ECE-GY 9223 Reinforcement Learning Spring 2019

Lecture 7 — April 12, 2019

Prof. Quanyan Zhu *Scribe: Zijie Xia*

1 Overview

In the last lecture we first talked about the **Lyapunov criteria for dynamic systems** which included two important theories: first is **Lyapunov global asymptotic stability theorem** and the second is **Lasalle's theorem**. Then we talked about **Q-learning** and Q-learning with value iteration and simulation. Finally, we talked about approximate Q-learning in value iteration.

In this lecture we are continue talking about **approximate Q-learning** in policy iteration and give some proof about feasible of the algorithm. In the end, we talked about some basic concept about **Temporal Difference**.

2 Approximate Q-learning

2.1 Recall Q-learning

According to the previous lectures, regarding a Markov decision process, we define a value function as following:

$$J(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)[g(i,u,j) + \alpha J(j)]$$

Define an operator to describe this operation of value function:

$$(TJ)(i) := \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)[g(i,u,j) + \alpha J(j)]$$

We get the Q-function here as:

$$Q(i,u) = \sum_{j=1}^{n} p_{ij}(u)[g(i,u,j) + \alpha J(j)]$$

Where $J(j) = \min_{u \in U(j)} Q(j,u)$. To solve this Q-learning problem, we are commonly use two methods: first is value iteration and the other is policy iteration. Here, we talked about the value iteration first.

2.1.1 Value Iteration

For the bellman equation, we use the value iteration and we can get:

$$\begin{aligned} Q^{k+1}(i,u) &= (1-\lambda)Q^k(i,u) + \lambda[\sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + \min_{v\in U(j)} Q^k(j,v))] \\ &\approx (1-\lambda)Q^k(i,u) + \lambda(g(i,u,\bar{j}) + \min_{v\in U(\bar{j})} Q^k(\bar{j},v)) \end{aligned}$$

Where $\bar{j}$ is the state after sampling. So for the approximate Q-learning in value iteration. We are evaluate with the approximate Q-function $\tilde{Q}(i,u,r)$ and we look for r and update r.

2.1.2 Policy Iteration

The main idea of policy iteration is evaluate the policy through the Q function and update the policy by choosing the policy which get the minimum Q value, which is :

Evaluate the policy μ^k *through* Q^k

Update the policy $\mu : \mu^{k+1}(i) \in arg\min_u Q^k(i,u)$

Iterate the instructions until finding the optimal policy μ^*

2.2 Approximate Q-learning in policy iteration

As in value iteration, we approximate a Q-function as following:

$$\begin{aligned} \tilde{Q}(i,u,r) &= \sum_{k=1}^{K} \phi_k(i,u) r_k \\ &= \phi^T \mathbf{r} = \mathbf{r}^T \phi \end{aligned}$$

Where K is the number of basis functions.
We are going to find the $\tilde{Q}$ which best approximate the Q-function. So we can define a function which describe the difference between the approximate Q-function and the real Q-function and find the minimum of that function:

$$\min_r \frac{1}{2} \sum_{(i,u)} w(i,u)(\mathbf{r}^T\phi(i,u) - Q(i,u))^2$$

$w(i,u)$ are weights. It can also be written as vector form:

$$\min_r \frac{1}{2} \|\mathbf{r}^T\phi - Q\|_{w,2}^2 \tag{1}$$

For a Markov decision process

$$(T^\mu Q)(i,u) = \mathbb{E}(g(i,u,j) + \alpha Q(j,v))$$

j is the next state and v is the action generated by $\mu(j)$
We have:

$$Q_{k+1} = T^{\mu}(Q_k) \tag{2}$$

Using $\mathbf{r}_k^T \phi$ to approximate each Q_k. Define a function to describe the difference of the real Q-function to the approximate Q-function:

$$c(r; Q_{k+1}) = \frac{1}{2}\|\mathbf{r}^T\phi - Q\|_{w,2}^2 r_{k+1} = arg\min_r \frac{1}{2}\|\mathbf{r}^T\phi - Q\|_{w,2}^2$$

To find the minimum difference, we need to calculate:

$$\Delta_r c(r; Q_{k+1}) = 0$$

The total gradient is

$$\sum_{(i,u)} w(i,u)(\mathbf{r}^T\phi(i,u) - Q_{k+1}(i,u))\phi(i,u) = 0$$

$$\sum_{(i,u)} w(i,u)(\mathbf{r}^T\phi(i,u) - \mathbb{E}[g(i_k,\mu_k,j) + \alpha Q_k(j,v)])\phi(i,u) = 0$$

$$\sum_{(i,u)} w(i,u)(\mathbf{r}^T\phi(i,u) - \mathbb{E}[g(i_k,\mu_k,j) + \alpha r_k^T\phi(j,v)])\phi(i,u) = 0$$

So, we can get the r_{k+1}

$$r_{k+1} = r_k - \epsilon_k(r_k^T\phi(i_k,u_k) - \mathbb{E}(g(i_k,u_k,j) + \alpha r_k^T\phi(j,v))\phi(i_k,u_k)) \tag{3}$$

With Robins-Monro algorithm

$$r_{k+1} = r_k - \epsilon_k(r_k^T\phi(i_k,u_k) - g(i_k,u_k,\tilde{j}) - \alpha r_k^T\phi(\tilde{j},\tilde{v}))\phi(i_k,u_k) \tag{4}$$

$\tilde{j}$ is the sampled by the Robins-Monro algorithm and $\tilde{v}$ is generated by the policy, $\tilde{v} = \mu(\tilde{j})$ and $r_k^T\phi(i_k,u_k) - g(i_k,u_k,\tilde{j}) - \alpha r_k^T\phi(\tilde{j},\tilde{v})$ is so called **Temporal Difference**. Consider a projection operator$P(z)$, equation 1 can be written as

$$\min \frac{1}{2}\|\hat{Q} - z\|_{\pi,2}^2$$

Where

$$\hat{Q} = \theta^T\phi$$

$$\hat{Q}_{k+1} = P(T_\mu(\hat{Q}_k)) = (P \cdot T)\hat{Q}_k$$

π is the weights $w(i,u)$
Claim: $P \cdot T_\mu$ is a contraction with respect to $\|\hat{Q} - z\|_{\pi,2}$ We separate into two step to prove the claim, first is to prove that the projection P is nonexpansive, which means $\|P(\hat{Q}_1) - P(\hat{Q}_2)\|_{\pi,2} \leq \|\hat{Q}_1 - \hat{Q}_2\|_{\pi,2}$, and the second is prove that T_μ is contraction. Proof P is nonexpansive:
From the optimality condition for convex optimization problem,

$$< P(\hat{Q}_1) - \hat{Q}_1, Q_1 - P(\hat{Q}_1) >_\pi \geq 0 \quad \forall Q_1 \in R \; and \; \hat{Q}_1$$

Then we have

$$< P(\hat{Q}_1) - \hat{Q}_1, P(\hat{Q}_2) - P(\hat{Q}_1) >_\pi \geq 0 \tag{5}$$

$$< P(\hat{Q}_2) - \hat{Q}_2, P(\hat{Q}_1) - P(\hat{Q}_2) >_\pi \geq 0 \tag{6}$$

Add equation 5 and 6

$$\begin{aligned}
&< P(\hat{Q}_1) - \hat{Q}_1 + \hat{Q}_2 - P(\hat{Q}_2), P(\hat{Q}_2) - P(\hat{Q}_1) >_\pi \geq 0 \\
&< P(\hat{Q}_1 - P(\hat{Q}_2), P(\hat{Q}_2) - P(\hat{Q}_1) >_\pi + < \hat{Q}_2 - \hat{Q}_1, P(\hat{Q}_2) - P(\hat{Q}_1) >_\pi \\
&- \|P(\hat{Q}_1 - P(\hat{Q}_2)\|_{\pi,2}^2 + < \hat{Q}_2 - \hat{Q}_1, P(\hat{Q}_2) - P(\hat{Q}_1) >_\pi
\end{aligned}$$

According to Cauchy-Schwarz inequality

$$< \hat{Q}_2 - \hat{Q}_1, P(\hat{Q}_2) - P(\hat{Q}_1) >_{pi} \geq \|P(\hat{Q}_1 - P(\hat{Q}_2)\|_{\pi,2}^2$$

We can get

$$\|\hat{Q}_2 - \hat{Q}_1\|_{\pi,2} \|P(\hat{Q}_2) - P(\hat{Q}_1)\|_{\pi,2} \ \geq < \hat{Q}_2 - \hat{Q}_1, P(\hat{Q}_2) - P(\hat{Q}_1) >_\pi$$

Done.

Proof T_μ is contraction

$$\|T_\mu(Q_1) \cdot T_\mu(Q_2)\|_{\pi,2}^2 = \|\alpha \sum_j P_{ij}(u)(Q_1(j,v) - Q_2(j,v))\|_{\pi,2}^2 \tag{7}$$

$$= \|\alpha P(\theta_1 - \theta_2)\|_{\pi,2}^2 \tag{8}$$

Where u is generated by $u = \mu(i)$, v is generated by $v = \mu(j)$, P is transition matrix.
According to Jensen's inequality, which is for convex function $f(x)$, we have $\mathbb{E}(f(x)) \geq f(\mathbb{E}(x))$, so equation 8 can be written as

$$\begin{aligned}
\|T_\mu(Q_1) \cdot T_\mu(Q_2)\|_{\pi,2}^2 =& \alpha \sum_i \pi_i (\sum_j P_{ij}(Q_{1,j} - Q_{2,j}))^2 \\
\leq& \alpha \sum_i \pi_i \sum_j P_{ij}(Q_{1,j} - Q_{2,j})^2 \\
=& \alpha \sum_j (Q_{1,j} - Q_{2,j})^2 \sum_i \pi_i P_{i,j} \\
=& \alpha \|Q_1 - Q_2\|_{\pi,2}^2
\end{aligned}$$

T_μ is contraction
So that $P \cdot T$ is a contraction mapping.
Now go back to approximate Q-learning algorithm. From equation 4, make some simplicity

$$\begin{aligned}
\phi(i_k, u_k) &= \phi(i_k, \mu(i_k)) \\
&= \phi(i_k) \\
g(i_k, u_k, \tilde{j}) &= g(i_k, \tilde{j})
\end{aligned}$$

ODE Approach

$$\dot{\mathbf{r}} = -f(\mathbf{r})$$

$$f(\mathbf{r}) = \mathbb{E}(\mathbf{r}^T \phi(i) - g(i,j) - \alpha \mathbf{r}^T \phi(j))\phi(i) \tag{9}$$

$$= \sum_i \pi_i \phi(i)(\mathbf{r}^T \phi(i) - \sum_j P_{ij}(g(i,j) + \alpha \mathbf{r}^T \phi(j))) \tag{10}$$

Where

$$\sum_j P_{ij}(g(i,j) + \alpha \mathbf{r}^T \phi(j)) = T_\mu(\mathbf{r}^T \phi(j))$$

$$= T_\mu \begin{bmatrix} \mathbf{r}^T\phi(1) \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{r}^T\phi(1) \end{bmatrix}$$

$$= T_\mu(\Phi_r)(i)$$

So that

$$f(\mathbf{r}) = \sum_i \pi_i \phi(i)(\mathbf{r}^T\phi(i) - T_\mu(\Phi_r)(i)) \tag{11}$$

Where

$$\Phi = \begin{bmatrix} \phi_1(1) & \phi_2(1) & \dots & \phi_n(1) \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \phi_1(|S|) & \phi_2(|S|) & \dots & \phi_n(|S|) \end{bmatrix}_{s \times n} = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(|S|) \end{bmatrix} \tag{12}$$

Where $s = |S|$ is the number of states. Define:

$$\Phi\mathbf{r} = \begin{bmatrix} \phi^T(1)\mathbf{r} \\ \phi^T(2)\mathbf{r} \\ \vdots \\ \phi^T(|S|)\mathbf{r} \end{bmatrix}_{s \times 1} \tag{13}$$

$$D = diag(\pi) = \begin{bmatrix} \pi_1 & & & \\ & \pi_2 & & \\ & & \ddots & \\ & & & \pi_{|S|} \end{bmatrix} \tag{14}$$

So that the Equation 10 can be written as

$$f(\mathbf{r}) = \Phi^T D \Phi\mathbf{r} - \Phi^T D T(\Phi\mathbf{r}) \tag{15}$$

Let $f^*(\mathbf{r}) = 0$

$$\Phi^T D \Phi\mathbf{r} - \Phi^T D T(\Phi\mathbf{r}) = 0$$

Which is

$$\mathbf{r}^* = (\Phi^T D \Phi)^{-1} \Phi^T D T(\Phi\mathbf{r}^*)$$
$$= P \cdot T(\Phi\mathbf{r}^*)$$

We can get the same conclusion with

$$\min_r \frac{1}{2} \|\Phi\mathbf{r} - x\|^2_{\pi,2}$$
$$= \min_r \frac{1}{2} (\Phi\mathbf{r} - x)^T D (\Phi\mathbf{r} - x)$$
$$First\ Order\ Condition \Rightarrow \mathbf{r} = (\Phi^T D \Phi)^{-1} \Phi^T D x$$

$\mathbf{r}^*$ is unique due to **Contraction Mapping Theorem**

$$\begin{aligned}\dot{\mathbf{r}} &= -f(\mathbf{r}) \\ &= -f(\mathbf{r}) + f(\mathbf{r}^*) \\ &= -\Phi^T D\Phi\mathbf{r} + \Phi^T DT(\Phi\mathbf{r}) + \Phi^T D\Phi\mathbf{r}^* - \Phi^T DT(\Phi\mathbf{r}^*) \\ &= -\Phi^T D\Phi(\mathbf{r} - \mathbf{r}^*) + \Phi^T D(T(\Phi\mathbf{r}) - T(\Phi\mathbf{r}^*))\end{aligned}$$

To proof the convergence, using Lyapunov method. First construct a Lyapunov function as follow:

$$V = \frac{1}{2}(\mathbf{r} - \mathbf{r}^*)^T(\mathbf{r} - \mathbf{r}^*)$$

Take the derivative of the Lyapunov function

$$\begin{aligned}\dot{V} &= -(\mathbf{r} - \mathbf{r}^*)^T\Phi^T D\Phi(\mathbf{r} - \mathbf{r}^*) + (\mathbf{r} - \mathbf{r}^*)\Phi^T D(T(\Phi\mathbf{r}) - T(\Phi\mathbf{r}^*)) \\ &\leq -\|\Phi(\mathbf{r} - \mathbf{r}^*)\|_{\pi,2}^2 + \|\Phi(\mathbf{r} - \mathbf{r}^*)\|_{\pi,2}\|T(\Phi\mathbf{r}) - T(\Phi\mathbf{r}^*)\|_{\pi,2} \\ &\leq -(1-\alpha)\|\Phi(\mathbf{r} - \mathbf{r}^*)\|^2\end{aligned}$$

$\dot{V} \leq 0$ and $\dot{V} = 0$ if and only if $\mathbf{r} = \mathbf{r}^*$ so that it will converge to $\mathbf{r}^*$.

There are following three remarks:
Remark 1:
Q-value here is so called **Actor-Critique** We choose policy from

$$\mu(i) = arg\min_u Q_k(i,u) \leftarrow Actor$$

Remark 2:
From equation 4, Temporal Difference is defined as follow:

$$r_{k+1} = r_k - \epsilon_k(r_k^T\phi(i_k,u_k) - g(i_k,u_k,\tilde{j}) - \alpha r_k^T\phi(\tilde{j},\tilde{v}))\phi(i_k,u_k)$$

Define

$$\delta_k := r_k^T\phi(i_k,u_k) - g(i_k,u_k,\tilde{j}) - \alpha r_k^T\phi(\tilde{j},\tilde{v})$$

So equation 4 can be written as:

$$r_{k+1} = r_k - \epsilon_k\delta_k\phi(i_k,u_k)$$

Where δ_k is **Temporal Difference Remark 3**:
Extension to TD(λ)

$$z_k = \sum_{t=0}^{n}\lambda^k\phi(i_k,t)$$

3 Temporal Difference

Consider a Bellman equation

$$J^\mu = T_\mu J^\mu$$

Using Monte Carlo sampling

$$J^{\mu} = g(i_0, i_1) + \alpha g(i_1, i_2) + \alpha^2 g(i_2, i_3) + \dots$$

Which

$$J^{\mu}(i_0) \approx \frac{1}{K} \sum_{m=1}^{K} c(i, m)$$

So that

$$J_{m+1}(i) = J_m(i) + \gamma_m(c(i, m) - J_m(i)) \quad m = 1, 2, \dots \tag{16}$$

Where $\gamma_m = \frac{1}{m}$ From equation 16 we can get

$$\begin{aligned} J_{m+1}(i_k) &= J_m(i_k) + \gamma_m(i_k)(g(i_k, i_{k+1}) + \alpha g(i_{k+1}, i_{k+2}) + \alpha^2 g(i_{k+2}, i_{k+3}) + \dots - J_m(i)) \\ &\quad After\ Filtering \\ &= J_m(i_k) + \gamma(g(i_k, i_{k+1}) + \alpha J_m(i_{k+1}) - J_m(i) + \alpha g(i_{k+1}, i_{k+2}) + \alpha^2 J_m(i_k + 2) \\ &\quad - \alpha J_m(i_{k+1}) + \dots) \end{aligned}$$

Like $g(i_k, i_{k+1}) + \alpha J_m(i_{k+1}) - J_m(i)$ is so called **Temporal Difference**

References

[1] Borkar, Vivek S. Stochastic approximation: a dynamical systems viewpoint. Vol. 48. *Springer*, 2009.