Structured Robust Submodular Maximization: Offline and Online Algorithms

Nima Anari
Stanford University

Nika Haghtalab
Microsoft Research

Joseph (Seffi) Naor
Technion

Sebastian Pokutta
Georgia Tech

Mohit Singh
Georgia Tech

Alfredo Torrico
Georgia Tech

Abstract

Constrained submodular function maximization has been used in subset selection problems such as selection of most informative sensor locations. While these models have been quite popular, the solutions obtained via this approach are unstable to perturbations in data defining the submodular functions. Robust submodular maximization has been proposed as a richer model that aims to overcome this discrepancy as well as increase the modeling scope of submodular optimization.

In this work, we consider robust submodular maximization with structured combinatorial constraints and give efficient algorithms with provable guarantees. Our approach is applicable to constraints defined by single or multiple matroids, knapsack as well as distributionally robust criteria. We consider both the offline setting where the data defining the problem is known in advance as well as the online setting where the input data is revealed over time. For the offline setting, we give a nearly optimal bi-criteria approximation algorithm that relies on new extensions of the classical greedy algorithm. For the online version of the problem, we give an algorithm that returns a bi-criteria solution with sub-linear regret.

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

1 Introduction

Constrained submodular function maximization has seen significant progress in recent years in the design and analysis of new algorithms with guarantees (Calinescu et al., 2011; Ene and Nguyen, 2016; Buchbinder and Feldman, 2016; Sviridenko, 2004), as well as numerous applications - especially in constrained subset selection problems (Powers et al., 2016a; Lin and Bilmes, 2009; Krause and Guestrin, 2005; Krause et al., 2009, 2008a,b) and more broadly machine learning. A typical example is the problem of picking a subset of candidate sensor locations for spatial monitoring of certain phenomena such as temperature, ph values, humidity, etc. (see (Krause et al., 2008a)). Here the goal is typically to find sensor locations that achieve the most coverage or give the most information about the observed phenomena. Submodularity naturally captures the decreasing marginal gain in the coverage, or the information acquired about relevant phenomena by using more sensors, (Das and Kempe, 2008). While submodular optimization offers an attractive model for such scenarios, there are a few key shortcomings, which motivated *robust submodular optimization* (see (Krause et al., 2008a)) in the cardinality case, so as to optimize against several functions *simultaneously*:

1. The sensors are typically used to measure various parameters at the same time. Observations for these parameters need to be modeled via different submodular functions.
2. Many of the phenomena being observed are non-stationary and highly variable in certain locations. To obtain a good solution, a common approach is to use different submodular functions to model different spatial regions.
3. The submodular functions are typically defined using data obtained from observations, and imprecise information can lead to unstable optimization

problems. Thus, there is a desire to compute *solutions that are robust* to perturbations of the submodular functions.

Our main contribution is the development of new algorithms with provable guarantees for robust submodular optimization under a large class of *combinatorial constraints*. These include partition constraints, where local cardinality constraints are placed on disjoint parts of the ground set. More generally, we consider matroid and knapsack constraints. We provide bi-criteria approximations that trade-off the approximation factor with the "size" of the solution, measured by the number ℓ of feasible sets $\{S_i\}_{i\in[\ell]}$ whose union constitutes the final solution S. While this might be nonintuitive at first, it turns out that the union of feasible sets corresponds to an appropriate relaxation of the single cardinality constraint. Some special cases of interest are:

1. *Partition constraints.* Given a partition of the candidate sensor locations, the feasible sets correspond to subsets that satisfy a cardinality constraint on each part of the partition. The union of feasible sets here corresponds to relaxing the cardinality constraints separately for each part. This results in a stronger guarantee than relaxing the constraint globally as would be the case in the single cardinality constraint case.
2. *Gammoid.* Given a directed graph and a subset of nodes T, the feasible sets correspond to subsets S that can reach T via disjoint paths in the graph. Gammoids appear in flow based models, for example in reliable routing. The union of feasible sets now corresponds to sets S that can reach T via paths such that each vertex appears in few paths.

We consider both offline and online versions of the problem, where the data is either known a-priori or is revealed over time, respectively. We give a simple and efficient greedy-like algorithm for the offline version of the problem. The analysis relies on new insights on the performance of the classical greedy algorithm for submodular maximization, when extended to produce a solution comprising of a union of multiple feasible sets. For the online case, we introduce new technical ingredients that might be broadly applicable in online robust optimization. Our work significantly expands on previous works on robust submodular optimization that focused on a single cardinality constraint (Krause et al., 2008a).

1.1 Problem Formulation

As we describe below, we study offline and online variations of *robust submodular maximization under structured combinatorial constraints*. While our results holds for more general constraints, we focus our attention first on *matroid* constraints that generalize the partition as well as the gammoid structural constraints mentioned above. We discuss extensions to other class of constraints in Appendix A.

Consider a non-negative set function $f : 2^V \to \mathbb{R}_+$. We denote the marginal value for any subset $A \subseteq V$ and $e \in V$ by $f_A(e) := f(A + e) - f(A)$, where $A + e := A \cup \{e\}$. Function f is *submodular* if and only if it satisfies the *diminishing returns property*. Namely, for any $e \in V$ and $A \subseteq B \subseteq V\backslash\{e\}$, $f_A(e) \geq f_B(e)$. We say that f is *monotone* if for any $A \subseteq B \subseteq V$, we have $f(A) \leq f(B)$. Most of our results are concerned with optimization of monotone submodular functions.

A natural class of constraints considered in submodular optimization are *matroid* constraints. For a ground set V and a family of sets $\mathcal{I} \subseteq 2^V$, $\mathcal{M} = (V, \mathcal{I})$ is a matroid if (1) for all $A \subset B \subseteq V$, if $B \in \mathcal{I}$ then $A \in \mathcal{I}$ and (2) for all $A, B \in \mathcal{I}$ with $|A| < |B|$, there is $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$. Sets in such a family $\mathcal{I}$ are called *independent* sets, or simply put, *feasible* sets for the purpose of optimization.

We consider the robust variation of submodular optimization. That is, for a matroid $\mathcal{M} = (V, \mathcal{I})$, and a given collection of k monotone submodular functions $f_i : 2^V \to \mathbb{R}_+$ for $i \in [k]$, our goal is to select a set S that maximizes $\min_{i\in[k]} f_i(S)$. We define a $(1-\epsilon)$-approximately optimal solution S as

$$\min_{i\in[k]} f_i(S) \geq (1-\epsilon) \max_{S\in\mathcal{I}} \min_{i\in[k]} f_i(S). \quad (1)$$

We also consider the online variation of the above optimization problem in presence of an adversary. In this setting, we are given a fixed matroid $\mathcal{M} = (V, \mathcal{I})$. At each time step $t \in [T]$, we choose a set S^t. An adversary then selects a collection of k monotone submodular functions $\{f_i^t\}_{i\in[k]} : 2^V \to [0, 1]$. We receive a reward of $\min_{i\in[k]} \mathbb{E}[f_i^t(S^t)]$, where the expectation is taken over any randomness in choosing S^t. We can then use the knowledge of the adversary's actions, i.e., oracle access to $\{f_i^t\}_{i\in[k]}$, in our future decisions. We consider non-adaptive adversaries whose choices $\{f_i^t\}_{i\in[k]}$ are independent of S^τ for $\tau < t$. In other words, an adversarial sequence of functions $\{f_i^1\}_{i\in[k]}, \dots, \{f_i^T\}_{i\in[k]}$ is chosen upfront without being revealed to the optimization algorithm.

Our goal is to design an algorithm that maximizes the total payoff $\sum_{t\in[T]} \min_{i\in[k]} \mathbb{E}[f_i^t(S^t)]$. Thus, we would like to obtain a cumulative reward that competes with that of the fixed set $S \in \mathcal{I}$ we should have played had we known all the functions f_i^t in advance, i.e., compete with $\max_{S\in\mathcal{I}} \sum_{t\in[T]} \min_{i\in[k]} f_i^t(S)$. As in the offline optimization problem, we also consider competing with

$(1-\epsilon)$ fraction of the above benchmark. In this case, $\mathbf{Regret}_{1-\epsilon}(T)$ denotes how far we are from this goal. That is,

$$\mathbf{Regret}_{1-\epsilon}(T) = (1-\epsilon)\cdot\max_{S\in\mathcal{I}}\sum_{t\in[T]}\min_{i\in[k]} f_i^t(S) - \sum_{t\in[T]}\min_{i\in[k]}\mathbb{E}\left[f_i^t(S^t)\right]. \quad (2)$$

We desire algorithms whose $(1-\epsilon)$-regret is sublinear in T. That is, we get arbitrarily close to a $(1-\epsilon)$ fraction of the benchmark as $T\to\infty$.

The offline (Equation 1), or online (Equation 2) variations of robust monotone submodular functions, are known to be NP-hard to approximate to any polynomial factor when the algorithm's choices are restricted to the family of independent sets $\mathcal{I}$ (Krause et al., 2008a). Therefore, to obtain any reasonable approximation guarantee we need to relax the algorithm's constraint set. Such an approximation approach is called a *bi-criteria* approximation scheme in which the algorithm outputs a set with a *nearly optimal objective value*, while ensuring that the set used is the *union of only a few independent sets in* $\mathcal{I}$. More formally, to get a $(1-\epsilon)$-approximate solutions, we may use a set S where $S = S_1\cup\cdots\cup S_\ell$ such that $S_1,\ldots,S_\ell\in\mathcal{I}$ and ℓ is a function of $\frac{1}{\epsilon}$ and other parameters.

1.2 Our Results and Contributions

We present (nearly tight) bi-criteria approximation algorithms for the offline and online variations of robust monotone submodular optimization under matroid constraints. Throughout the paper, we assume that the matroid is accessible via an independence oracle and the submodular functions are accessible via a value oracle. Moreover, we use log to denote logarithm with base 2 and ln to denote the natural logarithm.

For the offline setting of the problem we obtain the following result:

Theorem 1. *For the offline robust submodular optimization problem* (1), *for any* $0<\epsilon<1$, *there is a polynomial time algorithm that runs in* $O\left(nr\log\left(\frac{k}{\epsilon}\right)\log(n)\min\left\{\frac{nk}{\epsilon},\log_{1+\epsilon}(\max_{e,j} f_j(e))\right\}\right)$ *time and returns a set* S^{ALG}, *such that*

$$\min_{i\in[k]} f_i(S^{\text{ALG}}) \geq (1-\epsilon)\cdot\max_{S\in\mathcal{I}}\min_{j\in[k]} f_j(S),$$

where $S^{\text{ALG}} = S_1\cup\cdots\cup S_\ell$ *with* $\ell = O(\log\frac{k}{\epsilon})$, *and* $S_1,\ldots,S_\ell\in\mathcal{I}$.

The algorithm that achieves this result is an extension of the greedy algorithm. It reuses the standard greedy algorithm of Fisher et al. (1978) in an iterative scheme, so that it generates a *small family* of independent sets whose union achieves the $(1-\epsilon)$-guarantee. The argument is reminiscent of a well-known fact for submodular function maximization under cardinality constraints using the greedy algorithm: letting the greedy algorithm run longer results in better approximations at the expense of violating the cardinality constraint. Our extended greedy algorithm works in a similar spirit, however it iteratively produces independent sets in the matroid. We present the main results and the corresponding proofs in Section 2. Additionally, we also propose a second, randomized algorithm relying on continuous extensions of submodular functions that achieves tight bounds in line with the hardness result in (Krause et al., 2008a) (see Appendix B). This algorithm also forms the basis of the online algorithm that we present later in Section 3. One might hope that similar results can be obtained even when functions are non-monotone (but still submodular). As we show in Appendix B.3 this is not possible.

A natural question is whether our algorithm can be carried over into the online setting, where functions are revealed over time. For the online setting, we present the first results for robust submodular optimization.

Theorem 2. *For the online robust submodular optimization problem, for any* $0<\epsilon<1$, *there is a randomized polynomial time algorithm that returns a set* S^t *for each stage* $t\in[T]$, *we get*

$$\sum_{t\in[T]}\min_{i\in[k]}\mathbb{E}\left[f_i^t(S^t)\right] \geq (1-\epsilon)\cdot\max_{S\in\mathcal{I}}\sum_{t\in[T]}\min_{i\in[k]} f_i^t(S) - O\left(n^{\frac{5}{4}}\sqrt{T}\ln\frac{1}{\epsilon}\right),$$

where $S^t = S_1^t\cup\cdots\cup S_\ell^t$ *with* $\ell = O\left(\ln\frac{1}{\epsilon}\right)$, *and* $S_1^t,\ldots,S_\ell^t\in\mathcal{I}$.

We remark that the guarantee of Theorem 2 holds with respect to the minimum of $\mathbb{E}[f_i^t(S^t)]$, as opposed to the guarantee of Theorem 1 that directly bounds the minimum of $f_i(S)$. Therefore, the solution for the online algorithm is a union of only $O\left(\ln\frac{1}{\epsilon}\right)$ independent sets, in contrast to the offline solution which is the union of $O\left(\log\frac{k}{\epsilon}\right)$ independent sets.

The main challenge in the online algorithm is to deal with non-convexity and non-smoothness due to submodularity exacerbated by the robustness criteria. Our approach to coping with the robustness criteria is to use the *soft-min* function $-\frac{1}{\alpha}\ln\sum_{i\in[k]} e^{-\alpha g_i}$, defined for a collection of smooth functions $\{g_i\}_{i\in[k]}$ and a suitable parameter $\alpha>0$. While the choice of the specific soft-min function is seemingly arbitrary, one feature is crucial for us: its gradient is a convex combination of the gradients of the g_i's. Using this observation, we use parallel instances of the Follow-the-Perturbed-Leader (FPL) algorithm, presented by Kalai and Vempala (2005), one for each discretization step in the continuous greedy algorithm. We believe that the algorithm might be of independent interest to perform online learning over a minimum of many functions, a common feature in robust optimization. The main result and a summary of its proof appears in Section 3.

Our main results naturally extend to other types of combinatorial constraints, such as knapsack constraints or multiple matroids. We describe these extensions in the Supplemental Material (Appendix A) due to space restrictions.

1.3 Related Work

Building on the classical work of Nemhauser et al. (1978), constrained submodular maximization problems have seen much progress recently (see for example (Calinescu et al., 2011; Chekuri et al., 2010; Buchbinder et al., 2014, 2016)). Robust submodular maximization generalizes submodular function maximization under a matroid constraint for which a $(1-\frac{1}{e})$-approximation is known (Calinescu et al., 2011) and is optimal. The problem has been studied for constant k by Chekuri et al. (2010) who give a $(1-\frac{1}{e}-\epsilon)$-approximation algorithm with running time $O\left(n^{\frac{k}{\epsilon}}\right)$. Closely related to our problem is the submodular cover problem where we are given a submodular function f, a target $b \in \mathbb{R}_+$, and the goal is to find a set S of minimum cardinality such that $f(S) \geq b$. A simple reduction shows that robust submodular maximization under a cardinality constraint reduces to the submodular cover problem (Krause et al., 2008a). Wolsey (1982) showed that the greedy algorithm gives an $O(\ln \frac{n}{\epsilon})$-approximation, where the output set S satisfies $f(S) \geq (1-\epsilon)b$. Krause et al. (2008a) use this approximation to build a bi-criteria algorithm which achieves tight bounds. However, this approach falls short of achieving a tight bi-criteria approximation when the problem is defined over a matroid. Powers et al. (2016b) considers the same robust problem with matroid constraints. However, they take a different approach by presenting a bi-criteria algorithm that outputs a feasible set that is good only for a fraction of the k monotone submodular functions. A deletion-robust submodular optimization model is presented in (Krause et al., 2008a), which is later studied by Orlin et al. (2016); Bogunovic et al. (2017); Kazemi et al. (2018). Influence maximization (Kempe et al., 2003) in a network has been a successful application of submodular maximization and recently, He and Kempe (2016) and Chen et al. (2016) study the robust influence maximization problem. Robust optimization for non-convex objectives (including submodular functions) has been also considered by Chen et al. (2017), however with weaker guarantees than ours due to the extended generality. Specifically, their algorithm outputs $\frac{r \log k}{\epsilon^2 \, \mathrm{OPT}}$ feasible sets whose union achieves a factor of $(1-1/e-\epsilon)$. Finally, Wilder (2017) studies a similar problem in which the set of feasible solutions is the set of all distributions over independent sets of a matroid. In particular, for our setting Wilder (2017) gives an algorithm that outputs $O(\frac{\log k}{\epsilon^3})$ feasible sets whose union obtains $(1-1/e)^2$ fraction of the optimal solution. Our results are stronger than the ones obtained by Chen et al. (2017) and Wilder (2017), since we provide the same guarantees using the union of fewer feasible sets. Other variants of the robust submodular maximization problem are studied by Mitrovic et al. (2018); Staib et al. (2018).

There has been some prior work on online submodular function maximization that we briefly review here. Streeter and Golovin (2008) study the *budgeted maximum submodular coverage* problem and consider several feedback cases (denote B a integral bound for the budget): in the full information case, a $(1-\frac{1}{e})$-expected regret of $O(\sqrt{BT \ln n})$ is achieved, but the algorithm uses B experts which may be very large. In a follow-up work, Golovin et al. (2014) study the online submodular maximization problem under partition constraints, and then they generalize it to general matroid constraints. For the latter one, the authors present an online version of the continuous greedy algorithm, which relies on the Follow-the-Perturbed-Leader algorithm of Kalai and Vempala (2005) and obtain a $(1-\frac{1}{e})$-expected regret of $O(\sqrt{T})$. Similar to this approach, our bi-criteria online algorithm will also use the Follow-the-Perturbed-Leader algorithm as a subroutine.

2 The Offline Case

In this section, we consider offline robust optimization (Equation 1) under matroid constraints.

2.1 Offline Algorithm and Analysis

In this section, we present a procedure that achieves a (nearly) tight bi-criteria approximation for the problem of interest and prove Theorem 1. First, we extend the standard greedy algorithm for maximizing a *single* submodular function under matroid constraint to the bi-criteria setting and prove Theorem 3.

Observe that Algorithm 1 with $\ell = 1$ is just the greedy algorithm presented by Fisher et al. (1978), which gives a $\frac{1}{2}$-approximation. Extending the standard algorithm gives us the following result.

Theorem 3. *For any $\ell \geq 1$ and monotone submodular function $f : 2^V \to \mathbb{R}_+$ with $f(\emptyset) = 0$, the extended greedy Algorithm 1 returns sets $S_1, \ldots, S_\ell$ such that*

$$f\left(\cup_{\tau=1}^{\ell} S_\tau\right) \geq \left(1 - \frac{1}{2^\ell}\right) \max_{S \in \mathcal{I}} f(S).$$

Algorithm 1 Extended Greedy Algorithm for Submodular Optimization

Input: $\ell \geq 1$, monotone submodular function $f : 2^V \to \mathbb{R}_+$, Matroid $\mathcal{M} = (V, \mathcal{I})$.
Output: sets $S_1, \ldots, S_\ell \in \mathcal{I}$.
1: **for** $\tau = 1, \ldots, \ell$ **do**
2: $S_\tau \leftarrow \emptyset$
3: **while** S_τ is not a basis of $\mathcal{M}$ **do**
4: **Compute**

$$e^* = \mathrm{argmax}_{S_\tau + e \in \mathcal{I}}\, f(\cup_{j=1}^{\tau} S_j + e).$$

5: **Update** $S_\tau \leftarrow S_\tau + e^*$.

Proof. We use the following stronger statement that for any monotone non-negative submodular function (Fisher et al., 1978), the greedy algorithm when run for a single iteration returns a set $S_1 \in \mathcal{I}$ such that $f(S_1) - f(\emptyset) \geq \left(1 - \frac{1}{2}\right) \max_{S \in \mathcal{I}} \{f(S) - f(\emptyset)\}$. We use the above statement to prove our theorem by induction. For $\tau = 1$, the claim follows directly. Consider any $\ell \geq 2$. Observe that the algorithm in iteration $\tau = \ell$, is exactly the greedy algorithm run on submodular function $f' : 2^V \to \mathbb{R}_+$ where $f'(S) := f(S \bigcup \cup_{\tau=1}^{\ell-1} S_\tau)$. This procedure returns S_ℓ such that $f'(S_\ell) - f'(\emptyset) \geq \left(1 - \frac{1}{2}\right) \max_{S \in \mathcal{I}} \left(f'(S) - f'(\emptyset)\right)$, which implies that

$$f\left(\cup_{\tau=1}^{\ell} S_\tau\right) - f\left(\cup_{\tau=1}^{\ell-1} S_\tau\right) \geq \left(1 - \frac{1}{2}\right)\left(\max_{S \in \mathcal{I}} f(S) - f\left(\cup_{\tau=1}^{\ell-1} S_\tau\right)\right).$$

By induction we know $f\left(\cup_{\tau=1}^{\ell-1} S_\tau\right) \geq \left(1 - \frac{1}{2^{\ell-1}}\right) \max_{S \in \mathcal{I}} f(S)$. Thus we obtain

$$f\left(\cup_{\tau=1}^{\ell} S_\tau\right) \geq \frac{1}{2} \max_{S \in \mathcal{I}} f(S) + \frac{1}{2} f\left(\cup_{\tau=1}^{\ell-1} S_\tau\right) \geq \left(1 - \frac{1}{2^\ell}\right) \max_{S \in \mathcal{I}} f(S).$$

□

We now apply Theorem 3 for the robust submodular problem, in which we are given monotone submodular functions $f_i : 2^V \to \mathbb{R}_+$ for $i \in [k]$. First, given parameter $\epsilon > 0$, we obtain an estimate γ on the value of the optimal solution $\mathrm{OPT} := \max_{S \in \mathcal{I}} \min_{i \in [k]} f_i(S)$ via a binary search with a relative error of $1 - \frac{\epsilon}{2}$, i.e., $\left(1 - \frac{\epsilon}{2}\right) \mathrm{OPT} \leq \gamma \leq \mathrm{OPT}$. As in (Krause et al., 2008a), let $g : 2^V \to \mathbb{R}_+$ be defined for any $S \subseteq V$ as follows

$$g(S) := \frac{1}{k} \sum_{i \in [k]} \min\{f_i(S), \gamma\}. \tag{3}$$

Observe that $\max_{S \in \mathcal{I}} g(S) = \gamma$ whenever $\gamma \leq \mathrm{OPT}$. Moreover, note that g is also a monotone submodular function.

Proof of Theorem 1. Consider the family of monotone submodular functions $\{f_i\}_{i \in [k]}$ and define g as in equation (3) considering parameter γ with relative error of $1 - \frac{\epsilon}{2}$. If we run the extended greedy algorithm 1 on g with $\ell \geq \lceil \log \frac{2k}{\epsilon} \rceil$, we get a set $S^{\mathrm{ALG}} = S_1 \cup \cdots \cup S_\ell$, where $S_j \in \mathcal{I}$ for all $j \in [\ell]$. Moreover, Theorem 3 implies that

$$g(S^{\mathrm{ALG}}) \geq \left(1 - \frac{1}{2^\ell}\right) \max_{S \in \mathcal{I}} g(S) \geq \left(1 - \frac{\epsilon}{2k}\right) \gamma.$$

Now, we will prove that $f_i(S^{\mathrm{ALG}}) \geq \left(1 - \frac{\epsilon}{2}\right) \gamma$, for all $i \in [k]$. Assume by contradiction that there exists an index $i^* \in [k]$ such that $f_{i^*}(S^{\mathrm{ALG}}) < \left(1 - \frac{\epsilon}{2}\right) \gamma$. Since, we know that $\min\{f_i(S^{\mathrm{ALG}}), \gamma\} \leq \gamma$ for all $i \in [k]$, then

$$g(S^{\mathrm{ALG}}) \leq \frac{1}{k} f_{i^*}(S^{\mathrm{ALG}}) + \frac{k-1}{k}\gamma < \frac{1 - \epsilon/2}{k}\gamma + \frac{k-1}{k}\gamma = \left(1 - \frac{\epsilon}{2k}\right)\gamma,$$

contradicting $g(S^{\mathrm{ALG}}) \geq \left(1 - \frac{\epsilon}{2k}\right) \gamma$. Therefore, we obtain $f_i(S^{\mathrm{ALG}}) \geq \left(1 - \frac{\epsilon}{2}\right) \gamma \geq (1 - \epsilon)\, \mathrm{OPT}$, for all $i \in [k]$ as claimed. □

Running time analysis. In this section, we study the running time of the bi-criteria algorithm we just presented. To show that a set of polynomial size of values for γ exists such that one of them satisfies $(1 - \epsilon/2)\, \mathrm{OPT} \leq \gamma \leq \mathrm{OPT}$, we simply try $\gamma = n f_i(e)(1 - \epsilon/2)^j$ for all $i \in [k]$, $e \in V$, and $j = 0, \ldots, \lceil \ln_{1-\epsilon/2}(1/n) \rceil$. Note that there exists an index $i^* \in [k]$ and a set $S^* \in \mathcal{I}$ such that $\mathrm{OPT} = f_{i^*}(S^*)$. Now let $e^* = \mathrm{argmax}_{e \in S^*} f_{i^*}(e)$. Because of submodularity and monotonicity we have $\frac{1}{|S^*|} f_{i^*}(S^*) \leq f_{i^*}(e^*) \leq f_{i^*}(S^*)$. So, we can conclude that $1 \geq \mathrm{OPT} / n f_{i^*}(e^*) \geq 1/n$, which implies that $j = \lceil \ln_{1-\epsilon/2}(\mathrm{OPT} / n f_{i^*}(e^*)) \rceil$ is in the correct interval, obtaining

$$(1 - \epsilon/2)\, \mathrm{OPT} \leq n f_{i^*}(e^*)(1 - \epsilon/2)^j \leq \mathrm{OPT}\,.$$

We remark that the dependency of the running time on ϵ can be made logarithmic by running a binary search on j as opposed to trying all $j = 0, \ldots, \lceil \ln_{1-\epsilon/2}(1/n) \rceil$. This would take at most $\frac{nk}{\epsilon} \log n$ iterations. We could also say that doing a binary search to get a value up to a relative error of $1 - \epsilon/2$ of OPT would take $\log_{1+\epsilon} \mathrm{OPT}$. So, we consider the minimum of those two quantities $\min\{\frac{nk}{\epsilon} \log n, \log_{1+\epsilon} \mathrm{OPT}\}$. Given that the extended greedy algorithm runs in $O(nr\ell)$ time, where r is the rank of the matroid and $\ell = O(\log \frac{k}{\epsilon})$ is the number of rounds, we conclude that the bi-criteria algorithm runs in $nr \log \frac{k}{\epsilon} \min\{\frac{nk}{\epsilon} \log n, \log_{1+\epsilon} \mathrm{OPT}\}$.

Continuous offline algorithm. As a final remark, we give a randomized version of the extended greedy based on the continuous extensions of submodular functions. This algorithm outputs a random set S^{ALG} which is the union of $O(\ln \frac{k}{\epsilon})$ independent sets and such that with constant probability has value close to the true optimum. The number of independent sets required for obtaining this result is smaller up to a constant than the number of sets obtained by the extended greedy and optimal given the hardness results. The design of the continuous offline algorithm and its analysis are in Appendix B.

2.2 Experimental results

In this section, we provide a simple computational experiment to exemplify our theoretical guarantees. Moreover, it illustrates that our algorithm performs much better on practical instances, in both the running time as well as degree of the violation of the constraints as compared to the worst-case guarantees given by Theorem 1.

We consider the movie recommendation problem, in which there is a ground set of n movies V and a set of users U. Each user $u \in U$ rate a group of movies, by assigning a value $r_{e,u} \in \{1, \ldots, 5\}$, or zero, if that user did not rate the movie. Our interest is to select a subset of the movies that are the most representative of all users' ratings. To approach this idea, we consider a *facility-location* function, i.e., $f(A) := \frac{1}{5|U|}\sum_{u\in U} \max_{e\in A} r_{e,u}$. Observe that we scale by the maximum rating and the number of users. From this, we consider a collection of monotone submodular functions that are perturbed versions of the facility-location objective, i.e., problem (1) corresponds to $\max_{A\in\mathcal{I}} \min_{i\in[k]}\{f(A) + \sum_{e\in A\cap\Lambda_i} \xi_e\}$, where f is the function defined above, Λ_i is a random set of fixed size different for each $i \in [k]$, and $\xi \sim [0,1]^V$ is an error vector. For experiments, we consider partition constraints. Formally, there is a partition $\{P_1, \ldots, P_q\}$ of the movies and $\mathcal{I} = \{S : \ |S \cap P_j| \leq b, \ \forall j \in [q]\}$ (same budget b for each part). We run the bi-criteria algorithm with the following parameters: number of rounds for the Extended Greedy $\ell = \lceil \log \frac{2k}{\epsilon} \rceil$, and approximation $1 - \epsilon = 0.99$.

We used the MovieLens dataset of Harper and Konstan (2016) with $n = 1,000$ movies and $|U| = 1,000$ users. We consider $k = 20$ objective functions, where the random sets are of size $|\Lambda_i| = 100$. We fixed the number of parts to be $q = 10$ (but not the composition) and the budget $b = 5$. We created 20 random instances in total, where each instance corresponds to a different composition $\{P_1, \ldots, P_q\}$.

An optimal solution to this problem has size $q \cdot b = 50$, and Theorem 1 shows that the bi-criteria algorithm outputs a set that contains at most $b \cdot \lceil \log \frac{2k}{\epsilon} \rceil = 60$ movies in each part (instead of 5), which leads to selecting 600 movies in total. However, in our experimental results we get a much smaller set that on the average has 14.90 movies per part (with a standard deviation of 0.22). We also report results in terms of CPU time and number of function calls in Figure 1. The average CPU time is 21.67 seconds with a standard deviation of 5.22. The average number of function evaluations is $42.79 \cdot 10^4$ with a standard deviation of $7.07 \cdot 10^4$.

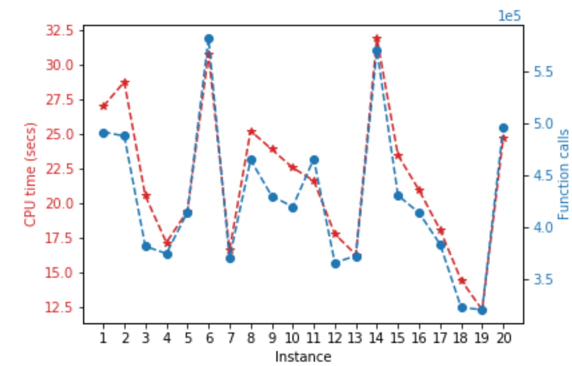

Figure 1: In this figure we report CPU time in seconds (red) and number of function calls (blue) per instance (x-axis)

3 The Online Case

In this section, we consider the online robust optimization problem (Equation 2) under matroid constraints. We introduce an online bi-criteria algorithm that achieves a sublinear $(1-\epsilon)$-regret while using solution S^t at time t that is a union of $O(\ln \frac{1}{\epsilon})$ independent sets from $\mathcal{I}$. To start, let us first present definitions and known results that play a key role in this online optimization problem.

3.1 Background

For a set function f, its *multilinear extension* F : $[0,1]^V \to \mathbb{R}_+$ is defined for any $y \in [0,1]^V$ as the expected value of $f(S_y)$, where S_y is the random set generated by drawing independently each element $e \in V$ with probability y_e. Formally,

$$F(y) = \mathbb{E}_{S\sim y}[f(S)] = \sum_{S\subseteq V} f(S) \prod_{e\in S} y_e \prod_{e\notin S}(1 - y_e). \tag{4}$$

Note that F is an extension of f, since for any subset $S \subseteq V$, we have $f(S) = F(\mathbf{1}_S)$, where $\mathbf{1}_S(e) = 1$ if $e \in S$ and zero otherwise. For all $e \in V$ let

$$\Delta_e F(y) := \mathbb{E}_{S\sim y}[f(S+e) - f(S)] = (1 - y_e)\nabla_e F(y). \tag{5}$$

We use $\Delta F(y)$ to denote the vector whose e^{th}-coordinate is $\Delta_e F(y)$ as defined above. Furthermore, for a matroid $\mathcal{M}$, we denote by $\mathcal{P}(\mathcal{M}) = \text{conv}\{\mathbf{1}_I \mid I \in \mathcal{I}\}$ its matroid polytope. For any $\tau > 0$ we denote by $\tau \cdot \mathcal{P}(\mathcal{M}) = \text{conv}\{\tau \cdot \mathbf{1}_I \mid I \in \mathcal{I}\}$ the scaling of the matroid polytope.

Multilinear extension plays a crucial role in designing approximation algorithms for various constrained submodular optimization problems (see Appendix B.1 for a list of its useful properties). Notably, Vondrák (2008) introduced the *discretized continuous greedy* algorithm that achieves a $1 - 1/e$ approximate solution for maximizing a single submodular function under matroid constraints (see (Feldman et al., 2011) for the variant of the continuous greedy that we use). At a high level, this algorithm discretizes interval $[0, 1]$ into points $\{0, \delta, 2\delta, \ldots, 1\}$. Starting at $y_0 = 0$, for each $\tau \in \{\delta, 2\delta, \ldots, 1\}$ the algorithm uses an LP to compute the direction $z_\tau = \text{argmax}_{z\in\mathcal{P}(\mathcal{M})} \Delta F(y_{\tau-\delta}) \cdot z$. Then the algorithm takes a step in the direction of z_τ by setting $y_{\tau,e} \leftarrow y_{\tau-\delta,e} + \delta z_{\tau,e}(1 - y_{\tau-\delta,e})$ for all $e \in V$. Finally, it outputs a set S by rounding the fractional solution y_1. We will use this discretized version of the continuous greedy to construct our online algorithm in the following section.

3.2 Online Algorithm and Analysis

In Appendix B we provide a continuous randomized algorithm for the offline problem. Broadly speaking, at every step, this algorithm finds a feasible direction that improves all k functions and moves in that direction. We use an LP to find this direction similar to the approach of Vondrák (2008) for the case of $k = 1$. However, for the online robust optimization problem, we immediately face with two challenges. First, it is not clear how to find a feasible direction z_t (as was found via an LP for the offline problem) that is good for all k submodular functions. To resolve this issue, we use a *soft-min* function that converts robust optimization over k functions into optimizing of a single function. Secondly, robust optimization leads to non-convex and non-smooth optimization combined with online arrival of such submodular functions. To deal with this, we use the Follow-the-Perturbed-Leader (FPL) online algorithm introduced by Kalai and Vempala (2005).

For any collection of monotone submodular functions $\{f_i^t\}_{i\in[k]}$ played by the adversary, we define the *soft-min* function with respect to the corresponding multilinear extensions $\{F_i^t\}_{i\in[k]}$ as

$$H^t(y) := -\frac{1}{\alpha} \ln \sum_{i\in[k]} e^{-\alpha F_i^t(y)},$$

where $\alpha > 0$ is a suitable parameter. Recall we assume functions f_i^t taking values in $[0, 1]$, then their multilinear extensions F_i^t also take values in $[0, 1]$. The following properties of the soft-min function as defined above are easy to verify and crucial for our result.

1. Approximation:

$$\min_{i\in[k]} F_i^t(y) - \frac{\ln k}{\alpha} \le H^t(y) \le \min_{i\in[k]} F_i^t(y). \quad (6)$$

2. Gradient:

$$\nabla H^t(y) = \sum_{i\in[k]} p_i^t(y) \nabla F_i^t(y),$$

where $p_i^t(y) \propto e^{-\alpha F_i^t(y)}$ for all $i \in [k]$.

Note that as α increases, the soft-min function H^t becomes a better approximation of $\min_{i\in[k]}\{F_i^t\}_{i\in[k]}$, however, its smoothness degrades (see Property (17) in Appendix C.1). On the other hand, the second property shows that the gradient of the soft-min function is a convex combination of the gradients of the multilinear extensions, which allows us to optimize all the functions at the same time. Indeed, define $\Delta_e H^t(y) := \sum_{i\in[k]} p_i^t(y) \Delta_e F_i^t(y) = (1 - y_e) \nabla_e H^t(y)$. At each stage $t \in [T]$, we use the information from the gradients previously observed, in particular, $\{\Delta H^1, \cdots, \Delta H^{t-1}\}$ to decide the set S^t. To deal with adversarial input functions, we use the FPL algorithm (Kalai and Vempala, 2005) and the following guarantee about the algorithm.

Theorem 4 ((Kalai and Vempala, 2005)). *Let $s_1, \ldots, s_T \in \mathcal{S}$ be a sequence of rewards. The FPL algorithm 4 (see Appendix C.3) with parameter $\eta \le 1$ outputs decisions $d_1, \ldots, d_T$ with regret*

$$\max_{d\in\mathcal{D}} \sum_{t\in[T]} s_t \cdot d - \mathbb{E}\left[\sum_{t\in[T]} s_t \cdot d_t\right] \le O\left(poly(n)\left(\eta T + \frac{1}{T\eta}\right)\right).$$

For completeness, we include the original setup and the algorithm in Appendix C.3.

Our online algorithm works as follows: first, given $0 < \epsilon < 1$ we denote $\ell := \lceil \ln \frac{1}{\epsilon} \rceil$. We consider the following discretization indexed by $\tau \in \{0, \delta, 2\delta, \ldots, \ell\}$ and construct fractional solutions y_τ^t for each iteration t and discretization index τ. At each iteration t, ideally we would like to construct $\{y_\tau^t\}_{\tau=0}^{\ell}$ by running the continuous greedy algorithm using the soft-min function H^t and then play S^t using these fractional solutions. But in the online model, function H^t is revealed only after playing set S^t. To remedy this, we aim to construct y_τ^t using FPL algorithm based on gradients

$\{\nabla H^j\}_{j=1}^{t-1}$ obtained from previous iterations. Thus we have multiple FPL instances, one for each discretization parameter, being run by the algorithm. Finally, at the end of iteration t, we have a fractional vector y_ℓ^t which belongs to $\ell \cdot \mathcal{P}(\mathcal{M}) \cap [0,1]^V$ and therefore can be written, fractionally, as a union of ℓ independent sets using the matroid union theorem (Schrijver, 2003).

We round the fractional solution y_ℓ^t using the randomized swap rounding proposed by Chekuri et al. (2010) for matroid $\mathcal{M}_\ell$ to obtain the set S^t to be played at time t. The following theorem from (Chekuri et al., 2010) gives the necessary property of the randomized swap rounding that we use.

Theorem 5 (Theorem II.1, (Chekuri et al., 2010))**.** *Let f be a monotone submodular function and F be its multilinear extension. Let $x \in \mathcal{P}(\mathcal{M}')$ be a point in the polytope of matroid $\mathcal{M}'$ and S' a random independent set obtained from it by randomized swap rounding. Then, $\mathbb{E}[f(S')] \geq F(x)$.*

Below, we formalize the details in Algorithm 2 (observe that $\ell/\delta \in \mathbb{Z}_+$). Now, we provide a summary of the proof of Theorem 2, for a complete version we refer to Appendix C.2.

Algorithm 2 OnlineSoftMin algorithm

Input: learning parameter $\eta > 0$, $\epsilon > 0$, $\alpha = n^2T^2$, discretization $\delta = n^{-6}T^{-3}$, and $\ell = \lceil \ln \frac{1}{\epsilon} \rceil$.
Output: sequence of sets $S_1, \ldots, S_T$.
1: Sample $q \sim [0, 1/\eta]^V$
2: **for** $t = 1$ to T **do**
3: $y_0^t = 0$
4: **for** $\tau \in \{\delta, 2\delta, \ldots, \ell\}$ **do**
5: **Compute**

$$z_\tau^t = \text{argmax}_{z \in \mathcal{P}(\mathcal{M})} \left[\sum_{j=1}^{t-1} \Delta H^j(y_{\tau-\delta}^j) + q \right] \cdot z.$$

6: **Update** For each $e \in V$,

$$y_{\tau,e}^t = y_{\tau-\delta,e}^t + \delta(1 - y_{\tau-\delta,e}^t) z_{\tau,e}^t.$$

7: **Play** $S^t \leftarrow$ SwapRounding (y_ℓ^t). **Receive** and **observe** new collection $\{f_i^t\}_{i \in [k]}$.

Proof of Theorem 2, summary. By applying a simple Taylor approximation and using the update rule, for any $\tau \in \{\delta, \ldots, \ell\}$ we have

$$\sum_{t \in [T]} H^t(y_\tau^t) - H^t(y_{\tau-\delta}^t) \geq \delta \sum_{t \in [T]} \Delta H^t(y_{\tau-\delta}^t) \cdot z_\tau^t - O(Tn^3\delta^2\alpha). \quad (7)$$

Observe in Algorithm 2 that a different FPL is implemented for each $\tau \in \{\delta, \ldots, \ell\}$, so we can state a regret bound for each τ by using Theorem 4. Specifically,

$$\mathbb{E}\left[\sum_{t \in [T]} \Delta H^t(y_{\tau-\delta}^t) \cdot z_\tau^t\right] \geq \max_{z \in \mathcal{P}(\mathcal{M})} \mathbb{E}\left[\sum_{t \in [T]} \Delta H^t(y_{\tau-\delta}^t) \cdot z\right] - R_\eta,$$

where R_η is the regret guarantee for a given learning parameter $\eta > 0$. By taking expectation in (7) and using the regret bound we just mentioned, we obtain

$$\mathbb{E}\left[\sum_{t \in [T]} H^t(y_\tau^t) - H^t(y_{\tau-\delta}^t)\right] \geq \delta \left(\max_{z \in \mathcal{P}(\mathcal{M})} \mathbb{E}\left[\sum_{t \in [T]} \Delta H^t(y_{\tau-\delta}^t) \cdot z\right]\right) - \delta R_\eta - O(Tn^3\delta^2\alpha). \quad (8)$$

After using submodularity and monotonicity of each f_i^t (specifically Fact 1 in Appendix B.1), and properties of the soft-min function, we arrive to the following recurrence for each $\tau \in \{\delta, \ldots, \ell\}$

$$\mathbb{E}\left[\sum_{t \in [T]} H^t(y_\tau^t)\right] - \mathbb{E}\left[\sum_{t \in [T]} H^t(y_{\tau-\delta}^t)\right] \geq \delta \left(\sum_{t \in [T]} H^t(x^*) - \mathbb{E}\left[\sum_{t \in [T]} H^t(y_{\tau-\delta}^t)\right]\right) - 2\delta R_\eta,$$

where x^* is the optimal solution of $\max_{x \in \mathcal{P}(\mathcal{M})} \sum_{t \in [T]} \min_{i \in [k]} F_i^t(x)$. This recurrence is similar to the one shown in (Vondrák, 2008) for the discretized continuous greedy. Then, by iterating $\frac{\ell}{\delta}$ times in τ, we get

$$\mathbb{E}\left[\sum_{t \in [T]} H^t(y_\ell^t)\right] - \mathbb{E}\left[\sum_{t \in [T]} H^t(y_0^t)\right] \geq (1-\epsilon) \sum_{t \in [T]} H^t(x^*) - O\left(R_\eta \ln \frac{1}{\epsilon}\right),$$

The term $\mathbb{E}\left[\sum_{t \in [T]} H^t(y_0^t)\right]$ is small, so it is bounded by $O\left(R_\eta \ln \frac{1}{\epsilon}\right)$. Since α is sufficiently large, we can apply property (6) of the soft-min to obtain

$$\mathbb{E}\left[\sum_{t \in [T]} \min_{i \in [k]} F_i^t\left(y_\ell^t\right)\right] \geq (1-\epsilon) \cdot \sum_{t \in [T]} \min_{i \in [k]} F_i^t(x^*) - O\left(R_\eta \ln \frac{1}{\epsilon}\right).$$

Finally, by doing swap rounding on each y_ℓ^t, and applying Theorem 5, we get the desired result. ☐

Acknowledgements

This research was partially supported by NSF Award CCF-1717947, NSF CAREER Award CMMI-1452463, an IBM Ph.D. fellowship, and a Microsoft Research Ph.D. fellowship.

References

Ilija Bogunovic, Slobodan Mitrovic, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 508–516, 2017.

Niv Buchbinder and Moran Feldman. Deterministic algorithms for submodular maximization problems. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 392–403, 2016.

Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1433–1452, 2014.

Niv Buchbinder, Moran Feldman, and Roy Schwartz. Comparing apples and oranges: Query trade-off in submodular maximization. *Mathematics of Operations Research*, 42(2):308–329, 2016.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the 51st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 575–584, 2010.

Robert S. Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 4708–4717, 2017.

Wei Chen, Tian Lin, Zihan Tan, Mingfei Zhao, and Xuren Zhou. Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 795–804, 2016.

Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the 40th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 45–54, 2008.

Alina Ene and Huy L. Nguyen. Constrained submodular maximization: Beyond 1/e. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 248–257, 2016.

Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 570–579, 2011.

Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. An analysis of approximations for maximizing submodular set functions—ii. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.

Daniel Golovin, Andreas Krause, and Matthew Streeter. Online submodular maximization under a matroid constraint with application to learning assignments. Technical Report, arXiv, 2014.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4): 19:1–19, 2016.

Xinran He and David Kempe. Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 885–894, 2016.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 2544–2553, 2018.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2003.

Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 324–331, 2005.

Andreas Krause, H. Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9:2761–2801, 2008a.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008b.

Andreas Krause, Ram Rajagopal, Anupam Gupta, and Carlos Guestrin. Simultaneous placement and scheduling of sensors. In *Proceedings of the 8th*

ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN), pages 181–192, 2009.

Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the 41st Annual ACM Symposium on the Theory of Computing (STOC)*, pages 323–332, 2009.

Hui Lin and Jeff A. Bilmes. How to select a good training-data subset for transcription: submodular active selection for sequences. In *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2859–2862, 2009.

Marko Mitrovic, Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Data summarization at scale: A two-stage submodular approach. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 3593–3602, 2018.

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

James B. Orlin, Andreas S. Schulz, and Rajan Udwani. Robust monotone submodular function maximization. In *Proceedings of the 18th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 312–324, 2016.

Thomas Powers, Jeff Bilmes, David W. Krout, and Les Atlas. Constrained robust submodular sensor selection with applications to multistatic sonar arrays. In *Proceedings of the 19th International Conference on Information Fusion (FUSION)*, pages 2179–2185, 2016a.

Thomas Powers, Jeff Bilmes, Scott Wisdom, David W Krout, and Les Atlas. Constrained robust submodular optimization. In *NIPS OPT2016 workshop*, 2016b.

Alexander Rakhlin. Lecture notes on online learning. *Draft, April*, 2009.

Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

Matthew Staib, Bryan Wilder, and Stefanie Jegelka. Distributionally robust submodular maximization. *CoRR*, abs/1802.05249, 2018.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Proceedings of the 21st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1577–1584, 2008.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 67–74, 2008.

Bryan Wilder. Equilibrium computation and robust optimization in zero sum games with submodular structure. arXiv, 2017.

Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.