Multi-armed Bandits

RLI Study

Dongmin Lee

Reference

- Reinforcement Learning : An Introduction,
  Richard S. Sutton & Andrew G. Barto
  (Link)

- 멀티 암드 밴딧(Multi-Armed Bandits), 송호연
  (https://brunch.co.kr/@chris-song/62)

Index

1. Why do you need to know Multi-armed Bandits(MAB)?

1. Why do you need to know MAB?

- Reinforcement Learning(RL) uses training information that 'Evaluate'('Instruct' X) the actions

- Evaluative feedback indicates 'How good the action taken was'

- Because of simplicity, 'Nonassociative' one situation

- Most prior work involving evaluative feedback

- 'Nonassociative', 'Evaluative feedback' -> MAB

- In order to Introduce basic learning methods in later chapters

1. Why do you need to know MAB?

In my opinion,

- I think we can't seem to know RL without knowing MAB.

- MAB deal with 'Exploitation & Exploration' of the core ideas in RL.

- In the full reinforcement learning problem, MAB is always used.

- In every profession, MAB is very useful.

2. A k-armed Bandit Problem

2. A k-armed Bandit Problem

Do you know what MAB is?

Do you know what MAB is?

- Slot Machine -> Bandit
- Slot Machine's lever -> Armed
- N slot Machine

→ Multi-armed Bandits

Source : Multi-Armed Bandit
Image source : https://brunch.co.kr/@chris-song/62

Do you know what MAB is?

Among the various slot machines,
which slot machine
should I put my money on
and lower the lever?

Source : Multi-Armed Bandit
Image source : https://brunch.co.kr/@chris-song/62

Do you know what MAB is?

How can you make the best return on your investment?

Source : Multi-Armed Bandit
Image source : https://brunch.co.kr/@chris-song/62

Do you know what MAB is?

MAB is a algorithm created to optimize investment in slot machines

Source : Multi-Armed Bandit
Image source : https://brunch.co.kr/@chris-song/62

2. A k-armed Bandit Problem

A K-armed Bandit Problem

A K-armed Bandit Problem

$$q_*(a) \doteq \mathbb{E}[R_t \mid A_t = a] .$$

t – Discrete time step or play number
A_t - Action at time t
R_t - Reward at time t
$q_*(a)$ – True value (expected reward) of action a

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

A K-armed Bandit Problem

$$q_*(a) \doteq \mathbb{E}[R_t \mid A_t = a] .$$

In our k-armed bandit problem, each of the k actions has an expected or mean reward given that that action is selected; let us call this the value of that action.

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

3. Simple-average Action-value Methods

3. Simple-average Action-value Methods

Simple-average Method

Simple-average Method

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}},$$

$Q_t(a)$ converges to $q_*(a)$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

3. Simple-average Action-value Methods

Action-value Methods

Action-value Methods

Action-value Methods

- Greedy Action Selection Method

- ε-greedy Action Selection Method

- Upper-Confidence-Bound(UCB) Action Selection Method

Action-value Methods

Action-value Methods

- Greedy Action Selection Method

- ε-greedy Action Selection Method

- Upper-Confidence-Bound(UCB) Action Selection Method

Greedy Action Selection Method

$$A_t \doteq \underset{a}{\arg\max}\, Q_t(a),$$

$argmax_a f(a)$ - a value of a at which $f(a)$ takes its maximal value

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Greedy Action Selection Method

$$A_t \doteq \underset{a}{\arg\max}\, Q_t(a),$$

Greedy action selection always exploits
current knowledge to maximize immediate reward

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Greedy Action Selection Method

Greedy Action Selection Method’s disadvantage

Is it a good idea to select greedy action,
exploit that action selection
and maximize the current immediate reward?

Action-value Methods

Action-value Methods

- Greedy Action Selection Method
- ε-greedy Action Selection Method
- Upper-Confidence-Bound(UCB) Action Selection Method

ε-greedy Action Selection Method

Exploitation is the right thing to do to maximize
the expected reward on the one step,
but Exploration may produce the greater total reward in the long run.

ε-greedy Action Selection Method

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1 - \varepsilon \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

ε – probability of taking a random action in an ε-greedy policy

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

ε-greedy Action Selection Method

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

→ Exploitation

→ Exploration

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Action-value Methods

Action-value Methods

- Greedy Action Selection Method

- ε-greedy Action Selection Method

- Upper-Confidence-Bound(UCB) Action Selection Method

Upper-Confidence-Bound(UCB) Action Selection Method

$$A_t \doteq \underset{a}{\arg\max}\left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}\right]$$

$\ln t$ - natural logarithm of t
$N_t(a)$ – the number of times that action a has been selected prior to time t
c – the number $c > 0$ controls the degree of exploration

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Upper-Confidence-Bound(UCB) Action Selection Method

The probability that the slot machine
may be the optimal slot machine

$$A_t \doteq \underset{a}{\arg\max}\left[Q_t(a) + \boxed{c\sqrt{\frac{\ln t}{N_t(a)}}}\right]$$

The idea of this UCB action selection is that
The square-root term is a measure of
the uncertainty(or potential) in the estimate of a's value

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Upper-Confidence-Bound(UCB) Action Selection Method

UCB Action Selection Method's disadvantage

UCB is more difficult than ε-greedy to extend beyond bandits
to the more general reinforcement learning settings

One difficulty is in dealing with nonstationary problems
Another difficulty is dealing with large state spaces

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

4. A simple Bandit Algorithm

4. A simple Bandit Algorithm

- Incremental Implementation

- Tracking a Nonstationary Problem

4. A simple Bandit Algorithm

- Incremental Implementation
- Tracking a Nonstationary Problem

Incremental Implementation

$$Q_n \doteq \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}.$$

Q_n denote the estimate of R_{n-1}'s action value
after R_{n-1} has been selected $\mathrm{n}-1$ times

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Incremental Implementation

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\Big(R_n + (n-1)Q_n\Big) \\
&= \frac{1}{n}\Big(R_n + nQ_n - Q_n\Big) \\
&= Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],
\end{aligned}
$$

$$
Q_n \doteq \frac{R_1 + R_2 + \cdots + R_{n-1}}{n-1}.
$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Incremental Implementation

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\Big(R_n + (n-1)Q_n\Big) \\
&= \frac{1}{n}\Big(R_n + nQ_n - Q_n\Big) \\
&= Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],
\end{aligned}
$$

$Q_{n+1} = Q_n + \frac{1}{n}[R_n - Q_n]$
holds even for $n = 1$,
obtaining $Q_2 = R_1$ for arbitrary Q_1

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Incremental Implementation

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

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

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

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Incremental Implementation

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

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

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

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Incremental Implementation

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

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

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

Available on stationary problem

Unstable(↔Constant)

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Incremental Implementation

A simple bandit algorithm

Initialize, for $a = 1$ to k:
$Q(a) \leftarrow 0$
$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

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

$$NewEstimate \leftarrow OldEstimate + StepSize\left[Target - OldEstimate\right]$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Incremental Implementation

$$NewEstimate \leftarrow OldEstimate \; + \; StepSize \boxed{\left[Target - OldEstimate \right]}$$

The expression $[Target - OldEstimate]$ is an error in the estimate.
The target is presumed to indicate a desirable direction
in which to move, though it may be noisy.

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

4. A simple Bandit Algorithm

- Incremental Implementation
- Tracking a Nonstationary Problem

Tracking a Nonstationary Problem

$$Q_{n+1} \quad = \quad Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],$$

$$Q_{n+1} \doteq Q_n + \alpha\Big[R_n - Q_n\Big],$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Tracking a Nonstationary Problem

$$Q_{n+1} \quad = \quad Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],$$

$$Q_{n+1} \doteq Q_n + \alpha\Big[R_n - Q_n\Big],$$

Why do you think it should be changed from $\frac{1}{n}$ to α?

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Tracking a Nonstationary Problem

Why do you think it should be changed from $\frac{1}{n}$ to α?

We often encounter RL problems that are effectively nonstationary.

In such cases it makes sense to give more weight to recent rewards than to long-past rewards.

One of the most popular ways of doing this is to use a constant step-size parameter.

The step-size parameter $\alpha \in (0,1]$ is constant.

Tracking a Nonstationary Problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha R_{n-2} + \\
&\qquad\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i.
\end{aligned}
$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Tracking a Nonstationary Problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)\underline{Q_n} \quad ? \\
&= \alpha R_n + (1-\alpha)\,\underline{[\alpha R_{n-1} + (1-\alpha)Q_{n-1}]} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2\alpha R_{n-2} + \\
&\qquad\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i.
\end{aligned}
$$

$$Q_n = \alpha R_{n-1} + (1-\alpha)Q_{n-1} \ ?$$

$$
\begin{aligned}
\boxed{Q_{n+1} \doteq Q_n + \alpha\Big[R_n - Q_n\Big],} \\
&= Q_n + \alpha R_n - \alpha Q_n \\
&= \alpha R_n + (1-\alpha)Q_n
\end{aligned}
$$

$$\therefore Q_n = \alpha R_{n-1} + (1-\alpha)Q_{n-1}$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Tracking a Nonstationary Problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + \underline{(1-\alpha)^2 Q_{n-1}} \quad ? \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + \underline{(1-\alpha)^2 \alpha R_{n-2}} + \\
&\qquad\qquad \underline{\cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1} \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha (1-\alpha)^{n-i} R_i .
\end{aligned}
$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Tracking a Nonstationary Problem

$$(1-\alpha)^2 Q_{n-1} = (1-\alpha)^2 \alpha R_{n-2} + (1-\alpha)^3 \alpha R_{n-3} + \cdots$$
$$+ (1-\alpha)^{n-1} \alpha R_1 + (1-\alpha)^n Q_1 \ ?$$

$$= (1-\alpha)^2 \{\alpha R_{n-2} + (1-\alpha)\alpha R_{n-3} + \cdots$$
$$+ (1-\alpha)^{n-3} \alpha R_1 + (1-\alpha)^{n-2} Q_1\}$$

$$\therefore Q_{n-1} = \alpha\{R_{n-2} + (1-\alpha)R_{n-3} + \cdots + (1-\alpha)^{n-3} R_1\} + (1-\alpha)^{n-2} Q_1$$

Tracking a Nonstationary Problem

$$Q_{n+1} \doteq Q_n + \alpha \Big[R_n - Q_n \Big],$$

Sequences of step-size parameters often converge very slowly or need considerable tuning in order to obtain a satisfactory convergence rate.

Thus, step-size parameters should be tuned effectively.

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

5. Gradient Bandit Algorithm

5. Gradient Bandit Algorithm

In addition to a simple bandit algorithm,
there is another way to use the gradient method as a bandit algorithm

5. Gradient Bandit Algorithm

We consider learning a numerical $preference$
for each action a, which we denote $H_t(a)$.

The larger the preference, the more often that action is taken,
but the preference has no interpretation in terms of reward.

In other wards, just because the preference($H_t(a)$) is large,
the reward is not unconditionally large.
However, if the reward is large, It can affect the preference($H_t(a)$)

5. Gradient Bandit Algorithm

The action probabilities are determined according to
a $soft - max\ distribution$ (i.e., Gibbs or Boltzmann distribution)

$$\Pr\{A_t = a\} \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} \doteq \pi_t(a),$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

5. Gradient Bandit Algorithm

$$\Pr\{A_t = a\} \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} \doteq \pi_t(a),$$

$\pi_t(a)$ – Probability of selecting action a at time t

Initially all action preferences are the same
so that all actions have an equal probability of being selected.

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

5. Gradient Bandit Algorithm

There is a natural learning algorithm for this setting based on the idea of stochastic gradient ascent.

On each step, after selecting action A_t and receiving the reward R_t, the action preferences are updated.

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), \qquad \text{and}$$ → Selected action A_t

$$H_{t+1}(a) \doteq H_t(a) - \alpha\big(R_t - \bar{R}_t\big)\pi_t(a), \qquad \text{for all } a \neq A_t,$$ → Non-selected actions

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

5. Gradient Bandit Algorithm

1) What does $\overline{R_t}$ mean?

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), \qquad \text{and}$$

$$H_{t+1}(a) \doteq H_t(a) - \alpha\big(R_t - \bar{R}_t\big)\pi_t(a), \qquad \text{for all } a \neq A_t,$$

2) What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\boxed{\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big)},$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

5. Gradient Bandit Algorithm

1) What does $\overline{R_t}$ mean?

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), \qquad \text{and}$$
$$H_{t+1}(a) \doteq H_t(a) - \alpha\big(R_t - \bar{R}_t\big)\pi_t(a), \qquad \text{for all } a \neq A_t,$$

?

2) What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\boxed{\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big)},$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $\overline{R_t}$ mean?

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), \qquad \text{and}$$

$$H_{t+1}(a) \doteq H_t(a) - \alpha\big(R_t - \bar{R}_t\big)\pi_t(a), \qquad \text{for all } a \neq A_t,$$

$\overline{R_t} \in \mathbb{R}$ is the average of all the rewards.

The $\overline{R_t}$ term serves as a baseline.

If the reward is higher than the baseline,
then the probability of taking A_t in the future is increased,
and if the reward is below baseline, then probability is decreased.
The non-selected actions move in the opposite direction.

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

5. Gradient Bandit Algorithm

1) What does $\overline{R_t}$ mean?

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), \qquad \text{and}$$
$$H_{t+1}(a) \doteq H_t(a) - \alpha\big(R_t - \bar{R}_t\big)\pi_t(a), \qquad \text{for all } a \neq A_t,$$

2) What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha\boxed{\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big)},\ ?$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$H_{t+1}(a) \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)},$$

- Stochastic approximation to gradient ascent in Bandit Gradient Algorithm

$$\mathbb{E}[R_t] = \sum_x \pi_t(x) q_*(x),$$

- Expected reward

$$\mathbb{E}[R_t] = \mathbb{E}[\ \mathbb{E}[R_t|A_t]\ ]$$

- Expected reward by Low of total expectation

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$H_{t+1}(a) \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}$$

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \left[\sum_x \pi_t(x) q_*(x) \right]$$

$$= \sum_x q_*(x) \frac{\partial \pi_t(x)}{\partial H_t(a)}$$

$$= \sum_x (q_*(x) - B_t) \frac{\partial \pi_t(x)}{\partial H_t(a)},$$

$$\mathbb{E}[R_t] = \sum_x \pi_t(x) q_*(x),$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$H_{t+1}(a) \doteq H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}$$

$$\begin{aligned}
\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \left[\sum_x \pi_t(x) q_*(x) \right] \\
&= \sum_x \underline{q_*(x)} \frac{\partial \pi_t(x)}{\partial H_t(a)} \\
&\quad ? \\
&= \sum_x \underline{(q_*(x) - B_t)} \frac{\partial \pi_t(x)}{\partial H_t(a)},
\end{aligned}$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$
\begin{aligned}
\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)}\left[\sum_x \pi_t(x) q_*(x)\right] \\
&= \sum_x \underline{q_*(x)} \frac{\partial \pi_t(x)}{\partial H_t(a)} \\
&\qquad ? \\
&= \sum_x \underline{\big(q_*(x) - B_t\big)} \frac{\partial \pi_t(x)}{\partial H_t(a)},
\end{aligned}
$$

The gradient sums to zero over all the actions, $\sum_x \frac{\partial \pi_t(x)}{\partial H_t(a)} = 0$

– as $H_t(a)$ is changed, some actions' probabilities go up and some go down, but the sum of the changes must be zero because the sum of the probabilities is always one.

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \left(q_*(x) - B_t\right) \frac{\partial \pi_t(x)}{\partial H_t(a)},$$

$$\downarrow$$

$$\begin{aligned}
\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} &= \sum_x \underline{\pi_t(x)}\left(q_*(x) - B_t\right) \frac{\partial \pi_t(x)}{\partial H_t(a)} \underline{/\pi_t(x)}. \\
&= \mathbb{E}\left[\left(q_*(A_t) - B_t\right) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right] \\
&= \mathbb{E}\left[\left(R_t - \bar{R}_t\right) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t)\right],
\end{aligned}$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_x \pi_t(x)\big(q_*(x) - B_t\big)\frac{\partial \pi_t(x)}{\partial H_t(a)}/\pi_t(x).$$

$$= \mathbb{E}\left[\big(\underline{q_*(A_t)} - B_t\big)\frac{\partial \pi_t(A_t)}{\partial H_t(a)}/\pi_t(A_t)\right]$$

$$= \mathbb{E}\left[\big(\underline{R_t} \overset{?}{-} \bar{R}_t\big)\frac{\partial \pi_t(A_t)}{\partial H_t(a)}/\pi_t(A_t)\right],$$

$\rightarrow$ $\mathbb{E}[R_t|A_t] = q_*(A_t).$

$\rightarrow$ $\mathbb{E}[R_t] = \mathbb{E}[\,\mathbb{E}[R_t|A_t]\,]$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \mathbb{E}\left[(R_t - \bar{R}_t) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t) \right]$$

$$\frac{\partial \pi_t(x)}{\partial H_t(a)} = \pi_t(x)\left(\mathbb{1}_{a=x} - \pi_t(a)\right)$$

$\mathbb{1}_{a=x}$ is defined to be 1 if $a = x$, else 0.

Please refer page 40 in link of reference slide

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

What does $(R_t - \overline{R_t})(1 - \pi_t(A_t))$ mean?

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \mathbb{E}\left[(R_t - \bar{R}_t) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t) \right]$$

$$= \mathbb{E}\left[(R_t - \bar{R}_t) \pi_t(A_t) \big(\mathbb{1}_{a=A_t} - \pi_t(a) \big) / \pi_t(A_t) \right]$$

$$= \mathbb{E}\left[(R_t - \bar{R}_t) \big(\mathbb{1}_{a=A_t} - \pi_t(a) \big) \right].$$

We can substitute a sample of the expectation above

for the performance gradient in $H_{t+1}(a) \cong H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}$

$$\therefore \; H_{t+1}(a) = H_t(a) + \alpha \big(R_t - \bar{R}_t \big) \big(\mathbb{1}_{a=A_t} - \pi_t(a) \big), \qquad \text{for all } a,$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

6. Summary

6. Summary

In this chapter, ‘Exploitation & Exploration‘ is the core idea.

6. Summary

Action-value Methods

- Greedy Action Selection Method
- ε-greedy Action Selection Method
- Upper-Confidence-Bound(UCB) Action Selection Method

6. Summary

A simple bandit algorithm : Incremental Implementation

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$$Q(a) \leftarrow 0$$
$$N(a) \leftarrow 0$$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1 - \varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$
$$R \leftarrow bandit(A)$$
$$N(A) \leftarrow N(A) + 1$$
$$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}\big[R - Q(A)\big]$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

6. Summary

A simple bandit algorithm : Incremental Implementation

A simple bandit algorithm

Initialize, for $a = 1$ to k:

$Q(a) \leftarrow 0$

$N(a) \leftarrow 0$

Loop forever:

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1-\varepsilon \quad \text{(breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

$R \leftarrow bandit(A)$

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

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

$$NewEstimate \leftarrow OldEstimate + StepSize\left[Target - OldEstimate\right]$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

6. Summary

A simple bandit algorithm : Tracking a Nonstationary Problem

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha\Big[R_n - Q_n\Big] \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)\left[\alpha R_{n-1} + (1-\alpha)Q_{n-1}\right] \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1} \\
&= \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 \alpha R_{n-2} + \\
&\qquad\qquad \cdots + (1-\alpha)^{n-1}\alpha R_1 + (1-\alpha)^n Q_1 \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i.
\end{aligned}
$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

6. Summary

A simple bandit algorithm : Tracking a Nonstationary Problem

$$Q_{n+1} \doteq Q_n + \alpha \Big[R_n - Q_n \Big],$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

6. Summary

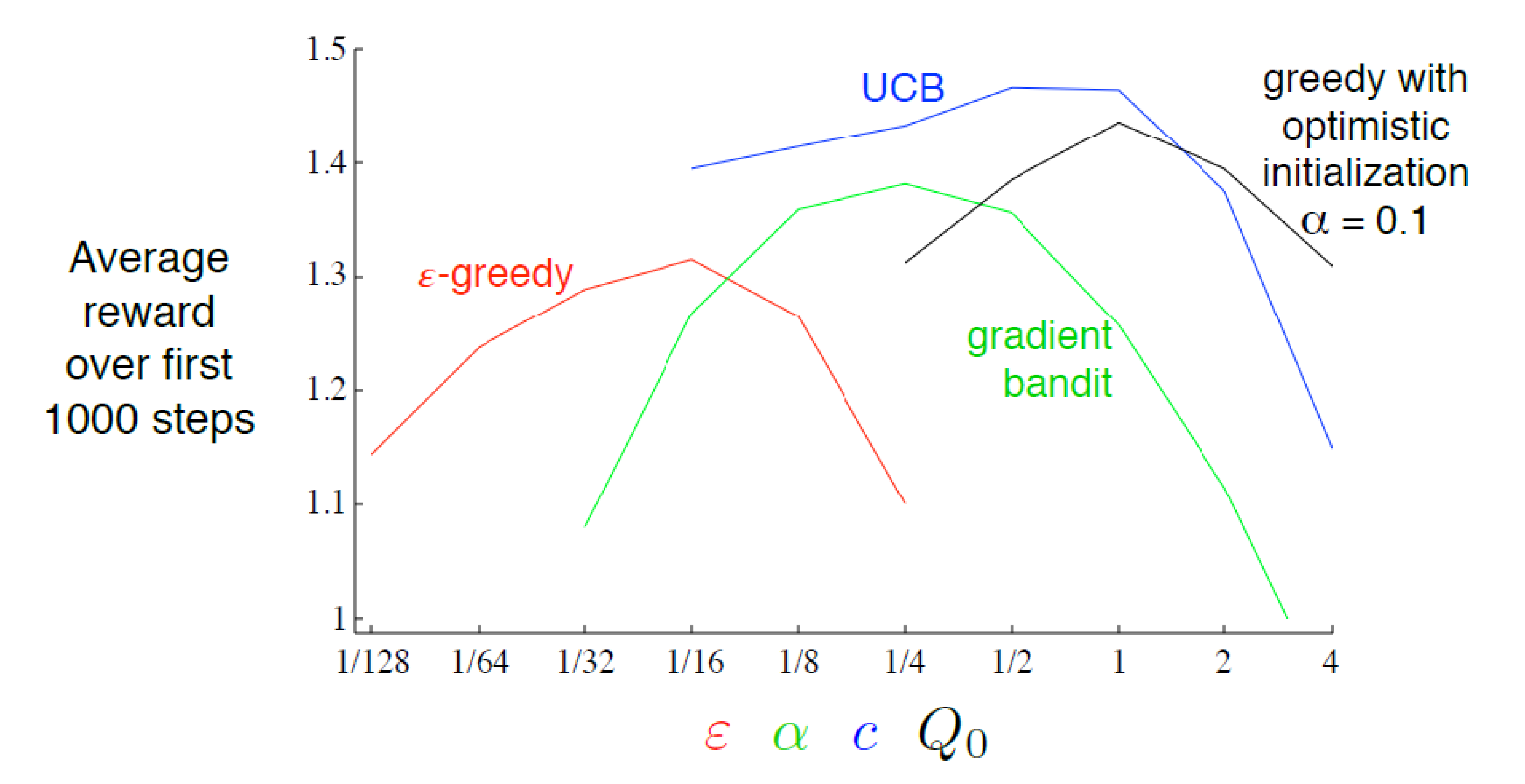
Gradient Bandit Algorithm

$$
\begin{aligned}
H_{t+1}(A_t) &\doteq H_t(A_t) + \alpha\big(R_t - \bar{R}_t\big)\big(1 - \pi_t(A_t)\big), && \text{and} \\
H_{t+1}(a) &\doteq H_t(a) - \alpha\big(R_t - \bar{R}_t\big)\pi_t(a), && \text{for all } a \neq A_t,
\end{aligned}
$$

Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

6. Summary

A parameter study of the various bandit algorithms



Source : Reinforcement Learning : An Introduction, Richard S. Sutton & Andrew G. Barto
Image source : https://drive.google.com/file/d/1xeUDVGWGUUv1-ccUMAZHJLej2C7aAFWY/view

Reinforcement Learning is LOVE♥

Thank you