Sample-Based Matroid Prophet Inequalities

HU FU, Shanghai University of Finance and Economics, China
PINYAN LU, Shanghai University of Finance and Economics, China
ZHIHAO GAVIN TANG, Shanghai University of Finance and Economics, China
HONGXUN WU, UC Berkeley, USA
JINZHAO WU, Yale University, USA
QIANFAN ZHANG, Princeton University, USA

We study matroid prophet inequalities when distributions are unknown and accessible only through samples. While single-sample prophet inequalities for special matroids are known, no constant-factor competitive algorithm with even a sublinear number of samples was known for general matroids. Adding more to the stake, the single-sample version of the question for general matroids has close (two-way) connections with the long-standing matroid secretary conjecture.

In this work, we give a $(1/4 - \varepsilon)$-competitive matroid prophet inequality with only $O_\varepsilon(\text{poly} \log n)$ samples. Our algorithm consists of two parts: (i) a novel *quantile-based* reduction from matroid prophet inequalities to online contention resolution schemes (OCRSs) with $O_\varepsilon(\log n)$ samples, and (ii) a $(1/4 - \varepsilon)$-selectable matroid OCRS with $O_\varepsilon(\text{poly} \log n)$ samples which carefully addresses an adaptivity challenge.

Authors' Contact Information: Hu Fu, fuhu@mail.shufe.edu.cn, ITCS, Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai University of Finance and Economics, China; Pinyan Lu, lu.pinyan@mail.shufe.edu.cn, ITCS, Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai University of Finance and Economics, China; Zhihao Gavin Tang, tang.zhihao@mail.shufe.edu.cn, ITCS, Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai University of Finance and Economics, China; Hongxun Wu, wuhx@berkeley.edu, UC Berkeley, USA; Jinzhao Wu, jinzhao.wu@yale.edu, Yale University, USA; Qianfan Zhang, qianfan@princeton.edu, Princeton University, USA.

1   Introduction

In the classical prophet inequality problem introduced by Krengel and Sucheston [43, 44], n items, with values $v_1, v_2, \ldots, v_n$ drawn independently from known distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$, arrive one by one. An algorithm, upon observing the value of each item, must decide irrevocably whether to take it or discard it forever. The algorithm can take at most one item in total, and aims to maximize the expected value of the item taken. we measure the performance of an algorithm via its *competitive ratio*: an algorithm is *c-competitive* (for $c \in [0, 1]$) if its expected value is at least c fraction of what can be achieved by a *prophet*, who knows the realizations of the values and always takes the highest value $\max_i v_i$, getting in expectation $\mathbf{E}[\max_i v_i]$. Krengel, Sucheston, and Garling [43, 44] showed that $1/2$ is the optimal competitive ratio in this so-called single-item setting.

Beyond interest of their own as canonical optimal stopping problems, prophet inequalities also have strong connections with online mechanism design and posted price mechanisms [11, 34]. Many variants beyond the single-item case have been studied extensively (e.g., [1, 2, 12, 15, 22, 23, 26, 27, 29, 31, 33, 38, 42, 50–52]). One of the significant extensions is *matroid prophet inequalities* discovered by Kleinberg and Weinberg [42], where the set of items taken is subject to a matroid feasibility constraint. The single-item setting is the special case when the matroid is the 1-uniform matroid. Kleinberg and Weinberg [42] presented a $1/2$-competitive algorithm that works for any matroid, which matches the best possible ratio even in the single-item case.

Many existing works on prophet inequalities assumed complete knowledge of distributions, which may be unrealistic both in practice and for some applications such as prior-independent mechanism design [4, 19, 37]. It is natural to ask whether similar performance can be achieved with only limited access to distributions (e.g., historical data points). This question was first formulated as *prophet inequalities with limited information* (specifically, *sample-based* prophet inequalities) by Azar et al. [3], where the online algorithm only has access to a limited number of samples from each distribution $\mathcal{D}_i$. Such a sample-based paradigm is closer to practice and naturally provides some robustness against distribution shifts.

More specifically, there are two particularly interesting facets of the problem:

1. *What is the best competitive ratio achievable with a single/constant number of samples?*
2. *What is the least number of samples needed to achieve a constant competitive ratio?*

The first facet was studied by Azar et al. [3]. They showed that a single sample from each distribution suffices for constant-factor competitive prophet inequalities for a number of feasibility constraints. Since then, there have been many subsequent works on different settings such as single-item [53], identical distributions [13, 14, 17, 40], non-adversarial arrivals [16, 18], and various other feasibility constraints [9, 41]. While some of these works studied sample-based prophet inequalities for specific classes of matroids, such as graphic or transversal matroids, not much was known for general matroids. Notably, Azar et al. [3] design single-sample prophet inequalities via a black-box reduction to a large subclass of algorithms for *secretary problems*[1]. Based on the reduction, they obtained a single-sample $1/O(\log \log \text{rank})$-competitive prophet inequality for general matroids from the state-of-the-art matroid secretary algorithm [30, 45] with the same ratio.

It turns out that the sample-based prophet inequalities and the secretary problem are deeply connected. In fact, solving either facets completely requires resolving the famous matroid secretary conjecture by Babaioff et al. [5]. For example, Caramanis et al. [9] discovered a partial reverse to the aforementioned reduction of Azar et al. [3]. Very recently, Li [48] obtained a black-box

---

[1]The secretary problem [25] is another classical model in online decision making that is closely related to prophet inequalities, where values are chosen adversarially rather than sampled from distributions, but items arrive in a uniformly random order.

reduction from the secretary problem to single-sample prophet inequalities (which also extends to the constant-sample prophet inequalities). As a consequence, simultaneously achieving constant competitive ratio and constant number of samples implies a constant-factor competitive algorithm for matroid secretary problem, which would resolve the matroid secretary conjecture.

In this work, we make progress on the second facet of the problem. We show that $O_\varepsilon(\log^4 n)$ samples suffice for $(1/4 - \varepsilon)$-competitive prophet inequalities for general matroids. Before our work, with even $o(n)$ samples, no constant-factor competitive prophet inequalities for general matroids were known. This is due to the limitations of previous techniques:

- In the original algorithm of Kleinberg and Weinberg [42], upon the arrival of each item, its value is compared against a threshold. Crucially, this threshold is the expectation of a random variable that depends on the items currently taken.
  - There is the *lack-of-concentration issue* with this strategy: Since the values of the items are unbounded, there is no concentration with any bounded number of samples. This prevents us from estimating the expectation threshold to any non-trivial accuracy.
  - Even if the values are bounded and we have concentration, there is still the *adaptivity issue*: The random variable for the item i depends on whether items $1, 2, \ldots, i-1$ are accepted and, therefore, depends on previous estimates. Even when there is concentration, one still cannot union bound over such adaptive estimations.
- The approach of Feldman et al. [31] reduces prophet inequalities to online contention resolution schemes (OCRSs). The reduction also needs to set similar thresholds.
  - It partially gets around the *lack-of-concentration issue* by directly using the realization of some random variable as the threshold (rather than the expectation). To generate one such realization, they need one sample from each of the distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$. Due to the requirement of OCRSs, they must use independent realizations for the n items, this already requires $\Omega(n)$ samples.
  - Although there is no adaptivity in the reduction, their OCRSs also require full knowledge of the distributions. If one tries to implement the OCRSs with samples, a similar *adaptivity issue* shows up.

We overcome these two issues with new techniques. Our first technical contribution is a quantile-based strategy for setting thresholds (in the reduction to OCRSs) and its analysis. Roughly speaking, we use the *median* of some random variables as thresholds, which is reminiscent of the classical Samuel-Cahn [54] result for the rank-1 case. With a few samples, the estimate of the median is guaranteed to be a $(1/2 - \varepsilon, 1/2 + \varepsilon)$ quantile for some $\varepsilon > 0$ with high probability. This overcomes the *lack-of-concentration issue*. The analysis of this strategy is highly nontrivial and involves a weighted generalization of matroid strong basis exchange lemma from submodular optimization literature. To the best of our knowledge, it has not been applied to matroid prophet inequalities before. More details are discussed in Section 3.1.

Our second contribution is a sample-based matroid OCRS overcoming the *adaptivity issue*, which is an interesting result on its own. To obtain such OCRS, we necessarily need to randomize the matroid OCRS of Feldman et al. [31] in an ingenious way to handle adaptivity. The approach we use is specific to OCRSs and differs from the usual randomization methods in adaptive data analysis. More details are discussed in Section 4. In conclusion, we believe that this work brings interesting techniques to the study of matroid prophet inequalities with limited information.

1.1 Further Related Works

Feldman et al. [31] introduce the concept of Online Contention Resolution Schemes (OCRSs) and present a $\frac{1}{4}$-selectable OCRS for matroids. Our sampled-based OCRS in Section 4 is based on their

algorithm. This algorithm is later improved by Lee and Singla [46] who obtain a $\frac{1}{2}$-selectable matroid OCRS by a clever reduction to the matroid prophet inequalities problem. OCRSs are also considered in other settings including matching [27], knapsack problem [39], and k-uniform matroids [20, 39]. Recent studies have expanded to investigate OCRSs with limited information. Fu et al. [32] show that there exists a $\frac{1}{e}$-selectable oblivious OCRS for rank-1 matroids. In contrast, for graphical and transversal matroids, they prove that no OCRS utilizing a constant number of samples can attain a constant selectability. Santiago et al. [55] address a scenario in which the marginal active probability x_e is disclosed upon the arrival of element e. They provide a constant selectable random-order OCRS for graphical matroids.

Online decision making with limited information is attracting growing attention. In addition to the sample-based prophet inequalities discussed earlier, Li et al. [47] explore I.I.D. prophet inequalities with only access to quantile queries and Nuti and Vondrák [49] study maximizing the probability of selecting the largest number in the secretary problem with a single sample. There are some works on posted-price mechanisms that also imply prophet inequalities with polynomially many samples [6, 22].

2 Preliminaries

Throughout the paper, we use bold letters v, x to denote vectors and v_i, x_i to denote their entries.

Prophet Inequalities. In the prophet inequality problem, we are given a ground set of items $U = [n]$, a downward-closed family of feasible sets $\mathcal{F} \subseteq 2^U$, and a distribution $\mathcal{D}_i$ for each $i \in U$. Items arrive in a fixed order $1, 2, \ldots, n$.[2] As each item arrives, a value v_i independently drawn from $\mathcal{D}_i$ is revealed to the algorithm, and an irrevocable decision must be made whether to include the item in the output A, while keeping $A \in \mathcal{F}$.

For $c \in [0, 1]$, we say the algorithm induces a *c-competitive* prophet inequality for $\mathcal{F}$ if, for any $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$,

$$\mathbf{E}\left[\sum_{i \in A} v_i\right] \geq c \, \mathbf{E}\left[\max_{S \in \mathcal{F}} \sum_{i \in S} v_i\right]$$

where the expectation is taken with respect to the joint distribution of $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$ and internal randomness of the algorithm (including samples in the limited information setting).

In the limited information setting, instead of distributions $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$, the algorithm will take a number of samples from each $\mathcal{D}_i$ as the offline input. We say an algorithm uses k samples if it takes k samples from each $\mathcal{D}_i$.

Online Contention Resolution Schemes (OCRSs). Given a ground set of elements $U = [n]$ and a downward-closed family of feasible sets $\mathcal{F} \subseteq 2^U$, we can define the polytope of $\mathcal{F}$ as the convex hull of all characteristic vectors of feasible sets:

$$\mathcal{P}_{\mathcal{F}} = \text{conv}(\{\mathbf{1}_I \mid I \in \mathcal{F}\}) \subseteq [0, 1]^n.$$

An OCRS for $\mathcal{P}_{\mathcal{F}}$ takes an vector $x \in \mathcal{P}_{\mathcal{F}}$ as input. Let $R(x) \subseteq U$ be a random set where each element $i \in U$ is in $R(x)$ with probability x_i independently. The algorithm sees the membership in $R(x)$ of the elements in U arriving in a fixed order $1, 2, \ldots, n$;[2] when each element $i \in U$ arrives, if it is active (i.e., $i \in R(x)$), the OCRS must decide irrevocably whether to include i in its output A while keeping $A \in \mathcal{F}$.

---

[2]Our sample-based algorithm for matroid prophet inequality/OCRS actually works against *almighty adversary*, who determines the arrival order adaptively with full knowledge of all realizations of randomness and the decisions made by the algorithm (see Remark 3.7 for details). Nevertheless, we assume a fixed order (i.e., against *offline adversary*) throughout the paper for ease of reading.

For $c \in [0, 1]$, an OCRS is said to be *c-selectable* for $\mathcal{F}$ if, for any $\boldsymbol{x} \in \mathcal{P}_{\mathcal{F}}$,

$$\mathbf{Pr}[i \in A \mid i \in R(\boldsymbol{x})] \geq c \quad \forall i \in U$$

where the probability is measured with respect to $R(\boldsymbol{x})$ and internal randomness of the OCRS (including samples in the limited information setting).

In the limited information setting, instead of $\boldsymbol{x}$, the algorithm will take a number of samples of $R(\boldsymbol{x})$ as the offline input. We say an OCRS uses k samples if it takes k samples of $R(\boldsymbol{x})$.

Matroids. A matroid $\mathcal{M} = (U, \mathcal{I})$ is defined by a ground set of elements $U = [n]$ and a non-empty downward-closed family of *independent sets* $\mathcal{F} \subseteq 2^U$ with the *exchange property*, i.e., for every $A, B \in \mathcal{I}$ where $|A| > |B|$, there exists some $i \in A \setminus B$ such that $B \cup \{i\} \in \mathcal{I}$. Some notations for matroids are used throughout the paper:

- The *rank* of a set $S \subseteq U$ is the size of the largest independent set contained in S:

$$\mathrm{rank}(S) = \max\{|I| \mid I \in \mathcal{I}, I \subseteq S\}.$$

- The *span* of a set $S \subseteq U$ is the set of elements that is not independent from S:

$$\mathrm{span}(S) = \{i \in U \mid \mathrm{rank}(S) = \mathrm{rank}(S \cup \{i\})\}.$$

- An independent set $S \in \mathcal{I}$ is called a *basis* if it spans the ground set, i.e., $\mathrm{span}(S) = U$.
- The *restriction* of $\mathcal{M}$ to a set $S \subseteq U$ is a matroid $\mathcal{M}|_S = (S, \mathcal{I}|_S) = (S, \{I \in \mathcal{I} \mid I \subseteq S\})$.
- The *contraction* of $\mathcal{M}$ by a set $S \subseteq U$ is a matroid $\mathcal{M}/S = (U \setminus S, \mathcal{I}')$ where

$$\mathcal{I}' = \{I \in \mathcal{I}|_{U \setminus S} \mid \mathrm{rank}(I) + \mathrm{rank}(S) = \mathrm{rank}(I \cup S)\}.$$

The polytope of a matroid $\mathcal{M}$ can be represented as

$$\mathcal{P}_{\mathcal{M}} = \left\{ \boldsymbol{x} \in [0, 1]^n \;\middle|\; \sum_{i \in S} x_i \leq \mathrm{rank}(S), \forall S \subseteq U \right\}.$$

3 Matroid Prophet Inequalities from Samples

In this section, we reduce the matroid prophet inequality to matroid OCRS using only $O_\varepsilon(\log n)$ samples. Together with our $O_\varepsilon(\log^4 n)$-sample matroid OCRS, which we will present in Section 4, it proves the following theorem.

Theorem 3.1. *With $O_\varepsilon(\log^4 n)$ samples, there is a $(\frac{1}{4} - \varepsilon)$-competitive prophet inequality for general matroids of size n and any $\varepsilon > 0$.*

3.1 Overview of Our Techniques

Before getting into our matroid prophet inequalities, let us first sketch the main idea and comparison with prior works.

Notations. For simplicity, we now define a minimum set of notations that we will only use in this subsection. In the following, we assume that the items arrive in order $1, 2, \ldots, n$ and A_i is the set of items accepted by our prophet inequalities among the first i items.

For subsets $S, A \subseteq [n]$, we will use the random variable

$$\mathrm{OPT}^v(S, A) \stackrel{\text{def}}{=} \max\{v(I) \mid I \subseteq S, I \cup A \in \mathcal{I}\}$$

to denote the maximum-weight subset $I \subseteq S$, such that, $I \cup A$ is an independent set. For any independent set I, we define

$$\tau_i^{\boldsymbol{v}}(I) \stackrel{\text{def}}{=} \min\{v_j \mid j \in I, I - j + i \in \mathcal{I}\}$$

to be the minimum value among the elements $j \in I$ that i can exchange with. The existence of j is guaranteed by the exchange property of matroids.

Approaches in prior works. Most matroid prophet inequalities follow a very general threshold-based framework:

- Before the arrival of an item i, select a threshold T_i. If the value of i exceeds the threshold, we call it an active item.
- Then either accepts all active items whenever possible, or performs contention resolution over the active items.

Therefore, the key to designing matroid prophet inequalities is choosing the appropriate thresholds $\{T_i\}_{i \in [n]}$. Below, we list several popular choices. and explain why they break down in the sampled-based setting:

- (*Expectation-based.*) In the optimal matroid prophet inequalities by Kleinberg and Weinberg [42], T_i is set to $\frac{1}{2} \mathbf{E}_v[\tau_i^v(\text{OPT}^v([n] \setminus A_{i-1}, A_{i-1}))]$. After their arrivals, all active items are accepted.

  Such expectation-based thresholds T_i are ubiquitous in the prophet inequalities and posted-price mechanisms literature (e.g. [29, 33]). However, in sample-based setting, consider an item which has a large value n^{10} with probability $1/n$ and otherwise has a value of 0. With only $O(\text{poly} \log n)$ samples, one cannot hope to see it realized with the large value, and therefore cannot estimate the expectation.

- (*Fresh-samples-based.*) In the reduction to OCRS by Feldman et al. [31], they draw a fresh sample $v'_{-i} \sim \mathcal{D}_{-i}$ for each arriving item i. The threshold T_i is set to $\tau_i^{v'}(\text{OPT}^{v'})$, where $\text{OPT}^{v'} = \text{OPT}^{v'}([n], \emptyset)$ is the maximum-weight independent set with respect to v'. Afterwards, the algorithm performs online contention resolution among those active items.

  This approach is later used in prophet inequalities for k-uniform matroids [20, 39] and bipartite matching [28]. By definition, each threshold in $\{T_i\}_{i \in [n]}$ requires one sample of the entire valuation vector v. In total, the reduction needs $\Omega(n)$ samples.

- (*Single-sample-based.*) Naturally, one can try to instantiate the same fresh-samples-based thresholds with only a single sample v' shared among all items. This idea was explored by Dughmi [21] in matroid secretary, which is a slightly different setting.

  The problem is that OCRS requires the events E_i that item i is active to be independent for different $i \in [n]$. Using a single sample introduces positive correlations between these events. Dughmi [21] shows that such positive correlated items still admit a constant-selectable CRS, but a constant-selectable OCRS for these items will imply the resolution of matroid secretary conjecture.

Our Approach. (Quantile-based.) Historically, for single-item prophet inequalities, the seminal result by Samuel-Cahn [54] shows that the median of the maximum value distribution gives the optimal 2-approximation. Such quantile-based thresholds are extended to the k-uniform matroids by Chawla et al. [10]. Naturally, one might wonder: Is such a choice still applicable to general matroids? To the best of our knowledge, this question still remains unexplored (no matter whether you allow contention resolution at the end or not).

In the sample-based setting, because previous thresholds break down, we are forced to understand quantile-based thresholds. Recall that $\text{OPT}^v = \text{OPT}^v([n], \emptyset)$ is the maximum-weight independent set with respect to random valuation v. In this work, we essentially set the threshold T_i to be the (approximate) median of $\tau_i^v(\text{OPT}^v)$. (For simplicity, let us assume all T_i's are the exact

medians throughout this overview.) Then apply contention resolution. At first glance, this expression may look similar to the fresh-samples-based thresholds of Feldman et al. [31]. However, note that they are using *the realization* of a fresh sample v' for each threshold T_i. But here we estimate *the medians* used to set all T_i's using the same set of $O_\varepsilon(\log n)$ samples.

Main Difficulty. To show that such thresholds, together with OCRS, provide a constant approximation, one needs to lower bound the total weight of active elements. That is, we wish to show that the contribution from nonactive items, $\sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i < T_i]$, is at most that from active items $\sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i \geq T_i]$. This turns out to be a non-trivial task.

For example, let us examine the case of k-uniform matroids. In this case, $\tau_i^v(\mathrm{OPT}^v)$ is always equal to the k-th largest value among all the items, denoted by $v[k]$. Let T' be the median of this value $v[k]$. We first set a deterministic benchmark, $k \cdot T'$:

- With probability $1/2$, we know that $v[k] \geq T'$ and that the contribution of active elements $\sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i \geq T_i]$ is at least $k \cdot T'$ in this case. There is no contribution from nonactive items.
- With the other $1/2$ probability, we know that $v[k] < T'$ and the contribution of nonactive items $\sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i < T_i]$ can be at most $k \cdot T'$ in this case.

Thus, we know that $\sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i \geq T_i] \geq k \cdot T'/2 \geq \sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i < T_i]$. We note that it is crucial that we are using a global argument: It first "gathers" the contribution of those elements i with $v_i \geq T'$, comparing it against a benchmark $k \cdot T'$. Then it "gathers" all the contribution of those elements with $v_i < T'$ and upper bounds it with the benchmark.

We note that such a global argument is necessary in a certain sense. Consider the following example: We add a new element that is deterministically $T' - \varepsilon$ to the k-uniform matroid. It is always below the threshold T', but is in the optimal solution with probability $1/2$. This results in a non-negligible contribution of $(T' - \varepsilon)/2$. Since it is never active, we are forced to "gather" the contribution of (possibly many) other active elements to compare with it.

Now, generalizing this argument to an arbitrary matroid, it is a priori not clear what the deterministic benchmark should be. We show that the weight of the optimal independent set w.r.t. weights $\{T_i\}_{i \in [n]}$, i.e. OPT^T is a good benchmark. To prove that $\sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i \geq T_i] \geq \mathrm{OPT}^T/2 \geq \sum_{i \in [n]} v_i \cdot \mathbb{1}[v_i < T_i]$, we need a weighted generalization of the strong basis exchange property (Lemma 3.8), which was used in submodular maximization [8] and has not been applied to prophet inequalities before.

Finally, to avoid losing competitive ratio in this reduction, we slightly generalize this approach and set multiple thresholds using different quantiles.

3.2 Reduction from Matroid Prophet Inequality to OCRS with Samples

Recall that we use $\mathcal{M} = (U, \mathcal{I})$ to denote a matroid (where $U = [n]$), and for each item $i \in U$, its value v_i is drawn independently from $\mathcal{D}_i$. For simplicity, we assume that all $\mathcal{D}_i$'s are continuous distributions. For general distributions with point masses, a standard tie-breaking technique can be applied. For completeness, we include it in Appendix A.1.

We first give some useful notations that we will use in this subsection. Given the realization of values v, we define $\mathrm{OPT}(v) \stackrel{\mathrm{def}}{=} \max_{S \in \mathcal{I}} \sum_{j \in S} v_j$ as the value of the maximum independent set and $\mathrm{OPT}_i(v_{\text{-}i}) \stackrel{\mathrm{def}}{=} \max_{S \in \mathcal{I}, i \notin S} \sum_{j \in S} v_j$ as the optimal value when item i is removed. We assume w.l.o.g. that every $\mathrm{OPT}_i(v_{\text{-}i})$ is a basis. We can always add dummy elements with zero value to ensure it. For simplicity of notation, we omit v when there is no obfuscation and abuse $\mathrm{OPT}, \mathrm{OPT}_i$ to denote both optimal sets of items and their total values.

Given $\boldsymbol{v}_{-i}$, define τ_i to be the smallest value among the items in OPT_i that i can exchange with. That is to say, $\tau_i(\boldsymbol{v}_{-i}) \overset{\text{def}}{=} \min\{v_j \mid \text{OPT}_i - j + i \in \mathcal{I}\}$. Again, we omit $\boldsymbol{v}_{-i}$ when it is clear from the context.

The following observations regarding τ_i are useful in our later analysis.

Observation 3.2. *Fixing $\boldsymbol{v}_{-i}$, if $v_i > \tau_i(\boldsymbol{v}_{-i})$, $i \in \text{OPT}(\boldsymbol{v})$; if $v_i < \tau_i(\boldsymbol{v}_{-i})$, $i \notin \text{OPT}(\boldsymbol{v})$.*[3]

PROOF. Let j be the item whose value equals τ_i. In the case when $v_i > \tau_i$, consider exchanging i with j in OPT_i. This results in a basis of value $\text{OPT}_i - \tau_i + v_i > \text{OPT}_i$. Hence OPT_i is not optimal, and consequently $i \in \text{OPT}$.

For the other direction, we prove by contradiction. Suppose $i \in \text{OPT}$, by Lemma A.1, there is an element $j \in \text{OPT}_i$ such that $\text{OPT} - i + j$ and $\text{OPT}_i - j + i$ are both bases. Since $\text{OPT} - i + j \in \mathcal{I}$, by optimality of OPT, $v_j \leq v_i$. Moreover, since $\text{OPT}_i - j + i \in \mathcal{I}$, by the definition of τ_i, we have $\tau_i \leq v_j$. Hence $\tau_i > v_j$ implies $i \notin \text{OPT}$. □

Observation 3.3. *Suppose $i \in \text{OPT}$, $j \notin \text{OPT}$ and $\text{OPT} - i + j \in \mathcal{I}$. Then $v_i \geq \tau_j$.*

PROOF. Since $j \notin \text{OPT}$, we have $\text{OPT}_j = \text{OPT}$. Together with the fact that $\text{OPT} - i + j$ is also an independent set, we have $\tau_j \leq v_i$ (by the definition of τ_j). □

Our algorithm consists of two stages. In the first stage, we estimate the thresholds $\{T_i^{(k)}\}$ from $O(\log n)$ samples. In the second stage, we use an OCRS to select a subset of active elements, which is sampled by an activation rule defined with thresholds. We present the first stage in the rest of this section. Let $m = \lfloor \log_{1+\varepsilon}(1/\varepsilon) \rfloor$.

Learning Thresholds. Given N i.i.d. samples $\boldsymbol{v}^s \sim \times_i \mathcal{D}_i$, where $s \in [N]$. For each item i, let $T_i^{(k)}$ be the $\lceil \varepsilon(1+\varepsilon)^k N \rceil$-th smallest value in $\{\tau_i(\boldsymbol{v}_{-i}^s) \mid s \in [N]\}$ for $0 \leq k < m$ and $T_i^{(m)} = \infty$.

The next lemma states that the thresholds $\{T_i^{(k)}\}_{i \in U}$ approximate the functions $p^i(v) \overset{\text{def}}{=} \mathbf{Pr}[i \in \text{OPT} \mid v_i = v]$ within ε error. The proof is a direct application of Chernoff bound and we defer it to Appendix A.2. Let $p_k = \varepsilon(1+\varepsilon)^k - \varepsilon^2$ for $0 \leq k < m$. Note by Observation 3.2, we have $p^i(v) = \mathbf{Pr}[v > \tau_i]$ for any fixed v, and $\{T_i^{(k)}\}$ are fixed values (instead of random variables) here.

Lemma 3.4. *For any $\varepsilon \in (0,1)$, with $N = O(\log(\frac{2nm}{\varepsilon}) \cdot \varepsilon^{-4})$ samples, $p^i(T_i^{(k)}) = \mathbf{Pr}[T_i^{(k)} > \tau_i] \in [p_k, p_k + 2\varepsilon^2]$ for all $i \in U$ and $0 \leq k < m$ with probability at least $1 - \varepsilon$. We refer to such thresholds as good thresholds.*

Once we have estimated these thresholds, we view them as fixed value rather than random variables (i.e. conditioning on the value of these thresholds). This is important for the independence required by OCRSs.

Activation Rule. With these thresholds, we specify a rule of determining whether an item i is active. If $v_i \in [T_i^{(k)}, T_i^{(k+1)})$ for some $0 \leq k < m$, let i be active with probability p_k; else $v_i \in [0, T_i^{(0)})$, let i be active with probability 0. Let $x_i = \mathbf{Pr}[i \text{ is active}]$, where the probability space only involves randomness of $v_i \sim \mathcal{D}_i$.

For any $\boldsymbol{v} \sim \times_i \mathcal{D}_i$, we can then apply this activation rule to each of the items to generate a set of active items. It is important that in this stage, we do not reuse the samples from the first stage. The reason is that there are correlations between the thresholds we learnt for different items, but OCRS requires each item to be active independently. Thus we treat the thresholds now as fixed values and use the real realization of $\boldsymbol{v}$ to generate $R(\boldsymbol{x})$. Observe that, in this way, each element is active independently.

---

[3]We ignore the equality case which happens with zero probability.

Lemma 3.5. *If $\{T_i^{(k)}\}$ are good thresholds, then $\boldsymbol{x} \in \mathcal{P}_M$.*

PROOF. Let $x_i^* = \mathbf{Pr}[i \in \text{OPT}]$ for every $i \in U$. Since $\text{OPT} \in \mathcal{I}$, we have $\boldsymbol{x}^* \in \mathcal{P}_M$. Consider the conditional probability of i being active when $v_i = v$.

- If $v < T_i^{(0)}$, $\mathbf{Pr}[i \text{ is active} \mid v_i = v] = 0 \leq p^i(v)$.
- If $v \in [T_i^{(k)}, T_i^{(k+1)})$ for $0 \leq k < m$, we have

$$\mathbf{Pr}\left[i \text{ is active} \mid v_i = v\right] = p_k \leq p^i(T_i^{(k)}) \leq p^i(v),$$

  where the first inequality follows from the definition of $T_i^{(k)}$ and the second inequality follows from the monotonicity of the function p^i.

Therefore, for each $i \in U$,

$$x_i = \int_v \mathbf{Pr}\left[i \text{ is active} \mid v_i = v\right] \mathrm{d}\mathcal{D}_i(v) \leq \int_v p^i(v) \mathrm{d}\mathcal{D}_i(v) = x_i^*,$$

that concludes the proof of the lemma. $\square$

Furthermore, the expected values of active items are large compared to the expected optimum. This is the most technical lemma and the proof is built on a monotone basis exchange lemma (Lemma 3.8) for matroids from [8].

Lemma 3.6. *If $\{T_i^{(k)}\}$ are good thresholds, then $\mathbf{E}[\sum_{i \in R(\boldsymbol{x})} v_i] \geq (1 - O(\varepsilon)) \cdot \mathbf{E}[\text{OPT}]$.*

PROOF. We defer its proof to Section 3.3. $\square$

Now, we have all the components to prove our main theorem.

Theorem 3.1. *With $O_\varepsilon(\log^4 n)$ samples, there is a $(\frac{1}{4} - \varepsilon)$-competitive prophet inequality for general matroids of size n and any $\varepsilon > 0$.*

PROOF. We first use $O_\varepsilon(\log n)$ samples to learn the thresholds $\{T_i^{(k)}\}$. By Lemma 3.4 and Lemma 3.5, the thresholds are good with at least $(1-\varepsilon)$ probability and the corresponding activation rule results in a valid $\boldsymbol{x} \in \mathcal{P}_M$. We then use $O_\varepsilon(\log^4 n)$ samples to learn the corresponding $(\frac{1}{4} - \varepsilon)$-selectable OCRS by Theorem 4.1. Finally, in the real run of the prophet inequality, upon the arrival of an item i, we use the activation rule constructed above to decide whether i active and feed it to the OCRS.

Let ALG be the random set of elements selected by the algorithm. Let $\mathcal{E}$ be the event that $\{T_i^{(k)}\}$ are good thresholds.

$$
\begin{aligned}
\mathbf{E}\left[\sum_{i \in \text{ALG}} v_i\right] &\geq \mathbf{Pr}\left[\mathcal{E}\right] \cdot \mathbf{E}\left[\sum_{i \in \text{ALG}} v_i \,\middle|\, \mathcal{E}\right] \\
&\geq (1 - \varepsilon) \cdot \left(\frac{1}{4} - \varepsilon\right) \mathbf{E}\left[\sum_{i \in R(\boldsymbol{x})} v_i \,\middle|\, \mathcal{E}\right] && \text{(Lemma 3.4 and Theorem 4.1)} \\
&\geq (1 - \varepsilon) \cdot \left(\frac{1}{4} - \varepsilon\right) \cdot (1 - O(\varepsilon)) \cdot \mathbf{E}\left[\text{OPT}\right] && \text{(Lemma 3.6)} \\
&= \left(\frac{1}{4} - O(\varepsilon)\right) \cdot \mathbf{E}\left[\text{OPT}\right],
\end{aligned}
$$

that concludes the proof of the theorem. $\square$

Remark 3.7 (Against almighty adversary). *Although here we state the prophet inequality problem with offline adversary, i.e., assuming the items arrive in a fixed order $1, 2, \ldots, n$, our sample-based algorithm actually works against stronger almighty adversary, who determines the arrival order adaptively with full knowledge of all realizations of randomness and the decisions made by the algorithm. The reason is that the thresholds are computed in a non-adaptive way (by using samples only), and the sample-based OCRS is a greedy OCRS, which works against almighty adversary (see [31] for details).*

3.3 Proof of Lemma 3.6

Observe that we select items in a conservative way, in the sense that $\mathbf{Pr}[i \text{ is active} \mid v_i = v] \leq \mathbf{Pr}[i \in \mathrm{OPT} \mid v_i = v]$ for all i, v. Nevertheless, we prove that the expected values of the active items are ε-close to the optimum.

We will need the following monotone basis exchange lemma for matroids. For completeness, we include a proof of it in Appendix A.3.

Lemma 3.8 (Lemma 2.4, [8]). *Let $\mathcal{M} = (U, \mathcal{I})$ be a weighted matroid with weight function $w : U \to \mathbb{R}_{\geq 0}$. Suppose $A \in \mathcal{I}$ is the basis with maximum total weight. For every basis B, there exists a bijection $f : A \to B$ such that for all $x \in A$, $B - f(x) + x$ is a basis and $w(f(x)) \leq w(x)$. Moreover, the bijection f satisfies $f(x) = x$ for all $x \in A \cap B$.*

It is worthwhile to make a comparison to the following lemma in the literature.

Lemma 3.9 (Lemma 1, [42]). *Let $\mathcal{M} = (U, \mathcal{I})$ be a weighted matroid with weight function $w : U \to \mathbb{R}_{\geq 0}$. Suppose $A \in \mathcal{I}$ is the basis with maximum total weight. For every basis B, there exists a bijection $g : B \to A$ such that for all $y \in B$, $A - g(y) + y$ is a basis and $w(g(y)) \geq w(y)$.*

Observe that the only difference between the two lemmas is the direction of the mappings f, g. We note that for every bijection g such that $A - g(y) + y$ is a basis, we must have $w(g(y)) \geq w(y)$ as otherwise $w(A - g(y) + y) > w(A)$. In contrast, Lemma 3.8 does not hold for every bijection f such that $B - f(x) + x$ is a basis.

Now, we are ready to prove Lemma 3.6. The crucial part is to bound the contribution of those items with $v_i < T_i^{(0)}$.

Lemma 3.10. *If $\{T_i^{(k)}\}$ are good thresholds,*

$$\mathbf{E}\left[\sum_{i \in \mathrm{OPT}} v_i \cdot \mathbb{1}\left[v_i < T_i^{(0)}\right] \right] \leq O(\varepsilon) \cdot \mathbf{E}\left[\mathrm{OPT}\right].$$

PROOF. Let $W = \mathrm{argmax}_{S \in \mathcal{I}} \sum_{i \in S} T_i^{(0)}$ be the maximum weighted basis where item i has value $T_i^{(0)}$. Fix arbitrary $\boldsymbol{v}$ and $\mathrm{OPT}(\boldsymbol{v})$, by Lemma 3.8, there exists a bijection $f : W \to \mathrm{OPT}$ such that for any $j \in W$, $\mathrm{OPT} - f(j) + j \in \mathcal{I}$ and $T_{f(j)}^{(0)} \leq T_j^{(0)}$. Moreover, for all $j \in W \cap \mathrm{OPT}$, $f(j) = j$.

By Observation 3.2, for every $j \in W \cap \mathrm{OPT}$, we have $f(j) = j$ and $v_j \geq \tau_j(\boldsymbol{v}_{-j})$. For every $j \in W \setminus \mathrm{OPT}$, by Observation 3.3, we also have $v_{f(j)} \geq \tau_j(\boldsymbol{v}_{-j})$. Hence, $v_{f(j)} \geq \tau_j$ for all $j \in U$. Consequently,

$$\mathbf{Pr}\left[v_{f(j)} \geq T_j^{(0)}\right] \geq \mathbf{Pr}\left[\tau_j(\boldsymbol{v}_{-j}) \geq T_j^{(0)}\right] \geq 1 - \varepsilon - \varepsilon^2, \tag{1}$$

where the last inequality holds when $\{T_i^{(k)}\}$ are good. Then,

$$\mathbf{E}\left[\sum_{i \in \mathrm{OPT}} v_i\right] = \mathbf{E}\left[\sum_{j \in W} v_{f(j)}\right] \geq \mathbf{E}\left[\sum_{j \in W} T_j^{(0)} \cdot \mathbb{1}\left[v_{f(j)} \geq T_j^{(0)}\right]\right] \geq (1 - \varepsilon - \varepsilon^2)\sum_{j \in W} T_j^{(0)}. \tag{2}$$

Recall that by the construction of f, we have that $T_{f(j)}^{(0)} \le T_j^{(0)}$, $\forall j \in W$. Equivalently, we have $T_{f^{-1}(i)}^{(0)} \ge T_i^{(0)}$, $\forall i \in \text{OPT}$. Finally,

$$
\mathbf{E}\left[\sum_{i \in \text{OPT}} v_i \mathbb{1}\left[v_i < T_i^{(0)}\right]\right] \le \mathbf{E}\left[\sum_{i \in \text{OPT}} v_i \mathbb{1}\left[v_i < T_{f^{-1}(i)}^{(0)}\right]\right] \le \mathbf{E}\left[\sum_{i \in \text{OPT}} T_{f^{-1}(i)}^{(0)} \mathbb{1}\left[v_i < T_{f^{-1}(i)}^{(0)}\right]\right]
$$

$$
= \mathbf{E}\left[\sum_{j \in W} T_j^{(0)} \mathbb{1}\left[v_{f(j)} < T_j^{(0)}\right]\right] \overset{(1)}{\le} (\varepsilon + \varepsilon^2) \sum_{j \in W} T_j^{(0)}
$$

$$
\overset{(2)}{\le} \frac{\varepsilon + \varepsilon^2}{1 - \varepsilon - \varepsilon^2} \mathbf{E}\left[\text{OPT}\right] \le O(\varepsilon) \cdot \mathbf{E}\left[\text{OPT}\right]. \qquad \square
$$

PROOF OF LEMMA 3.6. For every $i \in U$ and $v \in [T_i^{(k)}, T_i^{(k+1)})$, we have

$$
p^i(v) \le p^i(T_i^{(k+1)}) \le \varepsilon(1 + \varepsilon)^{k+1} + \varepsilon^2 \qquad (\{T_i^{(k)}\} \text{ are good})
$$

$$
\le (1 + O(\varepsilon)) \cdot (\varepsilon(1 + \varepsilon)^k - \varepsilon^2)
$$

$$
= (1 + O(\varepsilon)) \mathbf{Pr}\left[i \in R(\mathbf{x}) \mid v_i = v\right]. \qquad (\text{activation rule})
$$

Therefore,

$$
\mathbf{E}\left[\sum_{i \in \text{OPT}} v_i \mathbb{1}\left[v_i \ge T_i^{(0)}\right]\right] = \sum_{i \in U} \int_{T_i^{(0)}}^{\infty} v p^i(v) \mathrm{d}F_i(v)
$$

$$
\le (1 + O(\varepsilon)) \sum_{i \in [n]} \int_{T_i^{(0)}}^{\infty} v \mathbf{Pr}\left[i \in R(\mathbf{x}) \mid v_i = v\right] \mathrm{d}F_i(v)
$$

$$
= (1 + O(\varepsilon)) \mathbf{E}\left[\sum_{i \in R(\mathbf{x})} v_i\right].
$$

Combining this inequality with Lemma 3.10, we conclude that

$$
\mathbf{E}\left[\sum_{i \in R(\mathbf{x})} v_i\right] \ge \frac{1}{1 + O(\varepsilon)} \mathbf{E}\left[\sum_{i \in \text{OPT}} v_i \mathbb{1}\left[v_i \ge T_i^{(0)}\right]\right]
$$

$$
= \frac{1}{1 + O(\varepsilon)}\left(\mathbf{E}\left[\text{OPT}\right] - \mathbf{E}\left[\sum_{i \in \text{OPT}} v_i \mathbb{1}\left[v_i < T_i^{(0)}\right]\right]\right) \ge \frac{1}{1 + O(\varepsilon)} \mathbf{E}\left[\text{OPT}\right]. \qquad \square
$$

4 Matroid OCRS from Samples

Motivated by our sample-based reduction, our goal now is to design a constant selectable matroid OCRS using polylogarithmic samples. A previous work [32] studied lower bounds for this problem. Upon careful inspection, their counterexample shows that any constant selectable matroid OCRS requires $\Omega(\log n)$ samples. (Details are given in Appendix C.) In this work, we design a matroid OCRS with $O_\varepsilon(\log^4 n)$ samples:

Theorem 4.1. *With $O_\varepsilon(\log^4 n)$ samples, there is a $(\frac{1}{4} - \varepsilon)$-selectable OCRS for general matroids of size n and any $\varepsilon > 0$.*

Our OCRS is based on the matroid OCRS proposed by Feldman et al. [31]. Recall that in online contention resolution, each element $e \in N$ is active independently with probability x_e. Let $R(\mathbf{x})$ be the random set of active elements. For sample-based schemes, this vector $\mathbf{x}$ is hidden, instead the

algorithm gets s samples of $R(\boldsymbol{x})$. To implement the algorithm in [31], it suffices to implement an oracle that answers queries of form $\mathbf{Pr}[e \in \mathrm{span}(R(\boldsymbol{x}) \cup S)]$ for element $e \in N$ and subset $S \subseteq N$ with error up to ε.

For a single fixed query, evaluating the empirical probability over $\frac{\log(n)}{\varepsilon^2}$ samples guarantees the success with high probability. At first sight, it is tempting to argue that these samples suffice by a union bound over all $\mathrm{poly}(n)$ queries made by their algorithm. However, such argument is only true for non-adaptive queries. For adaptive queries, the error in the answers to previous queries can reveal information about the samples that leads to a problematic query. Resolving this issue is a central problem in adaptive data analysis. As the OCRS by Feldman et al. [31] dynamically updates the set S based on previous queries, it makes $\mathrm{poly}(n)$ highly adaptive queries. Studies in adaptive data analysis show that, in general, answering these many adaptive queries up to $\varepsilon = 0.1$ error already requires $\mathrm{poly}(n)$ samples [36, 57]. In our work, we protect against such adaptivity with new ideas that are specific to OCRSs.

4.1 Recap: Chain Decomposition

In this section, we briefly sketch the idea of the OCRS by Feldman et al. [31]. We are given a matroid $\mathcal{M} = (U, \mathcal{I})$, where $U = [n]$ is the universe, and $\mathcal{I}$ is the family of independent sets. Suppose $\boldsymbol{x} \in b \cdot \mathcal{P}_{\mathcal{M}}$ is a vector within the matroid polytope shrank by a factor of b, where b is a constant factor that we will pick later. Each element $e \in U$ is active independently with probability x_e. Let $R(\boldsymbol{x})$ be the random set of active elements (w.r.t. the vector $\boldsymbol{x}$).

Protected elements. Consider the greedy algorithm that always takes an active element whenever possible. Its selectability for an element $e \in U$ can be lower-bounded:

$$\mathbf{Pr}\left[e \text{ is accepted by greedy} \mid e \text{ is active}\right] \geq 1 - \mathbf{Pr}[e \in \mathrm{span}(R(\boldsymbol{x}) \setminus \{e\})].$$

The idea of chain decomposition is that we are going to protect those elements $e \in U$ with a selectability less than $1 - c$. We prioritize those elements by adding them to a set of protected elements S. Now the modified greedy algorithm is going to pretend that all elements in S are already taken at the beginning, then take any active element whenever possible. As the set S is now nonempty, the selectability for an element $e \in U \setminus S$ becomes

$$\mathbf{Pr}\left[e \text{ is accepted by modified greedy} \mid e \text{ is active}\right] \geq 1 - \mathbf{Pr}[e \in \mathrm{span}(R(\boldsymbol{x}) \cup S \setminus \{e\})].$$

There may be new problematic elements with this probability less than $1 - c$. We then have to enlarge the set S by adding those elements. In this way, the set S is *dynamically* changing based on the result of the previous queries of such a probability. Then new queries depend on the changing S, which makes them highly *adaptive* in nature. This process is summarized in Algorithm 1. It takes a submatroid $(N, \mathcal{I}|_N)$ for $N \subseteq U$ and returns the protected set S for that submatroid. We note that although the choice of the element e at Line 3 is arbitrary, the returned set S is always unique. (See Appendix B.1.)

ALGORITHM 1: Select protected elements for N

---

1 **Function** Select(N, c)
2      $S \leftarrow \emptyset$
3   **while** $\exists e \in N \setminus S$, $\mathbf{Pr}\left[e \in \mathrm{span}(((R(\boldsymbol{x}) \cap N) \cup S) \setminus \{e\})\right] > c$ **do**
4   add such element e to S.
5   **end**
6   **return** S

Chain decomposition. After selecting the protected set S, note that the elements outside S can be handled by the modified greedy algorithm and achieve $(1 - c)$-selectability. Furthermore, the following lemma guarantees that set S returned by $\text{Select}(N, c)$ is always of a smaller rank than N when $b \leq c$.

Lemma 4.2 (Section 2.1.1, [31]). *For any submatroid $(N, \mathcal{I}|_N)$, vector $\boldsymbol{x} \in b \cdot \mathcal{P}_M$, and $c > 0$, we always have*

$$\text{rank}(S) < \frac{b}{c} \cdot \text{rank}(N) \; \text{where } S = \text{Select}(N, c).$$

PROOF. We include a proof in Appendix B.2 for completeness. $\square$

Therefore, as long as we pick $c \geq b$, the subproblem in S is always strictly smaller. We can recursively apply the idea to the submatroid $(S, \mathcal{I}|_S)$. This leads to the following algorithm:

ALGORITHM 2: Chain decomposition in Feldman et al. [31]

1 **Procedure** Decompose(U, c)
2 $\quad$ $N_0 \leftarrow U$
3 $\quad$ $\ell \leftarrow 0$
4 $\quad$ **while** $N_\ell \neq \emptyset$ **do**
5 $\quad\quad$ $N_{\ell+1} \leftarrow \text{Select}(N_l, c)$
6 $\quad\quad$ $\ell \leftarrow \ell + 1$
7 $\quad$ **end**

By Lemma 4.2, when $b < c - \varepsilon$, it is guaranteed to terminate in $\ell = O_\varepsilon(\log n)$ steps and produce the following chain decomposition

$$\emptyset = N_\ell \subsetneq N_{\ell-1} \subsetneq \cdots \subsetneq N_1 \subsetneq N_0 = U \text{ where } N_{i+1} = \text{Select}(N_i, c) \text{ for } 0 \leq i < \ell.$$

The final OCRS in [31] is simply running Algorithm 3, a modified greedy algorithm, for each layer $N_i \setminus N_{i+1}$ separately.

ALGORITHM 3: Modified greedy algorithm for $N_i \setminus N_{i+1}$

1 $A \leftarrow \emptyset$; $\qquad\qquad\qquad\qquad$ // Here A is the set of accepted elements.
2 **for** *each arriving element $e \in N_i \setminus N_{i+1}$* **do**
3 $\quad$ **if** $A \cup N_{i+1} \in \mathcal{I}$ **then**
4 $\quad\quad$ $A \leftarrow A \cup \{e\}$
5 $\quad$ **end**
6 **end**
7 **return** A

In other words, Algorithm 3 is equivalent to a greedy algorithm running on the matroid $\mathcal{M}|_{N_i}/N_{i+1}$, which is $\mathcal{M}$ restricted to N_i and then contracted by N_{i+1}. It is easy to see that running Algorithm 3 for every $0 \leq i < \ell$ together always produces an independent set of $\mathcal{M}$. And the selectability of any element $e \in N_i \setminus N_{i+1}$ is

$$\mathbf{Pr}\left[e \text{ is accepted} \mid e \text{ is active}\right] \geq 1 - \mathbf{Pr}\left[e \in \text{span}(((R(\boldsymbol{x}) \cap N_i) \cup N_{i+1})) \setminus \{e\})\right] \geq 1 - c.$$

4.2 Selecting Protected Elements with Samples

Now, we move on to introduce our sampled-based OCRS. As a natural first step, let us first try to implement Algorithm 1 with samples:

ALGORITHM 4: Select protected elements for N, with samples

---

1 **Function** $\widehat{\text{Select}}(N, c)$
2   Let $R_1, R_2, \ldots, R_s$ be s *fresh* realized samples from $R(\boldsymbol{x})$
3   For any event $E(R)$, define $\widehat{\mathbf{Pr}}[E] := \frac{1}{s} \sum_{i=1}^{s} \mathbb{1}[E(R_i)]$
4   $\widehat{S} \leftarrow \emptyset$
5   **while** $\exists e \in N \setminus S, \widehat{\mathbf{Pr}}[e \in \text{span}(((R \cap N) \cup \widehat{S}) \setminus \{e\})] > c$ **do**
6   add such element e to $\widehat{S}$
7   **end**
8   **return** $\widehat{S}$

Suppose that the algorithm returns a final set $\widehat{S}$. In order for the chain decomposition approach to work, we need the following property: For some constant c' strictly less than 1,

$$\mathbf{Pr}_{R(\boldsymbol{x})}\left[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup \widehat{S}) \setminus \{e\})\right] \leq c', \quad \forall e \in N \setminus \widehat{S}. \tag{3}$$

It is tempting to think that if we take $s = \frac{\log n}{\delta^2}$ samples, by concentration, we will be able to use $\widehat{\mathbf{Pr}}[E]$ to answer polynomially many $\mathbf{Pr}[E]$ queries up to δ error with high probability, which ensures this property with $c' = c - \delta$. This is only true for non-adaptive queries. Here, in Algorithm 4, it is crucial that the algorithm maintains a dynamic set $\widehat{S}$ and adaptively discovers new elements that need to be protected. The issue with adaptive queries is that the error in the previous answers reveals information about the samples.

Looking closer, Algorithm 4 compares the answer to those $\mathbf{Pr}[E]$ queries with a threshold c. Consider a single $\mathbf{Pr}[E]$ query for estimating $\mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup \widehat{S}) \setminus \{e\})]$. If the true answer is less than $c - \delta$ (or greater than $c + \delta$), with high probability, the estimate returned by $\widehat{\mathbf{Pr}}$ will also be below (or above) the threshold c. Adaptivity appears when the true answer lies within $[c - \delta, c + \delta]$, whether the element e is added to $\widehat{S}$ or not depends now on the error on the samples. It affects the set $\widehat{S}$, which determines the later queries.

Remark 4.3 (Adaptive Data Analysis). *$\widehat{\mathbf{Pr}}[E]$ asks for the fraction of samples that satisfies an event E. These are called statistical queries. Answering statistical queries against adaptivity is a central problem in the field of adaptive data analysis. Lower bounds from there show, in general, with s samples, one can answer at most s^2 many such adaptive queries [36, 57]. There are a few exceptions, e.g. when the support of the sample is polynomially bounded [7], or when all queries are threshold queries, and they are sparse [24, 35] (in the that sense that, only a bounded number of queries have above-threshold answers, which can be solved using the sparse vector technique). However, in our case, the support of $R(\boldsymbol{x})$ can be as large as 2^n. Although the queries are threshold queries, they are not sparse. As these existing tools seem insufficient, novel ideas are needed to resolve the adaptivity issue in Algorithm 4.*

To handle the adaptivity issue, let us observe a surprising property of $\widehat{\text{Select}}(N, c)$: despite the presence of adaptivity, $\widehat{\text{Select}}(N, c)$ is still sandwiched between two sets that satisfy Property (3).

Lemma 4.4. *Fix any set $N \subseteq U$ and constants $0 < c_1 < c_2 < 1$. We define $\widehat{c} = \frac{c_1 + c_2}{2}$ and $\delta = \frac{c_2 - c_1}{2}$. Using $O\left(\frac{\log n \log \varepsilon^{-1}}{\delta^2}\right)$ samples,*

$$\text{Select}(N, c_2) \subseteq \widehat{\text{Select}}(N, \widehat{c}) \subseteq \text{Select}(N, c_1)$$

holds with probability at least $1 - \frac{\text{poly}(\varepsilon)}{\text{poly}(n)}$.

Proof. We prove these two cases separately:

- $\widehat{\text{Select}}(N, \widehat{c}) \subseteq \text{Select}(N, c_1)$.

  Let $S_1 = \text{Select}(N, c_1)$. Consider the process of $\widehat{\text{Select}}(N, \widehat{c})$ which dynamically maintains the set $\widehat{S}$. We claim that $\widehat{S} \subseteq S_1$ throughout the process. We prove this via induction. Initially, we have $\widehat{S} = \emptyset \subseteq S_1$. Then at any step, if we are going to add e to $\widehat{S}$, we must have

  $$\widehat{\mathbf{Pr}}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup \widehat{S}) \setminus \{e\})] > \widehat{c}.$$

  By induction hypothesis, we know that $\widehat{S} \subseteq S_1$, which implies $(R(\boldsymbol{x}) \cap N) \cup \widehat{S} \subseteq (R(\boldsymbol{x}) \cap N) \cup S_1$. Hence, by monotonicity,

  $$\widehat{\mathbf{Pr}}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_1) \setminus \{e\})]] > \widehat{c}.$$

  Note $S_1 = \text{Select}(N, c_1)$ is a fixed set that do not depend on samples. As we take $O\left(\frac{\log n \log \varepsilon^{-1}}{\delta^2}\right)$ samples, by Chernoff bound, with probability at least $1 - \frac{\text{poly}(\varepsilon)}{\text{poly}(n)}$, we have

  $$\left|\widehat{\mathbf{Pr}}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_1) \setminus \{e\})]] - \mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_1) \setminus \{e\})]\right| \leq \delta \quad (4)$$

  A union bound over all n possibilities of e shows that with probability $1 - \frac{\text{poly}(\varepsilon)}{\text{poly}(n)}$, Equation (4) holds throughout the process. Hence, for the newly added element e, we have

  $$\mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_1) \setminus \{e\})] > \widehat{c} - \delta \geq c_1.$$

  Since all elements in $U \setminus S_1$ are $(1 - c_1)$-selectable, we know that e must be in S_1.

- $\text{Select}(N, c_2) \subseteq \widehat{\text{Select}}(N, \widehat{c})$.

  First of all, let $\widehat{S} = \widehat{\text{Select}}(N, \widehat{c})$. When Algorithm 4 terminates, we know that for all $e \in U \setminus \widehat{S}$,

  $$\widehat{\mathbf{Pr}}[e \in \text{span}(((R \cap N) \cup \widehat{S}) \setminus \{e\})] \leq c.$$

  Consider the process of $\text{Select}(N, c_2)$, which dynamically maintains a set S_2. We claim that with high probability over the randomness of $\widehat{S}$, it holds that $S_2 \subseteq \widehat{S}$ through out this process. Again, we prove this by induction. Initially, we have $S_2 = \emptyset \subseteq \widehat{S}$. At any step, if we are going to add e to S_2, we must have

  $$\mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_2) \setminus \{e\})] > c_2.$$

  Note that the process of generating S_2 (i.e., the procedure $\text{Select}(N, c_2)$) does not depend on our samples. Similar to the previous case, by Chernoff bound, we get that with probability at least $1 - \frac{\text{poly}(\varepsilon)}{\text{poly}(n)}$,

  $$\widehat{\mathbf{Pr}}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_2) \setminus \{e\})] > c_2 - \delta > c_2 - \delta > \widehat{c}.$$

  By the induction hypothesis, we know that with $1 - \frac{\text{poly}(\varepsilon)}{\text{poly}(n)}$ probability, $S_2 \subseteq \widehat{S}$, which implies $(R(\boldsymbol{x}) \cap N) \cup S_2 \subseteq (R(\boldsymbol{x}) \cap N) \cup \widehat{S}$. Hence, with the same high probability,

  $$\widehat{\mathbf{Pr}}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup \widehat{S}) \setminus \{e\})] > \widehat{c}.$$

  If this happens, e must be in the set $\widehat{S}$. Finally, we finish the proof by observing that we use union bound over at most $\text{poly}(n)$ induction steps. □

According to Lemma 4.4, if we let $S_{c-\delta} = \text{Select}(N, c - \delta)$ and $S_{c+\delta} = \text{Select}(N, c + \delta)$, the set $\widehat{S}_c = \widehat{S}(N, c)$ must satisfy $S_{c+\delta} \subseteq \widehat{S}_c \subseteq S_{c-\delta}$ with high probability. Note that $S_{c-\delta}$ and $S_{c+\delta}$ are both valid choices for the set S that satisfy Equation (3) with different constants. Since $\widehat{S}_c$ is sandwiched between them, it is very tempting to think that the set $\widehat{S}_c$ must also satisfy Equation (3). If so, we would have proved that the original OCRS of Feldman et al. [31] (Algorithm 5) works with samples. However, this is not the case.

To see this, consider the quantity $r_e := \mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup \widehat{S}_c) \setminus \{e\})]$ for every element $e \in N \setminus \widehat{S}_c$:

- It cannot be the case that $e \in S_{c+\delta}$. If this is the case, we are guaranteed that $e \in \widehat{S}_c$ as well, which contradicts $e \in N \setminus \widehat{S}_c$.
- If $e \in N \setminus S_{c-\delta}$, we are guaranteed by Algorithm 1 that
$$\mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_{c-\delta}) \setminus \{e\})] \leq c - \delta.$$
Since $S_c \subseteq S_{c-\delta}$, by monotonicity, the quantity r_e that we care about is also at most $c - \delta$.
- If $e \in S_{c-\delta} \setminus S_{c+\delta}$, on the one hand, e is in $S_{c-\delta}$, so we do not have any non-trivial upper bound for
$$\mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_{c-\delta}) \setminus \{e\})].$$
On the other hand, $e \in N \setminus S_{c+\delta}$, although we know that
$$\mathbf{Pr}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup S_{c+\delta}) \setminus \{e\})] \leq c + \delta.$$
But $\widehat{S}_c$ is (potentially) a larger set than $S_{c+\delta}$. It could be the case that r_e is much larger than $c + \delta$. So we cannot hope to get a non-trivial upper bound for r_e.

In conclusion, the only problematic case is when $e \in S_{c-\delta} \setminus S_{c+\delta}$, where we do not have any guarantee of their selectability.

4.3 Against Adaptivity using Randomization

To illustrate our idea, we will first present a simple toy example to warm up. Then we will slightly generalize it to our actual construction.

Toy Example. Our main idea for resolving such an adaptivity issue is simple to state: by choosing the protected set $\widehat{S}$ to be either $\widehat{S}_c$ or $\widehat{S}_{c+2\delta}$ uniformly at random, i.e.

$$\widehat{S} = \begin{cases} \widehat{\text{Select}}(N, c) & \text{w.p. } \frac{1}{2}, \\ \widehat{\text{Select}}(N, c + 2\delta) & \text{w.p. } \frac{1}{2}. \end{cases}$$

By Lemma 4.4 and union bound, we know with high probability,

$$S_{c+3\delta} \subseteq \widehat{S}_{c+2\delta} \subseteq S_{c+\delta} \subseteq \widehat{S}_c \subseteq S_{c-\delta}.$$

We know if one chooses to protect $\widehat{S}_c$ (i.e., $\widehat{S} = \widehat{\text{Select}}(N, c)$), we have no guarantee for the elements in $S_{c-\delta} \setminus S_{c+\delta}$. Similarly, if one chooses to protect $\widehat{S}_{c+2\delta}$ (i.e., $\widehat{S} = \widehat{\text{Select}}(N, c + 2\delta)$), elements in $S_{c+\delta} \setminus S_{c+3\delta}$ are problematic. So neither of these two works. But the crucial observation is that $S_{c-\delta} \setminus S_{c+\delta}$ and $S_{c+\delta} \setminus S_{c+3\delta}$ are two disjoint sets. If we just randomize between these two, we can imagine having guarantees for all elements with only a factor-of-2 loss in selectability.

More specifically, for any element $e \in N$,

- with probability $1/2$, it is not in the problematic set of elements, which means with high probability, either $e \in \widehat{S}$ or $\mathbf{Pr}_{R(\boldsymbol{x})}[e \in \text{span}(((R(\boldsymbol{x}) \cap N) \cup \widehat{S}) \setminus \{e\})] \leq c + 3\delta$;
- with the other $1/2$ probability, it is in the problematic set and we do not have any guarantee.

Note that this only achieves Equation (3) with probability $1/2$. There is still a probability of $1/2$ that a bad event occurs: $e \in N \setminus \widehat{S}$ but $\mathbf{Pr}_{R(\boldsymbol{x})}[e \in \mathrm{span}(((R(\boldsymbol{x}) \cap N) \cup \widehat{S}) \setminus \{e\})]$ is too large.

This becomes a problem if we plug this procedure into the chain decomposition procedure (Algorithm 2). From Lemma 4.2, we know that the chain decomposition procedure has at most $\ell = O_\varepsilon(\log n)$ rounds. So it calls the procedure $\widehat{\text{Select}}$ ℓ times. Let us imagine the worst case: For every round in which e is not in the problematic set, let us say that e is in $\widehat{S}$ and the decomposition procedure continues with $e \in N$. Once we meet a round where e is in the problematic set (with probability $1/2$), we lose all guarantees for the element e. But it is highly likely (with probability $1 - (1/2)^{O_\varepsilon(\log n)}$) that we will see one such problematic round.

Our Construction. In order to fix this issue and achieve constant selectability, our final scheme needs a sequence $c_1 < c_2 < \cdots < c_{k+1}$ of $k+1$ different thresholds that are equally spaced in $[c_1, c_{k+1}]$. The scheme picks $j \in [k]$ uniformly at random and then uses $\widehat{c} = \frac{c_j + c_{j+1}}{2}$. Note that for each round i, we are using s fresh samples for $\widehat{\text{Select}}(\widehat{N}_i, \widehat{c})$. Finally, we are going to take $k = O_\varepsilon(\log n)$, so that the probability that the bad event happens, $1 - (1 - 1/k)^{O_\varepsilon(\log n)}$, is small enough.

ALGORITHM 5: Chain decomposition, with samples

---

1 **Procedure** $\widehat{\text{Decompose}}(U, c_1, c_2, \ldots, c_{k+1})$
2      $\widehat{N}_0 \leftarrow U$
3      $\ell \leftarrow 0$
4   **while** $\widehat{N}_\ell \neq \emptyset$ **do**
5   Sample $j_\ell \sim \mathrm{Uniform}([k])$ and let $\widehat{c} \leftarrow \frac{c_{j_\ell} + c_{j_\ell + 1}}{2}$
6          $\widehat{N}_{\ell+1} \leftarrow \widehat{\text{Select}}(\widehat{N}_\ell, \widehat{c})$
7          $\ell \leftarrow \ell + 1$
8   **end**

4.4 Proof of Theorem 4.1

In this section, we complete the proof of Theorem 4.1 using Algorithm 5. In our proof, we set $b = \frac{1}{2}$, which means that we shrink the ex-ante active probability $\boldsymbol{x}$ by half to ensure $\boldsymbol{x} \in \frac{1}{2} \cdot \mathcal{P}_{\mathcal{M}}$. Let

$$\varnothing = \widehat{N}_\ell \subsetneq \widehat{N}_{\ell-1} \subsetneq \cdots \subsetneq \widehat{N}_0 = U = [n]$$

be the chain decomposition generated by Algorithm 5. Based on such decomposition, we aim to demonstrate that running the modified greedy algorithm, Algorithm 3, for each layer $\widehat{N}_i \setminus \widehat{N}_{i+1}$ separately achieves $(\frac{1}{4} - \varepsilon)$-selectability using $O_\varepsilon(\log^4 n)$ samples.

We first delineate the parameters for Algorithm 5, setting k as $\frac{\log n}{\log(1+\varepsilon)\log\frac{1}{4}(1-\varepsilon)} = \Theta_\varepsilon(\log n)$ and defining $\{c_j\}_{j=1}^{k+1}$ as an arithmetic sequence with initial term $c_1 = \frac{1}{2} + \frac{\varepsilon}{2}$ and terminal term $c_{k+1} = \frac{1}{2} + \varepsilon$. In each layer i, $s = \Theta\left(\frac{k^2 \log n \log \varepsilon^{-1}}{\varepsilon^2}\right) = \Theta_\varepsilon(\log^3 n)$ fresh samples are needed for $\widehat{\text{Select}}(\widehat{N}_i, \widehat{c})$.

Lemma 4.5. *Let ℓ denote the number of layers in the chain decomposition generated by Algorithm 5, with parameters set as above. The following holds with probability at least $1 - \frac{\mathrm{poly}(\varepsilon)}{\mathrm{poly}(n)}$:*

- $\ell = O_\varepsilon(\log n)$.
- $(1 - \frac{1}{k})^\ell \geq 1 - \varepsilon$.

Proof. Consider any protected set $\widehat{N}_i$ and an index j_i uniformly selected from $[k]$ in layer i. Let $\delta = \frac{\varepsilon}{4k}$ represent the gap between $\frac{c_{j_i}+c_{j_i+1}}{2}$ and both c_{j_i} and c_{j_i+1}. Given the parameters previously defined, we take $s = \Theta\left(\frac{k^2 \log n \log \varepsilon^{-1}}{\varepsilon^2}\right) = \Theta\left(\frac{\log n \log \varepsilon^{-1}}{\delta^2}\right)$ new samples during the execution of $\widehat{\text{Select}}(\widehat{N}_i, \widehat{c})$. By applying Lemma 4.4, it is established that with probability at least $1 - \frac{\text{poly}(\varepsilon)}{\text{poly}(n)}$:

$$\text{Select}(\widehat{N}_i, c_{j_i+1}) \subseteq \widehat{N}_{i+1} = \widehat{\text{Select}}(\widehat{N}_i, \frac{c_{j_i} + c_{j_i+1}}{2}) \subseteq \text{Select}(\widehat{N}_i, c_{j_i}).$$

It always holds that $\text{Select}(\widehat{N}_i, c_{j_i}) \subseteq \text{Select}(\widehat{N}_i, c_1)$. Therefore,

$$\text{rank}\left(\widehat{N}_{i+1}\right) \leq \text{rank}\left(\text{Select}(\widehat{N}_i, c_1)\right)$$
$$< \frac{1}{1+\varepsilon} \text{rank}\left(\widehat{N}_i\right).$$

where the last inequality follows from Lemma 4.2 with $b = \frac{1}{2}$ and $c = \frac{1}{2} + \frac{1}{2}\varepsilon$. Since Algorithm 5 terminates when $\text{rank}(\widehat{N}_\ell) < 1$, it follows that

$$\ell \leq \log_{1+\varepsilon} \text{rank}\left(\widehat{N}_0\right) \leq \frac{\log n}{\log(1+\varepsilon)} = O_\varepsilon(\log n).$$

Now, to demonstrate the second property, recall that k is defined as $\frac{\log n}{\log(1+\varepsilon) \log_{1/4}(1-\varepsilon)}$. It thus follows that

$$\frac{\ell}{k} \leq \log_{1/4}(1-\varepsilon). \tag{5}$$

Therefore,

$$\left(1 - \frac{1}{k}\right)^\ell = \left(\left(1 - \frac{1}{k}\right)^k\right)^{\frac{\ell}{k}} \geq \left(\frac{1}{4}\right)^{\frac{\ell}{k}} \geq 1 - \varepsilon,$$

where the first inequality leverages the property that $\left(1 - \frac{1}{k}\right)^k$ is an increasing function for $k \geq 2$, and k is trivially at least 2. The final inequality is justified by Equation (5). □

Our algorithm employs $O_\varepsilon(\log^3 n)$ samples per layer across ℓ layers in total. Combining with Lemma 4.5, it is evident that the total number of samples required by our algorithm does not exceed $O_\varepsilon(\log^4 n)$.[4]

The remaining task is to demonstrate the selectability of Algorithm 5. Consider any element $e \in U$; our objective is to establish that

$$\mathbf{Pr}[e \text{ is accepted } | e \text{ is active}] \geq \frac{1}{2} - O(\varepsilon).$$

Given that x is contained within $\frac{1}{2} \cdot \mathcal{P}_\mathcal{M}$, it is deduced that our algorithm achieves a selectability of $\frac{1}{4} - O(\varepsilon)$. For this purpose, we will introduce a stochastic process to simulate the execution of Algorithm 5 with respect to a fixed element e.

---

[4]Should the conditions specified in Lemma 4.5 not be met, Algorithm 5 terminates immediately and announces failure. This approach has a negligible impact on the algorithm's selectability (in the order of $\frac{\text{poly}(\varepsilon)}{\text{poly}(n)}$) and guarantees that the algorithm deterministically requires $O_\varepsilon(\log^4 n)$ samples.

Stochastic Process. Fix an element $e \in U$. Initiating with $i = 0$ and $\widehat{N}_0 = U = [n]$, our stochastic process sequentially reveals the randomness, i.e., $j_0, j_1, \ldots, j_\ell$, across each layer. Assuming the process is at layer i, with the protected set $\widehat{N}_i$ containing element e, we proceed to define $N_{i+1,t} = \mathsf{Select}(\widehat{N}_i, c_t)$ for each $t \in [k+1]$. Specifically, we set $N_{i+1,0} = \widehat{N}_i$ and $N_{i+1,k+2} = \varnothing$. It is evident that the sequence $\{N_{i,t}\}_{t=1}^{k+2}$ satisfies the following property:

$$\varnothing = N_{i,k+2} \subseteq N_{i,k+1} \subseteq \cdots \subseteq N_{i,1} \subseteq N_{i,0} = \widehat{N}_i.$$

Given that $e \in \widehat{N}_i$, there must exist a unique $t^* \in \{0, 1, \ldots, k+1\}$ such that $e \in N_{i,t^*} \setminus N_{i,t^*+1}$. We proceed by revealing the randomness and selecting j_i uniformly at random from the set $[k]$. With the choice of j_i, the next step involves defining $\widehat{N}_{i+1} = \widetilde{\mathsf{Select}}(\widehat{N}_i, (c_{j_i} + c_{j_i+1})/2)$. Lemma 4.5 ensures that[5]

$$N_{i,j_i+1} = \mathsf{Select}(\widehat{N}_i, c_{j_i+1}) \subseteq \widehat{N}_{i+1} \subseteq \mathsf{Select}(\widehat{N}_i, c_{j_i}) = N_{i,j_i}.$$

Depending on the specific outcome of j_i, we examine the subsequent scenarios:

- **Case 1:** If $j_i < t^*$, this indicates that

$$e \in N_{i,t^*} \subseteq N_{i,j_i+1} \subseteq \widehat{N}_{i+1}.$$

  Consequently, e progresses to a subsequent layer. We then increase i by 1 to move to layer $i + 1$.
- **Case 2:** If $j_i > t^*$, it implies that

$$\widehat{N}_{i+1} \subseteq N_{i,j_i} \subseteq N_{i,t^*+1}.$$

  Observe that $e \notin N_{i,t^*+1}$. This observation implies that e is also not a member of $\widehat{N}_{i+1}$ and N_{i,j_i}. Consequently, the probability of e being selected in this scenario can be expressed as follows:

$$\begin{aligned} \mathbf{Pr}[e \text{ is selected } | \ e \text{ is active}] &\geq 1 - \mathbf{Pr}[e \in \mathsf{span}(((R(\boldsymbol{x}) \cap \widehat{N}_i) \cup \widehat{N}_{i+1}) \setminus \{e\})] \\ &\geq 1 - \mathbf{Pr}[e \in \mathsf{span}(((R(\boldsymbol{x}) \cap \widehat{N}_i) \cup N_{i,j_i}) \setminus \{e\})] \\ &\geq \frac{1}{2} - \varepsilon. \end{aligned}$$

  The first inequality is derived from the premise that $e \in \widehat{N}_i \setminus \widehat{N}_{i+1}$ while we are employing a greedy strategy on the matroid $\mathcal{M}|_{\widehat{N}_i}/\widehat{N}_{i+1}$. The subsequent inequality emerges from the inclusion $\widehat{N}_{i+1} \subseteq N_{i,j_i}$, and the final inequality is in accordance with the definition that

$$N_{i+1,j_i} = \mathsf{Select}(\widehat{N}_i, c_{j_i}).$$

  Hence, in this scenario, our algorithm demonstrates effectiveness, thereby concluding the stochastic process.
- **Case 3:** If $j_i = t^*$, the stochastic process immediately terminates with no guarantee for e.

Our algorithm fails at element e only when the stochastic process ends in Case 3. As j_i is drawn uniformly at random from $[k]$, the stochastic process falls into Case 3 with a probability of $\frac{1}{k}$ in each layer. As there are at most ℓ layers in total, the selectability is given as

$$\mathbf{Pr}[e \text{ is selected } | \ e \text{ is active}] \geq \left(\frac{1}{2} - \varepsilon\right) \cdot \left(1 - \frac{1}{k}\right)^\ell \geq \frac{1}{2} - O(\varepsilon).$$

where the last inequality follows from Lemma 4.5. We hereby conclude our proof.

---

[5]Similarly, should this property not be upheld, both our algorithm and the stochastic process will be terminated immediately. This affects the algorithm's selectability by only a negligible margin.

References

[1] Saeed Alaei. 2014. Bayesian Combinatorial Auctions: Expanding Single Buyer Mechanisms to Many Buyers. *SIAM J. Comput.* 43, 2 (2014), 930–972.

[2] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. 2012. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 18–35.

[3] Pablo D Azar, Robert Kleinberg, and S Matthew Weinberg. 2014. Prophet inequalities with limited information. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 1358–1377.

[4] Pablo D Azar, Robert Kleinberg, and S Matthew Weinberg. 2019. Prior independent mechanisms via prophet inequalities with limited information. *Games and Economic Behavior* 118 (2019), 511–532.

[5] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. 2007. Matroids, secretary problems, and online mechanisms. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, Nikhil Bansal, Kirk Pruhs, and Clifford Stein (Eds.). SIAM, 434–443.

[6] Kiarash Banihashem, MohammadTaghi Hajiaghayi, Dariusz R Kowalski, Piotr Krysta, and Jan Olkowski. 2024. Power of Posted-price Mechanisms for Prophet Inequalities. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 4580–4604.

[7] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. 2016. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 1046–1059.

[8] Niv Buchbinder, Moran Feldman, and Mohit Garg. 2019. Deterministic $(1/2+\varepsilon)$-Approximation for Submodular Maximization over a Matroid. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 241–254.

[9] Constantine Caramanis, Paul Dütting, Matthew Faw, Federico Fusco, Philip Lazos, Stefano Leonardi, Orestis Papadigenopoulos, Emmanouil Pountourakis, and Rebecca Reiffenhäuser. 2022. Single-sample prophet inequalities via greedy-ordered selection. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1298–1325.

[10] Shuchi Chawla, Kira Goldner, Anna R Karlin, and J Benjamin Miller. 2020. Non-adaptive matroid prophet inequalities. *arXiv preprint arXiv:2011.09406* (2020).

[11] Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. 2010. Multi-parameter mechanism design and sequential posted pricing. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, Leonard J. Schulman (Ed.). ACM, 311–320.

[12] José Correa and Andrés Cristi. 2023. A Constant Factor Prophet Inequality for Online Combinatorial Auctions. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 686–697.

[13] José Correa, Andrés Cristi, Boris Epstein, and José A Soto. 2023. Sample-driven optimal stopping: From the secretary problem to the iid prophet inequality. *Mathematics of Operations Research* (2023).

[14] José Correa, Paul Dütting, Felix Fischer, Kevin Schewior, and Bruno Ziliotto. 2021. Unknown IID Prophets: Better Bounds, Streaming Algorithms, and a New Impossibility. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[15] José Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. 2017. Posted price mechanisms for a random stream of customers. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 169–186.

[16] José R Correa, Andrés Cristi, Boris Epstein, and José A Soto. 2020. The two-sided game of googol and sample-based prophet inequalities. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2066–2081.

[17] José R. Correa, Paul Dütting, Felix A. Fischer, and Kevin Schewior. 2019. Prophet Inequalities for I.I.D. Random Variables from an Unknown Distribution. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, Anna Karlin, Nicole Immorlica, and Ramesh Johari (Eds.). ACM, 3–17.

[18] Andrés Cristi and Bruno Ziliotto. 2023. Prophet Inequalities Require Only a Constant Number of Samples. *arXiv preprint arXiv:2311.09141* (2023).

[19] Peerapong Dhangwatnotai, Tim Roughgarden, and Qiqi Yan. 2010. Revenue maximization with a single sample. In *Proceedings of the 11th ACM conference on Electronic commerce*. 129–138.

[20] Atanas Dinev and S Matthew Weinberg. 2023. Simple and Optimal Online Contention Resolution Schemes for k-Uniform Matroids. *arXiv preprint arXiv:2309.10078* (2023).

[21] Shaddin Dughmi. 2019. The outer limits of contention resolution on matroids and connections to the secretary problem. *arXiv preprint arXiv:1909.04268* (2019).

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

[23] Paul Dütting, Thomas Kesselheim, and Brendan Lucier. 2020. An o (log log m) prophet inequality for subadditive combinatorial auctions. *ACM SIGecom Exchanges* 18, 2 (2020), 32–37.

[24] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 381–390.

[25] Evgenii Borisovich Dynkin. 1963. The optimum choice of the instant for stopping a Markov process. *Soviet Mathematics* 4 (1963), 627–629.

[26] Soheil Ehsani, MohammadTaghi Hajiaghayi, Thomas Kesselheim, and Sahil Singla. 2018. Prophet secretary for combinatorial auctions and matroids. In *Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms*. SIAM, 700–714.

[27] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. 2020. Online Stochastic Max-Weight Matching: Prophet Inequality for Vertex and Edge Arrival Models. In *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, Péter Biró, Jason Hartline, Michael Ostrovsky, and Ariel D. Procaccia (Eds.). ACM, 769–787.

[28] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. 2022. General graphs are easier than bipartite graphs: tight bounds for secretary matching. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 1148–1177.

[29] Michal Feldman, Nick Gravin, and Brendan Lucier. 2014. Combinatorial auctions via posted prices. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 123–135.

[30] Moran Feldman, Ola Svensson, and Rico Zenklusen. 2014. A simple O (log log (rank))-competitive algorithm for the matroid secretary problem. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 1189–1201.

[31] Moran Feldman, Ola Svensson, and Rico Zenklusen. 2016. Online Contention Resolution Schemes. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, Robert Krauthgamer (Ed.). SIAM, 1014–1033.

[32] Hu Fu, Pinyan Lu, Zhihao Gavin Tang, Abner Turkieltaub, Hongxun Wu, Jinzhao Wu, and Qianfan Zhang. 2022. Oblivious online contention resolution schemes. In *Symposium on Simplicity in Algorithms (SOSA)*. SIAM, 268–278.

[33] Nikolai Gravin and Hongao Wang. 2019. Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 93–109.

[34] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. 2007. Automated online mechanism design and prophet inequalities. In *AAAI*, Vol. 7. 58–65.

[35] Moritz Hardt and Guy N Rothblum. 2010. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st annual symposium on foundations of computer science*. IEEE, 61–70.

[36] Moritz Hardt and Jonathan Ullman. 2014. Preventing false discovery in interactive data analysis is hard. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 454–463.

[37] Jason D Hartline and Tim Roughgarden. 2009. Simple versus optimal mechanisms. In *Proceedings of the 10th ACM conference on Electronic commerce*. 225–234.

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

[39] Jiashuo Jiang, Will Ma, and Jiawei Zhang. 2022. Tight Guarantees for Multi-unit Prophet Inequalities and Online Stochastic Knapsack. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1221–1246.

[40] Haim Kaplan, David Naori, and Danny Raz. 2020. Competitive analysis with a sample and the secretary problem. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2082–2095.

[41] Haim Kaplan, David Naori, and Danny Raz. 2022. Online weighted matching with a sample. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1247–1272.

[42] Robert Kleinberg and S. Matthew Weinberg. 2012. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. 123–136.

[43] Ulrich Krengel and Louis Sucheston. 1977. Semiamarts and finite values. (1977).

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

[45] Oded Lachish. 2014. O (log log rank) competitive ratio for the matroid secretary problem. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 326–335.

[46] Euiwoong Lee and Sahil Singla. 2018. Optimal Online Contention Resolution Schemes via Ex-Ante Prophet Inequalities. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland (LIPIcs, Vol. 112)*, Yossi Azar, Hannah Bast, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 57:1–57:14.

[47] Bo Li, Xiaowei Wu, and Yutong Wu. 2023. Prophet Inequality on I.I.D. Distributions: Beating 1-1/e with a Single Query. arXiv:2205.05519 [cs.DS]

[48] Wenzheng Li. 2023. A Reduction from Matroid Secretary Problem to Single-Sample Matroid Prophet Secretary. personal communication.

[49] Pranav Nuti and Jan Vondrák. 2023. Secretary problems: The power of a single sample. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2015–2029.

[50] Bo Peng and Zhihao Gavin Tang. 2022. Order selection prophet inequality: From threshold optimization to arrival time design. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 171–178.

[51] Aviad Rubinstein. 2016. Beyond matroids: Secretary problem and prophet inequality with general constraints. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 324–332.

[52] Aviad Rubinstein and Sahil Singla. 2017. Combinatorial prophet inequalities. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1671–1687.

[53] Aviad Rubinstein, Jack Z. Wang, and S. Matthew Weinberg. 2020. Optimal Single-Choice Prophet Inequalities from Samples. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA (LIPIcs, Vol. 151)*, Thomas Vidick (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 60:1–60:10.

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

[55] Richard Santiago, Ivan Sergeev, and Rico Zenklusen. 2023. Simple Random Order Contention Resolution for Graphic Matroids with Almost no Prior Information. In *Symposium on Simplicity in Algorithms (SOSA)*. SIAM, 84–95.

[56] Alexander Schrijver. 2003. Combinatorial Optimization Polyhedra and Efficiency Volume B Matroids, Trees, Stable Sets Chapters 39-69. *Algorithms and Combinatorics* 24, 1 (2003), ALL–ALL.

[57] Thomas Steinke and Jonathan Ullman. 2015. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Conference on learning theory*. PMLR, 1588–1628.

A Missing Proofs from Section 3

A.1 Tie-Breaking for Distributions with Point Masses

Our algorithm for prophet inequality in Section 3 can only work with continuous distributions. For distributions with point masses, we use the following tie-breaking technique from [53] to make them continuous.

For a distribution $\mathcal{D}$ with point masses, we can define $\mathcal{D}'$ as the bivariate distribution $\mathcal{D} \times U([0,1])$, the product of $\mathcal{D}$ and the uniform distribution over $[0,1]$. Then for $(x,t), (y,u)$ sampled from $\mathcal{D}'$, define $(x,t) > (y,u)$ if either $x > y$ or $x = y$ and $t > u$. Observe that the equality holds with zero probability. Therefore, if we define $F(x,t) = \mathbf{Pr}_{(y,u)\sim\mathcal{D}'}[(x,t) > (y,u)]$, we gain the desired property that F is continuous and $F(x,t) > F(y,u) \Leftrightarrow (x,t) > (y,u)$.

A.2 Proof of Lemma 3.4

PROOF OF LEMMA 3.4. Let the C.D.F. of τ_i be G_i. From Observation 3.2, we know that $i \in$ OPT if and only if $v_i > \tau_i$.[6] Therefore, by definition, $p^i(v) = \mathbf{Pr}[i \in \text{OPT} \mid v_i = v] = \mathbf{Pr}[v_i > \tau_i \mid v_i = v] = G_i(v)$. Together with $p_k = \varepsilon(1+\varepsilon)^k - \varepsilon^2$, we have that $p^i(T_i^{(k)}) \in [p_k, p_k + 2\varepsilon]$ if and only if $G_i(T_i^{(k)}) \in [\varepsilon(1+\varepsilon)^k - \varepsilon^2, \varepsilon(1+\varepsilon)^k + \varepsilon^2]$.

For each sample $1 \leq s \leq N$, $i \in U$ and $0 \leq k < m$, let $l_{i,k}^s = \mathbb{1}[G_i(\tau_i^s) < \varepsilon(1+\varepsilon)^k - \varepsilon^2]$ and $r_{i,k}^s = \mathbb{1}[G_i(\tau_i^s) < \varepsilon(1+\varepsilon)^k + \varepsilon^2]$. By Chernoff bound we have

$$\mathbf{Pr}\left[\sum_{s=1}^{N} l_{i,k}^s \geq \varepsilon(1+\varepsilon)^k N\right] \leq e^{-N\varepsilon^4/3} \leq \frac{\varepsilon}{2nm},$$

$$\mathbf{Pr}\left[\sum_{s=1}^{N} r_{i,k}^s \leq \varepsilon(1+\varepsilon)^k N\right] \leq e^{-N\varepsilon^4/2} \leq \frac{\varepsilon}{2nm}.$$

---

[6]We neglect the case where $v_i = \tau_i$ since it has zero probability.

Here n is the number of elements in U, and N is the number of samples. Observe that if $G_i(T_i^{(k)}) \notin [\varepsilon(1+\varepsilon)^k - \varepsilon^2, \varepsilon(1+\varepsilon)^k + \varepsilon^2]$, we either have $\sum_{s=1}^{N} l_{i,k}^s \geq \varepsilon(1+\varepsilon)^k N$ or $\sum_{s=1}^{N} r_{i,k}^s \leq \varepsilon(1+\varepsilon)^k N$, thus

$$\mathbf{Pr}\left[G_i(T_i^{(k)}) \notin [\varepsilon(1+\varepsilon)^k - \varepsilon^2, \varepsilon(1+\varepsilon)^k + \varepsilon^2]\right] \leq \frac{\varepsilon}{nm}.$$

Then by a union bound over all $i \in U$ and $0 \leq k < m$, we conclude that

$$\mathbf{Pr}\left[\exists i, k \text{ s.t. } G_i(T_i^{(k)}) \notin [\varepsilon(1+\varepsilon)^k - \varepsilon^2, \varepsilon(1+\varepsilon)^k + \varepsilon^2]\right] \leq \varepsilon,$$

which implies $\mathbf{Pr}[\forall i, k \text{ s.t. } G_i(T_i^{(k)}) \in [\varepsilon(1+\varepsilon)^k - \varepsilon^2, \varepsilon(1+\varepsilon)^k + \varepsilon^2]] \geq 1 - \varepsilon$.

Thus $\mathbf{Pr}[\forall i, k \text{ s.t. } p^i(T_i^{(k)}) \in [p_k, p_k + 2\varepsilon]] \geq 1 - \varepsilon$. $\qquad\square$

A.3 Proof of Lemma 3.8

We will need the strong basis exchange property of matroids.

Lemma A.1 (Strong basis exchange, Theorem 39.12 in [56]). *Let $\mathcal{M} = (U, \mathcal{I})$ be a matroid. Let A and B be two bases and $x \in A \setminus B$. Then there exists $y \in B \setminus A$ such that $A - x + y$ and $B - y + x$ are both bases.*

Proof of Lemma 3.8. Let $U = [n]$ and $w(1) \geq w(2) \geq \cdots \geq w(n)$. Suppose the rank of the matroid is r. We denote the elements in A by $\{a_i\}_{i \in [r]}$ where a_i's are in ascending order. Consequently, we have $w(a_1) \geq w(a_2) \geq \cdots \geq w(a_r)$. Without loss of generality, we can assume that the basis A is generated by feeding $1, 2, \ldots, n$ to the greedy algorithm sequentially.

Next, we state our algorithm for constructing the bijection f. We will need the strong basis exchange property stated in Lemma A.1.

ALGORITHM 6: Construct mapping f

---
1   Recall r is the rank of $\mathcal{M}$
2   $A_{r+1} \leftarrow A$
3   **for** $i \leftarrow r, r-1, \ldots, 1$ *in descending order* **do**
4   **if** $a_i \in A_{i+1} \setminus B$ **then**
5   By Lemma A.1, there exists $b_i \in B \setminus A_{i+1}$ such that both $A_{i+1} - a_i + b_i$ and $B - b_i + a_i$ are bases
6   Let $f(a_i) \leftarrow b_i$
7          $A_i \leftarrow A_{i+1} - a_i + f(a_i)$
8   **end**
9   **else**
10   Let $f(a_i) \leftarrow a_i$
11          $A_i \leftarrow A_{i+1}$
12   **end**
13   **end**

We first prove the following facts about Algorithm 6.

Fact A.2. $\{f(a_i), f(a_{i+1}), \ldots, f(a_r)\} \subset A_i$.

PROOF. For all $j \geq i$, if $a_j \in A_{j+1} \setminus B$, we have $f(a_j) \in B \setminus A_{j+1}$ which implies $f(a_j) \notin \{a_i, \ldots, a_j\}$. Together with $f(a_j) \in A_j$, we can see that $f(a_j)$ is never eliminated and $f(a_j) \in A_i$. If $a_j \notin A_{j+1} \setminus B$, we know $f(a_j) = a_j$, and it is in both A_j and A_i. $\qquad\square$

Fact A.3. *The resulting mapping f is a bijection.*

PROOF. If $a_i \in A_{i+1} \setminus B$, since $b_i \in B \setminus A_{i+1}$, we know $f(a_i) = b_i \notin \{f(a_{i+1}), \ldots, f(a_r)\}$ (by Fact A.2). If $a_i \notin A_{i+1} \setminus B$, $f(a_i)$ equals a_i itself. Note for all $j > i$, $A_{j+1} - a_j + f(a_j)$ is a basis and $a_i \in A_{j+1}$. Therefore $f(a_i) = a_i$ cannot equal to any of such $f(a_j)$.

Thus $f(a_i) \neq f(a_j)$ for all $j > i$. f is an injection. Since A and B are two bases and $|A| = |B|$, f is a bijection. □

Fact A.4. *For all $a_i \in A$, $w(f(a_i)) \leq w(a_i)$.*

PROOF. If $w(f(a_i)) > w(a_i)$, we know $f(a_i) < a_i$. Thus $f(a_i)$ is fed to the greedy algorithm before a_i. On the other hand, since $f(a_i) \neq a_i$, we must have $a_i \in A_{i+1} \setminus B$ and $A_{i+1} - a_i + f(a_i)$ is a basis. Therefore $\{a_1, a_2, \ldots, a_{i-1}, f(a_i)\} \in \mathcal{I}$. Then our greedy algorithm should have selected $f(a_i)$ before it selected a_i. Since we enumerate a_i in decreasing order of their index, if our algorithm selected $f(a_i)$, we must have $f(a_i) \in A_{i+1}$. This contradicts with the fact $f(a_i) \in B \setminus A_{i+1}$. □

Finally, we verify that for all a_i, $B - f(a_i) + a_i$ is a basis. It is trivial when $f(a_i) = a_i$. Otherwise, $B - f(a_i) + a_i$ is also a basis according to our construction. This finishes the proof of the lemma. □

B Missing Proofs from Section 4

B.1 Uniqueness of $\mathrm{Select}(N, c)$ and $\widehat{\mathrm{Select}}(N, c)$

In this section, we show $\mathrm{Select}(N, c)$ can be uniquely defined by the set S returned by Algorithm 1, no matter which element e is chosen each time at Line 3.

Let $\mathcal{T}$ be the family of sets that do not satisfy the condition in Line 3, i.e., for every $T \in \mathcal{T}$, $e \in N \setminus T$,

$$\mathbf{Pr}\left[e \in \mathrm{span}(((R(\boldsymbol{x}) \cap N) \cup T) \setminus \{e\})\right] \leq c.$$

By definition, the set S returned by Algorithm 1 always belongs to $\mathcal{T}$. We claim such S is the unique minimum set in $\mathcal{T}$. The plan is to prove by induction that the changing S in always remains a subset of T during the execution of Algorithm 1 for every $T \in \mathcal{T}$. Initially, $S = \emptyset$ and the statement trivially holds. Each time an element is added to S, we must have

$$\mathbf{Pr}\left[e \in \mathrm{span}(((R(\boldsymbol{x}) \cap N) \cup T) \setminus \{e\})\right] \geq \mathbf{Pr}\left[e \in \mathrm{span}(((R(\boldsymbol{x}) \cap N) \cup S) \setminus \{e\})\right] > c,$$

where the first inequality is due to the monotonicity of span and the induction hypothesis $S \subseteq T$. That implies $e \in T$ by definition of the family $\mathcal{T}$. Therefore, $S \cup \{e\} \subseteq T$ and we have completed the proof. The same argument holds if we replace $\mathbf{Pr}$ with $\widehat{\mathbf{Pr}}$, and thus $\widehat{\mathrm{Select}}(N, c)$ is also uniquely defined by Algorithm 4 (for any fixed samples $R_1, R_2, \ldots, R_s$).

B.2 Proof of Lemma 4.2

PROOF OF LEMMA 4.2. Let r be $\mathrm{rank}(S)$. Take $e_1, e_2, \ldots, e_r$ from S as follows: e_i is the item that increases the rank of S from $i - 1$ to i during the algorithm.

Before the formal proof, we first explain the intuition. Note we know that $\mathbf{Pr}[e_i \notin \mathrm{span}(((R(\boldsymbol{x}) \cap N) \cup \{e_1, e_2, \ldots, e_{i-1}\}) \setminus \{e_i\})] < 1 - c$ holds for all $i \in [r]$ since e_i is added to S by our algorithm. So instead of directly consider $\mathrm{rank}(S)$, we start from the random set $R(\boldsymbol{x}) \cap N$ and add $e_1, e_2, \ldots, e_r$ to it in order. Each item we add e_i, the probability that it increase the rank by one is strictly smaller than $1 - c$. Finally, we get a set with rank $\mathbf{E}[\mathrm{rank}(S \cup (R(\boldsymbol{x}) \cap N))]$ in expectation. This is an upper bound of $\mathrm{rank}(S)$, and we can further upper bound the rank of it by $\mathrm{rank}(N)$ using the fact that e_i increase its rank with probability strictly less than $1 - c$.

Formally, we have

$$\mathrm{rank}(S) \leq \mathbf{E}\left[\mathrm{rank}(S \cup (R(\boldsymbol{x}) \cap N))\right]$$

$$\leq \mathbf{E}\left[|R(\boldsymbol{x}) \cap N|\right] + \sum_{i=1}^{r} \mathbf{Pr}\left[e_i \notin \mathrm{span}(((R(\boldsymbol{x}) \cap N) \cup \{e_1, e_2, \ldots, e_{i-1}\}) \setminus \{e_i\})\right]$$

$$< b\,\mathrm{rank}(N) + (1 - c)\,\mathrm{rank}(S),$$

where we used the fact that $\mathbf{E}[|R(\boldsymbol{x}) \cap N|] \leq \sum_{i \in N} x_i \leq b\operatorname{rank}(N)$ and $\mathbf{Pr}[e_i \notin \operatorname{span}(((R(\boldsymbol{x}) \cap N) \cup \{e_1, \cdots, e_{i-1}\}) \setminus \{e_i\})] < 1 - c$. As a result, $\operatorname{rank}(S) < \frac{b}{c} \cdot \operatorname{rank}(N)$. $\qquad\square$

C Non-existence of $\Omega(1)$-Balanced Matroid CRS with $o(\log n)$ Samples

A previous work [32] showed the non-existence of $\Omega(1)$-balanced matroid contention resolution scheme (CRS) with $O(1)$ samples. We note that their impossibility result can be extended to any $\Omega(1)$-balanced matroid CRS with $o(\log n)$ samples by using the same hard instance. As a direct consequence, it is also impossible to obtain a constant-selectable OCRS with $o(\log n)$ samples.

One of their hard instance is the graphic matroid $\mathcal{M} = (E, \mathcal{I})$ defined on the complete bipartite graph $K_{N,M} = (U \cup V, E)$ with bipartition $U = \{u_1, u_2, \ldots, u_N\}$ and $V = \{v_1, v_2, \ldots, v_M\}$. Let $\mathcal{P}_{\mathcal{M}}$ be the polytope of the matroid and $\delta(u)$ be the set of edges incident to a vertex $u \in U \cup V$.
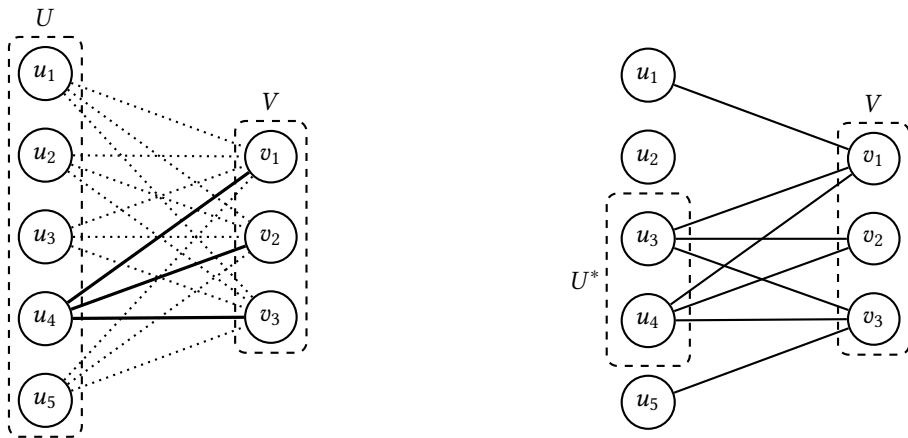
(a) The complete bipartite graph $K_{N,M}$ and $\boldsymbol{x}^i$. Here $i = 4$, edges adjacent to u_i has probability $x_e^i = 1$ of being active, while other edges each only has probability $\frac{1}{M}$ of being active. In any $\Omega(1)$-balanced CRS, they have to be selected with constant probability.

(b) A realization $R(\boldsymbol{x}^i)$ of this instance. Let U^* be the set of all vertices on left side of degree M. If N is large enough, there will be many vertices happen to be in U^*. These vertices in U^* are indistinguishable to CRS, and u_i ($i = 4$) is hidden between them.

Fig. 1. The hard instance for graphic matroids in [32]

Consider N points $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N$ where for every $i \in [N]$ and $e \in E$,

$$x_e^i = \begin{cases} 1 & \text{if } e \in \delta(u_i), \\ \frac{1}{M} & \text{otherwise.} \end{cases}$$

Their intermediate result can be summarized as follows:

Lemma C.1 (Section 3.1, [32]). *The points $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N$ are in the matroid polytope $\mathcal{P}_{\mathcal{M}}$. Moreover, for any $N, M \geq 1$, $s \geq 0$, and $0 < c \leq 1$, if there exists a c-balanced CRS with s samples for $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N \in \mathcal{P}_{\mathcal{M}}$, then*

$$c \leq \frac{1}{M} + \frac{M^{(s+1)M}}{N}.$$

Then they concluded the non-existence of $\Omega(1)$-balanced CRS for any constant s by letting $N \gg M^M$. The same argument, notably, remains valid to cases where s is beyond constant. Specifically,

for any $s \leq \frac{\log N}{2M \log M} - 1$, the inequality above becomes

$$c \leq \frac{1}{M} + \frac{1}{\sqrt{N}}.$$

In other words, a $(\frac{1}{M} + \frac{1}{\sqrt{N}})$-balanced CRS for $\mathcal{M}$ requires at least $\frac{\log N}{2M \log M}$ samples. By choosing $N \gg M^M$ (and note that $n = |E| = NM$), we conclude that $\Omega(1)$-balanced matroid CRS is impossible given $o(\log n)$ samples.