

Chapter 2. Multi-armed Bandits

강준하

Part I : Tabular Solution Methods

- state & action space가 table로 표현할 수 있을 만큼 충분히 작은 공간일 때 사용하는 방법
- often find exact solutions (the optimal value function & the optimal policy)
↔ approximated methods : 단지 대략적인 정답만 알 수 있음, 대신 더 넓은 space에서의 문제를 효과적으로 해결할 수 있음

Part I : Tabular Solution Methods

Chap 2. state가 하나뿐인 간단한 문제에 대한 해결방안

Chap 3. 일반적인 problem formulation 소개 : Markov Decision Process(MDP), Bellman Equations, Value Functions

Part I : Tabular Solution Methods

Chap 4~6. Chap 3.에서 제시한 MDP 문제를 푸는 세 가지 방법 소개

- Dynamic Programming : 수학적으로 정확 / 환경에 대한 정확한 정보 필요
- Monte Carlo Methods : 모델이 필요 없음, 간단한 컨셉 / computational cost가 급격히 증가
- Temporal-Difference Learning : 모델 필요 없음, cost 문제 해결 / 분석하기에 복잡함

Chap 7~8. 위의 세 가지 기법을 적절히 조합하여 사용하는 방법

- Chap 7. Monte Carlo + TD via multi-step bootstrapping methods
- Chap 8. TD 통한 model learning & planning methods

Chapter overview

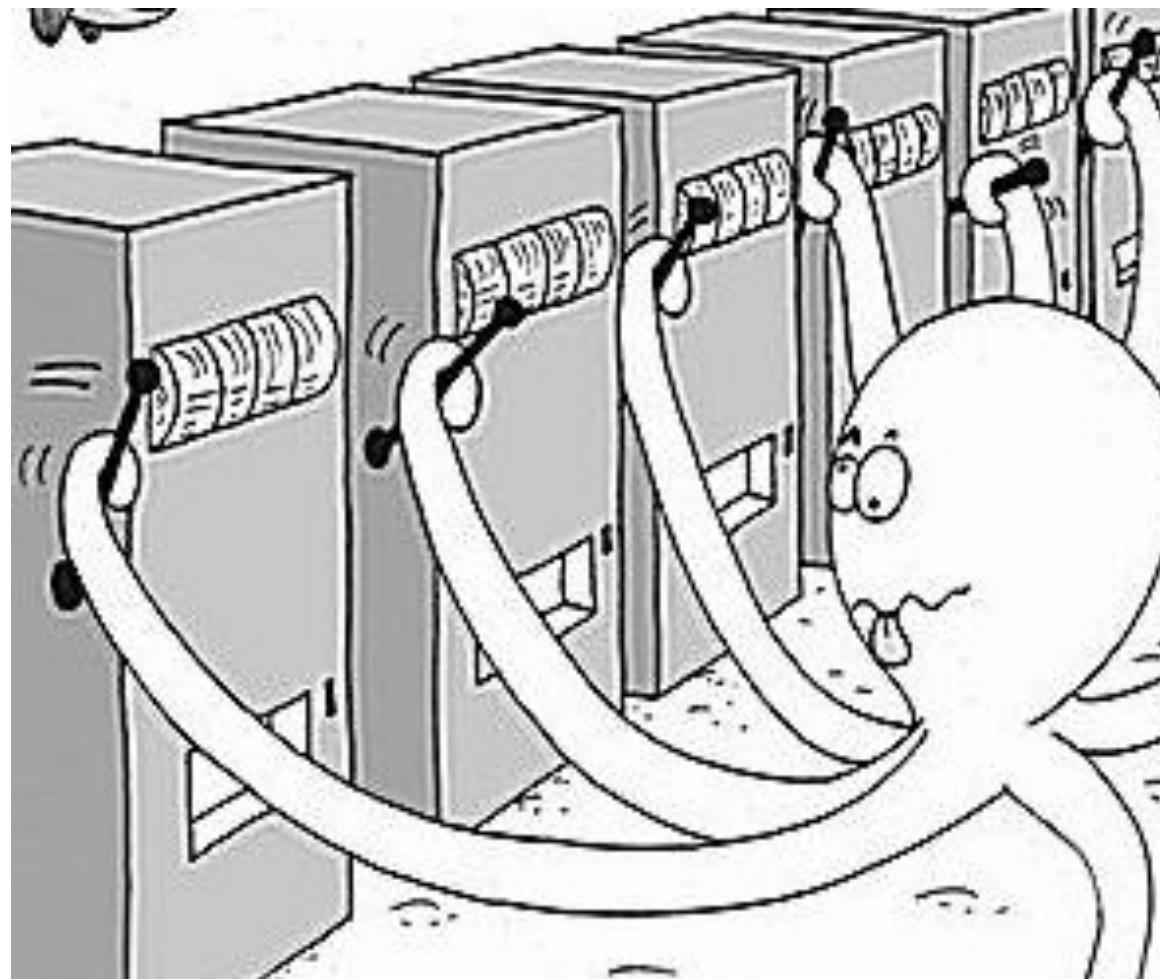
- RL이 다른 learning methods와 구별되는 중요한 특징들
 - training information을 통해 옳은 action을 지시하는 것이 아니라, action을 평가한다는 점이다(exploration을 장려해야 함!).
 - 얼마나 그 action이 좋았는지 evaluation할 뿐, correct/wrong action을 가르지 않음
- 간단한 문제를 RL을 통해 해결해보면서 대충 감을 잡자!

One-armed bandit?



slot machine

A k -armed Bandit Problem?



A k -armed Bandit Problem

each one-armed bandit has stationary probability distribution



mean : μ_1
variance : σ_1^2



mean : μ_2
variance : σ_2^2



mean : μ_3
variance : σ_3^2



mean : μ_4
variance : σ_4^2

...

A k -armed Bandit Problem

each one-armed bandit has stationary probability distribution

a_1



a_2



a_3



a_4



a_5

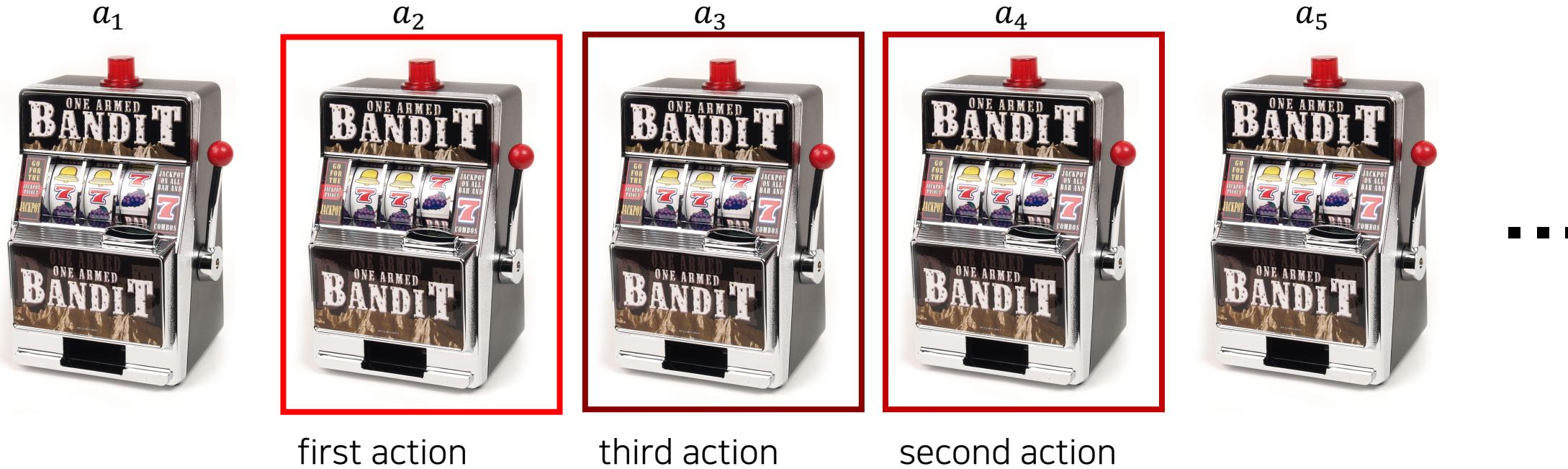


⋮ ⋮ ⋮

get expected(mean) reward
→ value of action a_2

A k -armed Bandit Problem

each one-armed bandit has stationary probability distribution



- t 시간에 선택된 action을 A_t , 따르는 reward를 R_t 라고 하자
- 임의의 action a 의 value를 $q_*(a)$ 라고 표기하면, a 를 선택했을 때의 expected value는 $q_*(a) = E[R_t | A_t = a]$

A k -armed Bandit Problem

- 모든 action에 대한 $q_*(a)$ 를 정확하게 알 수 있다면 trivial한 해를 찾을 수 있으나... 불가능
- estimated value of action a at time t : $Q_t(a)$
- 우리의 목표 : $Q_t(a)$ 를 $q_*(a)$ 에 맞추는 것

A k -armed Bandit Problem

each one-armed bandit has stationary probability distribution

a_1



a_2



a_3



a_4



a_5



⋮ ⋮ ⋮

value : 1

value : 1.5

value : 0.5

mean : μ_1
variance : σ_1^2

mean : μ_2
variance : σ_2^2

mean : μ_3
variance : σ_3^2

mean : μ_4
variance : σ_4^2

mean : μ_5
variance : σ_5^2

A k -armed Bandit Problem

- 해결 방안 1) 가장 value가 높았던 action만 선택 (only exploitation)
→ greedy method
 - 다음 스텝의 value는 최대화 가능
- 해결 방안 2) 적극적인 exploration
 - 장기적으로 봤을 때 더 많은 reward 획득 가능
 - 초반에는 효과가 안좋지만 다양한 action 체험 가능
- "conflict" between exploration & exploitation

Action-Value Methods

1. actions들의 value를 추정하는 방법
2. action 선택 방법을 추정을 이용해 만드는 방법

Action-Value Methods 1

- Recall - action의 true value는 action이 선택되었을 때의 평균 reward
→ 그냥 경험들을 모두 평균내자!

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}}, \quad (2.1)$$

- 분모 $\rightarrow \infty$ 이면 큰 수의 법칙에 의해 $Q_t(a) \rightarrow q_*(a)$
- sample-average method

Action-Value Methods 2

- greedy action : action들 중 value가 가장 높게 예측된 action 선택

$$A_t \doteq \arg \max_a Q_t(a)$$

- ε -greedy method : ε 의 확률로 greedy한 선택이 아니라 모든 action 들 중 랜덤하게 하나의 action을 선택함 ($0 < \varepsilon < 1$)
→ optimal action 뽑을 확률이 $1 - \varepsilon$ 보다 큰 확률로 수렴한다.

Exercise 2.1 In ε -greedy action selection, for the case of two actions and $\varepsilon = 0.5$, what is the probability that the greedy action is selected? □

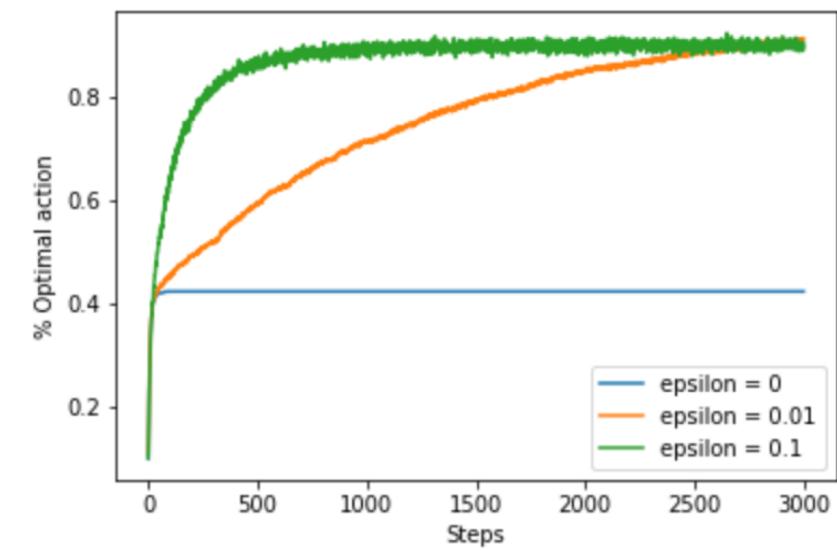
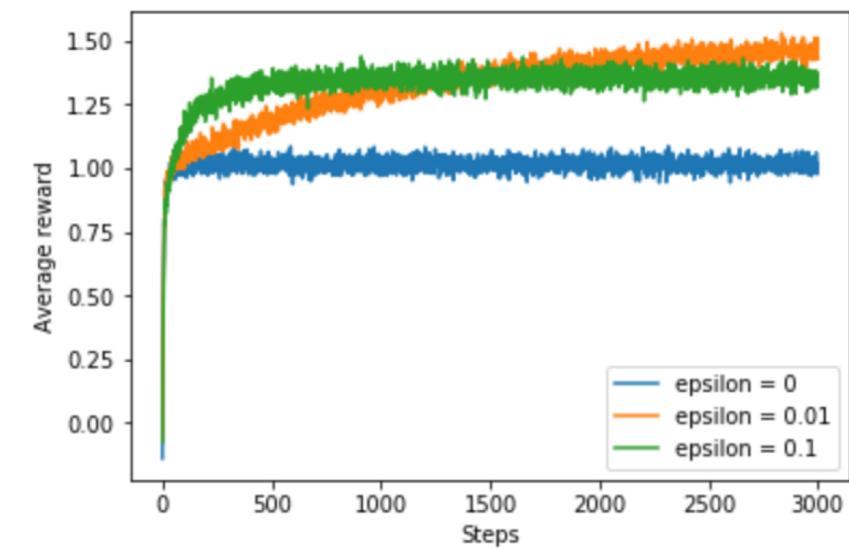
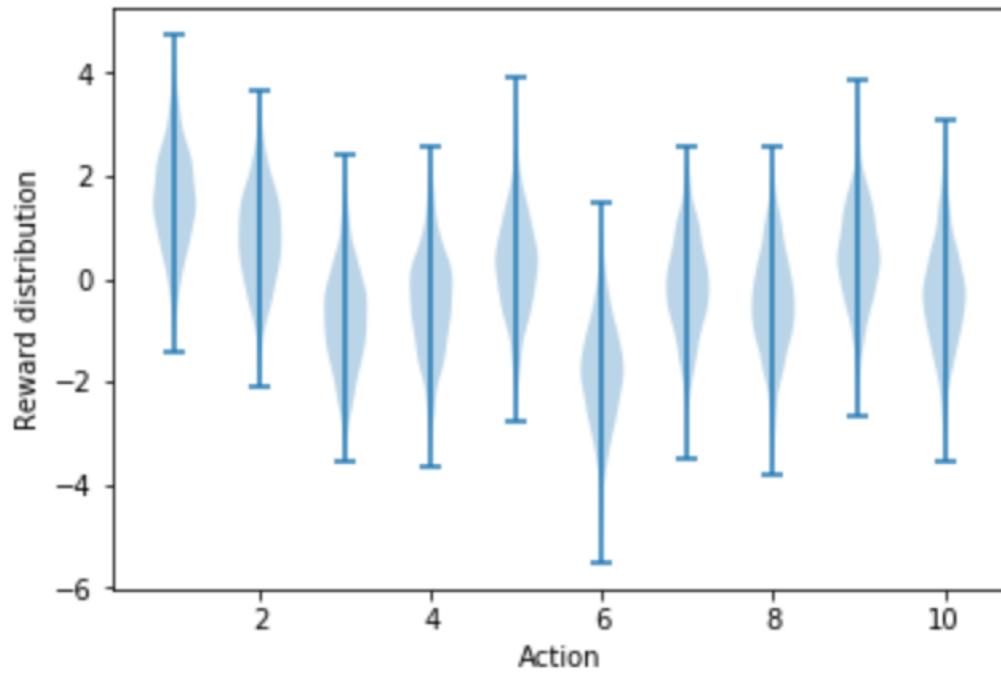
The 10-armed Testbed



The 10-armed Testbed

1. 10개의 action에 대한 optimal action value($q_*(a)$, $a = 1, \dots, 10$)의 값을 mean = 0, variance = 1인 normal distribution을 이용해 설정한다.
2. 각 action에 대한 time step t 에서의 reward R_t 는 mean = $q_*(a)$, variance = 1의 normal distribution을 따른다 하자.
3. 2000번의 실험, 1번의 실험은 3000 time step
4. $\varepsilon = 0.1, 0.01, 0$ 세 번 해보자.

The 10-armed Testbed



Incremental Implementation

- 기존의 $Q_t(a)$ 계산

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

- n 이 증가할수록 연산량이 늘어남
- 연산량을 일정하게 유지하고 싶다!

Incremental Implementation

- 간단한 수식 전개를 통하면...

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

Q_n과 n만으로 R_n 계산 가능

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize} [\text{Target} - \text{OldEstimate}].$$

Incremental Implementation

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \underline{\text{StepSize}} \left[\text{Target} - \text{OldEstimate} \right].$$

- 이제까지의 방법에 의하면 $\text{StepSize} = \frac{1}{n}$
- general 하게 $\text{StepSize} = \alpha_t(a)$ 로 표현함 (앞으로는 이 표기법 이용)

중간 요약

A simple bandit algorithm

Initialize, for $a = 1$ to k :

$$Q(a) \leftarrow 0$$

$$N(a) \leftarrow 0$$

Loop forever:

ε -greedy method

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

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

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

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

Tracking a Nonstationary Problem

- 앞서 살펴본 문제에서는 각 슬롯머신의 확률 분포가 고정 (stationary)
- $StepSize = \frac{1}{n}$: 모든 reward에 똑같은 가중치 적용
- Nonstationary problem에서는 최근의 reward가 더 의미있음
- step size를 constant value로 두자($StepSize = \alpha$)!

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

Tracking a Nonstationary Problem

- Then...

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

시간이 지난 만큼 영향이 적어진다는 의미

Tracking a Nonstationary Problem

- $Q_t(a)$ 가 수렴한다면 큰 수의 법칙에 의해 true action value로 수렴
- 단, stationary problem일 때
- nonstationary problem에서는 $Q_t(a)$ 가 수렴한다면 reward의 확률 분포가 변했을 때 이를 반영 못함
- 수렴 조건 : $\sum_{n=1}^{\infty} \alpha_n(a) = \infty$ and $\sum_{n=1}^{\infty} \alpha_n^2(a) < \infty.$

Optimistic Initial Values

- 이제까지 논의한 모든 방법들 : 모든 $Q_1(a)$ 를 0으로 초기화
- $Q_1(a)$ 의 초기화에 이후 값들이 dependent \rightarrow biased!
- sample-average method에서는 모든 action이 적어도 한 번씩은 선택되어야 bias가 사라짐 ($n \rightarrow \infty$)

Optimistic Initial Values

- 하지만 bias가 $StepSize = \alpha$ 일 때는 영구적으로 남음
- 추정하려는 action value에 대한 사전 지식이 있다면 이런 bias가 도움이 되기도 함
- bias를 exploration을 촉진시키기 위한 목적으로 이용해 보자!

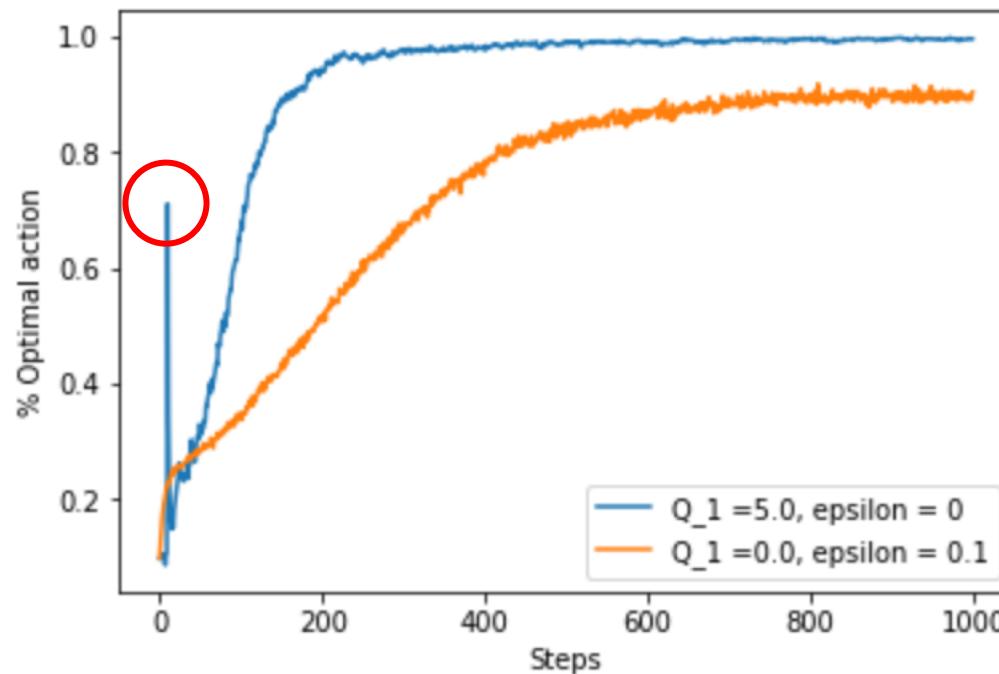
Optimistic Initial Values

- initial value를 0 대신 +5로 모두 설정했다고 생각해 보자
 1. 모든 action에 대해 $Q_1(a) = 5$ 이므로 무작위로 action 선택
 2. 무조건 reward는 5 이하이므로 선택된 action의 value estimation은 작은 쪽으로 업데이트됨
 3. 이 action은 non-greedy action으로 취급되고, 다음에는 나머지 중 하나의 action이 random하게 선택됨
 4. 반복

Optimistic Initial Values

- 학습 초기에 exploration 끝낼 수 있음
- optimistic initial values : 초기 상태에 중점을 둔다는 의미
- stationary problem에서는 잘 작동함
- nonstationary problem에서는 잘 작동하지 않음
 - 초기 상태가 시간이 변하면서 바뀔 수 있으므로

Optimistic Initial Values



Exercise 2.6: Mysterious Spikes The results shown in Figure 2.3 should be quite reliable because they are averages over 2000 individual, randomly chosen 10-armed bandit tasks. Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps? □

Upper-Confidence-Bound Action Selection

- exploration은 필수적임! (action-value 추정의 불확실성으로 인해)
- ϵ -greedy method는 강제로 non-greedy action을 선택하도록 함
- 하지만 어떤 action이 '얼마나 non-greedy한지' 정도 차이를 고려하지 않음
 - 적게 시행된 action부터 시도해봐야 하지 않나...

Upper-Confidence-Bound Action Selection

- UCB는 선택된 횟수가 적으면 탐색이 쉽게 이루어지도록 이를 반영

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

- 얼마나 explore할지 정해주는 hyperparameter

- 양으로 발산하는 함수이므로 결국 모든 action 선택됨
- 증가폭이 시간이 갈수록 감소하므로 시간이 오래 지나서도 추정치가 낮은 action들이 적게 선택되었다는 이유만으로 선택되지 않도록 함

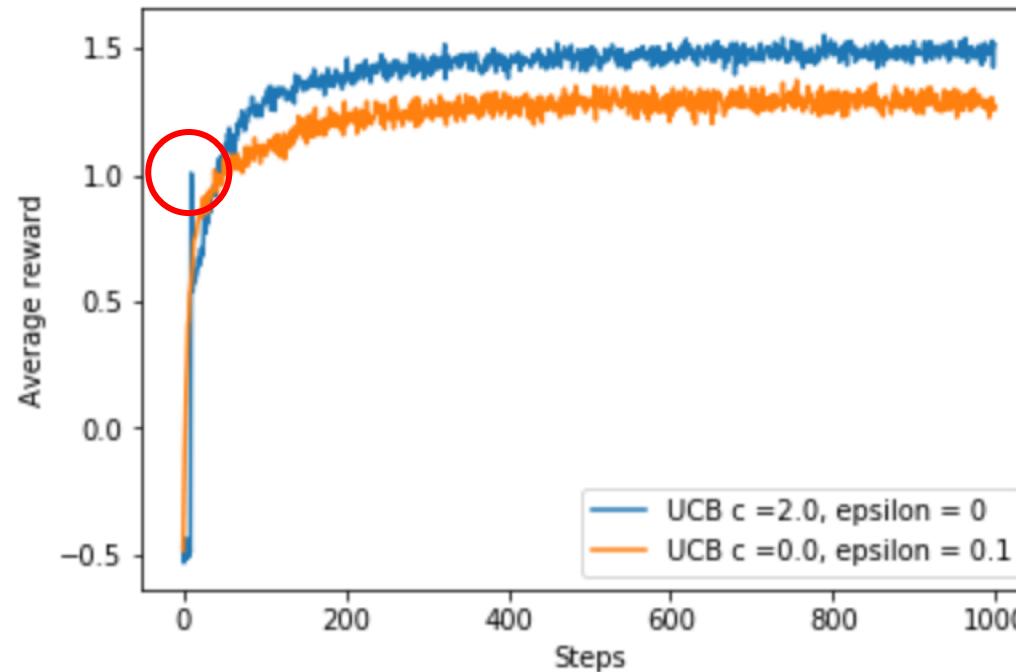
- t 이전에 a 가 선택된 횟수
- $N_t(a) = 0$ 이면 무조건 이러한 action 먼저 탐색

$c\sqrt{\frac{\ln t}{N_t(a)}}$: action a 의 추정치의 불확실성/variance를 나타냄

Upper-Confidence-Bound Action Selection

- nonstationary 문제 / large space 문제에서 적용 힘들
 - action의 의미가 바뀌니깐
 - 전체 space를 explore만 할 수는 없으니깐

Upper-Confidence-Bound Action Selection



Exercise 2.8: UCB Spikes In Figure 2.4 the UCB algorithm shows a distinct spike in performance on the 11th step. Why is this? Note that for your answer to be fully satisfactory it must explain both why the reward increases on the 11th step and why it decreases on the subsequent steps. Hint: if $c = 1$, then the spike is less prominent. \square

Gradient Bandit Algorithms

- 이제까지 : action value 추정 \rightarrow 좋은 action 선택
- 이번 파트 : action에 대한 선호도(numerical preference) $\rightarrow H_t(a)$
- preference가 클수록 그 action이 자주 선택되었다는 의미
- 즉 reward에 대한 표현이 아니며 의미를 담고 있지 않음
- 그저 다른 action보다 얼마나 상대적으로 중요한지를 의미함

Gradient Bandit Algorithms

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

- $\pi_t(a)$: 상대적인 선호도에 따라 time t 에서 action a 가 선택될 확률
- 모든 action a 에 대해 $H_1(a) = 0$ 으로 초기화

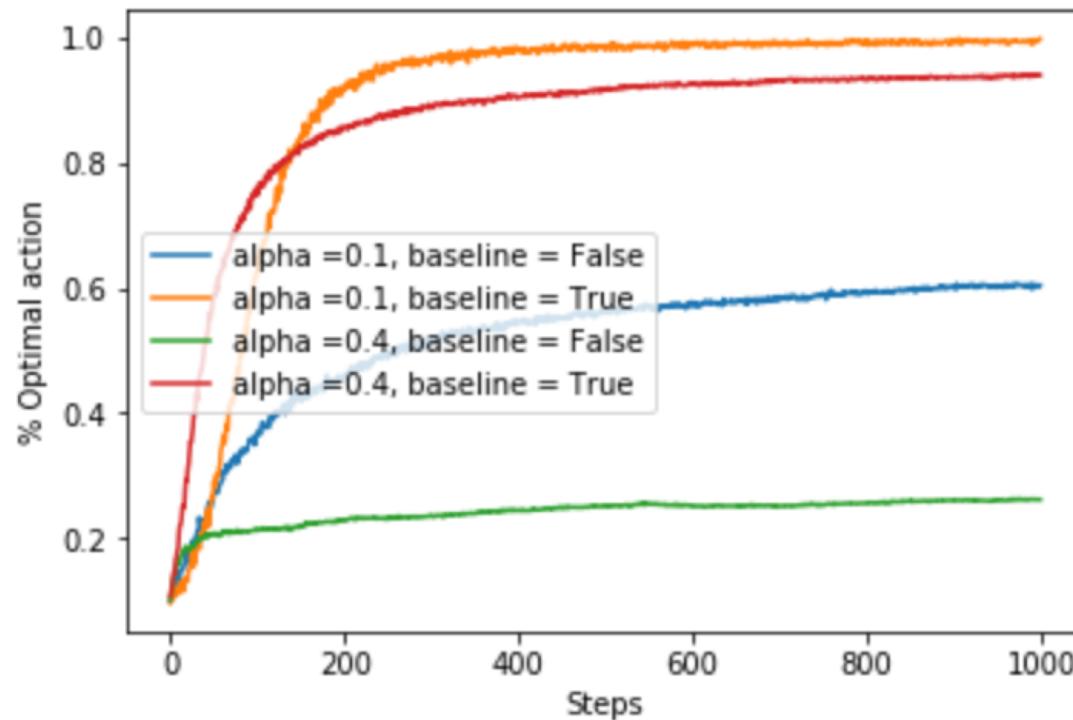
$$H_{t+1}(A_t) \doteq H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(A_t)), \quad \text{and}$$

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

$\bar{R}_t \in \mathbb{R}$: time t 까지의 모든 reward의 평균

R_t 가 \bar{R}_t 보다 크면 나중에 A_t 가 선택될 확률이 높아짐 ($R_t - \bar{R}_t > 0$)

Gradient Bandit Algorithms



(without baseline : $\bar{R}_t = 0$)

Stochastic Gradient Ascent?

- Gradient Ascent?

Stochastic Gradient Ascent

- main assertion : numerical preference learning method is equal to stochastic gradient ascent

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

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

R_t 의 기대값을 극대화!

Stochastic Gradient Ascent

- 증명은 책 참조...

Associative Search (contextual bandit)

- k -armed bandit problem은 고정된 하나의 상황에서 서로 독립된 k 개의 action 중 가장 높은 기댓값을 갖는 action을 선택하는 문제
- action과 situation(state?) 간의 연관성이 존재하지 않음
→ nonassociative task

Associative Search (contextual bandit)

- 만약 슬롯머신이 각 time step에서 여러 상태를 가질 수 있다면?
- 슬롯머신의 상태에 따라 action의 value가 달라진다면?
→ associative search task (contextual bandit)

Associative Search (contextual bandit)

- k -armed bandit problem ~ associative search ~ full RL
- action i reward와 다음 state까지 결정하는 요인이 된다면?
→ Full reinforcement learning problem

Summary

