The Limitations of Optimization from Samples

Eric Balkanski[*] Aviad Rubinstein[†] Yaron Singer[‡]

Abstract

As we grow highly dependent on data for making predictions, we translate these predictions into models that help us make informed decisions. But what are the guarantees we have? Can we optimize decisions on models learned from data and be guaranteed that we achieve desirable outcomes? In this paper we formalize this question through a novel model called optimization from samples (OPS). In the OPS model, we are given sampled values of a function drawn from some distribution and our objective is to optimize the function under some constraint. Our main interest is in the following question: are functions that are learnable (from samples) and approximable (given oracle access to the function) also optimizable from samples?

We show that there are classes of submodular functions which have desirable approximation and learnability guarantees and for which no reasonable approximation for optimizing from samples is achievable. In particular, our main result shows that even for maximization of coverage functions under a cardinality constraint k, there exists a hypothesis class of functions that cannot be approximated within a factor of $n^{-1/4+\epsilon}$ (for any constant $\epsilon > 0$) of the optimal solution, from samples drawn from the uniform distribution over all sets of size at most k. In the general case of monotone submodular functions, we show an $n^{-1/3+\epsilon}$ lower bound and an almost matching $\tilde{\Omega}(n^{-1/3})$-optimization from samples algorithm. Additive and unit-demand functions can be optimized from samples to within arbitrarily good precision. Finally, we also consider a corresponding notion of additive approximation for continuous optimization from samples, and show near-optimal hardness for concave maximization and convex minimization.

1 Introduction

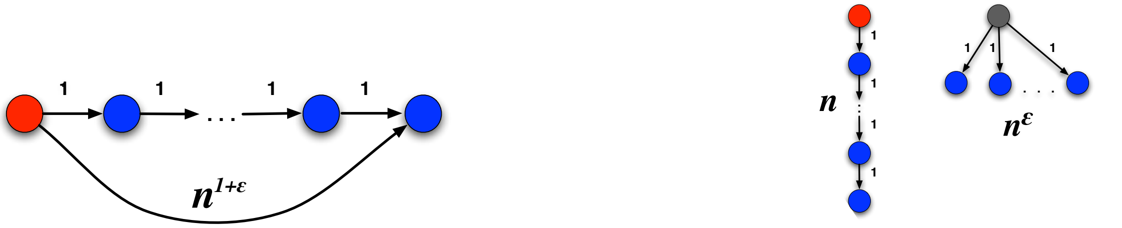
The traditional approach in optimization typically assumes some underlying model that is known to the algorithm designer, and the focus is on optimizing an objective defined on the given model. In the shortest path problem, for example, we are given a weighted graph and the goal is to find the shortest weighted path from a source to a destination. In the influence problem we are given a weighted graph in which the weights encode the likelihood of one node to forward information to its neighbors, and the goal is to select a subset of nodes to spread information to maximize the expected number of nodes in the network that eventually receive information.

In many applications we do not know the models we optimize over, and employ machine learning techniques to approximate them. This is often a reasonable approach since machine learning makes predictions by observing data and estimating parameters of the model in sophisticated ways. For

[*] School of Engineering and Applied Sciences, Harvard University. ericbalkanski@g.harvard.edu. This research was supported by Smith Family Graduate Science and Engineering Fellowship and by NSF grant CCF-1301976, CAREER CCF-1452961.

[†] Department of Electrical Engineering and Computer Sciences, UC Berkeley. aviad@eecs.berkeley.edu. This research was supported by Microsoft Research PhD Fellowship, NSF grant CCF1408635, and Templeton Foundation grant 3966, and done in part at the Simons Institute for the Theory of Computing.

[‡] School of Engineering and Applied Sciences, Harvard University. yaron@seas.harvard.edu. This research was supported by NSF grant CCF-1301976, CAREER CCF-1452961.



(a) Shortest path

(b) Influence maximization

Figure 1: Example of shortest path and influence maximization. The example on the left illustrates a shortest path problem instance where we are given a graph with edge weights. Here the edges on the top chain all have weight 1, and the bottom edge has weight of $n^{1+\epsilon}$. The example on the right is an instance of the influence problem where we are again given a weighted graph and the weights encode the probability of one node to forward information to its neighbors. The graph is composed of a chain (left) where every node receives information with probability 1, and a star of degree n^{ϵ} (right) where the nodes receive information from the grey node with probability 1 as well.

finding good driving routes, for example, we could first observe road traffic and fit weights to a graph that represents the travel time on a road network, and then apply a shortest path calculation. To find influential individuals, we would observe messages forwarded on Twitter, fit weights to a graph that represents the diffusion model, and then apply an influence maximization algorithm. Naturally, the way we estimate the model has an effect on our decision, but by how much? To start this discussion, let's consider the examples below which illustrate two extreme scenarios.

Example: small estimation errors can lead to poor decisions. Figure 1 illustrates two networks, one for the shortest path problem (Figure 1a) and the other for the influence maximization problem (Figure 1b). Suppose that in both networks the weights are estimated up to $(1 \pm \epsilon)$ factor of error, for some fixed $\epsilon > 0$. How does this estimation error affect the decision made by the algorithm? In the shortest path problem, if every edge weight was estimated to be $(1+\epsilon)$ of its true weight, applying the shortest path algorithm will be a good solution even on the distorted model: in the worst case the path selected will be a $(1+\epsilon)$ approximation. In the influence maximization example, a small error in the estimation of the parameters leads the algorithm to make a very poor decision. For the problem of selecting a single node to spread information, the correct solution is to select the red node, and the value in this case is n. But suppose now that the weight of every edge in the chain is estimated to be $(1-\epsilon)$ instead of 1, and the n^{ϵ} edges incident to the grey node are all estimated correctly as having weight 1. In this case, the estimated expected number of nodes that receive information from the red node is evaluated as only a constant rather than n, and for any fixed ϵ and sufficiently large n, the algorithm will select the grey node instead, which reaches n^{ϵ} nodes. Thus, a small estimation error would result in a decision that is a $n^{1-\epsilon}$ approximation to the optimal solution.

The general consensus is to treat modeling and decisions separately: machine learning is typically responsible for creating a model from observed data, and algorithms then optimize the decision using this model as if it were exact.[1] But as shown above, this approach can be problematic as the optimization critically depends on the guarantees of the learned model.

[1] Note that unlike bandit and experts settings, we are considering settings in which decisions can be made offline. We allow the prediction algorithm to train offline and generate a model, and then apply the optimization.

Optimization from samples: from predictions to decisions. In this paper we investigate the limitations and possibilities of optimization from sampled data. At a high level, we are interested in understanding whether the fact that we can statistically learn a model, and optimize over the model when given full information, implies that we can optimize decisions from sampled data. In other words, does our ability to make good predictions imply we can make good decisions?

To formalize our question, we consider a *model* as a real-valued function. In the shortest path example, the function is given a path as input and returns its expected latency. For maximizing influence, the function receives a set of individuals and returns the expected number of individuals that receive information. A natural starting point is to consider functions which are both *approximable* and *learnable.* If a decision cannot be optimized on a given model it would be impossible to optimize the same decision with partial information. Similarly, if the behavior of the model cannot be predicted from observations, then (at least intuitively) optimizing decisions on the model from observations seems hard.

1.1 Optimization from Samples

We say that a class of functions is *α-approximable* under some constraint if there is a polynomial-time algorithm that approximates the optimal solution up to some factor $\alpha > 0$. For learnability, we employ the framework of *Probably Mostly Approximately Correct* (PMAC) due to Balcan and Harvey [2] which is a generalization of Valiant's celebrated *Probably Approximately Correct* (PAC) learnability [25]. Informally, these concepts of learnability guarantee that after observing polynomially-many samples of sets and their function values, one can construct a surrogate function that is likely to mimic the behavior of the function observed from the samples. Does PMAC learnability and α-approximability, for some reasonable α, imply that we can approximate our objective well by observing samples?

A model for optimization from samples (OPS). To formalize our goal clearly, we define the *optimization from samples* criteria. We say that a hypothesis class of functions is *optimizable from samples (OPS) under constraint $\mathcal{M}$* for some given distribution $\mathcal{D}$, if there exists an algorithm which receives samples $\{S_i, f(S_i)\}$ as input, where the sets $\{S_i\}_{i=1}^{m}$ are drawn i.i.d. from a distribution $\mathcal{D}$ over $\mathcal{M}$ and where the function f is in the hypothesis class of functions, and w.h.p. the algorithm returns $S \in \mathcal{M}$ s.t. $f(S) \geq \alpha \max_{T \in \mathcal{M}} f(T)$, for some constant $\alpha > 0$.

Does learnability and approximablity imply optimizability from samples?

Our main result is a spoiler: we show that there are interesting classes of submodular functions which are not optimizable from samples. These impossibility results are information theoretic lower bounds. That is, the limitations of optimization from samples are not due to computational complexity but to sample complexity. In particular, the limitation is due to the fact that the samples simply do not contain sufficient information for optimization. Even though computational complexity is not important for our impossibility results, the problems that we consider are approximately optimizable in the sense that it requires polynomially-many queries to a value oracle of the function we aim to optimize to obtain a solution that is a constant factor away from optimal.

Submodular optimization. Traditionally, machine learning has been harnessing convex optimization to design fast algorithms with provable guarantees for a broad range of applications. In recent years however, there has been a surge of interest in applications that involve *discrete optimization.* For discrete domains, the analog of convexity is considered to be *submodularity*, and the evolving theory of submodular optimization has been a catalyst for progress in extraordinarily varied application areas. A set function $f : 2^N \to \mathbb{R}$ defined on subsets S of a ground set N of elements is *submodular* if for all $S \subseteq T \subseteq N$ and $a \notin T$, it holds that:

$$f(S \cup a) - f(S) \geq f(T \cup a) - f(T).$$

This class of functions is known to have many desirable approximation guarantees (see e.g. [12] and references therein) and their property of *diminishing returns* makes submodular functions an appealing model in a variety of application areas. From a learnability perspective, Balcan and Harvey show that a broad class of submodular functions is PMAC-learnable under mild assumptions on product distributions. Since their work there has been a great deal of interest in the learnability of submodular functions [1, 8, 3, 24, 16, 14, 13, 15]. Their approximation properties together with the learnability guarantees make submodular functions an ideal class of functions to consider for studying optimization from samples.

1.2 Our Results

For our results, the constraint $\mathcal{M}$ is a cardinality constraint, i.e., sets of size at most k, and the distribution $\mathcal{D}$ is the uniform distribution over all feasible sets. Our main result is for coverage functions, which are a canonical subclass of monotone submodular functions. We show that coverage functions cannot be $n^{-1/4+\epsilon}$-optimized from samples. This means in a strong sense that learnability and approximability do not imply optimizability from samples since coverage functions can be approximated to within $1 - 1/e$ via the greedy algorithm and they are PMAC-learnable (under any distribution) to within an arbitrary $(1 + \epsilon)$-approximation factor [1]. Coverage functions have applications to facility locations, privacy, and mechanism design ([13] and references therein). In addition, coverage functions capture most models of influence maximization in social networks, as in the example above. Our construction for this information-theoretic lower bound uses a convex combinations of a new class of coverage functions.

Theorem 1. *For every constant $\epsilon > 0$, there exists a hypothesis class of coverage functions that is not $n^{-1/4+\epsilon}$-optimizable from samples.*

For general monotone submodular functions, we show an even stronger bound on the factor of inapproximability for optimization from samples, $n^{-1/3+\epsilon}$, and we also show a near-matching $\tilde{\Omega}\left(n^{-1/3}\right)$-optimization from samples algorithm. The main ingredient of this algorithm is to estimate the expected marginal contribution of an element to a random set. Note that under product distributions, monotone submodular functions are also PMAC-learnable to within arbitrary approximation factor [14], and they also admit a $(1 - 1/e)$-approximation algorithm [23]. Furthermore, our lower bound holds for the special case of OXS valuations (which are a special case of Gross Substitutes). Like coverage functions, OXS valuations are PMAC-learnable on any (not necessarily product) distribution [1].

Theorem 2. *For monotone submodular functions, there exists an $\tilde{\Omega}(n^{-1/3})$-optimization from samples algorithm. Furthermore, no algorithm can do better than $n^{-1/3+\epsilon}$ for any constant $\epsilon > 0$.*

For the more general class of monotone subadditive functions, a known result in the value query model [21] implies $n^{-1/2+\epsilon}$-inapproximability and we show a near-matching $n^{-1/2}$-algorithm. We also give positive results: additive and unit-demand functions are optimizable from samples to within arbitrarily good factors. The result for unit-demand is particularly interesting because it separates the problem of optimizing from samples and the problem of *recoverability* which we introduce and which is similar to learning the function everywhere [17]. Finally, we extend our model to continuous optimization, where we consider an additive approximation notion of optimization from samples. In this regime, we give near-optimal impossibility results for the problems of maximizing concave functions and minimizing convex functions under a linear constraint.

1.3 Additional Related Work

The discrepancy between the model on which algorithms optimize and the true state of nature has recently been studied in algorithmic mechanism design. Most closely related to our work are several recent papers (e.g. [9, 11, 6, 19, 22, 7]) that also consider models that bypass the learning algorithm, and let the mechanism designer access samples from a distribution rather than an explicit Bayesian prior. In contrast to our negative conclusions, these papers achieve mostly positive results. In particular, Huang et al [19] show that the obtainable revenue is much closer to the optimum than the information-theoretic bound on learning the valuation distribution.

2 The model

2.1 Preliminaries

Approximability. To quantify the quality of a decision, we will consider the notion of approximation, primarily in the context of maximization problems. For a function $f : 2^N \to \mathbb{R}$ and family of sets $\mathcal{M}$, we say that the optimization problem is α-*approximable* if there exists an algorithm which runs in time polynomial in n and returns a set $S \in \mathcal{M}$ for which $f(S) \geq \alpha \max_{T \in \mathcal{M}} f(T)$. If the function is exponentially expressive we assume we have a *value oracle* which given a set S, returns $f(S)$ in polynomial time. For such functions, α-*approximable* means that the algorithm uses polynomially-many oracle calls.

Learnability. As a model for statistical learnability we use the notion of PAC learnability due to Valiant [25] and its generalization to real-valued functions PMAC learnability, due to Balcan and Harvey [2]. Let $\mathcal{F}$ be a hypothesis class of functions $\{f_1, f_2, \ldots\}$ where $f_i : 2^N \to \mathbb{R}$. Given precision parameters $\epsilon > 0$ and $\delta > 0$, the input to a learning algorithm is samples $\{S_i, f(S_i)\}_{i=1}^t$ where the S_i's are drawn i.i.d. from from some distribution $\mathcal{D}$, and the number of samples t is polynomial in $1/\epsilon, 1/\delta$ and n. The learning algorithm outputs a function $\widetilde{f} : 2^N \to \mathbb{R}$ that should approximate f in the following sense.

- $\mathcal{F}$ is PAC-learnable on distribution $\mathcal{D}$ if there exists a (not necessarily polynomial time) learning algorithm such that for every $\epsilon, \delta > 0$:

$$\Pr_{S_1,\ldots,S_t \sim \mathcal{D}} \left[\Pr_{S \sim \mathcal{D}} \left[\widetilde{f}(S) \neq f(S) \right] \geq 1 - \epsilon \right] \geq 1 - \delta$$

- $\mathcal{F}$ is α-PMAC-learnable on distribution $\mathcal{D}$ if there exists a (not necessarily polynomial time) learning algorithm such that for every $\epsilon, \delta > 0$:

$$\Pr_{S_1,\ldots,S_t \sim \mathcal{D}} \left[\Pr_{S \sim \mathcal{D}} \left[\widetilde{f}(S) \leq f(S) \leq \alpha \cdot \widetilde{f}(S) \right] \geq 1 - \epsilon \right] \geq 1 - \delta$$

A class $\mathcal{F}$ is PAC (or α-PMAC) learnable if it is PAC- (α-PMAC)-learnable on every distribution $\mathcal{D}$.

Submodularity. For the purpose of our discussion, we are interested in models that are both *learnable* and *approximable*. In particular, we will focus on the class of monotone submodular functions. A function $f : 2^N \to \mathbb{R}$ over a ground set N is *submodular* if for any two sets $S, T \subseteq N$ we have that $f(S \cup T) \leq f(S) + f(T) - f(S \cap T)$. The function is *monotone* if $S \subseteq T$ implies $f(S) \leq f(T)$. Submodular functions are well known to have many desirable approximation guarantees. In particular, it is well known that for any matroid $\mathcal{M}$ the problem of $\max_{S \in \mathcal{M}} f(S)$ is approximable as there exists a $1 - 1/e$-approximation [5] which is optimal using a polynomial number of queries. From a learnability perspective, Balcan and Harvey show that a broad class of submodular functions are $O(1)$-`PMAC`-learnable under mild conditions on product distribution that generates the samples.

2.2 Optimization from Samples (`OPS`)

Let $\mathcal{F}$ be a hypothesis class of functions $\{f_1, f_2, \ldots\}$ where $f_i : 2^N \to \mathbb{R}$, and let $\mathcal{D}$ be a distribution over sets in a given matroid $\mathcal{M}$. The input to an *optimization* algorithm under constraint $\mathcal{M}$ is samples $\{S_i, f(S_i)\}_{i=1}^t$ where $f \in \mathcal{F}$, $S_i \in \mathcal{M}$ is drawn i.i.d. from $\mathcal{D}$, and the number of samples t is polynomial in $1/\delta$ and n. We say that the hypothesis class $\mathcal{F}$ is α-**optimizable from samples** on distribution $\mathcal{D}$ if there exists a (not necessarily polynomial time) optimization algorithm which finds a set S such that for every $\delta > 0$:

$$\Pr_{S_1, \ldots, S_t \sim \mathcal{D}} \left[\mathbf{E}[f(S)] \geq \alpha \max_{T \in \mathcal{M}} f(T) \right] \geq 1 - \delta,$$

where the expectation is over the randomness of the algorithm. Since our main interest is in showing a lower bound, we focus on a simplified special case where $\mathcal{M}$ is a uniform matroid (all sets of size at most k) and the distribution $\mathcal{D}$ from which the sets are drawn from is simply the uniform distribution of all sets of size at most k.

Discussion about the model. Before we continue to discuss the impossibility results it would be useful to justify our modeling decision in which we insist on having the sets sampled *uniformly at random* from the *same family of sets* we aim to optimize over. We emphasize the reasoning through the following points.

- First, we argue that it makes sense to fix *s*ome distribution. We do this to avoid trivialities in our model. Clearly, one cannot hope to optimize a function from *e*very distribution: consider for example the degenerate distribution which simply returns the empty set on every sample. Similarly, it is not so interesting to ask about optimizing from *a*ny distribution: this is trivial when the distribution always returns the optimal set.

- Once we agreed that we should fix some distribution, two natural choices come to mind: the uniform distribution over all feasible sets, and the uniform distribution over all sets. We argue that the former is more interesting (in what context would you observe the values of infeasible solutions?). Nevertheless, we emphasize the robustness of our model by showing a simple impossibility result for the uniform distribution over all sets in the appendix.

3 Main Result: Lower Bound for Coverage Functions

A function $f : 2^N \to \mathbb{R}$ is a *coverage* function if there exists a bipartite graph divided between *parents* $\{e_1, \cdots, e_n\} = N$ and *children* $\{u_1, \cdots, u_l\}$ with value $v(u_i) \in \mathbb{R}^+$ such that $f(S) = \sum_{u_i \in \mathcal{N}(S)} v(u_i)$ where $\mathcal{N}(S)$ denotes the neighborhood of the set S of parents. Coverage functions are an important subclass of submodular functions with a simple structure which seems to indicate that they might be optimizable from samples. Our main result shows that this is not the case, there exists coverage functions with strong impossibility results for optimization from samples.

Theorem 3. *For every constant $\epsilon > 0$, there exists a hypothesis class of coverage functions that is not $n^{-1/4+\epsilon}$-optimizable from samples.*

We start by giving a, rather conceptual, general construction to obtain impossibility results. We show that given two functions that satisfy certain properties, it is possible to construct a hypothesis class of functions over which optimization from samples is hard. Then, we provide a technical construction for such functions.

3.1 A Framework for Impossibility Results

We wish to obtain two functions, called *good* and *bad* and denoted by $g(\cdot)$ and $b(\cdot)$, such that small sets of equal size have equal value on both $g(\cdot)$ and $b(\cdot)$, and such that $g(\cdot)$ has much larger value than $b(\cdot)$ over large sets. We then construct a collection of functions, using these good and bad functions, that with high probability cannot be distinguished given access to random samples, but that have very different maximizers.

Consider parameters $k, m > 0$ such that $m \cdot k < n$. We build a collection of m functions $f_i(\cdot)$. First, there are $n - m \cdot k$ elements that have a marginal contribution of 0 to any solution on any f_i, which are called the zero elements. Partition the remaining elements into m sets $T_1, \cdots, T_m$, each of size k. For function f_i, the elements in T_i are called the *good* elements; the value of a subset $S_i \subset T_i$ is $f_i(S_i) = g(S_i)$. The elements in T_j, for all $j \neq i$ are called the *bad* elements; the value of a subset $S_j \subset T_j$ is $f_i(S_j) = b(S_j)$. This construction is given formally in Definition 1.

Definition 1. *Given two functions $g(\cdot)$ and $b(\cdot)$ and a collection $\{T_i\}_{i=1}^m$ of m disjoint sets of size k, the hypothesis class of functions $\mathcal{F}(g, b)$ is defined to be $\mathcal{F}(g, b) := \{f_1, \cdots, f_m\}$ with*

$$f_j(S) := g(S \cap T_j) + \sum_{i=1, i\neq j}^{m} b(S \cap T_i).$$

The desired properties for the good and bad functions are formally given below. We consider functions $b(\cdot)$ and $g(\cdot)$ that are symmetric, i.e., functions that have the same value on subsets of same size. We abuse the notation and denote by $b(y)$ and $g(y)$ the value of a set of size y.

Definition 2. *Two functions $g(\cdot)$ and $b(\cdot)$ have an (α, β, k, m)-gap if*

- ***Identical on small sets.*** *$g(y) = b(y)$ for all $0 \leq y \leq \log\log n$.*
- ***Gap α between good and bad function.*** *$g(k/m) \geq \alpha \cdot b(k/m)$.*
- ***Curvature β of good function for large values.*** *$\frac{g(k)}{g(k/m)} = (1 - \beta)m$*

The following concentration bound gives a simple condition so that the intersection of a random sample with any set T_i is of size at most $\log\log n$ with high probability. The "identical on small sets property" then implies that good and bad elements cannot be differentiated from samples.

Lemma 1. *Let $T \subseteq N$ and S be sampled uniformly from all sets of size at most k. If $k \cdot |T| \leq n^{1-\epsilon}$ for some constant $\epsilon > 0$, then $\Pr(|S \cap T| \geq \log\log n) = n^{-\Omega(\log\log n)}$.*

Proof. We start by considering a subset L of T of size $\log\log n$. We first bound the probability that L is a subset of a sample S,

$$\Pr(L \subseteq S) \leq \prod_{e\in L} \Pr(e \in S) \leq \prod_{e\in L} \frac{k}{n} = \left(\frac{k}{n}\right)^{\log\log n}.$$

We then bound the probability that $|S \cap T| > \log\log n$ with a union bound over the events that a set L is a subset of S, for all subsets L of T of size $\log\log n$:

$$\begin{aligned}
\Pr(|S\cap T| > \log\log n) &\leq \sum_{L\subseteq T\,:\,|L|=\log\log n} \Pr(L \subseteq S) \\
&\leq \binom{|T|}{\log\log n}\left(\frac{k}{n}\right)^{\log\log n} \\
&\leq \left(\frac{k\cdot|T|}{n}\right)^{\log\log n} \\
&\leq n^{-\epsilon\log\log n}
\end{aligned}$$

where the last inequality follows from the assumption that $k \cdot |T| \leq n^{1-\epsilon}$. □

The following result gives an impossibility result for optimizing $\mathcal{F}(g, b)$ from samples in terms of the (α, β, k, m)-gap of $g(\cdot)$ and $b(\cdot)$.

Theorem 4. *Let $g(\cdot)$ and $b(\cdot)$ be two submodular functions with an (α, β, k, m)-gap for $0 < m < k < n^{1/2-\Omega(1)}$. Then the class $\mathcal{F}(g, b)$ defined above is not $\big((1-\beta)\cdot\min\{\alpha, m\}/2\big)^{-1}$-optimizable from samples.*

Proof. Choose $f_r(\cdot) \in \mathcal{F}(g, b)$ u.a.r. to be optimized. Consider an algorithm that observes polynomially many samples drawn uniformly from the collection of feasible solutions. We show that it holds with high probability over the samples that the value of f_r on those samples is independent from r. In particular, this means that the algorithm can learn no information about r.

By Lemma 1, every sample S has intersection at most $\log\log n$ with each of the T_i's, except with probability $n^{-\Omega(\log\log n)}$. By a union bound, the same holds simultaneously for all polynomially many samples and all the T_i's. Therefore, with probability at least $1 - n^{-\Omega(\log\log n)}$, the algorithm's output is independent of r; we henceforth assume this is the case. Let S be a set returned by any algorithm. Since S is independent of r, we have $\mathbf{E}_r[|S \cap T_r|] \leq k/m$. Thus,

$$\begin{aligned}
\mathbf{E}_r[f_r(S)] &= \mathbf{E}_r\left[g(S\cap T_r) + \sum_{i\neq r} b(S\cap T_i)\right] \\
&\leq g(k/m) + m\cdot b(k/m) && \text{(submodularity)} \\
&\leq g(k/m) + \frac{m}{\alpha}\cdot g(k/m) && \text{(gap } \alpha \text{ between } g(\cdot) \text{ and } b(\cdot)) \\
&\leq \left(1+\frac{m}{\alpha}\right)\frac{1}{(1-\beta)m} g(k) && \text{(curvature } \beta) \\
&\leq \frac{2}{\min(\alpha, m)}\frac{1}{1-\beta} f_r(T_r).
\end{aligned}$$

□

3.2 Constructing the Good and the Bad Functions

The main difficulty with coverage functions is to construct a good and a bad function that combine the “identical on small sets” and a large “gap α” properties. In particular, in order for the bad function to increase slowly (or not at all) for large sets, there has to be a non-trivial interaction between elements even on small sets (this is related to cover functions being *second-order supermodular* [20]). Our main result (Theorem 3) follows from the next lemma combined with Theorem 4.

Lemma 2. *For every constant $\epsilon > 0$, there exists coverage functions $b(\cdot)$ and $g(\cdot)$ that admit an $(\alpha = n^{1/4-\epsilon}, \beta = o(1), k = n^{1/2-\epsilon}, m = n^{1/4-\epsilon})$-gap.*

The rest of this section is devoted to the proof of Lemma 2. We construct a good and a bad coverage function that are convex combinations of symmetric coverage functions C_{1/t_j} defined below, and the symmetric additive function. The weights are obtained by solving a system of $\log\log n$ linear equations $Mx = y$ where M is a square matrix with column j corresponding to a symmetric coverage function C_{1/t_j}, and y is the symmetric additive function. The $\log\log n$ rows correspond to the $\log\log n$ constraints to obtain the “identical on small sets” property.

Let $x^\star$ be the solution to this system of linear equations. The bad function is then the sum of the weighted coverage functions C_{1/t_j} that have positive weight, and the good function is the sum of the additive function and the weighted coverage functions with negative weights. Formally, the bad function is the symmetric coverage function such that a set of cardinality y has value:

$$b(y) := \sum_{j\,:\,x_j^\star>0} x_j^\star C_{1/t_j}(y);$$

and the good function is

$$g(y) := y + \sum_{j\,:\,x_j^\star<0} (-x_j^\star) C_{1/t_j}(y).$$

The crucial difference between these two functions is the symmetric additive term y in the good function. At a high level, the three properties required for the (α, β, k, m)-gap are satisfied for the following reasons.

- The **identical on small sets** property is guaranteed by the system of linear equations.
- The **gap $\alpha = n^{1/4-\epsilon}$ between good and bad function** is satisfied by constructing functions $C_{1/t_j}(y)$ that are much smaller than y for large y, and by bounding the coefficients $x_j^\star$.
- The **curvature $\beta = o(1)$ of good function for large values** is satisfied similarly by showing that for large y, $\sum_{j\,:\,x_j^\star<0}(-x_j^\star)C_{1/t_j}(y)$ is much smaller than y, which has curvature 0.

Additionally, the coverage functions C_{1/t_j} also need to be symmetric and be such that M is invertible. These coverage functions are in the following family of functions.

Definition 3. *The cover function $C_{1/t}$ for $t \geq 1$ over parents N is such that*

- *for each set S of parents, there is a child u_S that is covered by exactly S,*
- *child u_S has value $v(u_S) = t \cdot \Pr(S \sim \mathrm{B}(N, 1/t))$ where the binomial distribution $\mathrm{B}(N, 1/t)$ picks each parent $e_i \in N$ independently with probability $1/t$.*

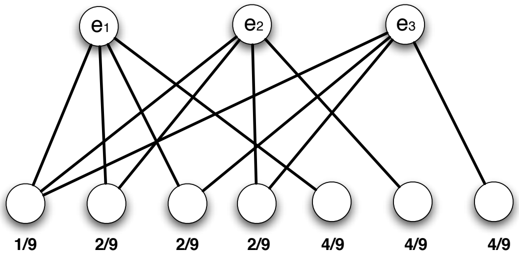


Figure 2: The coverage functions $C_{1/3}$ with $N = \{e_1, e_2, e_3\}$.

As an example, Figure 2 illustrates cover function $C_{1/3}$ with 3 parents. We make a few observations about coverage functions $C_{1/t}$. First, $C_{1/t}(y)$ is upper bounded by t for all positive integer y. The value of a single parent $C_{1/t}(\{e\})$ is the sum of all children u_S such that $e \in S$, so the value of a single parent is $t \cdot \Pr_{S \sim \mathrm{B}(N,1/t)}[e \in S] = t \cdot 1/t = 1$. The marginal contribution of adding a parent e to a set S of parents is the probability that a set drawn from $\mathrm{B}(N, 1/t)$ contains e and does not contain any element in S, multiplied by t. For small t, this value decreases quickly as $|S|$ increases. More precisely, the value $C_{1/t}(y)$ of a set of size y has a nice closed form given by the following claim and is illustrated in Figure 3.

Claim 1. *The cover function $C_{1/t}$ is a symmetric function where any set of size y has value*

$$C_{1/t}(y) = t \cdot (1 - (1 - 1/t)^y) .$$

Proof. Let S be a set of size y, then

$$\begin{aligned}
C_{1/t}(S) &= \sum_{T : |T \cap S| \geq 1} v(u_T) \\
&= t \cdot \sum_{T : |T \cap S| \geq 1} \Pr(T \sim \mathrm{B}(N, 1/t)) \\
&= t \cdot \left(1 - \sum_{T : |T \cap S| = 0} \Pr(T \sim \mathrm{B}(N, 1/t))\right) \\
&= t \cdot \left(1 - \Pr_{T \sim \mathrm{B}(N,1/t)}[|T \cap S| = 0]\right) \\
&= t \cdot \left(1 - \left(1 - \frac{1}{t}\right)^y\right) .
\end{aligned}$$

□

Note that these values are independent of n. Using the previous claim, we can formally define matrix M with the value of each of its entries.

Definition 4. *Let $M(\{t_j\}_{j=1}^{\ell})$ be the matrix with entries $M_{i,j} = C_{1/t_j}(i) = t_j \cdot \left(1 - (1 - 1/t_j)^i\right)$, i.e., the matrix where entry (i, j) corresponds to the value of a set of cardinality i in coverage function C_{1/t_j}.*

Next, we show that for any ℓ, there exists bounded $\{t_j\}_{j=1}^{\ell}$ such that $M(\{t_j\}_{j=1}^{\ell})$ is invertible.

Lemma 3. *Matrix $M(\{t_j\}_{j=1}^{\ell})$ is invertible for some set of integers $\{t_j\}_{j=1}^{\ell}$ such that $j \leq t_j \leq j(j + 1)$ for all $1 \leq j \leq \ell$.*

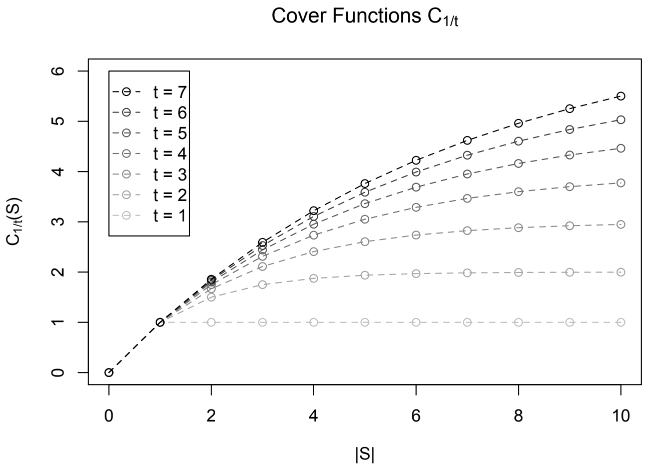


Figure 3: The value of coverage functions $C_{1/t}(S)$ for $1 \leq t \leq 7$ and sets S of size at most 10.

Proof. The proof goes by induction on ℓ and shows that it is possible to pick t_ℓ such that the rows of $M(\{t_j\}_{j=1}^{\ell})$ are linearly independent. The base case is trivial. In the inductive step, assume $t_1, \cdots, t_{\ell-1}$ have been picked so that the $(\ell-1) \times (\ell-1)$ matrix $M(\{t_j\}_{j=1}^{\ell-1})$ is invertible. We show that for some choice of integer $t_\ell \in [t_{\ell-1}, \ell(\ell+1)]$ there does not exist a vector z such that $\sum_{i \leq \ell} z_i M_{i,j} = 0$ for all $j \leq \ell$ where $M = M(\{t_j\}_{j=1}^{\ell})$. We write the first $\ell - 1$ entries of row M_ℓ as a linear combination of the other $\ell - 1$ rows:

$$\sum_{i<\ell} z_i M_{i,j} = M_{\ell,j} \quad \forall j < \ell.$$

Since $M(\{t_j\}_{j=1}^{\ell-1})$ is invertible by the inductive hypothesis, there exists a unique solution $z^\star$ to the above system of linear equations. It remains to show that $\sum_{i<\ell} z_i^\star M_{i,\ell} \neq M_{\ell,\ell}$, which by the uniqueness of $z^\star$ implies that there does not exist a vector z such that $\sum_{i \leq \ell} z_i M_{i,j} = 0$ for all $j \leq \ell$.

Observe that $M_{\ell,\ell} + \sum_{i<\ell} z_i^\star M_{i,\ell} = (t_\ell^\ell - (t_\ell - 1)^\ell + \sum_{i<\ell} z_i^\star (t_\ell^i - (t_\ell - 1)^i) t_\ell^{\ell-i}) / t_\ell^{\ell-1}$ and that $t_\ell^\ell - (t_\ell - 1)^\ell + \sum_{i<\ell} z_i^\star (t_\ell^i - (t_\ell - 1)^i) t_\ell^{\ell-i}$ is a non-zero polynomial of degree ℓ that has at most ℓ roots. Therefore, there exists t_ℓ such that $t_{\ell-1} < t_\ell \leq t_{\ell-1} + \ell + 1$ and $\sum_{i<\ell} z_i^\star M_{i,\ell} \neq M_{\ell,\ell}$. So the rows of $M(\{t_j\}_{j=1}^{\ell})$ are linearly independent and the matrix is invertible. We get the bounds on t_ℓ by the induction hypothesis, $t_\ell \leq t_{\ell-1} + \ell + 1 \leq (\ell-1)\ell + \ell + 1 \leq \ell(\ell+1)$. □

We now denote by $\{t_j\}_{j=1}^{\ell}$ a set of integers such that $M(\{t_j\}_{j=1}^{\ell})$ is invertible, which exists by Lemma 3. We bound the entries of the coefficients $x^\star$ to obtain the gap and curvature properties.

Lemma 4. *Let $x^\star$ be the solution to the system of linear equations $\left(M(\{t_j\}_{j=1}^{\ell})\right) x^\star = [1, \cdots, \ell]$, then $|x_i^\star| \leq \ell^{O(\ell^4)}$.*

Proof. Denote $M := M(\{t_j\}_{j=1}^{\ell})$. By Lemma 3, M is invertible, so let $x^\star = (M)^{-1}[1, \cdots, \ell]$. By Cramer's rule, $x_i^\star = \frac{\det M_i}{\det M}$, where M_i is M with the i-th column replaced by the vector y. Using the bound from Lemma 3, every entry in M can be represented as a rational number, with numerator and denominator bounded by $\ell^{O(\ell)}$. We can multiply by all the denominators, and get an integer matrix with positive entries bounded by $\ell^{O(\ell^3)}$. Now, by Hadamard's inequality, the determinants of the integral M and all the M_i's are integers bounded by $\ell^{O(\ell^4)}$. Therefore every entry in $x^\star$ can be written as a rational number with numerator and denominator bounded by $\ell^{O(\ell^4)}$. □

The two following lemmas for the gap and curvature properties complete the proof of Lemma 2.

Lemma 5. *The gap between the good and the bad functions $g(\cdot)$ and $b(\cdot)$ is at least $\alpha = n^{1/4-\epsilon}$.*

Proof. We show the gap between the good and the bad function on a set of size k/m. Recall that $b(k/m) = \sum_{j : x^\star_j > 0, j \leq \ell} x^\star_j C_{1/t_j}(k/m)$. We can bound each summand as:

$$\begin{aligned} x^\star_j C_{1/t_j}(k/m) &\leq x^\star_j t_j && \text{(Definition of } C_{1/t}\text{)} \\ &\leq x^\star_j \ell(\ell+1) && \text{(Lemma 3)} \\ &\leq \ell^{O(\ell^4)} && \text{(Lemma 4),} \end{aligned}$$

and therefore also $b(k/m) \leq \ell^{O(\ell^4)}$. On the other hand, the good function is bounded from below by the cardinality: $g(k/m) \geq k/m$. Plugging in $k = n^{1/2-\epsilon}$, $m = n^{1/4-\epsilon}$, and $\ell = \log\log n$, we get:

$$\frac{g(k/m)}{b(k/m)} \geq \frac{n^{1/4}}{(\log\log n)^{\log^4 \log n}} \gg n^{1/4-\epsilon} = \alpha.$$

□

Lemma 6. *The curvature of the good function for large value is $\beta = o(1)$.*

Proof. The curvature β is

$$\frac{g(k)}{g(k/m)} \geq \frac{k}{k/m + (\log\log n)^{\log^4 \log n}} \geq \frac{k}{(1+o(1))k/m} = (1-o(1))m,$$

where the first inequality follows a similar reasoning as the one used to upper bound $b(k/m)$ in Lemma 5. □

4 Tight Bounds for Submodular Functions

We improve the $n^{-1/4}$ impossibility result for coverage functions to $n^{-1/3}$ for submodular functions and, surprisingly, we show that this is almost tight.

4.1 Improving the Impossibility Result for Coverage Functions

First note that there exists a construction less involved than for coverage functions to obtain an $n^{-1/4+\epsilon}$ impossibility result for submodular functions using the framework from Section 3.1. Let the bad function be a budget-additive function with budget $\log\log n$, i.e., $b(y) = \min\{y, \log\log n\}$, and the additive function be an additive function, i.e., $g(y) = y$. These functions have an $(\alpha = n^{1/4-\epsilon}, \beta = 0, k = n^{1/2-\epsilon}, m = n^{1/4-\epsilon})$-gap, the properties for such a gap can easily be verified, which gives an $n^{-1/4+\epsilon}$ impossibility result for submodular functions by Theorem 4.

To improve to $n^{-1/3}$, we again construct a bad function that is budget-additive and a good function that is additive. The main difference is that the bad elements are not partitioned, so the value of the bad elements is $b(\cup_{i=1, i\neq j}^m S_i)$ instead of $\sum_{i=1, i\neq j}^m b(S_i)$. This is possible because the algorithm cannot even learn the partition (unlike in the coverage function case).

Theorem 5. *For every constant $\epsilon > 0$, there exists a hypothesis class of submodular functions that is not $n^{-1/3+\epsilon}$-optimizable from samples.*

Proof. As for coverage functions, the construction contains good, bad, and zero elements, but the bad elements are not partitioned. Fix the set T of good and bad elements of size $n^{2/3-\epsilon}$. The good elements are $n^{1/3}$ random elements from T, the remaining elements in T are bad elements, and the elements not in T are zero elements. Formally, the hypothesis class of function $\mathcal{F}$ contains function f^G, for any subset G of T of size $n^{1/3}$, with value

$$f^G(S) = |S \cap G| + \min\{|S \cap (T \setminus G)|, \log\log n\}.$$

Choose $f^G \in \mathcal{F}$ uniformly at random to be optimized under cardinality constraint $k = |G| = n^{1/3}$. By Lemma 1, a sample S_i has intersection of size at most $\log\log n$ with T, except with probability $n^{-\Omega(\log\log n)}$. By a union bound, the same holds simultaneously for all polynomially many samples. Therefore, with probability at least $1 - n^{-\Omega(\log\log n)}$, the algorithm's output is independent of G; we henceforth assume this is the case. Let S be a set returned by any algorithm. Since S is independent of G, we have $\mathbf{E}_G[|S \cap G|] \leq k \cdot |G|/|T| = n^{1/3} \cdot n^{1/3}/n^{2/3-\epsilon} = n^{\epsilon}$. Thus,

$$\mathbf{E}_G\left[f^G(S)\right] = \mathbf{E}_G[|S \cap G| + \min\{|S \cap (T \setminus G)|, \log\log n\}] \leq n^{\epsilon} + \log\log n$$

whereas the set of good elements is feasible and has value $n^{1/3}$. □

4.2 An Algorithm which Computes Expected Marginal Contributions

We provide an algorithm which achieves an $\tilde{\Omega}(n^{-1/3})$-optimization from samples for submodular functions. The algorithm is described formally below as Algorithm 2. Note that in the appendix, we give a simple $n^{-1/2}$-optimization from samples algorithm for subadditive functions. To improve from subadditive functions, we compute arbitrarily good estimates $\hat{v}_i$ of the expected marginal contribution of an element to a random set of size $k-1$ (Algorithm 1). The true marginal contribution of an element to a random set of size $k-1$ is denoted by $v_i := \mathbf{E}_{S\,:\,|S|=k-1, i \notin S}[f_S(i)]$.

Algorithm 1 MargCont: partitions elements into bins according to their estimated expected marginal contributions to a random set of size $k-1$.

```
Input: $\mathcal{S} = \{S_i : (S_i, f(S_i)) \text{ is a sample})\}$
  $\mathcal{F}_i \leftarrow \{S_j \in \mathcal{S} : i \in S_j, |S_j| = k\}$
  $\mathcal{F}_{-i} \leftarrow \{S_j \in \mathcal{S} : i \not\in S_j, |S_j| = k-1\}$
  $\hat{v}_i \leftarrow \frac{1}{|\mathcal{F}_i|}\sum_{S_j \in \mathcal{F}_i} f(S_j) - \frac{1}{|\mathcal{F}_{-i}|}\sum_{S_j \in \mathcal{F}_{-i}} f(S_j)$
  $\hat{v}_{max} \leftarrow \max_i \hat{v}_i$
  $L \leftarrow \left(\hat{v}_{max}/2^0, \hat{v}_{max}/2^1, \cdots, \hat{v}_{max}/2^{3\log n}\right)$
  $B_j \leftarrow \{i : L[j-1] \leq \hat{v}_i < L[j]\}$
  return $(B_1, \cdots, B_{3\log n})$
```

Algorithm 2 An $\tilde{\Omega}(n^{-1/3})$-optimization from samples algorithm for submodular functions.

```
Input: $\mathcal{S} = \{S_i : (S_i, f(S_i)) \text{ is a sample})\}$
  With probability 1/3:
      return  $\text{argmax}_{S_i \in \mathcal{S}} f(S_i)$
  With probability 1/3:
      return  $S$, a uniformly random set of size $k$
  With probability 1/3:
      $(B_1, \cdots, B_{3\log n}) \leftarrow \textsc{MargCont}(\mathcal{S})$
      Pick $j \in \{1, 2, \cdots, 3\log n\}$ u.a.r.
      return  a uniformly random subset of $B_j$ of size $\min\{|B_j|, k\}$
```

Theorem 6. *Let $f(\cdot)$ be a submodular function, then Algorithm 2 is an $\tilde{\Omega}(n^{-1/3})$-optimization from samples algorithm.*

We show that Algorithm 2 either obtains a $1/k$-approximation with the sample of largest value, or an $\tilde{\Omega}(t/n)$-approximation with a random set, or an $\tilde{\Omega}(k/t)$-approximation by picking a random subset from a random bin. Since $(1/k) \cdot (t/n) \cdot (k/t) = 1/n$, at least one of $1/k, t/n$, or k/t is at least $1/n^{1/3}$, so we obtain an $\tilde{\Omega}(n^{-1/3})$-approximation. The following concentration bound shows that the expected marginal contributions of elements to a random set are estimated well.

Lemma 7. *Let f be a monotone subadditive function. Then, with high probability, the estimations $\hat{v}_i$ are ϵ-close, for any $\epsilon \geq f(N)/poly(n)$, to the true expected marginal contribution of element i to a random set S of size $k-1$, i.e.,*

$$|\hat{v}_i - v_i| \leq \epsilon.$$

Proof. We assume wlog that $k \leq n/2$ (otherwise, a random subset of size k is a 1/2-approximation). The size of a sample which is the most likely is k, so the probability that a sample is of size k is at least $2/n$. Since $\binom{n}{k-1} \geq \binom{n}{k}/n$, the probability that a sample is of size $k-1$ is at least $2/n^2$. A given element i has probability at least $1/n$ of being in a sample and probability at least 1/2 of not being in a sample. Therefore, to observe at least n^c samples of size k which contain i and at least n^c samples of size $k-1$ which do not contain i, n^{c+3} samples are sufficient with high probability.

We assumed that $\epsilon \geq f(N)/n^{c'}$ for some constant c', we pick $c \geq 2c'+1$. Then by Hoeffding's inequality (Lemma 14) with $m = n^c$ and $b = f(N)$,

$$\Pr\left(\left|\frac{1}{|\mathcal{F}_i|}\sum_{S_j \in \mathcal{F}_i} f(S_j) - \mathbf{E}_{S\,:\,|S|=k, i \in S}[f(S)]\right| \geq \epsilon/2\right) \leq 2e^{-2n^c(\epsilon/2)^2/f(N)^2} \leq 2e^{-n^c/(2n^{2c'})} \leq 2e^{-n/2}$$

and similarly,

$$\Pr\left(\left|\frac{1}{|\mathcal{F}_{-i}|}\sum_{S_j \in \mathcal{F}_{-i}} f(S_j) - \mathbf{E}_{S\,:\,|S|=k-1, i \not\in S}[f(S)]\right| \geq \epsilon/2\right) \leq 2e^{-n/2}.$$

The claim holds then with high probability since $\hat{v}_i = \frac{1}{|\mathcal{F}_i|}\sum_{S_j \in \mathcal{F}_i} f(S_j) - \frac{1}{|\mathcal{F}_{-i}|}\sum_{S_j \in \mathcal{F}_{-i}} f(S_j)$ and $v_i = \mathbf{E}_{S\,:\,|S|=k-1, i \not\in S}[f_S(i)] = \mathbf{E}_{S\,:\,|S|=k, i\in S}[f(S)] - \mathbf{E}_{S\,:\,|S|=k-1, i \not\in S}[f(S)]$.

□

Let B_q be the bin that contains the largest value from the optimal solution denoted by $S^\star$, i.e., $q = \text{argmax}_j f(S^\star \cap B_j)$, we call this bin the optimal bin. We also define $S^\star_q := S^\star \cap B_q$ and denote by t the number of elements in bin B_q. For simplicity, we denote the expected value of a uniformly random set S of size k by $\mathbf{E}_S[f(S)]$.

Claim 2. *With the notation defined above, we have*

$$f(S_q^\star) \geq \Omega\left(\frac{1}{\log n}\right)\left(f(S^\star) - \mathbf{E}_S[f(S)]\right).$$

Proof. Let $S^\star_{>3\log n}$ be the set of elements in $S^\star$ that are not in any bin. Then, by subadditivity, $f(S_q^\star)$ is a $1/(3\log n)$ approximation to $f(S^\star \setminus S^\star_{>3\log n})$, and thus also a $1/(3\log n)$ approximation to $f(S^\star) - f(S^\star_{>3\log n})$. We upper bound $f(S^\star_{>3\log n})$ with the value of a random set of size k as follow

$$\begin{aligned}
f(S^\star_{>3\log n}) &\leq \mathbf{E}_S\left[f(S^\star_{>3\log n} \cup S)\right] && \text{(monotonicity)}\\
&\leq \mathbf{E}_S\left[f(S) + \sum_{i \in S^\star_{>3\log n}\setminus S} f_S(i)\right] && \text{(submodularity)}\\
&\leq \mathbf{E}_S\left[f(S) + \sum_{i \in S^\star_{>3\log n}\setminus S} v_i\right] && \text{(submodularity)}\\
&\leq \mathbf{E}_S\left[f(S) + \sum_{i \in S^\star_{>3\log n}\setminus S} \hat{v}_i\right] + O(f(N)/n^2) && \text{(Lemma 7)}\\
&\leq (1+o(1))\mathbf{E}_S[f(S)] + k \cdot \frac{\hat{v}_{max}}{n^3} && \text{(Not in a bin)}\\
&\leq (1+o(1))\mathbf{E}_S[f(S)]. && (\text{argmax}_i\, \hat{v}_i \in S \text{ w.p. at least } 1/k)
\end{aligned}$$

□

Lemma 8. *For any monotone subadditive function $f(\cdot)$, the sample S with the largest value among at least $(n/k)\log n$ samples is a $1/k$-approximation to $f(S^\star)$ with high probability.*

Proof. By subadditivity, there exists an element $i^\star$ such that $\{i^\star\}$ is a $1/k$-approximation to the optimal solution. By monotonicity, any set which contains $i^\star$ is a $1/k$-approximation to the optimal solution. After observing $(n/k)\log n$ samples, the probability of never observing a set that contains $i^\star$ is polynomially small. □

Lemma 9. *Let f be a monotone submodular function. Then a uniformly random subset of size k is an $\tilde{\Omega}(t/n)$-approximation to $f(S^\star)$.*

Proof. We show that a uniformly random subset is an $\Omega(t/n)$-approximation to $f(S_q^\star)$, and then the lemma follows by Claim 2. We first upper bound the value of $S_q^\star$,

$$\begin{aligned}
f(S_q^\star) &\leq \mathbf{E}_S\left[f(S_q^\star \cup S)\right] && \text{(monotonicity)}\\
&\leq \mathbf{E}_S\left[f(S) + \sum_{i \in S_q^\star \setminus S} f_S(i)\right] && \text{(submodularity)}\\
&\leq \mathbf{E}_S[f(S)] + \sum_{i \in S_q^\star} v_i && \text{(linearity of expectation)}\\
&\leq \mathbf{E}_S[f(S)] + \sum_{i \in S_q^\star} \hat{v}_i + O\left(f(N)/n^2\right) && \text{(Lemma 7)}\\
&\leq \mathbf{E}_S[f(S)] + kL[q] + O\left(f(N)/n^2\right) && \text{(Definition of } B_q)\\
&= \mathbf{E}_S[f(S)] \cdot (1+o(1)) + kL[q],
\end{aligned}$$

Next, we lower bound the expected value of random subset S of size k.

$$\begin{aligned}
\mathbf{E}_S[f(S)] &\geq \mathbf{E}_S\left[\sum_{i\in S} v_i\right] && \text{(submodularity)}\\
&\geq \mathbf{E}_S\left[\sum_{i\in S\cap B_q} \hat{v}_i\right] - O\left(f(N)/n^2\right) && \text{(Lemma 7)}\\
&\geq \mathbf{E}_S[|S\cap B_q|]\cdot L[q+1] - O\left(f(N)/n^2\right) && \text{(Definition of } B_q\text{)}\\
&= \frac{kt}{n}L[q+1] - O\left(f(N)/n^2\right) && (|S| = k \text{ and } |B_q| = t)\\
&= \frac{kt}{2n}L[q]\cdot(1-o(1))
\end{aligned}$$

By combining the lower bound of $\mathbf{E}_S[f(S)]$ and the upper bound of $f(S_q^\star)$, we get that

$$f(S_q^\star) \leq \mathbf{E}_S[f(S)]\cdot(1+o(1)) + kL[q] \leq O\left(n/t\right)\mathbf{E}_S[f(S)]\,.$$

□

The approximation obtained from a random set in a random bin follows from the next lemma, which will also be useful for other classes of functions.

Lemma 10. *For any monotone subadditive function $f(\cdot)$, a uniform random set S of size k is a k/n-approximation to $f(N)$.*

Proof. Partition the ground set into sets of size k uniformly at random. A uniform random set of this partition is a k/n-approximation to $f(N)$ in expectation by subadditivity. A uniform random set of this partition is also a uniform random set of size k. □

Corollary 1. *The better of a uniformly random feasible set, and a uniformly random subset of size $\min\{k, |B_j|\}$ of a random bin B_j, is an $\tilde{\Omega}(k/t)$-approximation to $f(S^\star)$ assuming that $t \geq k$.*

Proof. With probability $\Omega(1/\log n)$, the random bin is B_q. The expected value of a random subset of B_q of size k is a k/t-approximation to $f(B_q)$ by Lemma 10, so a k/t-approximation to $f(S_q^\star)$ by monotonicity. Finally, by Claim 2, $f(S_q^\star)$ is an $\Omega(1/\log n)$-approximation to $(f(S^\star) - \mathbf{E}_S[f(S)])$. □

5 Recoverability

The largely negative results from the above sections lead to the question of how well must a function be learned for it to be optimizable from samples? One extreme is a notion we refer to as recoverability (REC). A function is recoverable if it can be learned *everywhere* within an approximation of $1 \pm 1/n^2$ from samples. Does a function need to be learnable everywhere for it to be optimizable from samples?

Definition 5. *A function $f(\cdot)$ is recoverable for distribution $\mathcal{D}$ if there exists an algorithm which, given a polynomial number of samples drawn from $\mathcal{D}$, outputs a function $\tilde{f}(\cdot)$ such that for all sets S in the support of $\mathcal{D}$,*

$$\left(1-\frac{1}{n^2}\right)f(S) \leq \tilde{f}(S) \leq \left(1+\frac{1}{n^2}\right)f(S)$$

with high probability over the samples and the randomness of the algorithm.

This notion of recoverability is similar to the problem of approximating a function everywhere from Goemans et al. [17]. The differences are that recoverability is from samples whereas their setting allows value queries, and that recoverability requires to be within an approximation of $1 \pm 1/n^2$. It is important for us to be within such bounds and not within some arbitrarily small constant because such perturbations can still lead to an $O(n^{-1/2+\delta})$ impossiblity result for approximation [18]. We show that if a monotone submodular function $f(\cdot)$ is recoverable then it is optimizable from samples by using the greedy algorithm on the recovered function $\tilde{f}(\cdot)$. The proof is similar to the classical analysis of the greedy algorithm and is deferred to the appendix.

Theorem 7. *Let $\mathcal{D}$ be a distribution over feasible sets under a cardinality constraint. If a monotone submodular function $f(\cdot)$ is recoverable for $\mathcal{D}$, then it is $1 - 1/e - o(1)$-optimizable from samples from $\mathcal{D}$. For additive functions, it is $1 - o(1)$-optimizable from samples.*

We show that additive functions are in REC under some mild condition. A function $f(\cdot)$ is additive if there exists $v_1, \ldots, v_n$ such that for all subsets S, $f(S) = \sum_{i \in S} v_i$. The previous result implies that additive functions are optimizable from samples.

Lemma 11. *Let $f(\cdot)$ be an additive function with values $v_1, \ldots, v_n$ and with $v_{max} = \max_i v_i$, $v_{min} = \min_i v_i$ and let $\mathcal{D}$ be the uniform distribution over feasible sets under a cardinality constraint. If $v_{min} \geq v_{max}/poly(n)$, then $f(\cdot)$ is recoverable for $\mathcal{D}$.*

Proof. We have already shown that the expected marginal contribution of an element to a random set of size $k-1$ can be estimated from samples for submodular functions[2]. In the case of additive functions, this marginal contribution of an element is its value v_i.

We apply Lemma 7 with $\epsilon = v_i/n^2$, which satisfies $\epsilon \geq f(S^\star)/poly(n)$ since $v_{min} \geq v_{max}/poly(n)$, to compute $\hat{v}_i$ such that $|\hat{v}_i - v_i| \leq v_i/n^2$. Let $\tilde{f}(S) = \sum_{i \in S} \hat{v}_i$, then $\tilde{f}(S) \leq \sum_{i \in S}(1 + 1/n^2)v_i = (1 + 1/n^2)f(S)$ and $\tilde{f}(S) \geq \sum_{i \in S}(1 - 1/n^2)v_i = (1 - 1/n^2)f(S)$. □

Corollary 2. *Let $f(\cdot)$ be an additive function with values $v_1, \ldots, v_n$ and with $v_{max} = \max_i v_i$, $v_{min} = \min_i v_i$ and let $\mathcal{D}$ be the uniform distribution over feasible sets under a cardinality constraint. If $v_{min} \geq v_{max}/poly(n)$, then $f(\cdot)$ is $1 - o(1)$-optimizable from samples from $\mathcal{D}$.*

The previous results lead us to the question of whether a function needs to be recoverable to be optimizable from samples. We show that it is not the case since unit demand functions are optimizable from samples and not recoverable. A function $f(\cdot)$ is a unit demand function if $f(S) = \max_{i \in S} v_i$ for some $v_1, \ldots, v_n$.

Lemma 12. *Unit demand functions are not recoverable for $k \geq n^\epsilon$ but are 1-optimizable from samples.*

Proof. We first show that unit demand functions are not recoverable. Define a hypothesis class of functions $\mathcal{F}$ which contains n unit demand functions $f_j(\cdot)$ with $v_1 = j/n$ and $v_i = 1$ for $i \geq 2$, for all integers $1 \leq j \leq n$. We wish to recover function $f_j(\cdot)$ with j picked uniformly at random. With high probability, the sample $\{e_1\}$ is not observed when $k \geq n^\epsilon$, so the values of all observed samples are independent of j. Unit demand functions are therefore not recoverable.

Unit demand functions, on the other hand, are 1-optimizable from samples. With at least $n \log n$ samples, at least one sample contains, with high probability, the best element $e^\star := \text{argmax}_{e_i} v_i$.

[2] For simplicity, this proof uses estimations that we know how to compute. However, The values v_i can be recovered exactly by solving a system of linear equations where each row corresponds to a sample, provided that the matrix for this system is invertible, which is the case with a sufficently large number of samples by using results from random matrix theory such as in the survey by Blake and Studholme [4].

Any set containing the best element is an optimal solution. Therefore, an algorithm which returns the sample with largest value obtains an optimal solution with high probability. □

We conclude that functions do not need to be learnable everywhere to be optimizable from samples.

6 Limitations of Convex Optimization from Samples

In this section we translate our results to the world of continuous optimization. The analogue of submodularity in the continuous world is convexity which has numerous applications in machine learning. We show a similar lower bound for convex optimization as the ones from previous sections. This implies that continuous optimization is prone to the same vulnerabilities as combinatorial optimization when it comes to optimization from samples. Recall that a function f is *convex* if:

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y})$$

for all $\mathbf{x}, \mathbf{y} \in [0,1]^n$, and $0 \leq \alpha \leq 1$. Similarly a function is *concave* if for all $\mathbf{x}, \mathbf{y} \in [0,1]^n$, and $0 \leq \alpha \leq 1$ we have that:

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \geq \alpha f(\mathbf{x}) + (1-\alpha)f(\mathbf{y}).$$

Convex optimization from samples. As an analogy to the submodular case, we now consider optimization over the polytope $[0,1]^n$ under a linear constraint. The input is a set of samples $\{\mathbf{x_i}, f(\mathbf{x_i})\}_{i=1}^m$, where $f : [0,1]^n \to [0,1]$ and $\mathbf{x_i}$ is drawn from a distribution $\mathcal{D}$ over $\mathbf{x}$ in some matroid polytope $P(\mathcal{M})$ corresponding to some matroid $\mathcal{M}$. Our goal is to find $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in P(\mathcal{M})} f(\mathbf{x})$. We say that an optimization problem $(f, \mathcal{M})$ is **ϵ-absolute optimizable from samples** for a distribution $\mathcal{D}$ if there is an ϵ-additive approximation for the problem above. We focus on the the matroid polytope corresponding to uniform matroids, $P(\mathcal{M}) = \{\mathbf{x} : \sum_i x_i \leq k\}$ for some k.

The lower bound. The construction for submodular functions conveniently extends to concave function maximization and convex function minimization under a linear constraint.

Corollary 3. *There exist hypothesis classes of convex and concave functions that are not $1 - o(1)$-absolute optimizable from samples for convex minimization and concave maximization under a uniform matroid polytope.*

Proof. For concave function maximization, we extend the construction for submodular functions to vectors $\mathbf{x} \in [0,1]^n$ with $k = |G| = n^{1/3}$ and $|T| = n^{2/3+\epsilon}$ as follow

$$f^G(\mathbf{x}) = \frac{1}{n^{1/3} + 2n^\epsilon}\left(\sum_{i\in G} x_i + \min\{\sum_{i\in T\setminus G} x_i, 2n^\epsilon\}\right).$$

The functions $f^G(\cdot)$ are clearly concave. Observe that:

$$\mu := \mathbf{E}\left[\sum_{i\in T} x_i\right] \leq k \cdot \frac{|T|}{n} \leq n^\epsilon.$$

By Corollary 4 with constant $0 < \delta < 1$, $\sum_{x_i \in T} \leq 2n^{\epsilon}$ with high probability. So $f^G(\mathbf{x}) \leq \frac{2n^{\epsilon}}{n^{1/3}+2n^{\epsilon}}$ for all samples $\mathbf{x}$ and their values are independent of G with high probability. The remaining of the analysis follows similarly as for submodular functions, any algorithm has expected value at most $o(1)$ while the set G has value $1 - o(1)$.

For convex function minimization, define:

$$f^G_{cvx}(\mathbf{x}) = 1 - f^G(\mathbf{x}).$$

Then, by symmetry, any algorithm has expected value at least $1 - o(1)$ while G's value is $o(1)$. □

References

[1] Ashwinkumar Badanidiyuru, Shahar Dobzinski, Hu Fu, Robert Kleinberg, Noam Nisan, and Tim Roughgarden. Sketching valuation functions. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1025–1035, 2012. URL http://portal.acm.org/citation.cfm?id=2095197&CFID=63838676&CFTOKEN=79617016.

[2] Maria-Florina Balcan and Nicholas J. A. Harvey. Learning submodular functions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 793–802, 2011. doi: 10.1145/1993636.1993741. URL http://doi.acm.org/10.1145/1993636.1993741.

[3] Maria-Florina Balcan, Florin Constantin, Satoru Iwata, and Lei Wang. Learning valuation functions. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 4.1–4.24, 2012. URL http://www.jmlr.org/proceedings/papers/v23/balcan12b/balcan12b.pdf.

[4] Ian F Blake and Chris Studholme. Properties of random matrices and applications. *Unpublished report available at http://www. cs. toronto. edu/˜ cvs/coding*, 2006.

[5] Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. doi: 10.1137/080733991. URL http://dx.doi.org/10.1137/080733991.

[6] Shuchi Chawla, Jason D. Hartline, and Denis Nekipelov. Mechanism design for data science. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*, pages 711–712, 2014. doi: 10.1145/2600057.2602881. URL http://doi.acm.org/10.1145/2600057.2602881.

[7] Yu Cheng, Ho Yee Cheung, Shaddin Dughmi, Ehsan Emamjomeh-Zadeh, Li Han, and Shang-Hua Teng. Mixture selection, mechanism design, and signaling. In *FOCS*, 2015. To appear.

[8] Mahdi Cheraghchi, Adam Klivans, Pravesh Kothari, and Homin K. Lee. Submodular functions are noise stable. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1586–1592, 2012. URL http://portal.acm.org/citation.cfm?id=2095242&CFID=63838676&CFTOKEN=79617016.

[9] Richard Cole and Tim Roughgarden. The sample complexity of revenue maximization. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 243–252. ACM, 2014.

[10] Devdatt Dubhashi, Volker Priebe, and Desh Ranjan. Negative dependence through the fkg inequality. In *RESEARCH REPORT MPI-I-96-1-020, MAX-PLANCK-INSTITUT FUR INFORMATIK, SAARBRUCKEN*, 1996.

[11] Shaddin Dughmi, Li Han, and Noam Nisan. Sampling and representation complexity of revenue maximization. In *Web and Internet Economics - 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings*, pages 277–291, 2014. doi: 10.1007/978-3-319-13129-0_22. URL http://dx.doi.org/10.1007/978-3-319-13129-0_22.

[12] Moran Feldman. *Maximization Problems with Submodular Objective Functions*. PhD thesis, Technion - Israeli Institute of Technology, 2013. URL http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2013/PHD/PHD-2013-08.pdf.

[13] Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, pages 679–702, 2014. URL http://jmlr.org/proceedings/papers/v35/feldman14a.html.

[14] Vitaly Feldman and Jan Vondrák. Optimal bounds on approximation of submodular and XOS functions by juntas. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 227–236, 2013. doi: 10.1109/FOCS.2013.32. URL http://dx.doi.org/10.1109/FOCS.2013.32.

[15] Vitaly Feldman and Jan Vondrák. Tight bounds on low-degree spectral concentration of submodular and XOS functions. *CoRR*, abs/1504.03391, 2015. URL http://arxiv.org/abs/1504.03391.

[16] Vitaly Feldman, Pravesh Kothari, and Jan Vondrák. Representation, approximation and learning of submodular functions using low-rank decision trees. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, pages 711–740, 2013. URL http://jmlr.org/proceedings/papers/v30/Feldman13.html.

[17] Michel X Goemans, Nicholas JA Harvey, Satoru Iwata, and Vahab Mirrokni. Approximating submodular functions everywhere. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 535–544. Society for Industrial and Applied Mathematics, 2009.

[18] Avinatan Hassidim and Yaron Singer. Submodular optimization under noise. 2015. Working paper.

[19] Zhiyi Huang, Yishay Mansour, and Tim Roughgarden. Making the most of your samples. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 45–60, 2015. doi: 10.1145/2764468.2764475. URL http://doi.acm.org/10.1145/2764468.2764475.

[20] Nitish Korula, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 889–898, 2015. doi: 10.1145/2746539.2746626. URL http://doi.acm.org/10.1145/2746539.2746626.

[21] Vahab Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 70–77. ACM, 2008.

[22] Jamie Morgenstern and Tim Roughgarden. The pseudo-dimension of nearly-optimal auctions. In *NIPS*, page Forthcoming, 12 2015. URL papers/auction-pseudo.pdf.

[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions ii. *Math. Programming Study 8*, pages 73–87, 1978.

[24] Sofya Raskhodnikova and Grigory Yaroslavtsev. Learning pseudo-boolean k-dnf and submodular functions. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1356–1368, 2013. doi: 10.1137/1.9781611973105.98. URL http://dx.doi.org/10.1137/1.9781611973105.98.

[25] Leslie G. Valiant. A Theory of the Learnable. *Commun. ACM*, 27(11):1134–1142, 1984. doi: 10.1145/1968.1972. URL http://doi.acm.org/10.1145/1968.1972.

Appendix

A Concentration Bounds

Lemma 13 (Chernoff Bound)**.** *Let $X_1, \ldots, X_n$ be independent indicator random variables such that* $\Pr(X_i = 1) = 1$*. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[X]$. For $0 < \delta < 1$,*

$$\Pr(|X - \mu| \geq \delta\mu) \leq 2e^{-\mu\delta^2/3}.$$

Lemma 14 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_n$ be independent random variables with values in* $[0, b]$*. Let $X = \frac{1}{m}\sum_{i=1}^{m} X_i$ and $\mu = \mathbf{E}[X]$. Then for every $0 < \epsilon < 1$,*

$$\Pr\left(|\bar{X} - \mathbf{E}\left[\bar{X}\right]| \geq \epsilon\right) \leq 2e^{-2m\epsilon^2/b^2}.$$

Correlated variables

For sets of bounded size (which are the focus of this paper) the corresponding indicator variables are not exactly independent, so the standard Chernoff bound does not apply. There are many ways to go around this obstacle, the most convenient seems to be the following generalization of the concentration bound due to Dubhashi, Priebe, and Ranjan which combines the Chernoff bound with the FKG inequality. First, we require a definition:

Definition 6 (negative association [10])**.** *We say that random variables $X_1, \ldots, X_n$ are* negatively-associated *if for every $I \subset N$ and every two monotone non-decreasing functions $f : \mathbb{R}^I \to \mathbb{R}$ and $g : \mathbb{R}^{N \setminus I} \to \mathbb{R}$,*

$$\mathbf{E}[f(X_i, i \in I) g(X_j, j \notin I)] \leq \mathbf{E}[f(X_i, i \in I)]\, \mathbf{E}[g(X_j, j \notin I)]\,.$$

Theorem 8 (Negative-association Chernoff Bound [10, Theorem 14])**.** *Let $X_1, \ldots, X_n$ be negatively-associated random variables. Then for any $t > 0$,*

$$\mathbf{E}\left[\exp\left(t \sum_i X_i\right)\right] \leq \prod_i \mathbf{E}[\exp\left(tX_i\right)]$$

In order to apply this theorem to our case, we will use the following claim:

Claim 3. *Let $x_1, \ldots, x_n$ be fixed values in* $[0, 1]$*. Let $S \subset N$ be a uniformly random subset of size exactly ℓ, and let $X_i = \begin{cases} x_i & i \in S \\ 0 & otherwise \end{cases}$. Then $X_1, \ldots, X_n$ are negatively associated.*

Finally, we have the following concentration bound:

Corollary 4. *Let $X_1, \ldots, X_n$ be sampled uniformly from* $[0, 1]^n$*, conditioning on $\sum_i X_i \leq k$, and let $S \subset N$ be of size ℓ. Then for any $\delta \in [0, 1]$,*

$$\Pr_{X_1,\ldots,X_n \sim \mathcal{U}([0,1]^n)}\left(\sum_{i \in S} X_i \geq (1+\delta)k\ell/n \,\middle|\, \sum_{i=N} \leq k\right) \leq \exp\left(-\delta^2\mu/3\right)$$

Proof. By symmetry, we can first sample n values $x_1, \ldots, x_n$, and then assign k of them at random to elements in S. Thus by Claim 3, the values of the elements in S are negatively associated (so Theorem 8 holds), and their expected sum is $\ell k/n$. The bound 4 follows by standard manipulation of the Chernoff bound. □

B Linear Algebra

We recall a couple of basic results from linear algebra:

Theorem (Cramer's rule)**.** *Let M be an invertible matrix. The solution to the linear system $Mx = y$ is given by $x_i = \frac{\det M_i}{\det M}$, where M_i is the matrix M with the i-th column replaced by the vector y.*

Theorem (Hadamard's inequality)**.** $\det M \leq \prod \|v_i\|$, *where $\|v_i\|$ denotes the Euclidean norm of the i-th column of M.*

C Missing Discussion about the Model

We show a simple impossibility result for the uniform distribution over all sets, which motivates our choice of the uniform distribution over all feasible sets.

Theorem 9. *There exists a hypothesis class of submodular functions that is not $4n^{-1/2}$-optimizable from samples from the uniform distribution over* **all** *sets.*

Proof. Assume that samples are drawn uniformly from all sets of elements, or equivalently, from the product distribution with marginal probabilities 1/2. Consider the case where $k = n^{1/2}$ and the hypothesis class of functions $\mathcal{F}$ contains function $f^T(\cdot)$, for any set T of size $n^{1/2}$, defined as

$$f^T(S) = \min\left\{|S \cap T|, \frac{n^{1/2}}{4}\right\}.$$

We pick a set T of size $n^{1/2}$ uniformly at random. By Lemma 13 with X_i indicating if element e_i is in $S \cap T$, $\mu = n \cdot \Pr(e \in S) \Pr(e \in T) = n^{1/2}/2$, and constant $0 < \delta < 1$, $|S \cap T| \geq n^{1/2}/4$ with high probability, for all samples. So $f(S) = n^{1/2}/4$, which is independent from T, for all samples S with high probability. It follows that any algorithm is independent from T with high probability. An algorithm therefore picks in expectation, over the choice of T and the randomness of the algorithm, $n^{1/2} \cdot n^{1/2}/n = 1$ element in T. So for any algorithm, there is some function $f^T(\cdot) \in \mathcal{F}$ for which the algorithm outputs a set of expected value at most 1. The optimal solution is T and has value $n^{1/2}/4$. □

We also note that for $k \leq n/2$, and with n^c samples from the uniform distribution over all feasible sets under a cardinality constraint k, there are at least n^{c-1} samples of size exactly k in expectation. So the distribution over all feasible sets gives more information than the distribution over all sets of size **exactly** k.

D Tight Bounds for Subadditive Functions

A function $f : 2^N \rightarrow \mathbb{R}$ is *subadditive* if for any $S, T \subseteq N$ we have that $f(S \cup T) \leq f(S) + f(T)$. Since subadditive functions are a superclass of submodular functions, all the lower bounds apply to subadditive functions as well. Still, it seems natural to ask whether non-trivial guarantees are obtainable for subadditive functions.

Theorem 10. *For subadditive functions, there exists an $n^{-1/2}$-optimization from samples algorithm. Furthermore, no algorithm can do better than $n^{1/2-\epsilon}$ for any constant $\epsilon > 0$.*

Proof. In the value query model where $f(S)$ can be queried for any set S, subadditive functions cannot be $n^{-1/2+\epsilon}$-approximated for any constant $\epsilon > 0$ by a polynomial-time algorithm, which follows from [21]. Any algorithm for optimization from samples can trivially be extended for optimization in the value query model, so the $n^{1/2-\epsilon}$ lower bound for optimizing subadditive functions from samples follows.

For the upper bound, if $k \leq n^{1/2}$, we return the sample with largest value, which by Lemma 8 is a $1/k$-approximation. If $k \geq n^{1/2}$, then we return a random set of size k, which by Lemma 10 is a k/n-approximation to $f(N)$. By monotonicity, it is a k/n-approximation to the optimum. □

E Missing Proof from Section 5

We restate Theorem 7 for convenience.

Theorem 7. *Let $\mathcal{D}$ be a distribution over feasible sets under a cardinality constraint. If a monotone submodular function $f(\cdot)$ is recoverable for $\mathcal{D}$, then it is $1 - 1/e - o(1)$-optimizable from samples from $\mathcal{D}$. For additive functions, it is $1 - o(1)$-optimizable from samples.*

Proof. We show that the greedy algorithm with $\tilde{f}(\cdot)$ for a recoverable function performs well. The proof follows similarly as the classical analysis of the greedy algorithm. We start with submodular functions and denote by $S_i = \{e_1, \cdots, e_i\}$ the set obtained after the ith iteration. Let $S^\star$ be the optimal solution, then by submodularity,

$$\begin{aligned} f(S^\star) &\leq f(S_{i-1}) + \sum_{e \in S^\star \setminus S_{i-1}} f_{S_{i-1}}(e) \\ &\leq f(S_{i-1}) + \sum_{e \in S^\star \setminus S_{i-1}} \left(\left(\frac{1 + 1/n^2}{1 - 1/n^2} \right) f(S_i) - f(S_{i-1}) \right) \end{aligned}$$

where the second inequality follows from $\tilde{f}(S_i) \geq \tilde{f}(S_{i-1} \cup \{e\})$ for all $e \in S^\star \setminus S_{i-1}$ by the greedy algorithm, so $(1 + 1/n^2) f(S_i) \geq (1 - 1/n^2) f(S_{i-1} \cup \{e\})$. We therefore get that

$$f(S^\star) \leq (1 - k) f(S_{i-1}) + k \left(\frac{1 + 1/n^2}{1 - 1/n^2} \right) f(S_i).$$

By induction and similarly as in the analysis of the greedy algorithm, we then get that

$$f(S_k) \geq \left(\frac{1 - 1/n^2}{1 + 1/n^2} \right)^k \left(1 - (1 - 1/k)^k \right) f(S^\star).$$

Since

$$\left(\frac{1 - 1/n^2}{1 + 1/n^2} \right)^k \geq \left(1 - \frac{2}{n^2} \right)^k \geq 1 - 2k/n^2 \geq 1 - 2/n$$

and $\left(1 - (1 - 1/k)^k\right) \geq 1 - 1/e$, the greedy algorithm achieves an $(1 - 1/e - o(1))$-approximation for submodular functions.

For additive functions, let S be the set returned by the greedy algorithm and $\hat{v}_i = \tilde{f}(\{i\})$, then

$$f(S) = \sum_{i \in S} v_i \geq \left(\frac{1}{1 + 1/n^2} \right) \sum_{i \in S} \hat{v}_i \geq \left(\frac{1}{1 + 1/n^2} \right) \sum_{i \in S^\star} \hat{v}_i \geq \left(\frac{1 - 1/n^2}{1 + 1/n^2} \right) f(S^\star).$$

We therefore get a $(1 - o(1))$-approximation for additive functions.

□