

# Reinforcement Learning Formula Sheet

Eddie Guo

## Multi-Armed Bandit Problem

Expected reward of action  $a$ :  $q_*(a) \equiv \mathbb{E}[R_t \mid A_t = a]$

Estimate of  $q_*(a)$  at time  $t$ :  $Q_t(a) \equiv \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}}$

Optimization:  $Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$

$\lim_{t \rightarrow \infty} Q_t(a) = q_*(a)$  by LLN

Greedy action selection:  $A_t = \underset{a}{\operatorname{argmax}} Q_t(a)$

$\epsilon$ -greedy selection: greedy most of time but selects random action w/ small probability  $\epsilon$

Nonstationary problems: constant step-size parameter

$Q_{n+1} \equiv Q_n + \alpha(R_n - Q_n), \quad \alpha \in [0, 1)$

$Q_{n+1} = (1 - \alpha)Q_1 + \sum_{i=1}^n \alpha(1 - \alpha)^{n-i} R_i$

Notice exponentially decaying past rewards.

### 1: A simple bandit algorithm

Initialize, for  $a = 1$  to  $k$ :

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop:

$A \leftarrow \begin{cases} \underset{a}{\operatorname{argmax}} Q(a), & \text{with probability } 1 - \epsilon \\ \text{random action,} & \text{with probability } \epsilon \end{cases}$

$R \leftarrow \text{bandit}(A)$

$N(A) \leftarrow N(A) + 1$

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}[R - Q(A)]$

## Upper Confidence Bound (UCB) Action Selection

“Optimism in the face of uncertainty”

Same as greedy except initialize  $Q_t(a)$  to a high value, select value that optimizes an action  $A_t$ , and updates the upper bound to  $Q_t(a)$ .

$A_t \equiv \underset{a}{\operatorname{argmax}} \left[ Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$

## Finite Markov Decision Processes

State:  $S_t \in \mathcal{S}$ , Action:  $A_t \in \mathcal{A}(s)$ , Reward:  $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

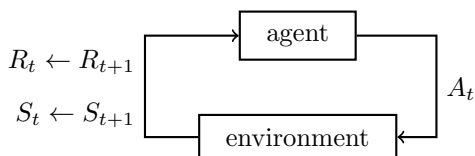
Transition dynamics fn (joint PMF):

Joint prob of next state  $s'$  and reward  $r$  given state  $s$  and action  $a$ .

$p(s', r \mid s, a) \equiv \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$

$p: \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$

$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s)$



## State-Transition Probabilities (Alternative Forms)

$p(s', r \mid s, a) \equiv \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$

$p(s' \mid s, a) = \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r \mid s, a)$

$r(s, a) \equiv \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r \mid s, a)$

$r(s, a, s') \equiv \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in \mathcal{R}} r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)}$

Markov property: future states of Markov process depend only on present state and not on past events.

Agent-envir interactions: episode  $\rightarrow$  terminal state  $\rightarrow$  reset

Goal of agent: maximize expected return,  $G_t$

Episodic tasks:  $G_t \equiv R_{t+1} + R_{t+2} + \dots + R_T$

## Continuing Tasks (no terminal state)

$G_t \equiv R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^{k-1} R_{t+k} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$

$G_t = R_{t+1} + \gamma G_{t+1}, \quad \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1 - \gamma}, \quad \gamma \in [0, 1) \text{ is discount rate}$

$G_t \equiv \sum_{k=t+1}^T \gamma^{k-t-1} R_k \quad T = \infty \text{ or } \gamma = 1 \text{ (but not both)}$

Notice that future rewards are discounted more.

$\gamma = 0$ : agent only cares about immediate reward (greedy).

$\gamma \rightarrow 1$ : future rewards contribute more.

## Policies

Law of total expectation:  $\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]]$

Partition formula:  $\mathbb{E}[X] = \sum_i \mathbb{E}[X \mid A_i] P(A_i)$

Policy: mapping from states to probs of selecting each possible action.

$\pi(a \mid s) = p(a \mid s) = \Pr\{A_t = a \mid S_t = s\}$

Expectation of  $R_{t+1}$  in terms of  $\pi$  and  $p$ :

$\mathbb{E}[R_{t+1} \mid S_t = s] = \sum_a \pi(a \mid S_t) \sum_{s', r} p(s', r \mid s, a) r$

## Value Functions

Value fns give expected return  $G_t$  when starting in state  $s$  and following policy  $\pi$  thereafter.

State-value fn:  $v_\pi(s) \equiv \mathbb{E}_\pi[G_t \mid S_t = s] \quad G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$

Value of terminal state is always 0.

Action-value fn:  $q_\pi(s, a) \equiv \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$

$v_\pi$  in terms of  $q_\pi$  and  $\pi$ :  $v_\pi(s) = \sum_a \pi(a \mid S_t) q_\pi(s, a)$

$q_\pi$  in terms of  $v_\pi$  and  $p$ :  $q_\pi(s, a) = \sum_{r, s'} p(s', r \mid s, a) [r + \gamma v_\pi(s')]$

## Bellman Equations

$$v_{\pi}(s) = \sum_a \pi(a | s) \sum_{s', r} p(s', r | s, a) [r + \gamma v_{\pi}(s')]$$

$$q_{\pi}(s, a) = \sum_{s', r} p(s', r | s, a) \left[ r + \gamma \sum_{a'} \pi(a' | s') q(s', a') \right]$$

*Optimal value fns:*  $\pi_1 \geq \pi_2 \iff v_{\pi_1}(s) \geq v_{\pi_2}(s), \quad \forall s \in \mathcal{S}$

$$v_*(s) = \max_{\pi} v_{\pi}(s) = \max_a \sum_{s', r} p(s', r | s, a) [r + \gamma v_*(s')], \quad \forall s \in \mathcal{S}$$

$$q_*(s, a) = \max_{\pi} q_{\pi}(s, a) = \sum_{s', r} p(s', r | s, a) [r + \gamma \max_{a'} q_*(s', a')]$$

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$$

.....

## Policy Evaluation

$$\pi_* = \operatorname{argmax}_a \sum_{s', r} p(s', r | s, a) [r + \gamma v_*(s')]$$

### 2: Iterative Policy Evaluation

Input  $\pi$ , the policy to be evaluated

$$\vec{V} \leftarrow \vec{0}, \vec{V}' \leftarrow \vec{0}$$

loop:

$$\Delta \leftarrow 0$$

loop for each  $s \in \mathcal{S}$ :

$$V'(s) \leftarrow \sum_a \pi(a | s) \sum_{s', r} p(s', r | s, a) [r + \gamma V(s')]$$

$$\Delta \leftarrow \max(\Delta, |V'(s) - V(s)|)$$

$$V \leftarrow V'$$

until  $\Delta < \theta$  (small positive number)

return  $V \approx v_{\pi}$

*Policy improvement thm:*  $q_{\pi}(s, \pi'(s)) \geq v_{\pi}(s), \quad \forall s \in \mathcal{S}$

$$\pi'(s) \equiv \operatorname{argmax}_a q_{\pi}(s, a) = \operatorname{argmax}_a \sum_{s', r} p(s', r | s, a) [r + \gamma v_{\pi}(s')]$$

### 3: Policy Iteration

1. Initialization

$V(s) \in \mathbb{R}$  and  $\pi(s) \in \mathcal{A}(s)$  arbitrarily  $\forall s \in \mathcal{S}$

2. Policy Evaluation

Loop:

$$\Delta \leftarrow 0$$

Loop for each  $s \in \mathcal{S}$ :

$$v \leftarrow V(s)$$

$$V(s) \leftarrow \sum_{s', r} p(s', r | s, \pi(s)) [r + \gamma V(s')]$$

$$\Delta \leftarrow \max(\Delta, |v - V(s)|)$$

3. Policy Improvement

*policy-stable*  $\leftarrow$  true

For each  $s \in \mathcal{S}$ :

$$\text{old-action} \leftarrow \pi(s)$$

$$\pi(s) \leftarrow \operatorname{argmax}_a \sum_{s', r} p(s', r | s, a) [r + \gamma V(s')]$$

If *old-action*  $\neq \pi(s)$ , then *policy-stable*  $\leftarrow$  false

If *policy-stable*, then stop and return  $V \approx v_*$  and  $\pi \approx \pi_*$ ; else go to 2.