# RL Discussion Notes

## Table of contents



## 1 Markov decision process

Standard RL involves 2 components:

- Agent (learned)
- Environment (user-defined)

Formally, at every timestep $t$, the agent receives $S_t$ (obtained from the environment), takes action $A_t$ (gets sent to the environment), receives reward $R_{t+1}$.

Here, $S_t$, $A_t$ and $R_{t+1}$ are all random variables.

Here's an example using Python API:

```python
env = gym.make("CartPole-v0")
state = env.reset()
while True:
    action = agent(state)
    next_state, reward, done, _ = env.step(action)
    if done:
        break
    state = next_state
```

We assume that the set of values $S_t$ can take on, $\mathcal{S}$, is finite. Similarly, we assume that the set of values $A_t$ can take on, $\mathcal{A}$, is also finite. Aso, we assume that

$$S_{t+1} | S_t, A_t, S_{t-1}, A_{t-1}, \ldots = S_{t+1} | S_t, A_t$$

which is called the *Markov property*. In other words, $S_t$ must be completely summarize the history.

We call this continual interaction between agent and environment a finite Markov Decision Process.

## 2 Transition model

对于这样的一个交互过程，我们可以建立environment的transition概率模型:

$$p(s', r | s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a\}$$

where $s', s \in \mathcal{S}$, $a \in \mathcal{A}$ and $r \in \mathbb{R}$.。

在大部分真实的强化学习应用中，我们是不会have access to这个模型的，即便这个模型是真实存在的。

# 3 Return

Intuitively, this is the "total sum of reward" from now on.

Finite time (the agent is given a finite number of timesteps to interact with the environment):

$$G_t = R_{t+1} + R_{t+2} + \cdots + R_T$$

Infinite time:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k}$$

where we use $\gamma \in [0, 1)$ to keep the sum finite.

# 4 Policies and Value Functions

## 4.1 Policies 策略

Definition: A policy is a mapping from states to distributions over actions. A policy can be good or bad.

Notation: $\pi(a|\, s)$, conditional probability distribution over actions given state

Example: Playing cards. At each timestep, the cards on the table and in your hand is the *state*. Given a specific state, you can of course choose actions uniformly. This counts as a *policy,* albeit a pretty bad one. To be good, you must choose an *action* (among many actions that are good and possible) that's suitable to that state.

Practice: In research, an agent includes a policy along with the algorithm used to learn it.

## 4.2 Value functions 价值函数

**Definition 1.** *The value of state $s$ under policy $\pi$, $v_\pi(s)$ is defined as the expected return of starting in $s$ and follow $\pi$ thereafter. Formally,*

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t|\, S_t = s].$$

*Note that this definition holds only when termination is solely dependent on state.*

**Example 2.** Why is $v_\pi(s)$ independent of $t$? Here's a perhaps crude example.

Suppose there's a race that can last arbitrarily long. A timer starts counting from zero as soon as the race starts. Then, it's straightforward that a person that starts running from $t = 100$ and another person that starts running from $t = 1000$ takes the same time delta to finish the race.

**Definition 3.** *The action-value of state-action pair $(s, a)$ under policy $\pi$, $q_\pi(s, a)$ is defined as the expected return of starting in $s$, take $a$ and follow $\pi$ thereafter. Formally,*

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t|\, S_t = s, A_t = a].$$

TODO: def 3 interpretation leave util next week.

## 4.3 Bellman equation 贝尔曼方程

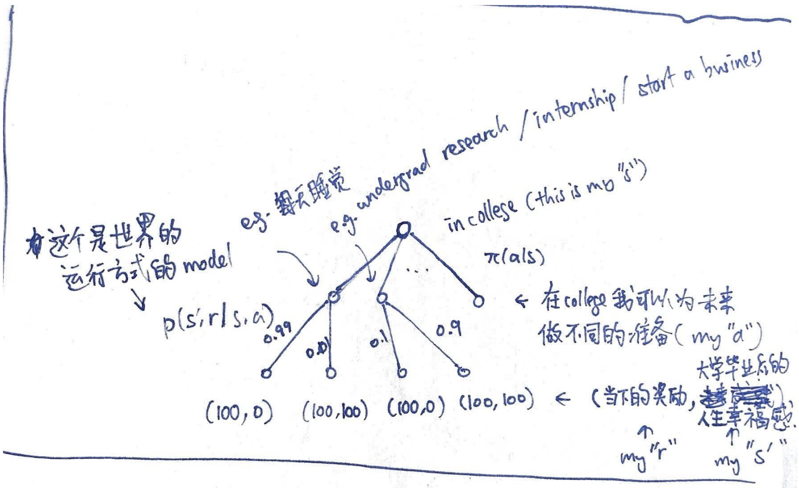
**Theorem 4.** *Bellman equation for $v_\pi$.*

*$v_\pi$ satisfies the following recursive property:*

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|\, s) \sum_{(s', r) \in \mathcal{S} \times \mathbb{R}} p(s', r|\, s, a) \Big[ r + \gamma v_\pi(s') \Big]$$

**Example 5.** Interpretation of the Bellman equation for $v_\pi$.

Suppose we are interested my expected happiness after high school (just starting college).




**Proof.** We proceed by expanding the definition of $v_\pi(s)$:

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi[G_t|\, S_t = s] \\
&= \mathbb{E}_\pi[R_t + \gamma G_{t+1}|\, S_t = s] \\
&= \mathbb{E}_\pi[R_t|\, S_t = s] + \gamma \mathbb{E}_\pi[G_{t+1}|\, S_t = s] \\
&= \mathbb{E}_\pi[R_t|\, S_t = s] + \gamma \mathbb{E}_\pi[\mathbb{E}_\pi[G_{t+1}|\, S_t, S_{t+1}]|\, S_t = s] \quad \text{(Adam's law variant*)} \\
&= \mathbb{E}_\pi[R_t|\, S_t = s] + \gamma \mathbb{E}_\pi[\mathbb{E}_\pi[G_{t+1}|\, S_{t+1}]|\, S_t = s] \quad \text{(Markov property)} \\
&= \mathbb{E}_\pi[R_t|\, S_t = s] + \gamma \mathbb{E}_\pi[v_\pi(S_{t+1})|\, S_t = s] \\
&= \mathbb{E}_\pi[R_t + \gamma v_\pi(S_{t+1})|\, S_t = s] \\
&= \sum_a \pi(a|\, s) \sum_{s', r} p(s', r|\, s, a)[r + \gamma v_\pi(s')]
\end{aligned}$$

(*) Theorem 3.9.8 from here. □

## 4.4 Bellman optimality equation 贝尔曼最优化方程

TODO