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

Jian Gao, Chi-Guhn Lee

University of Toronto
1. Introduction

2. Thompson sampling with belief update

3. Numerical Studies

4. Conclusions
Introduction
Introduction
Introduction
**Introduction**

**Multi-armed Bandit Problem**

---

**Objective** is to minimize the regret after \( n \) plays

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

Upper Confidence Bound (UCB\(^1\))

An index approach

Exponential-weight algorithm (Exp3\(^2\))

A randomization approach

\[ P_j(t) = (1 - \gamma) \frac{w_j}{\sum_{i=1}^{K} w_i} + \frac{\gamma}{K} \]

For each article \( j \), choose a hyperparameter \( \gamma \in (0,1) \) to control uniform distribution. To control exploration, select the article \( i_t \) randomly according to the probabilities \( P_1(t), ..., P_K(t) \).

Receive the result \( x_{i_t}(t) \in [0,1] \)

For each article \( j \), \( \hat{x}_j(t) = \begin{cases} x_j(t) & \text{if } j = i_t \\ 0 & \text{otherwise} \end{cases} \)

\[ w_j(t + 1) = w_j(t) \exp(\gamma \hat{x}_j(t)/K) \]

Increase the weight \( w_{i_t} \) if the result is positive.

Repeat

---

Algorithms for MAB

Thompson Sampling\textsuperscript{3}

A Bayesian approach

Initially, suppose $\pi_i \sim \text{Beta}(1,1)$

Randomly generate $\hat{\theta}_i$ from $\pi_i$

*Pull the arm $j$ with the largest $\hat{\theta}_j$*

Win: Update $\pi_j \rightarrow \pi'_j \sim \text{Beta}(2,1)$

Add 1

Loss: Update $\pi_j \rightarrow \pi'_j \sim \text{Beta}(1,2)$

\textsuperscript{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

\[ K \geq 2 \text{ articles} \]

State 1

\[
\begin{array}{ccc}
0.08 & 0.04 & 0.05 \\
\end{array}
\]

At some time \( t \)

\[ K \geq 2 \text{ articles} \]

State 2

\[
\begin{array}{ccc}
0.02 & 0.06 & 0.08 \\
\end{array}
\]
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
CUMSUM-UCB

- Change point detected if the cumulative drift \( (g_t^+ \text{ or } g_t^-) \geq \) 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 $\theta_t$ according to $P(\theta|D)$
- Select $a_t = \arg\max_{a} \mathbb{E}(r|a, \theta_t)$
- Observe $r_t$
- $D = D \cup (a_t, r_t)$

end

**TS for the Bernoulli Bandit**

**Input:** $\alpha, \beta$ 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 $\theta_t$ from Beta($S_i + \alpha, F_i + \beta$)
  - Select $a = \arg\max_i \theta_i$
  - 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

1: procedure TS-BU-Fin($T, K, \gamma, L$=None)
2: $t \leftarrow 0$ and $\forall k, s, \alpha^0_{s,k} \leftarrow 1, \beta^0_{s,k} \leftarrow 1$
3: $P(s_1|s_1) = P(s_2|s_2) \leftarrow 1 - \gamma$
4: $P(s_2|s_1) = P(s_1|s_2) \leftarrow \gamma$
5: $b_0(s_1) \leftarrow 1$ and $b_0(s_2) \leftarrow 0$
6: $N \leftarrow 1$
7: for $t \leq T$ do
8: $k_t \leftarrow \text{SelectArm}({\alpha^t}, {\beta^t}, \{b_t\})$
9: $x_t \leftarrow \text{Playarm}(k_t)$
10: $b_t \leftarrow \text{UpdateBelief}({\alpha^t}_{k_t}, {\beta^t}_{k_t}, \{b_t\}, x_t)$
11: $\alpha^{t+1}_{k_t}, \beta^{t+1}_{k_t} \leftarrow \text{UpdateArm}({\alpha^t}_{k_t}, {\beta^t}_{k_t}, \{b_t\}, x_t)$
12: $b_{t+1} \leftarrow \text{NextBelief}(\{b_t\}, P)$
13: if $L$ is None or $N < L$ then
14: if $\sum_{n=1}^N b_{t+1}(s_n) < b_{t+1}(s_{N+1})$ then
15: $b_{t+1}(s_{N+1}) \leftarrow 1$
16: $b_{t+1}(s_n) \leftarrow 0 \ \forall n \in \{1, \ldots, N, N+2\}$
17: $\alpha^{t+1}_{k_t}(s_{N+2}) \leftarrow 1, \beta^{t+1}_{k_t}(s_{N+2}) \leftarrow 1$
18: $P(s_n|s_n) \leftarrow 1 - \gamma, \ \forall n$
19: $P(s_m|s_n) \leftarrow \frac{\gamma}{N+1}, \forall m \neq n$
20: $N \leftarrow N + 1$
21: end if
22: end if
23: end for

**TSBU-Infinite**

Algorithm 2 TS-BU-Inf

1: procedure TS-BU-Inf($T, K, \gamma$)
2: $t \leftarrow 0$ and $\forall k, s, \alpha^0_{s,k} \leftarrow 1, \beta^0_{s,k} \leftarrow 1$
3: $P(s_1|s_1) = P(s_2|s_2) \leftarrow 1 - \gamma$ and $P(s_2|s_1) = P(s_1|s_2) \leftarrow \gamma$
4: $b_0(s_1) \leftarrow 1$ and $b_0(s_2) \leftarrow 0$
5: for $t \leq T$ do
6: $k_t \leftarrow \text{SelectArm}({\alpha^t}, {\beta^t}, \{b_t\})$
7: $x_t \leftarrow \text{Playarm}(k_t)$
8: $b_t \leftarrow \text{UpdateBelief}({\alpha^t}_{k_t}, {\beta^t}_{k_t}, \{b_t\}, x_t)$
9: $\alpha^{t+1}_{k_t}, \beta^{t+1}_{k_t} \leftarrow \text{UpdateArm}({\alpha^t}_{k_t}, {\beta^t}_{k_t}, \{b_t\}, x_t)$
10: $b_{t+1} \leftarrow \text{NextBelief}(\{b_t\}, P)$
11: if $b_{t+1}(s_1) < b_{t+1}(s_2)$ then
12: $b_{t+1}(s_1) \leftarrow 1$ and $b_{t+1}(s_2) \leftarrow 0$
13: $\alpha^{t+1}_{k_t}(s_1) \leftarrow \alpha^{t+1}_{k_t}(s_2), \beta^{t+1}_{k_t}(s_1) \leftarrow \beta^{t+1}_{k_t}(s_2)$
\{Move knowledge\}
14: $\alpha^{t+1}_{k_t}(s_2) \leftarrow 1, \beta^{t+1}_{k_t}(s_2) \leftarrow 1$ \{Re-initialize $s_2$\}
15: end if
16: end for
Numerical Studies
**Regret** $R_t$

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

$$R(T) = \sum_{t=1}^{T} \mu^*_t - E \left( \sum_{t=1}^{T} x_{k_t} \right)$$

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

**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

\[
\begin{array}{|c|ccc|}
\hline
 & a_1 & a_2 & a_3 \\
\hline
s_1 & 0.1 & 0.9 & 0.3 \\
\hline
s_2 & 0.8 & 0.2 & 0.4 \\
\hline
s_3 & 0.2 & 0.1 & 0.9 \\
\hline
\end{array}
\]

- 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.
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 \((\gamma = 1/5000)\).
Yahoo! Dataset

Yahoo Data - Simulated Average Reward for 6 arms in 5 days, starting from 1th day

Legend:
- article 575000
- article 574957
- article 575934
- article 569479
- article 569364
- article 568470
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

Cumulative Expected Regret vs Round Index
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


