Statistical Techniques in Robotics (16-831, S22)	Lecture #14 (Monday, March 14)
RL & MDP	
Lecturer: Kris Kitani	*Scribes: Yimin Tang, Zilin Si*

1 Review

In the last lecture, we learnt the Thompson Sampling in the Stochastic environment, EXP3 and EXP4 in the adversarial environment which is context-free and contextual respectively.

1.1 Thompsons Sampling: Beta-Bernoulli Bandit

We use beta-bernoulli distribution to estimate the each arm's reward function since these two are conjugate. Therefore the algorithm is shown as below:

Algorithm 1 Bern-Beta Thompsons Sampling

1: **for** $t = 1...T$ **do**
2: $\theta_t \sim p(\theta; \alpha_k, \beta_k), \forall k$ ▷ sample from posterior
3: $a^{(t)}_{\hat{k}} = argmax_k \mathbb{E}_{p(r|a_k,\theta_k)}[r|a_k, \theta_k]$ ▷ predict
4: $REVEIVE(r^{(t)})$ ▷ get reward
5: $\alpha_{\hat{k}} = \alpha_{\hat{k}} + r^{(t)}$
6: $\beta_{\hat{k}} = \beta_{\hat{k}} + 1 - r^{(t)}$ ▷ update posterior
7: **end for**

The regret of Thompson Sampling is $O(\sqrt{KTlogT})$

1.2 EXP3

EXP3 stands for "exponential-weight update algorithm for exploration and exploitation".

Algorithm 2 EXP3($\gamma \in [0, 1]$)

1: $\omega^{(1)} \leftarrow \{\omega^{(1)}_k = 1\}^K_{k=1}$ ▷ wights over actions
2: **for** $t = 1...T$ **do**
3: $p^{(t)} = \frac{\omega^{(t)}}{\sum_k \omega^{(t)}_k}$ ▷ probability over actions
4: $k \sim MULTINOMIAL(p^{(t)})$
5: $a^{(t)} = a_k$ ▷ take and draw action
6: $REVEIVE(r^{(t)} \in [0, 1])$ ▷ get reward
7: $\omega^{(t+1)}_k = \omega^{(t)}_k exp\{\gamma \cdot r^{(t)}/p^{(t)}_k\}$ ▷ update weight for one arm
8: **end for**

where the update term $r^{(t)}/p^{(t)}_k$ is the unbiased estimator.

The regret of EXP3 is $O(\sqrt{TKlogK})$ and it is a no regret algorithm.

1.3 EXP4

EXP4 stands for "exponential-weight update algorithm for exploration and exploitation with experts".

Algorithm 3 EXP4($\gamma \in [0,1], T$)

$\omega^{(1)} \leftarrow \{\omega_k^{(1)} = 1\}_{k=1}^{K}$ ▷ wights over experts
for $t = 1...T$ **do**
  $RECEIVE(X^{(t)} \in \mathbb{R}^{N \times K})$ ▷ advice from N experts
  $q^{(t)} = \frac{\omega^{(t)}}{||\omega^{(t)}||} \cdot X^{(t)} \in \Delta^K$ ▷ probability over actions
  $k^{(t)} \sim MULTINOMIAL(q^{(t)})$ ▷ draw action
  $RECEIVE(r^{(t)})$ ▷ get reward
  $\hat{r}^{(t)} = \frac{r^{(t)}}{q_k^{(t)}} \mathbb{I}[k = k^{(t)}] \in \mathbb{I}^K$ ▷ reward over all arms
  $g^{(t)} = X^{(t)} \cdot \hat{r}^{(t)} \in \mathbb{R}^N$ ▷ per expert reward
  $\omega_n^{(t+1)} = \omega_n^{(t)} exp\{\gamma \cdot g_n^{(t)}\} \forall n$ ▷ update weight for all arms
end for

The regret of EXP4 is $O(\sqrt{TKlogN})$ where K is the number of arms, T is the time, and N is the number of experts.

2 Summary

2.1 Sequence Feedback Learning Problems

To distinguish the sequence feedback or one-shot feedback, a good indicator would be whether the data generation (feedback) a sequentially dependent process.

One-Shot Feedback Supervised learning is one kind of one-shot feedback. If we draw samples from the data distribution as

$$x, a \sim D(x, a) \tag{1}$$

here we assume x is the state and a is the action. then we can get a set of identically and independently distributed (i.i.d) samples as $\{(x_1, a_1), (x_2, a_2), ..., (x_N, a_N)\}$. This means the action from the previous sample a_{i-1} won't affect the state in the next sample x_i. For one-shot feedback, we don't need to worry about issues like co-variate shift or temporal credit assignment.

Sequence Feedback Reinforcement learning is one kind of sequence feedback. If we draw samples from the data distribution as

$$\zeta, R \sim D(\zeta, R) \tag{2}$$

where $\zeta = \{(x_1, a_1), (x_2, a_2), ..., (x_N, a_N)\}$ is a trajectory of samples where all samples are correlated, $R_i \in \mathbb{R}$ is the reward for the entire sequence. Under this circumstance, the action affects the next

Problem	Sampled	Evaluative	Sequential
PWEA	$\times$	$\times$	$\times$
OLC/OMD	$\surd$	$\triangle$	$\times$
3 MAB	$\times$	$\surd$	$\times$
C-MAB	$\surd$	$\surd$	$\times$
RL	$\surd$	$\surd$	$\surd$
IL	$\surd$	$\surd$	$\triangle$

Table 1: Table for decision-making problems classification.

state and we need to address covariate shift, temporal credit assignment, very large "trajectory" space.

2.2 Review of Learning Problems

As shown in the Table 1, we classified the decision-making problems covered so far into sampled/exhausted, evaluative/instructive, sequential/one-shot. Note that:
1) For PWEA, the loss function is fully observed so all parameters could be updated at every step. So it is instructive.
2) For MAB, C-MAB, the reward function is only partial observed so we could only update one (arm) parameter at a time. So it is evaluative.
3) For all problems including PWEA, OLC/OMD, MAB, C-MAB, their feedback is one-shot.
4) For sequential problem such as RL, we have to reason about the impact of decisions on the entire sequence including the future. So we obtain a sequence of rewards, update the future predictor (value function), and then update the action predictor (policy).

2.3 Markov Decision Process

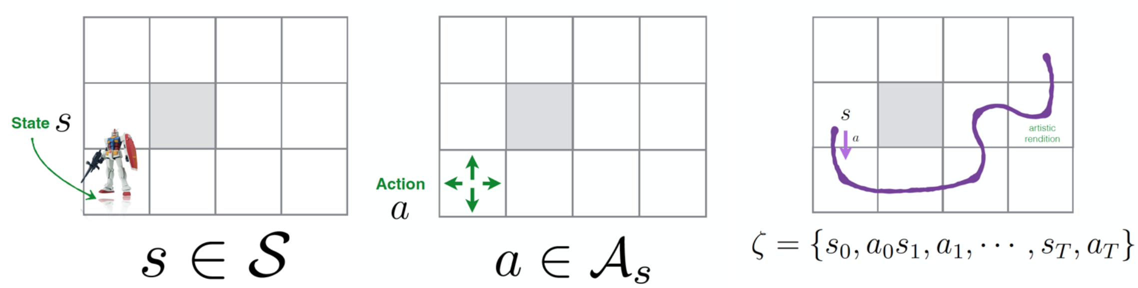

Figure 1: State s, action a, and trajectory ζ defined in a grid world.

First we use a grid world as example to define the components of markov decision process (MDP) as shown in Fig. 1. Each grid is one state s. Each movement is one action a including moving up, moving down, moving left and moving right. A trajectory is a sequence of states and actions as $\zeta = \{s_0, a_0, s_1, a_1, ..., s_T, a_T\}$.

Here we consider the joint distribution

$$p(s_0, a_0, s_1, a_1, ..., s_T, a_T) \tag{3}$$

as the probability of a state-action trajectory. We can factorize it as

$$p(s_0, a_0, s_1, a_1, ..., s_T, a_T) = p_0(s_0) \prod_t p(s_{t+1}|s_t, a_t) p(a_t|s_t) \tag{4}$$

where $p_0(s_0)$ is the prior state. $p(s_{t+1}|s_t, a_t)$ is the state transition dynamic which describe the probability of transition to another state. $p(a_t|s_t)$ is the policy which describes which action to take in a given state which can be stochastic or deterministic.

And a reward function

$$r(s_0, a_0, s_1, a_1, ..., s_T, a_T) \tag{5}$$

as a scalar value for one trajectory. We can factorize it as

$$\begin{aligned} r(s_0, a_0, s_1, a_1, ..., s_T, a_T) &= r(s_0, a_0, s_1) + r(s_1, a_1, s_2) + ... \\ &\Leftrightarrow r(s_0, a_0) + r(s_1, a_1) + ... \\ &\Leftrightarrow r(s_0) + r(s_1) + ... \end{aligned} \tag{6}$$

where $r(s)$ maps a state to a real value.

We can model the markov decision process as a graphical model as shown in Fig. 2.

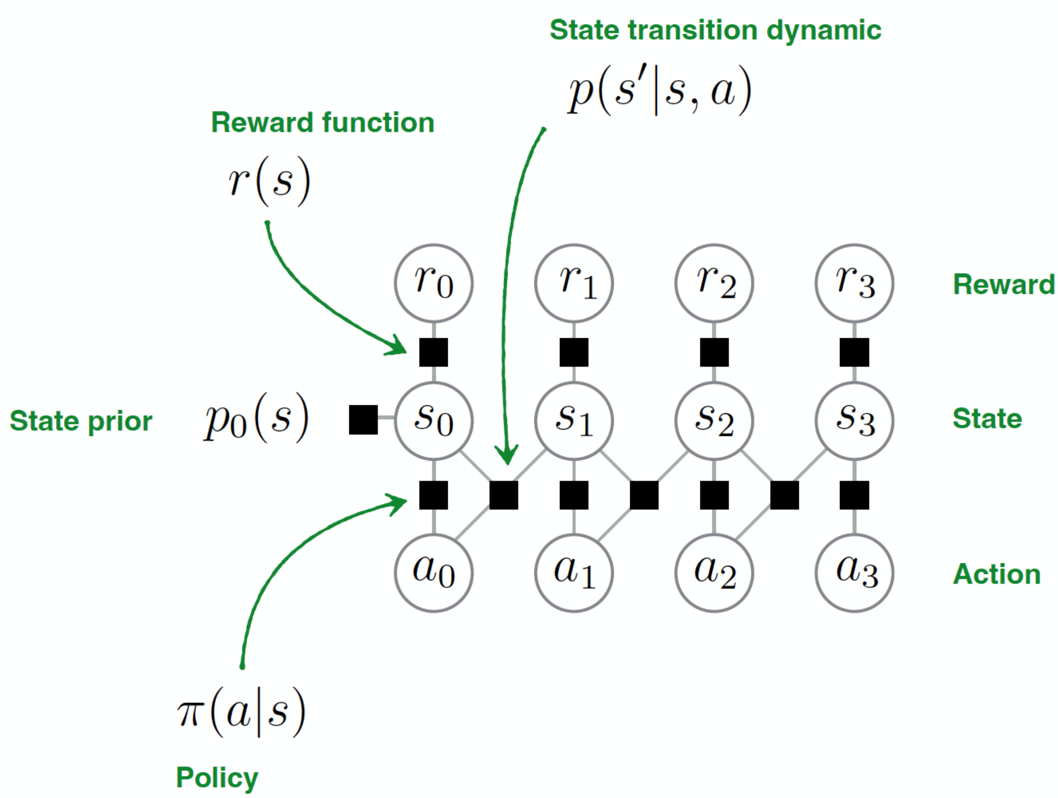

Figure 2: Graphical model for markov decision process.

2.4 Mathematic of the MDP

2.4.1 Value Function

We can define a value function for one policy on states:

$$V^{\pi}(s)=\mathbb{E}_p\left[r_0+r_1+r_2+\cdots \mid s_0=s\right] \tag{7}$$

π is our policy, $r_0, r_1, r_2, \cdots$ is all rewards for one trajectory generated by our policy and the environment when start state is s, and p is the trajectory probability which can be writen by:

$$p\left(s_0, a_0, s_1, a_1, \cdots\right)=p_0\left(s_0\right) p\left(s_1 \mid s_0, a_0\right) p\left(a_0 \mid s_0\right) p\left(s_2 \mid s_1, a_1\right) p\left(a_1 \mid s_1\right) \cdots \tag{8}$$

Notice $r_0, r_1, r_2, \cdots$, we can choose different time horizons:

- **Infinite horizon return**: $V^{\pi}(s)=\mathbb{E}_p\left[r_0+r_1+r_2+\cdots \mid s_0=s\right]$
- **Finite horizon return**: $V^{\pi}(s)=\mathbb{E}_p\left[r_0+r_1+r_2+\cdots+r_T \mid s_0=s\right]$
- **Infinite horizon discounted return**: $V^{\pi}(s)=\mathbb{E}_p\left[\gamma^0 r_0+\gamma^1 r_1+\gamma^2 r_2+\cdots \mid s_0=s\right]$

We can also define state-action value function based on our policy in infinite horizon discounted return form:

$$Q^{\pi}(s, a)=\mathbb{E}_p\left[\gamma^0 r\left(s_0\right)+\gamma^1 r\left(s_1\right)+\gamma^2 r\left(s_2\right)+\cdots \mid s_0=s, a_0=a\right] \tag{9}$$

Consider $Q^{\pi}(s, a)$ and $V^{\pi}(s)$, we can get:

$$\begin{aligned}
V^{\pi}(s) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0=s\right] \\
&= \sum_{s_{1:\infty}, a_{0:\infty}} p\left(s_{1:\infty}, a_{0:\infty}\right)\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0=s\right] \\
&= \sum_{s_{1:\infty}, a_0:\infty} \pi\left(a_0 \mid s_0=s\right) p\left(s_{1:\infty}, a_{1:\infty}\right)\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0=s\right] \\
&= \sum_a \pi\left(a_0=a \mid s_0=s\right) \sum_{s_{1:\infty}, a_{1:\infty}} p\left(s_{1:\infty}, a_{1:\infty}\right)\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0=s, a_0=a\right] \\
&= \sum_a \pi\left(a_0=a \mid s_0=s\right) \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0=s, a_0=a\right] \\
&= \sum_a \pi(a \mid s) Q^{\pi}(s, a)
\end{aligned}$$

2.4.2 Bellman Equation

We can change:

$$V^\pi(s)=\sum_a \pi(a\mid s)Q^\pi(s,a)$$

to:

$$V^\pi(s)=\sum_a \pi(a\mid s)\sum_{s'} p\left(s'\mid s,a\right)\left[r\left(s',a,s\right)+\gamma V^\pi\left(s'\right)\right] \tag{10}$$

First we can change Q to:

$$\begin{aligned}
Q^\pi\left(s_0,a_0\right) &= \mathbb{E}\left[\gamma^0 r_0+\gamma^1 r_1+\gamma^2 r_2+\cdots\mid s_0,a_0\right]\\
&= \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_t\mid s_0,a_0\right]\\
&= \mathbb{E}\left[r_0+\sum_{t=1}^{\infty}\gamma^t r_t\mid s_0,a_0\right]\\
&= \sum_{s_{1:\infty}} p\left(s_{1:\infty},a_{1:\infty}\right)\left[r_0+\sum_{t=1}^{\infty}\gamma^t r_t\mid s_0,a_0\right]\\
&= \sum_{s_1} p\left(s_1\mid s_0,a_0\right)\left\{r_0+\mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^t r_t\mid s_1\right]\right\}\\
&= \sum_{s_1} p\left(s_1\mid s_0,a_0\right)\left\{r_0+\gamma V^\pi(s_1)\right\}
\end{aligned}$$

So we get our V value:

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

And then we continue to change Q value:

$$\begin{aligned}
Q^\pi\left(s_0,a_0\right) &= \sum_{s_1} p\left(s_1\mid s_0,a_0\right)\left\{r_0+\sum_{a_1}\pi\left(a_1\mid s_1\right)\mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^t r_t\mid s_1,a_1\right]\right\}\\
&= \sum_{s_1} p\left(s_1\mid s_0,a_0\right)\left\{r_0+\sum_{a_1}\pi\left(a_1\mid s_1\right)Q^\pi\left(s_1,a_1\right)\right\}
\end{aligned}$$

For summary:

$$V^\pi(s)=\sum_a \pi(a\mid s)\sum_{s'} p\left(s'\mid s,a\right)\left[r\left(s',a,s\right)+\gamma V^\pi\left(s'\right)\right] \tag{11}$$

$$Q^\pi(s,a)=\sum_{s'} p\left(s'\mid s,a\right)\left\{r\left(s',a,s\right)+\sum_{a'}\pi\left(a'\mid s'\right)Q^\pi\left(s',a'\right)\right\} \tag{12}$$

2.4.3 Bellman Optimality Equations

We can have a max policy π^* and calculate its value functions:

$$\begin{aligned} V^{\pi^*}(s) &= \max_\pi V^\pi(s) \quad \forall s \\ Q^{\pi^*}(s,a) &= \max_\pi Q^\pi(s,a) \quad \forall s,a \end{aligned} \tag{13}$$

If our max policy π^* is optimal, we can rewrite our value equations to:

$$\begin{aligned} V^{\pi^*}(s) = \max_a \sum_{s'} p\left(s' \mid s,a\right)\left[r_t + \gamma V^{\pi^*}\left(s'\right)\right] \\ Q^{\pi^*}(s,a) = \sum_{s'} p\left(s' \mid s,a\right)\left[r(s) + \gamma \max_{a'} Q^{\pi^*}\left(s',a'\right)\right] \end{aligned} \tag{14}$$

Proof:

$$\begin{aligned} V^{\pi^*}(s) &= \sum_a \pi(a \mid s) Q^{\pi^*}(s,a) \\ &= \max_a Q^{\pi^*}(s,a) \\ &= \max_a \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a\right] \\ &= \max_a \mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a\right] \\ &= \max_a \sum_{s'} p\left(s_1 = s' \mid s,a\right)\left[r_0 + \mathbb{E}\left\{\sum_{t=1}^{\infty} \gamma^t r_t \mid s_1 = s'\right\}\right] \\ &= \max_a \sum_{s'} p\left(s' \mid s,a\right)\left[r_0 + \gamma V^{\pi^*}\left(s'\right)\right] \end{aligned}$$