_{n+1}}[G_{n+1} - V_n]$. The above formula will also sork for on-policy by letting $W_n = 1, \forall n$

4.6 Important (for off-policy methods)

Assumption of coverage:

If $\pi(a|s) > 0$ for any $a \in A(s)$ then $b(a|s) > 0$ for that $a \in A(s) \implies$ support of b should contain the support of π

5 (09/03/2023)

We need to show that

$$| \min_{v \in A(j)} Q(j,v) - \min_{v \in A(j)} \overline{Q}(j,v) | \leq \max_{v \in A(j)} | Q(j,v) - \overline{Q}(j,v) |$$

Note: If $A \subset B$, then

$$\inf_{x \in A} f(x) \geq \inf_{x \in B} f(x)$$

infimum -> greatest lower bound supremum -> least upper bound
Thus,

$$\inf_{x \in A}(f(x) + g(x)) = \inf_{x \in A, y = x}(f(x) + g(y)) \geq \inf_{x, y \in A}(f(x) + g(y))$$

$$\implies \inf_{x \in A}((f-g)(x) + g(x)) \geq \inf_{x \in A} g(x)$$
$$\implies \inf_{x \in A}(f-g)(x) \leq \inf_{x \in A} f(x) - \inf_{x \in A} g(x)$$
$$\text{Let } h(x) = -g(x)\, \forall x$$
$$\text{Then } \sup_{x \in A} h(x) = \sup_{x \in A}(-g(x)) = -\inf_{x \in A} g(x)$$
$$\implies \inf_{x \in A}(f(x) + h(x)) \leq \inf_{x \in A} f(x) + \sup_{x \in A} h(x)$$
$$\implies \inf_{x \in A}(f(x) + h(x)) - \inf_{x \in A} f(x) \leq \sup_{x \in A} h(x)$$
$$\text{Let } h(x) = g(x) - f(x)$$
$$\implies \inf_{x \in A} g(x) - \inf_{x \in A} f(x) \leq \sup_{x \in A}(g(x) - f(x))$$
$$\implies \inf_{x \in A} g(x) - \inf_{x \in A} f(x) \leq \sup_{x \in A} \mid g(x) - f(x) \mid$$

Claim:

$$\mid \sup_{x \in A}(g(x) - f(x)) \mid \leq \sup_{x \in A} \mid g(x) - f(x) \mid$$

Case (i):

$$\sup_{x \in A}(g(x) - f(x)) \geq 0$$
$$\implies \sup_{x \in A}(g(x) - f(x)) \leq \sup_{x \in A} \mid g(x) - f(x) \mid$$

Case (ii):

$$\sup_{x \in A}(g(x) - f(x)) < 0$$
$$\mid g(x) - f(x) \mid = -(g(x) - f(x))\, \forall x$$

$$\implies |\sup_{x\in A}(g(x)-f(x))| = -\sup_{x\in A}(g(x)-f(x)) = \inf_{x\in A}(-(g(x)-f(x))) = \inf_{x\in A} | g(x)-f(x) | \sup_{x\in A} | g(x)$$

$$\implies \inf_{x\in A} g(x) - \inf_{x\in A} f(x) \leq \sup_{x\in A} | g(x)-f(x) |$$

Also,

$$\implies \inf_{x\in A} f(x) - \inf_{x\in A} g(x) \leq \sup_{x\in A} | g(x)-f(x) |$$

$$\implies | \inf_{x\in A} f(x) - \inf_{x\in A} g(x) | \leq \sup_{x\in A} | g(x)-f(x) |$$

Thus it follows that

$$| \min_{v\in A(j)} Q(j,v) -_{v\in A(j)} \overline{Q}(j,v) | \leq \max_{v\in A(j)} | Q(j,v) - \overline{Q}(j,v) |$$

6 Function Approximations based approaches for Reinforcement Learning (09/03/2023)

Suppose each route has a buffer that can store 1000 packets. Q-learning and Sarsa algorithms, based on lookup table updates cannot be applied.

We need to resort to approximations

- Value function approximations (Temporal difference learning, Q-Learning, ...)
- Policy approximations (policy gradient methods, actor critic methods, ...)

6.1 Value function approximations (09/03/2023)

Given policy π, value function

$$v_\pi(s) = \lim_{N\to\infty} \mathbb{E}\left[\sum_{k=0}^{N-1} \gamma^k g(i_k, \pi(i_k), i_{k+1}) \middle| i_0 = s\right] \forall s \in S$$

Let $v_\pi(s) \approx \hat{v}(s,w)$ where $w \in \mathbb{R}^d$ is a parameter Invariably, $d << |s|$

Examples:

6.1.1 (i) Linear approximation architectures

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

Where $\phi(s) = (\phi_1(s), \phi_2(s), ..., \phi_d(s))^T$(feature of state s, can be highly non linear), $w = (w_1, w_2, ..., w_d)^T$

Examples of LFA:

1. (a) polynomial features suppose $s = (s_1, s_2)^T$

   Polynomial representations:

   $$\rightarrow \phi(s) = (1, s_1, s_2, s_1 s_2)^T$$
   $$\rightarrow \phi(s) = (1, s_1, s_2, s_1 s_2, s_1^2 s_2, s_1 s_2^2, s_1^2 s_2^2)^T$$

2. (b) Fourier bases Example: Let $s = (s_1, s_2, ..., s_k)^T$ with each $s_i \in [0, 1]$, Then $\phi_i(s) = cos(\pi s^T c^i)$, where $c^i = (c_1^i, ..., c_k^i)^T$ with $c_j^i \in \{0, 1, ..., n\}, j = 1, ..., k$ c^i takes $(n+1)^k$ values $s^T c^i$ has the effect of assuming an integer in $\{0, 1, ...n\}$ to each _ of s The integer determines the feature frequency along that dim

6.1.2 (ii) Nonlinear approximaiton architectures (neural nets based architectures)

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

Prediction Error objective:

$$\overline{VE}(w) = \sum_{s \in S} \mu(s)(v_\pi(s) - \hat{v}(s,w))^2$$

Here, $\mu(s), s \in S$ is the steady state distribution of the markov chain unser the given policy Let $\mu(s) > 0 \, \forall s \in S$

$$\{x_t\} \text{ or } \{S_t\} \rightarrow p^\pi(s, s') = \sum_{a \in A(s)} \pi(a|s) p(s'|s < a)$$

Goal: Find w^* that minimizes $\overline{VE}(w)$ which implies that distribution of $\hat{v}(s, w^*)$ from $v_\pi(s)$ is the minimum over all $\hat{v}(s, w)$

Lets use gradient search

$$w_{t+1} = w_t - \frac{1}{2}\alpha\nabla\overline{VE}(w_t)$$

$$\begin{aligned}
\nabla\overline{VE}(w_t) &= \nabla_w\left(\sum_{s\in S}\mu(s)(v_\pi(s)-\hat{v}(s,w))^2\right)\\
&= \sum_{s\in S}\mu(s)\nabla_w(v_\pi(s)-\hat{v}(s,w))^2\\
&= -2\sum_{s\in S}\mu(s)(v_\pi(s)-\hat{v}(s,w))\nabla_w\hat{v}(s,w)
\end{aligned}$$

The algorithm then is

$$w_{t+1} = w_t + \alpha\sum_{s\in S}\mu(s)(v_\pi(s)-\hat{v}(s,w))\nabla_w\hat{v}(s,w)$$

Problems with this update rule: (i) we don't know $\mu(s)$ (ii) we don't know $v_\pi(s)$

Use stochastic approximation (i.e. we use SGD(stochastic gradient descent))

$$w_{t+1} = w_t + \alpha(v_\pi(s_t)-\hat{v}(s_t,w_t))\nabla_w\hat{v}(s_t,w_t),\ s_t \text{ is the state visited at time t}$$

Also, $\mathbb{E}_0[v_\pi(s_t)] = \sum_{s\in S}\mu(s)v_\pi(s)$ where $\mathbb{E}_0$ is the expectation under the stationary list of the Markov chain $\{S_t\}$

2nd Problem: Instead of $v_\pi(s_t)$ use G_t (gradient Monte-Carlo)

6.2 Prediction Error Objective (14/03/2023)

$$\overline{VE}(w) = \sum_s\mu(s)\left(v_\pi(s)-\hat{v}(s,w)\right)^2$$

$\mu(s)$: average time spent in state s by the Markov chain $\{S_t\}$. $\hat{v}(s,w)$: approximate value function is a parameterized space with parameter $w\in\mathbb{R}^l$

Relaxed objective: Find a local minimum instead **Update rule**: Gradient Search

$$\begin{aligned}
w_{t+1} &= w_t - \frac{1}{2}\alpha\nabla\overline{VE}(w_t) &(1)\\
&= w_t + \alpha\sum_{s\in S}\mu(s)\left(v_\pi(s)-\hat{v}(s,w_t)\right)\nabla\hat{v}(s,w_t) &(2)
\end{aligned}$$

$\mu(s)$ is not known

Sample based update

$$w_{t+1} = w_t + \alpha(v_\pi(s_t) - \hat{v}(s_t, w_t))\nabla\hat{v}(s_t, w_t)$$

s_t : state visited at time t

Will work because

Steady state expectation:

$$E_0[v_\pi(s_t)] = \sum_{s\in S}\mu(s)v_\pi(s)$$

problem is that we don't know $v_\pi(s_t)$

6.2.1 Gradient Monte-Carlo Algorithm

$W_{t+1} = w_t + \alpha\left(G_t - \hat{v}(s_t, w_t)\right)\nabla\hat{v}(s_t, w_t)$

$G_t = (\gamma(s_t, \pi(s_t), s_{t+1}) + Y\gamma(s_{t+1}, \pi(s_{t+1}), s_{t+2}) + ... + Y^{T-t-1}\gamma(s_{T-1}, \pi(s_{T-1}), s_T))$

G_t : return on the episode starting from state s_t (could be first visit return or that obtained using every visit procedure)

6.2.2 Alternative to trajectory-based methods (incremental update methods)

TE(0) with function approximation

Recall the Bellman Equation for a given policy π

$$v_\pi(s) == E_{s'}\left[\gamma(s, \pi(s), s') + Yv_\pi(s')\right]$$

Recall that in TD(0) without function approximation, Then estimate $v(s)$ of $v_\pi(s)$ is $\gamma(s, \pi(s), s') + Yv(s')$

$$\begin{aligned} v_\pi(s) &= E\left[G_t S_t = s\right] \\ &= E\left[\sum_{t=1}^{\infty} Y^t\gamma(s_t, \pi(s_t), s_{t+1})\, S_0 = s\right] \end{aligned}$$

1. TD(0) algorithm with function approximation

$$w_{t+1} = w_t + \alpha(\gamma(s_t, \pi(s_t), s_{t+1}) + Y\hat{v}(s_{t+1}, w_t) - \hat{v}(s_t, w_t))\nabla\hat{v}(s_t, w_t)$$

Important Special case (TD(0) with LFA):

Linear function approximation: $\hat{v}(s,w) = w^T\phi(s)$ $\phi(s)$: state features, $w \in \mathbb{R}^d$, $\phi(s) \in \mathbb{R}^d$

Under LFA, $\nabla\hat{v}(s_t, w_t) = \phi(s_t)$

$$\begin{aligned} w_{t+1} &= w_t + \alpha(\gamma(s_t, \pi(s_t), s_{t+1}) + Y w_t^T \phi(s_{t+1}) - w_t^T \phi(s_t))\phi(s_t) \\ &= w_t + \alpha(\gamma(s_t, \pi(s_t), s_{t+1}) + w_t^T(Y\phi(s_{t+1}) - \phi(s_t)))\phi(s_t) \\ &= w_t + \alpha\phi(s_t)(\gamma(s_t, \pi(s_t), s_{t+1}) + (Y\phi(s_{t+1}) - \phi(s_t))^T w_t) \end{aligned}$$

Consider the LFA architecture, $\hat{v}(i,w) = \phi(i)^T w$ Here, $w = (w_1, ..., w_d)^T$, $\phi(i) = (\phi_1(i), \phi_2(i), ..., \phi_d(i))^T$

Let the feature matrix $\Phi = \begin{bmatrix} \phi(1)^T \\ \phi(2)^T \\ \vdots \\ \phi(|s|)^T \end{bmatrix}_{|s|\times d}$

Let $\hat{v}_w = (\hat{v}(i,w), i \in S)^T$, then

$$\hat{v}_w = \Phi w = \begin{bmatrix} \phi_1(1) \\ \phi_1(2) \\ \vdots \\ \phi_1(|s|) \end{bmatrix} w_1 + \begin{bmatrix} \phi_2(1) \\ \phi_2(2) \\ \vdots \\ \phi_2(|s|) \end{bmatrix} w_2 + ... + \begin{bmatrix} \phi_d(1) \\ \phi_d(2) \\ \vdots \\ \phi_d(|s|) \end{bmatrix} w_d$$

Let $\phi_i = \begin{bmatrix} \phi_i(1) \\ \phi_i(2) \\ \vdots \\ \phi_i(|s|) \end{bmatrix}$: ith feature vector or ith basis vector

Let $S_0 = \{\Phi w | w \in \mathbb{R}^d\}$ denote hte space of linear function approximations parameterized by $w \in \mathbb{R}^d$

2. Assumptions:

   (a) The Markov Chain $\{S_n\}$ has steady-state probabilities $\zeta_1, \zeta_2, ..., \zeta_{|s|}$ with $\zeta_j > 0 \ \forall j \in S$

   (b) The matrix Φ has rank d and $|s| \geq d$

3. Projected Bellman Equation Define a weighted Euclidean norm on $\mathbb{R}^{|s|}$ as

$$\|V\|_x = \sqrt{V^T \times V} = \sqrt{\sum_{i=1}^{|s|} x_i (v(i))^2}$$

Here, $X = \begin{bmatrix} x_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & x_{|s|} \end{bmatrix}$ Assume $x_1, x_2, ..., x_{|s|} > 0$ Let π be the projection operator from $\mathbb{R}^{|s|}$ to s_0 w.r.t. $\|.\|_x$. Thus for any $v \in \mathbb{R}^{|s|}$, TTV is the unique vector is s_0 that minimizes $\|v - \hat{v}\|_x^2$ over all $\hat{v} \in S_0$. Since Φ has rank d, any $\hat{v} \in S_0$ is uniquely written as $\hat{v} = \Phi w$ for some $w \in \mathbb{R}^d$

$$\implies \|v - \hat{v}\|_x^2 = \|v - \Phi w\|_x^2 = (v - \Phi w)^T x (v - \Phi w)$$

Thus $\pi v = \Phi w_v$, where $w_v = \arg\min_{w \in \mathbb{R}^d} \|v - \Phi w\|_x^2$, $v \in \mathbb{R}^{|s|}$

In order to find w_v, compute $\nabla_w (\|v - \Phi w\|_x^2)$ and set it to 0, then

$$\begin{aligned} \nabla_w (\|v - \Phi w\|_x^2 &= \nabla_w (\|v - \Phi w\|_x^2) \\ &= \nabla_w ((v - \Phi w)^T \times (v - \Phi w)) \\ &= \nabla_w (v^T \times v - w^T \Phi^T v - v^T \times \Phi w + w^T \Phi^T \times \Phi w) \\ &= -2\Phi^T \times v + 2\Phi^T \times \Phi w = 0 \\ \implies \Phi^T \times v &= (\Phi^T \times \Phi) w_v \\ \implies w_v &= (\Phi^T \times \Phi)^{-1} \Phi^T v. \end{aligned}$$

Thus, the point $\hat{v}$ in s_0 correspoiding to parameter w_v is $\hat{v} = \Phi w_v = \Phi(\Phi^T \times \Phi)^{-1} \Phi^T v = \pi v$ Note: $(\Phi^T \times \Phi)$: positive definite matrix, since Φ has rank d and x has all positive values.

6.3 Projected Bellman Equation (16/03/2023)

Define a weighted Euclidean norm on $\mathbb{R}^{|s|}$ as

$$\|J\|_\xi = \sqrt{J^T D J} = \sqrt{\sum_{i=1}^{|s|} \xi_i J(i)^2}$$

$\xi = (\xi_1, ..., \xi_{|s|})^T$ is the stationary distribution of $\{S_t\}$ $D = \begin{bmatrix} \xi_1 & & 0 \\ & \ddots & \\ 0 & & \xi_{|s|} \end{bmatrix}$

Let π be the projection operator onto $S_0 = \{\Phi w | w \in \mathbb{R}^d\}$ for any $J \in \mathbb{R}^{|s|}$, ΠJ is the unique vector in S_0 that minimizes $\|J - \hat{J}\|_\xi$ over all $\hat{J} \in S_0$

Since Φ has rank d, any $\hat{J} \in S_0$ is uniquely written as $\hat{J} = \Phi w$ for some $w \in \mathbb{R}^d$

$$\|J - \hat{J}\|_\xi^2 = \|J - \Phi w\|_\xi^2 = (J - \Phi w)^T D (J - \Phi w)$$

$\therefore \Pi J = \Phi w_J$ where $w_J = \arg\min_{w \in \mathbb{R}^{|s|}} \|J - \Phi w\|_\xi^2$, $J \in \mathbb{R}^{|s|}$

In order to find w_J,

$$\nabla_w (\|J - \Phi w\|_\xi^2) = 0$$
$$\Phi^T D (J - \Phi w_J) = 0$$

For any $w \in \mathbb{R}^d$, $\Phi w \in S_0 \implies w^T \Phi^T D (J - \Phi w_J) = 0$

$$\implies w_J = (\Phi^T D \Phi)^{-1} \Phi^T D J$$
$$\implies \Phi w_J = \Phi (\Phi^T D \Phi)^{-1} \Phi^T D J$$
$$\implies \Pi = \Phi (\Phi^T D \Phi)^{-1} \Phi^T D J$$

Any vectors x, y are orthogonal if $x^T D y = 0 \implies \sum_{i=1}^{|s|} \xi_i x_i y_i = 0$

Recall Bellman Equation for policy π,

$$J = T_\pi J$$
$$\implies T_\pi J = ((T_\pi, J)(i), i \in S)^T$$

where $(T_\pi J)(i) = \sum_{j \in S} p_{ij}(\pi(i))(g(i, \pi(i), j) + \gamma J(j))$

Projected Bellman Equation: $\Phi w = \Pi T_\pi (\Phi w)$ View ΠT_π as a compositoin of Π and T_π

6.3.1 Lemma 1:

$\|P_\pi z\|_\xi \leq \|z\|_\xi$ $\forall z \in \mathbb{R}^{|s|}$, $P_\pi = \begin{bmatrix} P_\pi(1,1) & \dots & P_\pi(1,|s|) \\ \vdots & \ddots & \vdots \\ P_\pi(|s|,1) & \dots & P_\pi(|s|,|s|) \end{bmatrix}$

$$
\begin{aligned}
\|P_\pi z\|_\xi^2 &= \sum_{i=1}^{|s|} \xi_i \left(\sum_{j=1}^{|s|} p_{ij} z_j \right)^2 \leq \sum_{i=1}^{|s|} \xi_i \sum_{j=1}^{|s|} p_{ij} z_j^2 \\
&= \sum_{j=1}^{|s|} \left(\sum_{i=1}^{|s|} \xi_i p_{ij} \right) z_j^2 = \sum_{j=1}^{n} \xi_j z_j^2 = \|z\|_2^2
\end{aligned}
$$

$\xi = (\xi_1, \xi_2, ..., \xi_{|s|})^T$ is the stationary distribution of $\{S_n\}$ under policy π
$\xi^T P_\pi = \xi^T$ as $\xi(i)^T P_\pi = \xi(i+1)$

Thus $\|P_\pi z\|_\xi \leq \|z\|_\xi$

6.3.2 Lemma 2:

The projection map Π is non-expansive, i.e., $\|\Pi J - \Pi \overline{J}\|_\xi \leq \|J - \overline{J}\|_\xi \ \forall J, \overline{J} \in \mathbb{R}^{|s|}$ Note that,

$$
\begin{aligned}
\|\Pi(J - \overline{J})\|_\xi^2 &\leq \|\Pi(J - \overline{J})\|_\xi^2 + \|(I - \Pi)(J - \overline{J})\|_\xi^2 \\
&= \|\Pi(J - \overline{J})\|_\xi^2 + \|(J - \overline{J}) - \Pi(J - \overline{J})\|_\xi^2
\end{aligned}
$$

Note: $\Pi(J - \overline{J}) \perp ((J - \overline{J}) - \Pi(J - \overline{J}))$ Therefore by Pythagorean theorem,

$$
\begin{aligned}
\|\Pi(J - \overline{J})\|_\xi^2 &\leq \|\Pi(J - \overline{J})\|_\xi^2 + \|(I - \Pi)(J - \overline{J})\|_\xi^2 \\
&= \|\Pi(J - \overline{J}) + (I - \Pi)(J - \overline{J})\|_\xi^2 = \|J - \overline{J}\|_\xi^2 \\
\implies \|\Pi(J - \overline{J})\|_\xi^2 &\leq \|J - \overline{J}\|_\xi^2 \\
\implies \|\Pi(J - \overline{J})\|_\xi &\leq \|J - \overline{J}\|_\xi
\end{aligned}
$$

Proposition: Let Πr^* be the fixed point of ΠT_π. Then

$$
\|J_\pi - \Phi r^*\|_\xi \leq \frac{1}{\sqrt{1 - \gamma^2}} \|J_\pi - \Pi J_\pi\|_\xi
$$

Note that:

$$
\begin{aligned}
\|J_\pi - \Phi r^*\|_\xi^2 &= \|J_\pi - \Pi J_\pi\|_\xi^2 + \|\Pi J_\pi - \Phi r^*\|_\xi^2 \\
&= \|J_\pi -_\pi\|_\xi^2 + \|\Pi T_\pi J_\pi - \Pi T_\pi(\Phi r^*)\|_\xi^2
\end{aligned}
$$

(Since $J_\pi = T_\pi J_\pi$ and $\Phi r^* = \Pi(T_\pi(\Phi r^*))$

Note that:

$$\begin{aligned}\|\Pi T_\pi J_\pi - \Pi T_\pi(\Phi r^*)\|_\xi &\leq \|T_\pi J_\pi - T_\pi(\Phi r^*)\|_\xi \text{ (by non-expansivity of } \Pi) \\ &\leq \gamma \|J_\pi - \Phi r^*\|_\xi \text{ (by contraction property of } T_\pi)\end{aligned}$$

Chapter 6, Vol 2 of Bertsekas (Approximate DP)

$$\begin{aligned} \|J_\pi - \Phi r^*\|_\xi^2 &\leq \|J_\pi - \Pi J_\pi\|_\xi^2 + \gamma^2 \|J_\pi - \Phi r^*\|_\xi^2 \\ \implies (1-\gamma^2)\|J_\pi - \Phi r^*\|_\xi^2 &\leq \|J_\pi - \Pi J_\pi\|_\xi^2 \\ \implies \|J_\pi - \Phi r^*\|_\xi &\leq \frac{1}{\sqrt{(1-\gamma^2)}} \|J_\pi - \Pi J_\pi\|_\xi \end{aligned}$$

This is the error

$r^* = \arg\min_{w \in \mathbb{R}^d} \|\Phi w - (g + \gamma P_\pi \Phi r^*)\|_\xi^2$

$\Phi^T D(I - \gamma P_\pi)\Phi r^* = \Phi^T D g \implies C r^* = d \implies r^* = C^{-1} d$, where $C_{d\times d} = \Phi^T D (I - \gamma P_\pi)\Phi$, $d = \Phi^T D g$

True Bellman Solution: $J_\pi = (I - \gamma P_\pi)^{-1}_{|s|\times|s|} g$

Numerical Solution to the projected Bellman Equation: Projected value iteratoin (PVI):

$$\Phi r_{k+1} = \Pi T_\pi \Phi r_k, k = 0, 1, 2, ...$$

Select $r_0 \in \mathbf{R}^d$ arbitrarily We know that ΠT_π is a contraction

$$r_{k+1} = \arg\min_{w \in \setminus^d} \|\Phi w - (g + \gamma P_\pi \Phi r_k\|_\xi^2$$

Consider again

$$\begin{aligned} &\nabla_w (\Phi w - g - \gamma P_\pi \Phi r_k)^T D (\Phi w - g - \gamma P_\pi \Phi r_k)) = 0 \\ &\implies 2\Phi^T D(\Phi r_{k+1} - (g + \gamma P_\pi \Phi r_k)) = 0 \\ &\implies (\Phi^T D \Phi) r_{k+1} = \Phi^T D g + \gamma \Phi^T D P_\pi \Phi r_k \\ &\implies r_{k+1} = (\Phi^T D\Phi)^{-1} b + \gamma (\Phi^T D \Phi)_{-1} \Phi^T D P_\pi \Phi r_k \implies \quad r_{k+1} = r_k + (\Phi^T D\Phi)^{-1} b + (\Phi^T D\Phi)^{-1}(\Phi^T D \end{aligned}$$

7 Events

- ☒ Quiz 1: Jan 19
- ☒ Midterm 1: Feb 16
- ☐ Midterm 2 and Quiz 2: Mar 30
- ☒ Assignment 1: Feb 04
- ☐ Assignment 1: Mar 19
- ☐ Project
- ☐ Endterm