Online Sign Identification: Minimization of the Number of Errors in Thresholding Bandits

Reda Ouhamma[1], Rémy Degenne[1], Pierre Gaillard[2], Vianney Perchet[3]

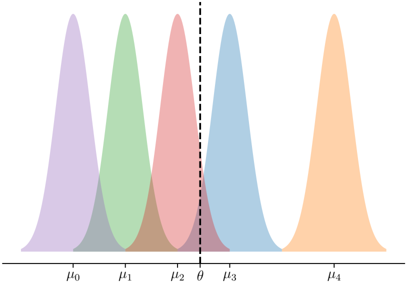
[1]Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[2]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France
[3]Crest, Ensae Criteo AI Lab

Agenda

- Thresholding bandits: Introduction
- Benchmarks
  - Non-adaptive oracle
  - Lower bound, result on "omni"-pulls
- Index-based algorithms for thresholding bandits
  - Generic algorithm design
  - Our algorithm: `FWT`
  - Recovering existing algorithms (`APT`, `LSA`)
- Loss upper bound
  - General analysis for a broad class of algorithms
  - Loss upper bound for `FWT`, and improvement for `APT` and `LSA`
  - Expandability: Sum-of-gaps as an example
- Benefits of adaptivity
- Discussion and future work

Thresholding K-armed bandits

- Arm k: reward distribution ν_k, mean μ_k
- Goal: predict $s_k = \text{sign}(\mu_k) \in \{-1, 1\}$.



Objectives: Given weights $(a_k)_{1 \leq k \leq K}$ and budget of T samples, minimize

$$L_T := \sum_{k=1}^{K} a_k \mathbb{I}\{\hat{s}_k \neq s_k\}$$

Benchmarks: Non-adaptive oracle

1/2

Assume known gaps ($\Delta_k = |\mu_k|/\sigma\sqrt{2}$) & fixed pull number $N_{k,T}$ of arm k.

$$\mathbb{E}\left[L_T\right] = \sum_{k=1}^{K} a_k \mathbb{P}\left(\text{sign}(\hat{\mu}_{k,T}) \neq \text{sign}(\mu_k)\right) \leq \sum_{k=1}^{K} a_k e^{-N_{k,T}\Delta_k^2}$$

Oracle: minimizes above upper-bound.

- Wlog $a_1\Delta_1^2 \leq \ldots \leq a_K\Delta_K^2$
- Oracle's strategy (for some k_0)

$$N_{k,T} = \begin{cases} \frac{c+\log\left(a_k\Delta_k^2\right)}{\Delta_k^2} & \text{if } k \geq k_0 \\ 0 & \text{otherwise} \end{cases}$$

Benchmarks: Non-adaptive oracle
2/2

Oracle strategy: $\exists k_0 \in [K]$

$$N_{k,T} = \begin{cases} \left(c_{k_0} + \log\left(a_k \Delta_k^2\right)\right) / \Delta_k^2 & \text{if } k \geq k_0 \\ 0 & \text{otherwise} \end{cases}$$

Expected loss:

$$\mathbb{E}\left[L_T\right] \leq \sum_{k<k_0} a_k + \sum_{k \geq k_0} a_k \exp\left(-\frac{T + \sum_{j \in S} \frac{1}{\Delta_j^2} \log\left(\frac{a_k \Delta_k^2}{a_j \Delta_j^2}\right)}{\sum_{j \in S} \frac{1}{\Delta_j^2}}\right)$$

Benchmarks: lower bound

Good algorithm pull all arms

Lower bound 1 (adaptation) Fix $\{\Delta_k, k \in [K]\}$ and $T \geq K$.

For any algorithm, there exists $\mu_k \in \{\Delta_k, -\Delta_k\}$ such that

$$\mathbb{E}[L_T] \geq \frac{1}{4} \min_{\sum_k N_k = T} \sum_{k=1}^{K} a_k e^{-4N_k\Delta_k^2}$$

Benchmarks: lower bound

Good algorithm pull all arms

Lower bound 1 (adaptation) Fix $\{\Delta_k, k \in [K]\}$ and $T \geq K$.

For any algorithm, there exists $\mu_k \in \{\Delta_k, -\Delta_k\}$ such that

$$\mathbb{E}\left[L_T\right] \geq \frac{1}{4} \min_{\sum_k N_k = T} \sum_{k=1}^{K} a_k e^{-4N_k\Delta_k^2}$$

Lower bound 2 (contribution)

There exists 4 mean vectors $\mu_1, \mu_2, \mu_{1,\epsilon}, \mu_{2,\epsilon}$
If $\max_{\tilde{\mu} \in \mu_1, \mu_2} \mathbb{E}_{\tilde{\mu}}\left[L_T\right] \leq c_1 \min_{\sum_k N_k = T} \sum_k e^{-c_0 N_k \Delta_k^2}$ then

$$\max_{\mu \in \{\mu_{1,\epsilon}, \mu_{2,\epsilon}\}} \mathbb{E}_\mu\left[\sum_{k=1}^{K_0} N_{k,T}\right] = \Omega(T)$$

Index-based algorithms for thresholding bandits

General structure

Index-based algorithms for thresholding bandits

General structure

Index-based algo: at $t+1$ pulls $i_{t+1} \in \arg\min_{k\in[K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$.

Index-based algorithms for thresholding bandits

General structure

Index-based algo: at $t+1$ pulls $i_{t+1} \in \arg\min_{k\in[K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$.

Algorithm Index-based algorithm for thresholding bandit

Input parameters: an index function $F : \mathbb{N} \times \mathbb{R}_+ \times \mathbb{R}_+^* \to \mathbb{R}$; $a_1, \dots, a_K \in \mathbb{R}_+^*$; $\sigma > 0$

for all $t \in [T]$ do

    for all $k \in [K]$ define

        $N_{k,t-1} = \sum_{s=1}^{t-1} \mathbb{I}\{k = i_s\}, \hat{\mu}_{k,t-1} = \frac{1}{N_{k,t-1}} \sum_{s=1}^{t-1} \mathbb{I}\{k = i_s\} X_s$, and $\hat{\Delta}^2_{k,t-1} = \frac{1}{2\sigma^2}\hat{\mu}^2_{k,t-1}$

    end for

    pull $i_t \in \operatorname{argmin}_{k\in[K]} F\left(N_{k,t-1}, N_{k,t-1}\hat{\Delta}^2_{k,t-1}; a_k\right)$.

    observe $X_t \sim \nu_{i_t}$

end for

Define $t_{\max} = \max_{t\in[T]} \min_{k\in[K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$

Return for each $k \in [K]$ the sign $\hat{s}_k = \text{sign}(\hat{\mu}_{k,t_{\max}})$

Frank-Wolfe for thresholding bandits

Intuition for FWT

Class of algorithms:

- Index-based: $i_{t+1} \in \arg\min_{k \in [K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$.
- $F(n, x; a)$ non-decreasing in n, x and $\lim_{n \to +\infty} F(n, ny; a) = +\infty \; \forall y, a > 0$.

Frank-Wolfe for thresholding bandits

Intuition for FWT

Class of algorithms:

- Index-based: $i_{t+1} \in \arg\min_{k\in[K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$.
- $F(n, x; a)$ non-decreasing in n, x and $\lim\limits_{n\to+\infty} F(n, ny; a) = +\infty \ \forall y, a > 0$.

Intuition behind FWT:

1. Recall the loss upper bound: $B\left(N_T\right) = \sum_{k=1}^{K} a_k e^{-N_{k,T}\Delta_k^2}$
2. Estimate its gradient sequentially: $\nabla B\left(N_t\right) = \left(-a_k\Delta_k^2 e^{-N_{k,t}\Delta_k^2}\right)_k$
3. Gaps must be estimated $\implies$ $\hat{\nabla} B\left(N_t\right)_k = -a_k\hat{\Delta}_k^2 e^{-N_{k,t}\hat{\Delta}^2_{k,t}}$
4. Frank-Wolfe recommends $F_0\left(n, x; a_k\right) = x - \log x + \log(n/a_k)$
5. F_0 is decreasing in x for $x \in (0, 1)$ so we propose the modification:

$$\boxed{F\left(n, x; a_k\right) = \max\{x, 1\} - \log(\max\{x, 1\}) + \log\left(n/a_k\right)}$$

Index-based algorithms for thresholding bandits

Existing algorithms

Class of algorithms:

- Index-based: $i_{t+1} \in \arg\min_{k \in [K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$.
- $F(n, x; a)$ non decreasing in n, x; $\lim_{n \to +\infty} F(n, ny; a) = +\infty \quad \forall y, a > 0$.

Existing algorithms:

Index-based algorithms for thresholding bandits

Existing algorithms

Class of algorithms:

- Index-based: $i_{t+1} \in \arg\min_{k\in[K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$.
- $F(n,x;a)$ non decreasing in n, x; $\lim_{n\to+\infty} F(n, ny; a) = +\infty \quad \forall y, a > 0$.

Existing algorithms:

- APT of [Locatelli et al., 2016]: pulls $i_{t+1} = \mathrm{argmin}_{k\in[K]} N_{k,t}\hat{\Delta}^2_k$.
  - Special case of Algorithm.1 $\quad F(n,x) = x$.
  - Intuition similar to `FWT` with the upper-bound:

$$\mathbb{E}[L_T] = \mathbb{E}\left[\sum_{k=1}^{K} a_k \mathbb{I}\{\hat{s}_k \neq s_k\}\right] \leq B(N_t) = \max_{k\in[K]} e^{-N_{k,t}\Delta_k^2}.$$

Index-based algorithms for thresholding bandits

Existing algorithms

Class of algorithms:

- Index-based: $i_{t+1} \in \arg\min_{k \in [K]} F\left(N_{k,t}, N_{k,t}\hat{\Delta}^2_{k,t}; a_k\right)$.
- $F(n,x;a)$ non decreasing in n, x; $\lim_{n \to +\infty} F(n, ny; a) = +\infty \quad \forall y, a > 0$.

Existing algorithms:

- APT of [Locatelli et al., 2016]: pulls $i_{t+1} = \text{argmin}_{k \in [K]} N_{k,t}\hat{\Delta}^2_k$.
  - Special case of Algorithm.1 $\quad F(n,x) = x$.
  - Intuition similar to FWT with the upper-bound:

$$\mathbb{E}[L_T] = \mathbb{E}\left[\sum_{k=1}^{K} a_k \mathbb{I}\{\hat{s}_k \neq s_k\}\right] \leq B(N_t) = \max_{k \in [K]} e^{-N_{k,t}\Delta_k^2}.$$

- LSA of [Tao et al., 2019]: pulls $i_{t+1} = \text{argmin}_{k \in [K]} \alpha N_{k,t}\hat{\Delta}^2_{k,t} + \log N_{k,t}$.
  - Corresponds to $F(n,x) = x + \log(n)$.
  - Intuition like FWT, they solve instability by estimating $\hat{\Delta}_i^{-1} \sim \sqrt{N_{i,t}}$.

Loss upper bound

Existing results: bounds for existing algorithms

Theorem 2 of [Locatelli et al., 2016]: Let $T \geq 2K$. APT's expected loss is upper-bounded as

$$\mathbb{E}[L_T] \leq \exp\left(-\frac{1}{32}\frac{T}{\sum_i 1/\Delta_i^2} + 2\log((\log(T)+1)K)\right)$$

LSA, adaptation of Theorem 1 of [Tao et al., 2019]: Let $\alpha = 1/20$. LSA's expected loss is upper-bounded as

$$\mathbb{E}[L_T] \leq \min_{N_1+\cdots+N_K=T} \sum_{i=1}^{K} \exp\left(-N_i\Delta_i^2/16020\right)$$

Loss upper bound

(new) General result: Index-based algorithms

Theorem. Let $F : \mathbb{N} \times \mathbb{R} \times \mathbb{R}_+^* \to \mathbb{R}$, $C_1, \ldots, C_K > \max_k F(0, 0; a_k)$. For all $j, k \in [K]$, define

- $t_j(C_k)$, solution of $F(t, t\Delta_j^2; a_j) = C_k$,
- $S_k \subseteq [K]$, $t_{j,0}(C_k) \in \mathbb{R}_+$ such that for $i \notin S_k$,

$$\mathbb{P}\left(\exists n \leq t_{i,0}(C_k), F(n, n\hat{\Delta}_{n,i}^2; a_i) \geq C_k\right) = 1.$$

Loss upper bound

(new) General result: Index-based algorithms

Theorem. Let $F : \mathbb{N} \times \mathbb{R} \times \mathbb{R}_+^* \to \mathbb{R}$, $C_1, \ldots, C_K > \max_k F(0, 0; a_k)$. For all $j, k \in [K]$, define

- $t_j(C_k)$, solution of $F(t, t\Delta_j^2; a_j) = C_k$,
- $S_k \subseteq [K]$, $t_{j,0}(C_k) \in \mathbb{R}_+$ such that for $i \notin S_k$,

$$\mathbb{P}\left(\exists n \leq t_{i,0}(C_k), F(n, n\hat{\Delta}_{n,i}^2; a_i) \geq C_k\right) = 1.$$

The expected loss is upper-bounded as

$$\mathbb{E}[L_T^{\mathbb{A}}] \leq e \sum_{k=1}^{K} a_k \exp\left(-\frac{\frac{1}{2}\left(T - \sum_{j \notin S_k} t_{j,0}(C_k)\right) - \sum_{j \in S_k} t_j(C_k)}{\sum_{j \in S_k} 1/\Delta_j^2}\right)$$

$$+ T \sum_{k=1}^{K} a_k e^{-t_k(C_k)\Delta_k^2}.$$

Loss upper bound

Proof sketch. Two parts:

1. For any arm $j \in [K]$, w.h.p, there is a time $\tau_j(C_k)$ s.t:
   - $F(\tau_j(C_k), \tau_j(C_k)\hat{\Delta}_{\tau_j(C_k),j}; a_j) \geq C_k$. We prove: $\forall j, k \in [K], \tau_j(C_k)$ has an exponential tail
   - the algorithm pulls the minimal index to control the probability that the minimum never reaches C_k.
2. If index of arm k is large, the probability of mistake on k is small.

Intuition behind the times $t_j(C_k)$

- The smallest # of samples s.t $t_j(C_k) \geq \tau_j(C_k)$ w.h.p.
- Determining $t_j(C_k) \implies$ explicit bounds if algorithm in our class.

Loss upper bound

(new) Specific results

Corollary: Assume that $a_k = 1$, it comes:

$$\mathbb{E}[L_T^{\text{APT}}] \leq 2\sqrt{eT} \sum_{k=1}^{K} a_k \exp\left(-\frac{1}{4} \frac{T}{\sum_{k=1}^{K} 1/\Delta_k^2}\right),$$

Loss upper bound

(new) Specific results

Corollary: Assume that $a_k = 1$, it comes:

$$\mathbb{E}[L_T^{\text{APT}}] \leq 2\sqrt{eT} \sum_{k=1}^{K} a_k \exp\left(-\frac{1}{4} \frac{T}{\sum_{k=1}^{K} 1/\Delta_k^2}\right),$$

for $T \geq 2 \sum_{j=1}^{K} \frac{1}{\Delta_j^2}(2 + \log \frac{a_j \Delta_j^2 \max_i a_i \Delta_i^2}{(\min_k a_k \Delta_k^2)^2} - \log \frac{T}{e^3})$, it comes:

$$\mathbb{E}[L_T^{\texttt{FWT}}] \leq 2\sqrt{eT} \sum_{k} a_k \exp\left(-\frac{1}{2} \frac{T/2 - \sum_j \frac{1}{\Delta_j^2} \log \frac{a_j \Delta_j^2}{a_k \Delta_k^2}}{\sum_j 1/\Delta_j^2}\right)$$

Loss upper bound

(new) Specific results

Corollary: Assume that $a_k = 1$, it comes:

$$\mathbb{E}[L_T^{\text{APT}}] \leq 2\sqrt{eT} \sum_{k=1}^{K} a_k \exp\left(-\frac{1}{4}\frac{T}{\sum_{k=1}^{K} 1/\Delta_k^2}\right),$$

for $T \geq 2\sum_{j=1}^{K} \frac{1}{\Delta_j^2}(2 + \log \frac{a_j \Delta_j^2 \max_i a_i \Delta_i^2}{(\min_k a_k \Delta_k^2)^2} - \log \frac{T}{e^3})$, it comes:

$$\mathbb{E}[L_T^{\texttt{FWT}}] \leq 2\sqrt{eT} \sum_{k} a_k \exp\left(-\frac{1}{2}\frac{T/2 - \sum_j \frac{1}{\Delta_j^2} \log \frac{a_j \Delta_j^2}{a_k \Delta_k^2}}{\sum_j 1/\Delta_j^2}\right)$$

Remarks:

- LSA's bound is less explicit, close to FWT's for large T.
- LSA and FWT recover the same exponent as the oracle (up to factor $1/4$).

Loss upper bound

Empirical comparison

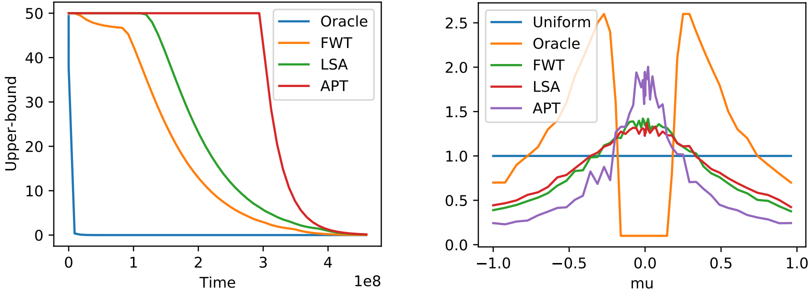


Figure: Gaps $\Delta_i = (i/K)^2$. [Left] comparison of the loss upper bounds. [right] oracle and empirical sampling distributions with respect to μ,

Generalization

Sum-of-gaps objective

The sum-of-gaps: $L_T = \sum_{k=1}^{K} \Delta_k \mathbb{I}\{\text{error on } k\}$.

Generalization

Sum-of-gaps objective

The sum-of-gaps: $L_T = \sum_{k=1}^{K} \Delta_k \mathbb{I}\{\text{error on } k\}$.

The algorithm: obtained following the steps for `FWT`:

$$F(n,x) = x' - \frac{3}{2}\log(x') + \frac{3}{2}\log(n), \text{ where } x' = \max\left(x, \frac{3}{2}\right).$$

Generalization

Sum-of-gaps objective

The sum-of-gaps: $L_T = \sum_{k=1}^{K} \Delta_k \mathbb{I}\{\text{error on } k\}$.

The algorithm: obtained following the steps for `FWT`:

$$F(n, x) = x' - \frac{3}{2}\log(x') + \frac{3}{2}\log(n), \text{ where } x' = \max\left(x, \frac{3}{2}\right).$$

Loss bound: for $T \geq 2\sum_{j=1}^{k} \frac{1}{\Delta_j^2}\left(3 + 3\log\frac{\Delta_j \max_i \Delta_i}{(\min_i \Delta_i)^2} - \log\frac{T}{e}\right)$, we show

$$\mathbb{E}\left[\sum_{k=1}^{K} \Delta_k E_k\right] \leq 2\sqrt{eT}\sum_k \Delta_k \exp\left(-\frac{1}{2}\frac{\frac{T}{2} + \sum_j \frac{3}{2}\frac{1}{\Delta_j^2}\log\frac{\Delta_k^2}{\Delta_j^2}}{\sum_j 1/\Delta_j^2}\right).$$

Beating the oracle

Toy experiment: arm k supported on $\{0, x_k\}$ s.t $x_k \in \mathbb{R}$; $\mathbb{P}(X_k = 0) = 1/2$, any non-adaptive oracle yields:

$$\mathbb{E}[L_T] = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{2^{N_k,T}} \geq \frac{K}{2^{(T/K)+1}}$$

Beating the oracle

Toy experiment: arm k supported on $\{0, x_k\}$ s.t $x_k \in \mathbb{R}$; $\mathbb{P}(X_k = 0) = 1/2$, any non-adaptive oracle yields:

$$\mathbb{E}\left[L_T\right] = \frac{1}{2}\sum_{k=1}^{K}\frac{1}{2^{N_k,T}} \geq \frac{K}{2^{(T/K)+1}}$$

Consider the algorithm: stop sampling arm k once a sample $X_k(t) \neq 0$, then

$$\mathbb{E}[L_T] \leq K\,\mathbb{P}(Z > T) \leq \frac{K}{2^{T/2}}\left(1 + \frac{1}{\sqrt{2}}\right)^K,$$

where Z: # of samples to classify all arms correctly, $Z \sim \mathrm{NB}(K, 1/2)$

Beating the oracle

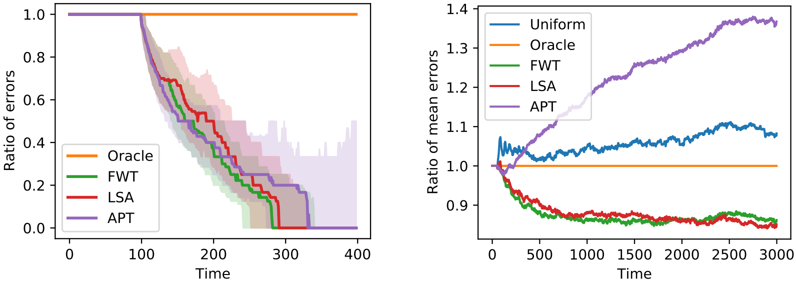


Figure: [left] Median (and 1st / 3rd quartiles) of the ratio: error suffered by algorithm over error of the non-adaptive oracle $(\mu_k)_k = \left((-1)^k\right)_{k=1,\dots,100}$. [right] Ratio of the averaged errors (500 runs) of each algorithm with that of the oracle $(\mu_k)_k = \left((-1)^k (k/K)^2\right)_{k=1,\dots,50}$.

Conclusion and perspectives

This paper:

◦ Proposes a generic method to design algorithms, with a generic proof, with demonstrated performance improvement on the weighted number of errors loss.

◦ For thresholding bandits:

1. We propose FWT that achieves explicit finite time loss bounds
2. We use our proof to improve the original bound of LSA by a factor of 4005 and APT by 8.
3. Our method, FWT, is within a factor 4 of the oracle.

◦ Shows the benefits of adaptivity, our algorithms surpass the optimal non-adaptive oracle empirically in certain settings.

◦ Could be complemented by deeper theoretical analyses of adaptivity.

◦ Could extend to general losses.

Thank you!

Questions?

Locatelli, A., Gutzeit, M., and Carpentier, A. (2016). An optimal algorithm for the thresholding bandit problem.
In International Conference on Machine Learning, pages 1690–1698. PMLR.

Tao, C., Blanco, S., Peng, J., and Zhou, Y. (2019).
Thresholding bandit with optimal aggregate regret.
arXiv preprint arXiv:1905.11046.