

Multi-Objective Maximization of Monotone Submodular Functions with Cardinality Constraint (Extended Abstract)

Rajan Udwani

rudwani@mit.edu

Operations Research Center, Massachusetts Institute of Technology

Abstract

We consider the problem of multi-objective maximization of monotone submodular functions subject to cardinality constraint, one formulation of which is $\max_{|A|=k} \min_{i \in \{1, \dots, m\}} f_i(A)$. Krause et al. (2008) showed that when the number of functions m grows as the cardinality k i.e., $m = \Omega(k)$, the problem is inapproximable (unless $P = NP$). For the more general case of matroid constraint, Chekuri et al. (2010) gave a randomized $(1 - 1/e) - \epsilon$ approximation for constant m . The runtime (number of queries to function oracle) scales exponentially as n^{m/ϵ^3} . We give the first polynomial time asymptotically constant factor approximations for $m = o(\frac{k}{\log^3 k})$: (i) A randomized $(1 - 1/e)$ algorithm based on Chekuri et al. (2010). (ii) A faster and more practical $\tilde{O}(n/\delta^3)$ time, randomized $(1 - 1/e)^2 - \delta$ approximation based on Multiplicative-Weight-Updates. Finally, we characterize the variation in optimal solution value as a function of the cardinality k , leading to a derandomized approximation for constant m .

1 Introduction

Optimization problems with submodular objective functions have received a lot of interest due to several applications where instances of these problems arise naturally. The problem of maximizing a monotone submodular function¹ on ground set N , subject to a cardinality constraint, written as:

$$P_0 := \max_{A \subseteq N, |A| \leq k} f(A),$$

has been very well studied. Nemhauser et al. [14, 13] showed that the greedy algorithm gives a guarantee of $(1 - 1/e)$ and this is best possible in the value-oracle model. Later, Feige [6] showed this is also the best possible approximation under standard complexity assumptions. Further, Badanidiyuru and Vondrák [2] found a faster algorithm for P_0 that improved the quadratic query complexity of the classical greedy algorithm to nearly linear complexity, by trading off on the approximation guarantee. For the more general problem $\max_{A \in \mathcal{I}} f(A)$, where \mathcal{I} is the collection of independent sets of a matroid; Calinescu et al. [3], obtained a $(1 - 1/e)$ approximation by (approximately) optimizing the *multilinear extension* of the submodular function, followed by suitable rounding. Based on this framework, tremendous progress has been made over the last decade for a variety of different settings [3, 19, 7, 20, 21, 4, 5].

Krause et al. [10] argue that in several practical applications where instances of P_0 arise naturally, it is often the case that the objective is uncertain. In such cases one often wishes to select a set robust against the worst-case objective function. This motivated them to consider the following:

$$P_1 = \max_{A \subseteq N, |A| \leq k} \min_{i \in \{1, 2, \dots, m\}} f_i(A),$$

where $f_i(\cdot)$ is monotone submodular for every i . They show that the problem is inapproximable unless $P = NP$. This is only when $m = \Omega(k)$. However, there are several applications [9, 11, 10] of the formulation where the number of functions m is a design choice. Further, Chekuri et al. [4] gave a randomized $(1 - 1/e) - \epsilon$ approximation for the more general case of matroid constraint, when m is constant. The runtime scales as n^{m/ϵ^3} . Incidentally, they consider a different but equivalent formulation of the problem that stems from an influential paper by Papadimitriou and Yannakakis [17], on multi-objective optimization of linear functions (more on this in Section 2). Recently [15] showed that the greedy algorithm can be generalized to achieve

¹ A set function $f : 2^N \rightarrow \mathbb{R}$ on the ground set N is submodular if: $f(A+a) - f(A) \leq f(B+a) - f(B)$ for all $B \subseteq A \subseteq N$ and $a \in N \setminus A$. The function is monotone if $f(B) \leq f(A)$ for all $B \subseteq A$.

a deterministic $1 - 1/e - \epsilon$ approximation for the special case of bi-objective maximization subject to a cardinality constraint. This still leaves open the question of whether the problem is approximable when $m = o(k)$.

Our main contributions: We focus on the cardinality constraint case and show that the problem is approximable up to a factor of $(1 - 1/e - \epsilon)(1 - \frac{m}{k\epsilon^3})$ which for $k \rightarrow \infty$, $m = o(\frac{k}{\log^3 k})$ and $\epsilon = \min\{\frac{1}{8 \ln m}, \sqrt[4]{\frac{m}{k}}\}$ tends to $1 - 1/e$. This implies a very sharp threshold, since the problem cannot be approximated to within any polynomial factor for $m = \Omega(k)$ [10]. However, the runtime is $O(n^5)$.

Further, using MWU we give a fast $O(\frac{n}{\delta^3} \log m \log \frac{n}{\delta})$ time $(1 - 1/e)^2(1 - m/k\epsilon^3) - \epsilon - \delta$ approximation. Under the same asymptotic conditions as above, the guarantee simplifies to $(1 - 1/e)^2 - \delta$.

Finally, we give an upper bound on the increase in optimal solution value as a function of cardinality k and essentially resolve a conjecture posed in [16], leading to a deterministic $1 - 1/e - \epsilon$ approximation for constant m (all other results are randomized).

2 Preliminaries

We introduce some notation followed by a brief overview of the algorithm in Chekuri et.al. [4], simplified to the case of cardinality constraint. Recall, we use N to denote the ground set of n elements and monotone submodular functions $f_i(\cdot), i \in \{1, \dots, m\}$. For real vectors \mathbf{x} , we use the short hand $\|\mathbf{x}\|$ to denote the ℓ_1 norm. The marginal increase in function value when set X is combined with set A is $f(X|A) = f(A \cup X) - f(A)$. Next, the *multilinear extension* of a set function $f(\cdot)$ over $\mathbf{x} = \{x_1, \dots, x_n\} \in [0, 1]^n$ is defined as, $F(\mathbf{x}) = \sum_{S \subseteq N} f(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j)$. While evaluating the exact value of this function is naturally hard, estimates from sampling oracles suffice for usage in optimization algorithms [2, 4, 3].

In another variant of the problem P_1 , which was introduced in [4], we are given a set of m positive values V_i and we wish to find a set S^* of size k , such that $f_i(S^*) \geq V_i, \forall i \in \{1, \dots, m\}$ or certify that no S^* exists. We call this variant P_2 . Since P_2 is clearly computationally hard, one must consider approximations instead. This involves efficiently finding a set S such that $f_i(S) \geq \alpha V_i$ for all i and some factor α , or certifying that there is no set S^* such that $f_i(S^*) \geq V_i, \forall i$. The two formulations P_1 and P_2 are equivalent. Simply observe that w.l.o.g. we can assume $V_i = 1, \forall i$ (since we can consider functions $f_i(\cdot)/V_i$ instead). Finally, for cardinality k , we denote an optimal solution to P_1/P_2 using S_k^* and let OPT_k denote the optimal solution value for formulation P_1 .

We now review the algorithm by Chekuri et.al [4] which is based on P_2 . It has three stages,

Stage 1: Intuitively, the purpose of this stage is to guess a small initial set of size at most m/ϵ^3 . Call this set S_1 , then the desired property of S_1 is that for every element $e \in N \setminus S_1$, $f_i(e|S_1) < \epsilon^3 V_i$ for every i . This is necessary for the rounding in Stage 3 to work. In particular, due to a matroid constraint in [4], it is not clear if one can do better than trying all initial sets of size $\leq m/\epsilon^3$. This stage is thus the bottleneck that contributes the n^{m/ϵ^3} term to runtime.

Stage 2: Given an initial set S_1 , this stage works with ground set $N \setminus S_1$, cardinality parameter $k_1 = k - |S_1|$ and outputs a point \mathbf{x} with $\|\mathbf{x}\| = k_1$ and $F_i(\mathbf{x}|S) \geq (1 - 1/e) f_i(S_k^*|S_1)$. It has runtime $O(n^5)$ [2, 8, 3].

Stage 3: Given that S_1 was suitably chosen i.e. for every $e \in N \setminus S_1$, $f_i(e|S_1) < \epsilon^3 V_i$ for every i , and $\epsilon < 1/8\gamma \ln m$. This stage decomposes the fractional solution \mathbf{x} into convex combination of sets of size k_1 . Then rounds it to a set S_2 of size k_1 such that w.p. at least $1/m^{\gamma-1}$, $f_i(S_2|S_1) \geq (1 - \epsilon) F_i(\mathbf{x}|S_1)$ for every i . The result is easily converted to one w.h.p. by standard repetition. The final output is $S_1 \cup S_2$.

3 Main Results

3.1 A $(1 - 1/e)$ algorithm for $m = o(\frac{k}{\log^3 k})$

We replace the enumeration in Stage 1 with a single starting set, obtained by scanning once over the ground set.

New Stage 1: Start with $S_1 = \emptyset$ and pass over all elements once, updating S_1 whenever an element is chosen to be added. For each element e , add it to S_1 if for some i , $f_i(e|S_1) \geq \epsilon^3 V_i$. We mention without proof

that the process adds at most m/ϵ^3 elements (at most $1/\epsilon^3$ for any one function) and when the subroutine terminates, for every remaining element $e \in N \setminus S_1$, $f_i(e|S_1) < \epsilon^3 V_i, \forall i$.

The idea here is that in case of cardinality constraint we can choose *any* small starting set such that the remaining elements have marginal values $\leq \epsilon^3 V_i$, whereas for a matroid constraint an arbitrary set satisfying the property doesn't suffice. We also need to make a minor modification to Stage 2 but we omit that detail here. The rounding process in Stage 3 remains unchanged.

Theorem 1. *For $\epsilon = \min\{\frac{1}{8 \ln m}, \sqrt[4]{\frac{m}{k}}\}$, with constant probability the algorithm outputs a $(1 - 1/e)(1 - \epsilon)(1 - m/k\epsilon^3)$ approximate solution. Asymptotically, $1 - 1/e$ approximate for $m = o(\frac{k}{\log^3 k})$.*

Owing to the limitation posed by the $O(n^5)$ runtime of Stage 2, we next introduce a simple and fast algorithm. The speedup however comes at the cost of an extra factor of $(1 - 1/e)$ in the approximation guarantee.

3.2 A fast $(1 - 1/e)^2 - \epsilon$ algorithm for $m = o(\frac{k}{\log^3 k})$

Let S_1 be the output of the New Stage 1 discussed above. We replace Stage 2 with a fast MWU based subroutine that runs for $T = O(\frac{\ln m}{\delta^2})$ rounds (for suitably chosen δ) and solves an instance of P_0 in each round. The currently fastest algorithm for P_0 , in [12], has runtime $O(n \log 1/\delta)$ and an *expected* guarantee of $(1 - 1/e - \delta)$. However, the slightly slower, but still nearly linear time $O(\frac{n}{\delta} \log \frac{n}{\delta})$ *thresholding* algorithm in [2], has the usual deterministic guarantee. Either of these would lead to a runtime of $T \times \tilde{O}(n/\delta) = \tilde{O}(\frac{n}{\delta^3})$ which is a vast improvement over $O(n^5)$.

Consider some algorithm \mathcal{A} for P_0 with guarantee α , and let $\mathcal{A}(f, k)$ denote its output given monotone submodular function f and cardinality constraint k as input. Note that α can be as large as $1 - 1/e$, and also $k_1 = k - |S_1|$ as before.

Stage 2: MWU

- 1: Initialize $T = \frac{2 \ln m}{\delta^2}, \lambda_i^1 = 1/m, \tilde{f}_i(\cdot) = \frac{f_i(\cdot|S_1)}{V_i - f_i(S_1)}$
 - 2: **while** $1 \leq t \leq T$ **do**
 - 3: $g^t(\cdot) = \sum_{i=1}^m \lambda_i^t \tilde{f}_i(\cdot)$
 - 4: $X^t = \mathcal{A}(g, k_1)$
 - 5: $m_i^t = \tilde{f}_i(X^t) - \alpha$
 - 6: $\lambda_i^{t+1} = \lambda_i^t (1 - \delta m_i^t)$
 - 7: $t = t + 1$
 - 8: Output: $\mathbf{x}_2 = \frac{1}{T} \sum_{t=1}^T X^t$
-

Observe that the “experts” in this instance of MWU are the normalized functions $\tilde{f}_i(\cdot)$. In the Plotkin-Shmoys-Tardos (PST) framework, each round involves solving an LP with a convex combination of linear constraints (the experts). Here we solve an instance of P_0 for the monotone submodular function given by a convex combination of the experts \tilde{f}_i . Further in the PST framework, the convex combination of solutions obtained over all rounds is approximately feasible due to linearity. Here the non-linearity leads to a loss in guarantee. The point \mathbf{x}_2 is rounded to a set S_2 in Stage 3 (which remains unchanged). The final output is $S_1 \cup S_2$. We summarize the result on approximation guarantee in the following lemmas.

Lemma 2. $\frac{\sum_t \tilde{f}_i(X^t)}{T} \geq \alpha(1 - m/k\epsilon^3) - \delta$.

Lemma 3. $\tilde{F}_i(\mathbf{x}_2) \geq (1 - 1/e) \frac{\sum_{t=1}^T \tilde{f}_i(X^t)}{T}$

Theorem 4. *For $\epsilon = \min\{\frac{1}{8 \ln m}, \sqrt[4]{\frac{m}{k}}\}$, the algorithm makes $O(\frac{n}{\delta^3} \log m \log \frac{n}{\delta})$ queries, and with constant probability, outputs a feasible $(1 - 1/e)^2(1 - \frac{m}{k\epsilon^3})(1 - \epsilon) - \delta$ approximate set. Asymptotically, $(1 - 1/e)^2 - \delta$ approximate for $m = o(\frac{k}{\log^3 k})$.*

Proof. (Sketch) Using the algorithm in [2] as \mathcal{A} , we have $\alpha = (1 - 1/e)(1 - \delta')$ with runtime $O(\frac{n}{\delta'} \log \frac{n}{\delta'})$. Recall that S_2 is the set obtained by rounding \mathbf{x}_2 . It follows from Lemmas 2 and 3 and the definition of \tilde{f}_i , that $f_i(S_2|S_1) \geq [(1 - 1/e)^2(1 - m/k\epsilon^3)(1 - \epsilon) - \delta](V_i - f_i(S_1)), \forall i$. Hence $f_i(S_1 \cup S_2) \geq [(1 - 1/e)^2(1 - m/k\epsilon^3)(1 - \epsilon) - \delta]V_i, \forall i$ as claimed. Stage 2 takes time $O((\frac{n}{\delta} \log \frac{n}{\delta}) \times \frac{\log m}{\delta^2}) = O(\frac{n}{\delta^3} \log m \log \frac{n}{\delta})$. Finally, it can be shown that this is indeed the dominant term in the overall runtime. \square

3.3 Variation in optimal solution value as k increases

Consider the problem P_0 with cardinality constraint k . Given an optimal solution, S_k^* with value OPT_k for the problem, it is known that for arbitrary $k' \leq k$, there is a subset $S_{k'} \subseteq S_k^*$ of size k' , such that $f(S_{k'}) \geq \frac{k'}{k} OPT_k$.

This raises a natural question: Can we generalize this bound on variation of optimal solution value with varying k , for multi-objective maximization? A priori, this isn't obvious even for modular functions. We claim the following,

Theorem 5. *Given that there exists a set S_k^* such that $f_i(S_k^*) \geq V_i, \forall i$ and $\epsilon < \frac{1}{8 \ln m}$. For every $k' \in [m/\epsilon^3, k]$, there exists $S_{k'} \subseteq S_k^*$ of size k' , such that,*

$$f_i(S_{k'}) \geq (1 - \epsilon) \left(\frac{k' - m/\epsilon^3}{k - m/\epsilon^3} \right) V_i, \forall i.$$

Observe that for $k' \gg m/\epsilon^3$, we have $f_i(S_{k'}) \gtrsim (1 - \epsilon) \frac{k'}{k} V_i, \forall i$. Previously, a stronger result was known from [16] but only for $m = 2$. It was also conjectured in [16] that a similar result could be shown for larger m , and that such a bound would lead to a deterministic $(1 - 1/e) - \epsilon$ approximation for the multi-objective maximization problem, when m is constant. Our result indeed suffices to obtain such a deterministic algorithm with runtime scaling as n^{m/ϵ^4} . Note that all previous algorithms, including the ones presented here are randomized.

Theorem 6. *For $k' = \frac{m}{\epsilon^4}$, choosing k' -tuples greedily w.r.t. $h(\cdot) = \min_i f_i(\cdot)$ yields approximation guarantee $(1 - 1/e)(1 - 2\epsilon)$ while making n^{m/ϵ^4} queries.*

4 Conclusion and Open Problems

In summary, we consider the problem of multi-objective maximization of monotone submodular functions subject to a cardinality constraint, when $m = o\left(\frac{k}{\log^3 k}\right)$. No polynomial time constant factor approximations or strong inapproximability results were known for the problem, though it was known that the problem is inapproximable when $m = \Omega(k)$ and admitted a nearly $1 - 1/e$ approximation for constant m . We showed that when $m = o\left(\frac{k}{\log^3 k}\right)$, one can indeed approach the best possible guarantee of $1 - 1/e$ and further also gave a nearly-linear time $(1 - 1/e)^2$ approximation for the same. Finally, we also established a natural bound on how the optimal solution value increases with increasing cardinality k of the set.

A natural question of course is whether one can achieve approximations right up to $m = o(k)$. Additionally, it also of interest to ask if there are fast algorithms with guarantee closer to $1 - 1/e$, in contrast to the guarantee of $(1 - 1/e)^2$ shown here. Further, most of the ideas, and all results here, are for the case of uniform matroid (cardinality constraint). It is unclear if similar results can also be shown for a general matroid constraint.

Acknowledgments

The author would like to thank James B. Orlin and Andreas S. Schulz for their insightful comments and feedback on drafts of this work.

References

- [1] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8:121–164, 2012.
- [2] A. Badanidiyuru and J. Vondrák. Fast algorithms for maximizing submodular functions. In *SODA '14*, pages 1497–1514. SIAM, 2014.
- [3] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.
- [4] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS 10*, pages 575–584. IEEE, 2010.
- [5] S. Dobzinski and J. Vondrák. From query complexity to computational complexity. In *STOC '12*, pages 1107–1116. ACM, 2012.

- [6] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
- [7] M. Feldman, J. Naor, and R. Schwartz. A unified continuous greedy algorithm for submodular maximization. In *FOCS '11*, pages 570–579. IEEE.
- [8] Y. Filmus and J. Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM Journal on Computing*, 43:514–542, 2014.
- [9] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. *UAI'05*, pages 324–331, 2005.
- [10] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9:2761–2801, 2008.
- [11] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.
- [12] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. In *AAAI*, 2015.
- [13] G. Nemhauser and L. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.
- [14] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.
- [15] J. Orlin, A. Schulz, and R. Udwani. Robust monotone submodular function maximization. In *IPCO 2016*, pages 312–324.
- [16] J. Orlin, A. Schulz, and R. Udwani. Robust monotone submodular function maximization. *arXiv preprint arXiv:1507.06616*, 2015.
- [17] C. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *FOCS*, pages 86–92, 2000.
- [18] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. In *FOCS*, pages 495–504, 1991.
- [19] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC '08*, pages 67–74. ACM.
- [20] J. Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.
- [21] J. Vondrák, C. Chekuri, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *STOC '11*, pages 783–792. ACM, 2011.