Thompson Sampling

JaeHong Ryu

‘확률형 아이템 정보 완전 공개’ 게임법 전부 개정안, 尹 정부서 처리될까

확률형 아이템 준수율 낮은 게임 17종… “대부분 외산”

게임 '확률형아이템' 다시 도마 위 올랐다

[이슈진단+] 게임법 개정안 본회의 상정 행보 시작

공정위, '확률형 아이템 논란' 넥슨 현장조사 착수

게임사, '확률형 아이템'에 답해야

“확률형아이템 규제 시급, 해외 게임사 구속력도 갖춰야”

넥슨에 칼 빼든 공정위…확률형아이템 조작의혹 현장조사

학생 모집
95/210
41,935,996
21,583
픽업1
픽업2
상시
2022/07/12 점검 후~2022/07/26 12:00까지
픽업 모집!
★3 체리노(온천) 출현 확률 UP!★3 치나츠(온천)도 모집가능!
10회 모집 시, ★2 이상 학생 1명 확정!
※ 이미 모집한 학생은 엘리그마와 해당 학생의 엘레프로 변환됩니다.
120
모집 1회
1200
모집 10회
Point
모집 포인트
0
학생 선택
확률 정보
학생 정보

How To Maximize The Payoff?

Since the number of rounds is fixed,

- Exploration : getting the information of the machine by experiments
- Exploiting : getting the best reward

‘Multi-Armed Bandit Problem’

- There are many slot machines. For each machine the algorithm is fixed, and unknown to the player.
- The player only can see the outcome.
- Opportunities are limited.
- How to get the best results within the given opportunity?

Bernoulli Bendit

- Suppose there are K actions, i.e. K-many slot machines.
- Any action yields a success (S) or a failure (F).
- Action k produces a success with probability $\theta_k \in [0,1]$.
- The success probability $\theta = (\theta_1, \cdots, \theta_k)$ is **unknown** and **fixed** over time.
- Total possible periods T is **fixed** and relatively large compared to K.
- How to maximize the cumulative number of successes over T periods?

Bernoulli Bendit

Exploration : learning the success probability $\theta = (\theta_1, \cdots, \theta_k)$

Exploitation : maximizing the payoff

"Machine"

Greedy Algorithm

ϵ-Greedy Algorithm

Thompson Sampling (TS)

Greedy Algorithm

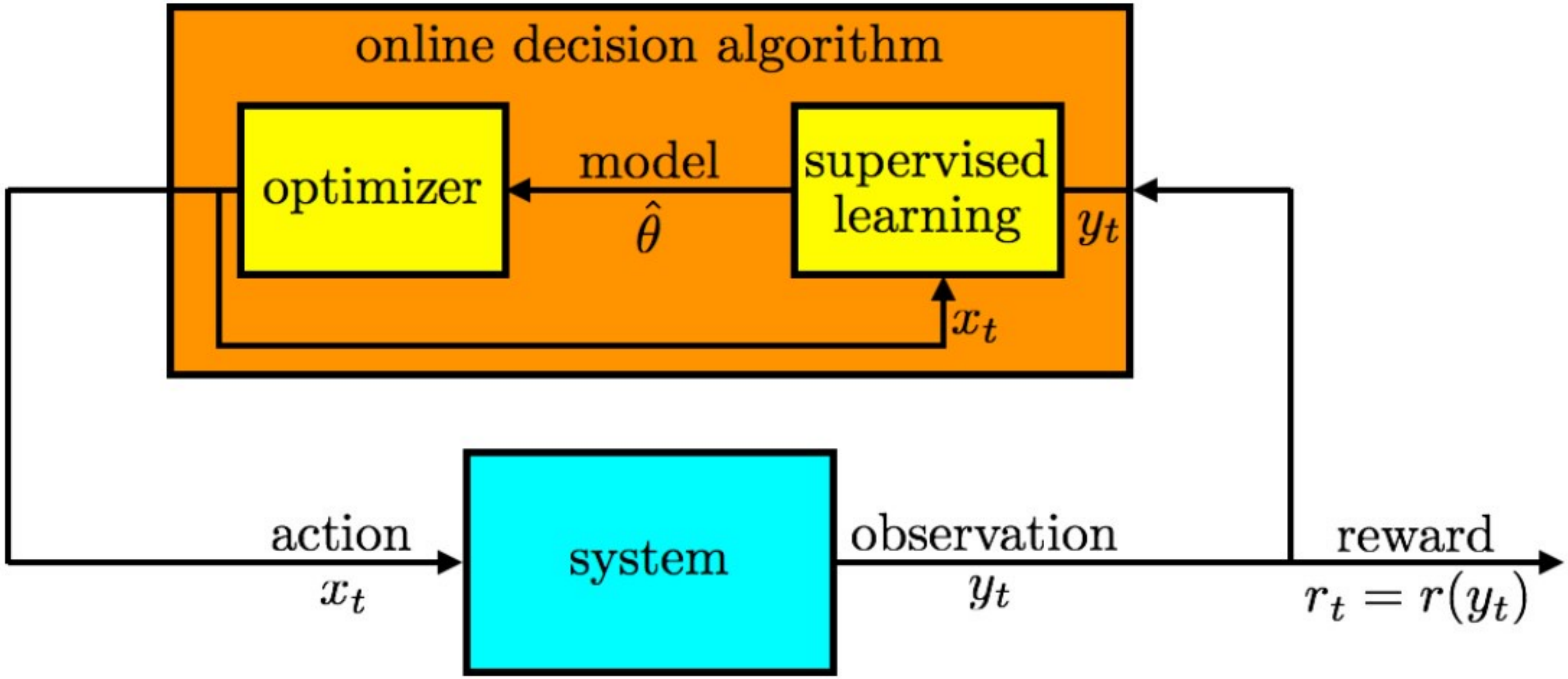
- The simplest and most common algorithm

For each time t, x_t is the action that the machine selects and y_t is an observed reward.

1. Estimate a model $\hat{\theta}$ from the history data $\boldsymbol{H}_{t-1}=\big((x_1, y_1), \cdots, (x_{t-1}, y_{t-1})\big)$.
2. Assume $\hat{\theta} = \theta$. Predict the reward $r_t = y_t$.
3. Using the predicton the machine selects the action x_t that maximizes r_t.
4. Observe y_t and add the result (x_t, y_t) to the history data.

This algorithm maximizes the immediate reward, thus greedy.

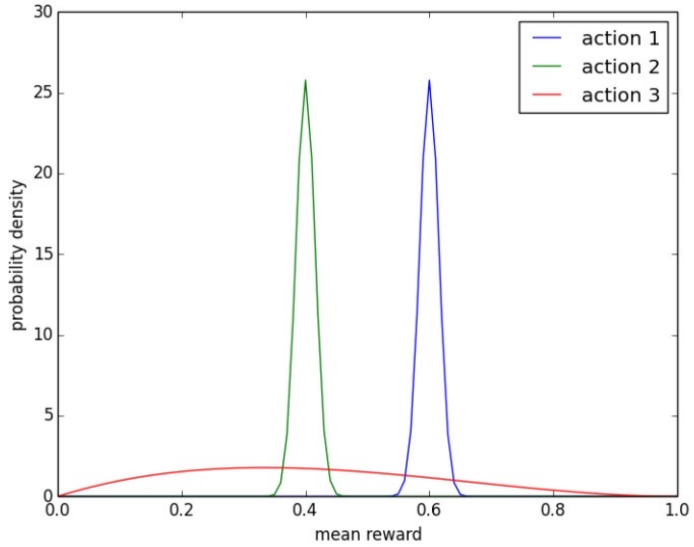
Greedy Algorithm



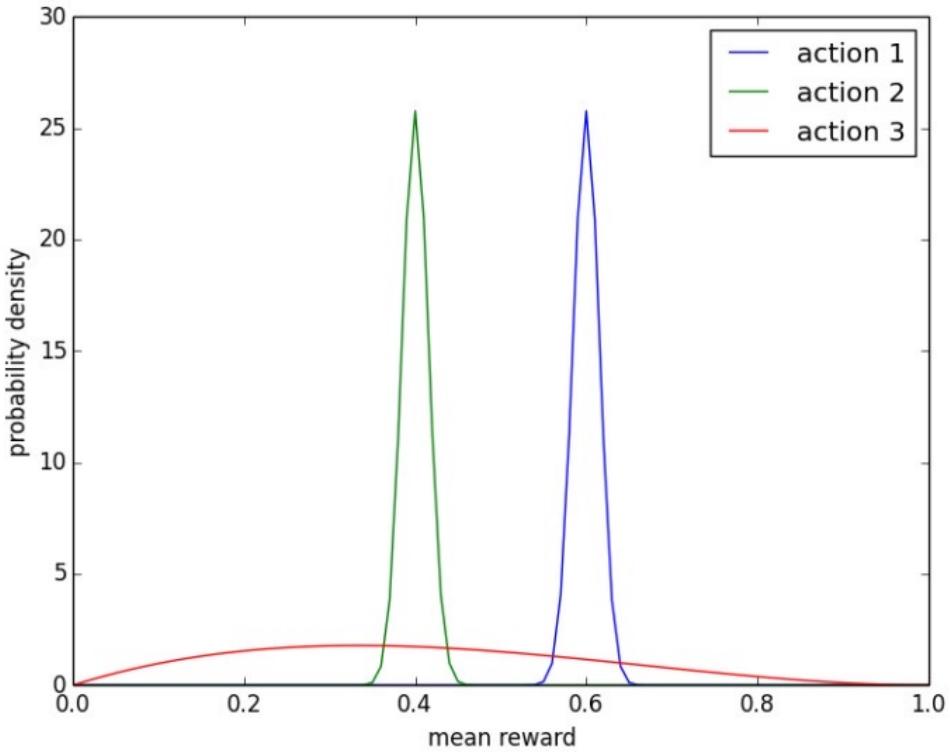
Greedy Algorithm : Example

- There are 3 actions; $K = 3$, with unknown mean rewards $\theta \in \mathbb{R}^3$.
- Any action k generates a reward 1 with probability θ_k , otherwise 0.
- For each time t, an action k is selected and the agent observes the reward.
- The agent believes the mean rewards θ can be expressed in terms of posterior distributions.

Action	1	2	3
#Tries	1000	1000	3
#Rewards	600	400	1



Greedy Algorithm : Example



- The machine would select the action 1.
- But there is a possibility : $\theta_3 > \theta_1$.
- So the machine should try the action 3 more, however, it will unlikely do that.
- The uncertainty in θ_3 will be remained.

∴There are some cases that the result with the greedy algorithm is not the maximum.

ϵ-Greedy Algorithm

- Dithering : Adding a random element to the greedy algorithm.
- ϵ -greedy exploration is the mixture of the greedy algorithm and the random.
- In ϵ-greedy exploration, the machine selects the greedy action with probability $1 - \epsilon$ and otherwise selects an action uniformly at random.
- Compared to the greedy algorithm, ϵ-greedy exploration reduces the uncertainty in the mean rewards.
- But ϵ-greedy exploration wastes more resources than the greedy algorithm.

Beta Bernoulli Bendit

- Suppose there are K actions, i.e. K-many slot machines.
- Any action yields a success (S) or a failure (F).
- Action k produces a success with probability $\theta_k \in [0,1]$.
- The success probability $\theta = (\theta_1, \cdots, \theta_k)$ is **unknown** and **fixed** over time.
- But the agent believes that for each k, θ_k is beta-distributed with parameters α_k and β_k.
- Total possible periods T is **fixed** and relatively large compared to K.
- How to maximize the cumulative number of successes over T periods?

Beta Bernoulli Bendit

- The probability density function of θ_k is

$$p(\theta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} {\theta_k}^{\alpha_k - 1}(1 - \theta_k)^{\beta_k - 1}$$

  where Γ denotes the gamma function.
- The distribution is updated according to Bayes' Rule.
- The parameters α_k and β_k can be updated according to the below rule.

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } x_t \neq k \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t) & \text{if } x_t = k \end{cases}$$

- The mean and variance of the distribution becomes decreased, as a result, the distribution becomes more concentrated as the period goes on.

Greedy Algorithm vs Thompson Sampling

Algorithm 1 BernGreedy(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #estimate model:
  for $k = 1, \ldots, K$ **do**
    $\boxed{\hat{\theta}_k \leftarrow \alpha_k / (\alpha_k + \beta_k)}$
  end for

  #select and apply action:
  $x_t \leftarrow \operatorname{argmax}_k \hat{\theta}_k$
  Apply x_t and observe r_t

  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t} + r_t, \beta_{x_t} + 1 - r_t)$
end for

Algorithm 2 BernTS(K, α, β)

for $t = 1, 2, \ldots$ **do**
  #sample model:
  for $k = 1, \ldots, K$ **do**
    $\boxed{\text{Sample } \hat{\theta}_k \sim \text{beta}(\alpha_k, \beta_k)}$
  end for

  #select and apply action:
  $x_t \leftarrow \operatorname{argmax}_k \hat{\theta}_k$
  Apply x_t and observe r_t

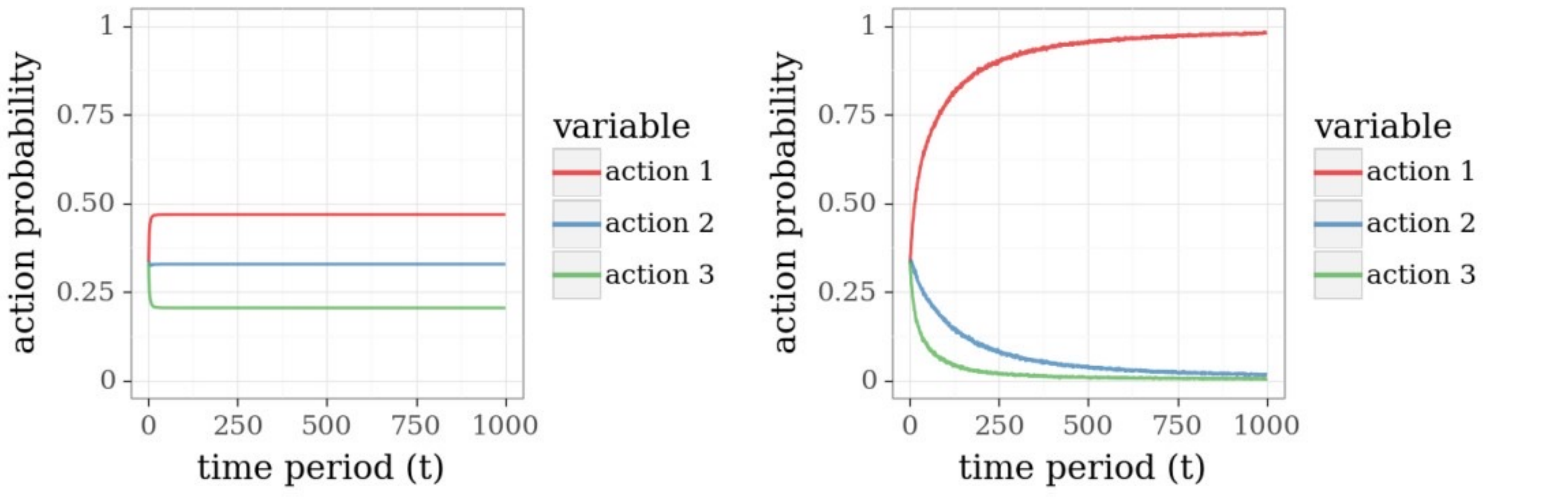
  #update distribution:
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t} + r_t, \beta_{x_t} + 1 - r_t)$
end for

- Estimate the model $\hat{\theta}_k$ using only the mean of the beta distribution.
- $\hat{\theta}_k = \alpha_k / (\alpha_k + \beta_k)$

- Estimate the model $\hat{\theta}_k$ sampling from the beta distribution.
- $\hat{\theta}_k \sim B(\alpha_k + \beta_k)$

Greedy Algorithm vs Thompson Sampling

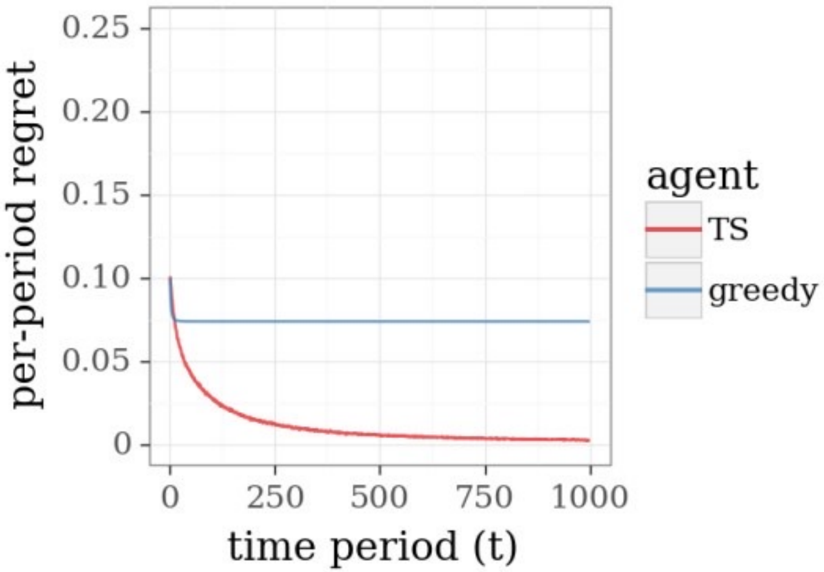
- Consider 3-armed beta-Bernoulli bandit with mean rewards $\theta_1 = 0.9$, $\theta_2 = 0.8$, and $\theta_3 = 0.7$.
- Let the prior distribution over each mean reward be uniform.
- The results of 1000 independent simulations of each algorithm are the below.



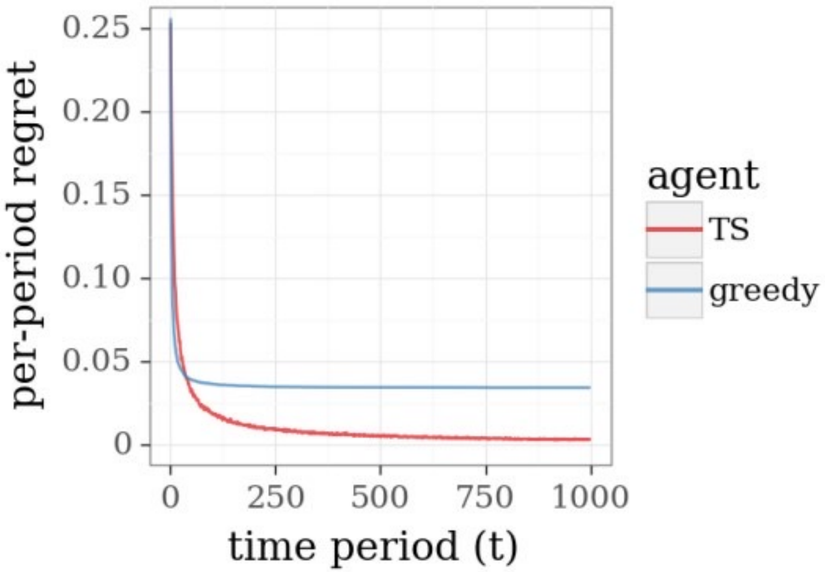
(a) greedy algorithm **(b)** Thompson sampling

Greedy Algorithm vs Thompson Sampling

- The per-period regret of an algorithm over a period t is the difference between the the mean reward of an optimal action and the action selected by the algotirhm
$$regret_t(\theta) = max_k\theta_k - \theta_{x_t}$$
- For 3-armed Bernoulli Bandit, the per-period regrets of the greedy algorithm and TS algorithm over 1000 periods are the below :



(a) $\theta = (0.9, 0.8, 0.7)$



(b) average over random θ

Greedy Algorithm vs Thompson Sampling

- Consider a more general setting. Let X be the action set, which can be finite or infinite.
- Applying action x_t, the agent observes an outcome y_t and enjoy the reward $r_t = r(y_t)$.
- The system randomly generates according to a conditional probability measure $q_\theta(\cdot\,|x_t)$.
- The value of θ is unknown but the agent represent his uncertainty using a prior distribution p.
- The machine generates model parameters $\hat{\theta}$ using the distribution.
- Using the model , the machine selects the action x_t maximizing r_t, observes the outcome $\hat{\theta}$ and updates the distribution p.

Greedy Algorithm vs Thompson Sampling

Algorithm 3 Greedy($\mathcal{X}, p, q, r$)

```
for t = 1, 2, ... do
    #estimate model:
    θ̂ ← E_p[θ]
    #select and apply action:
    x_t ← argmax_{x∈X} E_{q_θ̂}[r(y_t)|x_t = x]
    Apply x_t and observe y_t
    #update distribution:
    p ← P_{p,q}(θ ∈ ·|x_t, y_t)
end for
```

Algorithm 4 Thompson($\mathcal{X}, p, q, r$)

```
for t = 1, 2, ... do
    #sample model:
    Sample θ̂ ~ p
    #select and apply action:
    x_t ← argmax_{x∈X} E_{q_θ̂}[r(y_t)|x_t = x]
    Apply x_t and observe y_t
    #update distribution:
    p ← P_{p,q}(θ ∈ ·|x_t, y_t)
end for
```