Non-stationary MAB

day.lee

kakao

목차

Classical MAB

Non-stationary MAB

- 환경
    - Slowly-varying
    - Abruptly-changing
- Policies
    - Actively adaptive
    - Passively adaptive
- Summary

Classical MAB

Stationary

- 우리가 모르는 best arm이 존재함
- Learner 는 exploration과 exploitation을 적절히 사용해가면서 best arm을 찾아감

Stochastic:

- 각 arm 은 bandit이 알지 못하는 reward distribution을 가지고있음
- 매 라운드마다 K arm들 중 하나를 선택하여 그에 상응하는 실제 reward를 관측함

$$Regret = \sum_{t=1}^{T} E[\mu_{j*} - \mu_{j_t}]$$

Goal: minimize the regret over time

kakao

Classical MAB

- Classical bandit algorithm들의 regret lower bound (Lai and Robbins, 1985):

$$\Omega(log T)$$

- UCB, ThompsonSampling, Epsilon-n Greedy 알고리즘들의 tight bound (Agrawal and Goyal 2012):

$$O(\log T)$$

kakao

Non-stationary, Stochastic MAB

Rewards follow some unknown distribution

- Non-stationary 에서는 reward distribution이 시간에 따라 변화합니다
- best arm 이라고 생각했던 arm이 더이상 best arm이 아닐 수가 있기 때문

-> 기존의 MAB algorithm이 위 환경에서는 좋은 성능을 못 보임

Non-stationary, Stochastic MAB

$$Regret = \sum_{t=1}^{T} E[\mu_{j*} - \mu_{j_t}]$$

$$Regret = \sum_{t=1}^{T} \mu_{j_t *} - E[\sum_{j=1}^{N} \sum_{t=1}^{T} 1_{\{j_t = j\}} \mu_j(t)]$$

kakao

Non-stationary, Stochastic MAB

Expected cumulative regret

- Logarithmic bound ✘
- **Sublinear bound ✔** $O(\sqrt{T})$
    - Garivier와 Moulines (2008) 가 2008년에 낸 논문에의하면 non-stationary 환경에서는 그 어떤 알고리즘을 사용하더라도 이보다 낮은 regret bound를 성취할 수 없다고 증명함

kakao

Non-stationary (실험) 환경

kakao

Slowly-varying

급격한 change가 아닌 총 T시간동안 arm들의 reward distribution이 천천히 바뀜

- 어느 두 시간 대 사이에서의 arm의 reward가 바뀌는 수치가 아주 작고 아래와 같이 upper-bounded 되어있음

t와 t+1 사이에 arm의 mean reward는 최대 ϵ_T 만큼 변동 할 수 있음:

$$\epsilon_T \in O(T^{-\kappa})$$

kakao

Abruptly-changing

Piecewise, Switching

- arm의 rewards distribution이 특정 시간동안 일정하다가 알 수 없는 시간 대에 다른 값으로 변화한다
- distribution이 급격히 변화하는 구간 = breakpoint
- 총 T시간동안 발생하는 breakpoint의 수를 다음과 같이 정의:

$$\Upsilon_T \in O(T^v)$$

kakao

Non-stationary Policies

kakao

1. Passively Adaptive Policies

Passively Adaptive Policies

Passivley adaptive policy에서는 bandit이 앞서말한 reward distribution의 변화를 인지하지 못한다

- 수동적으로 reward distribution의 변화를 따라감
- 최근 데이터들만 사용해서 ‘현재’의 best arm 을 따라감

Discounted UCB

UCB 계산에 gamma term을 더해줌으로써 지난 관측 결과의 영향력을 감소시킨다

$$UCB = \bar{X}_t(\gamma, i) + c_t(\gamma, i)$$

aurochs 라이브러리에있는 dgp와 매우 비슷!

$$\bar{X}_t(\gamma, i) = \frac{1}{N_t(\gamma, i)} \sum_{s=1}^{t} \gamma^{t-s} X_s(i) \mathbb{1}_{\{I_s = i\}}\ , \quad N_t(\gamma, i) = \sum_{s=1}^{t} \gamma^{t-s} \mathbb{1}_{\{I_s = i\}},$$

$$c_t(\gamma, i) = 2B \sqrt{\frac{\xi \log n_t(\gamma)}{N_t(\gamma, i)}}\ , \quad n_t(\gamma) = \sum_{i=1}^{K} N_t(\gamma, i)\ ,$$

kakao

Discounted UCB

UCB 계산에 gamma term을 더해줌으로써 지난 관측 결과의 영향력을 감소시킨다

Remark 3 *If horizon T and the growth rate of the number of breakpoints Υ_T are known in advance, the discount factor γ can be chosen so as to minimize the RHS in Equation 2. Taking $\gamma = 1 - (4B)^{-1}\sqrt{\Upsilon_T/T}$ yields:*

$$\mathbb{E}_\gamma\left[\tilde{N}_T(i)\right] = O\left(\sqrt{T\Upsilon_T}\log T\right) .$$

Assuming that $\Upsilon_T = O(T^\beta)$ for some $\beta \in [0,1)$, the regret is upper-bounded as $O\left(T^{(1+\beta)/2}\log T\right)$. In particular, if $\beta = 0$, the number of breakpoints Υ_T is upper-bounded by Υ independently of T, taking $\gamma = 1-(4B)^{-1}\sqrt{\Upsilon/T}$, the regret is bounded by $O\left(\sqrt{\Upsilon T}\log T\right)$. Thus, D-UCB matches the lower-bound of Theorem 13 up to a factor $\log T$.

kakao

Sliding Window UCB

Sliding window는 지난 τ plays만을 이용해서 UCB를 계산합니다

$$UCB = \bar{X}_t(\tau, i) + c_t(\tau, i)$$

$$\bar{X}_t(\tau, i) = \frac{1}{N_t(\tau, i)} \sum_{s=t-\tau+1}^{t} X_s(i) \mathbb{1}_{\{I_s = i\}}, \quad N_t(\gamma, i) = \sum_{s=1}^{t} \gamma^{t-s} \mathbb{1}_{\{I_s = i\}},$$

$$c_t(\tau, i) = B \sqrt{\frac{\xi \log(t \wedge \tau)}{N_t(\tau, i)}},$$

kakao

Sliding Window UCB

Sliding window는 지난 τ plays만을 이용해서 local empirical average 를 트랙킹 한다

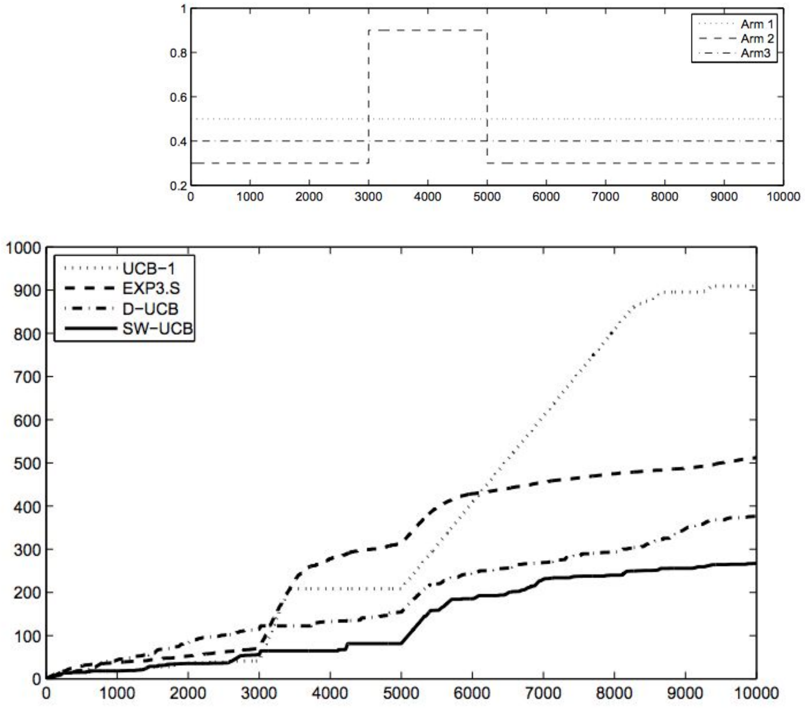
Remark 9 *If the horizon T and the growth rate of the number of breakpoints Υ_T are known in advance, the window size τ can be chosen so as to minimize the RHS in Equation (7). Taking $\tau = 2B\sqrt{T\log(T)/\Upsilon_T}$ yields*

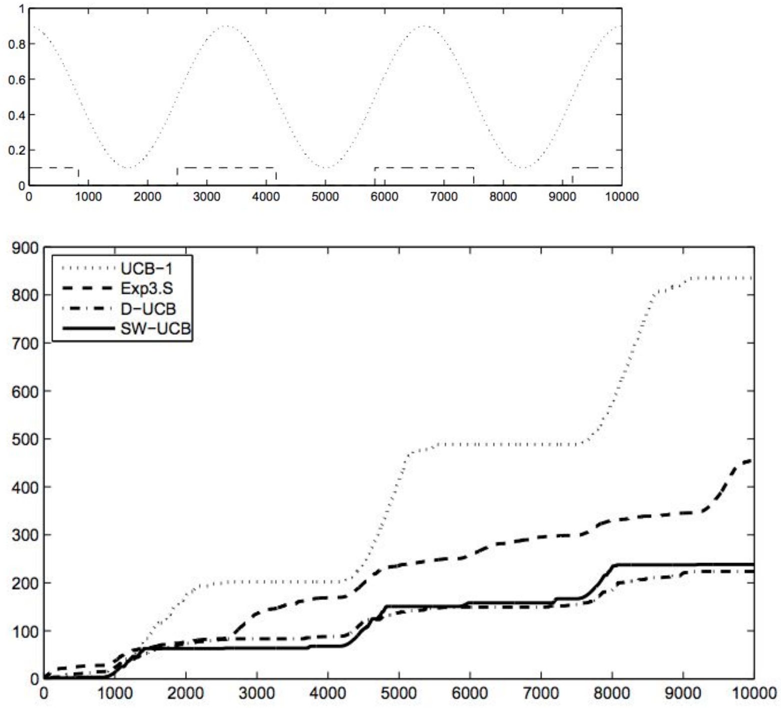
$$\mathbb{E}_\tau\left[\tilde{N}_T(i)\right] = O\left(\sqrt{\Upsilon_T T \log T}\right) .$$

Assuming that $\Upsilon_T = O(T^\beta)$ for some $\beta \in [0,1)$, the average regret is upper-bounded as $O\left(T^{(1+\beta)/2}\sqrt{\log T}\right)$. In particular, if $\beta = 0$, the number of breakpoints Υ_T is upper-bounded by Υ independently of T, then with $\tau = 2B\sqrt{T\log(T)/\Upsilon}$ the upper-bound is $O\left(\sqrt{\Upsilon T\log T}\right)$. Thus, SW-UCB matches the lower-bound of Theorem 13 up to a factor $\sqrt{\log T}$, slightly better than the D-UCB.

kakao

D-UCB vs SW-UCB





2. Actively Adaptive Policies

Actively Adaptive Policies

Change detection 알고리즘을 적용하여 환경에 일어나는 변화를 모니터링 해가며 일정 수치를 넘으면 bandit algorithm을 리셋팅해줌

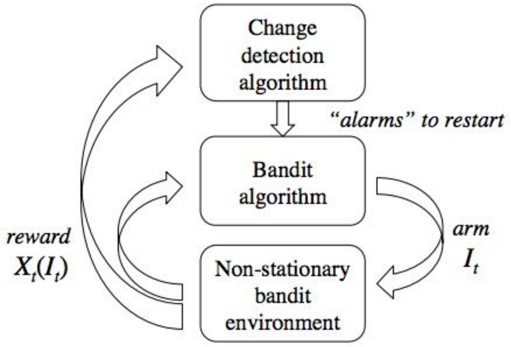


Figure 1: Change-detection based framework for non-stationary bandit problems

gamma-restart (D-UCB와 비슷) 등과 같은 방법으로 리셋

kakao

CD algorithm: Two-sided CUSUM UCB

Algorithm 2 Two-sided CUSUM

Require: parameters ϵ, M, h and $\{y_k\}_{k\geq 1}$

Initialize $g_0^+ = 0$ and $g_0^- = 0$.
for each k **do**
  Calculate s_k^- and s_k^+ according to (6).
  Update g_k^+ and g_k^- according to (7).
  if $g_k^+ \geq h$ or $g_k^- \geq h$ **then**
    Return 1
  end if
end for

맨 처음 M samples를
사용해서 구한 이 arm의
평균 reward 값

$$\boxed{(s_k^+, s_k^-) = (y_k - \hat{u}_0 - \epsilon, \hat{u}_0 - y_k - \epsilon)\mathbb{1}_{\{k>M\}}.}$$

$$\boxed{g_k^+ = \max(0, g_{k-1}^+ + s_k^+),\ g_k^- = \max(0, g_{k-1}^- + s_k^-).}$$

CD algorithm: PHT

현재의 best arm 을 통해 지난 T 라운드동안 얻은 reward를 series로 나타냈을 때 (i.e. x1, x2, x3, … ,xT) 이 series가 하나의 분포도로 설명될 수 있는 지 살펴본다 (null hypothesis)

$$\bar{x}_t = \frac{1}{t}\sum_{\ell=1}^{t} x_\ell$$

이제까지 (t) reward의 평균

$$m_T = \sum_{t=1}^{T}(x_t - \bar{x}_t + \delta)$$

지금 받은 reward와 평균 reward의 차이

$$M_T = \max\{m_t,\ t = 1 \ldots T\}$$

이제까지의 구한 평균과의 차이 중 최대 값

$$PH_T = M_T - m_T$$

이제까지 최대의 평균과의 차이와 현재의 평균과의 차이를 뺀 값

$$Return(PH_T > \lambda)$$

kakao

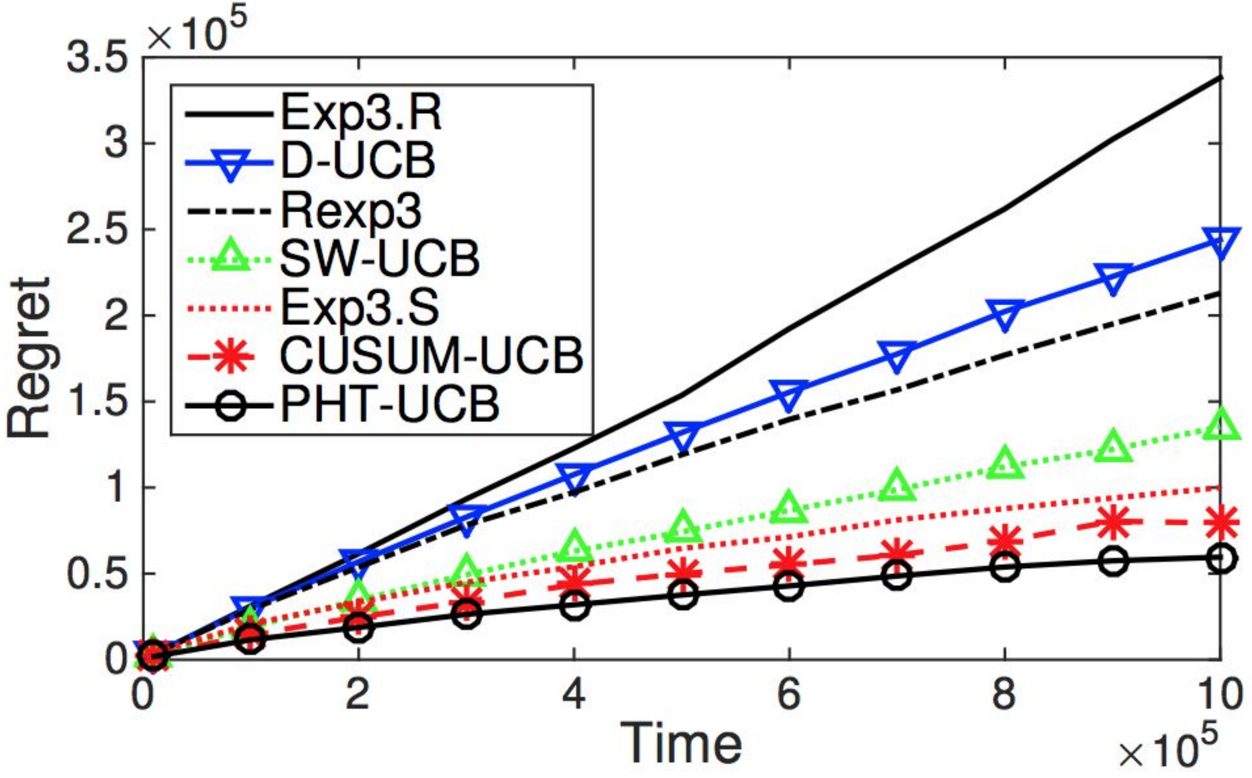
CUSUM vs PHT

현재 best arm 이 i*이고 best arm의 평균 reward를 μ* 라고 했을 때 3 종류의 변화가 일어날 수 있다

1. best arm은 그대로이지만 μ*가 바뀐다
2. best arm이 아닌 다른 arm의 reward가 best arm의 reward (μ*)를 뛰어넘을 정도로 상승한다
3. best arm의 reward인 μ*가 하락함에 따라 다른 arm이 best arm이 된다

PHT는 3번째 change만 고려함

kakao



(b) Under the switching environment

Figure 2: Regret over synthetic datasets

kakao

Summary

kakao

Abruptly-changing vs Slowly-varying

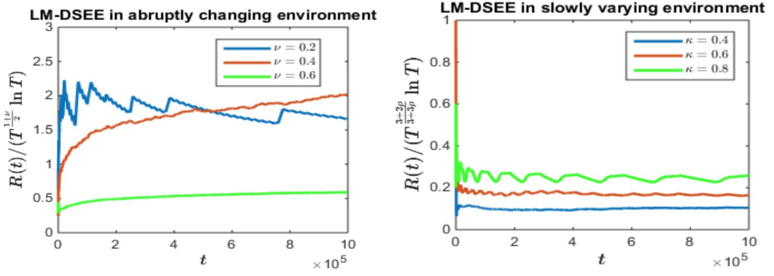


Fig. 1. The performance of the LM-DSEE algorithm in abruptly-changing and slowly-varying environments.

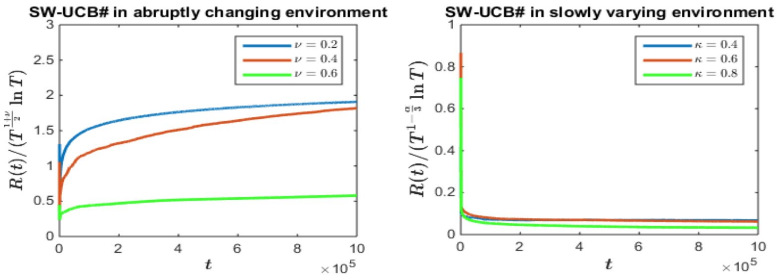


Fig. 2. The performance of the SW-UCB# algorithm in abruptly-changing and slowly-varying environments.

Passively Adaptive vs Actively Adaptive

Passive adaptive policy들의 이론적 regret 보장은 이미 수학적으로 증명되었다

- D-UCB, SW-UCB

하지만 actively adaptive policy는 regret 분석하는 것이 보다 힘들기 때문에 이론적 lower-bound/upper-bound가 증명되지않음

kakao

Passively Adaptive vs Actively Adaptive

Passive adaptive policy들의 이론적 regret 보장은 이미 수학적으로 증명되었다

- D-UCB, SW-UCB

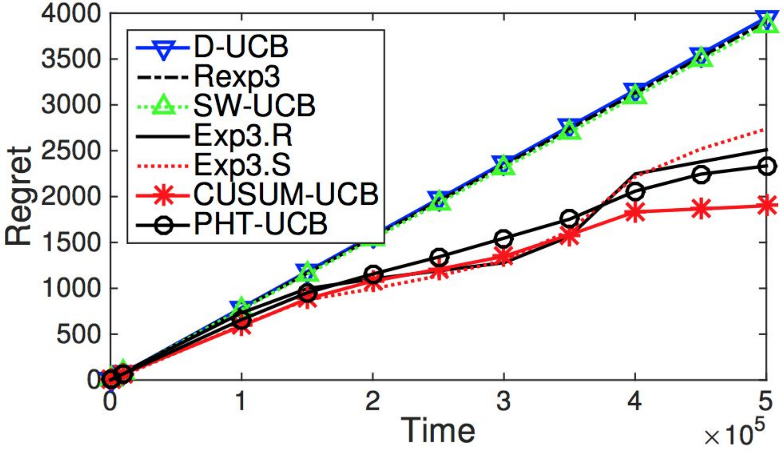
하지만 actively adaptive policy는 regret 분석하는 것이 보다 힘들기 때문에 이론적 lower-bound/upper-bound가 증명되지않음

- 예외) CUSUM-UCB in abruptly-changing

$$O\left(\sqrt{T\gamma_T \log \frac{T}{\gamma_T}}\right)$$

Real-world testing을 통해서 actively-adaptive policy들이 좋은 성능을 보인다는게 확인됨

kakao



(b) Regret

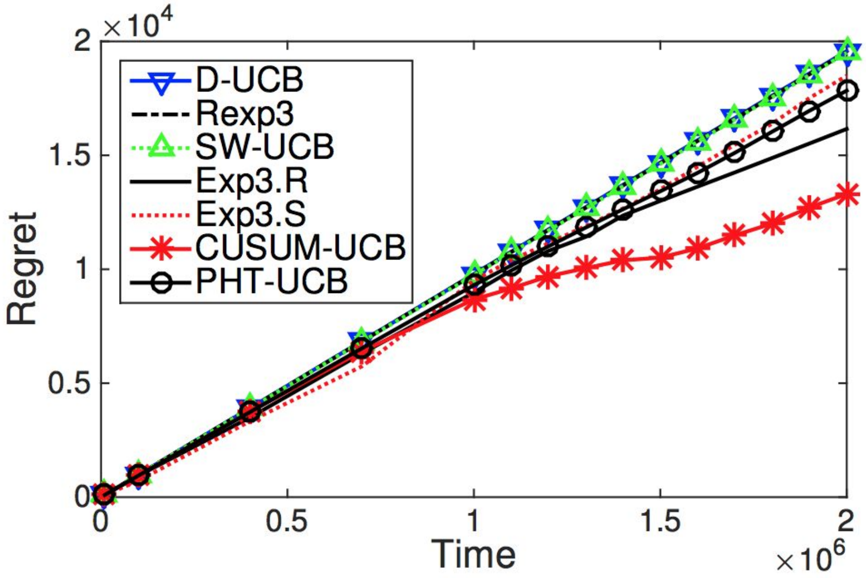


Figure 3: Rewards and regret over the Yahoo! dataset with $K = 5$

Figure 4: Regret over the Yahoo! dataset with $K = 100$

+

Non-stationary with contextual MAB

Adversarial MAB

Resources

https://arxiv.org/pdf/1711.03539.pdf - CUSUM
https://hal.archives-ouvertes.fr/hal-00113668/document - PHT
https://arxiv.org/pdf/1802.08380.pdf - Non-stationary (Abruptly-changing vs Slowly-varying)
https://arxiv.org/pdf/0805.3415.pdf - SW-UCB and D-UCB

감사합니다!

kakao