Optimal Sequential Maximization
One Interview is Enough!

Moein Falahatgar [1] **Alon Orlitsky** [2] **Venkatadheeraj Pichapati** [1]

Abstract

Maximum selection under probabilistic queries *(probabilistic maximization)* is a fundamental algorithmic problem arising in numerous theoretical and practical contexts. We derive the first query-optimal sequential algorithm for probabilistic-maximization. Departing from previous assumptions, the algorithm and performance guarantees apply even for infinitely many items, hence in particular do not require a-priori knowledge of the number of items. The algorithm has linear query complexity, and is optimal also in the streaming setting.

To derive these results we consider a probabilistic setting where several candidates for a position are asked multiple questions with the goal of finding who has the highest probability of answering interview questions correctly. Previous work minimized the total number of questions asked by alternating back and forth between the best performing candidates, in a sense, inviting them to multiple interviews. We show that the same order-wise selection accuracy can be achieved by querying the candidates sequentially, never returning to a previously queried candidate. Hence one interview is enough!

1. Introduction

Reinforcement learning, one of machine learning's tripodal paradigms, applies a sequence of actions and uses observations of their outcomes to learn the best possible strategy. It typically addresses two general scenarios that differ in the type of observations available to the learner. *Full knowledge*, where following each action, the learner observes the outcomes of all possible actions, such as the returns of all stocks on a given day; and *partial knowledge*, where the learner observes the outcomes of only a subset of the actions. The simplest and by far most popular partial-knowledge observation is that of just the action taken. For example, the effect of the administered medication, the click-rate of the placed ad, the performance of the routing algorithm utilized, or back to investment, the return of the strategy utilized.

The latter paradigm is captured by an idealized framework where a gambler can choose between k slot-machine arms, each with its own unknown return distribution. Through successive arm pulls, the gambler tries to maximize their return or find the most rewarding arm. The framework is commonly called the *multi-armed bandit (MAB)* as "in the long run... slot machines are as effective as human bandits in separating the victim from his money" (Lai & Robbins, 1985).

Two common measures evaluate the gambler's performance, and corresponding strategy. *Regret*, or *exploration-exploitation*, aims to maximize the gambler's expected total return over time (Auer et al., 2002; Bubeck et al., 2012); *Maximization*, or *pure exploration*, seeks the arm with the highest expected return (Bubeck et al., 2009; Karnin et al., 2013; Gabillon et al., 2012); We consider the latter. Maximum selection (maximization) arises in numerous applications ranging from medical trials (Robbins, 1952) to social choice (Caplin & Nalebuff, 1991), to wireless channel band selection (Audibert & Bubeck, 2010).

The typical approach for finding PAC maximum arm with linear query complexity (Even-Dar et al., 2006; Zhou et al., 2014) is to conduct the pulls (queries) in rounds. Starting with all n arms, in round i, all surviving arms are queried certain number of times, and the top half performing arms continue to the next round while the bottom half are discarded. Motivating this strategy is the goal of querying low-expectation arms only few times, while querying high-expectation arms successively more times, till the best is found. This approach inherently alternates between the arms, repeatedly looking for the best subset, and refining the selection in subsequent rounds.

For several applications, there is a cost associated with

[1] Apple Inc. [2] University of California, San Diego. Correspondence to: Venkatadheeraj Pichapati <dheerajpv7@ucsd.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

changing the queried alternatives. For example, switching back and forth between webpage layout styles frequently can annoy users; in manufacturing, switching alternatives might require reconfiguring entire production line.

One of the oldest branches of MAB research, has therefore considered *Bandits with switching costs* (Dekel et al., 2014; Koren et al., 2017). Our problem setup can be viewed in that context. Finding the best alternative clearly requires considering all possible alternatives, and if the switching cost is sufficiently high, this would be achieved by considering each alternative consecutively without ever returning back to the previously considered alternative.

In addition to reinforcement-learning motivation, the problem can be viewed from another perspective. The maximization model has also been likened to as an *interview process*, *e.g.,* (Schumann et al., 2017; David & Shimkin, 2014). An employer considers n applicants for a position, and asks each of them questions that for simplicity we assume have the same expected score, trying to find the one whose expected grade is at most ϵ away from the best.

The traditional approach (Even-Dar et al., 2006; Zhou et al., 2014) assumes the knowledge of n and that the various candidates can be interviewed at will. While typical interviews may not proceed in $\log n$ or $\log^* n$ rounds, many leading employers still conduct at least 2-round interviews exploring other candidates in between the rounds. More interview rounds are also common (fif).

Our results show that the knowledge of n is not necessary and further that a single interview round is order optimal.

We now define our problem more formally.

1.1. Traditional bandits sequential maximization

The company can interview one candidate at a time. We assume that each candidate interview consists of a sequence of queries, with each query providing probabilistic evidence about the candidate's merits. Each candidate c has a parameter $v_c \in [0, 1]$ indicating the probability that the candidate answers each question correctly. To each question asked, candidate c gives a correct response with probability v_c, and distinct questions are answered independently. The confidence in the candidate's evaluation improves as the number of queries increases, yet when each candidate is interviewed, it is not clear how many queries would truly suffice. At each interview, the administrator can ask any number of queries to evaluate the current candidate but once the interview ends, the candidate can't be called for further evaluation. Adopting the conventional PAC formulation, for given $\epsilon < 1/4$ and $\delta < 1/4$, we would like to find w.p. $\geq 1 - \delta$ an *ϵ-maximum*, i.e., one whose value is at most ϵ below that of the maximum value among all candidates. The goal is to minimize the total number of queries.

This is the traditional multi-armed bandits formulation, except that it is adapted for the streaming framework i.e., candidates come in a *uniformly random* sequence and one candidate can be interviewed at a time and once a candidate's interview is completed, they can't be recalled for further evaluation.

Under no constraints, (Mannor & Tsitsiklis, 2004) showed that maximization algorithms require $\Theta\big(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\big)$ queries to find an ϵ-maximum with probability $\geq 1 - \delta$. (Even-Dar et al., 2006; Zhou et al., 2014) provided the matching upper bound. Recall that these algorithms eliminate candidates in multiple rounds. To derive a sequential algorithm, these algorithms need to be modified in several ways. Only a single "round" can be performed, during which all but one item need to be discarded. Furthermore, we need to fix the number of queries of each item without knowing the performance of all subsequent items, let alone the best ones.

Questions In the sequential model, we ask the following questions: a) What is the optimal query complexity? b) Will the answer change if n is not known in advance?

Results In Theorems 10, 11 and 15, we derive optimal n-agnostic streaming maximization algorithm that w.p.$\geq$ $1 - \delta$ uses $\mathcal{O}(\frac{n}{\epsilon^2}\log\frac{1}{\delta})$ queries and outputs an ϵ-maximum. Notice that since query complexity is orderwise same as that of lower bound for traditional multi-armed bandits setting that need not be sequential and has a priori knowledge of n, we answered all questions above, with the same bound. Further it also implies that a candidate once interviewed doesn't need to be called for further evaluation. One interview is enough!

General Models For simplicity, we prove our results when each candidate has value v_i and for each query we observe a $Bernoulli(v_i)$ random variable. Essentially the same results hold even when for each candidate i, a query results in a random variable with an arbitrary distribution with bounded support, and the value of a candidate is the distribution's expected value. Query complexity scales according to bounds on the distribution's variance and domain size. We provide more explanation in Appendix.

In the process of designing optimal sequential maximization algorithm, we develop tools (ASYMMETRIC-THRESHOLD in Section 2.4) and proof techniques that we believe can be adapted to design optimal sequential algorithms even under other setups. To demonstrate this, we consider another variation of traditional multi-armed bandits, *dueling bandits* (Szörényi et al., 2015; Yue et al., 2012).

1.2. Dueling bandits sequential maximization

Here, in each interview the company can compare two candidates. To facilitate these "pairwise comparisons", the company is allowed to keep a "buffer" of one candidate and in each interview, it can compare the buffer candidate with a new candidate assigning them tasks to complete. For every independent task, candidate i will finish the task before candidate j with probability $p_{i,j}$, which is also referred to as the probability that i is *preferred* to j. If $p_{i,j} \geq \frac{1}{2}$, we say that i is *preferable* to j, denoted by $i \geq j$. Let $\tilde{p}_{i,j} = p_{i,j} - 1/2$ be the *centered preference probability*. Candidate i is ϵ-preferable to candidate j if $\tilde{p}_{i,j} \geq -\epsilon$. Our goal here is: given $\epsilon < 1/4$ and $\delta < 1/4$, to w.p.$\geq 1 - \delta$, find an ϵ-*maximum* candidate that is ϵ-preferable to every other candidate. The confidence in the candidates' comparison improves as the number of tasks increases, yet during each interview, it is not clear how many tasks would truly suffice. After each interview the administrator decides whether the newly-compared candidate is eliminated and the "buffer" candidate continues, or the "buffer" candidate is eliminated and replaced by the new candidate. Once a candidate is eliminated, they can't be recalled. The process may stop at any time, and at that point, the "buffer" candidate is declared as ϵ-maximum.

This is the dueling bandits formulation, except that it is adapted for the streaming framework i.e., candidates come in a uniformly random sequence and at a time only one interview can happen and after each interview a candidate is sent away and can never be recalled.

On the outset, this setting might look easier than the regular bandits setting since we can compare the current candidate with the "buffer" candidate, thereby getting more information about a previous ("buffer") candidate. But observe that this model is more general in the sense that it has $\Theta(n^2)$ parameters whereas the traditional bandits setting has only n.

Under this dueling bandits setting, to allow for the feasibility of existence of maximum, one needs to assume certain transitivity property among the elements. We assume one such property which has been used previously (Falahatgar et al., 2017b;a).

The model is said to satisfy *Strong Stochastic Transitivity (SST)* if there is an ordering $\succ$ among elements such that for all $i \succ j$ and $j \succ k$, $\tilde{p}_{i,k} \geq \max(\tilde{p}_{i,j}, \tilde{p}_{j,k})$.

For models with SST, (Falahatgar et al., 2017a) presented a min-max optimal maximization algorithm with comparison complexity of $\mathcal{O}\big(\frac{n}{\epsilon^2} \log \frac{1}{\delta}\big)$. This algorithm is neither streaming based nor n-agnostic. But in the process, for the same model (Falahatgar et al., 2017a) also presented a sub optimal min-max maximization algorithm, SEQ-ELIMINATE with comparison complexity of $\mathcal{O}\big(\frac{n}{\epsilon^2} \log \frac{n}{\delta}\big)$. This algorithm is streaming based but not n-agnostic.

Questions In the sequential scenario, a) what is the optimal comparison complexity under the dueling bandit settings with SST? b) will the answer change if n is not known in advance?

Results In Theorem 18, we derive an optimal n-agnostic streaming maximization algorithm that w.p.$\geq 1 - \delta$ uses $\mathcal{O}(\frac{n}{\epsilon^2} \log \frac{1}{\delta})$ comparisons and outputs an ϵ-maximum. Notice that since comparison complexity is orderwise same as that of lower bound for dueling bandits setting that need not be sequential and has a priori knowledge of n, we answered all questions above, with the same bound.

Outline In Section 2, we derive optimal sequential maximization algorithm under traditional bandits setting. In Section 3, we derive optimal sequential maximization algorithm under dueling bandits setting. In Section 4, we compare empirical performance of maximization algorithms. Finally, we provide our concluding remarks in Section 5.

2. Traditional Multi-armed bandits

2.1. Preliminaries

All sequential algorithms in this section share the same structure. They sequentially interview the candidates and maintain an *anchor* a deemed the *best* candidate interviewed thus far. Upon interviewing candidate c they approximate its value v_c by an estimate $\hat{v}_c$, and compare it to the current anchor's estimate $\hat{v}_a$, deciding whether to keep the current anchor, or replace it by c. They output the final anchor a^* to be the *best*.

The algorithm's *additive error* is $|v_b - v_{a^*}|$ where b is the candidate with highest value. We would like additive error to be $> \epsilon$ with probability $\leq \delta$ that we call *uncertainty*.

For simplicity, we say that candidate c is *better* than c' if $v_c > v_{c'}$, and *worse* if $v_c < v_{c'}$. Similarly we say that candidate c is ϵ-*better* than c' if $v_c > v_{c'} + \epsilon$, and ϵ-*worse* than c' if $v_c < v_{c'} - \epsilon$.

Hoeffding's Inequality (Hoeffding, 1994) states that if $X \sim \text{Binomial}(p, n)$, then

$$\begin{aligned} Pr(X \leq (p - \epsilon)n) &\leq e^{-2\epsilon^2 n} \\ Pr(X \geq (p + \epsilon)n) &\leq e^{-2\epsilon^2 n}. \end{aligned} \tag{1}$$

Hence with $\lceil \frac{1}{2\epsilon^2} \ln \frac{1}{\delta} \rceil$ queries, we can approximate a candidate's value to a one-sided additive accuracy $\leq \epsilon$ with error probability, or *uncertainty*, $\leq \delta$.

2.2. Suboptimal sequential maximization

We first consider a simple sequential maximization algorithm with suboptimal query complexity, and then build on it to derive an optimal one.

Notice that if we approximate all candidates' values to $\leq \epsilon/2$ additive accuracy, then the candidate with the highest approximated value will be an ϵ-maximum candidate.

2.2.1. Algorithm Suboptimal-Sequential

Algorithm Suboptimal-Sequential (S-S) maintains anchor a, a proxy for the candidate with highest approximated score so far. S-S updates a with current candidate if their approximated score is more than that of a. After interviewing the final candidate S-S outputs a.

From Hoeffding's inequality, it follows that with $\frac{2}{\epsilon^2} \ln \frac{1}{\delta'}$ queries, we can approximate a candidate's value to additive accuracy $\epsilon/2$ and confidence $1 - \delta'$. To ensure that w.p. $\geq 1 - \delta$, all candidate values are approximated to an additive accuracy of $\epsilon/2$, one can evaluate the ith candidate using $\delta_i = \delta/(2i^2)$, and then invoke the union bound. Pseudocode for algorithm S-S is given in Appendix.

By construction, $|\hat{v}_c - v_c| \leq \epsilon/2$, for every candidate c. Hence right after interviewing the best candidate b, $\hat{v}_a \geq \hat{v}_b \geq v_b - \epsilon/2$. Since $\hat{v}_a$ never decreases, the same inequality holds for the final anchor a^*, namely, $v_{a^*} \geq \hat{v}_{a^*} - \epsilon/2 \geq v_b - \epsilon$.

For $\delta < 1/n$, we have $\ln \frac{n}{\delta} = \Theta(\ln \frac{1}{\delta})$, hence S-S uses $\Theta\big(\frac{n}{\epsilon^2} \ln \frac{1}{\delta}\big)$ queries, within a constant factor from the (Mannor & Tsitsiklis, 2004) lower bound. However, for higher confidences δ, *e.g.,* constant, it may require up to $\ln n$ times more queries than the lower bound. The remainder of the section eliminates this extra factor.

2.3. Properties of Sequential Maximization

We identify sufficient conditions for correctness of any sequential maximization algorithm, point out shortcomings of S-S, and combine these observations to derive a query-optimal algorithm.

The following two properties ensure an ϵ-maximum output:

Lemma 1. *Suppose that (i) the anchor is never replaced by a worse candidate, and (ii) when the best candidate is interviewed, if it is ϵ-better than the anchor, then it replaces the anchor. Then the final anchor is an ϵ-maximum.*

The lemma holds because the first condition ensures that the anchor's values are a non-decreasing sequence, and the second condition guarantees that right after the best candidate is interviewed, the anchor is an ϵ-maximum.

We will ensure that if anchor's value is well approximated then Lemma 1's first condition fails for candidate i with probability $\leq \frac{\delta}{16i^2}$, and the second fails with probability $\leq \frac{\delta}{4}$.

Let $\hat{v}_c$ be our approximation of candidate c's value v_c. To ensure that with high probability the anchor is not updated with a lower value candidate, we could as in S-S, ensure that with probability $\geq 1 - \delta$, all candidate values are approximated to within $\pm\epsilon/4$, namely $|\hat{v}_c - v_c| < \epsilon/4 \ \forall c$, and update the anchor only if $\hat{v}_c \geq \hat{v}_a + \epsilon/2$. However, as noted earlier, this would entail an extra $\ln n$ factor.

To circumvent this issue we approximate candidate values that are significantly lower than that of the anchor to lower confidence. Assume that anchor's value is approximated to within $\pm\epsilon/4$ i.e., $|\hat{v}_a - v_a| < \epsilon/4$.

We update anchor a only when $\hat{v}_c > \hat{v}_a + \epsilon/2$. If this happens when the actual values satisfy $v_c \leq v_a$ (and hence $v_c \leq \hat{v}_a + \epsilon/4$), we call that an *overestimation*. We ensure that for ith candidate, overestimation happens with probability $\leq \delta/(16i^2)$. By the union bound over all candidates, overestimation happens with probability $\leq \sum_i \frac{\delta}{16i^2} < \frac{\delta}{8}$.

Similarly, the second condition of Lemma 1 fails only if the best candidate b satisfies $v_b > v_a + \epsilon$ (and hence $v_b > \hat{v}_a + 3\epsilon/4$) yet our evaluation of b concludes $\hat{v}_b \leq \hat{v}_a + \epsilon/2$. We call that an *underestimation*. Since the best candidate is interviewed at most once, we ensure that underestimation occurs with probability $\leq \delta/4$.

Note the asymmetry between the two mis-estimations. Overestimation of any candidate can be irreversibly harmful, hence we ensure that the ith candidate, is overestimated with very small probability $\leq \delta/(16i^2)$. By contrast, underestimation is harmful only for the single best candidate. We therefore ensure that any given candidate is underestimated with a larger probability bound $\delta/4$.

By Hoeffding's Inequality (1), using $\frac{8}{\epsilon^2} \ln \frac{16i^2}{\delta}$ queries for ith candidate ensures that overestimation happens with probability $\leq \delta/(16i^2)$ and underestimation happens with probability $\leq \delta/(16i^2)$. Since we are allowed more leeway in underestimation probability bound, we can stop earlier if we are in underestimation regime. Notice that overestimation can happen only if $\hat{v}_c > \hat{v}_a + \epsilon/2$ and underestimation can happen only if $\hat{v}_c \leq \hat{v}_a + \epsilon/2$. Hence we can stop earlier before using all allocated queries if $\hat{v}_c \leq \hat{v}_a + \epsilon/2$. Observe that stopping earlier might only result in underestimation and will never result in overestimation. To ensure that probability of underestimation is $\leq \delta/4$, for a given candidate, we check if $\hat{v}_c \leq \hat{v}_a + \epsilon/2$ at specific checkpoints and terminate if it is the case. We move ahead of a checkpoint only if $\hat{v}_c > \hat{v}_a + \epsilon/2$ at the checkpoint. The checkpoints are selected such that by union bound over all checkpoints, underestimation happens with probability $\leq \delta/4$. The checkpoints help in terminating much earlier than using all queries in one shot.

Observe that for overestimation, we want to bound probability of overestimation at final checkpoint over all candidates. In contrast for underestimation, we want to bound probability of underestimation over all checkpoints for a single candidate. There exists several ways to allocate checkpoints to achieve this goal. We now present one such subroutine that takes advantage of the asymmetry between overestimation and underestimation using checkpoints.

2.4. ASYMMETRIC-THRESHOLD

ASYMMETRIC-THRESHOLD (A-T) approximates a candidate c's value v_c by comparing it against the threshold t at multiple checkpoints. Its goal is to determine whether v_c is larger than $t+\epsilon$ or smaller than $t-\epsilon$. Since we are more concerned with overestimation than underestimation, we consider unbalanced estimators where if v_c is below $t-\epsilon$, we output a value smaller than t w.p.$\geq 1-\delta_o$, and if v_c exceeds $t+\epsilon$, then we output a value higher than t w.p. $\geq 1-\delta_u$ for some $\delta_o < \delta_u$. One can derive similar algorithm for the case $\delta_u < \delta_o$.

To achieve this, A-T maintains checkpoints at consecutive integral multiples of $\lceil \frac{1}{2\epsilon^2} \rceil$ queries with first checkpoint at $\lceil \frac{1}{2\epsilon^2} \rceil \left(1 + \lceil \ln \frac{1}{\delta_u} \rceil \right)$ queries and final checkpoint at $\lceil \frac{1}{2\epsilon^2} \rceil \max \left(1 + \lceil \ln \frac{1}{\delta_u} \rceil, \lceil \ln \frac{1}{\delta_o} \rceil \right)$ queries. The checkpoints are present at $n_j = \lceil \frac{1}{2\epsilon^2} \rceil \left(j + \lceil \ln \frac{1}{\delta_u} \rceil \right)$ for $1 \leq j \leq \max(1, \lceil \ln \frac{1}{\delta_o} \rceil - \lceil \ln \frac{1}{\delta_u} \rceil)$. Notice that the number of checkpoints is

$$\max(1, \lceil \ln \frac{1}{\delta_o} \rceil - \lceil \ln \frac{1}{\delta_u} \rceil) \tag{2}$$

To approximate v_c, A-T first considers the fraction of queries answered correctly until the first checkpoint. If this fraction falls below t, the algorithm stops and returns the fraction as the approximation of v_c. If the fraction exceeds t, the candidate *passes* the first checkpoint, and A-T queries the candidate till the second checkpoint. If the fraction of queries answered correctly from the very first query until the second checkpoint falls below t the algorithm stops and returns this fraction as the approximated v_c, and so on. If the candidate passes all $\max(1, \lceil \ln \frac{1}{\delta_o} \rceil - \lceil \ln \frac{1}{\delta_u} \rceil)$ checkpoints, the algorithm returns the final cumulative average as the approximation of v_c.

For simplicity, let $V(v_c, t, \epsilon, \delta_u, \delta_o)$ be the output of A-T$(c, t, \epsilon, \delta_u, \delta_o)$, and let $N(v_c, t, \epsilon, \delta_u, \delta_o)$ be the number of queries used.

We bound the number of queries used by A-T and prove the asymmetric probability error bounds of overestimation and underestimation.

Lemma 2. *$N(p, t, \epsilon, \delta_u, \delta_o) = \mathcal{O}\left(\frac{1}{\epsilon^2} \ln \frac{1}{\delta_o}\right)$, and*

$$V(p, t, \epsilon, \delta_u, \delta_o) \begin{cases} < t \text{ w.p. } \geq 1-\delta_o & \text{if } \quad p < t-\epsilon, \\ \geq t \text{ w.p. } \geq 1-\delta_u & \text{if } \quad p \geq t+\epsilon. \end{cases}$$

Algorithm 1 ASYMMETRIC-THRESHOLD (A-T)

```
inputs
    candidate $c$, threshold $t$, bias $\epsilon$, underestimation confidence $\delta_u$, overestimation confidence $\delta_o < \delta_u$
initialize
    $l \leftarrow \lceil \frac{1}{2\epsilon^2} \rceil$,   $t \leftarrow \lceil \frac{1}{2\epsilon^2} \rceil \left(1 + \lceil \ln \frac{1}{\delta_u} \rceil\right)$
Ask $c$, $t$ queries. $\hat{v}_c \leftarrow$ Fraction of correct responses
while $t < \lceil \frac{1}{2\epsilon^2} \rceil \lceil \ln \frac{1}{\delta_o} \rceil$ and $\hat{v}_c \geq t$ do
    Ask $c$, $l$ queries. $\hat{x} \leftarrow$ Fraction of correct responses
    $\hat{v}_c = \frac{t}{t+l}\hat{v}_c + \frac{l}{t+l}\hat{x}$
    $t \leftarrow t + l$
end while
return $\hat{v}_c$
```

Let $E_{\text{last}}(p, t, \epsilon, \delta_u, \delta_o)$ be the event that either last checkpoint is not invoked or candidate's value is approximated to an accuracy of ϵ. We now bound the probability of $E_{\text{last}}(p, t, \epsilon, \delta_u, \delta_o)$.

Lemma 3.

$$Pr(E_{\textit{last}}(p, t, \epsilon, \delta_u, \delta_o)) \geq 1 - 2\delta_o.$$

We prove the majorization property of queries used by A-T. These properties play a crucial role in bounding queries of our main algorithm. Notice that when A-T is called with overestimation confidence parameter as 0, it will have infinite allocated queries and will keep querying until candidate's estimated value falls below threshold at a checkpoint. We first show that worse candidates when queried against higher threshold use fewer queries.

Lemma 4. *For any $p' \leq p$, $t' \geq t$,*

$$\Pr(N(p', t', \epsilon, \delta_u, \delta_o) > m) \leq \Pr(N(p, t, \epsilon, \delta_u, 0) > m).$$

We now lower bound the probability of better candidates using all allocated queries by the probability that worse candidates using more queries when called with overestimation confidence parameter of 0.

Lemma 5. *For any $p' \geq p$, $t' \leq t$, $m \geq \lceil \frac{1}{2\epsilon^2} \rceil \lceil \ln \frac{1}{\delta_o} \rceil$,*

$$\Pr\left(N(p', t', \epsilon, \delta_u, \delta_o) \geq \lceil \frac{1}{2\epsilon^2} \rceil \lceil \ln \frac{1}{\delta_o} \rceil\right)$$
$$\geq \Pr\left(N(p, t, \epsilon, \delta_u, 0) \geq m\right).$$

2.5. OPTIMAL-SEQUENTIAL

We now present our main algorithm OPTIMAL-SEQUENTIAL (O-S).

As mentioned before, O-S ensures that for ith candidate, overestimation happens with probability $\leq \delta/(16i^2)$ and underestimation happens with probability $\leq \delta/4$. To achieve this, for ith candidate, O-S invokes A-T with threshold at $\hat{v}_a + \epsilon/2$, bias of $\epsilon/4$, underestimation confidence of $\delta/4$ and overestimation confidence of $\delta/(16i^2)$. From Equation (2), A-T compares the ith candidate's approximated value against the threshold at at most $\lceil \ln(4i^2) \rceil$ checkpoints and if the candidate's approximated value falls below threshold at any checkpoint, A-T will not invoke further checkpoints, thereby saving queries.

Algorithm 2 OPTIMAL-SEQUENTIAL (O-S)

```
inputs
  Set $S$, bias $\epsilon$, uncertainty $\delta$
initialize
  Anchor's estimated value $\hat{v}_a \leftarrow -\infty$, number of elements considered $i \leftarrow 0$
while $S \neq \emptyset$ do
  $c \leftarrow$ random element of $S$, $S \leftarrow S \setminus \{c\}$, $i \leftarrow i + 1$
  $\hat{v}_c \leftarrow$ A-T$(c, \hat{v}_a + \frac{\epsilon}{2}, \frac{\epsilon}{4}, \frac{\delta}{4}, \frac{\delta}{16i^2})$
  if $\hat{v}_c \geq \hat{v}_a + \epsilon/2$ then
    $\hat{v}_a \leftarrow \hat{v}_c$, $a \leftarrow c$
  end if
end while
return a
```

2.5.1. Correctness Proof

We first prove the correctness of O-S. To prove correctness, we never use randomness in candidates' arrival. Hence w.h.p., O-S outputs an ϵ-maximum even for adversarially picked sequence of candidates.

Recall that if we ensure conditions in Lemma 1, then the output is an ϵ-maximum.

To prove that w.h.p., anchor is never replaced by a worse candidate, we first show that for any candidate, w.h.p.,, either last checkpoint is not invoked or the candidate's value is approximated to an additive accuracy of $\epsilon/4$. Since anchor is updated only if the final checkpoint is invoked, w.h.p., anchor's value is always approximated to an additive accuracy of $\epsilon/4$. Notice that in O-S, when calling A-T, threshold is always set to be $\epsilon/2$ more than that of current anchor's approximated value. Therefore, w.h.p., threshold is always at least $\epsilon/4$ more than that of anchor's true value. Once again, recall that anchor is updated only when final checkpoint is invoked. If the value of current candidate is less than that of anchor, then w.h.p., either the last checkpoint is not invoked or candidate's value is approximated to an additive accuracy of $\epsilon/4$, and hence approximated value fails to be more than that of threshold. Therefore, anchor will never be replaced by a worse candidate. We prove the above arguments formally in below lemmas. Some definitions follow.

Let $E_{i,\text{last}}$ be defined as the event that either the last checkpoint was not invoked for candidate i, or its value is approximated to an additive accuracy of $\leq \epsilon/4$. We bound the probability of $E_{i,\text{last}}$ happening over all i.

Lemma 6. $\Pr(\bigcup_i E_{i,\mathit{last}}) \geq 1 - \delta/4$.

Now we show that w.h.p., all anchors' values are approximated to an additive accuracy of $\epsilon/4$.

Lemma 7. *Under event $\bigcup_i E_{i,\mathit{last}}$, values of all anchors are approximated to an additive accuracy of $\epsilon/4$ i.e.,*

$$|\hat{v}_a - v_a| \leq \epsilon/4.$$

Now we prove that anchor never gets replaced by a worse candidate.

Lemma 8. *Under event $\bigcup_i E_{i,\mathit{last}}$, anchor never gets worse.*

Now we prove that after best candidate is interviewed the anchor is an ϵ-maximum.

Event E_{best} : After the best candidate is interviewed, anchor will be an ϵ-maximum. We bound the probability of E_{best}.

Lemma 9. $\Pr(E_{\mathit{best}} | \bigcup_i E_{i,\mathit{last}}) \geq 1 - \delta/4$.

Now we prove the correctness of O-S.

Theorem 10. *W.p.$\geq 1 - \delta/2$,* O-S(S, ϵ, δ) *outputs an ϵ-maximum.*

2.5.2. Query Analysis

We now bound the query complexity of O-S. We first consider the case of low delta namely $\delta < 200/n^{1/3}$ and show that queries used by O-S is orderwise optimal.

Theorem 11. *For $\delta < 200/n^{1/3}$,* O-S(S, ϵ, δ) *uses $\mathcal{O}\big(\frac{n}{\epsilon^2} \ln \frac{1}{\delta}\big)$ queries.*

So from here on we assume $\delta > 200/n^{1/3}$ and bound the query complexity using the randomness of the sequence. We first outline the proof that bounds the query complexity.

Proof Sketch Recall from Algorithm O-S that for the i-th candidate, $\delta_u = \delta/4$ and $\delta_o = \delta/(16i^2)$. From Equation (2), candidates $\leq i$ will be interviewed at $\leq \lceil \ln(4i^2) \rceil$ checkpoints. We upper bound the number of later candidates (arrive after first i candidates) that are likely to be interviewed at checkpoint $\lceil \ln(4i^2) + 1 \rceil$ for each i. To achieve this we first lowerbound the threshold after interviewing first i candidates.

Recall that jth checkpoint is $n_j = \lceil \frac{8}{\epsilon^2} \rceil \big(j + \lceil \ln \frac{4}{\delta} \rceil\big)$. Let r_k be the candidate with the k-th highest value, where ties are broken arbitrarily. Omitting ϵ and δ for brevity, define

$$C_{k,l,\alpha} \overset{\text{def}}{=} \sup\{t : \Pr(N(v_{r_k}, t, \epsilon/4, \delta/4, 0) \geq n_l) \geq \alpha\}$$

to be the highest threshold against which the kth highest valued candidate will pass all first l checkpoints w.p. $\geq \alpha$.

Lemma 12 observes that if the threshold exceeds the candidate's value plus $\epsilon/4$, then with high probability the candidate will not pass even the first few checkpoints. More precisely that for every l and $\alpha > \delta/(4e^l)$, $C_{k,l,\alpha} \leq v_{r_k} + \epsilon/4$.

In Lemma 13 we combine this lemma, majorization property of A-T, and the sequence randomness to show that with high probability, the threshold after interviewing i candidates exceeds $C_{k,\lceil\ln(4i^2)\rceil,\sqrt{4n/ki}}$.

Lemma 14, then deduces that with high probability, at most $\mathcal{O}\big(\frac{n}{i^{1/3}}\big)$ candidates will be interviewed at $\lceil\ln(4i^2)+1\rceil$ checkpoint.

Finally, Theorem 15 bounds the total number of queries by summing over number of times each checkpoint is invoked.

Formal Proof We first upperbound $C_{k,l,\alpha}$ using value v_{r_k} of the kth ranked candidate. .

Lemma 12. *For any $\alpha > \frac{\delta}{4e^l}$,*

$$C_{k,l,\alpha} \le v_{r_k} + \epsilon/4.$$

Now we can lower bound the threshold after interviewing i candidates . Let t_i be the threshold after interviewing i candidates. The Lemma below lower bounds the value of t_i. This is the only Lemma that uses the randomness in the arrival of candidates.

Lemma 13. *For any i and k s.t. $\sqrt{\frac{4n}{ki}} > \frac{\delta}{4i^2}$, w.p. $\ge 1 - e^{-\frac{ki}{4n}} - e^{-\sqrt{\frac{ki}{64n}}}$,*

$$t_i \ge C_{k,\lceil\ln(4i^2)\rceil,\sqrt{\frac{4n}{ki}}}.$$

Now we bound the number of candidates invoked for a checkpoint. For this we use Lemma 13 to bound the threshold after first i candidates and bound the number of candidates ranked outside k candidates that can cross this threshold.

Lemma 14. *For any i and k s.t. $\sqrt{\frac{4n}{ki}} > \frac{\delta}{4i^2}$,*

w.p.$\ge 1 - e^{-\frac{ki}{4n}} - e^{-\sqrt{\frac{ki}{64n}}} - e^{-n\sqrt{\frac{4n}{ki}}}$, the number of times $\big(\lceil\ln(4i^2)\rceil + 1\big)$th checkpoint invoked is

$$\le k + n\sqrt{\frac{144n}{ki}}.$$

The below theorem establishes the query complexity of O-S.

Theorem 15. *For $\delta > 200/n^{1/3}$, w.p.$\ge 1 - \delta/2$, O-S(S,ϵ,δ) uses $\mathcal{O}\big(\frac{n}{\epsilon^2}\ln\frac{1}{\delta}\big)$ queries.*

3. Dueling Bandits Sequential Maximization

3.1. Tools

We use subroutine COMPARE (Falahatgar et al., 2017a) as a building block in our maximization algorithms. For the reader's convenience, we provide a brief outline of COMPARE here and state its guarantees in Lemma 16. We also present the algorithm COMPARE in Appendix.

For $\epsilon_u > \epsilon_l$, COMPARE$(i, j, \epsilon_l, \epsilon_u, \delta)$ compares elements i and j for $\mathcal{O}\Big(\frac{1}{(\epsilon_u-\epsilon_l)^2}\log\frac{1}{\delta}\Big)$ times and deems if $\tilde{p}_{i,j} \le \epsilon_l$ (returns 1) or $\tilde{p}_{i,j} \ge \epsilon_u$ (returns 2). The guarantees are presented in Lemma 16.

Lemma 16 (Lemma 1 (Falahatgar et al., 2017a))**.** *For $\epsilon_u > \epsilon_l$, COMPARE$(i, j, \epsilon_l, \epsilon_u, \delta)$ uses $\le \frac{2}{(\epsilon_u-\epsilon_l)^2}\log\frac{2}{\delta}$ comparisons and if $\tilde{p}_{i,j} \le \epsilon_l$, then w.p.$\ge 1-\delta$, returns 1, else if $\tilde{p}_{i,j} \ge \epsilon_u$, w.p.$\ge 1-\delta$, returns 2.*

3.2. Agnostic Version of SEQ-ELIMINATE (Falahatgar et al., 2017a)

Recall that under models with SST property, SEQ-ELIMINATE is a sub optimal maximization algorithm and is sequential and requires the knowledge of n a priori.

We first describe an outline of SEQ-ELIMINATE and present an easy fix to make it n-agnostic with orderwise same sample complexity. SEQ-ELIMINATE starts with the first element as the anchor r, sequentially compares r with elements of S using COMPARE$(S(i), r, 0, \epsilon, \delta/n)$, and updates r with $S(i)$ if COMPARE returns 2. This ensures that with probability $1-\delta/n$: 1) the updated anchor is at least as good as the previous anchor, and 2) the updated anchor is ϵ-preferable to $S(i)$. These two key properties along with SST property and the union bound, ensure that w.p.$\ge 1-\delta$, the final anchor is an ϵ-maximum. Notice that to ensure that the total error probability is bounded by δ, SEQ-ELIMINATE uses each instance of COMPARE with confidence parameter δ/n and hence requires knowing n beforehand. A simple fix is to use confidence parameter $\delta/(2i^2)$ (observe that $\sum_{i=1}^{\infty}\delta/(2i^2) \le \delta$) when using the ith instance of COMPARE and hence does not require knowing the value of n. Now we present the maximization algorithm AGNOSTIC-SEQ with this fix applied to SEQ-ELIMINATE. Notice that even the nth instance of COMPARE uses $\mathcal{O}\big(\frac{1}{\epsilon^2}\log\frac{n}{\delta}\big)$ comparisons and hence AGNOSTIC-SEQ has orderwise the same comparison complexity as SEQ-ELIMINATE. The pseudocode for AGNOSTIC-SEQ is provided in Appendix.

In the Lemma 17, we prove the correctness and bound the comparison complexity of AGNOSTIC-SEQ.

Lemma 17. *Under SST model,* AGNOSTIC-SEQ(S,ϵ,δ) *uses $\mathcal{O}\big(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\big)$ comparisons and w.p.$\ge 1-\delta$, outputs an ϵ-maximum.*

Observe that AGNOSTIC-SEQ is n-agnostic and min-max optimal for $\delta \le \frac{1}{n}$ but requires an extra multiplicative factor of $\log n$ comparisons than the known lower bound for constant δ.

3.3. Optimal Agnostic Sequential Maximization

In this subsection, for models with SST property, we present maximization algorithm OPT-AGNOSTIC-SEQ that is both sequential and n-agnostic and yet uses orderwise same comparisons as the min-max optimal maximization algorithm that has the knowledge of n and is not necessarily sequential. Hence OPT-AGNOSTIC-SEQ is also a min-max optimal maximization algorithm.

Due to lack of space, here we only provide a brief outline of our algorithm and state the main result. The motivation and analysis is very similar to that of OPTIMAL-SEQUENTIAL and is presented in detail in Appendix.

3.3.1. OPT-ANCHOR-UPDATE

Observe that in each instance to update the anchor, AGNOSTIC-SEQ uses COMPARE with confidence parameter $\frac{\delta}{8i^2}$. Here we present an alternative OPT-ANCHOR-UPDATE for using COMPARE in one shot. Similar to ASYMMETRIC-THRESHOLD, within each instance of OPT-ANCHOR-UPDATE, we use multiple rounds of COMPARE, decreasing the confidence parameter with each consecutive round such that overall comparisons used over all rounds are orderwise same as comparisons used in a single instance of COMPARE with confidence parameter $\Theta(\frac{\delta}{i^2})$. Within each instance, we move to the next COMPARE round only if the previous round returns 2. This helps in terminating much earlier than if only one round of COMPARE is used.

Algorithm 3 OPT-ANCHOR-UPDATE

```
inputs
    element $e$, element $f$, bias $\epsilon$, confidence $\delta$, number $i$
Initialize: $t \leftarrow 0$, $a \leftarrow 2$
while $a = 2$ and $t < \max(2, \log\log_{\frac{1}{\delta}} i^2 + 1)$ do
    $a \leftarrow$ COMPARE$(e, f, 0, \epsilon, \delta^{2^t+1}/8)$
    $t \leftarrow t + 1$
end while
if $a = 1$ then
    return $f$
else
    return $e$
end if
```

3.3.2. OPT-AGNOSTIC-SEQ

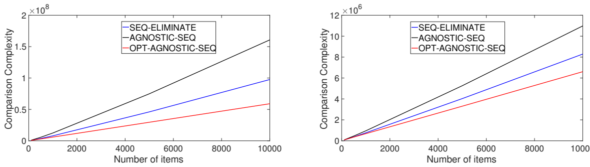
We now present our main algorithm OPT-AGNOSTIC-SEQ that uses OPT-ANCHOR-UPDATE as subroutine to update the anchor.

In the below Theorem, we provide guarantees for OPT-AGNOSTIC-SEQ.

Theorem 18. *Under SST models, w.p.$\geq 1 - \delta$,* OPT-AGNOSTIC-SEQ (S, ϵ, δ) *uses* $\mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\right)$ *comparisons and outputs an* $\epsilon-$*maximum.*

Algorithm 4 OPT-AGNOSTIC-SEQ

```
inputs
    Set $S$, bias $\epsilon$, confidence $\delta$
anchor $r \leftarrow S(1)$, $S = S \setminus \{r\}$, candidate number $i \leftarrow 0$
while $S \neq \emptyset$ do
    $c \leftarrow$ random element of $S$, $S = S \setminus \{c\}$, $i \leftarrow i + 1$
    $r \leftarrow$ OPT-ANCHOR-UPDATE$(c, r, \epsilon, \delta, i)$
end while
return $r$
```



(a) all elements are equal (b) strict ordering

Figure 1. Comparison of Maximization Algorithms

4. Experiments

In this section, we compare the performance of various sequential maximization algorithms SEQ-ELIMINATE (Falahatgar et al., 2017a), AGNOSTIC-SEQ and OPT-AGNOSTIC-SEQ. Note that SEQ-ELIMINATE uses the knowledge of n whereas AGNOSTIC-SEQ and OPT-AGNOSTIC-SEQ are n-agnostic. Further recall that SEQ-ELIMINATE and AGNOSTIC-SEQ are sub-optimal with query complexity of $\mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{n}{\delta}\right)$ and OPT-AGNOSTIC-SEQ is optimal with query complexity of $\mathcal{O}\left(\frac{n}{\epsilon^2}\log\frac{1}{\delta}\right)$. Experiments in (Falahatgar et al., 2017a; 2018) demonstrate that SEQ-ELIMINATE performs better than other maximization algorithms. Hence we don't compare with other maximization algorithms. In all the experiments in this section, we try to find an 0.05-maximum with $\delta = 0.1$. All results are averaged over 100 runs.

We first consider the model where all items are essentially equal i.e., $p_{i,j} = 1/2 \ \forall i, j$. Figure 1(a) show the performance of sequential maximization algorithms for this model. Notice that OPT-AGNOSTIC-SEQ uses significantly less comparisons than both SEQ-ELIMINATE and AGNOSTIC-SEQ. Notice that since AGNOSTIC-SEQ is an agnostic version of SEQ-ELIMINATE, AGNOSTIC-SEQ uses more comparisons than SEQ-ELIMINATE.

We now consider the model where $p_{i,j} = 0.6 \ \forall i < j$ same as in (Yue & Joachims, 2011; Falahatgar et al., 2017b;a; 2018). Figure 1(b) presents the performance of sequential maximization algorithms for this model. Notice again that OPT-AGNOSTIC-SEQ uses less comparisons than SEQ-ELIMINATE, that in turn uses fewer comparisons than AGNOSTIC-SEQ.

Since (Falahatgar et al., 2017a; 2018) showed that SEQ-ELIMINATE outperforms other maximization algorithms and empirical performance of OPT-AGNOSTIC-SEQ is better than SEQ-ELIMINATE, OPT-AGNOSTIC-SEQ outperforms even non-sequential maximization algorithms.

5. Conclusion and Future Work

We presented the first optimal sequential probabilistic maximization algorithm that works even without a-priori knowledge of number of items. The algorithm has linear complexity both under traditional- and dueling (with SST property)- bandits frameworks. In the future, we propose to extend these works to more general settings.

References

`https://www.forbes.com/sites/lizryan/2016/11/20/no-i-wont-come-back-for-a-fifth-interview/#34e50b5863d7`.

Audibert, J.-Y. and Bubeck, S. Best arm identification in multi-armed bandits. In *COLT-23th Conference on learning theory-2010*, pp. 13–p, 2010.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Bubeck, S., Munos, R., and Stoltz, G. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pp. 23–37. Springer, 2009.

Bubeck, S., Cesa-Bianchi, N., et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Caplin, A. and Nalebuff, B. Aggregation and social choice: A mean voter theorem. *Econometrica: Journal of the Econometric Society*, pp. 1–23, 1991.

David, Y. and Shimkin, N. Infinitely many-armed bandits with unknown value distribution. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 307–322. Springer, 2014.

Dekel, O., Ding, J., Koren, T., and Peres, Y. Bandits with switching costs: T 2/3 regret. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 459–467, 2014.

Even-Dar, E., Mannor, S., and Mansour, Y. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

Falahatgar, M., Hao, Y., Orlitsky, A., Pichapati, V., and Ravindrakumar, V. Maxing and ranking with few assumptions. In *Advances in Neural Information Processing Systems*, pp. 7060–7070, 2017a.

Falahatgar, M., Orlitsky, A., Pichapati, V., and Suresh, A. T. Maximum selection and ranking under noisy comparisons. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1088–1096. JMLR. org, 2017b.

Falahatgar, M., Jain, A., Orlitsky, A., Pichapati, V., and Ravindrakumar, V. The limits of maxing, ranking, and preference learning. In *International Conference on Machine Learning*, pp. 1426–1435, 2018.

Gabillon, V., Ghavamzadeh, M., and Lazaric, A. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, pp. 3212–3220, 2012.

Hoeffding, W. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pp. 409–426. Springer, 1994.

Karnin, Z., Koren, T., and Somekh, O. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, pp. 1238–1246, 2013.

Koren, T., Livni, R., and Mansour, Y. Multi-armed bandits with metric movement costs. In *Advances in Neural Information Processing Systems*, pp. 4119–4128, 2017.

Lai, T. L. and Robbins, H. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Mannor, S. and Tsitsiklis, J. N. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

Robbins, H. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

Schumann, C., Counts, S. N., Foster, J. S., and Dickerson, J. P. The diverse cohort selection problem: Multi-armed bandits with varied pulls. *CoRR*, abs/1709.03441, 2017. URL `http://arxiv.org/abs/1709.03441`.

Szörényi, B., Busa-Fekete, R., Paul, A., and Hüllermeier, E. Online rank elicitation for plackett-luce: A dueling bandits approach. In *Advances in Neural Information Processing Systems*, pp. 604–612, 2015.

Yue, Y. and Joachims, T. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 241–248, 2011.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zhou, Y., Chen, X., and Li, J. Optimal PAC multiple arm identification with applications to crowdsourcing. In *International Conference on Machine Learning*, pp. 217–225, 2014.