ECE-GY 9223 Reinforcement Learning	Spring 2019
Lecture 8 — April 26, 2019	
Prof. Quanyan Zhu	*Scribe: Yunian Pan*

1 Overview

In the last lecture we talked about **approximate Q-Learning** in policy policy iteration and give some proof about feasiblility of the algorithm and some basic concept about **Temporal Difference**.

In this lecture we continued **TD learning**, then we talked about **Average cost problems** and the resulting **ACOE**, in the end we had a beginning of discussion over two types of **Bandit Problems—stochastic bandit** and **adversarial bandit** and ended up with the introduction of a few machineries.

2 TD learning

We begin by recalling the stochastic Bellman equation:

$$J^\mu(i_k) = \mathbb{E}(g(i_k, i_{k-1}) + \alpha J^\mu(i_{k+1})) \tag{1}$$

where $\alpha \in (0,1)$ or $\alpha = 1$, this is again a Robbins Monro problem, $r = E(g(r,v))$ for which, from a Robbin's Monro's perspective we can do:

$$J^+(i_k) = J(i_k) + \gamma \underbrace{(g(i_k, i_{k-1}) + \alpha J^\mu(i_{k+1}) - J(i_k))}_{TD}$$

where the expectation of TD term under condition i_k should be 0. We further analyze $refSB$'s generalization:

$$\begin{aligned} J^\mu(i_k) &= \mathbb{E}(g(i_k, i_{k+1}) + \alpha g(i_{k+1}, i_{k+2}) + \alpha^2 J^\mu(i_{k+2})) \\ &= \mathbb{E}(\sum_{m=0}^{l} \alpha^m g(i_{k+m}, i_{k+m+1}) + \alpha^{l+1} J^\mu(i_{k+l+1})) \end{aligned}$$

For simplicity we get rid of α and consdier the total cost case:

$$J^\mu(i_k)\, \mathbb{E}(\underbrace{\sum_{m=0}^{l} g(i_{k+m}, i_{k+m+1})}_{roll-out\ term} + J^\mu(i_{k+l+1})) \tag{2}$$

The trick here is that we can multiply 2 $(1-\lambda)\lambda^l$ and sum over l, and then interchange the order of summation such that we can make use of the identity $\sum_{l=m}^{\infty} \lambda^l = \frac{\lambda^m}{1-\lambda}$, , thus:

$$
\begin{aligned}
\implies &= \mathbb{E}[(1-\lambda)\sum_{m=0}^{\infty}\sum_{l=m}^{\infty}\lambda^l g(i_{k+m}, i_{k+m+1}) + \sum_{l=0}^{m}\lambda^l(1-\lambda)KJ^\mu(i_{k+l+1})] \\
&= \mathbb{E}[\sum_{m=0}^{\infty} g(i_{k+m}, i_{k+m+1})\sum_{l=m}^{\infty}(1-\lambda)\lambda^l + \sum_{l=0}^{m}(\lambda^l - \lambda^{l+1})KJ^\mu(i_{k+l+1})] \\
&= \mathbb{E}[\sum_{m=0}^{\infty}(g(i_{k+m}, i_{k+m+1})\lambda^m + \lambda^m J^\mu(i_{k+m+1})) - \sum_{l=0}^{m}\lambda^{l+1}KJ^\mu(i_{k+l+1})] \\
&= \mathbb{E}[\sum_{m=0}^{\infty}\lambda^m(g(i_{k+m}, i_{k+m+1}) + J^\mu(i_{k+m+1})) - \sum_{l'=1}^{m}\lambda^{l'}KJ^\mu(i_{k+l'})] \\
&= \mathbb{E}[\underbrace{\sum_{m=0}^{\infty}\lambda^m(g(i_{k+m}, i_{k+m+1}) + J^\mu(i_{k+m+1} - J^\mu(i_{k+m})))}_{d_{k+m}\ TD} + J^\mu(i_k) \\
&= \underbrace{\mathbb{E}[\sum_{m=0}^{\infty}\lambda^m d_{k+m}]}_{should\ be\ equal\ to\ 1} + J^\mu(i_k) \\
&= J^\mu(i_k)
\end{aligned}
$$

where we define TD error term $d_m = g(i_m, i_{m+1}) + J^\mu(i_{m+1}) - J^\mu(i_m)$, resulting $TD(\lambda)$ algorithm, i.e. we do:

$$
\begin{aligned}
&J^+(i_k) = J(i_k) + \gamma\sum_{m=0}^{\infty}\lambda^m d_{k+m} \quad or\ if\ consider\ discounted\ case \\
&J^+(i_k) = J(i_k) + \gamma\sum_{m=0}^{\infty}(\alpha\lambda)^m d_{k+m}
\end{aligned}
$$

Fact.
For TD(λ) algorithm:
if $\lambda = 1$, we are doing value iteration;
if $\lambda = 0$, we are doing policy improvement.

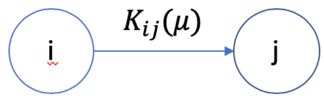
3 Average cost problem

Suppose we are interested in the average cost, i.e.

$$
\bar{J}(i) = \lim_{N\to\infty}\frac{1}{N}\mathbb{E}[\sum_{k=0}^{N-1} g(i_k, \mu(k), i_{k+1})|i_0 = i] \tag{3}
$$

which can be very dangerous since

- the limit may not exist



- the stationary policy may not be globally optimal
- For example

$$J^\mu = \underbrace{\lim_{n\to\infty} \frac{1}{N}\mathbb{E}[\sum_{k=1}^{K_{ij}(\mu)-1} g(i_k, \mu(k), i_{k+1})]}_{(1)} + \underbrace{\frac{1}{N}\mathbb{E}[\sum_{k=K_{ij}(\mu)}^{N-1} g(i_k, \mu(k), i_{k+1})|i_0 = i]}_{(2)}$$

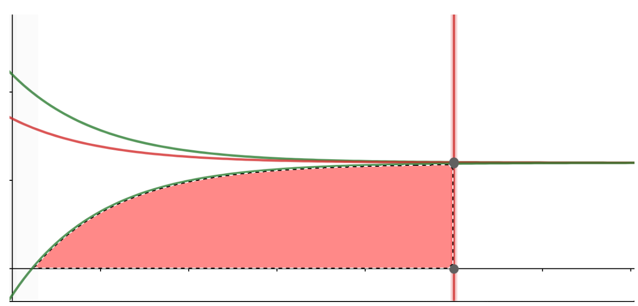
$\mathbb{E}(K_{ij}(\mu)) < \infty \implies (1) \to 0$, then J^μ should be indep. of the initial state, this is true if under a given policy μ, there is a state that can be reached from all other states with probability 1.

remark 1. *The (DP) equation:*

$$J(i) = \min \sum_j P_{ij}(\mu)(g(i,\mu,j) + \alpha J(j))$$

might not be correct

remark 2. *When we are only interested in average cost, i.e.* $\frac{1}{N}\sum_{k=0}^{N-1} g(i_k, \mu(k), i_{k+1})$



What's under this curve doesn't really matter, it goes to 0 multiplying $\frac{1}{N}$.

Proposition 3. *If there exists some bounded function h defined on nonnegative integers and a consitant λ s.t.*

$$\lambda + h(i) = \min[\sum_j P_{ij}(u)(g(i,u,j) + h(j))] \qquad ACOE(\star) \tag{4}$$

Then there exists a stationary policy μ_A such that

$$\lambda = \inf_\mu J_\mu(i) \quad \forall i \geq 0 \tag{5}$$

μ_A is the policy for each i, selects an action that minimizes the RHS of ACOE

3.1 Example

① A crowdsourcing worker is presented with job i w.p. p_i.

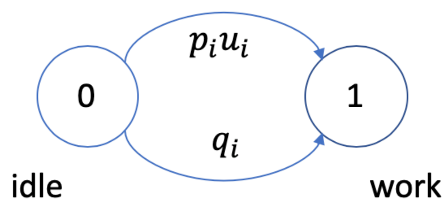
② A job of type i can be completed in a time slot w.p. q_i.

③ A reward r_i is received for completing a job of type i.

④ When taking job, one cannot take another job.

Q: Find the optimal policy/strategy to accept jobs to maximize the average expected reward.

First we have to do some modeling of the problem:



where

$$u_i = \begin{cases} 1 & \text{if job of type i is accepted} \\ 0 & \text{otherwise} \end{cases}$$

(ACOE) here

- $i = 0$, idling $\star$ $h(0) + \lambda = 0 + \sum_{i=1} p_i max\{ \overbrace{h(i)}^{accepted}, \underbrace{h(0)}_{reject} \}$
- $i = 1, 2, \ldots$, accept job i, $\star\star$ $h(i) + \lambda = \underbrace{q_i(r_i + h(0))}_{task\ completed} + \underbrace{(1 - q_i)h(i)}_{task\ not\ completed}$

For $\star$ is we add a constant c to $h(i)$ for every i, it will not change anything, therefore we set $h(0) = 0$.

$$\star \quad \lambda = \sum_i p_i \max(0, h(i))$$

$$\star\star \quad h(i) + \lambda = q_i r_i + (1 - q_i)h(i)$$

$$\lambda = q_i(r_i - h(i))$$

$$\star\star\star \quad \lambda = \sum_i p_i(\max(0, r_i - \frac{\lambda}{q_i}))$$

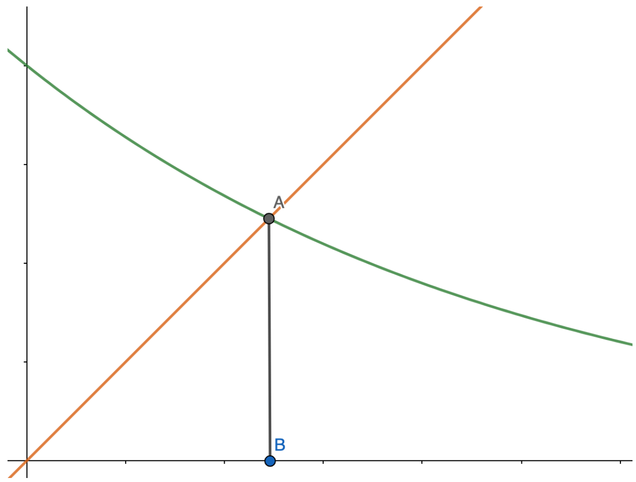
we have:

(1) λ is solution to a fixed point equation. (λ^*)

(2) the policy:

$$\begin{cases} \text{accept} & r_i \geq \frac{\lambda^*}{q_i} \\ \text{reject} & \text{otherwise} \end{cases} \tag{6}$$

The vertical axis represents $\sum_i p_i u_i$ and the horizontal axis represents λ, the solution exists for $h(i) = r_i - \frac{\lambda^*}{q_i}$.



There is a proof in [1] *Approximate Dynamic Programming Vol I*

4 Bandit Problems

(1) stochastic problem

(2) adversarial problem

Problem statements: K-arm. A carsino situation.

Consider k arms, each has an unknown distribution $\{\nu_k\}_{k=1,2,...}$ with values bounded in $[0,1]$, at each t, an agent pulls an arm $I_t \in \{1,\dots,K\}$ and observes a reward $X_t \sim \nu_{I_t}$ (i.i.d samples from ν_{I_t})

Objective: Maximize the expected sum of reward $\mathbb{E}(\sum_{t=1}^n X_t)$, the policy should be $\sigma : historical\ information \rightarrow some\ action.$

Here is the challenges: we don't know:

- ν_k
- mean of each arm: $\mu_k = \mathbb{E}_{X\sim\nu_k}[X]$
- mean of the best arm: $\mu^* = \max_k \mu_k$

Dynamic programming can hopefully solve this problem, at the very beginning we have to determine the states, we define a knowledge state: S^n and consider a thought experiment:

thought experiment 4. *There's only one arm, decide to continue or not to continue.*

thus our Bellman equation is:

$$V(S^n) = \max(\underbrace{V(S^n)}_{quit}, \underbrace{\mathbb{E}[w^{n+1} + V(S^n)|S^n]}_{continue}) \tag{7}$$

All of these make the problem extremely "hard" to solve, yet there are some genius people who demonstrated in a markovian framework that the optimal solution of the general case is an index policy whose "dynamic allocation index" is computable in principle for every state of each project as a function of the single project's dynamics called Gittins Index approach.[2]

Here's one question: Pick a strategy, can we evaluate it? Can we also compare with some nominated ν_k?

Define the regret:

$$R_n = n\mu^* - \sum_{t=1}^{n} X_t \tag{8}$$

The expectation of regret is taken w.r.t ,the sequence of the arms or, the randomness of arm and reward.

$$\mathbb{E}[R_n] = n\mu^* - \mathbb{E}[\sum_{t=1}^{n} X_t] \tag{9}$$

Let's be smarter in a way that instead of suming over the sequence X_t, we provide a measure couting from 1 to n.

$$T_k(n) = \sum_{t=1}^{n} \mathbb{1}(I_t = k) \tag{10}$$

which is a total number of times that I_t pulled up to time t. $\sum_{k=1}^{K} T_k(n) = n$. Thus in the regret 9,

$$\begin{aligned} RHS &= n\mu^* - \mathbb{E}[\sum_{k=1}^{K} T_k(n)] \\ &= \mathbb{E}[\sum_{k=1}^{K} T_k(n)(\mu^* - \mu_k)] \\ &= \mathbb{E}[\sum_{k=1}^{K} \underbrace{\Delta_\mu}_{gap}] \end{aligned}$$

and we have

- policy i: $\hat{\mu}_{k,s} = \frac{1}{s}\sum_{i=1}^{s} x_{k,i}$ up to time s, compute the empirical mean reward of arm k. and we choose $\underline{I_t = \arg\max_\mu \hat{\mu}_{k,s}}$, (Can we do better suppose the samples are huge and good? Yes.)

- policy ii: $I_t = \arg\max_\mu \hat{\mu}_{k,s} + prediction/correction$, when

  (1) s is large;

  (2) s is large w.r.t n

  i.e.

$$B_{t,s}(k) = \hat{\mu}_{k,s} + \sqrt{\frac{\alpha \log t}{s}} \tag{11}$$

$$I_t = \arg\max_{\mu} B_{t,T_k(t-1)}(k) \tag{12}$$

$$= \arg\max_{\mu} \frac{1}{T_k(t-1)} \sum_{i=1}^{T_k(t-1)} X_{\mu,i} + \sqrt{\frac{\alpha \log t}{T_k(t-1)}} \tag{13}$$

4.1 Some machineries

To get the things above we have to first introduce some apparatus.

- Markov Inequality
- Chernoff bound
- Hoeffding bound
- Chernoff bound

4.1.1 Markov Inequality

Theorem 5. *For a non-negative random variable X, the following Inequality holds for any $\epsilon > 0$.*

$$P(X \geq \epsilon) \leq \frac{\mathbb{E}(X)}{\epsilon} \tag{14}$$

Proof. Define indicator function $Y = \epsilon \mathbb{1}(X \leq \epsilon)$, $\mathbb{E}(Y) = \epsilon P(X \geq \epsilon) \leq \mathbb{E}(X)$, we are done. □

4.1.2 Chernoff bound

Theorem 6. *Consider a sequence of i.i.d R.V.s X_i, $\mu = \mathbb{E}(X_i)$, for any constance x,*

$$P(\sum_{i=1}^{n} X_i \geq nx) \leq \exp(-n \sup_{\theta \geq 0}(\theta x - \log M(\theta))) \tag{15}$$

where $M(\theta)$ is the M.G.F of X_i's

Proof. According to 14

$$
\begin{aligned}
P(\sum_{i=1}^{n} X_i \geq nx) \leq P(e^{\theta \sum_{i=1}^{n} X_i} \geq e^{\theta n x}) &\leq \frac{\mathbb{E}(e^{\theta \sum_{i=1}^{n} X_i})}{e^{\theta n x}} \\
&\leq \inf_{\theta \geq 0} e^{-\theta n x}\, \mathbb{E}(e^{\theta \sum_{i=1}^{n} X_i}) \\
&= \inf_{\theta \geq 0} e^{-n(\theta x - \log \mu(\theta))} \\
&= e^{-n \inf_{\theta \geq 0}(\theta x - \log \mu(\theta))}
\end{aligned}
$$

□

example. $\{X_i\}$ *are Bernoulli R.V's*

$$
\begin{aligned}
M(\theta) = E(e^{\theta x}) = pe^{\theta} + qe^{0} &= pe^{\theta} + 1 - p \\
\sup_{\theta}(\theta x - \log M(\theta)) = \sup_{\theta}(\theta x - \log(q + pe^{\theta})) &= D(x \| P) \\
&= x \log \frac{x}{p} + (1 - x) \log \frac{1 - x}{1 - p} \\
\theta^* &= \log \frac{n(1 - p)}{(1 - x)p}
\end{aligned}
$$

thus resulting the kL divergence.

4.1.3 Hoeffding bound

Theorem 7. *Consider a sequence of i.i.d R.V.s* X_i*,* $\mu = \mathbb{E}(X_i)$*,* X_i *takes values between* $[a_i, b_i]$*, then*

$$
\mathbb{P}(|\frac{\sum_i X_i - \mathbb{E}(\sum_i X_i)}{n}| \geq x) \leq \underbrace{2\exp(\frac{-2n^2x^2}{\sum_{i=1}^{n}(b_i - a_i)^2})}_{\star \text{ This will give us some predictions}} \tag{16}
$$

e.g. $\mathbb{P}(|\frac{1}{n}\sum_i X_i - \mu| \geq \epsilon) \leq 2\exp(-2n\epsilon^2)$

$$
|\min_f \frac{1}{n}\sum_{i=1}^{n} g(f(x_i), y_i) - \min_f \mathbb{E}(g(f(x), y))| \leq \epsilon
$$

References

[1] Bertsekas, Dimitri P. *" Approximate dynamic programming." (2008).*

[2] https://en.wikipedia.org/wiki/Gittins_index