Chapter 2. Multi-armed Bandits

Reinforcement Learning

발표자 정의정

목차

2.0 Introduction

- The most important feature distinguishing reinforcement learning from other types of learning

uses training information that evaluates the actions taken rather than instructs by giving correct actions

└ action에 evaluate을 주는 형태임

이게 중요

reinforcement

- Evaluative feedback vs instructive feedback

└ 선택한 action에 대한 feedback   └ 선택한 action과 독립적인 feedback

- In this chapter we study the evaluative aspect of reinforcement learning in a simplified setting

K-armed bandit problem

2.1 A k-armed Bandit Problem

- **Problem definition**

  1. a choice among k different options, or actions
  2. After each choice you receive a numerical reward (chosen from stationary probability distribution)

- **Object**

  To maximize the expected total reward over some time period

2.1 A k-armed Bandit Problem

- Each of the k actions has an expected or mean reward given that action is selected

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

이거의 기댓값..

- A_t : the action on time step t
- R_t : the reward on time step t
- $q_*(a)$: an expected or mean reward
- $Q_t(a)$: the estimated value of action a at time step t

closer

2.1 A k-armed Bandit Problem

- **Greedy action vs ε − greedy**

  Greedy action whose estimated value is greatest

  한 step에서 기대하는

  -> Exploitation

  ε − greedy is non greedy action

  -> Exploration

- **Exploitation vs Exploration**

  단기적

  Exploitation : to maximize the expected reward on the one step

  Exploration : the greater total reward in the long run 장기적인 보상

  -> balancing exploitation and exploration

2.2 Action-value Methods

- **True action-value** $q_*(a)$

  the mean reward when the action is selected

  How to estimate? Averaging the rewards "actually received"

- **Sample-average Method** $Q_t(a)$

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

where $\mathbb{1}_{predicate}$ denotes the random variable that is 1 if *predicate* is true and 0 if it is not.

If denominator is

1) $0 \rightarrow$ define $Q_t(a)$ as default value (such as 0)
2) $\infty \rightarrow Q_t(a)$ converges $q_*(a)$

2.2 Action-value Methods

- Greedy action selection

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

always exploits current knowledge to maximize immediate reward

- $\varepsilon-$greedy method

behave every once in a while with small probability ε

2.3 The 10-armed Testbed

- To compare the relative effectiveness of the greedy and $\varepsilon-$greedy
- A set of 2000 randomly generated k-armed bandit problems with k =10 (normal distribution, mean 0, var 1)

2.3 The 10-armed Testbed

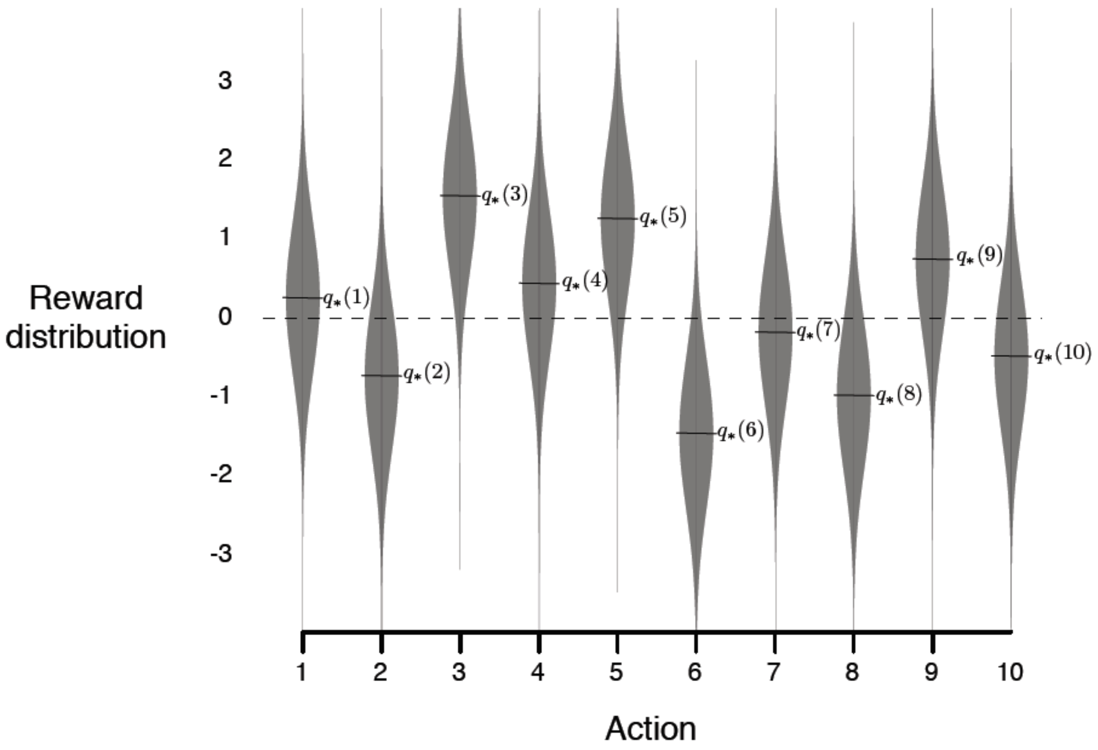


Figure 2.1: An example bandit problem from the 10-armed testbed. The true value $q_*(a)$ of each of th actions was selected according to a normal distribution with mean zero and unit variance, and then the a rewards were selected according to a mean $q_*(a)$ unit variance normal distribution, as suggested by these distributions.

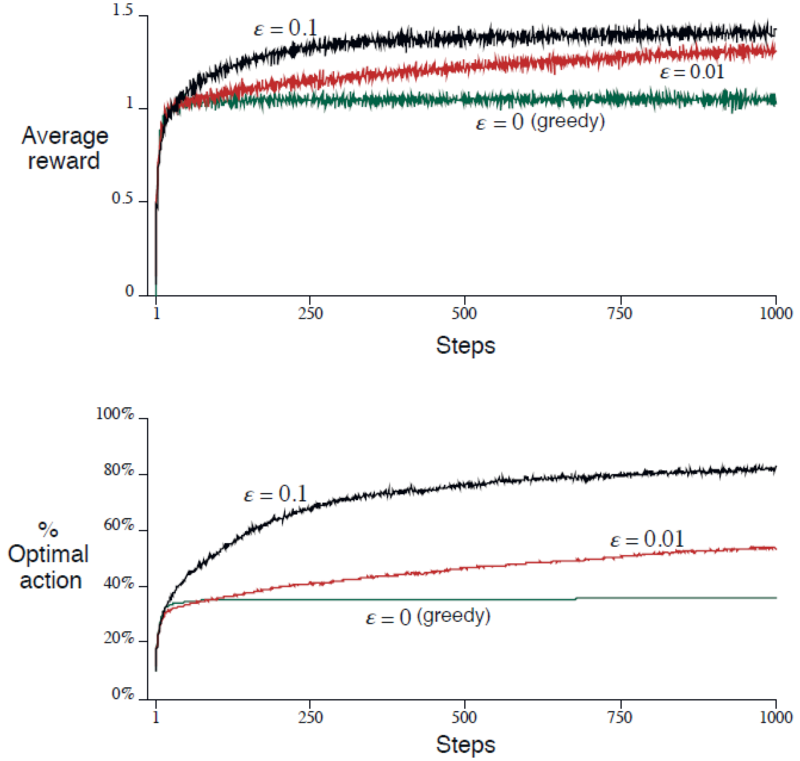


Figure 2.2: Average performance of ε-greedy action-value methods on the 10-armed testbed. These data are averages over 2000 runs with different bandit problems. All methods used sample averages as their action-value estimates.

2.4 Incremental Implementation

- Effective way that estimate action values as sample averages ?
- Q_n : estimate of its action value after it has been selected n-1 times

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

- **Problem:**

  1) memory : record of all the rewards

  2) computation : perform computation whenever the estimated value was needed

-> Incremental Implementation

2.4 Incremental Implementation

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

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

$$NewEstimate \leftarrow OldEstimate + StepSize \Big[Target - OldEstimate\Big].$$

→ generalization

2.5 Tracking a Nonstationary Problem

- Stationary bandit problems -> the reward probabilities do not change over time. //
- Reinforcement learning problems that are effectively nonstationary //

불안정함 :

- **Non stationary Problem**

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

  -> use a constant step-size parameter $\alpha \in (0,1]$

  ex)

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

2.5 Tracking a Nonstationary Problem

- step-size parameter $\alpha \in (0,1]$

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

- Convergence condition

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty \qquad \text{and} \qquad \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty.$$

- Weighted average

$$(1-\alpha)^n + \textstyle\sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} = 1.$$

- Sample-average case

$\alpha_n(a) = \frac{1}{n}$ → both

- Step-size parameter

$\alpha \in (0,1]$ → second condition x

2.6 Optimistic Initial Values

낙관적인

- $Q_1(a)$: initial action-value (initialize to 0)

if do not initialize $Q_1(a)$ to 0 → biased by initial estimate

→used as a simple way to encourage exploration

"Optimistic Initial Values!"

ex) 10-armed bandit problem

Suppose initializing $Q_1(a)$ to +5 about every action a.

1. because of $Q_1(a)$ = 5 about every action a, randomly select one action a
2. Whichever actions are initially selected, the reward is less than the starting estimates.
3. the learner switches to other actions (the selected action become non-greedy action)
4. repeat 1,2,3, about all actions

2.6 Optimistic Initial Values

- Optimistic initial values

  effective on stationary problems, but not on non-stationary problem

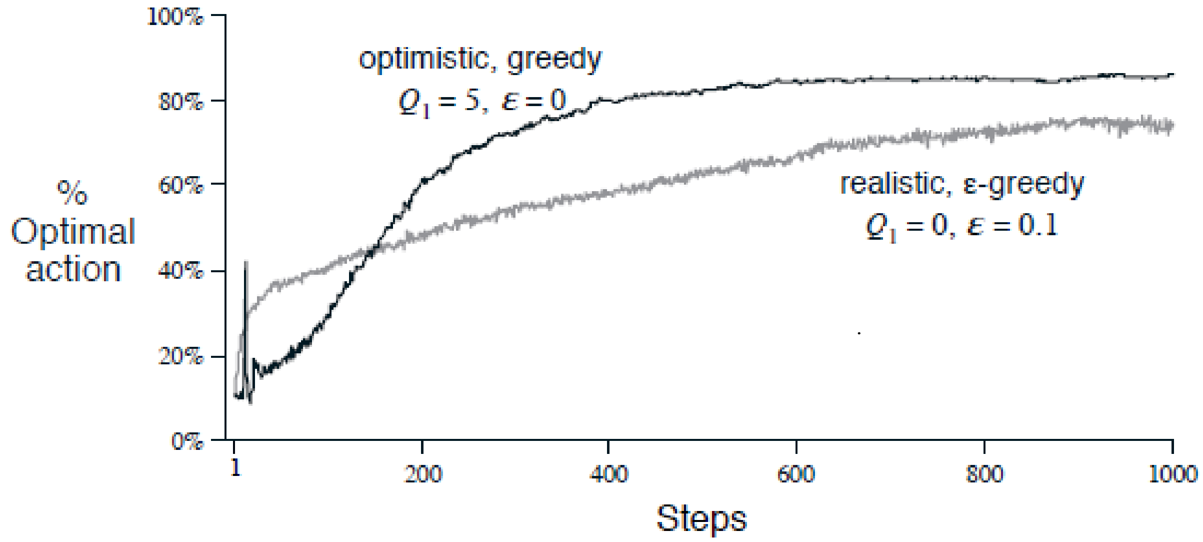


Figure 2.3: The effect of optimistic initial action-value estimates on the 10-armed testbed. Both methods used a constant step-size parameter, $\alpha = 0.1$.

2.7 Upper-Condition-Bound Action Selection

- Exploration is needed because there is always uncertainty about the accuracy of the action-value estimates.

→ lets study exploration methods

- $\varepsilon-$greedy action selection forces the non-greedy actions to be tried, but no preference for those that are nearly greedy or particularly uncertain.

→ Upper-confidence-bound action selection(UCB)

Optimistic initial values

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

The increases get smaller over time, but are unbounded

→ All actions will eventually be selected

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

- UCB often performs well, but is more difficult than $\varepsilon-$greedy

2.7 Upper-Condition-Bound Action Selection

- UCB vs e-greedy

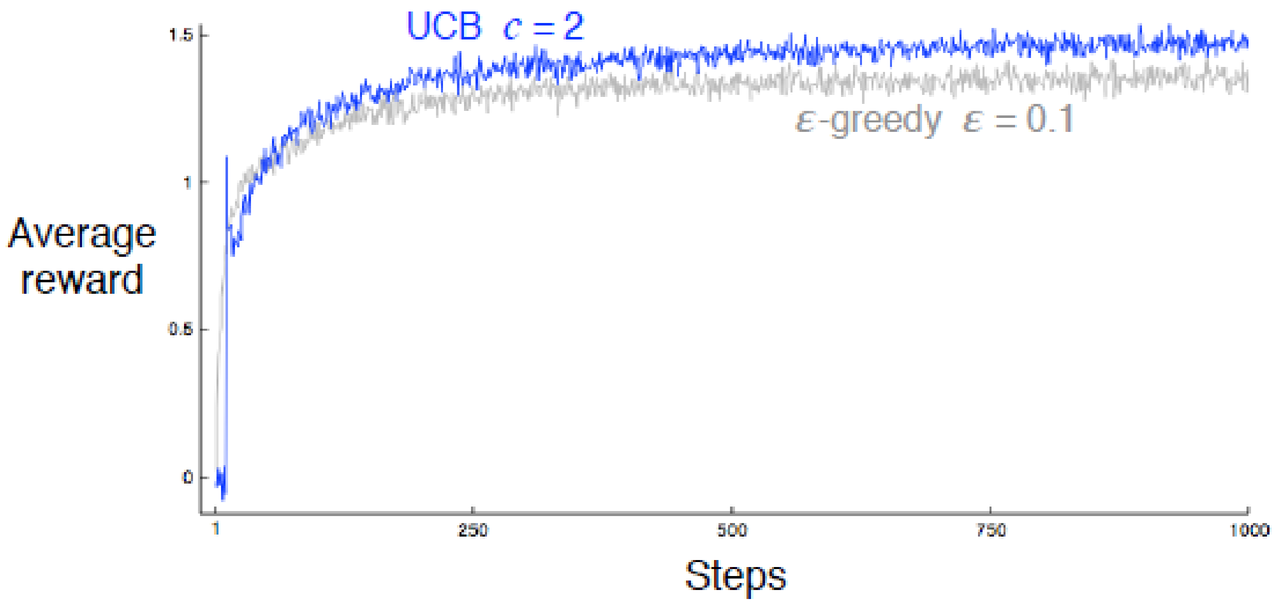


Figure 2.4: Average performance of UCB action selection on the 10-armed testbed. As shown, UCB generally performs better than ε-greedy action selection, except in the first k steps, when it selects randomly among the as-yet-untried actions.

2.8 Gradient Bandit Algorithms

- Method of learning a numerical preference (denote $H_t(a)$)

  Consider the relative preference of one action over another

- Action probabilities, which are determined according to a soft-max distribution

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

- $\pi_t(a)$: the probability of taking action a at time t
- Initially all preferences are the same ($H_1(a) = 0$)

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

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

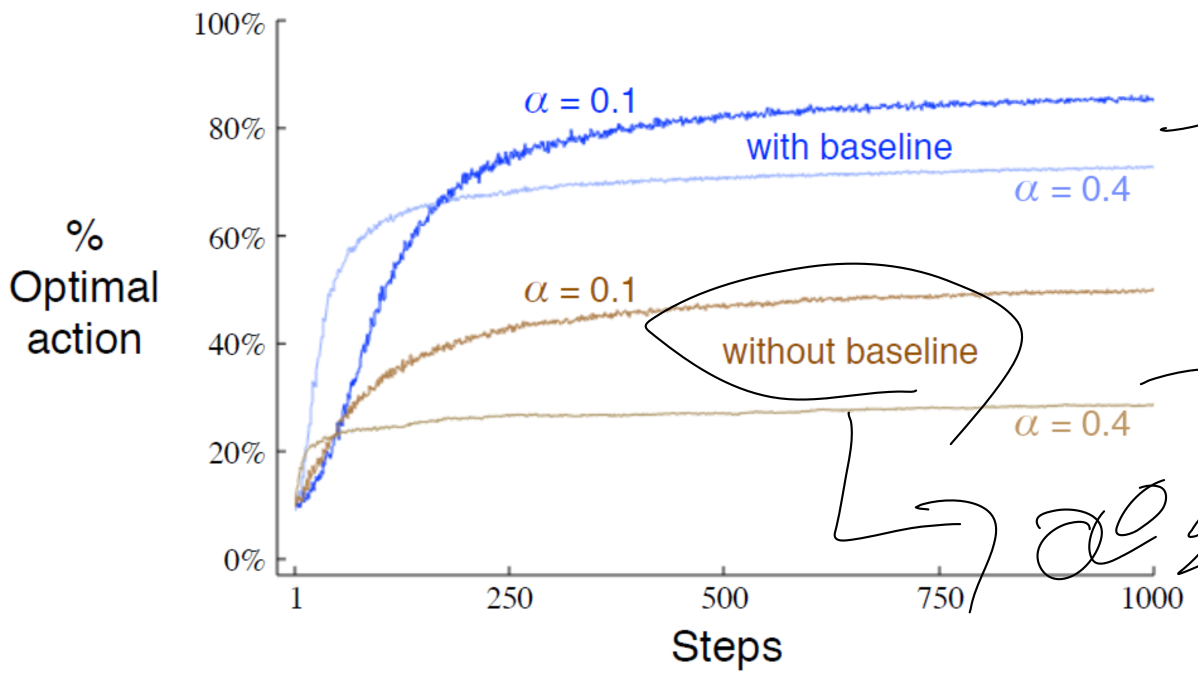
$R_t - \overline{R_t} > 0 \rightarrow \pi_t(a)$ ↑

$R_t - \overline{R_t} < 0 \rightarrow \pi_t(a)$ ↓

- Preferences are updated by the idea of stochastic gradient ascent.
- $\alpha > 0$: a step-size parameter
- $\bar{R}_t$: the average of all the rewards, as a baseline

2.8 Gradient Bandit Algorithms

- Baseline & step-size experiment



baseline에서
빼는 거리 아님
보기좋인건지??

α으로 설정

Figure 2.5: Average performance of the gradient bandit algorithm with and without a reward baseline on the 10-armed testbed when the $q_*(a)$ are chosen to be near +4 rather than near zero.

2.8 Gradient Bandit Algorithms

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

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \left[\sum_b \pi_t(b) q_*(b) \right]$$
$$= \sum_b q_*(b) \frac{\partial \pi_t(b)}{\partial H_t(a)}$$
$$= \sum_b \big(q_*(b) - X_t\big) \frac{\partial \pi_t(b)}{\partial H_t(a)},$$

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \sum_b \pi_t(b) \big(q_*(b) - X_t\big) \frac{\partial \pi_t(b)}{\partial H_t(a)} / \pi_t(b)$$
$$= \mathbb{E}\left[\big(q_*(A_t) - X_t\big) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t) \right]$$
$$= \mathbb{E}\left[\big(R_t - \bar{R}_t\big) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} / \pi_t(A_t) \right],$$
$$= \mathbb{E}\Big[\big(R_t - \bar{R}_t\big) \pi_t(A_t) \big(\mathbb{1}_{a=A_t} - \pi_t(a)\big) / \pi_t(A_t) \Big]$$
$$= \mathbb{E}\Big[\big(R_t - \bar{R}_t\big) \big(\mathbb{1}_{a=A_t} - \pi_t(a)\big) \Big].$$

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

2.8 Gradient Bandit Algorithms

$$
\begin{aligned}
\frac{\partial \pi_t(b)}{\partial H_t(a)} &= \frac{\partial}{\partial H_t(a)} \pi_t(b) \\
&= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(b)}}{\sum_{c=1}^{k} e^{H_t(c)}} \right] \\
&= \frac{\frac{\partial e^{H_t(b)}}{\partial H_t(a)} \sum_{c=1}^{k} e^{H_t(c)} - e^{H_t(b)} \frac{\partial \sum_{c=1}^{k} e^{H_t(c)}}{\partial H_t(a)}}{\left(\sum_{c=1}^{k} e^{H_t(c)} \right)^2} && \text{(by the quotient rule)} \\
&= \frac{\mathbb{1}_{a=b} e^{H_t(b)} \sum_{c=1}^{k} e^{H_t(c)} - e^{H_t(b)} e^{H_t(a)}}{\left(\sum_{c=1}^{k} e^{H_t(c)} \right)^2} && \left(\text{because } \frac{\partial e^x}{\partial x} = e^x\right) \\
&= \frac{\mathbb{1}_{a=b} e^{H_t(b)}}{\sum_{c=1}^{k} e^{H_t(c)}} - \frac{e^{H_t(b)} e^{H_t(a)}}{\left(\sum_{c=1}^{k} e^{H_t(c)} \right)^2} \\
&= \mathbb{1}_{a=b} \pi_t(b) - \pi_t(b) \pi_t(a) \\
&= \pi_t(b) \left(\mathbb{1}_{a=b} - \pi_t(a) \right). && \text{Q.E.D.}
\end{aligned}
$$

2.9 Associative Search (Contextual Bandits)

연관성 고려X

- K-armed bandit problem → nonassociative task

(개별 작업 상황에서 연관성이 없음)

- However, in a general reinforcement learning task there is more than one situation, and the goal is to learn a policy

- K-armed bandit problem → associative search task(contextual bandits) → full reinforcement learning.

THANK YOU