Efficient Pure Exploration in Adaptive Round model

Tianyuan Jin*,Jieming Shi+, Xiaokui Xiao+, Enhong Chen*

*School of Computer Science and Technology, University of Science and Technology of China, jty123@mail.ustc.edu.cn, cheneh@ustc.edu.cn
+School of Computing, National University of Singapore, {shijm, xkxiao}@nus.edu.sg

Introduction

Motivation

- Medical Trials, Crowdsourcing, Online advertisement
- Parallel Computing
  - Medical Trials
    - Multiple clinical subjects (e.g., mice)
    - Typically a waiting time (e.g., days) before the effects of drugs become observable to guide the design
    - It is important to minimize not only the total number of tests on clinical subjects (i.e., query complexity) but also the number of rounds, to identify the best drug within the shortest time frame

Problem Definition

Problem Setting

Under the standard setting of stochastic multi-armed bandit selection, there is a set S of n arms, such that each arm i is associated with an unknown reward distribution $\mathcal{D}_i$ supported on $[0,1]$ with unknown mean θ_i. Let i^* be the arm with i^{th} largest mean. We aim to identify the k arms with the largest means by pulling (i.e., sampling from) the arms in rounds. In each round, we can pull any number of arms for any number of times, such that (i) each pull of an arm i returns a reward that is an i.i.d. sample from $\mathcal{D}_i$, and (ii) the reward is only revealed at the end of the round.

PAC-top-*k* and RL-top-*k*

For *PAC subset selection*, we study two problems: (i) Problem 1: PAC Top-k Arm Selection with Adaptive Rounds(PAC-top-k), and (ii) Problem 2: Top-k Arm with a Round Limit R(RL-top-k). In both problems, the goal is to identify a set $V \subseteq S$ of k arms, such that for all $i \in [1,k]$, the i^{th} largest arm in V has mean larger than $\theta_{i^*} - \epsilon$ with probability at least $1-\delta$, where ϵ and δ are given constants. Specifically, for Problem 1, we aim to minimize the number of rounds performed, while achieving the best possible query complexity; for Problem 2, we expose a upper limit on the number of rounds that can perform, R, and aim to minimize the query complexity within R rounds.

Exact-top-*k*

For *exact top-k arm identification*, denoted as Problem 3(exact-top-k), we aim to minimize the number of rounds required as well as the query cost, for identifying the top-k arms with the largest means. We assume $\theta_{k^*} > \theta_{(k+1)^*}$, in order to ensure the uniqueness of the solution.

Our results

	Algorithm	Number of Rounds	Query Complexity
$k=1$	[5]	$\Theta(\log n)$	$O\left(\frac{n}{\epsilon^2}\cdot\log\frac{1}{\delta}\right)$
All $k \in [n]$	[6, 16, 14]	$\Theta(\log\frac{n}{k})$	$O\left(\frac{n}{\epsilon^2}\cdot\log\frac{k}{\delta}\right)$
	This paper (Algorithm 1)	$2\log^*_{\frac{k}{\delta}}(n)$	$O\left(\frac{n}{\epsilon^2}\cdot\log\frac{k}{\delta}\right)$

Table 1: Summary of algorithms for Problem 1: Top-k arms with adaptive rounds.

	Algorithm	Bound	Query Complexity
All $k \in [n]$	[4], assuming Δ_k is known	exact top-k	$O\left(\frac{n}{\Delta_k^2}\cdot(\log\frac{k}{\delta}+\text{ilog}^{(R)}(n))\right)$
	This paper (Algorithm 2)	(ϵ,δ)	$O\left(\frac{n}{\epsilon^2}\cdot(\log\frac{k}{\delta}+\text{ilog}^{(R)}_{\frac{k}{\delta}}(n))\right)$

Table 2: Summary of algorithms for Problem 2: Top-k arms with a round limit R

	Algorithm	Round Complexity	Query Complexity
$k=1$	[8]	$O(\log n\cdot\log\Delta_k^{-1})$	$O\left(\sum_{i=1}^n\Delta_i^{-2}\cdot\log\frac{\log\Delta_i^{-1}}{\delta}\right)$
	[17]	$O\left(\sum_{i=1}^n\Delta_i^{-2}\cdot\log\frac{\log\Delta_i^{-1}}{\delta}\right)$	$O\left(\sum_{i=1}^n\Delta_i^{-2}\cdot\log\frac{\log\Delta_i^{-1}}{\delta}\right)$
All $k \in [n]$	[10]	$O\left(\sum_{i=1}^n\Delta_i^{-2}\cdot\log\frac{\sum_{i=1}^n\Delta_i^{-1}}{\delta}\right)$	$O\left(\sum_{i=1}^n\Delta_i^{-2}\cdot\log\frac{\sum_{i=1}^n\Delta_i^{-1}}{\delta}\right)$
	[13]	$O(\log n\cdot\log\Delta_k^{-1})$	$O\left(\sum_{i=1}^n\Delta_i^{-2}\cdot\log\frac{k\cdot\log\Delta_i^{-1}}{\delta}\right)$
	This paper	$O(\log^*_{\frac{1}{\delta}} n\cdot\log\Delta_k^{-1})$	$O\left(\sum_{i=1}^n\Delta_i^{-2}\cdot\log\frac{k\cdot\log\Delta_i^{-1}}{\delta}\right)$

Table 3: Summary of algorithms for Problem 3: Exact top-k arm identification. (For $i \le k$, Δ_i denotes the difference between the means of the i^{th} and $(k+1)^{th}$ arms. For $i > k$, Δ_i denotes the difference between the means of the k^{th} and i^{th} arms.)

Algorithms

PAC-top-k

- Optimal query complexity and round complexity
- Two New Technique
  - **Double Test**
    - Keep the estimation of indicator unbiased
    - Each round, eliminate more arms
  - **Keep a global common list**
    - Each round, allow ϵ regret
    - Small constant in query complexity

Algorithm 1 Top-k δ-Elimination (k-δE)

1: **Input:** S, Q, k, ε and δ.
2: Initialize $r \leftarrow 1$, $\beta_1 \leftarrow 1$, $\delta_1 \leftarrow \delta/4$, $S_1 \leftarrow S$.
3: Initialize $S' \leftarrow \varnothing$.
4: **while** $S_r \neq \varnothing$ **do**
5: Sample each arm $i \in S_r$ for $Q_r \leftarrow \beta_r \cdot Q \cdot \log(\frac{k}{\delta_r})$ times; sort them decreasingly by empirical means $\hat{\theta}_i$;
6: $k' \leftarrow \min\{k, |S_r|\}$;
7: Uniformly sample k' arms from the top-$[\lceil(\delta_r/k)^{\beta_r}\cdot|S_r|/2\rceil + k' - 1]$ sorted arms as set $S^r_{k'}$;
8: For each arm $i \in S^r_{k'}$, double test by re-sampling it Q_r times and insert its new empirical mean into S';
9: Get the k-th largest mean in S' as $S'(k)$;
10: Set $S_{r+1} \leftarrow \{i \in S_r : \hat{\theta}_i \ge S'(k) + 3\epsilon/4\}$ and $S_{r+1} \leftarrow S_{r+1}\backslash S^r_{k'}$;
11: **if** $|S_{r+1}| \le \frac{2\delta}{k}|S_r|$ **then**
12: $\beta_{r+1} \leftarrow \beta_r\frac{|S_r|}{2|S_{r+1}|}$;
13: **else**
14: $\beta_{r+1} \leftarrow \beta_r\frac{|S_r|}{|S_{r+1}|}$;
15: **end if**
16: $\delta_{r+1} \leftarrow \delta/(2\cdot 2^r)$;
17: $r \leftarrow r+1$;
18: **end while**
19: **Return:** Top-k arms in S'.

Exact top-*k*

- Using Exponential-Gap-Elimination algorithm as framework
- Using Algorithm~1 as subroutine.
- Achieve near optimal query complexity, improve the round complexity

RL-top-k

- Delay all double tests until the final round(algorithm 3)
- Given round limit R, achieve near-optimal query complexity

Algorithm 2 Top-k δ-Elimination with Limited Rounds (k-δER)

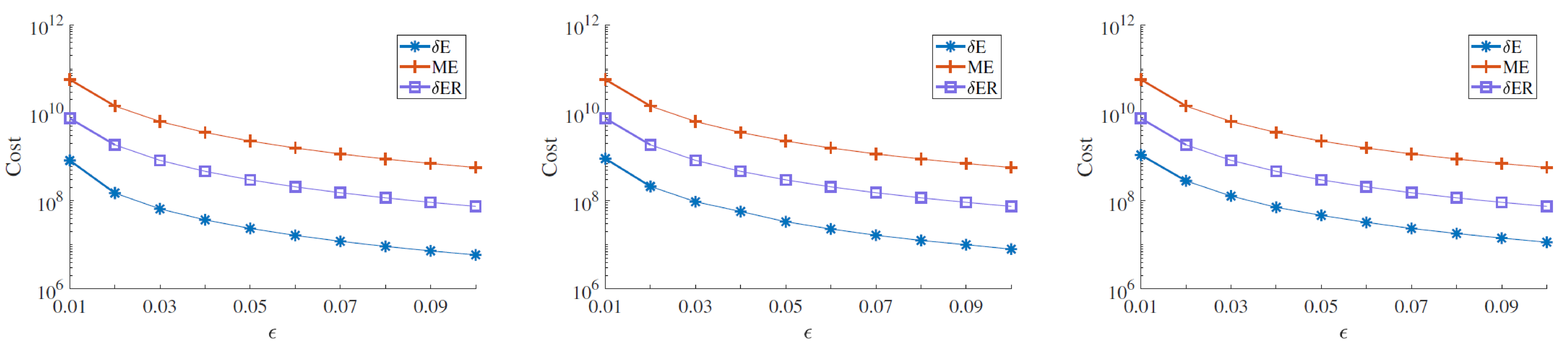
1: **Input:** S, R, k,Q, ϵ and δ.
2: Initialize $r \leftarrow 1$, $\delta_1 \leftarrow \delta/4$, $\beta_1 \leftarrow 1 + \text{i}\log^{(R)}_{\frac{k}{\delta}}(n)$, $S_1 \leftarrow S$, $S' \leftarrow \varnothing$.
3: **for** $r \le R-1$ **do**
4: Sample each arm in S_r by $Q_r \leftarrow \beta_r \cdot Q \cdot \log(k/\delta_r)$ times, and sort decreasingly by their empirical $\hat{\theta}_i$;
5: $k' \leftarrow \min\{k, |S_r|\}$.
6: Uniformly sample k' arms from the top-$[\lceil(\delta_r/k)^{\beta_r}\cdot|S_r|/2\rceil + k' - 1]$ sorted arms as set $S^r_{k'}$;
7: Add $S^r_{k'}$ into S';
8: Let S_{r+1} be set containing all the top-$[\lceil 2\cdot(\delta_r/k)^{\beta_r}|S_r|\rceil + k' - 1]$ sorted arms in S_r;
9: $S_{r+1} \leftarrow S_{r+1}\backslash S^r_{k'}$;
10: $\beta_{r+1} \leftarrow \beta_r\frac{|S_r|}{2|S_{r+1}|}$;
11: $\delta_{r+1} \leftarrow \delta/(2\cdot 2^r)$;
12: $r \leftarrow r+1$;
13: **end for**
14: **Return:** $\text{US}(S', S_R, Q, \beta_R, \delta, k)$.

Algorithm 3 Uniformly Sampling (US)

1: **Input:** $S', S_R, Q, \beta_R, \delta, k$.
2: Sample each arm $i \in S_R$ by $Q\cdot\beta_R\cdot\log\frac{2k\cdot 2^R}{\delta}$ times and sort decreasingly by their empirical $\hat{\theta}_i$;
3: Let S^R_k be the set of all the top-$\min\{k, |S_R|\}$ arms;
4: Sample each arm $i \in S'$ by $Q\log\frac{4|S'|}{\delta}$ times, and let $\hat{\theta}_i$ be its empirical mean;
5: **Return:** Top-k arms in $S^R_k \cup S'$.

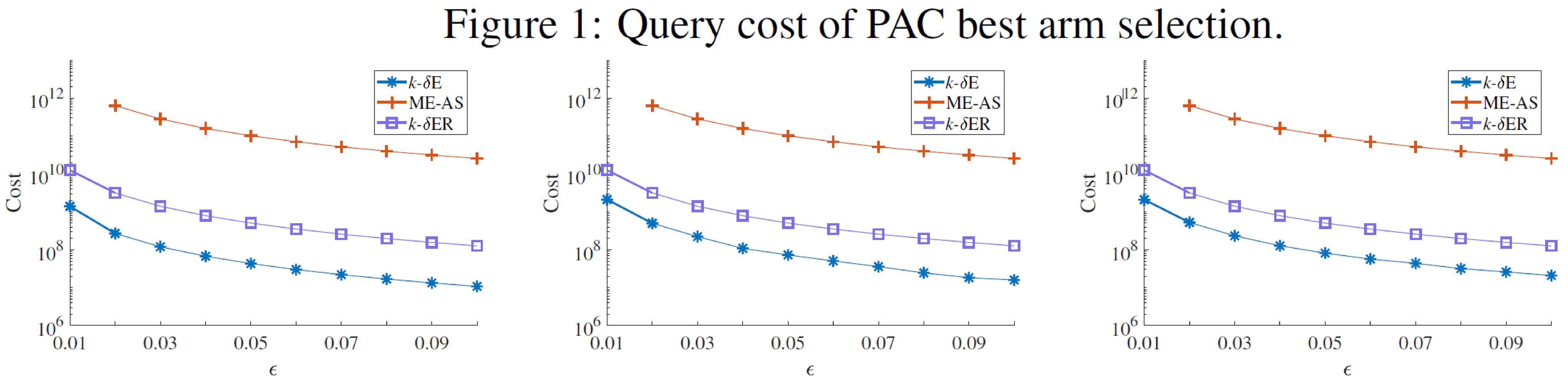
Experiments

- Settings: default values k=20, δ=0.1
- Dataset:
  - Uniform: $\theta_i \sim \text{Unif}[0,1]$. The mean of arms, θ_i, are uniformly distributed in $[0,1]$.
  - Normal: $\theta_i \sim TN(0.5, 0.2)$. Each θ_i is generated from a truncated normal distribution with mean 0.5, the standard deviation 0.2 and the support $[0,1]$.
  - Segment: $\theta_i = 0.5$ for $i = 1,\cdots,k$ and $\theta_i = 0.4$ for $i = k+1,\cdots,n$.



(a) Uniform Dataset. (b) Normal Dataset. (c) Segment Dataset.

Figure 1: Query cost of PAC best arm selection.



(a) Uniform Dataset. (b) Normal Dataset. (c) Segment Dataset.

Figure 2: Query cost of PAC top-k arm selection.

	Algorithm	Uniform	Normal	Segment
$k=1$	ME	11	11	11
	δE	2.2	3.4	3.9
	δER	2	2	2
$k=20$	ME-AS	6	6	6
	k-δE	2.1	3.0	3.8
	k-δER	2	2	2

Table 4: Number of rounds performed.

Dataset	Algorithm	Rounds	Query Cost
Normal	EG-δE	21	1.4×10^8
	[8]	36	6.7×10^9
	[17]	0.9×10^8	0.9×10^8
Uniform	EG-δE	27	2.8×10^9
	[8]	59	1.2×10^{11}
	[17]	2.4×10^9	2.4×10^9
Segment	EG-δE	6	5.6×10^7
	[8]	24	1.3×10^{10}
	[17]	2.2×10^8	2.2×10^8

Table 5: Exact top-k arms: rounds and query cost.