Robust Guarantees of Stochastic Greedy Algorithms

Avinatan Hassidim [1] **Yaron Singer** [2]

Abstract

In this paper we analyze the robustness of stochastic variants of the greedy algorithm for submodular maximization. Our main result shows that for maximizing a monotone submodular function under a cardinality constraint, iteratively selecting an element whose marginal contribution is approximately maximal *in expectation* is a sufficient condition to obtain the optimal approximation guarantee *with exponentially high probability*, assuming the cardinality is sufficiently large. One consequence of our result is that the linear-time STOCHASTIC-GREEDY algorithm recently proposed in (Mirzasoleiman et al., 2015) achieves the optimal running time while maintaining an optimal approximation guarantee. We also show that high probability guarantees cannot be obtained for stochastic greedy algorithms under matroid constraints, and prove an approximation guarantee which holds in expectation. In contrast to the guarantees of the greedy algorithm, we show that the approximation ratio of stochastic local search is arbitrarily bad, with high probability, as well as in expectation.

1. Introduction

In this paper we study the guarantees of stochastic optimization algorithms for submodular maximization. A function $f : 2^N \to \mathbb{R}$ is submodular if it exhibits a diminishing returns property. That is, for any $S \subseteq T \subseteq N$ and any $a \notin T$, the function respects:

$$f_S(a) \geq f_T(a)$$

where $f_H(x)$ denotes the marginal contribution of an element $x \in N$ to a set $H \subseteq N$, i.e. $f_H(x) = f(H \cup x) - f(H)$. Many fundamental measures such as entropy, diversity, and clustering can be modeled as submodular functions, and as a result submodular optimization is heavily studied in machine learning for well over a decade now.

It is well known that for the problem of maximizing a monotone ($S \subseteq T \implies f(S) \leq f(T)$) submodular function under a cardinality constraint, the celebrated greedy algorithm which iteratively adds the element whose marginal contribution is maximal, obtains an approximation guarantee of $1 - 1/e$ (Nemhauser et al., 1978). This is optimal unless P=NP (Feige) or alternatively, assuming polynomially-many function evaluations (Nemhauser & Wolsey, 1978).

In recent years, there have been various adaptations of the classic greedy algorithm to allow for scalable, distributed, and noise-resilient optimization. Most notably, the STOCHASTIC-GREEDY algorithm recently proposed by (Mirzasoleiman et al., 2015) is a linear-time algorithm which at every step takes an element, which approximates in expectation the element with the maximal marginal contribution. Mirzasoleiman et al. show that STOCHASTIC-GREEDY gives an approximation guarantee which is arbitrarily close to $1 - 1/e$ in expectation, and does very well in practice (Mirzasoleiman et al., 2015; Lucic et al., 2016). Variants of this algorithm are used in clustering (Malioutov et al., 2016), sparsification (Lindgren et al., 2016), Gaussian RBF kernels (Sharma et al., 2015), sensing (Li et al., 2016), and social data analysis (Zhuang et al., 2016).

More generally, a greedy algorithm that iteratively adds elements that are only approximately maximal in expectation, may not necessarily be due to a design decision, but rather an artifact of its application on large and noisy data sets (see e.g. (Azaria et al., 2013)). One can model this uncertainty with a probability distribution $\mathcal{D}$: at every iteration, a value $\xi \sim \mathcal{D}$ is being sampled, and the greedy algorithm adds an element whose marginal contribution is a ξ-approximation to the maximal marginal contribution.

In general, we refer to an algorithm that iteratively adds an element whose marginal contribution is approximately optimal in expectation as a *stochastic greedy* algorithm. It is easy to show that stochastic greedy algorithms give an approximation ratio of $1 - 1/e^{\mu}$ in expectation, where μ is the mean of the distribution modeling the uncertainty. This however is a weak guarantee as it leaves a non-negligible

*Equal contribution [1]Bar Ilan University and Google [2]Harvard University. Correspondence to: Avinatan Hassidim <avinatan@cs.biu.ac.il>, Yaron Singer <yaron@seas.harvard.edu>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

likelihood that the algorithm terminates with a solution with a poor approximation guarantee. Indeed, as we later show, there are cases where stochastic greedy algorithms have desirable guarantees in expectation, but with constant probability have arbitrarily bad approximation guarantees.

Do stochastic optimization algorithms for submodular maximization have robust approximation guarantees?

1.1. Our results

We prove the following results:

- **Optimization under cardinality constraints.** For the problem of maximizing a monotone submodular function under cardinality constraint k, with uncertainty distribution $\mathcal{D}$ with expectation μ, we show that for any $\varepsilon \in [0, 1]$, when $k \geq \frac{1}{\mu\varepsilon^2}$ a stochastic greedy algorithm obtains an approximation of $1 - 1/e^{(1-\varepsilon)\mu}$ w.p. at least $1 - e^{-\mu k \varepsilon^2/2}$. Furthermore, we prove that this bound is optimal by showing that for any $\delta > 0$ no algorithm can obtain an approximation ratio better than $1 - 1/e^{(1-\varepsilon)\mu}$ w.p. $1 - \delta$. For the special case in which the function is *modular*, we prove an improved bound of $(1 - \varepsilon)\mu$ w.p. at least $1 - e^{-\mu k \varepsilon^2/2}$;

- **Optimization under matroid constraints.** To further study the difference between guarantees that occur in expectation and guarantees which appear w.h.p, we study a generalization, where the greedy algorithm is used to maximize a monotone submodular function under intersection of matroid constraints, where a distribution $\mathcal{D}$ with mean μ generates uncertainty. We show that in this case, with P matroids the algorithm obtains an approximation ratio of $\mu/(P+1)$ in expectation. However, we show that even for a single matroid no algorithm can give a finite approximate guarantee w.p. at least $1-\mu-o(1)$, implying that in general stochastic greedy algorithms cannot obtain high probability guarantees under general matroid constraints;

- **Stochastic local search.** Finally, a natural alternative to greedy is local search. We show that even for cardinality constraints local search performs poorly, and does not give any meaningful approximation guarantees when there is probabilistic uncertainty about the quality of the elements. We contrast this with the case where there is deterministic uncertainty about the quality of the elements, in which we get an approximation ratio of $(1 + 1/\mu)^{-1}$, where μ is our uncertainty. This implies that local search does not enjoy the same robustness guarantees of the greedy algorithms under uncertainty of the quality of elements.

1.2. Applications

The above results have several immediate consequences:

- **Fast algorithms for submodular optimization.** Our analysis applies to the STOCHASTIC-GREEDY algorithm (Mirzasoleiman et al., 2015), thus showing its approximation guarantee holds with high probability, implying it is the optimal algorithm in terms of running time and approximation guarantee for the problem of maximizing a submodular function under a cardinality constraint; [1] The same guarantee holds for the variants studied in (Malioutov et al., 2016; Lindgren et al., 2016; Sharma et al., 2015; Li et al., 2016);

- **Submodular optimization under noise.** In (Hassidim & Singer, 2017) the problem of maximization of submodular functions under a cardinality constraint is considered when given access to a noisy oracle. Our result simplifies the analysis of one of the algorithms in this setting, and gives high probability results for the inconsistent noise model (Singla et al., 2016).

1.3. Organization of the paper

We describe the results for general monotone submodular functions in Section 2, and the improved analysis for modular functions in Section 3. In Section 4 we consider the more general problem of maximizing a submodular function under matroid constraints, and show the inapproximabilities of stochastic local search algorithms in Section 5. Finally, we discuss experiments in Section 6.

2. Submodular Functions

In this section we analyze the stochastic greedy algorithm for general monotone submodular functions. We first analyze the algorithm, and then show the bound is tight.

2.1. Upper bound

For a given cardinality constraint k, the standard greedy algorithm begins with the empty set as its solution and at each step $\{1, \ldots, k\}$ adds the element whose marginal contribution to the existing solution is largest. In the stochastic version, the algorithm may no longer add the element whose marginal contribution is largest. Rather, the algorithm adds the element whose marginal contribution is at least a factor of ξ from the maximal marginal contribution, where ξ is drawn i.i.d from some distribution $\mathcal{D}$ with mean μ. We give a formal description below.

[1]Formally, the algorithm in (Mirzasoleiman et al., 2015) does not assume that the expected marginal contribution of the element selected is approximately optimal, but rather that in expectation its marginal contribution approximates that of some element in the optimal solution. Nevertheless our analysis still applies.

Algorithm 1 STOCHASTIC-GREEDY

input k
$S \leftarrow \emptyset$
while $|S| < k$ **do**
  $\xi \sim \mathcal{D}$
  $S \leftarrow S \cup$ arbitrary a s.t. $f_S(a) \geq \xi \max_{x \in N} f_S(x)$
end while
output S

The following lemma shows that when the sampled mean of the distribution is close to 1, stochastic greedy algorithms obtain a near optimal performance guarantee.

Lemma 1. *Let S be the set of k elements selected by a stochastic greedy algorithm s.t. in each iteration $i \in [k]$ the algorithm selects an element whose marginal contribution is an ξ_i approximation to the marginal contribution of the element with the largest marginal contribution at that stage, and let $\hat{\mu} = \frac{1}{k}\sum_{i=1}^{k} \xi_i$. Then:*

$$f(S) \geq \left(1 - \frac{1}{e^{\hat{\mu}}}\right) \texttt{OPT}.$$

Proof. Let $S_i = \{a_1, \ldots, a_i\}$, and let the optimal solution be O, i.e. $O \in \operatorname{argmax}_{T:|T| \leq k} f(T)$. Note that for any $i < k$ we have that:

$$\begin{aligned}
f(S_{i+1}) - f(S_i) &= f_{S_i}(a_{i+1}) \\
&\geq \xi_{i+1} \max_{o \in O} f_{S_i}(o) \\
&\geq \frac{\xi_{i+1}}{k} f_{S_i}(O) \\
&= \frac{\xi_{i+1}}{k} \left(f(O \cup S_i) - f(S_i)\right) \\
&\geq \frac{\xi_{i+1}}{k} \left(f(O) - f(S_i)\right)
\end{aligned}$$

Rearranging, we get:

$$f(S_{i+1}) \geq \frac{\xi_{i+1}}{k}\Big(f(O) - f(S_i)\Big) + f(S_i) \tag{1}$$

We will show by induction that in every iteration $i \in [k]$:

$$f(S_i) \geq \left(1 - \prod_{j=1}^{i}\left(1 - \frac{\xi_j}{k}\right)\right) f(O)$$

The base case is when $i = 1$, and $S_0 = \emptyset$ and by (1):

$$f(S_1) \geq \frac{\xi_{i+1}}{k}\Big(f(O) - f(S_0)\Big) = 1 - \left(1 - \frac{\xi_1}{k}\right) f(O)$$

For a general iteration $i+1$, applying (1) for iteration $i+1$ and using the inductive hypothesis:

$$\begin{aligned}
&f(S_{i+1}) \\
&\geq \frac{\xi_{i+1}}{k} f(O) + \left(1 - \frac{\xi_{i+1}}{k}\right) f(S_i) \\
&\geq \frac{\xi_{i+1}}{k} f(O) + \left(1 - \frac{\xi_{i+1}}{k}\right) \left(1 - \prod_{j=1}^{i}\left(1 - \frac{\xi_j}{k}\right)\right) f(O) \\
&= \left(1 - \prod_{j=1}^{i+1}\left(1 - \frac{\xi_j}{k}\right)\right) f(O)
\end{aligned}$$

Since $1 - x \leq e^{-x}$, the above inequality implies:

$$\begin{aligned}
f(S) &= f(S_k) \\
&= \left(1 - \prod_{j=1}^{k}\left(1 - \frac{\xi_j}{k}\right)\right) f(O) \\
&\geq \left(1 - e^{-\frac{1}{k}\sum_{i=1}^{k}\xi_i}\right) f(O) \\
&= \left(1 - e^{-\hat{\mu}}\right) f(O).
\end{aligned}$$

□

We can now apply concentration bounds on the previous lemma and prove the main theorem.

Theorem 2. *Let f be a monotone submodular function, which is evaluated with uncertainty coming from a distribution $\mathcal{D}$ with mean μ. For any $\varepsilon \in (0, 1)$, suppose a stochastic greedy algorithm is being used with $k \geq \frac{2}{\varepsilon^2 \mu}$. Then, w.p. $1 - e^{-\frac{\varepsilon\mu\cdot k}{2}}$ the algorithm returns S s.t.:*

$$f(S) \geq \left(1 - \frac{1}{e^{(1-\varepsilon)\mu}}\right) \texttt{OPT}$$

Proof. Consider an application of the stochastic greedy algorithm with mean μ, and let $\xi_1, \ldots, \xi_k$ be the approximations to the marginal contributions made by i.i.d samples from a distribution in all iterations $1, \ldots, k$. Since all the values $\{\xi_i\}_{i=1}^{k}$ drawn from the distribution are bounded from above by 1, by the Chernoff bound we have:

$$\Pr\left[\frac{1}{k}\sum_{i=1}^{k}\xi_i < (1-\varepsilon)\mu\right] < e^{\frac{\varepsilon\mu\cdot k}{2}}$$

By applying Lemma 1 we get our result. □

2.2. Tight lower bound

Claim 3. *For any $\delta \in [0, 1)$ the competitive ratio of stochastic greedy with mean μ is at most $(1 - 1/e^{\mu}) + o(1)$ with probability at least $1 - \delta$.*

Proof. Consider maximizing a submodular function when the oracle has probability μ of returning the element with the largest marginal contribution, and probability $1-\mu$ to return a random element. We present an instance where greedy returns an approximation ratio of at most $1-1/e^{\mu}+o(1)$ with probability at least $1-1/k$. We use the same bad instance regardless of μ.

The construction is as follows. There are k special elements, $m=4k^3$ plain elements and $n=4(m+k)k^2$ dummy elements (note that the total number of elements is not n). The value $f(S)$ depends only on the number of special elements, plain elements and dummy elements contained in S (all special elements are identical). Moreover, dummy elements contribute nothing to f, and hence, we can write $f(S)=f(i,j)$, where i is the number of special elements in S, and j is the number of plain elements.

For $i\geq 1$, the value of f depends on j as follows:

$$f(i,j)=\begin{cases} k^k\text{-}(k\text{-}1)^j k^{k\text{-}j\text{-}1}(k\text{-}i) & 0\leq j\leq k\text{-}1\\ k^k\text{-}(k\text{-}1)^{k\text{-}2}((k\text{-}i)(2k\text{-}2\text{-}j)) & k\leq j\leq 2k\text{-}3\\ k^k\text{-}(k\text{-}1)^{k\text{-}2}(3k\text{-}i\text{-}j\text{-}3) & 2k\text{-}2\leq j\leq 3k\text{-}3\text{-}i\\ k^k & 3k-i-2\leq j \end{cases}$$

Note that for $i=0$, we have $f(0,j)=f(1,j-1)$, and $f(0,0)=0$. Also, one can verify, by case-by-case analysis that this function is indeed monotone and submodular.

Since $f(0,j)=f(1,j-1)$ for every $j\geq 1$, as long as the greedy algorithm did not choose any special element yet, the marginal contribution of a special elements is equal to the marginal contribution of a plain element.

Let t be the number of times in which the oracle supplied the greedy algorithm with the element with the maximal marginal contribution. By an additive version of the Chernoff bound we have that with probability at least $1-2^{4\log k}=1-k^4$:

$$t<k\mu+2\sqrt{k}\log k$$

We condition on this event. Next, we argue that with probability at least $1-1/3k$ all the elements for which the algorithm selected a random element (i.e. did not take an element whose marginal contribution is a μ approximation to the largest marginal contribution) are dummy elements. Consider one of the times in which greedy was given a random element. The probability that it was not a dummy element is at most $\frac{m+k}{4(m+k)k^2-k}$. Applying a union bound, the probability that in the $k-t$ cases in which greedy was supplied with a random element it was always a dummy element is at least $1-1/3k$. We condition on this event.

We will now argue that with probability at least $1-1/3k$ all the non-dummy elements selected are plain. Consider the first non dummy element chosen by the algorithm. With probability at least $1-k/4k^3$ it is a plain element. We condition on this event. Assuming by induction that the last $i-1$ non dummy elements which were chosen by greedy were plain, the probability that i'th non dummy element is plain is $1-\frac{k}{4k^3-1}$. Taking a union bound over all t such elements, gives that with probability $1-1/3k$ all t non-dummy elements chosen by greedy were plain.

Combining this together with a union bound, we get that with probability at least $1-1/k$ greedy chose no special elements, at most $t=\mu k+4\sqrt{k}\log k$ plain elements, and the rest are dummy elements that do not contribute to the value of the function. This means that the value of the solution is at most $k^k-(k-1)^t k^{k-t}$, and thus with probability at least $1-1/k$ the ratio between the value of greedy and the optimal value is at least:

$$\begin{aligned}\frac{k^k-(k-1)^t k^{k-t}}{k^k} &= 1-\left(\frac{k-1}{k}\right)^t\\ &\geq 1-\left(\left(1-\frac{1}{k}\right)^k\right)^{\mu+\frac{4\log k}{\sqrt{k}}}\\ &\geq 1-e^{-\left(\mu+4\log k/\sqrt{k}\right)}\\ &= 1-e^{-\mu}+o(1)\end{aligned}$$

Choosing $k=1/\delta$ we get our desired bound. □

3. Modular Functions

In this section we show a tight upper bound for the special case in which the function is *modular*. Recall that a function $f:2^N\to\mathbb{R}$ is modular if for every set $S\subseteq N$ we have that $f(S)=\sum_{a\in S}f(a)$. Note that in this case $f_S(a)=f(a)$ for all $S\subseteq N$ and $a\in N$.

Theorem 4. *Let $S\subseteq N$ be the set returned when applying a stochastic greedy algorithm with mean $\mu\in[0,1]$ on a modular function $f:2^N\to\mathbb{R}$. Then, for any $\varepsilon\in[0,1]$, when $k\geq\frac{2}{\varepsilon^2\mu}$ w.p. $1-e^{-\mu k\varepsilon^2/2}$:*

$$f(S)\geq(1-\varepsilon)\mu OPT.$$

Proof. Suppose that at every stage $i\in[k]$ element $a_i\in N$ is selected and its marginal contribution is at least ξ_i of the optimal marginal contribution at that stage. Let O be the optimal solution and $o^\star\in\operatorname{argmax}_{o\in O\setminus S}f(o)$, where S is the solution retuned by the algorithm, i.e. $S=\{a_1,\ldots,a_k\}$. The basic idea in this proof is to observe that since $o^\star$ is not in the solution and throughout the iterations of the algorithm is always a feasible candidate, this implies that every element a_i in the solution $S\setminus O$ has value at least as large as $\xi_i f(o^\star)$. Intuitively, if there are enough elements in $S\setminus O$ for concentration bounds to kick in, we have that $\frac{1}{|S\setminus O|}\sum_{j\in S\setminus O}\xi_j=(1-\varepsilon)\mu$ and we would be done since $f(S)=f(S\setminus O)+f(O\cap S)\geq(1-\varepsilon)\mu f(O)$.

The problem is that $S \setminus O$ may not be sufficiently large, and we therefore need slightly more nuanced arguments.

We will partition $O \cap S$ to two disjoint sets of *low* and *high* valued elements: $L = \{o \in O \cap S \;:\; f(o) < f(o^\star)\}$ and $H = \{o \in O \cap S \;:\; f(o) \geq f(o^\star)\}$. Notice that:

$$\begin{aligned}
&f(O)\\
&= f(O \setminus S) + f(L) + f(H)\\
&= \sum_{o \in O \setminus S} f(o) + \sum_{o \in L} f(o) + \sum_{o \in H} f(o)\\
&\leq \sum_{o \in O \setminus S} f(o^\star) + \sum_{o \in L} f(o^\star) + \sum_{o \in H} f(o^\star) + \Big(f(o)\text{-}f(o^\star)\Big)\\
&= k \cdot f(o^\star) + \sum_{o \in H} (f(o)\text{-}f(o^\star))
\end{aligned}$$

Since $o^\star \in O \setminus S$, it is a feasible choice for the algorithm at every stage, and therefore by the definition of the stochastic greedy algorithm, for every element $a_i \in S$ we have that:

$$f(a_i) \geq \xi_i \max_{a \in N} f_S(a) = \xi_i \max_{a \in N \setminus S} f(a) \geq \xi_i f(o^\star)$$

Thus, for $k \geq 2/\varepsilon^2$ with probability $1 - e^{-\frac{k\mu\varepsilon^2}{2}}$:

$$\begin{aligned}
&f(S)\\
&= f(S \setminus O) + f(H) + f(L)\\
&= \sum_{a_i \in S \setminus O} f(a_i) + \sum_{a_i \in L} f(a_i) + \sum_{o \in H} f(o)\\
&\geq \sum_{a_i \in S \setminus O} \xi_i f(o^\star) + \sum_{a_i \in L} \xi_i f(o^\star) + \sum_{o \in H} f(o)\\
&= f(o^\star) \left(\sum_{a_i \in (S \setminus O) \cup L} \xi_i \right) + \sum_{o \in H} f(o)\\
&= f(o^\star) \left(\sum_{a_i \in (S \setminus O) \cup L} \xi_i \right) + \sum_{o \in H} \Big(f(o^\star) + f(o)\text{-}f(o^\star)\Big)\\
&\geq f(o^\star) \left(\sum_{a_i \in (S \setminus O) \cup L \cup H} \xi_i \right) + \sum_{o \in H} \Big(f(o)\text{-}f(o^\star)\Big) \quad (2)\\
&\geq (1-\varepsilon)\mu \cdot k f(o^\star) + \sum_{o \in H} \Big(f(o)\text{-}f(o^\star)\Big) \quad (3)\\
&\geq (1-\varepsilon)\mu \left(k f(o^\star) + \sum_{o \in H} \Big(f(o)\text{-}f(o^\star)\Big) \right)\\
&\geq (1-\varepsilon)\mu f(O)
\end{aligned}$$

where inequality (2) is due to the fact that $f(o^\star) \geq \xi_i f(o^\star)$ since $\xi_i \leq 1$; inequality (3) is an application of the Chernoff bound for $k \geq 2/\varepsilon^2\mu$; the last inequality is due to the upper bound we established on $f(O)$. □

The upper bound is tight. An obvious lower bound holds for the degenerate case where in every stage the marginal contribution of the element returned is a $\mu \in [0, 1]$ approximation to the maximal marginal contribution with probability 1. Clearly in this case, the approximation ratio is no better than μ (consider $n = 2k$ elements where k elements have value 1 and k elements have value μ).

4. General Matroid Constraints

In this section we consider the more general problem of maximizing a monotone submodular function under matroid constraints. Recall that a matroid is a pair $(N, \mathcal{I})$ where N is the ground set and $\mathcal{I}$ is a family of subsets of N called *independent* that respects two axioms: (1) $A \in \mathcal{I}, A' \subset A \implies A' \in \mathcal{I}$ and (2) if $A, B \in \mathcal{I}$ and $|B| < |A|$ then $\exists x \in A \setminus B$, s.t. $B \cup \{x\} \in \mathcal{I}$. The *rank* of the matroid is the size of the largest set in $\mathcal{I}$. The cardinality constraint, is a special case of optimization under matroid constraints, where the matroid is *uniform*.

Submodular maximization under matroid constraints. The greedy algorithm for maximization under matroid constraints is a simple generalization of the uniform case: the algorithm iteratively identifies the element whose marginal contribution is maximal and adds it to the solution if it does not not violate the matroid constraint (i.e. if adding the element to the set keeps the set in the family of feasible sets $\mathcal{I}$). This algorithm obtains an approximation ratio of $1/2$. [2] More generally, for an intersection of P matroids, this algorithm obtains an approximation guarantee of $1/(1+P)$.

Stochastic greedy under intersection of matroids. Consider an intersection of matroids, a monotone submodular function f defined over the independent sets, and an uncertainty distribution $\mathcal{D}$ with mean μ. The stochastic greedy algorithm begins with the solution $S = \emptyset$ and set of elements not yet considered $X = N$. In every iteration the algorithm maintains a solution S of elements that are in the intersection of the P matroids and a value $\xi \sim \mathcal{D}$ is sampled. The algorithm then considers an arbitrary element $a \in X$ whose marginal contribution is $\xi \max_{x \in X} f_S(x)$, and adds a to S if $S \cup a$ is independent in all P matroids, and discards a from X.

4.1. Stochastic greedy fails with high probability

We first show that unlike the special case of uniform matroids, even for a single matroid, it is generally impossible to obtain high probability guarantees for maximization un-

[2] We note that unlike the uniform case, here greedy is not optimal. The optimal guarantee of $1 - 1/e$ can be obtained via an algorithm based on a continuous relaxation (Vondrák, 2008) or through local search (Filmus & Ward, 2012).

der matroid constraints, even when the function is modular and the rank of the matroid is sufficiently large.

Claim 5. *Even for a modular function and arbitrarily large k, a stochastic greedy algorithm with mean μ cannot obtain an approximation better than 0 with probability greater than $\mu + o(1)$, for maximization under matroid of rank k.*

Proof. Consider the following example, where the ground set has two types of elements $A = \{a_1, \ldots, a_m\}$, and $B = \{b_1, \ldots, b_{k-1}\}$ where $m = k^2$. The rank of the matroid is k, and a set is independent as long as it contains just a single $a_i \in A$. Define a modular function: $f(a_1) = 1$, but $f(a_j) = 0$ for $j \neq 1$, and also $f(b_j) = 0$ for any $j \in [k-1]$. The distribution returns 1 w.p. $p < 1/2$ and 0 otherwise.

In the first iteration of the algorithm, the element a_1 is correctly evaluated with probability p, and with probability $1 - p$ it is evaluated as having value 0, in which case we may assume that a random element is selected instead. Therefore, w.p. p the algorithm takes a_1, and obtains the optimal solution. However, if this is not the case, then w.p. $m-1/(m+k) = (k^2-1)/(k^2+k)$ the algorithm chooses an element from A whose value is 0. In this case, even if a_1 is later correctly evaluated it could not be considered into the solution since its inclusion violates independence. Hence, while the expected value of greedy is slightly larger than p, with probability at least $(1-p) - o(1)$ the value of the solution would be 0. □

4.2. The guarantee holds in expectation

Although the approximation guarantee cannot hold with high probability, we now show that in expectation stochastic greedy algorithms achieves the approximation guarantee of non-stochastic greedy when maximizing a monotone submodular functions under an intersection of P matroids.

Theorem 6. *Let $\mathcal{F}$ denote the intersection of $P \geq 1$ matroids on the ground set N, and $f : 2^N \to \mathbb{R}$ be a monotone submodular function. The stochastic greedy algorithm returns a solution $S \in \mathcal{F}$ s.t.:*

$$\mathbb{E}[f(S)] \geq \frac{\mu}{(P+1)} OPT$$

.

An equivalent algorithm. To simplify the analysis, it will be useful to consider the equivalent algorithm, which at every iteration when the existing solution is S, discards all elements x for which $S \cup x \notin \mathcal{F}$. The following claim due to Nemhauser et al. is later employed in our analysis:

Claim 7 (Prop. 2.2 in (Nemhauser et al., 1978))**.** *If for $\forall t \in [k]$ $\sum_{i=0}^{t-1} \sigma_i \leq t$ and $p_{i-1} \geq p_i$, with $\sigma_i, p_i \geq 0$ then:*

$$\sum_{i=0}^{k-1} p_i \sigma_i \leq \sum_{i=0}^{k-1} p_i.$$

Algorithm 2 STOCHASTIC-MATROID-GREEDY

1: $S \leftarrow \emptyset, X \leftarrow N$
2: **while** $X \neq S$ **do**
3: $\quad X \leftarrow X \setminus \{x : S \cup \{x\} \notin \mathcal{F}\}$
4: $\quad \xi \sim \mathcal{D}$.
5: $\quad S \leftarrow S \cup$ arbitrary a s.t. $f_S(a) \geq \xi \max_{x \in X} f_S(x)$
6: **end while**

Proof of Lemma 6. Consider the value obtained by the following procedure. An adversary chooses some maximal independent set $a_1, \ldots a_k$. Let $S_i = \{a_1, a_2, \ldots, a_i\}$ with $S_0 = \emptyset$, and for every $i \in [k]$ let $a_i^\star$ be the element that maximizes the marginal contribution given S_i, where the maximization is over elements a such that $S_i \cup \{a\}$ is independent in all P matroids. That is $a_i^\star$ is defined as:

$$\max_{S_i \cup \{a\} \in \mathcal{F}} f_{S_i}(a)$$

The value of the procedure is then:

$$\sum_{i=0}^{k-1} f_{S_i}(a_i^\star)$$

We will bound the value obtained by the procedure against that of the optimal solution, and then argue that the value obtained by the stochastic greedy is equivalent.

Let O denote the optimal solution. We have that:

$$f(O) \leq f(O \cup S_k) \leq f(S_k) + \sum_{x \in O \setminus S_k} f_{S_k}(x) \qquad (4)$$

For a set S and a matroid M_p in the family $\mathcal{F}$, we define $r_p(S)$, called the rank of S in M_p to be the cardinality of the largest subset of S which is independent in M_p, and define $sp_p(S)$, called the span of S in M_p by:

$$sp_p(S) = \{a \in N : r_p(S \cup a) = r_p(S)\}$$

If S is independent in M_p, we have that $r_p(sp_p(S)) = r_p(S) = |S|$. In particular, we have that $r_p(sp_p(S_i)) = i$ for every S_i. Now in each $1 \leq p \leq P$, since O is an independent set in M_p we have:

$$r_p(sp_p(S_i) \cap (O)) = |sp_p(S_i) \cap (O)|$$

which implies that $|sp_p(S_i) \cap (O)| \leq i$.

Define $U_i = \cup_{p=1}^{P} sp_p(S_i)$, to be the set of elements which are not part of the maximization in step $i+1$ of the procedure, and hence cannot give value at that stage. We have:

$$|U_i \cap O| = |(\cup_{p=1}^{P} sp_p(S^i)) \cap O| \leq \sum_{p=1}^{P} |sp_p(S^i) \cap O| \leq iP$$

Let $V_i = (U_i \setminus U_{i-1}) \cap O$ be the elements of O which are not part of the maximization in step i, but were part of the maximization in step $i-1$. If $x \in V_i$ then it must be that:

$$f_{S_k}(x) \leq f_{S_i}(x) \leq f_{S_i}(a_i^\star)$$

since x was not chosen in step i. Hence, we can upper bound:

$$\sum_{x \in O \setminus S_k} f_{S_k}(x) \leq \sum_{i=1}^{k} \sum_{x \in V_i} f_{S_i}(a_i^\star)$$
$$= \sum_{i=1}^{k} |V_i| f_{S_i}(a^\star) \leq P \sum_{i=1}^{k} f_{S_i}(a^\star)$$

where the last inequality uses $\sum_{t=1}^{i} |V_t| = |U_i \cap O| \leq Pi$ and the arithmetic claim proven in Claim 7 due to (Nemhauser et al., 1978). Together with (4), we get:

$$f(O) \leq (P+1) \sum_{i=1}^{k} f_{S_i}(a_i^\star)$$

Finally, note that STOCHASTIC-MATROID-GREEDY obtains a $\mu = \mathbb{E}_{\xi \sim \mathcal{D}}[\xi]$ approximation of the value of the procedure, in expectation. In each stage, one can add the element chosen by the algorithm to the procedure. Hence, at each stage STOCHASTIC-MATROID-GREEDY and the procedure have the same set of elements available, and the same $a_i^\star$ which maximizes the marginal contribution. □

5. Inapproximability of Local Search

In this section we consider variants of stochastic local search algorithms. We show that unlike the greedy algorithm, stochastic local search algorithms can end up with arbitrarily bad approximation guarantees.

Local search for submodular maximization. For $N = \{a_1, \ldots, a_n\}$, given a set of elements $T \subseteq N$ we will use T_{-i} to denote the set without element a_i, i.e. $T_{-i} = T \setminus \{a_i\}$. A solution S is a *local maximum* if no single element a_i in S can be exchanged for another element a_j not in S whose marginal contribution to S_{-i} is greater. That is, S is a local maximum if for every $a_i \in S$ we have that:

$$f_{S_{-i}}(a_i) \geq \max_{x \notin S} f_{S_{-i}}(x).$$

It is not hard to show that for any monotone submodular function, if S is a local maximum it is a $1/2$ approximation to the optimal solution. A local search algorithm begins with an arbitrarily set of size k, and at every stage exchanges one of its elements with the element whose marginal contribution is maximal to the set, until it reaches a local maximum. To guarantee that local search algorithms converge in polynomial time, the convention is to seek *approximate* local maxima. A solution S is an *α-approximate local maximum* if no element a_i in S can be exchanged for another element a_j not in S whose marginal contribution to S_{-i} is greater by a factor of α. It is easy to show that an α-approximate local maximum is a $(1+1/\alpha)^{-1}$ approximation (Filmus & Ward, 2012).

Stochastic local search. A natural question is whether local search enjoys the same robustness guarantees as the greedy algorithm. We say that a solution S is a *stochastic local maximum up to approximation* μ if no single element in S can be exchanged for another element not in S whose expected marginal contribution is greater by a factor μ. That is, S is a stochastic local maximum with mean μ if for every $a_i \in S$ we have that:

$$\mathbb{E}[f_{S_{-i}}(a_i)] \geq \mu \cdot \max_{x \notin S} f_{S_{-i}}(x)$$

If we have uncertaintiy modeled by a distribution $\mathcal{D} \subseteq [0,1]$, a solution is a stochastic local maximum w.r.t $\mathcal{D}$ if for every element a_i in S we draw $\xi_i \sim D$ s.t.

$$f_{S_{-i}}(a_i) \geq \xi_i \cdot \max_{b \notin S} f_{S_{-i}}(b)$$

A stochastic local search algorithm will therefore begin from an arbitrary solution S of size k, and at every iteration swap an element from the solution with an element outside the solution if S is not a stochastic local maximum w.r.t. $\mathcal{D}$. More specifically, the stochastic local search algorithm selects an element a_i from S and replaces it with another element a_j whose expected marginal contribution to S_{-i} is at least $f_{S_{-i}}(a_i)/\xi_i$, and repeats this process until no such elements are found. This is a similar abstraction of stochastic greedy algorithms, applicable in settings when one cannot evaluate the optimal marginal contribution exactly, but approximately well in expectation.

Consistent and inconsistent stochasticity. We consider two approaches to model the way in which the random variables ξ_i are assigned to an element a_i in the solution:

1. **Consistent:** For each element $a_i \in N$, ξ_i is a random variable drawn independently from $\mathcal{D} \subseteq [0,1]$, and fixed for the entire run of the algorithm;
2. **Inconsistent:** At each step of the algorithm, for every element $a_i \in S$, ξ_i is a random variable drawn independently from $\mathcal{D}$.

Note that the solution converges, as the distribution $\mathcal{D}$ makes the algorithm more conservative.

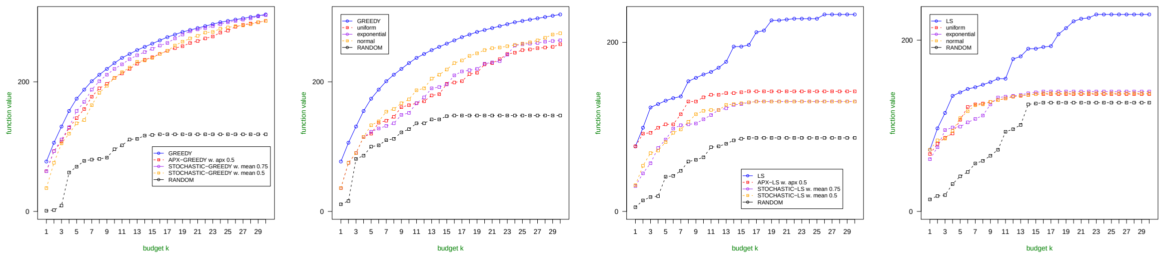

Figure 1: Function value as a function of solution size for greedy and local search with uncertainty

Inapproximability of stochastic local search. We show that in both consistent and inconsistent models, stochastic local search performs poorly, even for modular functions. Consider a setting where there are n elements, and a modular function. For every $i > 1$, we have $f(a_i) = \varepsilon/i$ for some negligible $\varepsilon > 0$ (e.g. $\varepsilon = 2^{-n}$), but $f(a_1) = 1$. As for $\mathcal{D}$, w.p. 0.99 it returns 1, and 0 o.w. Assume $k = 1$.

Lemma 8. *The expected approximation guarantee of stochastic local search is at most* $2^{-O(n)} + \varepsilon$.

Proof. At the first iteration, local search chooses a_n. If $\xi = 0$, we are done, and this is a local maxima. Otherwise, local search chooses a_{n-1}. At iteration i local search starts with a_i, halts w.p. 0.01 (the probability that $\mathcal{D}$ outputs 0), and otherwise continues. The probability that it will not halt for n steps and reach a_i is $0.99^n = 2^{-O(n)}$. ☐

We note that in the above proof we assumed that the local search algorithm chooses an arbitrary element at every iteration. If one allows the stochastic local search to randomly choose an element in every iteration a similar construction shows an inapproximability of $O(\frac{\log n}{n}) + \varepsilon$.

6. Experiments

We applied the algorithms to an ego-network from (Leskovec & Krevl, 2014). This network has 333 nodes and 5038 edges. The submodular function we used is coverage, which models influence in social networks. In order to emphasize the implications of having results w.h.p, the graphs do not depict the average of many runs, but instead each graph is a single run of the algorithm. In greedy, we present the value of the solution at each iteration k. In local search and in random we sort elements of the solution according to marginal contributions.

We start with greedy and describe the different distributions we used to model uncertainty. The same distributions were used for local search. Both left panes include the greedy algorithm without uncertainty (greedy, blue line), and choosing a random set (random, black line). When running stochastic greedy, we first sample a value $\xi \in \mathcal{D}$, and then pick a random element out of the elements that have marginal contribution at least $\xi \max_a f_S(a)$. The distribution $\mathcal{D}$ varies between the different lines. In the leftmost pane, APX (red line) depicts a $\mathcal{D}$ which is the constant distribution 0.5. In Stochastic greedy with mean 0.75, $\mathcal{D}$ is the uniform distribution on $[0.5, 1]$ (purple line), and in Stochastic greedy with mean 0.5, $\mathcal{D}$ is the uniform distribution on $[0, 1]$. It is expected that APX will behave smoothly, as $\mathcal{D}$ is a degenerate distribution in this case (note that there is still randomization in which element to choose at every stage out of the eligible elements). However, we see that the h.p. result kicks in, and the APX line is similar (across many values of k) to stochastic greedy with mean 0.5. Raising the mean to 0.75 makes stochastic greedy behave almost like greedy when k gets large, so in some cases stochastic greedy makes the same choice greedy would make.

In the second pane, The purple line (exponential) depicts $\mathcal{D}$ as an exponential distribution with $\lambda = 4$, which gives a mean of 0.25. The red line is uniform in $[0, 5]$, and the yellow is a Gaussian with $\mu = 0.25$ and $\sigma = 0.1$. We see that all graphs are further away from Greedy compared to the leftmost pane, and that higher variance is generally not a good thing, although the differences are small.

The two right panes depict the same noise distributions as the two left panes, but this time we use local search (or stochastic local search) instead of greedy. It is easy to see that $\mathcal{D}$ affects local search more than it affects greedy. The plateau is caused since we sort the final solution and then plot the elements, and since if some elements have a low value of ξ_i they are likely to stay in the solution even if they contribute very little, as in Lemma 8.

7. Acknowledgements

A.H. was supported by ISF 1394/16; Y.S. was supported by NSF grant CCF-1301976, CAREER CCF-1452961, a Google Faculty Research Award, and a Facebook Faculty Gift. We thank Andreas Krause for pointing the connection between our result and (Mirzasoleiman et al., 2015).

References

Azaria, Amos, Hassidim, Avinatan, Kraus, Sarit, Eshkol, Adi, Weintraub, Ofer, and Netanely, Irit. Movie recommender system for profit maximization. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 121–128. ACM, 2013.

Feige, Uriel. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4).

Filmus, Yuval and Ward, Justin. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *FOCS*, 2012.

Hassidim, Avinatan and Singer, Yaron. Submodular optimization under noise. In *COLT*, 2017.

Leskovec, Jure and Krevl, Andrej. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

Li, Chengtao, Sra, Suvrit, and Jegelka, Stefanie. Gaussian quadrature for matrix inverse forms with applications. In *ICML*, 2016.

Lindgren, Erik, Wu, Shanshan, and Dimakis, Alexandros G. Leveraging sparsity for efficient submodular data summarization. In *NIPS*, 2016.

Lucic, Mario, Bachem, Olivier, Zadimoghaddam, Morteza, and Krause, Andreas. Horizontally scalable submodular maximization. In *ICML*, 2016.

Malioutov, Dmitry, Kumar, Abhishek, and Yen, Ian En-Hsu. Large-scale submodular greedy exemplar selection with structured similarity matrices. In *UAI*, 2016.

Mirzasoleiman, Baharan, Badanidiyuru, Ashwinkumar, Karbasi, Amin, Vondrák, Jan, and Krause, Andreas. Lazier than lazy greedy. In *AAAI*, 2015.

Nemhauser, George L and Wolsey, Leonard A. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3): 177–188, 1978.

Nemhauser, George L, Wolsey, Laurence A, and Fisher, Marshall L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

Sharma, Dravyansh, Kapoor, Ashish, and Deshpande, Amit. On greedy maximization of entropy. In *ICML*, 2015.

Singla, Adish, Tschiatschek, Sebastian, and Krause, Andreas. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In *AAAI*, 2016.

Vondrák, Jan. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, 2008.

Zhuang, Hao, Rahman, Rameez, Hu, Xia, Guo, Tian, Hui, Pan, and Aberer, Karl. Data summarization with social contexts. In *CIKM*, 2016.