Threshold Testing and Semi-Online Prophet Inequalities

Martin Hoefer

Institute of Computer Science, Goethe University Frankfurt, Germany

Kevin Schewior

Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark

Abstract

We study *threshold testing*, an elementary probing model with the goal to choose a large value out of n i.i.d. random variables. An algorithm can *test* each variable X_i once for some threshold t_i, and the test returns binary feedback whether $X_i \geq t_i$ or not. Thresholds can be chosen adaptively or non-adaptively by the algorithm. Given the results for the tests of each variable, we then select the variable with highest conditional expectation. We compare the expected value obtained by the testing algorithm with expected maximum of the variables.

Threshold testing is a semi-online variant of the gambler's problem and prophet inequalities. Indeed, the optimal performance of *non-adaptive* algorithms for threshold testing is governed by the standard i.i.d. prophet inequality of approximately $0.745 + o(1)$ as $n \to \infty$. We show how *adaptive* algorithms can significantly improve upon this ratio. Our adaptive testing strategy guarantees a competitive ratio of at least $0.869 - o(1)$. Moreover, we show that there are distributions that admit only a constant ratio $c < 1$, even when $n \to \infty$. Finally, when each box can be tested multiple times (with n tests in total), we design an algorithm that achieves a ratio of $1 - o(1)$.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Online algorithms; Theory of computation → Markov decision processes

Keywords and phrases Prophet Inequalities, Testing, Stochastic Probing

Digital Object Identifier 10.4230/LIPIcs.ESA.2023.62

Related Version *Full Version*: `https://arxiv.org/abs/2307.01776`

Funding *Martin Hoefer*: Supported by DFG Research Unit ADYN (411362735) and grant 514505843.
Kevin Schewior: Supported by the Independent Research Fund Denmark, Natural Sciences, grant DFF-0135-00018B.

Acknowledgements The authors gratefully acknowledge discussions with Daniel Schmand and Luca von der Brelie.

1 Introduction

Consider an application process in which n job candidates are interviewed sequentially one by one for a single position. For each candidate, we assume the qualification for the job can be expressed by an i.i.d. non-negative random variable X_i with known distribution F. The goal is to maximize the expected value of the selected candidate. To which extent is the optimal achievable value harmed by the online arrival of the candidates? This is the classic *gambler's problem*, in which the loss in expected value is expressed by prophet inequalities [22, 26, 9]. More precisely, in this model one usually assumes (i) an interview fully reveals the realization of the respective variable, and (ii) the requirement of timely feedback forces the decision maker to irrevocably accept or reject the candidate upon seeing its realization. For i.i.d. variables, the best-possible prophet inequality states that a candidate σ can be selected such that $\mathbb{E}[X_\sigma] \geq \beta \cdot \mathbb{E}[\max\{X_1, \ldots, X_n\}]$, where $\beta \approx 0.745$ [19, 10].

31st Annual European Symposium on Algorithms (ESA 2023).
Editors: Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman; Article No. 62; pp. 62:1–62:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The gambler's problem has been extremely popular over the last decades, but assumptions (i) and (ii) are often unrealistic. Even after a long interview, an interviewer is usually not fully aware of the entire set of exact qualifications of a candidate. Moreover, in many selection processes a decision does not have to be taken instantaneously. In this paper, we examine the consequences of an arguably more realistic set of conditions. First, instead of (i), we assume that each candidate only partly reveals their realization in the form of a *single bit of information*. As we observe rather directly, this assumption is asymptotically equivalent to allowing to make a single threshold query to each candidate. Second, instead of (ii), we assume that the selection must be made only at the end of the process.

More formally, we consider the *threshold testing* model. We assume that (i′), instead of revelation of X_i, we perform a single *threshold test* with some arbitrary threshold t_i, and the feedback is binary ("positive" in case $X_i \geq t_i$ or "negative" otherwise), and (ii′) a candidate must be chosen only after completing *all* threshold tests. Again denoting the selected candidate by σ, we are interested in bounding the loss in expected value using an inequality of the form $\mathbb{E}[X_\sigma] \geq c \cdot \mathbb{E}[\max\{X_1, \ldots, X_n\}]$ for $c \in (0, 1)$. We call this a *semi-online prophet inequality*. A testing algorithm that satisfies it is called *c-competitive* and has a *competitive ratio* of c.

There are four possible models emerging from the different choices of (i) vs. (i′) and (ii) vs. (ii′). The remaining two models do not require substantial analytical efforts. Indeed, when we only replace (i) by (i′), the consequences are trivial: There is an optimal algorithm for the standard gambler's problem that uses threshold tests. Thus, the existing optimal algorithm and its guarantees continue to apply because the space of algorithms only shrinks when we require threshold tests. Also, only replacing (ii) by (ii′) implies a trivial problem – one can see and choose an option with maximum value at the end, a 1-competitive strategy. In contrast, the main contribution of this paper is to show that, with (i′) and (ii′) simultaneously, a mathematically interesting model arises.

Our results also imply a stark qualitative distinction to the standard model. It is well known that adaptivity, i.e., allowing decisions to depend on past observations, does not help for the standard gambler's problem. In our model, we observe rather easily that non-adaptive testing algorithms are unable to asymptotically improve upon the ratio of $\beta \approx 0.745$. Our main result is a set of adaptive algorithms that improves significantly upon this bound and achieves a ratio of approximately 0.869. To complement this result, we show that there are distributions that imply a non-trivial asymptotical upper bound on the ratio, i.e., there is no $(1 - o(1))$-competitive algorithm. We proceed to discuss our contributions in more detail.

1.1 Techniques and Contribution

Let F be the cumulative distribution function of the variables. For most of the paper we assume (essentially w.l.o.g.) that F is continuous. Our algorithms perform quantile testing, i.e., they use thresholds of the form $F^{-1}(1 - q)$ for $q \in (0, 1)$, oblivious of other properties of the distribution. It is straightforward to achieve a competitive ratio of $1 - 1/\mathsf{e} > 0.632$ by using threshold $t_i = F^{-1}(1 - 1/n)$ for all variables and then choosing any variable that has been tested positively (if any); see, e.g., [19]. The analysis of this strategy is asymptotically tight for each of the following two parametric distributions[1]:

$\boldsymbol{F_A}$: For some small $\varepsilon > 0$, with probability $1/\sqrt{n}$ choose a value uniformly from $[1-\varepsilon, 1+\varepsilon]$, and 0 otherwise. As $\varepsilon \to 0$ and $n \to \infty$, the algorithm gets a positive test and therefore value 1 with probability $1 - 1/\mathsf{e}$ while $\mathbb{E}[\max\{X_1, \ldots, X_n\}] = 1$.

[1] Strictly speaking, F_A and F_B are not continuous. For a rigorous argument, one can resort to an arbitrarily close continuous approximation of the distributions to obtain the same result.

F_B: Choose the value 1 with probability $1/n^2$ and 0 otherwise. The algorithm obtains a value 1 with probability $(1-(1-1/n)^n)/n$ while $\max\{X_1,\ldots,X_n\}=1$ with probability $1-(1-1/n^2)^n$. As $n\to\infty$, the ratio of both probabilities approaches $1-1/\mathrm{e}$.

To improve upon $1-1/\mathrm{e}$, an algorithm needs to test for *both* smaller and larger thresholds than $F^{-1}(1-1/n)$. Thresholds that are all larger than $F^{-1}(1-1/n)$ decrease the ratio for F_A; thresholds that are all smaller than $F^{-1}(1-1/n)$ decrease the ratio for F_B.

The class of algorithms we consider here is parameterized by $\alpha_1,\ldots,\alpha_k\in(0,1)$ with $\alpha_1>\alpha_2>\cdots>\alpha_k$. In the beginning, such an algorithm uses $F^{-1}(1-\alpha_1/n)$ as a threshold until it sees a positive test. Generally, after $i<k$ positive tests, it sets $F^{-1}(1-\alpha_{i+1}/n)$ as a threshold. After k positive tests (i.e., on $F^{-1}(1-\alpha_1/n),\ldots,F^{-1}(1-\alpha_k/n)$), the algorithm can make arbitrary tests. Indeed, we eventually choose $\alpha_1>1$ and $\alpha_2<1$.

In our analysis, we exactly calculate the asymptotic probability that the algorithm sees precisely i positive tests. Note that these probabilities asymptotically determine the probability density function of X_σ, the chosen variable. It is a step function in quantile space: The probability of making precisely i positive tests is spread uniformly over the interval $[1-\alpha_i/n,1]$ for all $i\in\{1,\ldots,k\}$.

We compare this probability density function of X_σ with that of $\max\{X_1,\ldots,X_n\}$ by stochastic dominance, leading to a tight analysis of such algorithms. For fixed values of $\alpha_1,\ldots,\alpha_k$, we can analyze the competitive ratio of the respective strategy by solving a piecewise convex optimization problem, where the $k+1$ pieces correspond to the $k+1$ steps of the step function. We numerically maximize the minimum of this function.

We execute this analysis in detail for $k\in\{2,3\}$. For $k=3$, we obtain a competitive ratio of approximately 0.869 by setting $\alpha_1\approx 2.035$, $\alpha_2\approx 0.506$, and $\alpha_3\approx 0.057$. Our numerical results for $k=4$ indicate only negligible improvement by further increasing k.

We complement this result by a constant upper bound on the competitive ratio, i.e., an impossibility of achieving a competitive ratio of $1-o(1)$. Intuitively, there is a trade-off inherent in every test: Testing for a smaller value yields a fallback option in case only one positive test is found at the end; testing for a larger value allows to differentiate between different variables when multiple of them have been tested positively. There are instances in which, irrespective of how the algorithm solves this trade-off, it loses a constant in the competitive ratio. For the proof we consider a distribution where values 1, 2, or 3 occur with probability $1/n$ each, and 0 otherwise. It is minimal in the sense that a competitive ratio of $1-o(1)$ is achievable for any distribution that uses only three values in the support, or whose parameters do not depend on n.

Finally, we establish that a competitive ratio of $1-o(1)$ *can* be achieved using n tests when a single variable can be tested multiple times (recall that the realization of each variable is only drawn *once initially* from the distribution). The idea is to drop $o(n)$ variables from consideration and test the remaining ones with a threshold such that, with high probability, $\max\{X_1,\ldots,X_n\}$ is larger than this threshold, but only $o(n)$ of these tests are positive. The additional $o(n)$ tests can then be used to find the maximum variable among those that have been tested positively.

1.2 Further Related Work

The original prophet inequality [22] states that there is a 1/2-competitive algorithm in the setting of independent random variables with arbitrary distributions. Initiated by the work of Hajiaghayi, Kleinberg, and Sandholm [18], prophet inequalities have seen a surge of interest in the TCS community over the past 15 years. This has, for instance, led to the development of the tight i.i.d. prophet inequality with competitive ratio approximately 0.745 [10] as well as almost-tight random-order [11, 5] and free-order [2, 25, 29, 5] prophet inequalities. Optimal or near-optimal prophet inequalities can be recovered without knowledge of the distribution but with $O(n)$ samples [7, 31, 8, 6]. Several works considered multiple-choice prophet inequalities with combinatorial constraints, e.g., [3, 13, 21, 12]. We also refer to the (at this point slightly outdated) surveys of Lucier [26] as well as Correa et al. [9] for additional references.

We compare our work with the two works from the prophet-inequality literature that seem closest. Orthogonally to samples, Li, Wu, and Wu [24] considered a version of the unknown i.i.d. setting in which quantile queries to the distribution can be made *before* the sequence of variables arrives. They as well as Perez-Salazar, Singh, and Toriello [30] also used a limited number of (quantile-based) thresholds to achieve near-optimal i.i.d. prophet inequalities. We are not aware of any version of the single-choice prophet inequality with general i.i.d. distributions to which the impossibility of approximately 0.745 does not apply.

In stochastic probing (e.g., [4, 15, 1, 16, 17]), information is also revealed online according to known distributions. The standard models are, however, quite different from our model: The decision maker gets to choose which variables to probe, and each probe entirely reveals the realization of the variable at hand. Eventually, the decision maker can pick a (set of) variable(s), much like in our model. Comparing with an omniscient optimum (like in the prophet inequality) is, however, usually hopeless in this setting. Instead, one focusses on computing or approximating the strategy that maximizes the expected selected (total) value, a task that is straightforward for our model.

In these probing models, the adaptivity gap measures the worst-case multiplicative gap between the value of the best adaptive and that of the best non-adaptive strategy. Note that, while our result does imply a nontrivial adaptivity gap (i.e., larger than 1) for our problem, we are studying a different question as we compare both adaptive and non-adaptive strategies with an omniscient optimum.

We are aware of two works in the probing literature in which tests do not eradicate all uncertainty about the respective variable. Hoefer, Schmand, and Schewior [20] considered a stochastic-probing model in which the first test to a variable only reveals whether the realization is above or below the median of the distribution, and additional tests can be used to further narrow down the realization in the same way applied to the conditional distribution. Gupta et al. [14] generalized the related classic Pandora's box problem due to Weitzman [32] and considered the Markovian model. There, a set of Markov chains, which correspond to variables that can eventually be chosen, is given and, in each step, a probe can be used to advance one of the Markov chains.

Threshold tests have also been considered in the context of estimating (properties of) a probability distribution. For example, Paes Leme et al. [23] gave bounds on the sample complexity, i.e., required number of such tests, to estimate the approximately optimal reserve price for certain types of distributions. Meister and Nietert [27] as well as Okoroafor et al. [28] investigated the sample complexity of estimating other objects, e.g., mean, median, or even full CDF, of the *empirical* distribution in a non-stochastic setting.

2 Preliminaries

We consider *threshold testing* defined as follows. We are given a distribution F on $\mathbb{R}_{\geq 0}$ with finite expectation. There are n *boxes*. Each box i contains a hidden realization $X_1, \dots, X_n$ drawn *once upfront* i.i.d. from F. A *testing algorithm* can apply a *threshold test* to each box $i \in [n] = \{1, \dots, n\}$ exactly once, in that order. To apply a test to box i, the algorithm chooses a threshold $t_i \geq 0$ and receives a binary feedback whether $X_i \geq t_i$ or not. Upon testing i, the algorithm learns if $X_i \geq t_i$ or not, but *not the precise value of* X_i. If $X_i \geq t_i$ we say the test was *positive*, otherwise it was *negative*. The algorithm may choose thresholds adaptively based on the feedback from earlier tests. Finally, after testing each box, the algorithm chooses one box $\sigma \in [n]$ and receives a reward of X_σ. Here, σ is a random variable based on the observed feedback and the internal randomization of the algorithm. We call an algorithm c-competitive if $\mathbb{E}[X_\sigma] \geq c \cdot \mathbb{E}[\max\{X_1, \ldots, X_n\}]$. We are interested in maxmimizing c in the limit as $n \to \infty$.

Non-adaptive Algorithms and Prophet Inequalities. Our testing problem has inherent connections to the classic prophet inequality for i.i.d. random variables. Consider the *non-adaptive* variant, in which the algorithm chooses thresholds $t_1, \ldots, t_n$ upfront. We observe that this problem is essentially the standard gambler's problem governed by prophet inequalities. The optimal algorithm for the gambler's problem emerges from straightforward backwards induction. For each box $i \in [n]$, the gambler sets a threshold t_i to the expected profit from the optimal algorithm for boxes $i+1, \ldots, n$. The algorithm accepts i if and only if $X_i \geq t_i$. It is straightforward to verify that this implies $t_1 \geq \ldots \geq t_n$. All t_i-values can be computed in advance. As such, a non-adaptive algorithm for threshold testing can use these thresholds and imitate the optimal algorithm for the gambler's problem.

▶ **Observation 1.** *The optimal non-adaptive testing algorithm for n boxes obtains at least the expected reward of the optimal algorithm for the gambler's problem with n boxes.*

We also observe the converse – for large n, the optimal reward of non-adaptive threshold testing is essentially the optimal reward in the gambler's problem.

▶ **Proposition 2.** *The optimal non-adaptive testing algorithm for n boxes obtains at most the expected reward of the optimal algorithm for the gambler's problem with $n+1$ boxes.*

Proof. Consider the optimal non-adaptive algorithm for threshold testing. W.l.o.g. we can assume that the chosen thresholds are ordered such that $t_1^* \geq \ldots \geq t_n^*$. If at least one test is positive, then among the positively tested boxes, the algorithm chooses the one with the highest threshold – which is the earliest one in the sequence. The gambler can imitate this in the online model by using thresholds $t_1^*, \ldots, t_n^*$ and accepting the first one with $X_i \geq t_i^*$. If all tests are negative, then the testing algorithm accepts X_1 – it failed the test with the highest threshold and, as such, has the highest conditional expectation. Clearly, this is less than the apriori expectation of F, which can be obtained by the gambler from accepting box X_{n+1}. Hence, the gambler with $n+1$ boxes obtains more expected value. ◀

For large n the best competitive ratio is approximately 0.745 by the optimal prophet inequality [19, 10]. For the rest of the paper we focus on *adaptive* testing algorithms.

Threshold Testing vs. General Binary Feedback. We discuss our scenario in the context of a more general model. In binary-feedback testing, the algorithm can choose a set $Y_i \subset \mathbb{R}_{\geq 0}$ and learns whether or not $X_i \in Y_i$. Note this model generalizes threshold testing – setting a threshold t_i can be simulated by choosing $Y_i = \{x \in \mathbb{R} \mid x \geq t_i\}$. Nevertheless, the competitive ratio achievable is asymptotically the same as for threshold testing. As such, we restrict attention to threshold testing.

▶ **Proposition 3.** *The optimal algorithm for binary-feedback testing with n boxes obtains at most the expected reward of the optimal algorithm for threshold testing with $n+1$ boxes.*

Table 1 Numerically optimized parameters and competitive ratios for different values of k.

k	α_1	α_2	α_3	α_4	comp. ratio as $n \to \infty$
1	1	–	–	–	$1 - 1/e \approx 0.63212$
2	1.83298	0.35932	–	–	> 0.84005
3	2.035135	0.5063	0.05701	–	> 0.86933
4	2.038	0.508	0.058	0.0002	> 0.86956

Proof. Consider an optimal algorithm for binary-feedback testing with n boxes. We modify this algorithm to obtain an algorithm for threshold testing with $n+1$ boxes. We assume w.l.o.g. that, whenever the original algorithm chooses a set Y_i to test the i-th box, then $\mathbb{E}[X_i \mid X_i \in Y_i] \geq \mathbb{E}[X_i]$ and, therefore, $\mathbb{E}[X_i \mid X_i \notin Y_i] \leq \mathbb{E}[X_i]$. We replace any such test with a threshold test for a threshold t_i such that $\Pr[X_i \geq t_i] = \Pr[X_i \in Y_i]$, i.e., both tests are positive with precisely the same probability and $\mathbb{E}[X_i \mid X_i \geq t_i] \geq \mathbb{E}[X_i \mid X_i \in Y_i]$. We continue in the same way as the original algorithm would upon a positive or negative test, modifying subsequent tests in the same way. If the original algorithm eventually picks a box $i^\star$ with a positive test result, the new algorithm picks the same box. Thereby it obtain at least the same value since, by our choice of $t_{i^\star}$, $\mathbb{E}[X_{i^\star} \mid X_{i^\star} \geq t_{i^\star}] \geq \mathbb{E}[X_{i^\star} \mid X_{i^\star} \in Y_{i^\star}]$. Similarly, if the original algorithm would pick a box with a negative result, the new algorithm picks box $n+1$, obtaining $\mathbb{E}[X_{n+1}] = \mathbb{E}[X_{i^\star}] \geq \mathbb{E}[X_{i^\star} \mid X_{i^\star} \notin Y_{i^\star}]$ by our assumption above. ◀

3 Adaptive Testing

In this section, we prove the following theorem. For simplicity, we consider a continuous distribution F throughout the proof. In the following section, we discuss that the result also generalizes to finite discrete distributions.

▶ **Theorem 4.** *There is an efficient $(0.869 - o(1))$-competitive algorithm for threshold testing with a continuous distribution.*

Proof. We consider a class of algorithms that is parameterized by a monotone sequence of quantile parameters $q_1, \ldots, q_k \in (0,1)$ where $q_1 > \ldots > q_k$. For convenience, we assume $q_0 = 1$ and $q_{k+1} = \ldots = q_n = 0$. The algorithm starts by testing for the $1 - q_1$ quantile of F. Since the distribution is continuous, q_1 corresponds to a threshold τ_1 (i.e., τ_1 is such that $Pr[X_i \geq \tau_1] = q_1$). Then for any $j \geq 1$, if the algorithm sees a negative test for τ_j, it continues testing with τ_j. If it sees a positive test for τ_j, it increments j to $j+1$ (i.e., continues testing with the next threshold τ_{j+1}). After having tested each box, it selects the one with the best conditional expectation. This is either the box tested positively for the threshold corresponding to the largest quantile, or any box (when all tested negative for τ_1).

We consider the values of q_j in the form $q_j = \alpha_j/n$ for some $\alpha_j \in (0, n)$, for all $j \in [k]$. In Table 1, we give example values of α_j and the resulting competitive ratios for different values of k. We obtained these values by numerical optimization over a bounded interval.

We use F to denote the CDF, i.e., $F(x) = \Pr[X_i < x]$, for each $i \in [n]$ and $x \in [0,1]$. For the maximum over n i.i.d. draws, we obtain the CDF $F_m(x) = (F(x))^n = (\Pr[X_i < x])^n = \prod_i(\Pr[X_i < x]) = \Pr[\max_i X_i < x]$. We denote the complementary CDF by $G(x) = \Pr[X_i \geq x] = 1 - F(x)$ and $G_m(x) = \Pr[\max_i X_i \geq x] = 1 - F_m(x)$. Since F is continuous, threshold $\tau_j = G^{-1}(q_j) = F^{-1}(1-q_j)$, i.e., $G(\tau_j) = q_j$ and $F(\tau_j) = 1 - q_j$. Similarly, $F_m(\tau_j) = (1-q_j)^n$ and $G_m(\tau_j) = 1 - (1-q_j)^n$. We here restrict attention to values of $\alpha_j \in o(n)$, we will assume these are constants throughout. This implies $\lim_{n\to\infty} G_m(\tau_j) = 1 - \mathrm{e}^{-\alpha_j}$.

Our analysis proceeds via stochastic dominance. For any given threshold $t \geq 0$ we compare the complementary CDF $G_m(t)$ to the complementary CDF of our algorithm. We denote the latter by $A(t) = \Pr[X_\sigma \geq t]$, where σ is the box chosen by our algorithm. If $A(t) \geq c \cdot G_m(t)$ for all $t \geq 0$, then the algorithm is c-competitive by stochastic dominance.

For any given $t \in [0, \infty)$ let $q = G(t) = 1 - F(t)$ and $\alpha = n \cdot q$. We will conduct our analysis with respect to $\alpha \in [0, n]$ instead of $t \in [0, \infty)$. We split $[0, n]$ into intervals $I_j = [\alpha_{j+1}, \alpha_j]$ for $j = 0, \dots, k$, where we use $\alpha_0 = n$ and $\alpha_{k+1} = 0$. Suppose we see a positive test for α_j. Then, between the positive test for α_{j-1} and the one for α_j, assume there are $\ell_j \geq 0$ negative tests.

Two Thresholds. We start by discussing an algorithm with $k = 2$ thresholds. Suppose $\alpha \in I_2$. First, let's assume we only have a positive test for t_1 but not for t_2. We call this event $\mathcal{E}_{10}$. It happens with probability

$$\begin{aligned}\Pr[\mathcal{E}_{10}] &= \sum_{\ell_1=0}^{n-1} (1-q_1)^{\ell_1} q_1 \cdot (1-q_2)^{n-1-\ell_1} = q_1 \cdot (1-q_2)^{n-1} \cdot \frac{1 - \left(\frac{1-q_1}{1-q_2}\right)^n}{1 - \frac{1-q_1}{1-q_2}} \\ &= q_1 \cdot \frac{(1-q_2)^n - (1-q_1)^n}{q_1 - q_2} \quad = \quad \alpha_1 \cdot \frac{\left(1 - \frac{\alpha_2}{n}\right)^n - \left(1 - \frac{\alpha_1}{n}\right)^n}{\alpha_1 - \alpha_2} \quad .\end{aligned}$$

In this case, the algorithm selects the box that was tested positive for τ_1. It has a value at least t with probability $q/q_1 = \alpha/\alpha_1$.

Otherwise, we have a positive test for τ_1 and τ_2, which we call event $\mathcal{E}_{11}$. The event that we have a positive test for τ_1 (irrespective of what happens for τ_2) is called $\mathcal{E}_1$. Clearly,

$$\Pr[\mathcal{E}_{11}] = \Pr[\mathcal{E}_1] - \Pr[\mathcal{E}_{10}].$$

In case $\mathcal{E}_{11}$ happens, we select the box that tested positive for τ_2. It has a value at least t with probability $q/q_2 = \alpha/\alpha_2$.

Overall, for $\alpha \in I_2$ we see

$$\begin{aligned}A(\alpha) &= \frac{\alpha}{\alpha_1} \Pr[\mathcal{E}_{10}] + \frac{\alpha}{\alpha_2} \Pr[\mathcal{E}_{11}] = \alpha \cdot \left(\frac{\Pr[\mathcal{E}_{10}]}{\alpha_1} + \frac{\Pr[\mathcal{E}_1] - \Pr[\mathcal{E}_{10}]}{\alpha_2} \right) \\ &= \alpha \cdot \left(\frac{\Pr[\mathcal{E}_1]}{\alpha_2} - \frac{(\alpha_1 - \alpha_2)\Pr[\mathcal{E}_{10}]}{\alpha_1 \alpha_2} \right) = \frac{\alpha}{\alpha_2} \cdot \left(1 - \left(1 - \frac{\alpha_2}{n}\right)^n\right) \\ &= c_2(\alpha) \cdot \left(1 - \left(1 - \frac{\alpha}{n}\right)^n\right) = c_2(\alpha) \cdot G_m(\alpha) \quad .\end{aligned}$$

Hence,

$$c_2(\alpha) = \frac{\alpha}{\alpha_2} \cdot \frac{1 - \left(1 - \frac{\alpha_2}{n}\right)^n}{1 - \left(1 - \frac{\alpha}{n}\right)^n} \geq \lim_{\alpha \to 0} c_2(\alpha) = \frac{1 - \left(1 - \frac{\alpha_2}{n}\right)}{\alpha_2} \geq \frac{1 - \mathrm{e}^{-\alpha_2}}{\alpha_2} \quad ,$$

since for every given $n \geq 1$ and every $\alpha > 0$, the ratio $\alpha/(1 - (1 - \alpha/n)^n) > 1$, because $\alpha \geq 1 - (1 - \alpha/n)^n$ by concavity of the latter function.

Now for $\alpha \in I_1$, we consider the case with a positive test on τ_1 but not on τ_2. In this case, the box has a value of at least t with probability $q/q_1 = \alpha/\alpha_1$. Alternatively, if we see a positive test for τ_1 and τ_2, the algorithms selects a box with a value of at least t with probability 1. Overall, for $\alpha \in I_1$

$$A(\alpha) = \frac{\alpha}{\alpha_1} \cdot \Pr[\mathcal{E}_{10}] + \Pr[\mathcal{E}_{11}] \;=\; \Pr[\mathcal{E}_1] - \left(\frac{\alpha_1 - \alpha}{\alpha_1}\right) \cdot \Pr[\mathcal{E}_{10}]$$

$$
\begin{aligned}
&= 1 - \left(1 - \frac{\alpha_1 - \alpha}{\alpha_1 - \alpha_2}\right)\left(1 - \frac{\alpha_1}{n}\right)^n - \frac{\alpha_1 - \alpha}{\alpha_1 - \alpha_2}\left(1 - \frac{\alpha_2}{n}\right)^n \\
&= c_1(\alpha) \cdot \left(1 - \left(1 - \frac{\alpha}{n}\right)^n\right).
\end{aligned}
$$

Since $\alpha \in [\alpha_2, \alpha_1]$ is a constant,

$$
\lim_{n\to\infty} c_1(\alpha) = \frac{1}{1 - \mathsf{e}^{-\alpha}} \cdot \left(1 - \mathsf{e}^{-\alpha_1} - \frac{(\alpha_1 - \alpha)(\mathsf{e}^{-\alpha_2} - \mathsf{e}^{-\alpha_1})}{\alpha_1 - \alpha_2}\right).
$$

Finally, for $\alpha \in I_0$, we see that

$$
\begin{aligned}
A(\alpha) = \frac{\alpha - \alpha_1}{n - \alpha_1}(1 - \Pr[\mathcal{E}_1]) + \Pr[\mathcal{E}_1] &= \frac{\alpha - \alpha_1}{n - \alpha_1}\left(1 - \frac{\alpha_1}{n}\right)^n + \left(1 - \left(1 - \frac{\alpha_1}{n}\right)^n\right) \\
&= c_0(\alpha) \cdot \left(1 - \left(1 - \frac{\alpha}{n}\right)^n\right).
\end{aligned}
$$

Thus,

$$
c_0(\alpha) \geq \frac{1 - \left(1 - \frac{\alpha_1}{n}\right)^n}{1 - \left(1 - \frac{\alpha}{n}\right)^n} \geq \frac{1 - \mathsf{e}^{-\alpha_1}}{1 - \mathsf{e}^{-\alpha}},
$$

where the latter bound holds for any $n \geq 1$ and any constant α. Indeed, when $\alpha \in \omega(1)$, we obtain a bound of $1 - \mathsf{e}^{-\alpha}$ in the limit for $n \to \infty$.

As a sanity check, observe that $c_1(\alpha_1) = c_0(\alpha_1) = 1$. Indeed, suppose we have a box with value $t \geq \tau_1$. Then either this box is tested positive for τ_1, or some other box was tested positive for τ_1 before. In either case, the algorithm indeed selects a box of value at least τ_1. Similarly, observe that $c_2(\alpha_2) = 1$ as well. Indeed, suppose we have a box with value $t \geq \tau_2$. Suppose (1) this box is tested positive for τ_2. Then it is selected. Suppose (2) the box is tested positive for τ_1. Then it is selected, unless some later box is tested positive for τ_2. Either way, we eventually obtain a value of at least τ_2. Finally, suppose (3) the box is not tested at all. Then we have already selected a box of value at least τ_2 before.

To obtain the best ratio, we strive to select constants $0 < \alpha_2 < \alpha_1$ in order to

$$
\max_{\alpha_1, \alpha_2} \{\min_{\alpha \in I_2} c_2(\alpha),\ \min_{\alpha \in I_1} c_1(\alpha),\ \min_{\alpha \in I_0} c_0(\alpha)\}.
$$

For $c_2(\alpha)$ and $c_0(\alpha)$ we obtain fairly clear lower bounds, which even hold pointwise for any n. It seems unpromising to obtain an insightful analytic formula for the minimum of $c_1(\alpha)$ as a function of α_1 and α_2. Instead, we numerically optimized parameters α_1, α_2 and used standard solver software to minimize $c_1(\alpha)$. The lower bounds for c_2 and c_0 then amount to $(1 - \mathsf{e}^{-0.35932})/0.35932 \geq 0.8400637\ldots$ and $1 - \mathsf{e}^{1.83298} \geq 0.8400564\ldots$. The minimum of $\lim_{n\to\infty} c_1(\alpha)$ is located roughly at $\alpha^* \approx 0.832961265\ldots$ with a value for $c_1(t) = 0.8400569\ldots$ For a plot of the ratios see Figure 1.

Along similar lines, we analyze the case with $k = 3$ thresholds in the full version, which yields a ratio of at least $0.869 - o(1)$ (see Table 1). Based on similar calculations, we also numerically optimized the case with $k = 4$, but we see only very slight improvements. Intuitively, the probability to reach a state with positive tests for all k thresholds becomes extremely small. Increasing this probability requires to decrease the value to be tested for in the first $k - 1$ tests. However, the possibility to obtain an improvement in this way seems to vanish very quickly as k grows larger. We conjecture that for all values of k, we cannot significantly improve the competitive ratio beyond 0.869 as $n \to \infty$. ◀

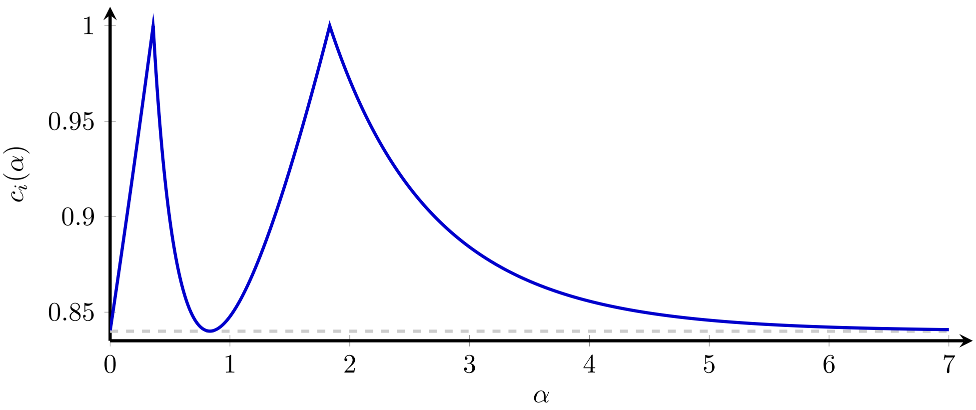

Figure 1 Ratios c_2, c_1 and c_0 in the limit for $n \to \infty$ using $\alpha_1 = 1.83298$ and $\alpha_2 = 0.35932$.

Observe that the analysis of our algorithms is tight. Consider the value of α' that yields the minimum of all $c_i(\alpha)$ in the respective intervals I_i. For a "golden nugget"-distribution, where each X_i has value 1 with probability α'/n and 0 otherwise, the above calculations become exact, and the analysis of the competitive ratio becomes tight. While, strictly speaking, this golden-nugget distribution is discrete, it is straightfoward to approximate it arbitrarily closely by a continuous distribution.

4 Discrete Distributions

Let us shift attention from a continuous distribution to a finite discrete distribution F. We assume F is represented in straightforward form as a list of (value, probability) pairs. We denote by m the number of distinct realizations, and we use $v_1 < v_2 < \ldots < v_m$ to denote the support of F.

Observe that w.l.o.g. we only need to test for these values v_j. If we test for a threshold t in between two consecutive $v_j < t \leq v_{j+1}$, we obtain the same result by testing for $t = v_{j+1}$ instead. As such, we restrict to tests for values in the support.

4.1 Testing Algorithms

We first observe that an optimal testing algorithm can be computed in polynomial time. Moreover, we show that this algorithm yields a competitive ratio of $0.869 - o(1)$.

▶ **Theorem 5.** *For finite discrete distributions, an optimal testing algorithm can be computed by dynamic programming in polynomial time.*

Proof. We use backwards induction. Consider the last test of box n. Clearly, given the previously tested boxes $1, \ldots, n-1$, we can restrict attention to the one with the highest conditional expectation. We denote this value by V^*_{n-1}. Since each box is tested for exactly one of the m realizations, there are $2m$ different possibilities for V^*_{n-1}. There are m possible tests for box n. We can enumerate all the $2m^2$ combinations. For each value of V^*_{n-1}, the optimal test of box n is the one leads to the highest expected value of the chosen item. Thus, to determine and describe the optimal decision for box n, we only need to consider $2m$ options of V^*_{n-1}, and for each option we record the best of the m possible tests for box n.

For the induction, let V^*_{i-1} and V^*_i be the conditional expectation of the best tested box before and after testing box i, resp. Suppose that for each possible value of V^*_i, we have computed an optimal testing strategy for subsequent boxes $i+1, \ldots, n$, along with the resulting expected value of X_σ. Now for box i, consider each of the $2m$ possible values for V^*_{i-1}. For each realization v_k, we can determine the effect when we test box i for v_k – i.e., the probability that $V^*_i = V^*_{i-1}$ (when the test on i implies the conditional expectation of i is at most V^*_{i-1}), as well as the probability that V^*_i becomes any higher value (otherwise). For the resulting V^*_i, we inspect the value obtained by an optimal testing strategy for boxes $i+1,\dots,n$. This serves to find the test of box i resulting in the optimal expected value.

Overall, to determine and describe the optimal decision for box i, we need to consider $2m$ options of V^*_{i-1}, and for each option we determine the best of the m possible tests for box i (using the results of the subsequent optimal testing strategy for boxes $i+1,\dots,n$). Finally, for box 1 V^*_0 is undefined. At this point, we only need to find the best of the m possible tests for box 1 (using the results of the subsequent optimal testing strategy for boxes $2,\dots,n$). This concludes the backwards induction.

We record for each possible value V^*_{i-1} the best threshold to test box i along with the resulting expected value emerging from an optimal algorithm for boxes $i+1,\dots,n$. Hence, we can describe an optimal testing strategy using $2\cdot(1+(n-1)\cdot 2m))$ entries. This strategy can be computed in polynomial time via dynamic programming as described above. ◀

At this point, it is unclear how to apply our algorithm from the previous section to finite discrete distributions since $F^{-1}(1-q)$ may not be defined for the relevant values of q. In fact, we will show that the optimal algorithm in Theorem 5 achieves a competitive ratio of at least $0.869 - o(1)$ for finite discrete distributions.

We consider the following different model for testing discrete distributions, called *probability testing*. It can be viewed as the limit that emerges from approximating discrete with continuous distributions arbitrarily close. Here a test requires an input value $q \in [0,1]$. It then returns whether or not the value v in the box lies in the top-q fraction of probability mass of F. For a finite discrete distribution F, let k be such that $\sum_{j=k+1}^m \Pr[v = v_j] < q \le \sum_{j=k}^m \Pr[v = v_j]$. Then the test is positive for $v \in \{v_{k+1},\dots,v_m\}$ and negative for $v \in \{v_1,\dots,v_{k-1}\}$. For $v = v_k$ it yields a randomized outcome, i.e., positive with probability $p_q = \left(q - \sum_{j=k+1}^m \Pr[v = v_j]\right) / \Pr[v = v_k]$ and negative otherwise. Hence, the overall probability that box i is tested positive is exactly q.

Clearly, our algorithm from Section 3 can be implemented with probability testing and obtains a competitive ratio of $0.8969 - o(1)$. Probability and threshold testing are equivalent for continuous distributions, since there is a bijection between thresholds and values for q. For finite discrete distributions we observe in Proposition 6 that any algorithm for probability testing can be simulated using randomized threshold tests. We then show that randomized tests are not beneficial, i.e., for any algorithm with randomized threshold tests, there is one with *deterministic* tests performing at least as good. For a formal proof, see the full version.

▶ **Proposition 6.** *If there is a c-competitive algorithm for probability testing, then there is a c-competitive algorithm for threshold testing.*

▶ **Corollary 7.** *The optimal algorithm for finite discrete distributions is at least* $(0.869-o(1))$*-competitive for threshold testing.*

4.2 Impossibility

Complementing our results in the previous subsection, we proceed to show a constant upper bound on the competitive ratio for $n \to \infty$.

▶ **Theorem 8.** *There exists no* $(1-o(1))$*-competitive algorithm for threshold testing.*

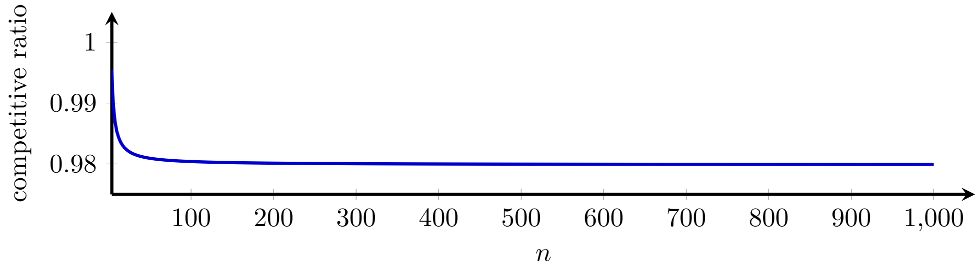

Figure 2 The best-possible competitive ratio on the instance used in the proof of Theorem 8 as a function of n.

To prove the theorem, we are going to construct a counter example that is a discrete distribution, which carries over to the continuous case by the arguments given in Section 4. We first observe that such a distribution needs to depend on n: Otherwise, the top realization appears with constant probability in each box, and an algorithm simply testing for that realization finds it with probability $1 - o(1)$. Furthermore, such a distribution needs to have a support of cardinality at least 4: If the cardinality of the support is 3, it is w.l.o.g. exactly 3, and the algorithm can obtain $\max\{X_1, \dots, X_n\}$ by testing for the middle realization and, upon a positive test, testing for the top realization. If it finds a positive test on the top realization, it clearly obtains $\max\{X_1, \dots, X_n\}$ by choosing the corresponding box. If it finds a positive test on the middle realization, the corresponding box is the only one that can possibly contain the top realization, which the algorithm obtains by picking it, so it also obtains $\max\{X_1, \dots, X_n\}$. In the final case, $\max\{X_1, \dots, X_n\}$ is only the lowest realization, which the algorithm will also obtain by choosing any box.

We consider boxes that contain a realization 3, 2, or 1 with probability $1/n$ each and 0 otherwise. Intuitively, any algorithm that does not always test for the value 1 before encountering a positive test runs the risk of missing a value 1. Similarly, any algorithm that does not always test for the value 2 afterwards and before encountering another positive test runs the risk of missing a value 2. Such an algorithm, however, with constant probability, gets into a situation in which it has encountered precisely two positive tests, specifically, for the values 1 and 2. In that situation, it is clearly optimal to choose the box that has been positively tested for the value 2. With a constant probability, the value of this box is, however, equal to 2 while the one that has been tested positively for value 1 is equal to 3. The conclusion is that the algorithm, in any case, loses a constant fraction of $\mathbb{E}[\max\{X_1, \dots, X_n\}]$. In the full version, we present a formal version of this argument.

We have verified numerically (by solving the dynamic program from Theorem 5) that for this distribution the achievable competitive ratio decreases in n in the interval $n = 2, \dots, 1000$. For $n = 1000$, the optimal competitive ratio is ca. 0.9799 (computed with full precision). See Figure 2 for the results.

5 Multiple Tests per Box

In this section, we consider a setting with n boxes and a budget of n threshold tests. Each box can be tested an arbitrary number of times with different thresholds[2] as long as there are still tests available. We again assume continuous distributions and show the following result.

[2] Recall that nature draws *initially a single* value $X_i \sim F$ inside each box i. All tests on the same box are evaluated accordingly. The results of multiple tests on the same box are all consistent with the single unknown X_i drawn upfront.

▶ **Theorem 9.** *There is an efficient* $(1 - o(1))$*-competitive algorithm for threshold testing with multiple tests per box and a continuous distribution.*

Proof. In the first step, our algorithm discards the last $\lceil n^{2/3} \rceil$ boxes, losing only a $\lceil n^{2/3} \rceil / n$ fraction of the value. The remaining ones are tested for the threshold $F^{-1}(1 - n^{-1/3})$. Let P be the set of boxes that were tested positively. For each box $i \in P$, the algorithm next searches the integers $\{0, \ldots, \lfloor n^{2/3} \rfloor\}$ to find the largest j such that the test for $F^{-1}(1 - n^{-1/3} + j/n)$ is negative. Then[3], $F^{-1}(1 - n^{-1/3} + j/n) < X_i \leq F^{-1}(1 - n^{-1/3} + (j+1)/n)$. Using a binary search, this requires at most $\lceil 2/3 \cdot \log n \rceil$ tests for each box $i \in P$. Using the result, we say that box i is of type j. Since there are potentially up to n boxes in P, the algorithm may well run out of tests during this process. Eventually, if the algorithm succeeds to determine the type of each box in P, it picks a box from P with the highest type. If no such box exists, it is not unique, or the algorithm ran out of tests before determining the type of each box in P, it may choose an arbitrary box.

To analyze our algorithm, we fix any $v \in [F^{-1}(1 - n^{-1/3}), F^{-1}(1))$. We denote by $\mathcal{M}_v$ the event that $\max\{X_1, \ldots, X_n\} = v$, and by X_σ the value obtained by the algorithm. Our goal is to show that, whenever an optimal box has a high value v, the probability that we choose an optimal box is

$$\Pr[X_\sigma = v \mid \mathcal{M}_v] = 1 - o(1). \tag{1}$$

Now we see an optimal box with a high value with probability

$$\Pr[\max\{X_1, \ldots, X_n\} \geq F^{-1}(1 - n^{-1/3})] = 1 - (1 - n^{-1/3})^n = 1 - o(1),$$

so proving Eq. (1) indeed suffices to prove the theorem. To show Eq. (1), we define two additional events:

- $\mathcal{E}_1$ is the event that $|P| \leq n^{1/2}$ (in particular, this implies that the algorithm does not run out of tests for large-enough n),
- $\mathcal{E}_2$ is the event that only a single box has the largest type.

Note that $\Pr[X_\sigma = v \mid \mathcal{M}_v] \geq \Pr[\mathcal{E}_1 \cap \mathcal{E}_2 \mid \mathcal{M}_v]$ since our algorithm chooses the box with the maximum value if both $\mathcal{E}_1$ and $\mathcal{E}_2$ occur. We finalize the argument by observing that

$$\begin{aligned} \Pr[\mathcal{E}_1 \cap \mathcal{E}_2 \mid \mathcal{M}_v] &= \Pr[\mathcal{E}_1 \mid \mathcal{M}_v] \cdot \Pr[\mathcal{E}_2 \mid \mathcal{M}_v \cap \mathcal{E}_1] \\ &\geq \left(1 - e^{-\frac{n^{1/3}}{3}}\right) \cdot \left(1 - \frac{n^{-2/3}}{1 - n^{-1/3}}\right)^{n^{1/2}} = 1 - o(1). \end{aligned}$$

For the inequality, we bound the first probability using a one-sided multiplicative Chernoff bound with $\mu = n^{1/3}$ and factor 2. We bound the second probability by observing that, conditioned on $\mathcal{M}_v$, the probability that a single box other than that with realization v has the same type is at most $n^{-2/3}/(1 - n^{-1/3})$. Here, $n^{-2/3}$ is an upper bound on the probability of having the same type, and $1 - n^{-1/3}$ is a lower bound on the probability that an independently drawn value is below v (using that v is a high value). The additional condition on $\mathcal{E}_1$ does not increase the probability of $\mathcal{E}_2$. This shows Eq. (1) and thus completes the proof. ◀

It is rather straightforward to apply the insights from Sec. 4 to show similar results for testing with multiple tests per box and a finite discrete distribution. Our algorithm in the proof of Theorem 9 can be cast as a *sequential* testing algorithm: It tests boxes

[3] We use the convention $F^{-1}(x) = F^{-1}(1)$ for $x > 1$.

sequentially from box 1 to $n - \lceil n^{2/3} \rceil$. For each box i it applies tests to determine whether $i \in P$ or not, and then binary search the type of i (or aborts when it runs out of tests). For finite discrete distributions, we can optimize over such sequential testing algorithms using backwards induction, much like in the proof of Theorem 5. When considering box i, an optimal decision about the next test can be found by relying on three additional parameters. Apart from the best conditional expectation of a previous box V^*_{i-1}, we also consider the smallest realization for which we saw a positive test for i, the largest one for which we saw a negative test for i, as well as the number of tests we applied so far. These parameters sufficiently describe the current state of the system before applying the next test. Note that there is only a polynomial number of combinations of these parameters that need to be considered. Then the algorithm has up to m possible options for the next test of box i – or m possible options for the first test of box $i + 1$, thereby concluding the testing of box i. Hence, there are only polynomially many combinations that need to be considered to find the optimal decision for the current test (assuming that an optimal testing algorithm for the subsequent number of tests/boxes has already been computed via backwards induction).

We can also transfer the approximation guarantee for the algorithm from Theorem 9. We apply the algorithm in the model with probability tests and interpret them as randomized threshold tests. By applying the arguments of Proposition 6 to the sequential model with multiple tests per box, we see that for every randomized threshold testing algorithm there is a deterministic one that performs at least as good. Overall, this yields the following corollary.

▶ **Corollary 10.** *For finite discrete distributions, an optimal sequential testing algorithm for multiple tests per box can be computed by dynamic programming in polynomial time. It is at least* $(1 - o(1))$*-competitive for threshold testing with multiple tests per box.*

6 Conclusion

In this paper, we have initiated the study of threshold testing of i.i.d. random variables, a probing model with partial revelation and binary feedback. For non-adaptive algorithms, the model is essentially equivalent to the standard gambler's problem, and optimal performance is governed by the i.i.d. prophet inequality of approximately 0.745. For adaptive algorithms, we obtain a testing algorithm with competitive ratio of 0.869. This significantly outperforms 0.745, proves that there is a substantial adaptivity gap, and reveals the structural difference of the adaptive problem. Moreover, we show a constant upper bound on the ratio achievable by any adaptive testing algorithm. In contrast, when we can (adaptively) apply multiple tests to a single box, it is possible to achieve even a ratio of $1 - o(1)$.

There are many intriguing open problems arising from our work. Obviously, the current upper and lower bounds for the i.i.d. model are not tight. More generally, a simple argument similar to Observation 1 shows that free-order prophet inequalities [5] transfer directly to non-adaptive threshold testing, even for non-i.i.d. boxes. It is an intriguing open problem whether these guarantees can be strictly improved using an adaptive testing algorithm. Can we obtain a ratio strictly larger than 0.745 also for non-i.i.d. threshold testing?

In addition, there are many combinatorial versions of the problem that deserve attention, i.e., when the algorithm is allowed to select more than one box. Testing algorithms for, e.g., knapsack, matroid, or general downward-closed feasibility structures represent a natural and important direction for future research.

References

1 Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 29–40, 2014.

2 S. Agrawal, J. Sethuraman, and X. Zhang. On optimal ordering in the optimal stopping problem. In *ACM Conference on Economics and Computation (EC)*, pages 187–188, 2020.

3 Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM J. Comput.*, 43(2):930–972, 2014.

4 Arash Asadpour, Hamid Nazerzadeh, and Amin Saberi. Stochastic submodular maximization. In *Workshop on Internet and Network Economics (WINE)*, pages 477–489, 2008.

5 Archit Bubna and Ashish Chiplunkar. Prophet inequality: Order selection beats random order. *CoRR*, abs/2211.04145, 2022. `arXiv:2211.04145`.

6 José R. Correa, Andrés Cristi, Boris Epstein, and José A. Soto. Sample-driven optimal stopping: From the secretary problem to the i.i.d. prophet inequality. *CoRR*, abs/2011.06516, 2020. `arXiv:2011.06516`.

7 José R. Correa, Paul Dütting, Felix A. Fischer, and Kevin Schewior. Prophet inequalities for independent and identically distributed random variables from an unknown distribution. *Math. Oper. Res.*, 47(2):1287–1309, 2022.

8 José R. Correa, Paul Dütting, Felix A. Fischer, Kevin Schewior, and Bruno Ziliotto. Unknown I.I.D. prophets: Better bounds, streaming algorithms, and a new impossibility (extended abstract). In *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 86:1–86:1, 2021.

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

10 José R. Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. Posted price mechanisms and optimal threshold strategies for random arrivals. *Math. Oper. Res.*, 46(4):1452–1478, 2021.

11 José R. Correa, Raimundo Saona, and Bruno Ziliotto. Prophet secretary through blind strategies. *Math. Program.*, 190(1):483–521, 2021.

12 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.

13 Oliver Göbel, Martin Hoefer, Thomas Kesselheim, Thomas Schleiden, and Berthold Vöcking. Online independent set beyond the worst-case: Secretaries, prophets, and periods. In *International Colloqium on Automata, Languages, and Programming (ICALP)*, pages 508–519, 2014.

14 Anupam Gupta, Haotian Jiang, Ziv Scully, and Sahil Singla. The markovian price of information. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 233–246, 2019.

15 Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 205–216, 2013.

16 Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1731–1747, 2016.

17 Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1688–1702. SIAM, 2017.

18 Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 58–65, 2007.

19 T. P. Hill and R. P. Kertz. Comparisons of stop rule and supremum expectations of i.i.d. random variables. *Annals of Probability*, 10(2):336–345, 1982.

20 Martin Hoefer, Kevin Schewior, and Daniel Schmand. Stochastic probing with increasing precision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4069–4075, 2021.

21 Robert Kleinberg and S. Matthew Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games Econ. Behav.*, 113:97–115, 2019.

22 U. Krengel and L. Sucheston. Semiamarts and finite values. *Bull. Amer. Math. Soc.*, 83:745–747, 1977.

23 Renato Paes Leme, Balasubramanian Sivan, Yifeng Teng, and Pratik Worah. Pricing query complexity of revenue maximization. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 399–415. SIAM, 2023.

24 Bo Li, Xiaowei Wu, and Yutong Wu. Query efficient prophet inequality with unknown I.I.D. distributions. *CoRR*, abs/2205.05519, 2022. `arXiv:2205.05519`.

25 Allen Liu, Renato Paes Leme, Martin Pál, Jon Schneider, and Balasubramanian Sivan. Variable decomposition for prophet inequalities and optimal ordering. In *ACM Conference on Economics and Computation (EC)*, page 692, 2021.

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

27 Michela Meister and Sloan Nietert. Learning with comparison feedback: Online estimation of sample statistics. In *Algorithmic Learning Theory (ALT)*, volume 132, pages 983–1001, 2021.

28 Princewill Okoroafor, Vaishnavi Gupta, and Robert Kleinberg. Non-stochastic CDF estimation using threshold queries. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3551–3572. SIAM, 2023.

29 Bo Peng and Zhihao Gavin Tang. Order selection prophet inequality: From threshold optimization to arrival time design. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 171–178, 2022.

30 Sebastian Perez-Salazar, Mohit Singh, and Alejandro Toriello. The IID prophet inequality with limited flexibility. *CoRR*, abs/2210.05634, 2022. `arXiv:2210.05634`.

31 Aviad Rubinstein, Jack Z. Wang, and S. Matthew Weinberg. Optimal single-choice prophet inequalities from samples. In *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 60:1–60:10, 2020.

32 Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47:641–654, 1979.