arXiv:2205.05519v2 [cs.DS] 22 Aug 2023

Prophet Inequality on I.I.D. Distributions: Beating $1 - 1/e$ with a Single Query

Bo Li* Xiaowei Wu † Yutong Wu ‡

Abstract

In this work, we study the single-choice prophet inequality problem, where a gambler faces a sequence of n online i.i.d. random variables drawn from an unknown distribution. When a variable reveals its value, the gambler needs to decide irrevocably whether or not to accept the value. The goal is to maximize the competitive ratio between the expected gain of the gambler and that of the maximum variable. It is shown by Correa et al. [CDFS19] that when the distribution is unknown or only $o(n)$ uniform samples from the distribution are given, the best an algorithm can do is $1/e$-competitive. In contrast, when the distribution is known [CFH+17] or $\Omega(n)$ uniform samples are given [RWW20], the optimal competitive ratio of 0.7451 can be achieved. In this paper, we study a new model in which the algorithm has access to an oracle that answers quantile queries about the distribution and investigate to what extent we can use a small number of queries to achieve good competitive ratios. We first use the answers from the queries to implement the threshold-based algorithms and show that with two thresholds our algorithm achieves a competitive ratio of 0.6786. Motivated by the two-threshold algorithm, we design the observe-and-accept algorithm that requires only a single query. This algorithm sets a threshold in the first phase by making a query and uses the maximum realization from the first phase as the threshold for the second phase. It can be viewed as a natural combination of the single-threshold algorithm and the algorithm for the secretary problem. By properly choosing the quantile to query and the break-point between the two phases, we achieve a competitive ratio of 0.6718, beating the benchmark of $1 - 1/e$.

*The Hong Kong Polytechnic University. comp-bo.li@polyu.edu.hk
†University of Macau. xiaoweiwu@um.edu.mo
‡The University of Texas at Austin. yutong.wu@utexas.edu

1 Introduction

Prophet inequality is one the of most widely studied problems in optimal stopping theory. In the classic setting, a gambler faces a sequence of online non-negative random variables $x_1, \ldots, x_n$ independently drawn from distributions $D_1, \ldots, D_n$. The realizations of these variables represent potential rewards given to the gambler. After observing the realization of a variable, the gambler needs to make an irrevocable decision on whether to accept the reward and leave the game, or reject it and continue with the next variable. All rejected values cannot be collected anymore. A prophet knows all the realizations beforehand and thus can always select the highest reward. The expected reward of the prophet is then defined as $\mathsf{OPT} = \mathbb{E}_{x_1,\ldots,x_n}[\max_i\{x_i\}]$. The gambler's goal is to choose a stopping rule so that her expected reward, which we refer to as ALG, is as close to that of the prophet as possible. The performance is measured by the *competitive ratio*, which is defined as the worst case of $\mathsf{ALG}/\mathsf{OPT}$ over all possible distributions. The prophet inequality problem has received increasingly more attention due to its connections with Bayesian mechanism design, especially the posted price mechanism, where buyers arrive online and the seller provides each buyer with a take-it-or-leave-it offer [HKS07, CHMS10, Luc17].

It is proved in [KS77, KS78] and [SC84] that there exists a stopping rule such that the gambler's expected reward is at least $0.5 \cdot \mathsf{OPT}$, and this is the best possible competitive ratio when the distributions are distinct and the order of variables is adversarial. However, when the distributions are identical, i.e., $D_i = D$ for all i, much better competitive ratios can be achieved. Hill and Kertz[HK82] proved that there is an algorithm that achieves a competitive ratio of $1 - 1/e \approx 0.6321$ while Kertz [Ker86] showed that no algorithms can do better than 0.7451. The best-known competitive ratio remained $1 - 1/e$ until Abolhassani et al. [AEE+17] improved it to 0.738. Later, Correa et al. [CFH+17] proposed a *blind quantile strategy* with a tight competitive ratio of 0.7451. Informally speaking, a blind quantile strategy defines a sequence of increasing probabilities $p_1 < \cdots < p_n$ (which depend on n and the distribution D, but are independent of the realizations), so that the acceptance probability for each variable x_i equals p_i. This is done by setting the i-th threshold θ_i such that $\Pr_{x\sim D}[x > \theta_i] = p_i$ and accepting the i-th variable if $x_i > \theta_i$. However, computing the probabilities $p_1, \ldots, p_n$ that give the optimal ratio is highly non-trivial and requires complete information of the distribution D. As pointed out in a survey paper [CFH+18], an interesting research problem is to investigate the amount of information needed to achieve a good competitive ratio:

How much knowledge of the distributions is required to achieve a good competitive ratio?

To follow up this question, a line of recent works studies the prophet inequality problem on unknown distributions, most of which focus on the setting where uniform random samples from the distribution are given [AKW19, CDFS19, RWW20]. Particularly, it is shown in [RWW20] that $O(n)$ samples are sufficient to achieve a competitive ratio arbitrarily close to 0.7451, and in [CDFS19] that no algorithm can use $o(n)$ samples to ensure a result better than $1/e(\approx 0.368)$-competitive. However, in some applications obtaining random samples can be difficult. Consider the following analogy of prophet inequality: a seller sells a product with a posted price to a sequence of buyers whose values are independently drawn from a distribution. A buyer purchases the product if and only if the price is lower than her private value. In practice, the seller often does not have an accurate knowledge of the value distribution or random samples from this distribution. The seller can, however, observe the selling records to draw a relationship between a historic price and the fraction of buyers that have accepted the price, which is precisely a *quantile* of the value distribution. Since one price is usually offered over months and even years, which means that the number of historic prices is often small, such quantile information is also scarce. Therefore, a

natural question is how a seller can take advantage of the limited quantile information to set up a price that maximizes their expected revenue.

Motivated by the above discussion, in this work, we propose a new information model in which the algorithm has access to an oracle that answers quantile queries on the unknown distribution. Formally, given a quantile $q \in [0, 1]$, the oracle returns a value v such that $\Pr_{x \sim D}[x > v] = q$. In other words, the oracle answers queries in the form of "*what price should be set to generate a success probability of q when buyers' values are independently drawn from the distribution D?*" We aim to understand how much reward can be guaranteed using only a few queries and construct such algorithms. Our model aligns well with the study of query complexity in Bayesian auction design – a closely related area to prophet inequality. In particular, [ABB21] considered the revenue maximization problem with a single value-quantile pair, and showed that, for example, if we are given the value at quantile 0.5, we are able to guarantee 85% of the optimal revenue when there is a single buyer and the full distribution is known. As far as we know, the query model has not been studied in the context of prophet inequality.

Besides algorithms that use little knowledge, *simple* algorithms are often much preferred due to their easy implementation in real-world scenarios. For example, in the *secretary problem* [GM66], a widely used simple strategy is to discard the first $1/e$ fraction of the variables and select the first variable whose realization is greater than all the discarded ones. We call this strategy the *observe-then-accept* algorithm. It was proved in [CDFS19] that this algorithm is $1/e$-competitive and is optimal when the distribution is unknown or $o(n)$ uniform samples are given. For the prophet inequality problem, the *single-threshold* algorithm simply uses a fixed threshold for all variables and accepts the first variable whose realization exceeds the threshold. It is shown in [EHKS18] that by setting an appropriate threshold, the single-threshold algorithm ensures a competitive ratio of $1 - 1/e$, which is the best possible ratio using just one threshold[1]. In this work, we propose two simple algorithms for the query model that achieve good competitive ratios. We first use quantile queries to obtain thresholds and extend the single-threshold algorithm to a simple *multi-threshold algorithm*. The proposed multi-threshold algorithm naturally does better than $(1-1/e)$-competitive and we give a general framework to derive its competitive ratio. Besides using multiple thresholds, we prove that beating $1 - 1/e$ can also be done by using a single query via our *observe-and-accept algorithm*. Indeed, an interesting take-home message from this work is that a simple combination of the observe-then-accept algorithm and the single-threshold algorithm performs strictly better than $(1 - 1/e)$-competitive.

A review of more related works can be found in Appendix B.

1.1 Our Contribution

We study the prophet inequality problem with unknown i.i.d. distributions and propose simple algorithms that use a few queries to achieve competitive ratios strictly better than $1 - 1/e$. Our contribution is summarized as follows.

Two-threshold Algorithm. Our first result is to show that a simple two-threshold algorithm achieves a competitive ratio strictly better than $1 - 1/e$, the optimal ratio of single-threshold algorithms [EHKS18]. Specifically, the algorithm fixes two quantiles q_1, q_2 (which define two thresholds θ_1, θ_2) and divides the time horizon $\{1, \ldots, n\}$ into two phases. The threshold θ_1 (resp. θ_2) is then used to decide the acceptance of variables in the first (resp. second) phase. Our main technical contribution is a careful formulation of the lower bounds on ALG and upper bounds on OPT in

[1] We remark that their result holds for the more general setting of the prophet secretary problem. When the distributions are discrete, randomization is required to achieve the $1 - 1/e$ competitive ratio.

terms of the parameters, which then become the objective and constraints for a minimization linear program (LP) whose optimal solution provides a lower bound on the competitive ratio. By fixing appropriate parameters and solving the LP, we show that the competitive ratio of the algorithm is at least 0.6786.[2] We also complement our positive result with an upper bound on the competitive ratio: we show that any two-threshold algorithm cannot do better than 0.7081-competitive.

Observe-and-accept Algorithms. Our second result, which is the main contribution of this work, is showing an algorithm using a single query that can do strictly better than $(1 - 1/e)$-competitive. Most interestingly, we show that a natural combination of the single-threshold algorithm and the observe-then-accept algorithm gives one such algorithm, with a competitive ratio of at least 0.6718. The algorithm works in a similar way as the two-threshold algorithm in that it partitions the time horizon into two phases and uses the query result from the oracle as a threshold to decide the acceptance of variables in the first phase. In the second phase, however, instead of making a second query, the algorithm uses the maximum realization of variables in the first phase as the threshold, given that no variables are accepted during the first phase. We refer to this algorithm as *observe-and-accept*, as it is similar to the algorithm for the secretary problem but in addition, allows acceptance of variables in the observation phase. As before, we lower bound the competitive ratio by bounding ALG from below and OPT from above to formulate an LP. However, different from the analysis for the two-threshold algorithm, the main technical challenge is to lower bound ALG when the second threshold is a random variable that depends on the realizations of variables in the first phase, i.e., the algorithm is not a blind strategy algorithm. To overcome this main difficulty, we extend the LP by further dividing the domain of the second threshold into continuous segments, whose boundary points become additional variables of the LP. Actually, given any quantile, we can compute the optimal separation point of the two phases and obtain the corresponding optimal reward by running the observe-and-accept algorithm. Via carefully designing the query and the separation point of phases, our main result is a 0.6718-competitive observe-and-accept algorithm. To complement this result, we provide hard instances for which the observe-and-accept algorithm performs no better than 0.6921-competitive.

Robustness of the Quantile to Query. If we are not allowed to choose an arbitrary quantile but are instead given a specific one, we show in Figure 1.1 some examples when the given quantile is $1 - c/n$ for some constant c; a detailed setup will be introduced in Section 4.2. Recall that in our motivating example, the thresholds are the posted prices of an item for a sequence of n buyers. We can view c/n as c out of n buyers have private values higher than the price. As c increases, the corresponding price decreases. We can observe that as long as c is between 0.5 and 1.0, the expected reward from observe-and-accept is always greater than $(1 - 1/e)$. When c is between 0.1 and 2.5, the quantile information is still beneficial in the sense that the observe-and-accept algorithm has a competitive ratio better than $1/e$ – the best we can do when no information is given.

Bayesian Auctions with Queries. Besides prophet inequality, queries have also been studied in Bayesian auctions when the seller does not know the prior distributions. Similar to our motivation, Allouah et al. [ABB21] considered the single-item single-buyer revenue maximization problem with one value-quantile pair. They showed that, for example, if we are given the value at quantile 0.01, 51% of the optimal revenue is guaranteed when the full distribution is known, and if we are given the value at quantile 0.5, the guarantee is improved to 85%. Chen et al.[CLLL22] studied a more

[2] Our analysis extends naturally to the multi-threshold algorithm. In Appendix C, we show that better competitive ratios can be achieved with more queries and thresholds.

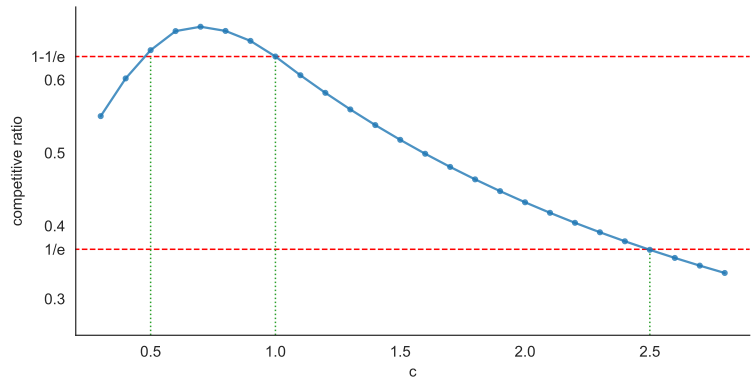

Figure 1.1: Competitive ratios for different quantiles at $1 - c/n$.

general setting of multi-item multi-buyer auctions and gave an almost tight number of queries to ensure a constant fraction of the optimal revenue. Hu et al. [HHSW21] explored the middle ground between random samples and queries by allowing the algorithm to specify a quantile interval and sample from the prior distribution restricted to the interval. In a different flavor from the quantile query oracle we study in this work, Leme et al. [LSTW21] defined *pricing queries* in that when the oracle is given a price, it generates a random sample and returns the sign of whether the sample is above the given price or not. Both query-related settings are of independent interest to be studied in prophet inequality.

1.2 Organization of the Paper

The rest of this paper is organized as follows. We introduce the necessary notations and some basic properties in Section 2. In Section 3, we review the analysis for proving the $1 - 1/e$ competitive ratio of the single-threshold algorithm (Section 3.1), introduce the two-threshold algorithm (Section 3.2), and show that it achieves a competitive ratio of at least 0.6786 using a factor revealing LP (Section 3.5). Built on top of the algorithm and analysis from Section 3, we propose in Section 4 the observe-and-accept algorithm that uses a single query to achieve a competitive ratio of at least 0.6718. Finally, we conclude our results and propose some open problems in Section 5.

2 Preliminaries

In our problem, there are n non-negative random variables $\{x_1, \ldots, x_n\}$ that are independently and identically drawn from an unknown distribution D. We slightly abuse the notation and use $x_i's$ to denote both random variables and their realizations. The realizations of the variables are revealed to the algorithm one by one. When the value of a variable is revealed, the algorithm needs to make an irrevocable decision on whether to *accept* the value and stop the algorithm, or *reject* this variable and proceed to the next one. The objective is to pick a realization that is as large as possible. For any integer t, we use $[t]$ to denote $\{1, 2, \ldots, t\}$. Throughout this paper, we use $\tau \in [n+1]$ to denote the stopping time of the algorithm. If the algorithm accepts variable x_i, then $\tau = i$; if the algorithm does not accept any variable, we set $\tau = n + 1$. We let ALG denote the expected gain of the algorithm. We compare the performance of the algorithm with a prophet, who sees all the realizations of variables before making a decision. In other words, the benchmark is

$$\mathsf{OPT} = \mathbb{E}_{(x_1,\ldots,x_n)\sim D^n}\left[\max_{i\in[n]}\{x_i\}\right].$$

The *competitive ratio* of the algorithm is defined as the minimum of ALG/OPT over all the distributions. In this work, we consider the setting where the distribution is unknown but the algorithm can make queries on the cumulative distribution function (CDF) of the distribution.

Query. We denote the CDF of the distribution by $F(\theta) = \Pr_{x\sim D}[x < \theta]$ and the survival function by $G(\theta) = \Pr_{x\sim D}[x \geq \theta]$. Note that for every distribution, the CDF $F(\cdot) : \mathbb{R} \to [0, 1]$ is non-decreasing. In this work, we assume the distribution does not contain mass points. That is, $\Pr_{x\sim D}(x = t) = 0$ for all t. When the distribution D is known, this assumption is w.l.o.g. because we can add an arbitrarily small random noise to the distribution. On the other hand, when the distribution D is unknown, the assumption is apparent[3] in order to achieve a good competitive ratio with few queries. For distributions without mass points, the CDF $F(\cdot)$ is strictly increasing. Given any quantile query at $q \in [0, 1]$, the distribution quantile oracle returns the corresponding value $v(q)$ such that

$$v(q) := \inf_{\theta\in\mathbb{R}} \{F(\theta) \geq q\} = F^{-1}(q). \tag{2.1}$$

Therefore, if $v(q)$ is used as a threshold to decide the acceptances of variables, each variable is accepted with probability $1 - q$. We note that $v(\cdot)$ is also strictly increasing.

For any number t, we use $(t)^+$ to denote $\max\{t, 0\}$. Since we often make queries at $q = 1 - c/n$ for some constant c (i.e., probability to accept is c/n), for convenience we define

$$\Delta(c) := n \cdot \mathbb{E}_{x\sim D}[(x - v(1 - \tfrac{c}{n}))^+].$$

Zero Queries. We briefly discuss the case when we cannot make any query to the distribution oracle. The problem then degenerates to the model where the algorithm has no information on the distribution [CDFS19]. The classic algorithm for the secretary problem [GM66] achieves the optimal competitive ratio of $1/e$, which also applies to our setting. Formally, we observe the first n/e variables without accepting any of them; for the following variables, we accept the first variable whose realization exceeds the maximum of the previous ones. Clearly, the algorithm retains the $1/e$ competitive ratio since it can be equivalently described as independently drawing n realizations from the distribution and giving them a random arrival order. Interestingly, Correa et al. [CDFS19] showed that this is the best possible ratio for all algorithms that have no information about the distribution.

Theorem 2.1 (Zero-Query) *There is a $1/e$-competitive algorithm that makes no queries for the prophet inequality problem on unknown i.i.d. distributions. Moreover, the competitive ratio is optimal.*

3 Warm-up Analysis and Techniques

We first provide a warm-up analysis for single-threshold algorithms in Section 3.1, which is the least a single query can do by using the query result as a threshold to accept the first variable whose realization exceeds the threshold. We show that the competitive ratio of the algorithm is $1 - 1/e$. We then show in Section 3.2 that by using two thresholds that are obtained by making two queries, we can do strictly better than $1 - 1/e$. Our main technique is based on the theory

[3] If there are mass points in the distribution, the query model will not be well-defined since for some quantile query there might not exist any value with this particular quantile. We will then have to adopt some fixed rule to return a nearby value as in Equation (2.1). In this way, the distributions would be distorted, and we demonstrate in Appendix A that this distortion can be arbitrarily large in some cases.

of factor revealing LPs. Both the algorithm and the technique from Section 3.2 will be important building blocks towards proving our main result in Section 4.

3.1 The Single-Threshold Algorithm

Single-threshold algorithms have been investigated by [EHKS18] and [CSZ19], where a $(1-1/e)$ competitive ratio was proved. Moreover, the competitive ratio is optimal for single-threshold algorithms [EHKS18]. For completeness, we give the formal proof here, which will also serve as a warm-up for our proofs in later sections.

Theorem 3.1 *There exists a $(1-1/e)$-competitive algorithm for the prophet inequality problem with unknown i.i.d. distributions that makes a single query to the distribution oracle.*

Proof: The algorithm works as follows:

1. Let $\theta = v(1-1/n)$. In other words, we have $\Pr_{x\sim D}[x \geq \theta] = 1/n$.
2. For the realization of each variable x_i, where $i = 1, 2, \ldots, n$, accept x_i and terminate if $x_i \geq \theta$; otherwise reject it and observe the next variable, if any.

We refer to θ as the *threshold* of the algorithm. Note that $F(\theta) = 1 - 1/n$ and $G(\theta) = 1/n$. We show that the competitive ratio for the above algorithm is at least $1 - 1/e$. We prove this by deriving an upper bound for OPT and a lower bound for ALG. Let $x^* = \max_i\{x_i\}$ denote the maximum of the n i.i.d. random variables. Then $\mathsf{OPT} = \mathbb{E}[x^*]$. Given threshold θ, we have

$$\begin{aligned}\mathsf{OPT} = \mathbb{E}[x^*] &= \mathbb{E}[x^* \mid x^* < \theta] \cdot \Pr[x^* < \theta] + \mathbb{E}[x^* \mid x^* \geq \theta] \cdot \Pr[x^* \geq \theta] \\ &= \mathbb{E}[x^* \mid x^* < \theta] \cdot \Pr[x^* < \theta] + (\theta + \mathbb{E}[x^* - \theta \mid x^* \geq \theta]) \cdot \Pr[x^* \geq \theta] \\ &= \mathbb{E}[x^* \mid x^* < \theta] \cdot \Pr[x^* < \theta] + \theta \cdot \Pr[x^* \geq \theta] + \mathbb{E}[x^* - \theta \mid x^* \geq \theta] \cdot \Pr[x^* \geq \theta].\end{aligned}$$

Note that the conditional expectation $\mathbb{E}[x^* \mid x^* < \theta]$ is at most θ. Moreover,

$$\mathbb{E}[(x^* - \theta)^+] = 0 \cdot \Pr[x^* \leq \theta] + \mathbb{E}[x^* - \theta \mid x^* \geq \theta] \cdot \Pr[x^* \geq \theta]. \tag{3.1}$$

Thus we obtain the following upper bound on OPT,

$$\mathsf{OPT} \leq \theta \cdot \Pr[x^* < \theta] + \theta \cdot \Pr[x^* \geq \theta] + \mathbb{E}[(x^* - \theta)^+] = \theta + \mathbb{E}[(x^* - \theta)^+].$$

Since x^* is the maximum among all x_i's and x_i's are i.i.d., we have

$$\mathbb{E}[(x^* - \theta)^+] \leq \sum_{i=1}^{n} \mathbb{E}[(x_i - \theta)^+] = n \cdot \mathbb{E}[(x - \theta)^+] = \Delta(1).$$

Overall, a valid upper bound for OPT is given by

$$\mathsf{OPT} \leq \theta + \Delta(1). \tag{3.2}$$

Recall $\tau \in [n+1]$ is the stopping time of the algorithm. The expected gain of the algorithm can then be written as

$$\mathsf{ALG} = \sum_{i=1}^{n} \Pr[\tau = i] \cdot \mathbb{E}[x_i \mid \tau = i]. \tag{3.3}$$

The algorithm accepts the i-th variable if and only if the first $i-1$ variables are smaller than the threshold θ and x_i is at least θ. In other words, we have

$$\Pr[\tau = i] = F(\theta)^{i-1} \cdot G(\theta). \tag{3.4}$$

Note that $\mathbb{E}[x_i \mid \tau = i] = \mathbb{E}[x_i \mid x_i \geq \theta]$. Conditioned on $x_i \geq \theta$, the algorithm receives θ and the expected difference between x_i and θ. That is,

$$\mathbb{E}[x_i \mid \tau = i] = \theta + \mathbb{E}[x_i - \theta \mid x_i \geq \theta] = \theta + \frac{\mathbb{E}[(x_i - \theta)^+]}{G(\theta)} = \theta + \frac{\Delta(1)/n}{G(\theta)}, \tag{3.5}$$

where the second last equality is derived from a similar argument to (3.1). By plugging Equations (3.4) and (3.5) into Equation (3.3), we have

$$\begin{aligned}\mathsf{ALG} &= \sum_{i=1}^{n} F(\theta)^{i-1} \cdot G(\theta) \cdot \left(\theta + \frac{\Delta(1)/n}{G(\theta)}\right) = \sum_{i=1}^{n} \left(1 - \frac{1}{n}\right)^{i-1} \left(\frac{\theta}{n} + \frac{\Delta(1)}{n}\right) \\ &= \frac{1 - (1 - \frac{1}{n})^n}{1/n} \cdot \frac{1}{n} \cdot (\theta + \Delta(1)) \geq (1 - (1 - 1/n)^n) \cdot \mathsf{OPT}.\end{aligned}$$

Since $(1 - 1/n)^n \leq 1/e$ for all $n \in \mathbb{N}_+$, we have $\mathsf{ALG} \geq (1 - 1/e) \cdot \mathsf{OPT}$. ■

3.2 The Two-Threshold Algorithm

Next, we show that one can beat the competitive ratio of $1 - 1/e$ with a two-threshold algorithm[4]. We first briefly explain why the performance of single-threshold algorithms is upper bounded by $1 - 1/e$, and how to get around this upper bound using multiple thresholds. We observe that any single-threshold algorithm faces the following trade-off: the threshold cannot be too high such that the algorithm terminates with no acceptance, and the threshold cannot be too low such that the algorithm terminates too early without observing sufficiently many variables. Recall that $\tau \in [n+1]$ is the stopping time. The trade-off is reflected by $\Pr[\tau \leq n]$ (the probability of accepting some variable) and $\frac{1}{n}\sum_{i=1}^{n} \Pr[\tau \geq i]$ (the average probability of "seeing" a variable). In fact, the competitive ratio of any threshold-based algorithm is upper bounded by these two terms. Suppose the threshold is $\theta = v(1 - c/n)$ for some c. It can be shown that $\Pr[\tau \leq n] \geq 1 - 1/e$ only if $c \geq 1$ while $\frac{1}{n}\sum_{i=1}^{n} \Pr[\tau \geq i] \geq 1 - 1/e$ only if $c \leq 1$. Therefore, one has to set $c = 1$ to achieve the competitive ratio of $1 - 1/e$, which is optimal. Now assume that we can use two thresholds: $\theta_1 = v(1 - c_1/n)$ and $\theta_2 = v(1 - c_2/n)$, for some $c_1 < c_2$. In order to beat $1 - 1/e$, we need both $\frac{1}{n}\sum_{i=1}^{n} \Pr[\tau \geq i]$ and $\Pr[\tau \leq n]$ to exceed $1-1/e$. To ensure $\Pr[\tau \leq n] > 1-1/e$, we must set $c_2 > 1$, as otherwise, we have $c_1 < c_2 \leq 1$ and the algorithm is strictly worse than the single-threshold algorithm with $c = 1$. Similarly, to make sure that $\frac{1}{n}\sum_{i=1}^{n} \Pr[\tau \geq i] > 1 - 1/e$, we must set $c_1 < 1$. By fixing an appropriate division point $\rho \in (0, 1)$ and using threshold θ_1 for variables $\{x_i\}_{i \leq \rho n}$, θ_2 for variables $\{x_i\}_{i > \rho n}$, we show that the algorithm does strictly better than $(1 - 1/e)$-competitive. The main challenge lies in characterizing ALG and OPT in terms of the thresholds and the division points, and choosing the parameters that optimize the competitive ratio.

Algorithm. The algorithm has three parameters c_1, c_2 and ρ, where $c_1 < c_2$ controls the quantiles of the two queries and ρ decides the fraction of variables on which we use the first threshold.

[4]The algorithm and analysis framework naturally extends to three or more thresholds, and the details are deferred to Section C in the appendix.

Specifically, let thresholds be $\theta_1 := v(1 - c_1/n)$ and $\theta_1 := v(1 - c_2/n)$ via querying the oracle. We use threshold θ_1 to decide the acceptance of variable x_i for $i \leq \rho \cdot n$ and threshold θ_2 for x_i's with $i \geq \rho \cdot n + 1$. For sufficiently large n, we can assume w.l.o.g. that $\rho \cdot n$ is an integer. We describe the algorithm formally in Algorithm 1.

Algorithm 1: Two-Threshold Algorithm

1 **Input:** Number of items n, parameters c_1, c_2 and ρ.
2 **Output:** A variable selected by the gambler.
3 Set $\theta_1 := v(1 - c_1/n)$ by making a query to the oracle;
4 Set $\theta_2 := v(1 - c_2/n)$ by making another query to the oracle;
5 **for** $i = 1, 2, \ldots, \rho \cdot n$ **do**
6 **if** $x_i \geq \theta_1$ **then**
7 **return** x_i and **terminate**; // phase 1
8 **for** $i = \rho \cdot n + 1, \rho \cdot n + 2, \ldots, n$ **do**
9 **if** $x_i \geq \theta_2$ **then**
10 **return** x_i and **terminate**; // phase 2

We refer to the first for-loop (line 5 - 7) as the first phase of the algorithm and the second for-loop (line 8 - 10) as the second phase. Since $c_1 < c_2$, we have $\theta_1 \geq \theta_2$. Our main goal is to characterize ALG and OPT using a factor revealing linear program (LP), whose optimal objective lower bounds the competitive ratio of the two-threshold algorithm.

Remark. The technique of *factor revealing LPs* was first introduced by [JMM+03] and further extended by [MY11]. Generally speaking, a family of LPs is called factor revealing if the infimum of the optimal values for these LPs serves as a lower bound on the competitive ratio for the original maximization problem. In our case, we construct factor revealing LPs for the adversary of the gambler. After the gambler has established an algorithm to select variables, the goal for the adversary is to find a distribution of the variables such that the gambler's gain is minimized. We approximate the nonlinear optimization problem faced by the adversary with an LP whose objective will be a lower bound for the competitive ratio of the gambler. The objective function and the constraints of the factor revealing LP are formulated using inequalities similar to but more advanced than the ones derived in Section 3.1. We then leverage the power of LP solvers to find the best algorithm for the gambler such that the optimal value of the family of LPs is maximized. In the following subsections, we will show that by carefully choosing the parameters, the optimal value of the LPs (and thus the competitive ratio) for the two-threshold algorithm is at least 0.6786.

3.3 Lower Bounding ALG

We derive a lower bound for the expected gain of the algorithm in the following lemma.

Lemma 3.1 *Let $\epsilon > 0$ be an arbitrarily small constant. For sufficiently large n, we have*

$$\mathsf{ALG} \geq (1 - \epsilon) \cdot \left((1 - e^{-\rho \cdot c_1}) \cdot \left(\theta_1 + \frac{\Delta(c_1)}{c_1} \right) + e^{-\rho \cdot c_1} \cdot (1 - e^{-(1-\rho) \cdot c_2}) \cdot \left(\theta_2 + \frac{\Delta(c_2)}{c_2} \right) \right). \quad (3.6)$$

Proof: Let $\tau \in [n + 1]$ denote the stopping time of the algorithm. Specifically, if $\tau \leq n$, the algorithm terminates after accepting variable x_τ; if $\tau = n + 1$, the algorithm terminates without

accepting any variable. Then we can write the expected gain of the algorithm as

$$\mathsf{ALG} = \sum_{i=1}^{n} \left(\Pr[\tau = i] \cdot \mathbb{E}[x_i \mid \tau = i]\right).$$

Recall that the algorithm has two phases.

1. For all $i \leq \rho \cdot n$, the algorithm accepts variable x_i if and only if $x_j < \theta_1$ and $x_i \geq \theta_1$.
2. For all $i \geq \rho \cdot n + 1$, the algorithm accepts the x_i if and only if (i) $x_j < \theta_1$ for all $j \leq \rho \cdot n$, (ii) $x_j < \theta_2$ for all $j \in [\rho \cdot n + 1, i - 1]$, and (iii) $x_i \geq \theta_2$.

Therefore, we have

$$\Pr[\tau = i] = \begin{cases} F(\theta_1)^{i-1} \cdot G(\theta_1), & \text{if } i \leq \rho \cdot n; \\ F(\theta_1)^{\rho \cdot n} \cdot F(\theta_2)^{i-1-\rho \cdot n} \cdot G(\theta_2), & \text{if } i \geq \rho \cdot n + 1. \end{cases}$$

Similar to Equation (3.5), the expected gain of the algorithm given that it accepts x_i is

$$\mathbb{E}[x_i \mid \tau = i] = \begin{cases} \theta_1 + \frac{\Delta(c_1)/n}{G(\theta_1)}, & \text{if } i \leq \rho \cdot n; \\ \theta_2 + \frac{\Delta(c_2)/n}{G(\theta_2)}, & \text{if } i \geq \rho \cdot n + 1. \end{cases}$$

Given that $F(\theta_1) = 1 - \frac{c_1}{n}$ and $F(\theta_2) = 1 - \frac{c_2}{n}$, the expected gain of the algorithm is

$$\begin{aligned} \mathsf{ALG} &= \sum_{i=1}^{\rho \cdot n} F(\theta_1)^{i-1} \cdot G(\theta_1) \cdot \left(\theta_1 + \frac{\Delta(c_1)/n}{G(\theta_1)}\right) \\ &\quad + \sum_{i=\rho \cdot n+1}^{n} F(\theta_1)^{\rho \cdot n} \cdot F(\theta_2)^{i-1-\rho \cdot n} \cdot G(\theta_2) \cdot \left(\theta_2 + \frac{\Delta(c_2)/n}{G(\theta_2)}\right) \\ &= \frac{1 - (1 - \frac{c_1}{n})^{\rho \cdot n}}{c_1/n} \cdot \frac{c_1}{n} \cdot \left(\theta_1 + \frac{\Delta(c_1)/n}{c_1/n}\right) \\ &\quad + (1 - \frac{c_1}{n})^{\rho \cdot n} \cdot \frac{1 - (1 - \frac{c_2}{n})^{(1-\rho) \cdot n}}{c_2/n} \cdot \frac{c_2}{n} \cdot \left(\theta_2 + \frac{\Delta(c_2)/n}{c_2/n}\right) \\ &\geq (1 - \epsilon) \cdot \left((1 - e^{-\rho \cdot c_1}) \cdot \left(\theta_1 + \frac{\Delta(c_1)}{c_1}\right) + e^{-\rho \cdot c_1} \cdot (1 - e^{-(1-\rho) \cdot c_2}) \cdot \left(\theta_2 + \frac{\Delta(c_2)}{c_2}\right)\right), \end{aligned}$$

where in the last inequality we use $1 + x \leq e^x$ for all $x > 0$ and $1 - \frac{c_1}{n} \geq (1 - \epsilon) \cdot e^{-\frac{c_1}{n}}$, which holds for sufficiently large n, e.g., $n = \omega(\frac{1}{\epsilon})$. ■

3.4 Upper Bounding OPT

Following the argument similar to the one given in Equation (3.2), we obtain the following upper bounds for OPT:

$$\mathsf{OPT} \leq \theta_1 + \Delta(c_1) \quad \text{and} \quad \mathsf{OPT} \leq \theta_2 + \Delta(c_2).$$

However, the above bounds (together with the lower bound on ALG given in Equation (3.6)) alone would not be sufficient to beat $1 - 1/e$. Intuitively speaking, this is because we do not have many constraints on the variables $\theta_1, \theta_2, \Delta(c_1)$ and $\Delta(c_2)$ (except that $\theta_1 \geq \theta_2$). In what follows,

we give a more careful analysis that leads to tighter upper bounds for OPT. We introduce a new variable δ defined as follows:

$$\delta = \int_{\theta_2}^{\theta_1} \Pr[x^* \geq t]dt.$$

Recall that $x^* = \max_{i\in[n]}\{x_i\}$. We have the following upper bound for OPT:

$$\begin{aligned}\mathsf{OPT} = \mathbb{E}[x^*] &= \int_0^\infty \Pr[x^* \geq t]dt \\ &= \int_0^{\theta_2} \Pr[x^* \geq t]dt + \int_{\theta_2}^{\theta_1} \Pr[x^* \geq t]dt + \int_{\theta_1}^{\infty} \Pr[x^* \geq t]dt \\ &\leq \theta_2 + \delta + \mathbb{E}[(x^* - \theta_1)^+] \leq \theta_2 + \delta + \Delta(c_1). \end{aligned} \tag{3.7}$$

In the next three lemmas, we establish three upper bounds for δ, which, when combined with Equation (3.7), will provide much better upper bounds on OPT. Intuitively, the variable δ helps establish some connection between θ_1 and θ_2, and between $\Delta(c_1)$ and $\Delta(c_2)$, which can be utilized to derive more constraints on these variables. Due to space limit, the proofs of Lemmas 3.2 to 3.4 can be found in Appendix D.

Lemma 3.2 *For arbitrarily small $\epsilon > 0$ and sufficiently large n, we have*

$$\delta \leq (1+\epsilon)\cdot(\theta_1 - \theta_2)\cdot(1 - e^{-c_2}). \tag{3.8}$$

Next, we observe a second upper bound for δ achieved by using the union bound.

Lemma 3.3 *We have*

$$\delta \leq \Delta(c_2) - \Delta(c_1). \tag{3.9}$$

So far we have developed two simple upper bounds for δ. In fact, with these two sets of upper bounds, we can already beat $1-1/e$ and achieve a competitive ratio of at least 0.65 by using factor revealing LPs. However, we observe that the ratio can be further improved by introducing a third upper bound[5] on δ.

Lemma 3.4 *Fix any constant $c \in [c_1, c_2]$, let $\gamma := \frac{c-c_1}{e^{-c}-e^{-c_2}}$. For an arbitrarily small constant $\epsilon > 0$ and sufficiently large n, we have*

$$\gamma\cdot\delta/(1+\epsilon) \leq \Delta(c_2) - \Delta(c_1) - (c_1 - \gamma(1-e^{-c}))\cdot(\theta_1 - \theta_2). \tag{3.10}$$

3.5 Lower Bounding the Competitive Ratios

So far, we have established a lower bound for ALG in Section 3.3 and several upper bounds for OPT in Section 3.4. More importantly, given parameters c_1, c_2 and ρ, these bounds are linear in $\theta_1, \theta_2, \Delta(c_1), \Delta(c_2)$ and δ. We can now construct a minimization LP taking Equation (3.6) as the objective and Equations (3.7), (3.8), (3.9), (3.10) as constraints with $\theta_1, \theta_2, \Delta(c_1), \Delta(c_2)$ and δ being the non-negative LP variables. In the following theorem, we prove that the optimal objective of the LP gives a lower bound on the competitive ratio for the algorithm.

[5] As we will discuss in Section 5, there is still room for improvement for this upper bound.

Theorem 3.2 *Given parameters c_1, c_2 and ρ of the two-threshold algorithm, the optimal value of the following LP provides a lower bound for the competitive ratio of the algorithm when $n \to \infty$.*

$$\begin{aligned}
\textit{minimize} \quad & (1 - e^{-\rho \cdot c_1}) \cdot \left(\theta_1 + \tfrac{\Delta(c_1)}{c_1}\right) + e^{-\rho \cdot c_1} \cdot (1 - e^{-(1-\rho)\cdot c_2}) \cdot \left(\theta_2 + \tfrac{\Delta(c_2)}{c_2}\right) \\
\textit{subject to} \quad & \theta_1 \geq \theta_2 \geq 0, \quad \Delta(c_1) \geq 0, \quad \delta \geq 0, \\
& 1 \leq \theta_2 + \delta + \Delta(c_1), \\
& \delta \leq (\theta_1 - \theta_2) \cdot (1 - e^{-c_2}), \\
& \delta \leq \Delta(c_2) - \Delta(c_1), \\
& \gamma \cdot \delta \leq \Delta(c_2) - \Delta(c_1) - (c_1 - \gamma(1 - e^{-c})) \cdot (\theta_1 - \theta_2), \ \forall c \in [c_1, c_2], \ \gamma = \tfrac{c - c_1}{e^{-c} - e^{-c_2}}.
\end{aligned}$$

Proof: The objective of the LP comes from Equation (3.6), the lower bound for ALG, where we omit the $(1-\epsilon)$ term since we have $\epsilon \to 0$ when $n \to \infty$. We will also omit the $(1+\epsilon)$ terms in the following for the same reason. The first set of constraints follows straightforwardly from the definitions of the parameters. By scaling we can assume w.l.o.g. that $\mathsf{OPT} = 1$. Note that our algorithm does not need to know the value of OPT to decide the parameters. Therefore, Equation (3.7) gives the second constraint. The three proceeding sets of constraints follow from Equations (3.8), (3.9) and (3.10), the three upper bounds on δ that we have established in Section 3.4.

Observe that every distribution D induces a set of variables $\{\theta_1, \theta_2, \Delta(c_1), \Delta(c_2), \delta\}$, which will form a feasible solution to the LP as they must abide by the corresponding constraints. Since the objective of any feasible solution induced by distribution D provides a lower bound on ALG/OPT, the optimal (minimum) objective of the LP provides a lower bound on the competitive ratio, i.e., the worst-case performance against all distributions. ■

With the above result, it remains to set the appropriate parameters of the two-threshold algorithm such that the optimal value of the LP is as large as possible.

Theorem 3.3 *The two-threshold algorithm achieves a competitive ratio of at least* 0.6786*, with parameters $c_1 = 0.7067$, $c_2 = 1.8353$, and $\rho = 0.6204$.*

Proof: By Theorem 3.2, given the values of c_1, c_2, and ρ, the objective value of the factor revealing LP bounds the competitive ratio from below. Using an online LP solver[6], it can be verified that the optimal value of the LP is at least 0.6786. ■

3.6 An Upper Bound for Two-Threshold Algorithms

Finally, we complement our algorithmic results with an upper bound of 0.7081 on the competitive ratio for any algorithm that uses two thresholds from two queries. Specifically, we construct three distributions and prove that any such algorithm, parameterized by c_1, c_2 and ρ, cannot achieve a competitive ratio better than 0.7081 in all three distributions.

Lemma 3.5 *Any algorithm that defines two thresholds by making two queries cannot achieve a competitive ratio better than* 0.7081*.*

4 Beating $1 - 1/e$ using a Single Query

In this section, we show that it is possible to beat the competitive ratio of $1-1/e$ using just one query. Our algorithm combines the single-threshold algorithm and the observe-then-accept algorithm for

[6]Online LP solver: https://online-optimizer.appspot.com/.

the secretary problem. Let $c \in [0, n]$ and $\rho \in [0, 1]$ be the parameters of the algorithm. Again, for sufficiently large n we can assume w.l.o.g. that $\rho \cdot n$ is an integer. The algorithm, denoted by *observe-and-accept*, works as follows. Let $\theta_1 = v(1 - c/n)$. We have $\Pr_{x \sim D}[x \geq \theta_1] = G(\theta_1) = c/n$. In the quantile-based phase, for $i = 1, 2, \ldots, \rho \cdot n$, we accept x_i if and only if $x_i \geq \theta_1$. If no variables are accepted in the quantile-based phase, we enter the observation-based phase and let $\theta_2 = \max_{i \leq \rho \cdot n}\{x_i\}$. Note that θ_2 is a random variable and $\theta_2 < \theta_1$. For $i = \rho \cdot n + 1, \ldots, n$, we accept x_i if and only if $x_i \geq \theta_2$ (see Algorithm 2).

Algorithm 2: Observe-and-Accept

```
Input: Number of items n, quantile choice c and observation portion ρ.
Output: A variable selected by the gambler.
Set θ1 := v(1 − c/n) by making a query to the oracle; for i = 1, 2, ..., ρ·n do
    if x_i ≥ θ1 then
        return x_i and terminate;          // quantile-based phase
Set θ2 := max_{i≤ρ·n}{x_i};
for i = ρ·n + 1, ..., n do
    if x_i ≥ θ2 then
        return x_i and terminate;          // observation-based phase
```

Theorem 4.1 *There exists a* 0.6718*-competitive algorithm for the prophet inequality problem on unknown i.i.d. distributions that makes a single query to the distribution oracle.*

We prove the theorem in the following two subsections.

4.1 Bounding ALG and OPT

We first give lower bounds for ALG and upper bounds for OPT, which will then become the constraints of the factor revealing LP we study in the next subsection. Recall Equation (3.3) from Section 3, in which we have $\mathsf{ALG} = \sum_{i=1}^{n} \Pr[\tau = i] \cdot \mathbb{E}[x_i \mid \tau = i]$, where $\tau \in [n+1]$ is the stopping time of the algorithm. For $i \leq \rho \cdot n$, the algorithm accepts variable x_i if its realization is at least θ_1, which happens with probability c/n. By a similar analysis seen in Equations (3.4) and (3.5), we have

$$\Pr[\tau = i] = \left(1 - \frac{c}{n}\right)^{i-1} \cdot \frac{c}{n}, \quad \forall i \leq \rho \cdot n, \qquad \text{and}$$
$$\mathbb{E}[x_i \mid \tau = i] = \theta_1 + \frac{\Delta(c)}{c}, \quad \forall i \leq \rho \cdot n.$$

Therefore, the expected gain of the algorithm from the quantile-based phase is given by

$$\begin{aligned}
\Lambda_1 &:= \sum_{i=1}^{\rho n} \Pr[\tau = i] \cdot \mathbb{E}[x_i \mid \tau = i] = \sum_{i=1}^{\rho n} (1 - \frac{c}{n})^{i-1} \cdot \left(\frac{c}{n} \cdot \theta_1 + \frac{\Delta(c)}{n}\right) \\
&= \frac{1 - (1 - c/n)^{\rho n}}{c/n} \cdot \left(\frac{c}{n} \cdot \theta_1 + \frac{\Delta(c)}{n}\right) = \left(1 - (1 - \frac{c}{n})^{\rho n}\right) \cdot \left(\theta_1 + \frac{\Delta(c)}{c}\right) \\
&\geq \left(1 - e^{-c\rho}\right) \cdot \left(\theta_1 + \frac{\Delta(c)}{c}\right). \qquad (4.1)
\end{aligned}$$

Next, we give a lower bound for the expected gain of the algorithm from the observation-based phase, which is given by $\Lambda_2 := \sum_{i=\rho n+1}^{n} \Pr[\tau = i] \cdot \mathbb{E}[x_i \mid \tau = i]$. Unlike the quantile-based phase, the threshold $\theta_2 = \max_{i \le \rho n}\{x_i\}$ is a random variable and thus $F(\theta_2)$ is not a fixed probability. Our algorithm enters the observation-based phase if and only if $\theta_2 < \theta_1$, i.e., no variable passes the threshold θ_1 in the quantile-based phase. Note that

$$\Pr[\theta_2 < \theta_1] = \Pr[\forall i \le \rho n, x_i < \theta_1] = (1 - \frac{c}{n})^{\rho n} \ge (1 - \epsilon) \cdot e^{-c\rho},$$

where $\epsilon > 0$ is an arbitrarily small constant given that n is sufficiently large. Moreover, given a realization of $\theta_2 < \theta_1$, we can express the gain of the algorithm $h(\theta_2)$ in the observation-based phase using a similar argument as above. Specifically, conditioned on a given $\theta_2 < \theta_1$, we have

$$h(\theta_2) := \sum_{i=\rho n+1}^{n} \Pr[\tau = i \mid \tau > \rho n] \cdot \mathbb{E}[x_i \mid \tau = i] = \sum_{i=1}^{(1-\rho)n} F(\theta_2)^{i-1} \cdot \left(G(\theta_2) \cdot \theta_2 + \mathbb{E}[(x - \theta_2)^+]\right).$$

In summary, we can lower bound the expected total gain of the algorithm by

$$\begin{aligned}
\mathsf{ALG} &= \left(1 - (1 - \frac{c}{n})^{\rho n}\right) \cdot \left(\theta_1 + \frac{\Delta(c)}{c}\right) + \left(1 - \frac{c}{n}\right)^{\rho n} \cdot \mathbb{E}_{\theta_2}[h(\theta_2) \mid \theta_2 < \theta_1] \\
&\ge \left(1 - e^{-c\rho}\right) \cdot \left(\theta_1 + \frac{\Delta(c)}{c}\right) + (1 - \epsilon) \cdot e^{-c\rho} \cdot \mathbb{E}_{\theta_2}[h(\theta_2) \mid \theta_2 < \theta_1].
\end{aligned}$$

Unfortunately, it is challenging to derive a closed-form expression of the conditional expectation $\mathbb{E}_{\theta_2}[h(\theta_2) \mid \theta_2 < \theta_1]$ in terms of c and ρ. To overcome this difficulty, we relax this term and lower bound it using a factor revealing LP. Specifically, consider the case when

$$v(1 - \frac{\beta_2}{n}) \le \theta_2 \le v(1 - \frac{\beta_1}{n}),$$

where β_1, β_2 are constants satisfying $c < \beta_1 < \beta_2$. Recall that $F(v)$ and $v(q) = F^{-1}(q)$ are increasing functions of v and q, and $G(v)$ and $\mathbb{E}[(x - \theta)^+]$ are decreasing functions of v and θ. We can lower bound $h(\theta_2)$ by

$$\begin{aligned}
h(\theta_2) &= \sum_{i=1}^{(1-\rho)n} F(\theta_2)^{i-1} \cdot \left(G(\theta_2) \cdot \theta_2 + \mathbb{E}[(x - \theta_2)^+]\right) \\
&\ge \sum_{i=1}^{(1-\rho)n} F\left(v(1 - \frac{\beta_2}{n})\right)^{i-1} \cdot \left\{G\left(v(1 - \frac{\beta_1}{n})\right) \cdot v(1 - \frac{\beta_2}{n}) + \mathbb{E}\left[\left(x - v(1 - \frac{\beta_1}{n})\right)^+\right]\right\} \\
&\ge \sum_{i=1}^{(1-\rho)n} \left(1 - \frac{\beta_2}{n}\right)^{i-1} \cdot \left(\frac{\beta_1}{n} \cdot v(1 - \frac{\beta_2}{n}) + \frac{\Delta(\beta_1)}{n}\right) \\
&\ge \frac{1 - \left(1 - \frac{\beta_2}{n}\right)^{(1-\rho)n}}{\beta_2/n} \cdot \left(\frac{\beta_1}{n} \cdot v(1 - \frac{\beta_2}{n}) + \frac{\Delta(\beta_1)}{n}\right) \\
&\ge \left(1 - \left(1 - \frac{\beta_2}{n}\right)^{(1-\rho)n}\right) \cdot \left(\frac{\beta_1}{\beta_2} \cdot v(1 - \frac{\beta_2}{n}) + \frac{\Delta(\beta_1)}{\beta_2}\right) \\
&\ge \left(1 - e^{-\beta_2(1-\rho)}\right) \cdot \left(\frac{\beta_1}{\beta_2} \cdot v(1 - \frac{\beta_2}{n}) + \frac{\Delta(\beta_1)}{\beta_2}\right) := H(\beta_1, \beta_2).
\end{aligned}$$

The important observation here is that $H(\beta_1, \beta_2)$ depends only on c, β_1, β_2, n and the distribution D, and is independent of θ_2. Moreover, it is linear in $v(1-\beta_2/n)$ and $\Delta(\beta_1)$. Note that conditioned on $\theta_2 < \theta_1$, the event of $v(1-\beta_2/n) \le \theta_2 \le v(1-\beta_1/n)$, where $c < \beta_1 < \beta_2$, happens with probability

$$\begin{aligned}
&\Pr\left[v(1-\frac{\beta_2}{n}) \le \theta_2 \le v(1-\frac{\beta_1}{n}) \mid \theta_2 < \theta_1\right] \\
&= \frac{\Pr[v(1-\frac{\beta_2}{n}) \le \theta_2 \le v(1-\frac{\beta_1}{n})]}{\Pr[\theta_2 < \theta_1]} = \frac{(1-\frac{\beta_1}{n})^{\rho n} - (1-\frac{\beta_2}{n})^{\rho n}}{(1-\frac{c}{n})^{\rho n}} := p(\beta_1, \beta_2).
\end{aligned}$$

Therefore, for any sequence $c = \beta_0 < \beta_1 < \beta_2 < \cdots < \beta_k$ and sufficiently large n, we can lower bound the gain of the observation-based phase by

$$\begin{aligned}
\Lambda_2 &\ge \left(1-\frac{c}{n}\right)^{\rho n} \cdot \sum_{i=1}^{k} p(\beta_{i-1}, \beta_i) \cdot H(\beta_{i-1}, \beta_i) \\
&= \sum_{i=1}^{k} \left((1-\frac{\beta_{i-1}}{n})^{\rho n} - (1-\frac{\beta_i}{n})^{\rho n}\right) \cdot H(\beta_{i-1}, \beta_i) \\
&\ge (1-\epsilon) \sum_{i=1}^{k} \left(e^{-\beta_{i-1}\rho} - e^{-\beta_i \rho}\right) \cdot \left(1 - e^{-\beta_i(1-\rho)}\right) \cdot \left(\frac{\beta_{i-1}}{\beta_i} \cdot v(1-\frac{\beta_i}{n}) + \frac{\Delta(\beta_{i-1})}{\beta_i}\right). \qquad (4.2)
\end{aligned}$$

So far, we have given a lower bound for ALG in Equations (4.1) and (4.2). In the following, we give upper bounds for OPT. Following similar analyses as in Section 3.2, we define

$$\delta_i = \int_{v(1-\frac{\beta_{i+1}}{n})}^{v(1-\frac{\beta_i}{n})} \Pr[x \ge t] dt, \qquad \forall 0 \le i \le k-1.$$

Then we have

$$\begin{aligned}
\mathsf{OPT} &= \int_0^{v(1-\frac{\beta_k}{n})} \Pr[x^* \ge t] dt + \sum_{i=0}^{k-1} \int_{v(1-\frac{\beta_{i+1}}{n})}^{v(1-\frac{\beta_i}{n})} \Pr[x^* \ge t] dt + \int_{v(1-\frac{\beta_0}{n})}^{\infty} \Pr[x^* \ge t] dt \\
&\le v(1-\frac{\beta_k}{n}) + \sum_{i=0}^{k-1} \delta_i + \Delta(\beta_0).
\end{aligned}$$

Following the same analyses for proving Lemmas 3.2, 3.3 and 3.4, we obtain the following: for arbitrarily small $\epsilon > 0$ and sufficiently large n and for all $i \in \{0, 1, \ldots, k-1\}$,

$$\begin{aligned}
&\delta_i \le (1+\epsilon) \cdot (1 - e^{-\beta_{i+1}}) \cdot \left(v(1-\frac{\beta_i}{n}) - v(1-\frac{\beta_i+1}{n})\right), \\
&\delta_i \le \Delta(\beta_{i+1}) - \Delta(\beta_i).
\end{aligned}$$

For all $\zeta \in [\beta_i, \beta_{i+1}]$ and $\gamma = \frac{\zeta - \beta_i}{e^{-\zeta} - e^{-\beta_{i+1}}}$, we have

$$\gamma \cdot \delta_i/(1+\epsilon) \le \Delta(\beta_{i+1}) - \Delta(\beta_i) - (\beta_i - \gamma(1-e^{-\zeta})) \cdot \left(v(1-\frac{\beta_i}{n}) - v(1-\frac{\beta_i+1}{n})\right).$$

4.2 Factor Revealing LP

We finish the proof of Theorem 4.1 by formulating the lower and upper bounds into a factor revealing LP and choosing appropriate parameters to obtain a lower bound of 0.6718 on the competitive ratio. In the following, we fix $\beta_i = \beta^i \cdot c$ for all $i = 0, 1, \ldots, k$, for some $\beta > 1$. For ease of notation, we use $v_i = v(1 - \beta_i/n)$ and $\Delta_i = \Delta(\beta_i)$, for all $i = 0, 1, \ldots, k$. Note that $c = \beta_0$, $\theta_1 = v(1 - \beta_0/n) = v_0$ and $\Delta(c) = \Delta_0$. Recall that $v_0 \geq v_1 \geq \cdots \geq v_k$ and $\Delta_0 \leq \Delta_1 \leq \cdots \leq \Delta_k$. By fixing c and k, we uniquely define a sequence $\beta_0, \beta_1, \ldots, \beta_k$. However, the values of $\{v_i, \Delta_i\}_{0 \leq i \leq k}$ depend on the unknown distribution D. As argued in Section 3.5, by taking the lower bound on ALG as the objective and upper bounds on OPT as constraints, we obtain an LP whose optimal value provides a lower bound for the competitive ratio of the algorithm. The LP variables are $\{v_i, \Delta_i\}_{0 \leq i \leq k}$ and $\{\delta_i\}_{0 \leq i < k}$.

Recall that we have lower bounded $\mathsf{ALG} = \Lambda_1 + \Lambda_2$, where

$$\Lambda_1 \geq \left(1 - e^{-\beta_0 \rho}\right) \cdot (v_0 + \Delta_0/\beta_0),$$

$$\Lambda_2 \geq (1 - \epsilon) \sum_{i=1}^{k} \left(e^{-\beta_{i-1}\rho} - e^{-\beta_i \rho}\right) \cdot \left(1 - e^{-\beta_i(1-\rho)}\right) \cdot ((\beta_{i-1}/\beta_i) \cdot v_i + \Delta_{i-1}/\beta_i),$$

are both linear in $\{v_i, \Delta_i\}_{0 \leq i \leq k}$. We have also upper bounded OPT by

$$\mathsf{OPT} \leq v(1 - \frac{\beta_k}{n}) + \sum_{i=0}^{k-1} \delta_i + \Delta_0,$$

and each δ_i, where $i \in \{0, 1, \ldots, k-1\}$, by (recall that $\gamma = \frac{\zeta - \beta_i}{e^{-\zeta} - e^{-\beta_{i+1}}}$)

$$\begin{aligned}
\delta_i &\leq (1 + \epsilon) \cdot (1 - e^{-\beta_{i+1}}) \cdot (v_i - v_{i+1}), \\
\delta_i &\leq \Delta(\beta_{i+1}) - \Delta(\beta_i), \\
\gamma \cdot \delta_i/(1 + \epsilon) &\leq \Delta_{i+1} - \Delta_i - (\beta_i - \gamma(1 - e^{-\zeta})) \cdot (v_i - v_{i+1}), \quad \forall \zeta \in [\beta_i, \beta_{i+1}].
\end{aligned}$$

Again, we can assume w.l.o.g. that $\mathsf{OPT} = 1$. Since $\epsilon > 0$ can be arbitrarily small when $n \to \infty$, the terms $(1 - \epsilon)$ and $(1 + \epsilon)$ can be removed. The factor revealing LP is as follows.

$$\begin{aligned}
\text{minimize} \quad & \Lambda_1 + \Lambda_2 \\
\text{subject to} \quad & \Lambda_1 \geq (1 - e^{-\beta_0 \rho}) \cdot (v_0 + \Delta_0/\beta_0), \\
& \Lambda_2 \geq \textstyle\sum_{i=1}^{k} \left(e^{-\beta_{i-1}\rho} - e^{-\beta_i \rho}\right) \cdot \left(1 - e^{-\beta_i(1-\rho)}\right) \cdot ((\beta_{i-1}/\beta_i) \cdot v_i + \Delta_{i-1}/\beta_i), \\
& 1 \geq v_k + \textstyle\sum_{i=0}^{k-1} \delta_i + \Delta_0 \\
& \delta_i \leq \left(1 - e^{-\beta_{i+1}}\right) \cdot (v_i - v_{i+1}), \qquad \forall 0 \leq i \leq k-1 \\
& \delta_i \leq \Delta_{i+1} - \Delta_i, \qquad \forall 0 \leq i \leq k-1 \\
& \gamma \cdot \delta_i \leq \Delta_{i+1} - \Delta_i - (\beta_i - \gamma(1 - e^{-\zeta})) \cdot (v_i - v_{i+1}), \quad \forall 0 \leq i \leq k-1, \zeta \in [\beta_i, \beta_{i+1}] \\
& v_0 \geq v_1 \geq \cdots \geq v_k \geq 0, \\
& \Delta_0, \delta_0, \ldots, \delta_{k-1} \geq 0,
\end{aligned}$$

where ρ, $\{\beta_i\}_{0 \leq i \leq k}$ and $\gamma = \frac{\zeta - \beta_i}{e^{-\zeta} - e^{-\beta_{i+1}}}$ are constants; Λ_1, Λ_2, $\{v_i, \Delta_i\}_{0 \leq i \leq k}$, $\{\delta_i\}_{0 \leq i < k}$ are variables of the LP. As argued, every distribution D induces a feasible solution to the LP. Therefore, the optimal value of the LP provides a lower bound for the competitive ratio of our algorithm when $n \to \infty$.

The following claim is verified by the online LP solver, which completes the proof of Theorem 4.1.

Claim 4.1 *By fixing constants* $c = 0.72941, \rho = 0.64863$, $k = 100$ *and* $\beta_i = 1.03^i \cdot c$ *for all* $0 \le i \le k$, *the optimal solution to the above LP has objective at least* 0.6718.

Remark Suppose we are given a fixed quantile (i.e., a fixed c) instead of being able to choose one, we may still beat $1 - 1/e$. Let us revisit the example in Figure 1.1 in the following updated Figure 4.1. By setting $k = 100$ and $\beta_i = 1.03^i \cdot c$ for all $0 \le i \le k$, we compute the corresponding separation point ρ (the green-cross curve) and the optimal reward using the above LP (the blue-dot curve). We see that for any c between 0.5 and 1, the competitive ratio is better than $1 - 1/e$. This suggests that if we are not allowed to make queries but are given a fixed quantile, as long as it is within this range, the observe-and-accept algorithm can still do strictly better than the optimal single-threshold algorithm. Taking a closer look at the cases when $c > 1$, we see that the optimal ρ is always 1, which means the algorithm only has the quantile-based phase using $v(1 - c/n)$ as the threshold. This is because for $c > 1$, $v(1 - c/n)$ is already smaller than the threshold used in the optimal single-threshold algorithm, and thus it is worse off to add the observation-based phase whose threshold is even smaller than $v(1 - c/n)$. Finally, it is clear to note that if the given c is very large, which means that the threshold for the quantile-based phase is too small, the observe-and-accept algorithm may accept a very small value and perform badly. As shown in Figure 4.1, when c is greater than 2.5, the competitive ratio for observe-and-accept is less than $1/e$.

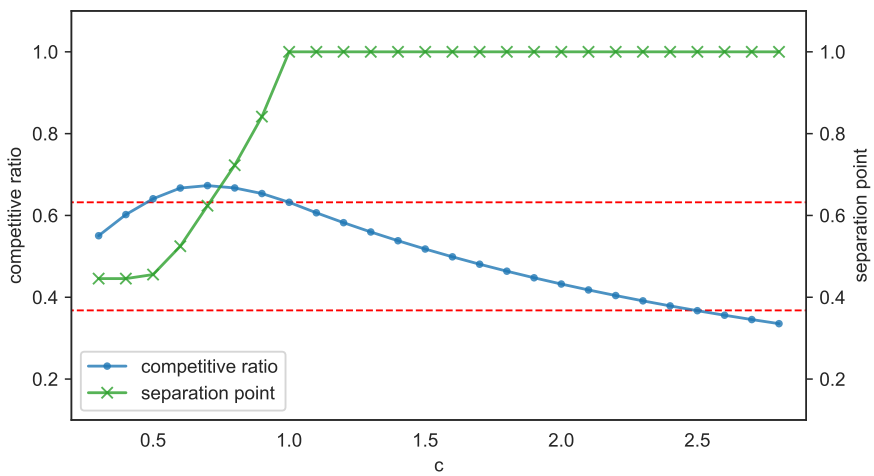

Figure 4.1: Competitive ratios, as well as the corresponding best separation point ρ, for different quantiles at $1 - c/n$, by setting $k = 100$ and $\beta_i = 1.03^i \cdot c$ for all $0 \le i \le k$. (Figure 1.1 revisited).

4.3 Upper Bound on the Competitive Ratio of the Algorithm

To complement our lower bound on the competitive ratio of the algorithm, we also explore upper bounds for this type of algorithm. In particular, we show the following.

Lemma 4.1 *No observe-and-accept algorithm can do better than* 0.6921*-competitive.*

Proof: Consider any observe-and-accept algorithm with parameters c and ρ. In the following, we construct two distributions D_1 and D_2, and show that the performance of the algorithm on one of the two distributions is at most 0.6921. Note that by construction both distributions have strictly increasing CDF and satisfy $\mathsf{OPT} = 1$.

Let the first distribution D_1 be a uniform distribution over $[1-\epsilon, 1+\epsilon]$, for some arbitrarily small $\epsilon > 0$. Then we have

$$\begin{aligned}\mathsf{ALG} &\leq 1+\epsilon - \Pr[\tau > n] = 1+\epsilon - \Pr[\forall i \leq \rho n, x_i < \theta_1 \text{ and } \forall i > \rho n, x_i < \max_{j \leq \rho n}\{x_j\}] \\ &= 1+\epsilon - (1-\frac{c}{n})^n \cdot \rho \leq 1+\epsilon - \rho \cdot e^{-c}, \end{aligned} \tag{4.3}$$

where the last equality holds because $\tau > n$ happens if and only if all variables are below θ_1 and the maximum of them appears in the first $\rho \cdot n$ variables.

Let the second distribution D_2 be as follows. Let $M \gg n$ be an arbitrarily large number and $\epsilon > 0$ be arbitrarily small. With probability $\frac{1}{n \cdot M}$, x is uniformly distributed over $[M-\epsilon, M+\epsilon]$; with probability $1 - \frac{1}{n \cdot M}$, x is uniformly distributed over $[0, \epsilon]$. Note that when $M \to \infty$ and $\epsilon \to 0$,

$$\mathsf{OPT} \approx M \cdot \left(1 - (1 - \frac{1}{n \cdot M})^n\right) = M \cdot \frac{1}{M} = 1.$$

Observe that for distribution D_2, we have $\theta_1 \leq \epsilon$ and $\Delta(c) = n \cdot \mathbb{E}[(x-\theta_1)^+] \approx 1$. Since θ_1 is close to 0, the gain of the algorithm is determined by how likely the algorithm accepts a variable with a value close to M. Therefore, we can upper bound ALG by

$$\mathsf{ALG} < \epsilon + M \cdot \sum_{i=1}^{n} \Pr[\tau > i-1] \cdot \Pr[x_i \geq M - \epsilon].$$

Note that $\Pr[x_i \geq M - \epsilon] = \frac{1}{n \cdot M}$ for all i. For $i \leq \rho n$, we have

$$\Pr[\tau > i-1] = (1 - \frac{c}{n})^{i-1}.$$

For $i > \rho n$, the event $\tau > i-1$ happens if and only if all variables $x_1, \ldots, x_{i-1}$ are below θ_1, and the maximum of them appears before index ρn. Hence,

$$\Pr[\tau > i-1] = (1 - \frac{c}{n})^{i-1} \cdot \frac{\rho n}{i-1}.$$

Putting everything together, as $n \to \infty$, we get

$$\begin{aligned}\mathsf{ALG} &< \epsilon + \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr[\tau > i-1] = \epsilon + \frac{1}{n} \cdot \sum_{i=1}^{\rho n} (1 - \frac{c}{n})^{i-1} + \frac{1}{n} \cdot \sum_{i=\rho n+1}^{n} (1 - \frac{c}{n})^{i-1} \cdot \frac{\rho n}{i-1} \\ &= \epsilon + \frac{1 - (1 - \frac{c}{n})^{\rho n}}{c} + \sum_{i=\rho n+1}^{n} (1 - \frac{c}{n})^{i-1} \cdot \frac{\rho}{i-1} \approx \epsilon + \frac{1 - e^{-c\rho}}{c} + \rho \cdot \int_{\rho}^{1} \frac{e^{-ct}}{t} dt. \end{aligned} \tag{4.4}$$

It is not difficult to check that when $c > 1$ the above upper bound is less than $1 - 1/e$. Thus any observe-and-accept algorithm with a competitive ratio $> 1 - 1/e$ must have $c < 1$. It remains to show that when $\epsilon \to 0$, $n \to \infty$ and $M \to \infty$, for any $c, \rho \in [0, 1]$, at least one of the two upper bounds given in Equations (4.3) and (4.4) has a value less than 0.6921:

$$\max_{c,\rho \in [0,1]} \left\{ \min \left\{ 1 - \rho \cdot e^{-c}, \frac{1 - e^{-c\rho}}{c} + \rho \cdot \int_{\rho}^{1} \frac{e^{-ct}}{t} dt \right\} \right\} \leq 0.6921. \tag{4.5}$$

Using a similar argument as in Section 3.6, we prove Equation (4.5) by discretizing the domain for c and ρ and using computational tools. For example, letting $I = \{\frac{i}{10^5} : 1 \leq i \leq 10^5\}$, we

can check that the LHS of Equation (4.5) can be approximated by replacing "$c, \rho \in [0, 1]$" with "$c, \rho \in I$", with an additive error of at most 3×10^{-5}. By enumerating all $c, \rho \in I$ and recording the maximum, we get an upper bound of 0.69204, which is achieved when $c = 0.37476$ and $\rho = 0.44799$. Combining the above discretized upper bound with the additive error of 3×10^{-5} gives an upper bound of $0.69207 < 0.6921$. ■

5 Conclusion and Open Problems

In this work, we study the single-choice prophet inequality problem with unknown i.i.d. distributions. We propose a new model in which an algorithm has access to an oracle that answers quantile queries about the distribution, with which we implement the multi-threshold blind strategies and show that with $k \geq 2$ thresholds, the competitive ratio is at least 0.6786. Moreover, we demonstrate that with queries, one can do more than just implementations of blind strategies, by proposing an algorithm that uses a single query to achieve a competitive ratio of 0.6718.

Our work uncovers many interesting open problems and future research directions.

The first direction is to improve the competitive ratios and the upper bounds. We believe that our analysis for the competitive ratio does not fully release the potential of the proposed algorithms, and improvements on the lower bounds of the competitive ratios are possible. In particular, our current technique of factor revealing LP hinders us from deriving tighter bounds on ALG and OPT. For example, the upper bound on δ we present in Lemma 3.4 is only a linear approximation of a stronger and more general (non-linear) bound. In order to formulate this upper bound into a linear constraint of the LP, we have to relax it into a linear form. It would be very interesting to investigate how much the competitive ratio can be improved if the analysis is able to incorporate such non-linear constraints.

Another natural open problem is to study more general forms of the observe-and-accept algorithm. For example, one can generalize the algorithm from a single query to multiple queries. Based on our current results and analyses, we believe that with two queries, it is possible to derive algorithms with a competitive ratio strictly above 0.68. One can also investigate alternative ways to utilize the queries. For instance, is it beneficial to introduce more phases for the single-query algorithm? That is, the first phase uses the threshold derived from a query to the oracle while each later phase uses the maximum realization from the previous phase as the threshold. Furthermore, if having more phases helps, what is the best possible competitive ratio one can achieve with a single query?

Finally, it is interesting to investigate the prophet secretary problem under the query model. In the prophet secretary problem, the n variables $x_1, x_2, \ldots, x_n$ are drawn from n (possibly different) distributions $D_1, D_2, \ldots, D_n$, and they arrive following a uniformly-at-random chosen order. The problem generalizes the prophet inequality problem on i.i.d. distributions. It can be shown that a single query on the distribution D^* of $x^* = \max_{1 \leq i \leq n}\{x_i\}$ suffices to achieve a competitive ratio of $1 - 1/e$. Whether we can beat $1 - 1/e$ with a single query on D^* would be an interesting open problem to investigate.

References

[ABB21] Amine Allouah, Achraf Bahamou, and Omar Besbes. Optimal pricing with a single point. In *EC*, page 50. ACM, 2021.

[ACK18] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Prophet secretary: Surpassing the 1-1/e barrier. In *EC*, pages 303–318. ACM, 2018.

[AEE+17] Melika Abolhassani, Soheil Ehsani, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Robert D. Kleinberg, and Brendan Lucier. Beating 1-1/e for ordered prophets. In *STOC*, pages 61–71. ACM, 2017.

[AKW19] Pablo Daniel Azar, Robert Kleinberg, and S. Matthew Weinberg. Prior independent mechanisms via prophet inequalities with limited information. *Games Econ. Behav.*, 118:511–532, 2019.

[BC22] Archit Bubna and Ashish Chiplunkar. Prophet inequality: Order selection beats random order. *CoRR*, abs/2211.04145, 2022.

[CDF+21] Constantine Caramanis, Paul Dütting, Matthew Faw, Federico Fusco, Philip Lazos, Stefano Leonardi, Orestis Papadigenopoulos, Emmanouil Pountourakis, and Rebecca Reiffenhäuser. Single-sample prophet inequalities via greedy-ordered selection. *CoRR*, abs/2111.03174, 2021.

[CDFS19] José R. Correa, Paul Dütting, Felix A. Fischer, and Kevin Schewior. Prophet inequalities for I.I.D. random variables from an unknown distribution. In *EC*, pages 3–17. ACM, 2019.

[CFH+17] José R. Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Posted price mechanisms for a random stream of customers. In *EC*, pages 169–186. ACM, 2017.

[CFH+18] José R. Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Recent developments in prophet inequalities. *SIGecom Exch.*, 17(1):61–70, 2018.

[CHMS10] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Multi-parameter mechanism design and sequential posted pricing. In *STOC*, pages 311–320. ACM, 2010.

[CLLL22] Jing Chen, Bo Li, Yingkai Li, and Pinyan Lu. Bayesian auctions with efficient queries. *Artif. Intell.*, 303:103630, 2022.

[CSZ19] José R. Correa, Raimundo Saona, and Bruno Ziliotto. Prophet secretary through blind strategies. In *SODA*, pages 1946–1961. SIAM, 2019.

[DFKL20] Paul Dütting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet inequalities made easy: Stochastic optimization by pricing nonstochastic inputs. *SIAM J. Comput.*, 49(3):540–582, 2020.

[DK15] Paul Dütting and Robert Kleinberg. Polymatroid prophet inequalities. In *ESA*, volume 9294 of *Lecture Notes in Computer Science*, pages 437–449. Springer, 2015.

[DK19] Paul Dütting and Thomas Kesselheim. Posted pricing and prophet inequalities with inaccurate priors. In *EC*, pages 111–129. ACM, 2019.

[DKL20] Paul Dütting, Thomas Kesselheim, and Brendan Lucier. An o(log log m) prophet inequality for subadditive combinatorial auctions. In *FOCS*, pages 306–317. IEEE, 2020.

[EHKS18] Soheil Ehsani, MohammadTaghi Hajiaghayi, Thomas Kesselheim, and Sahil Singla. Prophet secretary for combinatorial auctions and matroids. In *SODA*, pages 700–714. SIAM, 2018.

[EHLM15] Hossein Esfandiari, MohammadTaghi Hajiaghayi, Vahid Liaghat, and Morteza Monemizadeh. Prophet secretary. In *ESA*, volume 9294 of *Lecture Notes in Computer Science*, pages 496–508. Springer, 2015.

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

[HHSW21] Yihang Hu, Zhiyi Huang, Yiheng Shen, and Xiangning Wang. Targeting makes sample efficiency in auction design. In *EC*, pages 610–629. ACM, 2021.

[HK82] Theodore P Hill and Robert P Kertz. Comparisons of stop rule and supremum expectations of iid random variables. *The Annals of Probability*, pages 336–345, 1982.

[HKS07] Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI*, pages 58–65. AAAI Press, 2007.

[JMM+03] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50(6):795–824, 2003.

[Ker86] Robert P Kertz. Stop rule and supremum expectations of iid random variables: a complete comparison by conjugate duality. *Journal of multivariate analysis*, 19(1):88–112, 1986.

[KS77] Ulrich Krengel and Louis Sucheston. Semiamarts and finite values. *Bulletin of the American Mathematical Society*, 83(4):745–747, 1977.

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

[KW12] Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities. In *STOC*, pages 123–136. ACM, 2012.

[LSTW21] Renato Paes Leme, Balasubramanian Sivan, Yifeng Teng, and Pratik Worah. Pricing query complexity of revenue maximization. *CoRR*, abs/2111.03158, 2021.

[Luc17] Brendan Lucier. An economic view of prophet inequalities. *SIGecom Exch.*, 16(1):24–47, 2017.

[MY11] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 597–606. ACM, 2011.

[NV22] Pranav Nuti and Jan Vondrák. Secretary problems: The power of a single sample. *CoRR*, abs/2208.09159, 2022.

[PST22] Sebastian Perez-Salazar, Mohit Singh, and Alejandro Toriello. The IID prophet inequality with limited flexibility. *CoRR*, abs/2210.05634, 2022.

[PT22] Bo Peng and Zhihao Gavin Tang. Order selection prophet inequality: From threshold optimization to arrival time design. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 171–178. IEEE, 2022.

[RS17] Aviad Rubinstein and Sahil Singla. Combinatorial prophet inequalities. In *SODA*, pages 1671–1687. SIAM, 2017.

[RWW20] Aviad Rubinstein, Jack Z. Wang, and S. Matthew Weinberg. Optimal single-choice prophet inequalities from samples. In *ITCS*, volume 151 of *LIPIcs*, pages 60:1–60:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[SC84] Ester Samuel-Cahn. Comparison of threshold stop rules and maximum for independent nonnegative random variables. *the Annals of Probability*, pages 1213–1216, 1984.

A Justification of No Mass Point Assumption

We provide a justification for the no-mass-point assumption by showing that when the distribution contains mass points, every single-threshold algorithm that makes an arbitrary number of queries to the oracle performs arbitrarily badly. Consider an algorithm that makes queries at quantiles $0 < q_1 < \cdots < q_k < 1$ and sets a threshold θ for accepting variables. Note that θ does not have to be equal to any of $v(q_i)$'s. Now suppose that for all queries $\{q_i\}_{i\in[k]}$ the returned values are 1, i.e., $v(q_i) = 1$ for all $i \in [k]$. We consider how the algorithm sets the threshold.

If $\theta > 1$ then for the distribution D such that $x = 1$ with probability 1, we have $\mathsf{ALG} = 0$ because no variable will be accepted, and $\mathsf{OPT} = 1$. Thus the performance is arbitrarily bad. On the other hand, if $\theta \leq 1$ then we consider the following distribution. Let $M \gg \frac{n}{1-q_k^n}$ be an arbitrarily large number. Let $x = 1$ with probability q_k and $x = M$ otherwise. Since $\theta \leq 1$, we have $\mathsf{ALG} = 1$ while $\mathsf{OPT} = 1 \cdot q_k^n + M \cdot (1 - q_k^n) \gg n$. Again, the performance of the algorithm is arbitrarily bad.

Note that it is possible that the algorithm makes an additional query at $q_{k+1} = 1$. However, this does not change the result because we can modify the first distribution slightly such that $x = 1$ with probability $1 - \frac{1}{M^3}$ and $x = M$ otherwise. Note that for such a distribution we have $\mathsf{ALG} \approx 0$ and $\mathsf{OPT} \approx 1$. Then for both distributions the values returned by the oracle are exactly the same: $v(q_i) = 1$ for all $i \in [k]$ and $v(q_{k+1}) = M$. However, there is no way to distinguish these two distributions when setting the threshold.

B Related Works

Prophet Inequality with Complete Information. The theory of prophet inequality is initiated by [GM66] in the late sixties, and has been widely studied in seventies and eighties; see, e.g., [KS77, KS78, SC84, Ker86, HK82]. The problem regained significant interest in the last decade partly because of its application in posted price mechanisms that are widely adopted in (online) auctions [HKS07, CHMS10, Luc17]. For general (non-identical) distributions, the optimal competitive ratio is 0.5 when the ordering of the variables is adversarial [KS77, KS78, SC84]. In contrast, Chawla et al. [CHMS10] showed that when the algorithm can decide the arrival order of the variables, the competitive ratio can be improved to $1-1/e$. The ratio was further improved to 0.634 and 0.669 by [ACK18] and [CSZ19], respectively. These results also apply to the setting with random arrival orders, which is called the *prophet secretary problem* by [EHLM15]. Very recently, Peng and Tang [PT22] proposed a 0.7251-competitive algorithm that selects the arrival order of the variables based on a reduction to a continuous arrival time design problem. The ratio is further improved to 0.7258 by [BC22], who also provided a 0.7254 upper bound for the competitive ratio of algorithms for the prophet secretary problem, separating the best possible ratios for the two problems. Beyond the classic single-choice setting, recent works studied the setting where multiple variables can be selected subject to some combinatorial constraints [CHMS10, KW12, DK15, AKW19, DFKL20]. In the aforementioned works, the objective is additive over the selected variables and similar problems with non-additive objectives were investigated by [RS17] and [DKL20].

Prophet Inequality with Sample Queries. With unknown i.i.d. distributions, Correa et al. [CDFS19] showed that the best an algorithm can do is $1/e$-competitive, and the competitive ratio cannot be improved even if it can observe $o(n)$ random samples. However, if the algorithm has n samples, it can achieve a competitive ratio $\alpha \in [0.6321, 0.6931]$. If $O(n^2)$ samples are given, the competitive ratio can be further improved to $(0.7451 - \epsilon)$, the best possible ratio even if the full distribution is known to the algorithm. Later, Rubinstein et al. [RWW20] improved this result

by showing that $O(n)$ samples suffice to achieve the optimal ratio. With non-i.i.d. distributions, Rubinstein et al. [RWW20] proved that one random sample from each distribution is enough to define a 0.5-competitive algorithm. Nuti et al. [NV22] studied the prophet secretary version of the same problem and showed that the probability of selecting the maximum value is 0.25, which is optimal. For multi-choice prophet inequality problems with random samples, constant competitive algorithms have been analyzed [AKW19, CDF+21]. Beyond random samples, inaccurate prior distributions were considered by [DK19], where the algorithm knows some estimations of the true distributions under various metrics. A simultaneous and independent work by [PST22] also considered the multi-threshold algorithms and proved that the competitive ratio improves with more thresholds and approaches 0.7451.

C Improving the Ratio with Three or More Thresholds

We show in this section that the two-threshold algorithm and its analysis can be naturally extended to multiple thresholds, and better competitive ratios can be achieved. In general, the algorithm has k thresholds, and divides the time horizon $\{1, \ldots, n\}$ into k phases, where $k \geq 2$.

Algorithm. The algorithm has positive parameters $\{c_\ell, \rho_\ell\}_{1 \leq \ell \leq k}$, where $c_1 < \cdots < c_k$ and $\sum_{\ell=1}^{k} \rho_\ell = 1$. Each parameter c_ℓ corresponds to a threshold $\theta_\ell := v(1 - c_\ell/n)$ via a query to the oracle. The parameters $\rho_1, \ldots, \rho_k$ divide the time horizon into k phases: phase ℓ includes variables x_i satisfying $n \cdot \sum_{j<\ell} \rho_j < i \leq n \cdot \sum_{j \leq \ell} \rho_j$, and threshold θ_ℓ is used to decide the acceptance of these variables. For sufficiently large n, we can assume w.l.o.g. that $\rho_\ell \cdot n$ is an integer for all ℓ. Note that each phase ℓ contains exactly $\rho_\ell \cdot n$ variables. We describe the algorithm formally in Algorithm 3.

Algorithm 3: k-Threshold Algorithm

Input: Number of items n, quantile choices $\{c_\ell\}_{1 \leq \ell \leq k}$ with $c_1 < c_2 < \cdots < c_k$, and observation portions $\{\rho_\ell\}_{1 \leq \ell \leq k}$ with $\sum_{\ell=1}^{k} \rho_\ell = 1$.
Output: A variable selected by the gambler.
for $\ell = 1, 2, \ldots, k$ **do**
  Set $\theta_\ell := v(1 - c_\ell/n)$ by making a query to the oracle; // phase ℓ
  for $i = n \cdot \sum_{j<\ell} \rho_j + 1, \ldots, n \cdot \sum_{j \leq \ell} \rho_j$ **do**
    if $x_i \geq \theta_\ell$ **then**
      return x_i and **terminate**;

We call Algorithm 3 the *k-threshold algorithm* and refer to the ℓ-th for-loop as *phase ℓ* of the algorithm, during which variables in phase ℓ are evaluated. Note that θ_ℓ's forming a decreasing sequence of thresholds since c_ℓ's are increasing in ℓ.

In the following sections, we extend our analysis in Section 3 and provide a general framework that gives the competitive ratio of the k-threshold algorithm. We first obtain a lower bound in terms of $\{\theta_\ell, \Delta(c_\ell)\}_{1 \leq \ell \leq k}$ for ALG in Section C.1 in a similar way as in Section 3.3. Then we provide several upper bounds for OPT in Section C.2 by extending our analysis in Section 3.4. We ensure that all of these bounds are linear in terms of $\{\theta_\ell, \Delta(c_\ell)\}_{1 \leq \ell \leq k}$, which enables us to construct an LP in Section C.3 whose optimal objective value lower bounds the competitive ratio.

C.1 Lower Bounding ALG

We derive a lower bound for the expected gain of the k-threshold algorithm in the following lemma.

Lemma C.1 *Let $\{c_\ell, \rho_\ell\}_{1\le\ell\le k}$ be the positive parameters of a k-threshold algorithm and $\epsilon > 0$ be arbitrarily small. For sufficiently large n, we have*

$$\mathsf{ALG} \ge (1-\epsilon)\cdot \sum_{\ell=1}^{k} e^{-\sum_{z=1}^{\ell-1}\rho_z\cdot c_z}\cdot (1-e^{-\rho_\ell\cdot c_\ell})\cdot\left(\theta_\ell + \frac{\Delta(c_\ell)}{c_\ell}\right). \tag{C.1}$$

Proof: Let $\pi \in [k+1]$ denote the *stopping phase* of the above algorithm. For each $\ell \in [k]$, we let τ_ℓ be the stopping time *within* phase π: when $\pi = \ell$, we have $\tau_\ell \in [\rho_\ell \cdot n]$; otherwise τ_ℓ is undefined. We write the expected gain of the algorithm as

$$\mathsf{ALG} = \sum_{\ell=1}^{k}\sum_{i=1}^{\rho_\ell\cdot n} \Pr[\pi \ge \ell]\cdot \Pr[\tau_\ell = i \wedge \pi = \ell \mid \pi \ge \ell]\cdot \mathbb{E}[x_i \mid \tau_\ell = i \wedge \pi = \ell].$$

The algorithm accepts the i-th variable in phase ℓ if and only if (i) all variables from phase $z < \ell$ are less than their corresponding thresholds θ_z's; (ii) the first $(i-1)$ variables in phase ℓ are below θ_ℓ; and (iii) x_i is at least θ_ℓ. Therefore, we have

$$\Pr[\pi \ge \ell]\cdot \Pr[\tau_\ell = i \wedge \pi = \ell \mid \pi \ge \ell] = \prod_{z=1}^{\ell-1} F(\theta_z)^{\rho_z\cdot n}\cdot F(\theta_\ell)^{i-1}\cdot G(\theta_\ell).$$

Similar to Equation (3.5), the expected gain of the algorithm given that it accepts x_i is

$$\mathbb{E}[x_i \mid \tau_\ell = i \wedge \pi = \ell] = \theta_\ell + \frac{\Delta(c_\ell)/n}{G(\theta_\ell)}.$$

In summary, the expected gain of the algorithm is

$$\begin{aligned}\mathsf{ALG} &= \sum_{\ell=1}^{k}\sum_{i=1}^{\rho_\ell\cdot n}\prod_{z=1}^{\ell-1} F(\theta_z)^{\rho_z\cdot n}\cdot F(\theta_\ell)^{i-1}\cdot G(\theta_\ell)\cdot\left(\theta_\ell + \frac{\Delta(c_\ell)/n}{G(\theta_\ell)}\right)\\ &= \sum_{\ell=1}^{k}\prod_{z=1}^{\ell-1} F(\theta_z)^{\rho_z\cdot n}\sum_{i=1}^{\rho_\ell\cdot n} F(\theta_\ell)^{i-1}\cdot\left(G(\theta_\ell)\cdot\theta_\ell + \frac{\Delta(c_\ell)}{n}\right).\end{aligned}$$

Using $F(\theta_z) = 1 - \frac{c_z}{n}$ and for sufficiently large n we have

$$\begin{aligned}\mathsf{ALG} &= \sum_{\ell=1}^{k}\prod_{z=1}^{\ell-1}\left(1-\frac{c_z}{n}\right)^{\rho_z\cdot n}\sum_{i=1}^{\rho_\ell\cdot n}\left(1-\frac{c_\ell}{n}\right)^{i-1}\cdot\left(\frac{c_\ell}{n}\cdot\theta_\ell + \frac{\Delta(c_\ell)}{n}\right)\\ &\ge \sum_{\ell=1}^{k}\prod_{z=1}^{\ell-1}(1-\epsilon)\cdot e^{-\rho_z\cdot c_z}\cdot\frac{1-(1-\frac{c_\ell}{n})^{\rho_\ell\cdot n}}{c_\ell/n}\cdot\left(\frac{c_\ell}{n}\cdot\theta_\ell + \frac{\Delta(c_\ell)}{n}\right)\\ &\ge (1-\epsilon)\cdot\sum_{\ell=1}^{k} e^{-\sum_{z=1}^{\ell-1}\rho_z\cdot c_z}\cdot(1-e^{-\rho_\ell\cdot c_\ell})\cdot\left(\theta_\ell + \frac{\Delta(c_\ell)}{c_\ell}\right),\end{aligned} \tag{C.2}$$

where $\epsilon > 0$ is an arbitrarily small constant. ■

C.2 Upper Bounding OPT

Given threshold θ_ℓ for $1 \le \ell \le k$, we can reuse the argument in Equation (3.2) and obtain the following set of k upper bounds for OPT:

$$\mathsf{OPT} \le \theta_\ell + n \cdot \mathbb{E}[(x - \theta_\ell)^+] = \theta_\ell + \Delta(c_\ell), \quad \forall 1 \le \ell \le k.$$

However, as discussed in Section 3.4, the above bounds alone would not be enough to beat $1 - 1/e$. In what follows, we give a more careful analysis that leads to tighter upper bounds for OPT. Recall that we have $0 \le \theta_k < \theta_{k-1} < \cdots < \theta_1$. We introduce new variables δ_ℓ's for $1 \le \ell < k$,

$$\delta_\ell = \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x^* \ge t] dt, \quad \forall 1 \le \ell < k.$$

Recall that $x^* = \max_{i \in [n]}\{x_i\}$. We have the following upper bound for OPT:

$$\begin{aligned}
\mathsf{OPT} = \mathbb{E}[x^*] &= \int_0^\infty \Pr[x^* \ge t] dt \\
&= \int_0^{\theta_k} \Pr[x^* \ge t] dt + \sum_{\ell=1}^{k-1} \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x^* \ge t] dt + \int_{\theta_1}^\infty \Pr[x^* \ge t] dt \\
&\le \theta_k + \sum_{\ell=1}^{k-1} \delta_\ell + \mathbb{E}[(x^* - \theta_1)^+] \le \theta_k + \sum_{\ell=1}^{k-1} \delta_\ell + \Delta(c_1). \qquad \text{(C.3)}
\end{aligned}$$

In the next three lemmas, we establish three upper bounds for δ_ℓ's, which, when combined with Equation (C.3), provide much better upper bounds on OPT.

Lemma C.2 *For arbitrarily small $\epsilon > 0$ and sufficiently large n, we have*

$$\delta_\ell \le (1 + \epsilon) \cdot (\theta_\ell - \theta_{\ell+1}) \cdot (1 - e^{-c_{\ell+1}}), \quad \forall 1 \le \ell < k. \qquad \text{(C.4)}$$

Proof: For $1 \le \ell < k$, we have

$$\begin{aligned}
\delta_\ell &= \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x^* \ge t] dt \le \Pr[x^* \ge \theta_{\ell+1}] \cdot (\theta_\ell - \theta_{\ell+1}) \\
&= \Big(1 - (1 - \frac{c_{\ell+1}}{n})^n\Big) \cdot (\theta_\ell - \theta_{\ell+1}) \le (1 + \epsilon) \cdot (1 - e^{-c_{\ell+1}}) \cdot (\theta_\ell - \theta_{\ell+1}),
\end{aligned}$$

where the second equality holds due to $\Pr_{x \sim D}[x < \theta_{\ell+1}] = 1 - c_{\ell+1}/n$. ■

The second upper bound for δ_ℓ's can be achieved using the union bound.

Lemma C.3 *We have*

$$\delta_\ell \le \Delta(c_{\ell+1}) - \Delta(c_\ell), \quad \forall 1 \le \ell < k. \qquad \text{(C.5)}$$

Proof: For $1 \le \ell < k$, we have

$$\begin{aligned}
\delta_\ell &= \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x^* \ge t] dt = \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[(x_1 \ge t) \vee (x_2 \ge t) \vee \ldots \vee (x_n \ge t)] dt \\
&\le n \cdot \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x \ge t] dt = n \cdot \int_{\theta_{\ell+1}}^\infty \Pr[x \ge t] dt - n \cdot \int_{\theta_\ell}^\infty \Pr[x \ge t] dt \\
&= \Delta(c_{\ell+1}) - \Delta(c_\ell),
\end{aligned}$$

where the last equality uses $\int_{\theta_\ell}^\infty \Pr[x \ge t] dt = \mathbb{E}[(x - \theta_\ell)^+]$ and $\theta_\ell = v(1 - c_\ell/n)$. ■

Finally, we derive the third upper bound on δ_l's.

Lemma C.4 *For all $1 \le \ell < k$ and constant $c \in [c_\ell, c_{\ell+1}]$, let $\gamma := \frac{c-c_\ell}{e^{-c}-e^{-c_{\ell+1}}}$. For arbitrarily small $\epsilon > 0$ and sufficiently large n, we have*

$$\gamma \cdot \delta_\ell/(1+\epsilon) \le \Delta(c_{\ell+1}) - \Delta(c_\ell) - (c_\ell - \gamma(1-e^{-c})) \cdot (\theta_\ell - \theta_{\ell+1}). \tag{C.6}$$

Proof: Recall $\Pr[x \ge \theta_\ell] = c_\ell/n$, $\Pr[x \ge \theta_{\ell+1}] = c_{\ell+1}/n$ and that $\Pr[x \ge t]$ is strictly decreasing in t. For all $c \in [c_\ell, c_{\ell+1}]$, there must exist $t^* \in [\theta_{\ell+1}, \theta_\ell]$ such that $\Pr[x \ge t^*] = c/n$. Thus we have,

$$\Pr[x \ge t] \in \begin{cases} [\frac{c}{n}, \frac{c_{\ell+1}}{n}], & \forall t \in [\theta_{\ell+1}, t^*]; \\ [\frac{c_\ell}{n}, \frac{c}{n}), & \forall t \in (t^*, \theta_\ell]. \end{cases}$$

Since $\Pr[x^* \ge t] = 1 - (1 - \Pr[x \ge t])^n$, for sufficiently large n, we have:

$$\begin{aligned} \forall t \in [\theta_{\ell+1}, t^*], \quad & \Pr[x^* \ge t] \le 1 - \left(1 - \frac{c_{\ell+1}}{n}\right)^n \le (1+\epsilon) \cdot (1 - e^{-c_{\ell+1}}); \\ \forall t \in (t^*, \theta_\ell], \quad & \Pr[x^* \ge t] < 1 - \left(1 - \frac{c}{n}\right)^n \le (1+\epsilon) \cdot (1 - e^{-c}). \end{aligned}$$

For convenience, we define $\alpha := \frac{t^*-\theta_{\ell+1}}{\theta_\ell - \theta_{\ell+1}} \in [0,1]$. We have

$$\begin{aligned} \delta_\ell = \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x^* \ge t]dt &= \int_{\theta_{\ell+1}}^{t^*} \Pr[x^* \ge t]dt + \int_{t^*}^{\theta_\ell} \Pr[x^* \ge t]dt \\ &\le (1+\epsilon) \cdot (t^* - \theta_{\ell+1}) \cdot (1 - e^{-c_{\ell+1}}) + (1+\epsilon) \cdot (\theta_\ell - t^*) \cdot (1 - e^{-c}) \\ &= (1+\epsilon) \cdot (\theta_\ell - \theta_{\ell+1}) \cdot (1 - e^{-c} + \alpha \cdot (e^{-c} - e^{-c_{\ell+1}})). \end{aligned}$$

Rearranging the inequality gives

$$\alpha \ge \frac{1}{e^{-c} - e^{-c_{\ell+1}}} \cdot \left(\frac{\delta_\ell}{(1+\epsilon)(\theta_\ell - \theta_{\ell+1})} - (1 - e^{-c}) \right).$$

On the other hand, we have (recall that we define $\gamma = \frac{c-c_\ell}{e^{-c}-e^{-c_{\ell+1}}}$)

$$\begin{aligned} n \cdot \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x \ge t]dt &= n \cdot \int_{\theta_{\ell+1}}^{t^*} \Pr[x \ge t]dt + n \cdot \int_{t^*}^{\theta_\ell} \Pr[x \ge t]dt \\ &\ge (t^* - \theta_{\ell+1}) \cdot c + (\theta_\ell - t^*) \cdot c_\ell \\ &= (\theta_\ell - \theta_{\ell+1}) \cdot (c_\ell + \alpha \cdot (c - c_\ell)) \\ &\ge (\theta_\ell - \theta_{\ell+1}) \cdot \left\{ c_\ell + \frac{c - c_\ell}{e^{-c} - e^{-c_{\ell+1}}} \cdot \left(\frac{\delta_\ell}{(1+\epsilon)(\theta_1 - \theta_{\ell+1})} - (1 - e^{-c}) \right) \right\} \\ &= c_\ell(\theta_\ell - \theta_{\ell+1}) + \frac{\gamma}{1+\epsilon} \cdot \delta_\ell - \gamma(1 - e^{-c}) \cdot (\theta_\ell - \theta_{\ell+1}). \end{aligned}$$

Recall that $n \cdot \int_{\theta_{\ell+1}}^{\theta_\ell} \Pr[x \ge t]dt = \Delta(c_{\ell+1}) - \Delta(c_\ell)$. The above lower bound implies

$$\Delta(c_{\ell+1}) - \Delta(c_\ell) \ge (c_\ell - \gamma(1 - e^{-c})) \cdot (\theta_\ell - \theta_{\ell+1}) + \frac{\gamma}{1+\epsilon} \cdot \delta_\ell.$$

Rearranging the inequality completes the proof. ■

C.3 Competitive Ratios for k-Threshold Algorithms

To recap, we have established a lower bound for ALG in Section C.1 as well as several upper bounds for OPT in Section C.2. Note that given parameters $\{c_\ell, \rho_\ell\}_{1\le\ell\le k}$, these bounds are linear in $\{\theta_\ell, \Delta(c_\ell)\}_{1\le\ell\le k}$ and $\{\delta_\ell\}_{1\le\ell<k}$. We can now construct a minimization LP taking Equation (C.1) as the objective and Equations (C.3), (C.4), (C.5), (C.6) as constraints with non-negative LP variables $\{\theta_\ell, \Delta(c_\ell)\}_{1\le\ell\le k}$ and $\{\delta_\ell\}_{1\le\ell<k}$. In the following theorem, we prove that the optimal objective of the LP gives a lower bound on the competitive ratio for the algorithm.

Theorem C.1 *Given parameters $\{c_\ell, \rho_\ell\}_{1\le\ell\le k}$ of a k-threshold algorithm, the optimal value of the following LP provides a lower bound for the competitive ratio of the algorithm when $n \to \infty$.*

$$\begin{aligned}
&\textit{minimize} && \textstyle\sum_{\ell=1}^{k} e^{-\sum_{z=1}^{\ell-1}\rho_z\cdot c_z}\cdot(1-e^{-\rho_\ell\cdot c_\ell})\cdot\left(\theta_\ell+\frac{\Delta(c_\ell)}{c_\ell}\right)\\
&\textit{subject to} && 0\le\theta_k\le\cdots\le\theta_1,\\
& && \Delta(c_1),\delta_1,\ldots,\delta_k\ge 0,\\
& && 1\le\theta_k+\textstyle\sum_{\ell=1}^{k-1}\delta_\ell+\Delta(c_1),\\
& && \delta_\ell\le(\theta_\ell-\theta_{\ell+1})\cdot(1-e^{-c_{\ell+1}}),\qquad \forall\ell\in[k-1],\\
& && \delta_\ell\le\Delta(c_{\ell+1})-\Delta(c_\ell),\qquad \forall\ell\in[k-1],\\
& && \gamma\cdot\delta_\ell\le\Delta(c_{\ell+1})-\Delta(c_\ell)-(c_\ell-\gamma(1-e^{-c}))\cdot(\theta_\ell-\theta_{\ell+1}),\\
& && \qquad\qquad \forall\ell\in[k-1],\ \forall c\in[c_\ell,c_{\ell+1}],\ \gamma=\tfrac{c-c_\ell}{e^{-c}-e^{-c_{\ell+1}}}.
\end{aligned}$$

Proof: Let $\{c_\ell, \rho_\ell\}_{1\le\ell\le k}$ be the parameters of a k-threshold algorithm and $\epsilon > 0$ be arbitrarily small. The objective of the LP comes from Equation (C.1), the lower bound for ALG, where we omit the $(1-\epsilon)$ term since we have $\epsilon \to 0$ when $n \to \infty$. We will also omit the $(1+\epsilon)$ terms in the following for the same reason. The first two constraints follow straightforwardly from the definitions of the parameters. By scaling we can assume w.l.o.g. that OPT $= 1$. Therefore Equation (C.3) gives the third constraint. The three proceeding sets of constraints follow from Equations (C.4), (C.5) and (C.6), the three upper bounds on δ_ℓ's that we have established in Section C.2.

Observe that every distribution D induces a set of LP variables $\{\theta_\ell, \Delta(c_\ell)\}_{1\le\ell\le k}$ as well as $\{\delta_\ell\}_{1\le\ell<k}$. Moreover, by our analyses, they form a feasible solution to the LP. Since the objective of any feasible solution induced by distribution D provides a lower bound on ALG/OPT, the optimal (minimum) objective of the LP provides a lower bound on the competitive ratio, i.e., the worst-case performance against all distributions. ■

With the above result, it remains to set the appropriate parameters of the algorithm such that the optimal value of the LP is as large as possible. We demonstrate the effectiveness of our LP framework with the following theorem that gives lower bounds of the competitive ratio for the k-threshold algorithm with $k \in \{2, 3, 4, 5\}$.

Theorem C.2 *The k-threshold algorithm achieves a competitive ratio of at least* 0.6786 *when $k = 2$;* 0.6883 *when $k = 3$;* 0.6946 *when $k = 4$; and* 0.7004 *when $k = 5$.*

Proof: For two thresholds, we have three parameters for the algorithm, c_1, c_2, and $\rho := \rho_1$ (note that $\rho_2 = 1 - \rho_1 = 1 - \rho$). The LP has four variables: θ_1, θ_2, $\Delta_1 := \Delta(c_1)$ and $\Delta_2 := \Delta(c_2)$, and is

written as follows.

$$\begin{aligned}
\text{minimize} \quad & (1-e^{-\rho c_1})\theta_1 + \tfrac{1-e^{-\rho c_1}}{c_1}\Delta_1 + e^{-\rho c_1}(1-e^{-(1-\rho)c_2})\theta_2 + \tfrac{e^{-\rho c_1}(1-e^{-(1-\rho)c_2})}{c_2}\Delta_2, \\
\text{subject to} \quad & \theta_1 \geq \theta_2 \geq 0,\ \Delta_1 \geq 0,\ \delta_1 \geq 0, \\
& 1 \leq \theta_2 + \delta_1 + \Delta_1, \\
& \delta_1 \leq (\theta_1 - \theta_2)\cdot(1-e^{-c_2}), \\
& \delta_1 \leq \Delta_2 - \Delta_1, \\
& \gamma \cdot \delta_1 \leq \Delta_2 - \Delta_1 - (c_1 - \gamma(1-e^{-c}))\cdot(\theta_1-\theta_2),\ \forall c \in [c_1, c_2],\ \gamma = \tfrac{c-c_1}{e^{-c}-e^{-c_2}}.
\end{aligned}$$

By Theorem C.1, given values of c_1, c_2, ρ, the objective value of the above LP bounds the competitive ratio from below. Using an online LP solver, it can be verified that by fixing $c_1 = 0.7067$, $c_2 = 1.8353$ and $\rho = 0.6204$, the optimal value of the LP is at least 0.6786.

	2 thresholds	3 thresholds	4 thresholds	5 thresholds
c_1	0.7067	0.7204	0.6857	0.6561
c_2	1.8353	1.7551	1.4367	1.4082
c_3	–	3.2857	2.4417	2.2735
c_4	–	–	3.9036	3.4423
c_5	–	–	–	4.4783
ρ_1	0.6204	0.71	0.65	0.64
ρ_2	0.3796	0.195	0.19	0.17
ρ_3	–	0.095	0.1	0.11
ρ_4	–	–	0.06	0.06
ρ_5	–	–	–	0.02
ratio	**0.6786**	**0.6883**	**0.6946**	**0.7004**

Table 1: Parameters and the resulting competitive ratios of the k-threshold algorithms for various k.

The proofs for $k \geq 3$ are identical but with different choices of parameters. We summarize the choices of parameters we have found and the corresponding competitive ratios in Table 1. ■

We know from [EHKS18] and [CFH+17] that the optimal competitive ratios are $1 - 1/e$ and 0.7451 with 1 and n thresholds, respectively. Our results partially fill in the gap between these two results. While we can continue the numerical computation to achieve better ratios for $k \geq 6$, we believe that a more meaningful open problem is to improve the current analyses as we discuss in detail in Section 5, and see what value of k is sufficient to achieve a competitive ratio close to the upper bound of 0.7451.

D Missing Proofs from Section 3

D.1 Proof of Lemma 3.2

Proof: By definition of δ, we have

$$\begin{aligned}
\delta &= \int_{\theta_2}^{\theta_1} \Pr[x^* \geq t]dt \leq \Pr[x^* \geq \theta_2]\cdot(\theta_1-\theta_2) \\
&= \left(1-(1-\frac{c_2}{n})^n\right)\cdot(\theta_1-\theta_2) \leq (1+\epsilon)\cdot(1-e^{-c_2})\cdot(\theta_1-\theta_2),
\end{aligned}$$

where the last inequality holds for all $n = \omega(\frac{1}{\epsilon})$. ■

D.2 Proof of Lemma 3.3

Proof: By definition of δ, we have

$$\delta = \int_{\theta_2}^{\theta_1} \Pr[x^* \geq t]dt = \int_{\theta_2}^{\theta_1} \Pr[(x_1 \geq t) \vee (x_2 \geq t) \vee \ldots \vee (x_n \geq t)]dt$$
$$\leq n \cdot \int_{\theta_2}^{\theta_1} \Pr_{x\sim D}[x \geq t]dt = n \cdot \int_{\theta_2}^{\infty} \Pr_{x\sim D}[x \geq t]dt - n \cdot \int_{\theta_1}^{\infty} \Pr_{x\sim D}[x \geq t]dt = \Delta(c_2) - \Delta(c_1),$$

where the inequality holds by using the union bound. ■

D.3 Proof of Lemma 3.4

Proof: Recall that $\Pr[x \geq \theta_1] = c_1/n$, $\Pr[x \geq \theta_2] = c_2/n$ and that $\Pr[x \geq t]$ is strictly decreasing in t. For all $c \in [c_1, c_2]$, there must exist $t^* \in [\theta_2, \theta_1]$ such that $\Pr[x \geq t^*] = c/n$. Thus we have,

$$\Pr[x \geq t] \in \begin{cases} [\frac{c}{n}, \frac{c_2}{n}], & \forall t \in [\theta_2, t^*]; \\ [\frac{c_1}{n}, \frac{c}{n}), & \forall t \in (t^*, \theta_1]. \end{cases}$$

Since $\Pr[x^* \geq t] = 1 - (1 - \Pr[x \geq t])^n$, for sufficiently large n, we have:

$$\forall t \in [\theta_2, t^*], \quad \Pr[x^* \geq t] \leq 1 - \left(1 - \frac{c_2}{n}\right)^n \leq (1+\epsilon) \cdot (1 - e^{-c_2});$$
$$\forall t \in (t^*, \theta_1], \quad \Pr[x^* \geq t] < 1 - \left(1 - \frac{c}{n}\right)^n \leq (1+\epsilon) \cdot (1 - e^{-c}).$$

For convenience, we define $\alpha := \frac{t^* - \theta_2}{\theta_1 - \theta_2} \in [0, 1]$. We have

$$\delta = \int_{\theta_2}^{\theta_1} \Pr[x^* \geq t]dt = \int_{\theta_2}^{t^*} \Pr[x^* \geq t]dt + \int_{t^*}^{\theta_1} \Pr[x^* \geq t]dt$$
$$\leq (1+\epsilon) \cdot (t^* - \theta_2) \cdot (1 - e^{-c_2}) + (1+\epsilon) \cdot (\theta_1 - t^*) \cdot (1 - e^{-c})$$
$$= (1+\epsilon) \cdot (\theta_1 - \theta_2) \cdot (1 - e^{-c} + \alpha \cdot (e^{-c} - e^{-c_2})).$$

Rearranging the inequality gives

$$\alpha \geq \frac{1}{e^{-c} - e^{-c_2}} \cdot \left(\frac{\delta}{(1+\epsilon)(\theta_1 - \theta_2)} - (1 - e^{-c}) \right).$$

On the other hand, with the definition of $\gamma = \frac{c - c_1}{e^{-c} - e^{-c_2}}$, we have

$$n \cdot \int_{\theta_2}^{\theta_1} \Pr[x \geq t]dt = n \cdot \int_{\theta_2}^{t^*} \Pr[x \geq t]dt + n \cdot \int_{t^*}^{\theta_1} \Pr[x \geq t]dt$$
$$\geq (t^* - \theta_2) \cdot c + (\theta_1 - t^*) \cdot c_1$$
$$= (\theta_1 - \theta_2) \cdot (c_1 + \alpha \cdot (c - c_1))$$
$$\geq (\theta_1 - \theta_2) \cdot \left\{ c_1 + \frac{c - c_1}{e^{-c} - e^{-c_2}} \cdot \left(\frac{\delta}{(1+\epsilon)(\theta_1 - \theta_2)} - (1 - e^{-c}) \right) \right\}$$
$$= c_1(\theta_1 - \theta_2) + \frac{\gamma}{1+\epsilon} \cdot \delta - \gamma(1 - e^{-c}) \cdot (\theta_1 - \theta_2).$$

Recall that $n \cdot \int_{\theta_2}^{\theta_1} \Pr[x \geq t]dt = \Delta(c_2) - \Delta(c_1)$. The above lower bound implies

$$\Delta(c_2) - \Delta(c_1) \geq (c_1 - \gamma(1 - e^{-c})) \cdot (\theta_1 - \theta_2) + \frac{\gamma}{1+\epsilon} \cdot \delta.$$

Rearranging the inequality completes the proof. ■

D.4 Proof of Lemma 3.5

Proof: Given parameters c_1, c_2, and ρ, following the proof of Lemma 3.1, when $n \to \infty$, the expected gain of the algorithm is

$$\mathsf{ALG} = (1 - e^{-\rho c_1})\theta_1 + \frac{1 - e^{-\rho c_1}}{c_1}\Delta_1 + e^{-\rho c_1}(1 - e^{-(1-\rho)c_2})\theta_2 + \frac{e^{-\rho c_1}(1 - e^{-(1-\rho)c_2})}{c_2}\Delta_2. \tag{D.1}$$

Note that this is also the objective function of the factor revealing LP. Let $\epsilon > 0$ be an arbitrarily small constant and consider the following three distributions.

1. Let M be a sufficiently large number such that $c_1, c_2 \gg 1/M$. Consider the distribution D_1 as follows. With probability $\frac{1}{n \cdot M}$, x is uniformly distributed over $[M - \epsilon, M + \epsilon]$; with probability $1 - \frac{1}{n \cdot M}$, x is uniformly distributed over $[0, \epsilon]$. The CDF for D_1 is strictly increasing, and when $\epsilon \to 0$, $M \to \infty$, we have

$$\mathsf{OPT} \approx M \cdot \left(1 - (1 - \frac{1}{n \cdot M})^n\right) \approx M \cdot \frac{1}{M} = 1.$$

It is not hard to see that both θ_1 and θ_2 are at most ϵ. Moreover, $\Delta_1 = n \cdot \mathbb{E}[(x - \theta_1)^+] \approx n \cdot \mathbb{E}[x] = 1$ and similarly $\Delta_2 \approx 1$. By plugging these values in Equation (D.1), under $n \to \infty$,

$$\mathsf{ALG} \approx \frac{1 - e^{-\rho c_1}}{c_1} + \frac{e^{-\rho c_1}(1 - e^{-(1-\rho)c_2})}{c_2} := UB_1. \tag{D.2}$$

2. Consider the distribution D_2 such that x is uniformly distributed over $[1, 1 + \epsilon]$. Note that $\mathsf{OPT} \approx 1$ as $\epsilon \to 0$ and the CDF is strictly increasing. For this distribution, we have $\theta_1 \approx \theta_2 \approx 1$ while $\Delta_1 \approx \Delta_2 \approx 0$. Therefore, by Equation (D.1) and when $n \to \infty$,

$$\mathsf{ALG} \approx (1 - e^{-\rho c_1}) + e^{-\rho c_1}(1 - e^{-(1-\rho)c_2}) = 1 - e^{-\rho c_1 - (1-\rho)c_2} := UB_2. \tag{D.3}$$

3. Let $c \in (c_1, c_2)$. Consider the distribution D_3 as follows. With probability $\frac{c}{n}$, x_i is uniformly distributed over $[\frac{1}{1-e^{-c}} - \epsilon, \frac{1}{1-e^{-c}} + \epsilon]$; with probability $1 - \frac{c}{n}$, x_i is uniformly distributed over $[0, \epsilon]$. Note that the CDF for D_3 is strictly increasing. Meanwhile, when $n \to \infty$,

$$\mathsf{OPT} \approx \left(1 - (1 - \frac{c}{n})^n\right) \cdot \frac{1}{1 - e^{-c}} \approx (1 - e^{-c}) \cdot \frac{1}{1 - e^{-c}} = 1.$$

Since $\frac{c_1}{n} < \frac{c}{n} < \frac{c_2}{n}$, we have $\theta_1 \approx \frac{1}{1-e^{-c}}$ and $\theta_2 \approx 0$. Moreover, $\Delta_1 \approx 0$ and $\Delta_2 \approx n \cdot \mathbb{E}[x] \approx \frac{c}{1-e^{-c}}$. Again, using Equation (D.1), when $n \to \infty$ we have

$$\mathsf{ALG} \approx \frac{1 - e^{-\rho c_1}}{1 - e^{-c}} + \frac{e^{-\rho c_1}(1 - e^{-(1-\rho)c_2})}{c_2} \cdot \frac{c}{1 - e^{-c}} := UB_3(c).$$

We define our third upper bound as the minimum over all possible $UB_3(c)$'s, i.e.,

$$UB_3 := \min_{c \in (c_1, c_2)} \{UB_3(c)\}. \tag{D.4}$$

It remains to show that when $\epsilon \to 0$, $n \to \infty$, and $M \to \infty$, given any values of c_1, c_2 and $\rho \in [0,1]$, at least one of Equations (D.2), (D.3), and (D.4) is at most 0.7081. That is,

$$\max_{c_1 \in [0,n], c_2 \in [0,n], \rho \in [0,1]} \{\min\{UB_1, UB_2, UB_3\}\} \leq 0.7081. \tag{D.5}$$

We first observe that if $c_1 \geq 2$, UB_1 is at most $1/2$. Similarly, for $c_2 \geq 30$, UB_1 cannot be more than $1 - 1/e$. Therefore, we only need to consider the case where $c_1 \in [0,2]$ and $c_2 \in [0,30]$. We can prove (D.5) using computational tools by discretizing the domain for c_1, c_2, and ρ. Specifically, we replace the domains for c_1, c_2 and ρ with $I_{c_1} = \{\frac{2i}{10^5} : 1 \leq i \leq 10^5\}$, $I_{c_2} = \{\frac{3i}{10^6} : 1 \leq i \leq 10^6\}$ and $I_\rho = \{\frac{i}{10^5} : 1 \leq i \leq 10^5\}$, respectively. By enumerating the possible values of c_1, c_2 and ρ within I_{c_1}, I_{c_2} and I_ρ, we obtain an upper bound of 0.70804, which is achieved by $c_1 = 0.51904$, $c_2 = 2.32059$ and $\rho = 0.60473$. It can be verified that discretizing the domain incurs an additive error of at most 2×10^5. Therefore, the upper bound under the continuous domain is at most $0.70806 < 0.7081$. ■