Thompson Sampling with Belief Update for Non-stationary Multi-armed Bandit Problem

Jian Gao, Chi-Guhn Lee

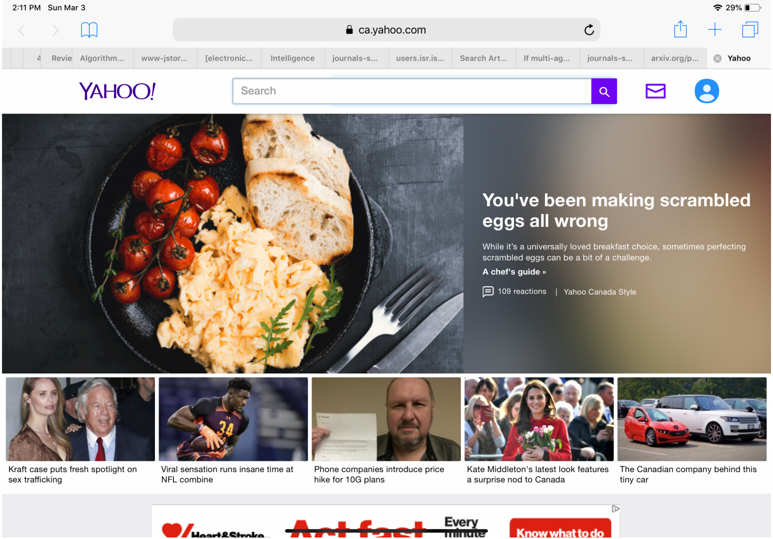
University of Toronto

Table of contents

Introduction

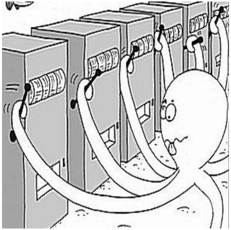
Introduction

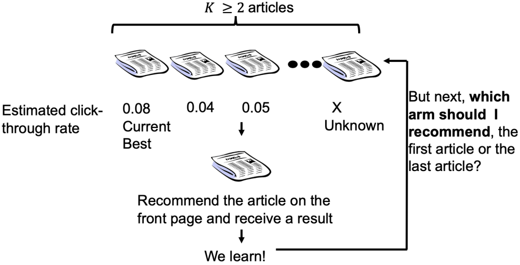
Introduction



Introduction

Multi-armed Bandit Problem





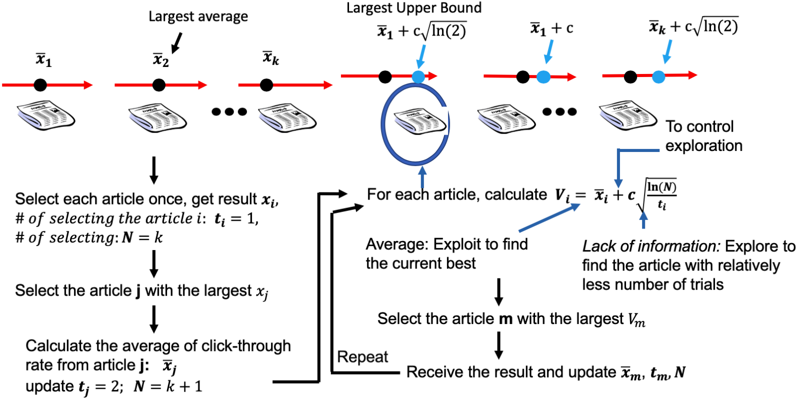
Objective is to minimize the regret after n plays

$$\mu^* n - \mu_j \sum_{j=1}^{K} \mathbb{E}[T_j(n)] \text{ where } \mu^* = \max_{1 \leq i \leq K} \mu_i$$

Algorithms for MAB

Upper Confidence Bound (UCB[1])

An index approach

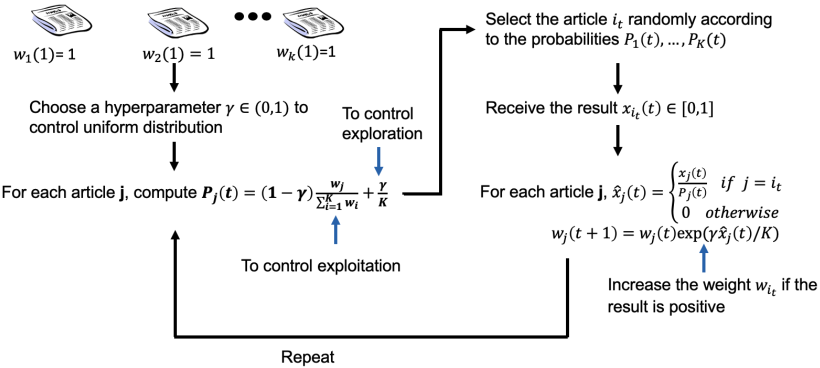


[1] Auer, Cesa-Bianchi & Fischer. "Finite-time analysis of the multiarmed bandit problem," Machine Learning, 47:235-256, 2002

Algorithms for MAB

Exponential-weight algorithm (Exp3[2])

A randomization approach

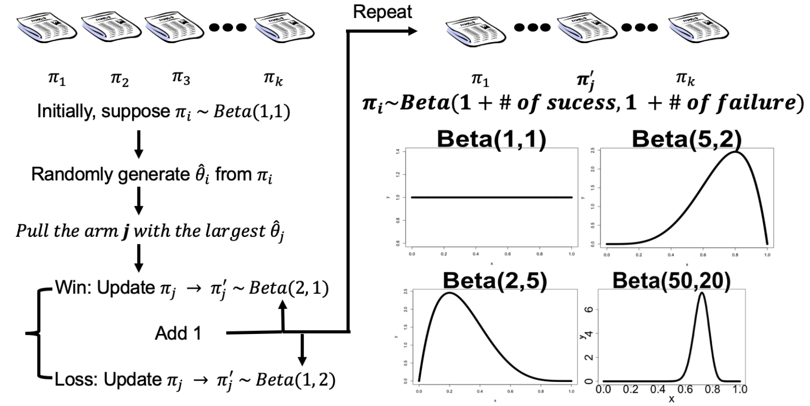


[2] Auer, Cesa-Bianchi, Freund & Schapire. "The nonstochastic multiarmed bandit problem," SIAM J.Comput., 32:48-77, 2003

Algorithms for MAB

Thompson Sampling[3]

A Bayesian approach

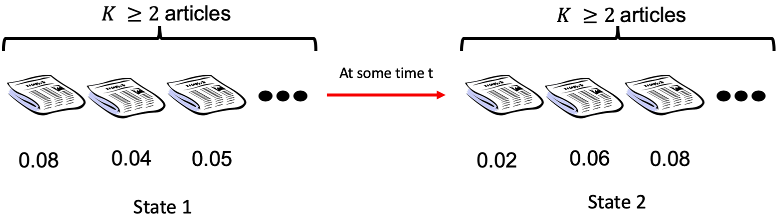


[3] Thompson. "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," Biometrika, 25:285?294, 1933

An Extension

Piecewise non-stationary environment

- Unrealistic to assume the static reward distributions
- Different users with different preference may access the page
- A piece-wise non-stationary environment
  - The preference can be static for some time before change



Existing Approaches

1. **Passive** approaches: decay or remove the weights on the rewards
   - Discounted UCB (D-UCB) [1]: UCB with a discount factor.
   - Sliding-Window UCB (SW-UCB) [2]: UCB with a time window.
   - Rexp3[3]: reset the Exp3 every T times.

2. **Active** approaches: monitor the rewards to detect a change point
   - EXP3.R [4]: reset EXP3 algorithm upon detecting a change point
   - M-UCB [5] and CUSUM-UCB[6] : reset UCB algorithm upon detecting a change point
   - Global Change-Point Thompson Sampling (Global-CTS) [7] and Global Switching Thompson Sampling with Bayesian Aggregation (Global-STS-BA) [8] : Thompson sampling with Bayesian change-point detection.

Drawbacks of the previous works:

- Passive approaches perform poorly
- Active approaches are expensive in computation & memory use
- Sensitive to hyperparameters

Change Point Detection

CUMSUM-UCB

- Change point detected if the cumulative drift (g_t^+ or g_t^-) $\geq$ threshold (h), where

$$g_t^+ = \max(0, g_{t-1}^+ + s_t^+), \text{ and } g_t^- = \max(0, g_{t-1}^- + s_t^-)$$

$$(s_t^+, s_t^-) = (y_t - \hat{\mu}_0 - \epsilon, \hat{\mu}_0 - y_t - \epsilon)1_{t>M}, \hat{\mu}_0 = \sum_{k=1}^{M} y_k/M, \epsilon > 0$$

Bayesian Change Detection

- Compute posterior for all possible run lengths r_t (# of steps since the last change point). That is, for each r_t and t,

$$P(r_t|x_{t-1}, D_{t-2}) = \frac{P(r_t, x_{t-1}, D_{t-2})}{P(x_{t-1}, D_{t-2})}$$

where x_{t-1} is the reward at time $t-1$, D_{t-2} is the reward history up to time $t-2$;

Thompson sampling with belief update

Thompson Sampling

Initialize $D = \emptyset$
for $i = 1, \ldots, T$ **do**
  Draw θ^t according to $\mathbb{P}(\theta|D)$
  Select $a_t = argmax_a \mathbb{E}(r|a, \theta^t)$
  Observe r_t
  $D = D \cup (a_t, r_t)$
end

TS for the Bernoulli Bandit

Input: α, β prior parameters of a Beta
Initialize $S_i = 0, F_i = 0$ /Counters
for $i = 1, \ldots, T$ **do**
  for $i = 1, \ldots, K$ **do**
    Draw θ^t from Beta$(S_i + \alpha, F_i + \beta)$
  end
  Select $a = argmax_i \theta_i$
  Observe r
  if r=1 **then**
    $S_a = S_a + 1$
  end
  else
    $F_a = F_a + 1$
  end
end

Belief update

Partially Observable Markov Decision Process (POMDP)

- State $s(\in S)$ of the environment is assumed **unobservable**
- Calculate posterior $b_{t+1}(s)$ given an observation x_t and a prior $b_t(s)$.
- An observation function $O(x_t|s,a)$ is unknown but transition function $P(ss')$ is assumed known.

That is,

$$b^{'}(s^{'}) = \eta O(x_t|s^{'},a) \sum_{s \in S} T(s^{'}|s,a) b(s)$$

where $\eta = \frac{1}{Pr(x_t|b,a)} = \frac{1}{\sum_{s^{'} \in S} O(x_t|s^{'},a) \sum_{s \in S} T(s^{'}|s,a) b(s)}$

Thompson Sampling with Belief Update

TSBU - Finite

Algorithm 1 TS-BU-Fin

procedure TS-BU-Fin(T, K, γ, L=None)
$t \leftarrow 0$ and $\forall k,\ \forall s,\ \alpha^0_{s,k} \leftarrow 1, \beta^0_{s,k} \leftarrow 1$
$P(s_1|s_1) = P(s_2|s_2) \leftarrow 1 - \gamma$
$P(s_2|s_1) = P(s_1|s_2) \leftarrow \gamma$
$b_0(s_1) \leftarrow 1$ and $b_0(s_2) \leftarrow 0$
$N \leftarrow 1$
for $t \leq T$ **do**
  $k_t \leftarrow \text{SelectArm}(\{\alpha^t\}, \{\beta^t\}, \{b_t\})$
  $x_t \leftarrow \text{Playarm}(k_t)$
  $b_t^{'} \leftarrow \text{UpdateBelief}(\{\alpha^t_{k_t}\}, \{\beta^t_{k_t}\}, \{b_t\}, x_t)$
  $\alpha^{t+1}_{k_t}, \beta^{t+1}_{k_t} \leftarrow \text{UpdateArm}(\{\alpha^t_{k_t}\}, \{\beta^t_{k_t}\}, \{b_t^{'}\}, x_t)$
  $b_{t+1} \leftarrow \text{NextBelief}(\{b_t^{'}\}, P)$
  if L is None or $N < L$ **then**
    if $\sum_{n=1}^{N} b_{t+1}(s_n) < b_{t+1}(s_{N+1})$ **then**
      $b_{t+1}(s_{N+1}) \leftarrow 1$
      $b_{t+1}(s_n) \leftarrow 0\ \forall n \in \{1, ..., N, N+2\}$
      $\alpha^{t+1}_k(s_{N+2}) \leftarrow 1, \beta^{t+1}_k(s_{N+2}) \leftarrow 1$
      $P(s_n|s_n) \leftarrow 1 - \gamma,\ \forall n$
      $P(s_m|s_n) \leftarrow \dfrac{\gamma}{N+1}, \forall m \neq n$
      $N \leftarrow N + 1$
    end if
  end if
end for

TSBU-Infinite

Algorithm 2 TS-BU-Inf

procedure TS-BU-Inf(T, K, γ)
$t \leftarrow 0$ and $\forall k,\ \forall s,\ \alpha^0_{s,k} \leftarrow 1, \beta^0_{s,k} \leftarrow 1$
$P(s_1|s_1) = P(s_2|s_2) \leftarrow 1 - \gamma$ and $P(s_2|s_1) = P(s_1|s_2) \leftarrow \gamma$
$b_0(s_1) \leftarrow 1$ and $b_0(s_2) \leftarrow 0$
for $t \leq T$ **do**
  $k_t \leftarrow \text{SelectArm}(\{\alpha^t\}, \{\beta^t\}, \{b_t\})$
  $x_t \leftarrow \text{Playarm}(k_t)$
  $b_t^{'} \leftarrow \text{UpdateBelief}(\{\alpha^t_{k_t}\}, \{\beta^t_{k_t}\}, \{b_t\}, x_t)$
  $\alpha^{t+1}_{k_t}, \beta^{t+1}_{k_t} \leftarrow \text{UpdateArm}(\{\alpha^t_{k_t}\}, \{\beta^t_{k_t}\}, \{b_t^{'}\}, x_t)$
  $b_{t+1} \leftarrow \text{NextBelief}(\{b_t^{'}\}, P)$
  if $b_{t+1}(s_1) < b_{t+1}(s_2)$ **then**
    $b_{t+1}(s_1) \leftarrow 1$ and $b_{t+1}(s_2) \leftarrow 0$
    $\alpha^{t+1}_k(s_1) \leftarrow \alpha^{t+1}_k(s_2), \beta^{t+1}_k(s_1) \leftarrow \beta^{t+1}_k(s_2)$ {Move knowledge}
    $\alpha^{t+1}_k(s_2) \leftarrow 1, \beta^{t+1}_k(s_2) \leftarrow 1$ {Re-initialize s_2}
  end if
end for

Numerical Studies

Performance Measure and Benchmark

Regret R_t

Instead of maximizing the rewards directly, the common measure of performance is cumulative regret:

$$R(T) = \sum_{t=1}^{T} \mu_t^* - E(\sum_{t=1}^{T} x_{k_t})$$

where $\mu_t^* = \max_{k \in \{1,\ldots,K\}} \mu_t^k$.

TS-oracle

Thompson Sampling Oracle (TS-oracle) [8] knows all the change points with certainty and resets Thompson sampling at these points.

Finite state space

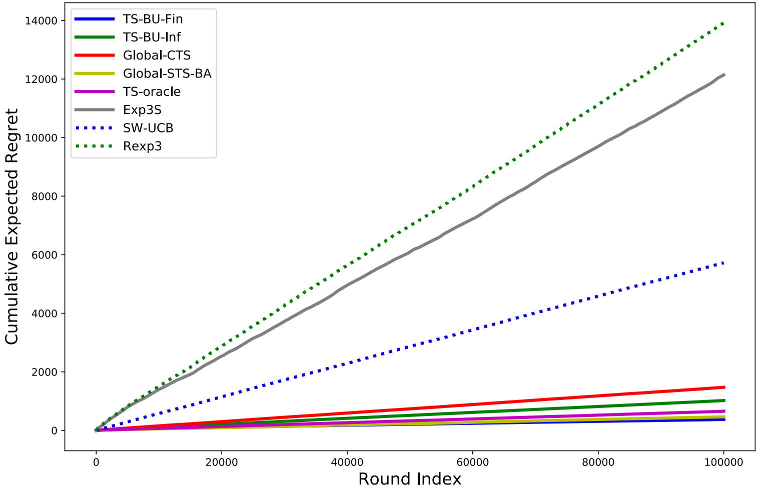
Abruptly Varying Environment

- Three states and three arms

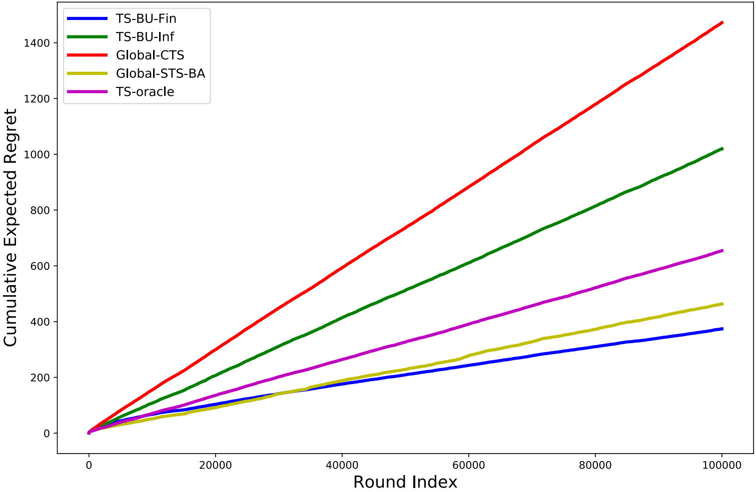
	a_1	a_2	a_3
s_1	0.1	0.9	0.3
s_2	0.8	0.2	0.4
s_3	0.2	0.1	0.9

- The change point occurs randomly at a given switching rate 10^{-3}
- State after change point is randomly chosen.
- The time horizon is $T = 10^5$

Finite state space



Finite state space



Finite state space

Observations:

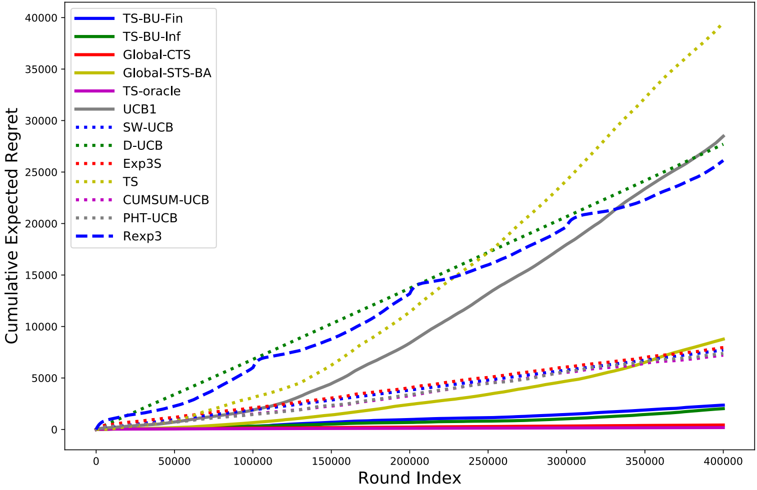
- Global-STS-BA shows the best performance early (approximately up to 5,000 rounds), but TS-BU-Fin eventually outperforms Global-STS-BA and all the other algorithms.
- The strong performance of Global-STS-BA at the beginning is possibly due to active exploration. That is, thanks to the large run-length support, Global-STS-BA detects new states more quickly.
- Eventually, TS-BU-Fin outperforms even TS-oracle, which does not memorize state specific information albeit it knows exactly when the change point occurs.

Infinite state space

Switching environment

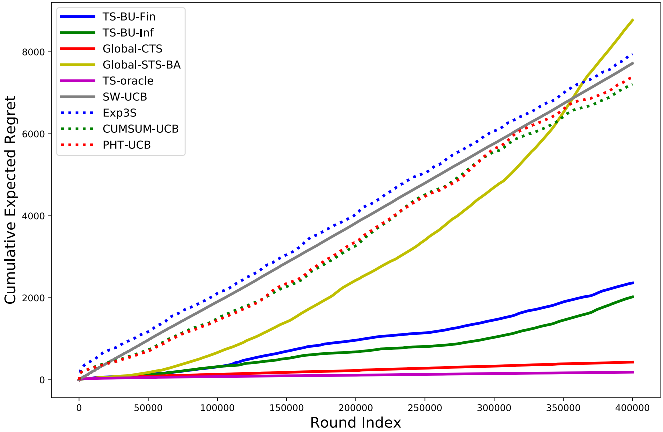
- The switching environment used in this section is adopted from [7]
- Five arms $K = 5$
- The rewards $\mu_{k,t}$ (the mean of an arm k) at time t changes abruptly and globally at a constant switching rate $\gamma = 10^{-5}$.
- The reward distributions are randomly generated from a uniform distribution $U(0, 1)$
- Horizon used is $T = 4 \times 10^5$

Experiment: Infinite state space



Experiment: Infinite state space

The algorithms with relatively strong performance



Observations:

- Performance of TS-BU-Inf is between Global-CTS and Global-STS-BA.
- TS-BU-Fin shows comparable performance with TS-BU-Inf.
- Global-CTS has a better performance than Global-STS-BA unlike the case of finite number of states; The possible reason is that the sampling step enhances the exploration efforts of Global-CTS, while the exploration of Global-STS-BA is less than Global-CTS due to the Bayesian aggregation step.
- When the switching rate is small, Global-STS-BA needs more time to detect change points. TS-BU-Inf, on the other hand, can balance the trade-off between exploration and exploitation better with less computation.

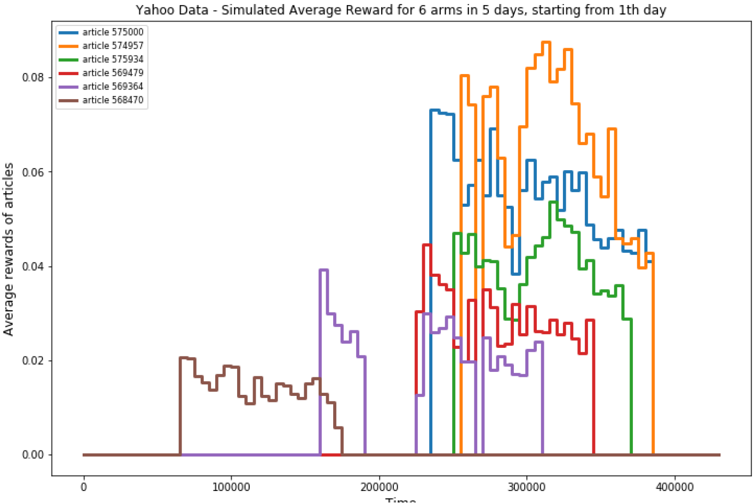
Yahoo! Dataset[4]

Set up:

- Binary value representing whether user clicked articles shown on the front page.
- Our goal is to maximize the click-through rate by selecting which article to be shown on the front page.
- We randomly choose a 5-day horizon ($T = 4.32 \times 10^5$) and six articles ($K = 6$) that were shown the most times during the chosen horizon.
- The click-through rates are computed by taking the average of the number of clicks on each article in every 5,000 seconds ($\gamma = 1/5000$).

[4]Yahoo! Front Page Today Module User Click Log Dataset on https://webscope.sandbox.yahoo.com

Experiment Yahoo! Dataset



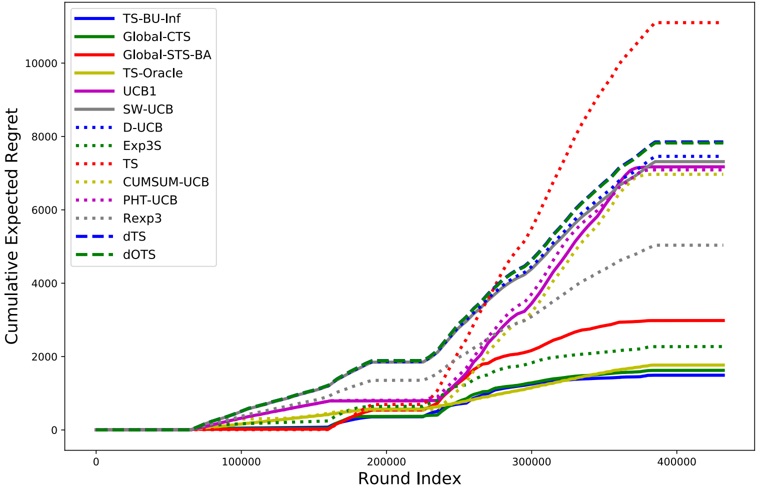
Experiment Yahoo! Dataset

Observations:

- We test our algorithm with other 11 algorithms, whose parameters were set up optimally or according to recommendations.
- TS-BU-Inf has the lowest cumulative regret (better than even TS-oracle)

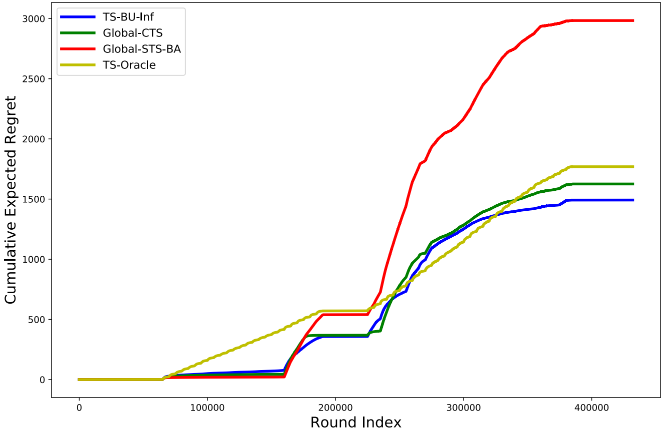
Experiment Yahoo! Dataset

Cumulative regret



Experiment Yahoo! Dataset

Cumulative regret of the best 3 algorithms



Conclusions

Conclusions

- We proposed new variants of Thompson sampling by integrating Bayesian belief updating capability
  - Thompson sampling with belief update - Finite (TS-BU-Fin)
- Numerical studies showed that TS-BU-Fin and TS-BU-Inf are competitive with the state-of-the-art algorithms
- TS-BU-Fin and TS-BU-Inf have significant benefits in computation and memory requirements.
- Unfortunately, due to the generality of Thompson sampling in the piece-wise stationary MAB studied in this paper, theoretical results on performance guarantee is still an open problem.

References

[1] Kocsis, L., Szepesv ari, *Discounted UCB*. In: 2nd PASCAL Challenges Workshop, Venice, Italy (April 2006)

[2] Garivier A., Moulines E. (2011) *On Upper-Confidence Bound Policies for Switching Bandit Problems*. In: Kivinen J., Szepesvri C., Ukkonen E., Zeugmann T. (eds) Algorithmic Learning Theory. ALT 2011. Lecture Notes in Computer Science, vol 6925. Springer, Berlin, Heidelberg

[3] Omar Besbes, Yonatan Gur, and Assaf Zeevi. *Stochastic multi-armed-bandit problem with non- stationary rewards*. In Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS14, pages 199207, Cambridge, MA, USA, 2014. MIT Press.

[4] Robin Allesiardo and Raphal Fraud. *Exp3 with drift detection for the switching bandit problem*. In Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, pages 17. IEEE, 2015.

[5] Yang Cao, Zheng Wen, Branislav Kveton, Yao Xie, *Nearly Optimal Adaptive Procedure for Piecewise-Stationary Bandit: a Change-Point Detection Approach*. arXiv preprint arXiv:1802.03692v2, 2018.

[6] F. Liu, J. Lee, and N. Shroff, *A change-detection based framework for piecewise-stationary multi-armed bandit problem*, arXiv preprint arXiv:1711.03539, 2017.

[7] Joseph Mellor and Jonathan Shapiro. *Thompson sampling in switching environments with bayesian online change point detection*. CoRR, abs/1302.3721, 2013.

[8] Rda Alami, Odalric Maillard, Raphael Fraud. *Memory Bandits: a Bayesian approach for the Switching Bandit Problem*. NIPS 2017 - 31st Conference on Neural Information Processing Systems, Dec 2017, Long Beach, United States. 2017. ¡hal-01811697¿