Oracle-Augmented Prophet Inequalities

Sariel Har-Peled* Elfarouk Harb† Vasilis Livanos‡

April 19, 2024

Abstract

In the classical prophet inequality settings, a gambler is given a sequence of n random variables $X_1, \ldots, X_n$, taken from known distributions, observes their values in this (potentially adversarial) order, and select one of them, immediately after it is being observed, so that its value is as high as possible. The classical *prophet inequality* shows a strategy that guarantees a value at least half of that an omniscience prophet that picks the maximum, and this ratio is optimal.

Here, we generalize the prophet inequality, allowing the gambler some additional information about the future that is otherwise privy only to the prophet. Specifically, at any point in the process, the gambler is allowed to query an oracle $\mathcal{O}$. The oracle responds with a single bit answer: YES if the current realization is greater than the remaining realizations, and NO otherwise. We show that the oracle model with m oracle calls is equivalent to the Top-1-of-$(m+1)$ model when the objective is maximizing the probability of selecting the maximum. This equivalence fails to hold when the objective is maximizing the competitive ratio, but we still show that any algorithm for the oracle model implies an equivalent competitive ratio for the Top-1-of-$(m+1)$ model.

We resolve the oracle model for any m, giving tight lower and upper bound on the best possible competitive ratio compared to an almighty adversary. As a consequence, we provide new results as well as improvements on known results for the Top-1-of-m model.

1. Introduction

The field of optimal stopping theory concerns optimization settings where one makes decisions in a sequential manner, given imperfect information about the future, in a bid to maximize a reward or minimize a cost. A canonical setting in this area is the *prophet inequality* [KS77, KS78]. In these settings, a gambler is presented with rewards $X_1, \ldots, X_n$, one after the other, drawn independently from known distributions. Upon seeing a reward X_i, the gambler must immediately make an irrevocable decision to either accept X_i, in which case the process ends, or to reject X_i and continue, losing the option to select X_i in the future. The goal of the gambler is to maximize

*Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@illinois.edu; http://sarielhp.org/. Work on this paper was partially supported by NSF AF award CCF-2317241.

†Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; eyharb2@illinois.edu; https://farouky.github.io/.

‡Department of Computer Science; University of Chile; Chile; livanos3@illinois.edu; https://livanos3.web.engr.illinois.edu/.

the selected reward comparing against a *prophet* who knows all realizations in advance and selects the maximum realized reward. Throughout, we assume, without loss of generality, that $X_1, \dots, X_n$ are continuous random variables.

The performance of the gambler can be measured in terms of several objectives. A common metric used in the literature is the *competitive ratio*, which is also known as the *Ratio of Expectations (RoE)* (see Definition 1.2). Another common distinction is between the case in which the given distributions are different and the case in which they are identical. For the former, Krengel *et al.* [KS77, KS78] showed an optimal strategy that is $1/2$-competitive. In this setting, the optimal competitive ratio can be achieved by simple, single-threshold algorithms [Sam84, KW19]. For IID and non-IID random variables, Hill and Kertz [HK82] initially gave a $(1-1/e)$-competitive algorithm. This was improved to ≈ 0.738 [AEE+17] and later ≈ 0.745 [CFH+21], which is tight, due to a matching upper bound [HK82, Ker86].

Another relevant metric, introduced by Gilbert and Mosteller [GM66] for IID random variables, is that of maximizing the *Probability of selecting the Maximum realization* ($\mathbb{P}_{\max}$) - see Definition 1.3. For this objective and IID random variables, Gilbert and Mosteller [GM66] gave an algorithm that achieves a probability of ≈ 0.58, which is the best possible. Later, Esfandiari, Hajiaghayi, Lucier and Mitzenmacher [EHLM17] studied the same objective for general random variables, obtaining a tight probability equal to $1/e$ when the random variables arrive in adversarial order and 0.517 when the random variables arrive in random order. The latter case was recently improved to the tight ≈ 0.58 by Nuti [Nut22], showing that the IID setting is not easier than the non-IID setting with random order. In this paper, we introduce a new model as a means to study variations of both the IID and the general settings, for both the RoE and $\mathbb{P}_{\max}$ objectives.

A setting that is related to ours is the Top-1-of-m model, formally introduced by Assaf and Samuel-Cahn [AS00] for IID random variables, although it had been studied initially by Gilbert and Mosteller [GM66]. In this setting, the algorithm is allowed to select $m \geq 1$ values, but the value it gets judged by is the maximum selected value. Gilbert and Mosteller [GM66] gave numerical approximations of the $\mathbb{P}_{\max}$ objective for $2 \leq m \leq 10$, using a simple, single-threshold algorithm. Later, Assaf and Samuel-Cahn [AS00] studied the RoE objective for general distributions and gave an elegant and simple $(1 - 1/m+1)$-competitive algorithm. This was improved [AGS02] by bounding the competitive ratio of the optimal algorithm by a recursive differential equation. They gave numerical approximations for $2 \leq m \leq 5$, but studying the asymptotic nature of the constants for large m turned out to be difficult. Ezra *et al.* [EFN18] later revisited the problem and gave a new algorithm for large m that is $1 - \mathcal{O}\left(e^{-m/6}\right)$-competitive for the same problem. This improves the error term from [AGS02] from linear in m to exponential in m. Harb [Har24] recently improved this into a $1 - e^{-mW_0\left(\frac{\sqrt[m]{m!}}{m}\right)}$-competitive algorithm, where W_0 is the Lambert-W function[1], and improved the lower bound for $m = 2$ separately. However, the asymptotic nature of this function is difficult to analyze.

Type of Adversary. In the context of prophet inequalities, the distinction between an offline adversary and an almighty adversary is crucial to understanding the competitive ratio bounds of prophet inequalities. An offline adversary, often considered less powerful, observes the *distributions* of the random variables, and chooses an adversarial order based on the distributions. An almighty adversary is stronger:

[1] The Lambert-W function is $W_0(x)$ defined as the solution y to the equation $ye^y = x$.

Definition 1.1. An ***almighty adversary*** possesses complete information, including the algorithm's random decisions, and can thus tailor the ordering of the sequence to worst-case scenarios with perfect foresight. In particular, the almighty adversary observes the gambler's algorithm, and the values of $X_1, \dots, X_n$, then chooses a permutation order $X_{\sigma(1)}, \dots, X_{\sigma(n)}$ to show the algorithm the values in that order.

Consequently, while both types of adversaries present different levels of challenge, the almighty adversary sets a far stricter benchmark, typically leading to lower competitive ratios for prophet inequalities. Unless stated otherwise, we work with the almighty adversary. For a discussion on the (very subtle) differences between the two adversaries for our model, see Appendix B.

Model. We introduce a new model that generalizes the standard prophet inequality setting, and analyze it as a means to obtain new results and improvements in the TOP-1-OF-m model. Our model allows the algorithm some information about the future that is otherwise privy only to the prophet. Specifically, at any point in the process, upon seeing a reward X_i, the algorithm is allowed to query an oracle $\mathcal{O}$. The oracle $\mathcal{O}$ responds with a single bit answer: YES if the current realization is larger than the remaining realizations, i.e., $X_i > \max_{j=i+1}^{n} X_j$ and NO otherwise. In other words, the oracle $\mathcal{O}$ informs the algorithm it should select X_i, or reject it, because there is a reward coming up that is at least as good[2]. Clearly, with no queries available, one recovers the classical prophet inequality setting, whereas with $n-1$ queries, the strategy of using a query on every X_i, for $i = 1, \dots, n-1$, leads to the algorithm selecting the highest realization always. Thus, this model interpolates nicely between the two extremes of full or no information about the future.

In this paper, we consider the following different settings.

Definition 1.2. The competitive ratio or *Ratio of Expectations* is denoted by RoE. An algorithm ALG is *α-competitive*, for $\alpha \in [0, 1]$, if $\mathbb{E}[ALG] \geq \alpha \cdot \mathbb{E}[\max_i X_i]$, and α is called the *competitive ratio*.

Definition 1.3. The *Probability of selecting the Maximum* realization is denoted by $\mathbb{P}_{\max}$. An algorithm ALG achieves a $\mathbb{P}_{\max}$ of α if it returns a value v such that $\mathbb{P}[v = Z] \geq \alpha$, where $Z = \max\{X_1, \dots, X_n\}$. Note that in some works (for example [GM66]), the notation PbM has also been used.

Definition 1.4. We use the term IID to refer to the setting where $X_1, \dots, X_n$ are independent and identically distributed random variables. We use non-IID to refer to the more general setting where $X_1, \dots, X_n$ are independent, but not necessarily identical.

Definition 1.5. We use PROPH_m to refer to the TOP-1-OF-m model, in which the algorithm is allowed to choose up to m values, and its payoff is the maximum of the chosen values. We use $\mathcal{O}_m$ refers to our oracle model where the algorithm has access to m oracle calls, and can only select one value.

Note that it makes sense to compare the model PROPH_{m+1} to $\mathcal{O}_m$ since in the former, the algorithm can choose $m+1$ values, where as the later can ask the oracle m times, and then choose

[2] There are *very subtle* differences between an oracle that answers $>$ queries, vs $\geq$ queries. See Appendix B for a discussion on this. In particular, the $>$ oracle is weaker than the $\geq$ oracle; imagine a stream of 1 values, the $>$ oracle will always answer NO, while the $\geq$ oracle answers YES on the first query

an item. To help distinguish between the different settings, we denote each model as $\mathcal{M}(x, y, z)$, where

- x is either Proph_m or $\mathcal{O}_m$ with $m \in \mathbb{N}$,
- y is either IID or $\mathsf{non\text{-}IID}$, and
- z is either $\mathbb{P}_{\max}$ or RoE.

Motivation. Our oracle choice was driven by our initial effort to reformulate the Top-1-of-m model in order to get a better understanding of that settings and improve the known bounds. While we initially thought the two models are the same, we later observed the subtle differences between the two models. Thankfully, it turns out that one can still use this oracle model to study the Top-1-of-m model and improve the previously known bounds, which was our original goal. We did consider a few other oracle models, which we briefly mention here: if the oracle predicts the maximum value, then there is a trivial solution of asking the oracle at X_1, and just waiting until the maximum value v arrives. Another option is for the oracle to predict a *range* for the maximum value, but formalizing this in a more general setting turns out to be difficult without assuming something about the support of each random variable. While we explored various other oracle formulations, we chose the simplest version that improved the lower bound for the Top-1-of-m model and for which we can also achieve a tight competitive ratio. We leave exploring more complex oracle models for future work.

1.1. Our contributions

In this paper, we study the oracle model for independent random variables following identical or general distributions with the $\mathbb{P}_{\max}$ and RoE objectives and make the following contributions:

(I) We establish an *equivalence* between the oracle model and the Top-1-of-m model for the $\mathbb{P}_{\max}$ objective.

(II) We show that this equivalence fails to hold for the RoE objective and that the best-possible competitive ratios in the two settings are quite separated. However, we show that guarantees for RoE in the oracle model translate to guarantees in the Top-1-of-m model, thus further motivating our study of the oracle model.

(III) We resolve the oracle model $\mathcal{M}(\mathcal{O}_m, \mathsf{non\text{-}IID}, \mathsf{RoE})$ by presenting a single-threshold algorithm. Our algorithm achieves a competitive ratio of $1 - e^{-\xi_m}$ for general m, where ξ_m is the unique *positive* solution[3] to the equation $1 - e^{-\xi_m} = \frac{\Gamma(m+1, \xi_m)}{m!}$[4]. Furthermore, we show that this lower bound is optimal by showing a construction that yields an equal upper bound. Since we showed that lower bound guarantees for $\mathcal{M}(\mathcal{O}_m, \mathsf{non\text{-}IID}, \mathsf{RoE})$ also hold for the alternative settings $\mathcal{M}(\text{Proph}_{m+1}, \mathsf{non\text{-}IID}, \mathsf{RoE})$, this strictly improves the current state of the art bounds of [Har24], even though the guarantees are obtained in the weaker oracle model.

(IV) We give a single-threshold algorithm for the oracle model and the $\mathbb{P}_{\max}$ objective $\mathcal{M}(\mathcal{O}_m, \mathsf{IID}, \mathbb{P}_{\max})$ that achieves a $1 - \mathcal{O}(m^{-m/5})$ probability of selecting the maximum, as well as providing an upper bound that is asymptotically (almost) tight. To the best of our knowledge, this is the first result for the $\mathbb{P}_{\max}$ objective and general m in the well studied Top-1-of-m model. Our algorithm achieves a probability of ≈ 0.797 even with $m = 1$ calls to the oracle, a significant improvement on the ≈ 0.58 achieved without oracle calls [GM66].

[3] In Section 3, we prove that there is indeed a unique positive solution.

[4] $\Gamma(n, x) = \int_x^\infty t^{n-1} e^{-t}\, dt$ denotes the upper incomplete gamma function

As discussed earlier, the main motivation behind our oracle model comes from our first two results which relate it to the TOP-1-OF-m model.

1.2. Our results in detail

1.2.1. Equivalence of models as far as $\mathbb{P}_{\max}$

In Theorem 2.3 we prove that $\mathcal{M}(\mathcal{O}_m, y, \mathbb{P}_{\max})$ model is equivalent to the $\mathcal{M}(\text{PROPH}_{m+1}, y, \mathbb{P}_{\max})$ model, where $y = \mathsf{IID}$ or $\mathsf{non\text{-}IID}$. In other words, the best algorithms in these models achieve the same probability of realizing the maximum.

This result might not seem that surprising due to the apparent similarity of the two models. However, thinking about the TOP-1-OF-m setting from the viewpoint of oracle calls allows for a different perspective that we exploit in our analysis.

1.2.2. Difference of the models as far as RoE

Perhaps more surprisingly, our oracle model and the TOP-1-OF-m model stop being equivalent when one considers the RoE objective. The oracle model is strictly weaker.

Specifically, in Theorem 2.8, we prove that there exists a prophet inequality instance, and an algorithm $\mathcal{A}$ for $\mathcal{M}(\text{PROPH}_{m+1}, \mathsf{non\text{-}IID}, \mathsf{RoE})$ instance for which no algorithm for $\mathcal{M}(\mathcal{O}_m, \mathsf{non\text{-}IID}, \mathsf{RoE})$ can achieve the same competitive ratio as that of $\mathcal{A}$. Furthermore, any algorithm for $\mathcal{M}(\mathcal{O}_m, y, \mathsf{RoE})$ can be modified to be an algorithm for $\mathcal{M}(\text{PROPH}_{m+1}, y, \mathsf{RoE})$ that achieves a competitive ratio that is at least as good.

1.2.3. Bounding the performance of the oracle model

After establishing the relationship between our oracle model and the TOP-1-OF-m model, we turn our attention to upper and lower bounds for the oracle model. First, for the non-IID setting and the RoE objective, we present an extremely simple single-threshold algorithm achieving a competitive ratio that approaches 1 exponentially in m. Even though our algorithm is for the oracle model, for which weaker guarantees are expected due to Theorem 2.8, it improves upon the best-known guarantee for the TOP-1-OF-m setting, due to Harb [Har24]. Our algorithm relies on two techniques; sharding and Poissonization, introduced by [Har24] for the analysis of threshold-based algorithms for prophet inequalities. As an added benefit, the algorithm's analysis is easy to understand.

Specifically, in Theorem 3.11, we show that there is a constant ξ_m, such that for the oracle model $\mathcal{M}(\mathcal{O}_m, \mathsf{non\text{-}IID}, \mathsf{RoE})$, there exists an algorithm with competitive ratio at least $1 - e^{-\xi_m}$. As $m \to +\infty$, this behaves as $1 - e^{-m/e + o(m)}$. The competitive ratio plot for $m = 1, \ldots 15$ is shown in Figure 1.1.

Matching upper bound. In addition, we provide a construction for every m that gives a matching upper bound to the competitive ratio, thus resolving the problem for the case of general distributions and the RoE objective. The construction we have is perhaps of independent interest in the design of counterexamples for other settings, as it combines and generalizes standard counterexamples of prophet inequalities.

Specifically, in Theorem 3.14, we show that for every $\delta > 0$, there exists an instance of $\mathcal{M}(\mathcal{O}_m, \mathsf{non\text{-}IID}, z)$, where $z = \mathsf{RoE}$ or $\mathbb{P}_{\max}$, in which no single-threshold algorithm can achieve a $\left(1 - e^{-\xi_m} + \delta\right)$-competitive ratio or select the maximum realization with probability $\geq \left(1 - e^{-\xi_m} + \delta\right)$.

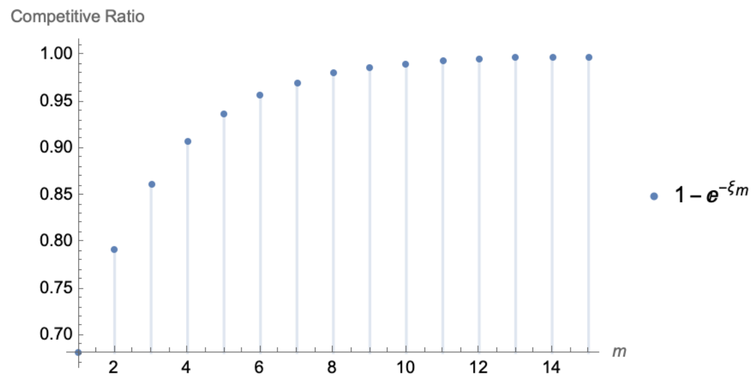

Figure 1.1: The value of $1 - e^{-\xi_m}$ for $m = 1, \dots, 15$.

Model	Lower Bound		Upper Bound	
	Prev. Best	Current Best	Prev.	Current Best
RoE General Settings	$1 - \mathcal{O}\big(e^{-m/6}\big)$ [EFN18]	$1 - e^{-m/e + o(m)}$	-	$1 - e^{-m/e + o(m)}$ single-threshold
$\mathbb{P}_{\max}$, IID Setting	≈ 0.58 [GM66]	≈ 0.797 $(m = 1)$ $1 - \mathcal{O}\big(m^{-m/5}\big)$	-	$1 - \mathcal{O}(m^{-m})$

Figure 1.2: State of the art.

Intuitively, the above follows since an algorithm for the oracle model performs poorly when, every time it uses an oracle call and gets a YES answer, the next value it sees that is at least the queried value is roughly equal, and thus the oracle call was used without any real gain. The idea behind the worst-case for this setting is to have what is essentially a Poisson random variable with rate ξ_m, providing the algorithm with several non-zero values, each roughly the same. By carefully selecting ξ_m in order to equate the probability of having no non-zero values and the probability of having more than m non-zero values, we are forcing the algorithm to use a query for every non-zero realization, thus rendering the oracle calls as useless as possible.

1.2.4. The benefit of several oracle calls

Next, we turn our attention to the IID setting with m oracles calls and the $\mathbb{P}_{\max}$ objective. We present a simple, single-threshold algorithm that selects the maximum realization with probability that approaches 1 in a super-exponential fashion. As a warm-up, we first present the analysis for $m = 1$ before generalizing it to all m.

Specifically, in Theorem 4.2, we show that for $\mathcal{M}(\mathcal{O}_m, \mathsf{IID}, \mathbb{P}_{\max})$, one can select the maximum realization with probability at least $1 - \mathcal{O}\big(m^{-m/5}\big)$.

We also present, in Theorem 4.3, an upper bound on the probability of success that is asymptotically tight, up to small multiplicative constants in the exponent. Because of Theorem 2.3, both upper and lower bounds on the probability of success carry over in the TOP-1-OF-m settings as well.

See Figure 1.2 for a summary of our results for the oracle model in the different settings.

1.3. Additional related work

We have already mentioned the related work on algorithms with predictions, as well as the works of Gilbert and Mosteller [GM66], Esfandiari, Hajiaghayi, Lucier and Mitzenmacher [EHLM17] and Nuti [Nut22] for the $\mathbb{P}_{\max}$ objective. Related work includes the study of order-aware algorithms by Ezra, Feldman *et al.* [EFGT23], algorithms with fairness guarantees by Correa *et al.* [CCDN21] and algorithms with a-priori information of some of the values by Correa *et al.* [BCGL22]. In addition to these, Esfandiari *et al.* [EHLM17] study a related but distinct variant to ours. They relax the objective to allow the return of one out of the top k realizations, and show exponential upper and lower bounds. Their model, however, is incomparable to ours.

Organization. In Section 2 we relate our model to TOP-1-OF-m model of Assaf and Samuel-Cahn [AS00] and prove the reductions. In Section 3 we present our tight algorithm for the non-IID setting. Section 4 contains our algorithms and upper bounds for the IID setting. Due to space constraints, we present some background on concentration inequalities that we use for our results in Appendix A. Finally, we present .

2. Reductions

To motivate our oracle model, we start by establishing an equivalence between $\mathcal{M}(\mathcal{O}_m, y, \mathbb{P}_{\max})$ and $\mathcal{M}(\text{PROPH}_{m+1}, y, \mathbb{P}_{\max})$, for both the $y =$ IID and $y =$ non-IID case (see Theorem 2.3 below). We also show that, perhaps surprisingly, this equivalence does not hold for the RoE objective; lower bound guarantees for $\mathcal{M}(\mathcal{O}_m, y, \text{RoE})$ translate to guarantees for $\mathcal{M}(\text{PROPH}_{m+1}, y, \text{RoE})$ (Theorem 2.8), but not the converse. Later, we use this result to improve the best-known lower bound guarantees on $\mathcal{M}(\text{PROPH}_{m+1}, y, \text{RoE})$.

2.1. The $\mathbb{P}_{\max}$ objective

Lemma 2.1. *Fix an instance of the prophet problem. Let $\mathcal{A}$ be an algorithm for this instance in $\mathcal{M}(\mathcal{O}_m, y, \mathbb{P}_{\max})$, where $y =$ IID or non-IID. Then, there exists an algorithm $\mathcal{B}$ for tor this instance in $\mathcal{M}(\text{PROPH}_{m+1}, y, \mathbb{P}_{\max})$, with black-box access to $\mathcal{A}$, such that $\mathbb{P}_{\max}(\mathcal{B}) \geq \mathbb{P}_{\max}(\mathcal{A})$.*

Proof: The idea is for $\mathcal{B}$ to simulate $\mathcal{A}$'s behavior by selecting each realization that $\mathcal{A}$ decides to query. Initially, $\mathcal{B}$ starts with an empty set S of selected values. Whenever $\mathcal{B}$ is presented with a realization X_i, it feeds it to $\mathcal{A}$. If $\mathcal{A}$ decides to select X_i or expend a query for X_i, regardless of the outcome of the query, $\mathcal{B}$ always selects X_i into S, otherwise $\mathcal{B}$ decides not to select X_i. By induction, S contains exactly all the realizations that were queried by $\mathcal{A}$ as well as at most one more realization that might have been selected by $\mathcal{A}$ if it run out of queries. Therefore, $|S| \leq m+1$.

Observe that $\mathcal{A}$ succeeds if and only if it selects the maximum, and it only selects a realization X_i if (i) it chose to expend a query on X_i, or (ii) when it observed X_i it run out of queries. In both cases, by the description of $\mathcal{B}$, we know that $X_i \in S$, and thus the probability that $\mathcal{B}$ succeeds is at least $\mathbb{P}_{\max}(\mathcal{A})$. ■

Lemma 2.2. *Fix an input instance of the prophet problem. Fix an algorithm $\mathcal{B}$ for $\mathcal{M}(\text{PROPH}_{m+1}, y, \mathbb{P}_{\max})$, where $y =$ IID or non-IID. Then, there exists an algorithm $\mathcal{A}$ for $\mathcal{M}(\mathcal{O}_m, y, \mathbb{P}_{\max})$, with black-box access to $\mathcal{B}$, such that such that $\mathbb{P}_{\max}(\mathcal{A}) \geq \mathbb{P}_{\max}(\mathcal{B})$.*

Proof: The idea is that $\mathcal{A}$ can simulate $\mathcal{B}$'s behavior using the oracle queries instead of storing the values like $\mathcal{B}$ does. Initially, $\mathcal{B}$ starts with an empty set S of selected values. Whenever $\mathcal{A}$ is presented with a realization X_i, it feeds it to $\mathcal{B}$. If $\mathcal{B}$ selects X_i into S, $\mathcal{A}$ performs an oracle $\mathcal{O}$ query whether $X_i > \max_{j=i+1}^{n} X_j$. Consider the first i where this happens. We distinguish between the two possible answers:

(I) If $\mathcal{O}$ answers YES, then we know that all future realizations are $\leq X_i$. However, we also know that since the objective is $\mathbb{P}_{\max}$, any optimal algorithm for $\textsc{Proph}_{m+1}$ will only select a value X_i if it is larger than any previously observed value (otherwise it "wastes" a spot in S for a value that is definitely not the maximum). Therefore, if $\mathcal{B}$ selects X_i, we know that $X_i > \max_{j<i} X_j$. In this case, both $\mathcal{B}$ and $\mathcal{A}$ succeed in selecting the maximum realization.

(II) If $\mathcal{O}$ answers NO, then we know that there exists a future realization that is $\geq X_i$. In this case, the instance for $\mathcal{B}$ reduces to $\mathcal{M}(\textsc{Proph}_m, y, \mathbb{P}_{\max})$ on $X_{i+1}, \ldots, X_n$, whereas the instance for $\mathcal{A}$ reduces to $\mathcal{M}(\mathcal{O}_{m-1}, y, \mathbb{P}_{\max})$. Since we know that $\mathcal{M}(\textsc{Proph}_1, y, \mathbb{P}_{\max}) = \mathcal{M}(\mathcal{O}_0, y, \mathbb{P}_{\max})$ by definition, we have that by induction, the probability that $\mathcal{A}$ succeeds is at least $\mathbb{P}_{\max}(\mathcal{B})$. ■

Combining the above two lemmas, we get the following result.

Theorem 2.3. *The $\mathcal{M}(\mathcal{O}_m, y, \mathbb{P}_{\max})$ model is equivalent to the $\mathcal{M}(\textsc{Proph}_{m+1}, y, \mathbb{P}_{\max})$ model, where $y =$ IID or non-IID. In other words, for every prophet inequality instance, the probability achieved by the best-possible algorithm in the $\mathcal{M}(\mathcal{O}_m, y, \mathbb{P}_{\max})$ model is the same as the one achieved by the best-possible algorithm in the $\mathcal{M}(\textsc{Proph}_{m+1}, y, \mathbb{P}_{\max})$ model.*

2.2. For the RoE objective $\mathcal{O}_m \leq \textbf{Proph}_{m+1}$

We demonstrate that the $\textsc{Proph}_m$ model strictly surpasses the $\mathcal{O}_m$ for non-IID random variables.

Definition 2.4. For two integers $i \leq j$, let $[\![i : j]\!] = \{i, i+1, \ldots, j\}$.

Definition 2.5. For an instance $\mathcal{I}$ of $\mathcal{M}(x, \text{non-IID}, \text{RoE})$, we denote $\text{RoE}(x, \mathcal{I})$ as the competitive ratio of an optimal algorithm for $\mathcal{I}$. For example, $\text{RoE}(\mathcal{O}_m, \mathcal{I})$ denotes the best competitive ratio on instance $\mathcal{I}$ in the oracle model.

Lemma 2.6. *For $m = 1$, there exists an input instance $\mathcal{I}$, made out of 3 non-IID random variables, such that $\text{RoE}(\mathcal{O}_1, \mathcal{I}) \leq \frac{3}{4}\text{RoE}(\textsc{Proph}_2, \mathcal{I})$.*

Proof: Consider the input instance $\mathcal{I}$ of three independent random variables X_1, X_2, X_3 with default value 0, such that

$$X_1 = 1, \qquad \mathbb{P}[X_2 = 1 + \varepsilon] = \frac{1}{2} - \varepsilon, \qquad \text{and} \qquad \mathbb{P}\left[X_3 = \frac{1}{\varepsilon}\right] = \varepsilon.$$

We have that

$$\mathbb{E}\Big[\max\{X_1, X_2, X_3\}\Big] = \frac{1}{\varepsilon}\varepsilon + (1+\varepsilon)(1-\varepsilon)\left(\frac{1}{2} - \varepsilon\right) + 1(1-\varepsilon)\left(\frac{1}{2} + \varepsilon\right) = 2 - O(\varepsilon).$$

For small ε, an algorithm $\mathcal{B}$ that is optimal for the $\textsc{Proph}_2$ model in this instance is to select X_1, ignore X_2 and then select X_3 if it is non-zero. This yields

$$\mathbb{E}[\mathcal{B}] = 1 \cdot (1-\varepsilon) + \frac{1}{\varepsilon} \cdot \varepsilon = 2 - \varepsilon.$$

However, the optimal $\mathcal{A}$ for the oracle model queries $\mathcal{O}$ at X_1. With probability $(1-\varepsilon)(1/2+\varepsilon)$, it stops and select X_1, getting a value of 1. Otherwise, it continues, with no oracle calls left. It ignores X_2 and select X_3. Thus,

$$\mathbb{E}[\mathcal{A}] = 1 \cdot \left(\frac{1}{2} + \varepsilon\right)(1-\varepsilon) + \frac{1}{\varepsilon} \cdot \varepsilon = \frac{3}{2} + \frac{\varepsilon}{2} - \varepsilon^2.$$

The competitive ratios of $\mathcal{A}$ is $\mathsf{RoE}(\mathcal{O}_1, \mathcal{I}) = \dfrac{\frac{3}{2} + \frac{\varepsilon}{2} - \varepsilon^2}{2 - O(\varepsilon)} = \dfrac{3}{4} + O(\varepsilon) \to \dfrac{3}{4}$ (the limit is for $\varepsilon \to 0$). The competitive ratio of $\mathcal{B}$ is

$$\mathsf{RoE}(\textsc{Proph}_2, \mathcal{I}) = \frac{2-\varepsilon}{2+O(\varepsilon)} = 1 - O(\varepsilon) \to 1.$$

■

The above example, appropriately generalized for $m > 1$ by having random variables

$$X_1 = 1 \quad \text{w.p. } 1, \quad X_i = \begin{cases} 1 + (i-1)\varepsilon & \text{w.p. } \frac{1}{2} - \varepsilon \\ 0 & \text{w.p. } \frac{1}{2} + \varepsilon \end{cases}, \quad \text{for } i = 2, \ldots, m+1, \text{ and}$$

$$X_{m+2} = \begin{cases} \frac{1}{\varepsilon} & \text{w.p. } \varepsilon \\ 0 & \text{w.p. } 1 - \varepsilon \end{cases},$$

shows that the gap between $\mathsf{RoE}(\mathcal{O}_m, \mathcal{I}))$ and $\mathsf{RoE}(\textsc{Proph}_{m+1}, \mathcal{I}))$ is at most $1 - 1/2^{m+1}$ for general m. The analysis of this example for general m is similar to the $m = 1$ case. We do not present it here as, even though this example is very simple, this gap is not the tightest one possible. For a tighter gap between the competitive ratio of the two models, see the example in the proof of Theorem 3.14.

Lemma 2.7. *For any input instance $\mathcal{I}$, we have $\mathsf{RoE}(\textsc{Proph}_{m+1}, \mathcal{I}) \geq \mathsf{RoE}(\mathcal{O}_m, \mathcal{I})$, for IID or non-IID variables.*

Proof: Let $\mathcal{A}$ be the algorithm in $\mathcal{M}(\mathcal{O}_m, \mathsf{RoE}, \mathcal{I})$ realizing the maximum RoE for $\mathcal{I}$. We construct an algorithm $\mathcal{B} \in \mathcal{M}(\mathcal{O}_m, \mathsf{RoE}, \mathcal{I})$.

The algorithm $\mathcal{B}$ simulates $\mathcal{A}$'s behavior by selecting each realization that $\mathcal{A}$ decides to query. Initially, $\mathcal{B}$ starts with an empty set S. Whenever $\mathcal{B}$ is presented with a realization X_i, it feeds it to $\mathcal{A}$. If $\mathcal{A}$ decides to return X_i, or performs an oracle query for X_i, the algorithm $\mathcal{B}$ adds X_i to S.

Observe that the algorithm $\mathcal{A}$ stops as soon as an oracle query returns NO. Thus, the simulation $\mathcal{B}$ of $\mathcal{A}$, assumes the oracle always answers YES (i.e., a larger value is coming up in the future). (i.e., the simulation replaces a call to the oracle by a function that always returns YES), as this enables it (potentially) to save more values into the available slots, thus increasing its RoE.

The set S contains exactly all the realizations that were queried by $\mathcal{A}$, as well as at most one additional realization returned by $\mathcal{A}$. Therefore, $|S| \leq m+1$.

Every possible sequence of realizations $\mathcal{A}$ queried (or selected to return) are in S. Therefore, if $V_{\mathcal{A}}$ is the value returned by $\mathcal{A}$ and $V_{\mathcal{B}}$ is the value returned by $\mathcal{B}$, we have $V_{\mathcal{B}} \geq V_{\mathcal{A}}$, which readily implies that $\mathsf{RoE}(\mathcal{B}) \geq \mathsf{RoE}(\mathcal{A})$. ■

Theorem 2.8. *For every $m \geq 1$, and for* all *input instances $\mathcal{J}$ (of IID or non-IID variables), we have $\textsf{RoE}(\mathcal{O}_m, \mathcal{J}) \leq \textsf{RoE}(\textsc{ProphH}_{m+1}, \mathcal{J})$, Furthermore, there* exists *an input instance $\mathcal{I}$, made out of $m+2$ non-IID random variables, such that $\textsf{RoE}(\mathcal{O}_m, \mathcal{I}) \leq (1 - 1/2^{m+1})\textsf{RoE}(\textsc{ProphH}_2, \mathcal{I})$.*

3. The non-IID settings

By Theorem 2.8, any guarantees we provide for the oracle model with the RoE objective can be directly translated to guarantees for the $\textsc{Top-1-of-}m$ model, improving upon the previous work on this model [AS00, AGS02, EFN18, Har24]. We provide a simple, single-threshold algorithm that resolves the RoE objective in the oracle model.

3.1. The exponent sequence

Definition 3.1. For every $m \geq 1$, let ξ_m denote the unique *positive* solution to the following equation:

$$1 - e^{-\xi_m} = \frac{\Gamma(m+1, \xi_m)}{m!},$$

where $\Gamma(m+1, x) = \int_{t=x}^{\infty} t^m e^{-t}\, dt$ denotes the *upper incomplete gamma* function. The ***exponent sequence*** is $\xi_1, \xi_2, \ldots$.

We show below that the optimal competitive ratio of $\mathcal{M}(\mathcal{O}_m, \textsf{non-IID}, \textsf{RoE})$ is exactly $1 - e^{-\xi_m}$. It is known that, for $x \geq 0$ and an integer $m+1 > 0$, we have

$$\Gamma(m+1, x) = m!\, e^{-x} \sum_{k=0}^{m} \frac{x^k}{k!} \leq m! e^{-x} e^{x} \leq m!. \tag{3.1}$$

As such, the above equation on the value of ξ_m, becomes

$$1 - e^{-\xi_m} = e^{-\xi_m} \sum_{k=0}^{m} \frac{(\xi_m)^k}{k!} \qquad \Longleftrightarrow \qquad \sum_{k=m+1}^{\infty} \frac{(\xi_m)^k}{k!} = 1.$$

This readily implies that the exponent sequence is monotonically increasing, and $m/e^2 \leq \xi_m \leq m$.

Definition 3.2. Let $\mathsf{q}_{k+1}(x) = \frac{\Gamma(k+1,x)}{k!} = e^{-x} \sum_{j=0}^{k} \frac{x^j}{j!}$. This implies $\mathsf{q}_{m+1}(\xi_m) = 1 - e^{-\xi_m}$.

Lemma 3.3. $\mathsf{q}'_{m+1}(x) = -e^{-x} \frac{x^m}{m!}$.

Proof: As $(e^{-x})' = -e^{-x}$, we have $\mathsf{q}'_{m+1}(x) = -e^{-x} + \sum_{j=1}^{m} \left(e^{-x} \frac{x^{j-1}}{(j-1)!} - e^{-x} \frac{x^j}{j!} \right) = -e^{-x} + e^{-x} - e^{-x} \frac{x^m}{m!} = -e^{-x} \frac{x^m}{m!}$. ■

Lemma 3.4. *For all $m \geq 1$, we have $(m!)^{1/m} < \xi_m < ((m+1)!)^{1/m+1}$.*

Proof: Guided by Eq. (3.1), define

$$h(x) = \frac{\Gamma(m+1, x)}{m!} - 1 + e^{-x} = \mathsf{q}_{m+1}(x) - 1 + e^{-x} = e^{-x} \left(\sum_{i=0}^{m} \frac{x^i}{i!} - e^x + 1 \right) = e^{-x}(1 - T(x)),$$

where $T(x) = \sum_{i=m+1}^{\infty} \frac{x^i}{i!}$. By Lemma 3.3, we have $h'(x) = -e^{-x}\frac{x^m}{m!} - e^{-x} < 0$. Namely, $h(\cdot)$ is a strictly decreasing function. Thus, ξ_m the positive root of $h(x) = 0$ is unique, as $h(0) = 1$, and $\lim_{x\to\infty} h(x) = -1$.

Setting $\beta = ((m+1)!)^{1/(m+1)}$, we have $T(\beta) > \frac{\beta^{m+1}}{(m+1)!} = \frac{(m+1)!}{(m+1)!} = 1$, which readily implies $h(\beta) < 0$. By the AM-GM inequality, we have that $\gamma = \sqrt[m]{m!} < \sum_{i=1}^{m} i/m = \frac{m+1}{2}$. In particular, we have

$$\frac{\gamma^{m+1}}{(m+1)!} = \frac{m! \cdot \gamma}{(m+1)!} = \frac{\gamma}{m+1} < \frac{1}{2}.$$

As such, we have

$$T(\gamma) \leq \sum_{i=m+1}^{\infty} \frac{\gamma^i}{i!} \leq \sum_{i=m+1}^{\infty} \frac{\gamma^{m+1}\gamma^{i-m-1}}{(m+1)!(m+1)^{i-m-1}} < \frac{1}{2}\sum_{j=0}^{\infty} \frac{\gamma^j}{(m+1)^j} < \frac{1}{2}\sum_{j=0}^{\infty} \frac{((m+1)/2)^j}{(m+1)^j} = 1.$$

Thus, $h(\gamma) > 0$. We conclude that $\gamma < \xi_m < \beta$. ■

Remark 3.5. Setting $\nu(x) = \nu(m, x) = \frac{\Gamma(m+1,x)}{m!}$, and arguing as in Lemma 3.4, we have $\nu'(x) < 0$, which readily implies that $\nu(x)$ is monotonically decreasing.

Stirling's formula applied to Lemma 3.4 readily implies the following.

Lemma 3.6. *We have* $\lim_{m\to\infty} \frac{\xi_m}{m} = \frac{1}{e}$.

Lemma 3.7. *For all $k, m \geq 0$ integers, we have* $f(k, m) = \sum_{j=1}^{k} \frac{\xi_m^j}{j!} - \sum_{j=m+1}^{m+k} \frac{\xi_m^j}{j!} \geq 0$.

Proof: By definition $f(0, m) = 0$. We have

$$f(k+1, m) - f(k, m) = \frac{\xi_m^{k+1}}{(k+1)!} - \frac{\xi_m^{m+k+1}}{(m+k+1)!}.$$

Thus $f(k+1, m) \geq f(k, m) \iff {}^{(m+k+1)!}/_{(k+1)!} \geq {}^{\xi_m^{m+k+1}}/_{\xi_m^{k+1}} = \xi_m^m$. for $k > 0$ and $m > 0$, we have

$$(m+k+1)! = (m+1)! \cdot 1 \cdot \underbrace{(m+2)}_{>2} \cdot \underbrace{(m+3)}_{>3} \cdots \underbrace{(m+k+1)}_{>k+1} > (m+1)!(k+1)!.$$

Thus, it sufficient to prove that $\xi_m^m < (m+1)! \iff \xi_m < \sqrt[m]{(m+1)!}$. The later is immediate from Lemma 3.4, as $\xi_m < \big((m+1)!\big)^{1/(m+1)} < \sqrt[m]{(m+1)!}$. ■

3.2. Background: Sharding, poissonization, and stochastic dominance

For a sequence of random variables $\mathbf{X} = X_1, \ldots, X_n$, let $|\alpha \leq \mathbf{X} \leq \beta| = |\{i \mid \alpha \leq X_i \leq \beta\}|$ denote the number of realizations in this sequence falling in the interval $[\alpha, \beta]$.

3.2.1. Sharding

For the lower bound, we use poissonization and sharding [Har24]. Given random variables $X_1, \ldots, X_n$ with cdfs $F_1, \ldots, F_n$, instead of sampling X_i from F_i, we instead replace it with a sequence of K independent random variables $\mathbf{H}_i = Y_{i,1}, \ldots, Y_{i,K}$, such that $\max_j Y_{i,j}$ has the same distribution as X_i. Specifically, the cdf of $Y_{i,j}$, for all j, is $F_i^{1/K}$. Thus, the distribution of $\max\{Y_{i,1}, \ldots, Y_{i,K}\}$ is the same as X_i. This creates a new sequence of Kn samples $\mathbf{S} = \mathbf{H}_1 \cdot \mathbf{H}_2 \cdots \mathbf{H}_n$, where $\cdot$ is the concatenation operator. Observe that for any $\alpha \geq 0$ and integer t, we have

$$\mathbb{P}[|\mathbf{X} \geq \alpha| > t] < \mathbb{P}[|\mathbf{S} \geq \alpha| > t].$$

This implies, that for threshold algorithms, running on $\mathbf{S}$ instead of $\mathbf{X}$ can only generate worst results. We emphasize that this sharding is done only for analysis purposes.

3.2.2. Poissonization

Definition 3.8. A random variable X has Poisson distribution with rate λ, denoted by $X \sim \mathrm{Pois}(\lambda)$, if $\mathbb{P}[X = i] = \lambda^k e^{-\lambda}/k!$. Conveniently, $\mathbb{E}[X] = \mathbb{V}[X] = \lambda$.

The purpose of the sharding is to be able to bound quantities of the form $\mathbb{P}[|\beta \leq \mathbf{S} \leq \tau| = t]$. As K grows, the underlying random variable $|\beta \leq \mathbf{S} \leq \tau|$ has a binomial distribution that converges to a Poisson distribution.

Observation 3.9. *For $c \in (0, 1]$, we have, using L'Hôpital's rule, that* $\lim_{x\to\infty} x(1 - c^{1/x}) = \lim_{x\to\infty} \frac{1-\exp(\log(c)/x)}{1/x} = \lim_{x\to\infty} \frac{\log(c)\exp(\log(c)/x)/x^2}{-1/x^2} = -\log c$, *where* $\log = \log_e$.

Let τ be a threshold such that $\sum_{i=1}^{n} \sum_{j=1}^{K} \mathbb{P}[Y_{i,j} \geq \tau] = c$ for some constant c to be determined shortly. We can rewrite this into the following.

$$\sum_{i=1}^{n} K\Big(1 - \mathbb{P}[X_i \leq \tau]^{1/K}\Big) = c. \tag{3.2}$$

The limit of Eq. (3.2), as $K \to +\infty$, is $\sum_{i=1}^{n} -\log \mathbb{P}[X_i \leq \tau] = c$, by Observation 3.9. Equivalently, for $Z = \max\{X_1, \ldots, X_n\}$, we have

$$e^{-c} = \exp\Big(\sum_{i=1}^{n} \log \mathbb{P}[X_i \leq \tau]\Big) = \prod_{i=1}^{n} \mathbb{P}[X_i \leq \tau] = \mathbb{P}[X_1, \ldots, X_n \leq \tau] = \mathbb{P}[Z \leq \tau].$$

In particular, the distribution of the number of indices j, such that $Y_{i,j} \geq \tau$ can be well approximated with a Poisson distribution. Specifically, let $V_{i,j} = 1 \iff Y_{i,j} \geq \tau$, and consider the sum $V_i = \sum_{j=1}^{K} V_{i,j}$. The variable $V_i \sim \mathrm{bin}(K, \psi_i)$, where $\psi_i = 1 - \mathbb{P}[X_i \leq \tau]^{1/K}$.

Let $\lambda_i = \psi_i K$, and consider the random variable $U_i \sim \mathrm{Pois}(\lambda_i)$ (i.e., U_i has a Poisson distribution with rate λ_i). Intuitively, V_i and U_i have similar distributions. Formally, Le Cam theorem implies that for any set $T \subseteq \{0, 1, \ldots, K\}$, we have $|\mathbb{P}[V_i \in T] - \mathbb{P}[U_i \in T]| \leq 2K\psi_i^2 = 2\lambda_i^2/K \leq 2c^2/K$, by Eq. (3.2). The later quantity goes to zero as K increases.

Thus, we get a variable U_i with a Poisson distribution for each shard sequence $\mathbf{H}_i$, with rate λ_i, where U_i models the number of times we encounter in $\mathbf{H}_i$ values larger than τ. Thus, $U_\tau = \sum_i U_i$ models the total number of times in the splintered sequence $\mathbf{S}$ that values encountered are larger than τ. The variable U_τ has a Poisson distribution with rate $\lambda_\tau = \sum_{i=1}^{n} \lambda_i$.

3.2.3. The distribution in a range

Repeating the same process with a bigger threshold $\beta > \tau$, would yield a similar Poisson random variable U_β with a *lower* rate λ_β. The quantity $\Delta = U_\tau - U_\beta$ is the number of values in $\mathbf{S}$ in the range $[\tau, \beta]$. Furthermore, Δ has a Poisson distribution with rate $\lambda_\tau - \lambda_\beta$. Specifically, $\mathbb{P}[|\beta \leq \mathbf{S} \leq \tau| = t] = \mathbb{P}[\Delta = t]$.

The key to our analysis is that the variables Δ and U_β are independent (in the limit as K increases).

3.2.4. Stochastic dominance

A standard observation is that for a non-negative random variable X, we have $\mathbb{E}[X] = \int_{x=0}^{\infty} \mathbb{P}[X \geq x] \, \mathrm{d}x$. Thus, for $Z = \max\{X_1, \ldots, X_n\}$, and for an algorithm $\mathcal{A}$, if one can guarantee that there is $c \in [0,1]$, such that for all $\nu \geq 0$, $\mathbb{P}[\mathcal{A} \geq \nu] \geq c\,\mathbb{P}[Z \geq \nu]$, then

$$\mathbb{E}[\mathcal{A}] = \int_0^\infty \mathbb{P}[\mathcal{A} \geq x] \, \mathrm{d}x \geq c \int_0^\infty \mathbb{P}[Z \geq x] \, \mathrm{d}x \geq c\,\mathbb{E}[Z] \,.$$

And hence c is a lower bound on the competitive ratio of $\mathcal{A}$. This argument is used in several results on prophet inequalities and is often referred to as *majorizing* $\mathcal{A}$ with Z.

3.3. An optimal single-threshold algorithm (lower bound)

Here, we describe a single-threshold algorithm that achieves the optimal competitive ratio in the oracle model.

Definition 3.10. A *single threshold* algorithm sets a threshold τ, and start reading the sequence. Whenever encountering a realization $> \tau$, the algorithm stops and consult with the oracle. The oracle query is whether all the values remaining in the suffix of the sequence are of value $\leq \tau$. If the oracle returns YES, the algorithm accepts the current value and stops. Otherwise it raises its threshold to $\tau = X_i$ and continues. If the oracle runs out of oracle calls, it returns the first value encountered after the last oracle call that is bigger than τ (which exists, since all oracle calls returned NO).

While technically, the querying threshold of the algorithm might change during its execution, we call the algorithm a single-threshold algorithm since it uses a single-threshold to decide whether to query the oracle or not, and this threshold *does not change with* i, unlike for example the optimal DP for the IID prophet inequality or the prophet secretary model. Our oracle model is quite different than most other prophet inequality models in the sense that the algorithm has some knowledge of the (true) future. Of course, any algorithm that knows that the maximum of $X_{i+1}, \ldots, X_n$ is larger than X_i would be wasting queries if it expended them on some $X_j < X_i$ for $j > i$, and thus the spirit of it being a single-threshold algorithm to decide whether to query the oracle or not remains.

Theorem 3.11. *Let $\alpha = 1 - e^{-\xi_m} = 1 - e^{-m/e + o(m)}$, see Definition 3.1. For any finite sequence $\mathbf{X}$ of non-IID variables, one can compute a value τ, such that the single-threshold algorithm (with initial threshold τ) has competitive ratio $\geq \alpha$. i.e., the competitive ratio of $\mathcal{M}(\mathcal{O}_m, \textsf{non-IID}, \textsf{RoE})$ is $\geq \alpha$.*

Proof: Let $\mathbf{X} = X_1, \ldots, X_n$, and $Z = \max_i X_i$. The threshold τ is the $e^{-\xi_m}$ quantile of the maximum, i.e. $\mathbb{P}[Z \leq \tau] = e^{-\xi_m}$. We use $\mathcal{A}(\mathbf{X})$ to denote the result of running the algorithm on $\mathbf{X}$.

As suggested in Section 3.2.1 (for the analysis), we imagine running the algorithm on the splintered sequence $\mathbf{S}$. Some counterintuitively, imagine first generating $\mathbf{S}$, and computing $X_i = \max_j Y_{i,j}$, see Section 3.2.1. Thus, $\max \mathbf{S} = \max \mathbf{X}$. For the sequence $\mathbf{S}$, let $\mathbf{S}_{\geq \tau}$ denote the subsequence of elements of $\mathbf{S}$ that their values are above τ. Observe that $\mathbf{X}_{\geq \tau}$ is a subsequence of $\mathbf{S}_{\geq \tau}$. Thus, we analyze the algorithm performance on $\mathbf{S}$.

Let $\beta \in [0, \tau]$. The probability the algorithm selects a value above β is equal to the probability it selects any value. Thus,

$$\mathbb{P}[\mathcal{A}(\mathbf{X}) \geq \beta] = \mathbb{P}[\mathcal{A} \geq \tau] = \mathbb{P}[Z \geq \tau] = 1 - e^{-\xi_m} \geq \left(1 - e^{-\xi_m}\right) \mathbb{P}[Z \geq \beta]. \tag{3.3}$$

For $\beta \in [\tau, +\infty)$, let $\mathbb{P}[Z \leq \beta] = e^{-q} > e^{-\xi_m}$, implying $\mathbb{P}[Z \geq \beta] = 1 - e^{-q}$. By sharding and Poissonization, the number of shards in the range $[\tau, \beta]$ (resp. $\geq \beta$) is a Poisson random variable Δ (resp. U_β) with rate $\xi_m - q$ (resp. q), see Section 3.2.3. Critically, U_β and Δ are independent. Consider the event of there being at most m values in the range $[\tau, \beta]$, and there being at least one value in $[\beta, +\infty)$. The value $\mathcal{A}(\mathbf{X}) \geq \beta$ in that case. Hence, by the independence of Δ and U_β, we have

$$\frac{\mathbb{P}[\mathcal{A}(\mathbf{X}) \geq \beta]}{\mathbb{P}[Z \geq \beta]} \geq \frac{\mathbb{P}[(U_\beta \geq 1) \cap (0 \leq \Delta \leq m)]}{\mathbb{P}[Z \geq \beta]} = \frac{\mathbb{P}[U_\beta \geq 1]}{\mathbb{P}[Z \geq \beta]} \mathbb{P}[0 \leq \Delta \leq m] = \mathbb{P}[0 \leq \Delta \leq m].$$

Now, we have

$$\mathbb{P}[0 \leq \Delta \leq m] = \sum_{i=0}^{m} e^{-(\xi_m - q)} \frac{(\xi_m - q)^i}{i!} = \frac{\Gamma(m+1, \xi_m - q)}{m!} \geq \frac{\Gamma(m+1, \xi_m)}{m!} = 1 - e^{-\xi_m}.$$

by Eq. (3.1), Remark 3.5 and Definition 3.1.

The above implies that, for any $\beta \geq 0$, we have $\mathbb{P}[\mathcal{A}(\mathbf{X}) \geq \beta] \geq (1 - e^{-\xi_m}) \mathbb{P}[Z \geq \beta]$, Namely, $\mathsf{RoE}(\mathcal{A}) \geq 1 - e^{-\xi_m}$. ∎

3.4. A matching upper bound for single-threshold algorithms

To this end, we present an input sequence for which no algorithm can do better for the oracle that answers if $X_i > \max_{j=i+1}^{n} X_j$, and against an almighty adversary.

Input instance. The input instance $\mathcal{I}$ is a sequence made out of $n+2$ random variables, for n sufficiently large. Each of these random variables can have only two values – either zero or some positive value. Specifically, for $\varepsilon > 0$ sufficiently small (e.g., $\varepsilon \ll 1/n^4$), let

$$X_1 = 1, \quad \mathbb{P}[X_i = 1] = \frac{\xi_m}{n}, \quad \text{for} \quad i \in [\![2 : n+1]\!], \quad \text{and} \quad \mathbb{P}\big[X_{n+2} = \tfrac{1}{\varepsilon}\big] = \varepsilon.$$

By Lemma 3.6, $\xi_m \approx m/e$, as such, the expected number of non-zero entries in this sequence is (roughly) $m/e + 1$.

Lemma 3.12. *For $Z = \max_i X_i$, we have $\mathbb{E}[Z] = 2$ as $\varepsilon \to 0$.*

Proof: Let $Z' = \max_{i \in [\![n+1]\!]} X_i$. Observe that $Z' = 1$. As such, for $Z = \max(Z', X_{n+2})$, we have $\mathbb{E}[Z] = \mathbb{E}[\max_i X_i] = (1/\varepsilon)\varepsilon + (1-\varepsilon)\,\mathbb{E}[Z'] \xrightarrow[\varepsilon \to 0]{} 2$. ∎

Observation 3.13. *Let $\widehat{X}_i$ be an indicator variable for the event that $X_i = 1$. For sufficiently large n, $\nabla = \sum_{i=2}^{n+1} \widehat{X}_i$ has a binomial distribution that can be well approximated by a Poisson distribution (see Theorem A.2) with rate ξ_m. That is, $\lim_{n\to\infty} \mathbb{P}\big[\nabla = k\big] = e^{-\xi_m}\frac{(\xi_m)^k}{k!}$.*

Observe that $\lim_{n\to\infty} \mathbb{P}[\nabla \le k] = \sum_{i=0}^{k} e^{-\xi_m}\frac{(\xi_m)^i}{i!} = \mathsf{q}_{k+1}(\xi_m)$. For simplicity of exposition, we will assume $n \to \infty$ in the following analysis and thus $\mathbb{P}[\nabla \le k] = \mathsf{q}_{k+1}(\xi_m)$, see Definition 3.2.

Theorem 3.14. *Consider any choice of $m \ge 1$, and $\delta > 0$, and the above input instance $\mathcal{I}$ formed by a sequence of* non-IID *random variables. Then, for any algorithm, against the almighty adversary (see Definition 1.1), we have $\mathcal{A} \in \mathcal{M}(\mathcal{O}_m, \textsf{non-IID}, \textsf{RoE})$ for $\mathcal{I}$, we have $\textsf{RoE}(\mathcal{A}) \le 1 - e^{-\xi_m} + \delta$.*

Proof: First, we discuss the strategy that the almighty adversary adopts. The adversary first observes all values. Suppose k nonzero values show up from $X_2, ..., X_n$ at indices $U = \{i_1, ..., i_k\}$, and all other $n-k$ values from $X_2, \ldots, X_{n+1}$ at indices $B = \{\hat{i}_1, \ldots, \hat{i}_{n-k}\}$ are zero. The adversary provides the random variables in the order $X_{\sigma(1)}, \ldots, X_{\sigma(n+2)}$ where σ is defined as $\sigma(1) = 1$, $\sigma(j) = i_j, j = 2, \ldots, k+1$, $\sigma(j) = \hat{i}_j$ and finally $\sigma(n+2) = n+2$. In other words, the adversary stacks all the k non zero values from $X_2, \ldots, X_{n+1}$ starting from index 2 to index $k+1$.

Now we consider any algorithm for this setting. We strengthen the algorithm by telling it the almighty adversary's strategy; this can never reduce its expected reward. Hence, the algorithm knows it will see $X_{\sigma(1)} = X_1$, then a stream of k ones (where it does not know k), then $n-k$ zeros, and finally $X_{\sigma(n+2)} = X_{n+2}$. The algorithm has two initial decisions to make; either query at X_1 and continue (if the answer is NO) to $X_{\sigma(2)}, \ldots X_{\sigma(n+1)}$ with $m-1$ oracle calls, or it can just proceed to $X_{\sigma(2)}, \ldots X_{\sigma(n+1)}$ with m oracle calls. Thus, the only difference in the two cases is that in the former, we have only $m-1$ oracle calls for $X_{\sigma(2)}, \ldots, X_{\sigma(n+1)}$ but we get an expected reward of 1 if $X_{\sigma(2)} = \cdots = X_{\sigma(n+2)} = 0$, and in the later case, we get m oracle calls for $X_{\sigma(2)}, \ldots, X_{\sigma(n+1)}$, but we get 0 reward if $X_{\sigma(2)} = \cdots = X_{\sigma(n+2)} = 0$.

Let k be the number of non-zeros in $X_2, \ldots, X_{n+1}$ (i.e., $X_{\sigma(k+1)}$ is the last 1). When the algorithm starts reading the stream of 1s from $X_{\sigma(2)}, ..., X_{\sigma(n+1)}$, it needs to decide indices $S \subseteq [\![2 : n+1]\!], |S| \le m$ where it will expend the oracle call. Further, it is suboptimal to use the oracle at a 0 value, since regardless, the algorithm will receive a value of 0 in the end if it fails. Consider what happens if the algorithm decides to query at index $i \in [\![2 : n+1]\!]$ with $X_{\sigma(i)} = 1$. If $X_{\sigma(i+1)} = \ldots X_{\sigma(n+1)} = 0$, then the algorithm gets on expectation $1/\varepsilon \cdot \varepsilon + (1-\varepsilon) \cdot 1 \xrightarrow[\varepsilon\to 0]{} 2$ reward on expectation. However, if $X_{\sigma(i+1)} = 1$, then the oracle will return NO because $1 = X_{\sigma(i)} \not> \max(X_{\sigma(i+1)}, \ldots, X_{\sigma(n+2)})$. On the other hand, if the algorithm does not query at index $k+1$ (i.e., $(k+1) \notin S$), then the algorithm gets on expectation $\mathbb{E}\big[X_{\sigma(n+2)}\big] = \mathbb{E}[X_{n+2}] = 1/\varepsilon \cdot \varepsilon = 1$.

Hence, the crucial observation is that an algorithm starting at $X_{\sigma(2)}$ that uses its query calls at indices $S \subseteq [\![2 : n+1]\!]$ gets on expectation 2 if and only if $(k+1) \in S$, and 1 otherwise. Thus, for algorithm $\mathcal{A}_1$ that skips $X_{\sigma(1)}$ and uses its oracles at indices $S, |S| = m$, it satisfies

$$\begin{aligned}
\mathbb{E}[\mathcal{A}_1] &= 2 \cdot \sum_{i \geq 0, (i+1) \in S} e^{-\xi_m} \frac{\xi_m^i}{i!} + 1 \cdot \sum_{i \geq 0, (i+1) \notin S} e^{-\xi_m} \frac{\xi_m^i}{i!} \\
&= \sum_{i \geq 0} e^{-\xi_m} \frac{\xi_m^i}{i!} + \sum_{i \geq 0, (i+1) \in S} e^{-\xi_m} \frac{\xi_m^i}{i!} \\
&= 1 + \sum_{(i+1) \in S} e^{-\xi_m} \frac{\xi_m^i}{i!}
\end{aligned}$$

On the other hand, for algorithm $\mathcal{A}_2$ that uses its oracle at $X_{\sigma(1)}$ and uses its remaining oracles at indices $S' \in [\![2 : n+1]\!], |S'| = m - 1$, it gets an extra benefit of getting a reward with expected value 2 (as $\varepsilon \to 0$) if $X_{\sigma(2)} = \cdots = X_{\sigma(n+1)} = 0$. Hence, it satisfies

$$\begin{aligned}
\mathbb{E}[\mathcal{A}_2] &= \left(e^{-\xi_m} \cdot 2\right) + \left(2 \cdot \sum_{i \geq 0, (i+1) \in S'} e^{-\xi_m} \frac{\xi_m^i}{i!}\right) + \left(1 \cdot \sum_{i \geq 1, (i+1) \notin S'} e^{-\xi_m} \frac{\xi_m^i}{i!}\right) \\
&= \left(\sum_{i \geq 0} e^{-\xi_m} \frac{\xi_m^i}{i!}\right) + e^{-\xi_m} + \sum_{(i+1) \in S'} e^{-\xi_m} \frac{\xi_m^i}{i!} \\
&= 1 + e^{-\xi_m} + \sum_{(i+1) \in S'} e^{-\xi_m} \frac{\xi_m^i}{i!}.
\end{aligned}$$

First, we show that the expression $\sum_{(i+1) \in S} e^{-\xi_m} \frac{\xi_m^i}{i!}$ subject to $S \subseteq [\![2 : n+1]\!], |S| = m$ is maximized for $S^* = [\![2 : m+1]\!]$. Note that it is easy to verify that for a Poisson distribution with rate λ, its probability mass function $e^{-\lambda} \lambda^i / i!$ is increasing for $i < \lambda$, and decreasing after $i > \lambda$. Hence, the optimal $S^* = [\![k : k+m-1]\!]$ for some $k \geq 2$ that "covers" the rate ξ_m (this is the region with the most mass for a Poisson distribution). The optimal choice of k is $k = 2$ because

$$\sum_{i=1}^{m} e^{-\xi_m} \frac{\xi_m^i}{i!} - \sum_{i=k-1}^{k+m-2} e^{-\xi_m} \frac{\xi_m^i}{i!} = \sum_{i=1}^{k-2} e^{-\xi_m} \frac{\xi_m^i}{i!} - \sum_{i=m+1}^{m+k-2} e^{-\xi_m} \frac{\xi_m^i}{i!} \geq 0,$$

where the last inequality holds by Lemma 3.7. Similarly, $k = 2$ is optimal for when $|S| = m - 1$. Hence, we get the inequalities

$$\begin{aligned}
\mathbb{E}[\mathcal{A}_1] &\leq 1 + \sum_{i=1}^{m} e^{-\xi_m} \frac{\xi_m^i}{i!} = 1 + \mathsf{q}_{m+1}(\xi_m) - e^{-\xi_m}, \\
\mathbb{E}[\mathcal{A}_2] &\leq 1 + e^{-\xi_m} + \sum_{i=1}^{m-1} e^{-\xi_m} \frac{\xi_m^i}{i!} = 1 + \mathsf{q}_m(\xi_m).
\end{aligned}$$

Thus, we have

$$\max(\mathbb{E}[\mathcal{A}_1], \mathbb{E}[\mathcal{A}_2]) \leq 1 - e^{-\xi_m} + \mathsf{q}_m(\xi_m) + e^{-\xi_m} \max\{1, \frac{\xi_m^m}{m!}\}$$

But recall from Lemma 3.4 that $\xi_m^m \geq m!$, thus

$$\begin{aligned}\max(\mathbb{E}[\mathcal{A}_1], \mathbb{E}[\mathcal{A}_2]) &\leq 1 - e^{-\xi_m} + \mathsf{q}_m(\xi_m) + e^{-\xi_m} \cdot \frac{\xi_m^m}{m!} \\ &= 1 - e^{-\xi_m} + \mathsf{q}_{m+1}(\xi_m) \\ &= 2\Big(1 - e^{-\xi_m}\Big).\end{aligned}$$

Therefore, the competitive ratio of every algorithm is

$$\mathsf{RoE} \leq \frac{2\big(1 - e^{-\xi_m}\big)}{2} = 1 - e^{-\xi_m}. \qquad \blacksquare$$

Remark 3.15. For a weaker adversary (i.e., offline adversary), one can so very slightly better than Theorem 3.14. See Appendix B for details.

4. The IID settings

Motivated by the early work of [GM66] for the TOP-1-OF-m model, in this section we study the IID setting and the $\mathbb{P}_{\max}$ objective. As a warm-up, we take a look at the IID setting with the $\mathbb{P}_{\max}$ objective and the case of $m = 1$, providing a simple single-threshold algorithm.

4.1. A single-threshold algorithm for $m = 1$

Our single-threshold algorithm $\mathcal{A}_p$ for $\mathcal{M}(\mathcal{O}_1, \mathsf{IID}, \mathbb{P}_{\max})$ selects a threshold τ equal to the pth quantile of the given distribution $\mathcal{D}$, for some $p \in [0, 1]$. In other words, τ is set such that $p = \mathbb{P}[X_i \geq \tau]$. The first time the algorithm observes a realization above τ, it queries the oracle to see whether the realization should be selected or not. If it continues, it simply accepts the first value encountered above the observed realization on which it queried $\mathcal{O}$.

Lemma 4.1. *There exists $p \in [0, 1]$ such that $\mathcal{A}_p$ selects the maximum realization with probability at least 0.797 in the $\mathcal{M}(\mathcal{O}_1, \mathit{IID}, \mathbb{P}_{\max})$ model for large n.*

Proof: Let Y be the total number of realizations above τ, and $i_1 < i_2 < \cdots < i_Y$ be the indices of the random variables above τ, i.e. $X_{i_t} > \tau$, for $t = 1, \ldots, Y$. Furthermore, let r_t be the rank of X_{i_t} in $\mathcal{X} = \{X_{i_1}, \ldots, X_{i_Y}\}$, i.e. the number k such that X_{i_t} is the kth largest number in $\mathcal{X}$, and Z be the maximum realization of $X_1, \ldots, X_n$.

X_{i_1} is the first realization we observe above τ. Notice that if $r_1 = 1$ or $r_1 = 2$ then the algorithm always selects the maximum realization Z. In other words, given that $Y = 1$ or $Y = 2$, the algorithm selects Z with probability 1. Consider the case $Y > 2$. Again, if $r_1 \leq 2$, the algorithm selects Z with probability 1. Otherwise, if $r_1 > 2$, the algorithm returns Z if and only if for all realizations above τ that appear after X_{i_1} and are also larger than X_{i_1}, the first to encounter is Z. In other words, for the algorithm to succeed in this case, it must be that among the $r_1 - 1$ values of rank smaller than r_1, the first one in the arrival order is the element of rank 1. Since the random variables are IID, the probability of this event is exactly $1/r_1-1$.

Let j be the first index such that $X_{i_j} > X_{i_1}$, and $\alpha(Y) = \mathbb{P}[\mathcal{A} \text{ selects } Z \mid Y]$. Conditioned on $Y \geq 3$, the probability that the algorithm selects Z is

$$\begin{aligned}
\alpha(Y \mid Y \geq 3) &= \mathbb{P}[r_1 = 1] + \mathbb{P}[r_1 = 2] + \sum_{t=3}^{Y} \mathbb{P}[r_1 = t]\, \mathbb{P}[r_j = 1 \mid r_1 = t] \\
&= \frac{2}{Y} + \sum_{t=3}^{Y} \frac{\mathbb{P}[r_z = 1 \mid r_1 = t]}{Y} \\
&= \frac{1}{Y}\left(2 + \sum_{t=3}^{Y} \mathbb{P}[r_z = 1 \mid r_1 = t]\right) \\
&= \frac{1}{Y}\left(2 + \sum_{t=3}^{Y} \frac{1}{t-1}\right) \\
&= \frac{1}{Y}\left(1 + \sum_{t=1}^{Y-1} \frac{1}{t}\right) \\
&= \frac{1}{Y}(1 + H_{Y-1}),
\end{aligned}$$

where H_n denotes the nth harmonic number. Recall also that $\alpha(Y \mid Y = 1) = \alpha(Y \mid Y = 2) = 1$.

Next, we estimate $\mathbb{P}[Y = i]$, by approximating Y with a Poisson distribution via Le Cam's theorem. Let $\delta_i = \left|\binom{n}{i} p^i (1-p)^{n-i} - e^{-np}\frac{(np)^i}{i!}\right|$. The idea is to set p such that $np = q$, where $q \geq 1$ is a fixed constant. We know that $\mathbb{P}[Y = i] = \binom{n}{i} p^i (1-p)^{n-i}$, and thus, by Theorem A.2, we have

$$\sum_{i=0}^{\infty} \delta_i = \sum_{i=0}^{\infty} \left|\mathbb{P}[Y = i] - e^{-np}\frac{(np)^i}{i!}\right| = \sum_{i=0}^{\infty} \left|\mathbb{P}[Y = i] - e^{-q}\frac{(q)^i}{i!}\right| \leq \frac{2qp}{\max\{1, q\}} \leq 2p = \frac{2q}{n}.$$

Overall, the probability that $\mathcal{A}$ selects Z is

$$\begin{aligned}
\alpha(Y) &= \sum_{i=0}^{n} \mathbb{P}[Y = i] \cdot \alpha(Y \mid Y = i) \\
&= \mathbb{P}[Y = 1] + \sum_{i=2}^{n} \mathbb{P}[Y = i] \cdot \alpha(Y \mid Y = i) \\
&\geq np(1-p)^{(n-1)} + \sum_{i=2}^{n} \left(e^{-q}\frac{q^i}{i!} - \delta_i\right) \cdot \alpha(Y \mid Y = i),
\end{aligned}$$

where the last inequality follows by the definition of δ_i. Thus,

$$\begin{aligned}
\alpha(Y) &= q(1 - q/n)^{(n-1)} + \sum_{i=2}^{n} e^{-q}\frac{q^i}{i!} \cdot \alpha(Y \mid Y = i) - \sum_{i=2}^{n} \delta_i \cdot \alpha(Y \mid Y = i) \\
&\geq q(1 - q/n)^{(n-1)} + \sum_{i=2}^{n} e^{-q}\frac{q^i}{i!}\frac{1 + H_{i-1}}{i} - \sum_{i=2}^{n} \delta_i \\
&\geq q(1 - q/n)^{(n-1)} + e^{-q}\sum_{i=2}^{n} \frac{q^i(1 + H_{i-1})}{i! \cdot i} - \frac{2q}{n}. \qquad (4.1)
\end{aligned}$$

It is not too difficult to see after some calculations that, as $n \to \infty$, Eq. (4.1) is maximized for $q \approx 2.435$, yielding $\alpha(Y) \approx 0.798$. ∎

It is easy to see that simply setting $q = 2$, which corresponds to $p = 2/n$ and τ being the $2/n$th quantile of $\mathcal{D}$, yields $\alpha(Y) > 0.5801$ for all $n \geq 20$. Thus, our simple single-threshold algorithm, augmented with a single oracle call, beats, even for small n, the optimal algorithm for the IID prophet inequality which uses different thresholds per distribution and achieves a probability of success approximately 0.5801 [GM66].

Since the worst-case probability of ≈ 0.5801 by [GM66] is achieved for $n \to \infty$, one might be interested in the asymptotic behavior of the probability of our algorithm, $\alpha(Y)$, for large n.

4.2. A single-threshold algorithm for general m

As we saw in the previous section, even for a simple, single-threshold algorithm, the analysis of the winning probability gets tedious quickly. In this section, we generalize our single-threshold algorithm to the case of general m, and use the fact that the maximum of a uniformly random permutation of n values changes $\mathcal{O}(\log n)$ times with high probability to obtain a guarantee on the winning probability that is super-exponential with respect to m.

As before, our algorithm selects a threshold τ such that $p = \mathbb{P}[X \geq \tau]$ and every time the algorithm observes a realization above τ, it uses an oracle query and asks $\mathcal{O}$ if the realization should be selected or not. If not, then it updates the threshold to the new higher value. If the algorithm runs out of oracle calls, then it selects the first element above the current threshold τ that is encounters, if any. In other words, the algorithm uses the oracle calls greedily for all realizations above τ.

Theorem 4.2. *For sufficiently large m, n, and an instance of $\mathcal{M}(\mathcal{O}_m, \mathsf{IID}, \mathbb{P}_{\max})$, there exists an algorithm that selects the maximum realization with probability at least $1 - \mathcal{O}\big(m^{-m/5}\big)$.*

Proof: Let $L = e^{\sqrt{m}}$. The idea is to set τ so that $p = \mathbb{P}[X \geq \tau] = L/n$. As before, let Y be the number of realizations above τ. By Theorem A.1, we have

$$\mathbb{P}[|Y - L| \geq \delta L] \leq 2e^{-\delta^2 L/3}.$$

Setting $\delta = 1$ yields that $1 \leq Y \leq 2L$ with probability at least $1-2e^{-L/3} = 1-2e^{-e^{\sqrt{m}}/3} \geq 1-m^{-m/4}$ for all m.

Next, let $X'_1, \ldots, X'_Y$ be the subsequence of all realizations larger than τ, according to their arrival order, and let $Z_i = 1$ if $X'_i > \max_{j=1}^{i-1} X'_j$, in other words if X'_i is larger than all previous realizations, and $Z_i = 0$ otherwise. Observe that $\mathbb{P}[Z_i = 1] = 1/i$, and that the random variables $Z_1, \ldots, Z_n$ are independent. Furthermore, let $M = \sum_i Z_i$ be the number of times that the maximum realization changes in the sequence $X'_1, \ldots, X'_Y$. Observe that if $M \leq m+1$, then m oracle queries are sufficient for the algorithm to always select the maximum realization. Therefore, our goal is to bound the probability that this event happens.

Conditioned on $1 \leq Y \leq 2L$, we have

$$\mathbb{E}[M] = \sum_{i=1}^{2L} \frac{1}{i} \leq \log(2L) + 1 \leq \sqrt{m} + 2.$$

For $\delta = {}^{m+2}/_{\mathbb{E}[M]} - 1$, we have

$$\mathbb{P}[M \geq m+2] = \mathbb{P}[M \geq (1+\delta)\,\mathbb{E}[M]].$$

Notice that for $m \geq 98$, we have $\delta \geq e^2$, and thus, by Theorem A.1, we obtain

$$\mathbb{P}[M \geq m+2] \leq e^{-\mathbb{E}[M]\delta \log \delta/2} \leq e^{-\frac{(m-\sqrt{m})(\log(m-\sqrt{m})-\log(m+2)/2)}{2}} \leq m^{-m/5}.$$

If we instead use the tight Chernoff bound in Theorem A.1, we can show that $\mathbb{P}[M \geq m+2] \leq m^{-m/4+\varepsilon}$ for all m and $\varepsilon > 0$.

Putting everything together, for our algorithm to succeed, it suffices to have $1 \leq Y \leq 2L$ and $M \leq m+1$, both of which happen together with probability at least $1 - \mathcal{O}\big(m^{-m/5}\big)$. ■

4.3. An (almost) tight upper bound

Now that we have presented a simple, single-threshold algorithm for the $\mathcal{M}(\mathcal{O}_m, \mathsf{IID}, \mathbb{P}_{\max})$ setting, a reasonable question to ask is how far it is from being optimal. As we show in this section, the algorithm is asymptotically almost optimal.

Theorem 4.3. *There exists an instance of $\mathcal{M}(\mathcal{O}_m, \mathsf{IID}, \mathbb{P}_{\max})$ for which no algorithm can select the maximum realization with probability greater than $1 - \mathcal{O}(m^{-m})$.*

Proof: To construct an instance in which no algorithm can achieve a high probability, fix m and consider n random variables $X_1, \dots, X_n$ drawn IID from the uniform distribution on $[0,1]$, where n is a sufficiently large number. We first divide $[0,1]$ into $k = {}^{n}/_{m \log m}$ intervals $B_1, \dots, B_k$ of length ${}^{m \log m}/_{n}$ each, with $B_i = \big((i-1) \cdot {}^{m \log m}/_{n}, i \cdot {}^{m \log m}/_{n}\big]$. For each $i = 1, \dots, n$, let Y_i denote the random variable that is equal to 1 if $X_i \in B_k$ and 0 otherwise, where B_k is the last interval. Also, let $Y = \sum_{i=1}^{n} Y_i$. Since the X_i's follow the uniform distribution, we have $\mathbb{P}[Y_i = 1] = \frac{m \log m}{n}$ for all i, and $\mathbb{E}[Y] = m \log m$.

Next, consider an algorithm $\mathcal{A}$ for $\mathcal{M}(\mathcal{O}_m, \mathsf{IID}, \mathbb{P}_{\max})$ on this instance, and assume that $Y \geq 1$, i.e. there exists at least one realization that falls in the last interval. Consider the moment that $\mathcal{A}$ observes a realization $X_i \in B_k$ that is larger than all previous realizations (including previous realizations in B_k). There are two cases:

- If $\mathcal{A}$ decides not to use a query to $\mathcal{O}$ for this realization and skip it, there is a chance it fails to select the highest realization. This definitely happens if no other realization in the future is in B_k, which occurs with probability

  $$(1 - X_i)^{n-i} \geq \left(1 - \frac{m \log m}{n}\right)^{n-i} \geq \left(1 - \frac{m \log m}{n}\right)^{n} \geq e^{-m \log m - 1} = \Omega\Big(m^{-m/1-\varepsilon}\Big)$$

  for sufficiently large n, for any $\varepsilon > 0$.
- If $\mathcal{A}$ decides to expend a query to $\mathcal{O}$ for this realization, there is a chance it fails to select the highest realization by running out of queries, deciding to select the next realization in B_k that is higher than all previous ones, and missing out on a higher realization in the future. For this to happen, it must be that $Y \geq m+2$. Let $\delta = 1 - {}^{(1+1/m)}/_{\log m}$. By Theorem A.1, this happens with probability

  $$\begin{aligned}\mathbb{P}[Y > m+1] = 1 - \mathbb{P}[Y \leq m+1] = 1 - \mathbb{P}[Y \leq (1-\delta)\,\mathbb{E}[Y]] \geq 1 - e^{-\frac{m \log m (\log m - 1 - 1/m)^2}{2 \log m^2}} \\ \geq 1 - m^{-m/4}.\end{aligned}$$

Given that $Y \geq m+2$, the probability that the first $m+2$ realizations arrive in increasing order is $1/(m+2)!$. Therefore, $\mathcal{A}$ misses out on the maximum realization in this case with probability at least (for $m \geq 6$)

$$\frac{1 - m^{-m/4}}{(m+2)!} \geq m^{-m}.$$

Therefore, $\mathcal{A}$ must miss the maximum realization with probability at least $\Omega(m^{-m})$. ■

References

[AEE+17] M. Abolhassani, S. Ehsani, H. Esfandiari, M. Hajiaghayi, R. D. Kleinberg, and B. Lucier. Beating 1-1/e for ordered prophets. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017,* 61–71, 2017.

[AGS02] D. Assaf, L. Goldstein, and E. Samuel-Cahn. Ratio prophet inequalities when the mortal has several choices. *The Annals of Applied Probability*, 12(3): 972–984, 2002.

[AS00] D. Assaf and E. Samuel-Cahn. Simple ratio prophet inequalities for a mortal with multiple choices. *Journal of Applied Probability*, 37(4): 1084–1091, 2000.

[BCGL22] S. Banerjee, V. Cohen-Addad, A. Gupta, and Z. Li. *Graph Searching with Predictions*. 2022.

[Cam60] L. L. Cam. An approximation theorem for the poisson binomial distribution. *Pacific Journal of Mathematics*, 10: 1181–1197, 1960.

[CCDN21] J. Correa, A. Cristi, P. Duetting, and A. Norouzi-Fard. Fairness and bias in online selection. *Proceedings of the 38th International Conference on Machine Learning,* vol. 139. 2112–2121, 2021.

[CFH+21] J. R. Correa, P. Foncea, R. Hoeksma, T. Oosterwijk, and T. Vredeveld. Posted price mechanisms and optimal threshold strategies for random arrivals. *Math. Oper. Res.*, 46(4): 1452–1478, 2021.

[Den12] F. Den Hollander. Probability theory: the coupling method, 2012. Notes available online: `https://mathematicaster.org/teaching/lcs22/hollander_coupling.pdf`.

[DP09] D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[EFGT23] T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang. "who is next in line?" on the significance of knowing the arrival order in bayesian online settings. *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023,* 3759–3776, 2023.

[EFN18] T. Ezra, M. Feldman, and I. Nehama. Prophets and secretaries with overbooking. *Proceedings of the 2018 ACM Conference on Economics and Computation,* 319–320, 2018.

[EHLM17] H. Esfandiari, M. Hajiaghayi, V. Liaghat, and M. Monemizadeh. Prophet secretary. *SIAM Journal on Discrete Mathematics*, 31(3): 1685–1701, 2017. eprint: https://doi.org/10.1137/15M1029394.

[GM66] J. P. Gilbert and F. Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61(313): 35–73, 1966.

[Har24] E. Harb. *New Prophet Inequalities via Poissonization and Sharding*. 2024. arXiv: 2307.00971 [cs.DS].

[HK82] T. P. Hill and R. P. Kertz. Comparisons of stop rule and supremum expectations of i.i.d. random variables. *Ann. Probab.*, 10(2): 336–345, 1982.

[Ker86] R. P. Kertz. Stop rule and supremum expectations of i.i.d. random variables: a complete comparison by conjugate duality. *Journal of Multivariate Analysis*, 19(1): 88–112, 1986.

[KS77] U. Krengel and L. Sucheston. Semiamarts and finite values. *Bull. Amer. Math. Soc.*, 83(4): 745–747, 1977.

[KS78] U. Krengel and L. Sucheston. On semiamarts, amarts, and processes with finite value. *Probability on Banach spaces*, 4: 197–266, 1978.

[KW19] R. Kleinberg and S. M. Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games Econ. Behav.*, 113: 97–115, 2019.

[Nut22] P. Nuti. The secretary problem with distributions. *Integer Programming and Combinatorial Optimization: 23rd International Conference, IPCO 2022, Eindhoven, The Netherlands, June 27-29, 2022, Proceedings*, 429–439, 2022.

[Sam84] E. Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *The Annals of Probability*, 12(4): 1213–1216, 1984.

A. Some tools from probability

Theorem A.1 (Chernoff's inequality [DP09]). *Let $Y_1, \dots, Y_n$ be independent indicator random variables with $p_i = \mathbb{P}[Y_i = 1]$ and $Y = \sum_i Y_i$. Let $\mu = \mathbb{E}[Y] = \sum_i p_i$. Then,*

(I) For $\delta \geq 0$: $\mathbb{P}[Y \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}$.

(II) For $\delta \geq 0$: $\mathbb{P}[Y \leq (1-\delta)\mu] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu}$.

(III) For $\delta \in (0,1]$, $\mathbb{P}[Y \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}$.

(IV) For $\delta \in (0,1]$ $\mathbb{P}[Y \leq (1-\delta)\mu] \leq e^{-\mu\delta^2/2}$.

(V) For $\delta > e^2$, $\mathbb{P}[Y \geq (1+\delta)\mu] < e^{-\frac{\mu\delta\log\delta}{2}}$.

The following is known as Le Cam's theorem, see [Cam60, Den12].

Theorem A.2 (Le Cam's theorem). *Let $X_1, \dots, X_n$ be independent Bernoulli random variables, with $p_i = \mathbb{P}[X_i = 1]$, for $i \in [\![n]\!]$. Let $S = \sum_i X_i$ and $\lambda = \sum_i p_i$. Then S has a Poisson binomial distribution with expectation λ. Furthermore, let $Y \sim \operatorname{Pois}\lambda$. Then we have*

$$\sum_{i=0}^{n} |\mathbb{P}[S = i] - \mathbb{P}[Y = i]| = \sum_{i=0}^{n} \left|\mathbb{P}[S = i] - e^{-\lambda}\frac{\lambda^i}{i!}\right| \leq 2\sum_{i=1}^{n} p_i^2.$$

B. Discussion on variants of the oracle and adversary choice.

In this discussion, we will weaken the adversary from an almighty adversary to an offline adversary. This can only increase the upper bound. We will also strengthen the oracle to answer $\geq$ instead of $>$ oracles. Note that the sharding analysis still holds, and the lower bound of $1 - e^{-\xi_m}$ still holds. We show that

1. Amongst single-threshold algorithms, our algorithm is still optimal. That is, no single threshold algorithm can get a competitive ratio $\geq 1 - e^{-\xi_m}$.
2. However, if we allow the algorithm to use thresholds that depend on i in the sequence $X_1, \ldots, X_n$, then our single-threshold ceases to be optimal. However, we show that the discrepancy from an optimal algorithm in this scenario, to an optimal single-threshold algorithm is very small[5]

Input instance. The input is a sequence made out of $n + 2$ random variables, for n sufficiently large. Each of these random variables can have only two values – either zero or some positive value. Specifically, for $\varepsilon > 0$ sufficiently small (e.g., $\varepsilon \ll 1/n^4$), let

$$X_1 = 1, \quad \mathbb{P}[X_i = 1 + \varepsilon(i-1)] = \frac{\xi_m}{n}, \quad \text{for} \quad i \in [\![2 : n+1]\!], \quad \text{and} \quad \mathbb{P}\big[X_{n+2} = \tfrac{1}{\varepsilon}\Psi\big] = \varepsilon,$$

where $\Psi = m!/\xi_m^m$. By Lemma 3.6, $\xi_m \approx m/e$, as such, the expected number of non-zero entries in this sequence is (roughly) $m/e + 1$.

Observation B.1. *We have that* $\Psi = m!/\xi_m^m < 1$ *by Lemma 3.4.*

Lemma B.2. *For* $Z = \max_i X_i$, *we have* $\mathbb{E}[Z] = 1 + \Psi$, *where* $\Psi = m!/\xi_m^m$.

Proof: Let $Z' = \max_{i \in [\![n+1]\!]} X_i$. Observe that $1 = X_1 \leq Z' \leq 1 + \varepsilon n$. Thus, $\lim_{\varepsilon \to 0} Z' = 1$. As such, for $Z = \max(Z', X_{n+2})$, we have $\mathbb{E}[Z] = \mathbb{E}[\max_i X_i] = (\Psi/\varepsilon)\varepsilon + (1-\varepsilon)\,\mathbb{E}[Z'] \xrightarrow[\varepsilon \to 0]{} \Psi + 1$. ∎

Observation B.3. *Let* $\widehat{X}_i$ *be an indicator variable for the event that* $X_i \geq 1$. *For sufficiently large* n, $\nabla = \sum_{i=2}^{n+1} \widehat{X}_i$ *has a binomial distribution that can be well approximated by a Poisson distribution (see Theorem A.2) with rate* ξ_m. *That is,* $\lim_{n \to \infty} \mathbb{P}\big[\nabla = k\big] = e^{-\xi_m} \dfrac{(\xi_m)^k}{k!}$.

Observe that $\lim_{n\to\infty} \mathbb{P}[\nabla \leq k] = \sum_{i=0}^{k} e^{-\xi_m} \frac{(\xi_m)^i}{i!} = \mathsf{q}_{k+1}(\xi_m)$. For simplicity of exposition, we are going to pretend that $\mathbb{P}[\nabla \leq k] = \mathsf{q}_{k+1}(\xi_m)$.

Theorem B.4. *Consider any choice of* $m \geq 1$, *and* $\delta > 0$, *and the above input instance* $\mathcal{I}$ *formed by a sequence of* non-IID *random variables. Then, for all* ***single-threshold algorithm*** $\mathcal{A} \in \mathcal{M}(\mathcal{O}_m)$ *for* $\mathcal{I}$, *we have* $\mathsf{RoE}(\mathcal{A}) \leq 1 - e^{-\xi_m} + \delta$ *and* $\mathbb{P}_{\max}(\mathcal{A}) \leq 1 - e^{-\xi_m} + \delta$.

[5] We observed that the maximum difference between the single-threshold optimal competitive ratio and the optimal multiple-threshold competitive ratio is at most ≤ 0.0018 for all m. In fact for $m = 1$, the discrepancy is ≤ 0.000000973! See Figure B.1. This very small discrepancy is separately interesting to address, especially that the discrepancy between optimal multiple-threshold algorithms and single-threshold algorithms for other prophet inequalities is usually quite large. For example, the Top-1-of-2 model has a discrepancy of almost 0.1 between single threshold and multiple-threshold algorithms.

Proof: Consider all the distinct values that might appear in $\mathcal{I}$ – there are $n+2$ such values. Thus, there are only $n+2$ single-threshold algorithms we need to consider (corresponding to each of these values). If the threshold is $\tau = \Psi/\varepsilon$, then the algorithm gets on expectation $\Psi < 1$, which is clearly suboptimal compared to accepting X_1 immediately.

Next, consider the threshold 1. If there are at most $m-1$ non-zero values in $X_2, \ldots, X_{n+1}$, then the algorithm continues to X_{n+2} and the algorithm gets at most $\max(X_{n+2}, 1+n\varepsilon)$. If there are at least m non-zero values in $X_2, \ldots, X_{n+1}$, then the algorithm gets at most $1+n\epsilon$. Hence, for

$$\rho = \Psi \frac{1}{\varepsilon} \cdot \varepsilon + (1-\varepsilon)(1+n\varepsilon) \xrightarrow{\varepsilon \to 0} 1 + \Psi, \tag{B.1}$$

we have

$$\mathbb{E}[\mathcal{A}] \leq \mathbb{P}[\nabla \leq m-1]\left(\Psi \frac{1}{\varepsilon} \cdot \varepsilon + (1-\varepsilon)(1+n\varepsilon)\right) + \mathbb{P}[\nabla \geq m](1+n\varepsilon) \underbrace{\leq}_{\varepsilon \to 0} \mathsf{q}_m(\xi_m)\Psi + 1.$$

Since $\mathsf{q}_{m+1}(\xi_m) = 1 - e^{-\xi_m}$, we have

$$\mathsf{q}_m(\xi_m)\Psi = \mathsf{q}_m(\xi_m)\frac{m!}{\xi_m^m} = \left(\mathsf{q}_{m+1}(\xi_m) - e^{-\xi_m}\frac{\xi^m}{m!}\right) \cdot \frac{m!}{\xi_m^m} = (1 - e^{-\xi_m})\Psi - e^{-\xi_m}.$$

Thus, we have (as $\varepsilon \to 0$) that

$$\mathbb{E}[\mathcal{A}] \leq 1 - e^{-\xi_m} + (1 - e^{-\xi_m})\Psi = (1 - e^{-\xi_m})(1 + \Psi).$$

Next, consider a threshold of $1 + (i-1)\varepsilon$. Here, the algorithm is "activated" at X_i, for $i \in [\![2 : n+1]\!]$. Let

$$\beta = (n - i + 2)\frac{\xi_m}{n}$$

denote the Poisson rate for $X_i, \ldots, X_{n+1}$, and let $\nabla_i = \sum_{j=i}^{n+1} X_j$. Clearly $\beta \leq \xi_m$. If $X_i, \ldots, X_{n+1}$ are all zeros, then the algorithm gets the expectation of X_{n+2}. If there are at least one, and at most m non-zero values in $X_i, \ldots, X_{n+1}$, then the algorithm continues to X_{n+2} and the algorithm gets at most $\mathbb{E}[\max(X_{n+2}, 1+n\varepsilon)] \leq \rho = 1 + \Psi$, see Eq. (B.1). If there are at least $m+1$ non-zero values in $X_i, \ldots, X_{n+1}$, then the algorithm gets at most $1+n\epsilon$. Hence, on expectation, the algorithm gets

$$\begin{aligned}
\mathbb{E}[\mathcal{A}] &\leq \mathbb{P}[\nabla_i = 0]\,\mathbb{E}[X_{n+2}] + \mathbb{P}\big[\nabla_i \in [\![1 : m]\!]\big]\rho + \mathbb{P}\big[\nabla_i > m\big](1+n\varepsilon) \\
&\leq \mathsf{q}_1(\beta)\Psi + (\mathsf{q}_{m+1}(\beta) - \mathsf{q}_1(\beta))(1+\Psi) + (1 - \mathsf{q}_{m+1}(\beta)) \cdot (1+n\varepsilon) \\
&\xrightarrow[\varepsilon \to 0]{} \mathsf{q}_1(\beta)\Psi + \mathsf{q}_{m+1}(\beta) + \mathsf{q}_{m+1}(\beta)\Psi - \mathsf{q}_1(\beta) - \mathsf{q}_1(\beta)\Psi + 1 - \mathsf{q}_{m+1}(\beta) \\
&= 1 + \mathsf{q}_{m+1}(\beta)\Psi - \mathsf{q}_1(\beta) = 1 + \mathsf{q}_{m+1}(\beta)\Psi - e^{-\beta}.
\end{aligned}$$

Let $f(x) = \mathsf{q}_{m+1}(x)\Psi - e^{-x}$. By Lemma 3.3, we have $f'(x) = -e^{-x}\frac{x^m}{m!}\Psi + e^{-x} = -e^{-x}\frac{x^m}{\xi_m^m} + e^{-x}$. As such, $f'(x) > 0$ for $x \in [0, \xi_m)$. Namely, f is increasing in this range, and this range contains the value β. Since $\mathsf{q}_{m+1}(\xi_m) = 1 - e^{-\xi_m}$, we have

$$\mathbb{E}[\mathcal{A}] \leq 1 + f(\beta) \leq 1 + f(\xi_m) \leq 1 + (1 - e^{-\xi_m})\Psi - e^{-\xi_m} = (1 - e^{-\xi_m})(1 + \Psi).$$

This implies, that for all cases, the competitive ratio is

$$\frac{\mathbb{E}[\mathcal{A}]}{\mathbb{E}[Z]} \leq \frac{1 - e^{-\xi_m} + (1 - e^{-\xi_m})\Psi}{1 + \Psi} = \frac{\big(1 - e^{-\xi_m}\big)(1 + \Psi)}{1 + \Psi} = 1 - e^{-\xi_m}. \qquad \blacksquare$$

B.1. On the optimal multiple-threshold algorithms

Intuitively, a better strategy than single-threshold, is using different thresholds (potentially based on the values seen so far).

New Input Instance The input is a sequence made out of $n+2$ random variables, for n sufficiently large. Each of these random variables can have only two values – either zero or some positive value. Specifically, for $\varepsilon > 0$ sufficiently small (e.g., $\varepsilon \ll 1/n^4$), let

$$X_1 = 1, \quad \mathbb{P}[X_i = 1 + \varepsilon(i-1)] = \frac{c_{1,m}}{n}, \quad \text{for} \quad i \in [\![2 : n+1]\!], \quad \text{and} \quad \mathbb{P}\big[X_{n+2} = \tfrac{c_{2,m}}{\varepsilon}\big] = \varepsilon,$$

Optimal algorithm. For a fixed instance of the prophet inequality problem, one can usually argue about the optimal algorithm for the instance using reverse dynamic programming. The argument is standard, but we include it here for the sake of completeness. Let $b_i = 1 + \varepsilon(i-1)$, for $i \in [\![1 : n+1]\!]$, and $b_{n+2} = c_{2,m}/\varepsilon$. Note that $b_1 \le \cdots \le b_{n+2}$. Similarly, let $p_1 = 1$, $p_i = c_{1,m}/n$ for $i \in [\![2 : n+1]\!]$, and finally $p_{n+2} = \varepsilon$. Now, the input is the sequence of random variables $X_1, \ldots, X_{n+2}$, with

$$\mathbb{P}[X_i = b_i] = p_i, \quad \text{for} \quad i = 1, \ldots, n+2.$$

Let $Z_i = \max(X_i, \ldots, X_n)$. Let $E_t(k)$ be the expected value of an *optimal algorithm* running on $X_k, \ldots, X_n$ having access to t oracle calls. Let $E_t^{\uparrow}(k)$ be the expected value of an *optimal algorithm* running on $X_k, \ldots, X_n$ with t oracle calls, given that $Z_k > 0$. We have the (mutual) recurrence

$$\begin{aligned} E_t(k) = {} & \mathbb{P}[X_k = 0] E_t(k+1) \\ & + \mathbb{P}[X_k > 0] \begin{cases} \max \begin{cases} E_t(k+1) \\ \mathbb{P}[Z_{k+1} = 0] b_k + \mathbb{P}[Z_{k+1} > 0] E_{t-1}^{\uparrow}(k+1) \end{cases} & t > 0 \\ \max\left(E_t(k+1), b_k\right) & t = 0. \end{cases} \end{aligned}$$

Let

$$\alpha_k = \mathbb{P}[X_k > 0 \mid Z_k > 0] = \frac{\mathbb{P}[(X_k > 0) \cap (Z_k > 0)]}{\mathbb{P}[Z_k > 0]} = \frac{\mathbb{P}[X_k > 0]}{\mathbb{P}[Z_k > 0]}.$$

Consider

$$\beta_{k+1} = \mathbb{P}[Z_{k+1} > 0 \mid (X_k > 0) \cap (Z_k > 0)] = \mathbb{P}[Z_{k+1} > 0 \mid X_k > 0] = \mathbb{P}[Z_{k+1} > 0].$$

We now have

$$E_t^{\uparrow}(k) = (1 - \alpha_k) E_t^{\uparrow}(k+1) + \begin{cases} \alpha_k \max \begin{cases} E_t(k+1), \\ (1 - \beta_{k+1}) b_k + \beta_{k+1} E_{t-1}^{\uparrow}(k+1) \end{cases} & t > 0 \\ \alpha_k \max\left(E_t(k+1), b_k\right) & t = 0 \end{cases}$$

It is straightforward to argue by reverse induction on k that $E_t(k)$ is the best any algorithm can get on expectation from $X_k, \ldots, X_{n+2}$ using t oracle queries, as the recurrence includes all possible outcomes of these queries. Also note that the recurrence can easily be evaluated in $O(nm)$ time.

m	$c_{1,m}$	$c_{2,m}$	OPT Competitive Ratio	$1 - e^{-\xi_m}$	**Difference**
1	1.146	0.872	0.682	0.682	0.000000973
2	1.685	0.779	0.792	0.792	0.000689
3	2.054	0.808	0.863	0.861	0.00178
4	3.250	0.682	0.909	0.907	0.00170
5	3.696	0.651	0.939	0.937	0.00170
6	3.826	0.628	0.959	0.958	0.00154
7	4.330	0.612	0.973	0.971	0.00131
8	4.195	0.682	0.982	0.980	0.00113
9	5.234	0.580	0.988	0.987	0.000846
10	5.854	0.571	0.992	0.991	0.000656
11	6.131	0.563	0.994	0.994	0.000500

Table B.1: Maximum discrepancy between single-threshold algorithm and multiple-threshold algorithm for $m = 1$ to $m = 11$. Rounded to 3 significant digits.

How far off is the optimal single-threshold algorithm It is natural to ask how far our optimal single-threshold algorithm is from the optimal multiple-threshold algorithm above. We set $n = 100000$ and $\varepsilon = 10^{-18}$, and performed experiments on $m = 1, \ldots, 11$. We observed that the maximum difference between the single-threshold optimal competitive ratio and the optimal multiple-threshold competitive ratio is at most ≤ 0.0018. See Figure B.1. This very small discrepancy is separately interesting to address, especially that the discrepancy between optimal multiple-threshold algorithms and single-threshold algorithms for other prophet inequalities is usually quite large. For example, the Top-1-of-2 model has a discrepancy of almost 0.1 between single threshold and multiple-threshold algorithms.