Structured Robust Submodular Maximization: Offline and Online Algorithms

Nima Anari* Nika Haghtalab† Joseph (Seffi) Naor‡
Sebastian Pokutta§ Mohit Singh¶ Alfredo Torrico‖

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.

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,c) 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*:

*Stanford University, Stanford. Email: anari@berkeley.edu
†Carnegie Mellon University, Email: nhaghtal@cs.cmu.edu
‡Technion, Haifa, Israel. Email: naor@cs.technion.ac.il
§Georgia Institute of Technology, Atlanta. Email: sebastian.pokutta@isye.gatech.edu
¶Georgia Institute of Technology, Atlanta. Email: mohitsinghr@gmail.com
‖Georgia Institute of Technology, Atlanta. Email: atorrico3@gatech.edu

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 Section 4.

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

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

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 $\{f_i^1\}_{i\in[k]}, \ldots, \{f_i^T\}_{i\in[k]}$ 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, namely, 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, $\textbf{Regret}_{1-\epsilon}(T)$ denotes how far we are from this goal. That is,

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

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*

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

where $S^{\mathrm{ALG}} = S_1 \cup \cdots \cup S_\ell$ *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 Section 2.3). 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 Section 2.4 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 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 Section 4.

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. (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., 2008b), which is later studied by (Orlin et al., 2016). 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 (Chen et al., 2017; Wilder, 2017), however with weaker guarantees than ours due to the extended generality. Particularly, (Chen et al., 2017) follows the same bi-criteria approach than us, in the case of submodular objectives.

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 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*

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 $S_1, \ldots, S_\ell \in \mathcal{I}$.

```
for τ = 1, ..., ℓ do
    S_τ ← ∅
    while S_τ is not a basis of M do
        Compute e* = argmax_{S_τ + e ∈ I} f(∪_{j=1}^{τ} S_j + e).
        Update S_τ ← S_τ + e*.
```

extended greedy Algorithm 1 returns sets $S_1, \ldots, S_\ell$ such that

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

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., $\left(1 - \frac{\epsilon}{2}\right) \mathrm{OPT} \leq \gamma \leq \mathrm{OPT}$. 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. 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 can also do a binary search to get a value up to a relative error of $1 - \epsilon/2$ of $\max_{e,j} n \cdot f_j(e)$ and this would take $O(\log_{1+\epsilon} \max_{e,i} n \cdot f_i(e))$ iterations. So, we consider the minimum of those two quantities $\log n \min\{\frac{nk}{\epsilon}, \log_{1+\epsilon} \max_{e,j} f_j(e)\}$. 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 $O(nr \log \frac{k}{\epsilon} \log(n) \min\{\frac{nk}{\epsilon} \log n, \log_{1+\epsilon} \max_{e,j} f_j(e)\})$.

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 the 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 $1-\epsilon=0.99$.

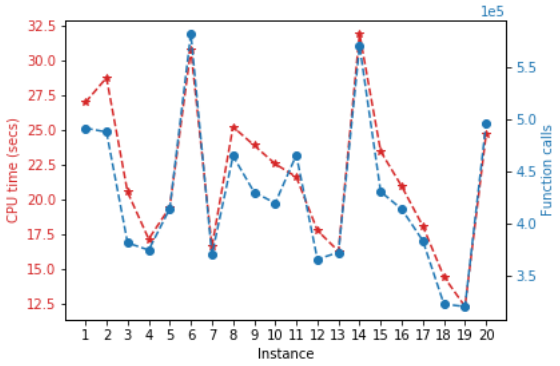

Figure 1: In this figure we report CPU time in seconds (red) and number of function calls (blue) per instance (x-axis)

We used the MovieLens dataset of (Harper and Konstan, 2015) 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 $42.79\cdot10^4$ with a standard deviation of $7.07\cdot10^4$.

2.3 Continuous Offline Algorithm

In this section, we construct a continuous randomized version of the extended greedy that achieves optimal bounds. This algorithm outputs a random set S^{ALG} which is the union of $O(\ln\frac{k}{\epsilon})$ independent sets (improving over the $O(\log\frac{k}{\epsilon})$ independent sets from above) and such that with constant probability has value close to the true optimum. This number of independent sets is tight (see Krause et al. (2008a)), but the algorithm is not as simple as the extended greedy procedure presented above. The main result of this section is Theorem 4, but first we give some preliminary concepts.

2.3.1 Preliminaries of Multilinear Extension

For a non-negative 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,

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

Observe, this is in fact 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 0 otherwise. For all $e \in V$ we define also the following expression:

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

Now, we state some facts about the multilinear extension when the set function is monotone and submodular. For more details see Calinescu et al. (2011).

Fact 1. *[Multilinear Extensions of Monotone Submodular Functions] Let f be a monotone submodular function and F its multilinear extension.*

1. *By monotonicity of f, we have $\frac{\partial F}{\partial y_e} \geq 0$ for any $e \in V$. This implies that for any $x \leq y$ coordinate-wise, $F(x) \leq F(y)$. On the other hand, by submodularity of f, F is concave in any positive direction, i.e., for any $e_1, e_2 \in V$ we have $\frac{\partial^2 F}{\partial y_{e_1} \partial y_{e_2}} \leq 0$.*

2. *Throughout the paper we will denote by $\nabla_e F(y) := \frac{\partial F(y)}{\partial y_e}$, and $\Delta_e F(y) := \mathbb{E}_{S \sim y}[f_S(e)]$. It is easy to see that $\Delta_e F(y) = (1 - y_e)\nabla_e F(y)$. Now, consider two points $x, y \in [0, 1]^V$ and two sets sampled independently from these vectors: $S \sim x$ and $U \sim y$. Then, by submodularity*

$$f(S \cup U) \leq f(S) + \sum_{e \in V} \mathbf{1}_U(e) f_S(e). \tag{6}$$

3. *By taking expectation over x and y in* (6)*, we obtain*

$$F(x \vee y) \leq F(x) + \sum_{e \in V} y_e \Delta_e F(x) \leq F(x) + \sum_{e \in V} y_e \nabla_e F(x).$$

Therefore, we get the following important property

$$F(x \vee y) \leq F(x) + y \cdot \nabla F(x). \tag{7}$$

Finally, we denote the matroid polytope by $\mathcal{P}(\mathcal{M}) = \text{conv}\{\mathbf{1}_I \mid I \in \mathcal{I}\}$ and for any $\tau > 0$ let $\tau \cdot \mathcal{P}(\mathcal{M}) = \text{conv}\{\tau \cdot \mathbf{1}_I \mid I \in \mathcal{I}\}$ be the τ-scaling of the matroid polytope.

2.3.2 Algorithm Analysis

In this section we prove the following theorem.

Theorem 4. *Let $(V, \mathcal{I})$ be a matroid and let $f_i : 2^V \to \mathbb{R}_+$ be a monotone submodular function for $i \in [k]$. Then, there is a randomized polynomial time algorithm that with constant probability returns a set S^{ALG}, such that for all $i \in [k]$, for a given $0 < \epsilon < 1$,*

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

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

Our overall approach is to first find a fractional solution with a desirable approximation guarantee and then round it to an integral solution. We use a relaxation of a matroid to its convex hull to accommodate the search for a fractional solution.

For this algorithm, we need an estimate γ of the value of the optimal solution which we denote by OPT. We prove the following lemma which solves an approximate decision version of our optimization problem. The proof of Theorem 4 follows from the lemma and a search over an approximate value for OPT.

Lemma 1. *There is a randomized polynomial time algorithm that given $\gamma \leq \mathrm{OPT}$ and $0 < \epsilon < 1$ returns with constant probability a set S^{ALG} such that for all $i \in [k]$,*

$$f_i(S^{\mathrm{ALG}}) \geq (1-\epsilon) \cdot \gamma,$$

where $S^{\mathrm{ALG}} = \bigcup_{j \in [\ell]} S_j$ with $\ell = O(\ln \frac{k}{\epsilon})$ and $S_j \in \mathcal{I}$ for each $j \in [\ell]$.

First, we finish the proof of Theorem 4 assuming Lemma 1.

Proof of Theorem 4. We apply the algorithm from Lemma 1 with approximation loss $\epsilon/2$ and with different values of γ, some of which may be larger than OPT, but at least one of them is guaranteed to satisfy $(1-\epsilon/2)\,\mathrm{OPT} \leq \gamma \leq \mathrm{OPT}$. At the end we return the set S^{ALG} from our runs with the highest value of $\min_{i \in [k]} f_i(S^{\mathrm{ALG}})$.

Before describing the set of candidate values of γ that we try, note that if the algorithm succeeds for the particular value of γ satisfying $(1-\epsilon/2)\,\mathrm{OPT} \leq \gamma \leq \mathrm{OPT}$, then we get

$$\min_{i \in [k]} f_i(S^{\mathrm{ALG}}) \geq (1-\epsilon/2) \cdot \gamma \geq (1-\epsilon)\,\mathrm{OPT},$$

and since we return the set with the highest $\min_{i \in [k]} f_i(S^{\mathrm{ALG}})$, the algorithm's output will have the desired approximation guarantee. The existence of such γ follows from the running time analysis made in Section 2.1, and this finishes the proof.

□

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 $j = 0, \ldots, \lceil \ln_{1-\epsilon/2}(1/n) \rceil$. We just need to run the algorithm from Lemma 1 for each γ polynomially many times to make the failure probability exponentially small whenever $\gamma \leq \mathrm{OPT}$.

The rest of this section is devoted to prove Lemma 1. To achieve a strong concentration bound when rounding the fractional solution, we truncate f_i to $\min\{\gamma, f_i\}$. Hereafter, we use f_i^γ to refer to $\min\{\gamma, f_i\}$. Note that submodularity and monotonicity is preserved under this truncation. Also, we denote by F_i^γ the corresponding multilinear extension of f_i^γ.

We describe the continuous process counterpart of the algorithm in this section. The discretization details follow using standard methods Vondrák (2008).

Continuous Greedy. We start a continuous gradient step process where $y(\tau)$ represents the point at time τ we are at. We start at $y(0) = 0$ and take continuous gradient steps in direction $\frac{dy}{d\tau} = v_{\mathrm{all}}(y)$, such that $v_{\mathrm{all}}(y)$ satisfies the following conditions:

(a) $v_{\mathrm{all}}(y) \cdot \nabla F_i^\gamma(y) \geq \gamma - F_i^\gamma(y)$ for all $i \in [k]$,

(b) $v_{\mathrm{all}}(y) \in \mathcal{P}(\mathcal{M})$, and

(c) $v_{\mathrm{all}}(y) + y \in [0,1]^V$.

First, we show that such v_{all} always exists. Take x^* to be the indicator vector corresponding to the optimal solution. For any y, $v^* = (x^* - y) \vee 0$ is a positive direction satisfying inequality (7), and for all $i \in [k]$:

$$v^* \cdot \nabla F_i^\gamma(y) \geq F_i^\gamma(y + v^*) - F_i^\gamma(y) = \gamma - F_i^\gamma(y), \tag{8}$$

where the last equality holds since $F_i^{\gamma}(y) \leq \gamma$ for all y. It is easy to check that v^* satisfies the rest of the constraints (a)-(c), implying that there exists a feasible solution to the above system of linear inequalities. Therefore, we can solve a linear program defined by these inequalities to obtain a solution $v_{\text{all}}(y)$.

The above continuous process goes on until time $\ell = O(\ln \frac{k}{\epsilon})$. We intentionally set $\ell > 1$ to obtain a (fractional) solution with a higher budget, which is useful for achieving a bi-criteria approximation. Next we show the following claim.

Claim 1. *For any $\tau \geq 0$, $y(\tau) \in \tau\mathcal{P}(\mathcal{M}) \cap [0,1]^V$ and for all $i \in [k]$,*

$$F_i^{\gamma}(y(\tau)) \geq (1 - e^{-\tau})\gamma.$$

Proof. For any $\tau \geq 0$, we have

$$y(\tau) = \int_0^{\tau} v_{\text{all}}(y(s))\, ds = \int_0^{1} \tau \cdot v_{\text{all}}(y(\tau s))\, ds.$$

So, $y(\tau)$ is a convex combination of vectors in $\tau\mathcal{P}(\mathcal{M})$. Moreover, $(v_{\text{all}}(y))_j = 0$ when $y_j = 1$, thus $y(\tau) \in [0,1]^V$ proving the first part of the claim.

For the second part, observe that for all $i \in [k]$ we have

$$\frac{dF_i^{\gamma}(y(\tau))}{d\tau} = \frac{dy(\tau)}{d\tau} \cdot \nabla F_i^{\gamma}(y(\tau)) = v_{\text{all}}(y(\tau)) \cdot \nabla F_i^{\gamma}(y(\tau)) \geq \gamma - F_i^{\gamma}(y(\tau)).$$

Moreover, $F_i^{\gamma}(0) = 0$. Now we solve the above differential equation to obtain

$$F_i^{\gamma}(y(\tau)) \geq (1 - e^{-\tau})\gamma$$

for each $i \in [k]$ as claimed. □

Thus, by setting $\ell = \ln \frac{k}{\epsilon} + \ln \frac{1}{c}$, we obtain $F_i^{\gamma}(y(\ell)) \geq \left(1 - \frac{\epsilon}{k} \cdot c\right) \cdot \gamma$ for all $i \in [k]$ and a desired constant $c < 1$. We next show how to obtain an integral solution.

Rounding. The next lemma summarizes our rounding. We first show that the fractional solution at time ℓ is contained in the matroid polytope of the ℓ-fold union of matroid $\mathcal{M}$. We then do randomized swap rounding, introduced by (Chekuri et al., 2010), in this matroid polytope. The truncation of the submodular functions, as well as properties of randomized swap rounding, play a crucial role in the proof.

Lemma 2. *Let $\ell = \lceil \ln \frac{k}{\epsilon} + \ln \frac{1}{c} \rceil$ be an integer and $y(\ell)$ be the output of the continuous greedy algorithm at time ℓ such that $F_i^{\gamma}(y(\ell)) \geq \left(1 - \frac{\epsilon}{k} \cdot c\right) \cdot \gamma$ for each $i \in [k]$ and some constant $c < 1$. Then, there exists a polynomial time randomized algorithm that outputs a set S such that with probability $\Omega(1)$, for each $i \in [k]$ we have*

$$f_i(S) \geq (1 - \epsilon) \cdot \gamma.$$

Moreover, S is a union of at most ℓ independent sets in $\mathcal{M}$.

Proof. Let $\mathcal{M}_\ell = \bigvee_\ell \mathcal{M}$ be the ℓ-fold union of matroid $\mathcal{M}$, i.e., I is an independent set in $\mathcal{M}_\ell$ if and only if I is a union of ℓ independent sets of $\mathcal{M}$. We denote by $\mathcal{I}_\ell$ the set of independent sets of $\mathcal{M}_\ell$. The rank function of $\mathcal{M}_\ell$ is given by $r_{\mathcal{M}_\ell}(S) = \min_{A \subseteq S} |S \setminus A| + \ell \cdot r_{\mathcal{M}}(A)$ (see (Schrijver, 2003)). We first show that $y = y(\ell)$ is in the convex hull of independent sets of matroid $\mathcal{M}_\ell$, i.e., $\mathcal{P}(\mathcal{M}_\ell)$. This polytope is given by $\mathcal{P}(\mathcal{M}_\ell) = \{x \in \mathbb{R}_+^V \mid \ x(S) \leq r_{\mathcal{M}_\ell}(S) \quad \forall\ S \subseteq V\}$, where $x(S) = \sum_{e \in S} x_e$. We now prove that $y \in \mathcal{P}(\mathcal{M}_\ell)$. For any $S \subseteq V$ and $A \subseteq S$, we have $y(S) = \sum_{e \in S \setminus A} y_e + y(A) \leq |S \setminus A| + \ell \cdot r_{\mathcal{M}}(A)$, where the last inequality is due to the fact that $y_e \leq 1$ for all e, and $y(A) \leq \ell \cdot r_{\mathcal{M}}(A)$ because $y \in \ell \cdot \mathcal{P}(\mathcal{M})$ by Claim 1. Therefore, $y \in \mathcal{P}(\mathcal{M}_\ell)$.

Next, we apply a randomized swap rounding (see (Chekuri et al., 2010)) in matroid $\mathcal{M}_\ell$ to obtain the solution. A feature of the randomized swap rounding is that it is oblivious to the specific function f_i used, and it is only a randomized function of the matroid space and the fractional solution. In (Chekuri et al., 2010) authors prove the following result.

Theorem 5 (Theorem II.1 of (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, $\mathbb{E}[f(S')] \geq F(x)$.*

Applying Theorem 5 to fractional solution $y(\ell)$ and matroid $\mathcal{M}_\ell$, we obtain a random set $S \in \mathcal{I}_\ell$ such that

$$\mathbb{E}[f_i^\gamma(S)] \geq F_i^\gamma(y(\ell)) \geq \left(1 - \frac{\epsilon}{k} \cdot c\right) \cdot \gamma$$

for all $i \in [k]$.

Due to the initial truncation, we have that $f_i^\gamma(S) \leq \gamma$ with probability one. Thus, using Markov's inequality for each $i \in [k]$, we obtain that with probability at least $1 - \frac{c}{k}$, we have $f_i^\gamma(S) \geq (1-\epsilon)\gamma$. Therefore, taking a union bound over k functions, we obtain $f_i^\gamma(S) \geq (1-\epsilon)\gamma$ for all $i \in [k]$ with probability at least $1 - c$, and since $f_i(S) \geq f_i^\gamma(S)$ we get an integral solution S with max-min value at least $(1-\epsilon)\gamma$ as claimed. □

2.4 Necessity of monotonicity

In light of the approximation algorithms for non-monotone submodular function maximization under matroid constraints (see, for example, (Lee et al., 2009)), one might hope that an analogous bi-criteria approximation algorithm could exist for robust non-monotone submodular function maximization. However, we show that even without any matroid constraints, getting any approximation in the non-monotone case is NP-hard.

Lemma 3. *Unless $P = NP$, no polynomial time algorithm can output a set $\tilde{S} \subseteq V$ given general submodular functions $f_1, \ldots, f_k$ such that $\min_{i \in [k]} f_i(\tilde{S})$ is within a positive factor of $\max_{S \subseteq V} \min_{i \in [k]} f_i(S)$.*

Proof. We use a reduction from SAT. Suppose that we have a SAT instance with variables $x_1, \ldots, x_n$. Consider $V = \{1, \ldots, n\}$. For every clause in the SAT instance we introduce a nonnegative linear (and therefore submodular) function. For a clause $\bigvee_{i \in A} x_i \vee \bigvee_{i \in B} \overline{x_i}$ define

$$f(S) := |S \cap A| + |B \setminus S|.$$

It is easy to see that f is linear and nonnegative. If we let S be the set of true variables in a truth assignment, then it is easy to see that $f(S) > 0$ if and only if the corresponding clause is satisfied. Consequently, finding a set S such that all functions f corresponding to different clauses are positive is as hard as finding a satisfying assignment for the SAT instance. □

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 properties and known results that play a key role in this online optimization problem.

3.1 Background

Submodular maximization. Multilinear extension plays a crucial role in designing approximation algorithms for various constrained submodular optimization problems (see Section 2.3.1 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 $z_\tau = \operatorname{argmax}_{z\in\mathcal{P}(\mathcal{M})} \Delta F(y_{\tau-\delta}) \cdot z$. 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.

Properties of the soft-min function. Consider a set of k twice differentiable, real-valued functions $g_1,\ldots,g_k$. Let g_{min} be the minimum among these functions, i.e., for each point x in the domain, define $g_{min}(x) := \min_{i\in[k]} g_i(x)$. This function can be approximated by using the so-called *soft-min* function H defined as

$$H(x) = -\frac{1}{\alpha}\ln\sum_{i\in[k]} e^{-\alpha g_i(x)},$$

where $\alpha > 0$ is a fixed parameter. We now present some of the key properties of this function in the following lemma.

Lemma 4. *For any set of k twice differentiable, real-valued functions $g_1,\ldots,g_k$, the soft-min function H satisfies the following properties:*

1. *Bounds:*
$$g_{min}(x) - \frac{\ln k}{\alpha} \le H(x) \le g_{min}(x). \tag{9}$$

2. *Gradient:*
$$\nabla H(x) = \sum_{i\in[k]} p_i(x)\nabla g_i(x), \tag{10}$$
where $p_i(x) := e^{-\alpha g_i(x)} / \sum_{j\in[k]} e^{-\alpha g_j(x)}$. Clearly, if $\nabla g_i \ge 0$ for all $i \in [k]$, then $\nabla H \ge 0$.

3. *Hessian:*
$$\frac{\partial^2 H(x)}{\partial x_{e_1}\partial x_{e_2}} = \sum_{i\in[k]} p_i(x)\left(-\alpha\frac{\partial g_i(x)}{\partial x_{e_1}}\frac{\partial g_i(x)}{\partial x_{e_2}} + \frac{\partial^2 g_i(x)}{\partial x_{e_1}\partial x_{e_2}}\right) + \alpha\nabla_{e_1}H(x)\cdot\nabla_{e_2}H(x) \tag{11}$$
Moreover, if for all $i \in [k]$ we have $\left|\frac{\partial g_i}{\partial x_{e_1}}\right| \le L_1$, and $\left|\frac{\partial^2 g_i}{\partial x_{e_1}\partial x_{e_2}}\right| \le L_2$, then $\left|\frac{\partial^2 H}{\partial x_{e_1}\partial x_{e_2}}\right| \le 2\alpha L_1^2 + L_2$.

4. Comparing the average of the g_i functions with H: given $T > 0$ we have

$$H(x) \leq \sum_{i\in[k]} p_i(x)g_i(x) \leq H(x) + \frac{n + \ln T}{\alpha} + \frac{\ln k}{\alpha} + \frac{ke^{-n}}{T}. \tag{12}$$

So, for $\alpha > 0$ sufficiently large $\sum_{i\in[k]} p_i(x)g_i(x)$ is a good approximation of $H(x)$.

Now, we present a lemma which is used to prove Theorem 2. This is done via a simple Taylor approximation.

Lemma 5. *Fix a parameter $\delta > 0$. Consider T collections of k twice-differentiable functions, namely $\{g_i^1\}_{i\in[k]}, \ldots, \{g_i^T\}_{i\in[k]}$. Assume $0 \leq g_i^t(x) \leq 1$ for any x in the domain, for all $t \in [T]$ and $i \in [k]$. Define the corresponding sequence of soft-min functions $H^1, \ldots, H^T$, with a common parameter $\alpha > 0$. Then, any two sequences of points $\{x^t\}_{t\in[T]}, \{y^t\}_{t\in[T]} \subseteq [0, 1]^V$ with $|x^t - y^t| \leq \delta$ satisfy*

$$\sum_{t\in[T]} H^t(y^t) - \sum_{t\in[T]} H^t(x^t) \geq \sum_{e\in V} \sum_{t\in[T]} \nabla_e H^t(x^t)(y_e^t - x_e^t) - O(Tn^3\delta^2\alpha).$$

For a proof of these lemmas, we refer to Appendix A.

3.2 Online Algorithm and Analysis

Turning our attention to the online robust optimization problem, we are immediately faced with two challenges. First, we need to find a direction z_t (as was found via an LP for a single function) that is good for all k submodular functions, not just one of them. 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. The following properties of the soft-min function as defined in the previous section are easy to verify and crucial for our result.

1. Approximation: $\min_{i\in[k]} F_i^t(y) - \dfrac{\ln k}{\alpha} \leq H^t(y) \leq \min_{i\in[k]} F_i^t(y)$.
2. Gradient: $\nabla H^t(y) = \sum_{i\in[k]} p_i^t(y)\nabla F_i^t(y)$, where $p_i^t(y) \propto e^{-F_i^t(y)}$ 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 (11) in the previous section). 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 6 ((Kalai and Vempala, 2005)). *Let $s_1, \ldots, s_T \in \mathcal{S}$ be a sequence of rewards. The FPL algorithm 4 (see Appendix B) with parameter $\eta \leq 1$ outputs decisions $d_1, \ldots, d_T$ with regret*

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

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

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. Theorem 5 from (Chekuri et al., 2010) gives the necessary property of the randomized swap rounding that we use.

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 $S_1, \ldots, S_T$.

Sample $q \sim [0, 1/\eta]^V$
for $t = 1$ to T **do**
  $y_0^t = 0$
  for $\tau \in \{\delta, 2\delta, \ldots, \ell\}$ **do**
    Compute $z_\tau^t = \operatorname{argmax}_{z \in \mathcal{P}(\mathcal{M})} \left[\sum_{j=1}^{t-1} \Delta H^j(y_{\tau-\delta}^j) + q\right] \cdot z$
    Update $y_{\tau,e}^t = y_{\tau-\delta,e}^t + \delta(1 - y_{\tau-\delta,e}^t) z_{\tau,e}^t$ for each $e \in V$.
  Play $S^t \leftarrow$ SwapRounding $\left(y_\ell^t\right)$. **Receive** and **observe** new collection $\{f_i^t\}_{i \in [k]}$.

We note that FPL was shown to be useful for online optimization of a single submodular function (Golovin et al., 2014). Therefore, our technique can be seen as a generalization of this result. The presence of the soft-min function of the multi-linear functions rather than a single multi-linear function presents technical challenges. We deal with them by using the above described properties of soft-min function and appropriate setting of the discretization parameter η and the parameter defining the soft-min function α.

In order to get sub-linear regret for the FPL algorithm, (Kalai and Vempala, 2005) assume a couple of conditions on the problem (see Appendix B). Similarly, for our online model we need to consider the following for any $t \in [T]$:

1. bounded diameter of $\mathcal{P}(\mathcal{M})$, i.e., for all $y, y' \in \mathcal{P}(\mathcal{M})$, $\|y - y'\|_1 \leq D$;
2. for all $x, y \in \mathcal{P}(\mathcal{M})$, we require $\left|y \cdot \Delta H^t(x)\right| \leq L$;

3. for all $y \in \mathcal{P}(\mathcal{M})$, we require $\|\Delta H^t(y)\|_1 \leq A$,

Now, we give a complete proof of Theorem 2 for any given learning parameter $\eta > 0$, but the final result follows with $\eta = \sqrt{D/LAT}$ and assuming $L \leq n$, $A \leq n$ and $D \leq \sqrt{n}$, which gives a $O(n^{5/4})$ dependency on the dimension in the regret.

Proof of Theorem 2. Consider the sequence of multilinear extensions $\{F_i^1\}_{i\in[k]}, \ldots, \{F_i^T\}_{i\in[k]}$ derived from the monotone submodular functions f_i^t obtained during the dynamic process. Since f_i^t's have value in $[0,1]$, we have $0 \leq F_i^t(y) \leq 1$ for any $y \in [0,1]^V$ and $i \in [k]$. Consider the corresponding soft-min functions H^t for collection $\{F_i^t\}_{i\in[k]}$ with $\alpha = n^2T^2$ for all $t \in [T]$. Denote $\ell = O(\ln \frac{1}{\epsilon})$ and fix $\tau \in \{\delta, 2\delta, \ldots, \ell\}$ with $\delta = n^{-6}T^{-3}$. According to the update in Algorithm 2, $\{y_\tau^t\}_{t\in[T]}$ and $\{y_{\tau-\delta}^t\}_{t\in[T]}$ satisfy conditions of Lemma 5. Thus, we obtain

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

Then, since the update is $y_{\tau,e}^t = y_{\tau-\delta,e}^t + \delta(1 - y_{\tau-\delta,e}^t)z_{\tau,e}^t$, we get

$$\begin{aligned}\sum_{t\in[T]} H^t(y_\tau^t) - H^t(y_{\tau-\delta}^t) &\geq \delta \sum_{t\in[T]} \sum_{e\in V} \nabla_e H^t(y_{\tau-\delta}^t)(1 - y_{\tau-\delta,e}^t) z_{\tau,e}^t - O(Tn^3\delta^2\alpha) \\ &= \delta \sum_{t\in[T]} \Delta H^t(y_{\tau-\delta}^t) \cdot z_\tau^t - O(Tn^3\delta^2\alpha). \end{aligned} \tag{13}$$

Observe that an FPL algorithm is implemented for each τ, so we can state a regret bound for each τ by using Theorem 6 with $s_t = \Delta H^t(y_{\tau-\delta}^t)$. 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 $R_\eta = \eta LAT + \frac{D}{\eta}$ is the regret guarantee for a given $\eta > 0$. By taking expectation in (13) and using the regret bound we just mentioned, we obtain

$$\begin{aligned}\mathbb{E}\left[\sum_{t\in[T]} H^t(y_\tau^t) - H^t(y_{\tau-\delta}^t)\right] &\geq \delta \left(\max_{z\in\mathcal{P}(\mathcal{M})} \mathbb{E}\left[\sum_{t\in[T]} \Delta H^t(y_{\tau-\delta}^t) \cdot z\right]\right) - \delta R_\eta - O(Tn^3\delta^2\alpha) \\ &\geq \delta\mathbb{E}\left(\sum_{t\in[T]} \left[H^t(x^*) - \sum_{i\in[k]} p_i^t(y_{\tau-\delta}^t) F_i^t(y_{\tau-\delta}^t)\right]\right) - \delta R_\eta - O(Tn^3\delta^2\alpha), \end{aligned} \tag{14}$$

where x^* is the true optimum for $\max_{x\in\mathcal{P}(\mathcal{M})} \sum_{t\in[T]} \min_{i\in[k]} F_i^t(x)$. Observe that (14) follows from

monotonicity and submodularity of each f_i^t, specifically we know that

$$\begin{aligned}
\Delta H^t(y)\cdot z &= \sum_{i\in[k]} p_i^t(y)\Delta F_i^t(y)\cdot z \\
&\geq \sum_{i\in[k]} p_i^t(y)F_i^t(x^*) - \sum_{i\in[k]} p_i^t(y)F_i^t(y) \qquad \text{(eq. (7))} \\
&\geq F_{min}^t(x^*) - \sum_{i\in[k]} p_i^t(y)F_i^t(y) \\
&\geq H^t(x^*) - \sum_{i\in[k]} p_i^t(y)F_i^t(y).
\end{aligned}$$

By applying property (12) of the soft-min in expression (14) we get

$$\begin{aligned}
\mathbb{E}\left[\sum_{t\in[T]} H^t(y_\tau^t) - H^t(y_{\tau-\delta}^t)\right] \;\geq\; & \delta\mathbb{E}\left(\sum_{t\in[T]} H^t(x^*) - H^t(y_{\tau-\delta}^t)\right) - \delta R_\eta - O(Tn^3\delta^2\alpha) \\
& -\delta T\left(\frac{n+\ln T}{\alpha} - \frac{\ln k}{\alpha} - \frac{ke^{-n}}{T}\right), \qquad (15)
\end{aligned}$$

Given the choice of α and δ, the last two terms in the right-hand side of inequality (15) are small compared to R_η, so by re-arranging terms we can state the following

$$\sum_{t\in[T]} H^t(x^*) - \mathbb{E}\left[\sum_{t\in[T]} H^t(y_\tau^t)\right] \leq (1-\delta)\left(\sum_{t\in[T]} H^t(x^*) - \mathbb{E}\left[\sum_{t\in[T]} H^t(y_{\tau-\delta}^t)\right]\right) + 2\delta R_\eta$$

By iterating $\frac{\ell}{\delta}$ times in τ, we get

$$\begin{aligned}
\sum_{t\in[T]} H^t(x^*) - \mathbb{E}\left[\sum_{t\in[T]} H^t(y_\ell^t)\right] &\leq (1-\delta)^{\frac{\ell}{\delta}}\left(\sum_{t\in[T]} H^t(x^*) - \sum_{t\in[T]} H^t(y_0^t)\right) + O\left(R_\eta \ln\frac{1}{\epsilon}\right) \\
&\leq \epsilon\left[\sum_{t\in[T]} H^t(x^*) + \frac{\ln k}{n^2 T}\right] + O\left(R_\eta \ln\frac{1}{\epsilon}\right),
\end{aligned}$$

where in the last inequality we used $(1-\delta)\leq e^{-\delta}$. Given that the term $\epsilon\frac{\ln(k)}{n^2T}$ is small (for T and n sufficiently large) we can bound it by $O(R_\eta \ln\frac{1}{\epsilon})$. Since α is sufficiently large, we can apply (9) to obtain the following regret bound

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

Since we are doing randomized swap rounding on each y_ℓ^t, Theorem 5 shows that there is a random set S^t that is independent in $\mathcal{M}_\ell$ (i.e., S^t is the union of at most ℓ independent sets in $\mathcal{I}$) such that $\mathbb{E}\left[f_i^t(S^t)\right] \geq F_i^t\left(y_\ell^t\right)$ for all $t\in[T]$ and $i\in[k]$. Thus, we finally obtain

$$(1-\epsilon)\cdot\max_{S\in\mathcal{I}}\sum_{t\in[T]}\min_{i\in[k]} f_i^t(S) - \sum_{t\in[T]}\min_{i\in[k]}\mathbb{E}\left[f_i^t(S^t)\right] \leq O\left(R_\eta \ln\frac{1}{\epsilon}\right).$$

□

Observation 1. *Theorem 2 could be easily extended to an adaptive adversary by sampling in each stage $t \in [T]$ a different perturbation $q_t \sim [0, 1/\eta]^V$ as shown in Kalai and Vempala (2005).*

4 Extensions

4.1 Knapsack constraint

Consider a knapsack constraint $\mathcal{K} = \{S \subseteq [n] : \sum_{e \in S} c_e \leq 1\}$, where $c_e > 0$ for all $e \in [n]$. Our interest is to solve the following robust problem

$$\max_{S \in \mathcal{K}} \min_{i \in [k]} f_i(S) \tag{16}$$

Corollary 1. *For Problem* (16)*, there is a polynomial time algorithm that returns a set S^{ALG}, such that for all $i \in [k]$, for a given $0 < \epsilon < 1$,*

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

and $\sum_{e \in S^{\mathrm{ALG}}} c_e \leq \ell$ for $\ell = O(\ln \frac{k}{\epsilon})$. Moreover, S^{ALG} can be covered by at most ℓ sets in $\mathcal{K}$.

Instead of using the standard greedy for every $\tau = \{1, \ldots, \ell\}$, we design an extended version of the "bang-per-buck" greedy algorithm. We formalize this procedure in Algorithm 3 below. Even though the standard "bang-per-buck" greedy algorithm does not provide any approximation factor, if we relax the knapsack constraint to be $\sum_{e \in S} c_e \leq 2$, then the algorithm gives a $1 - 1/e$ factor. There are other approaches to avoid this relaxation, see e.g. (Sviridenko, 2004).

Algorithm 3 Extended "Bang-per-Buck" Algorithm for Knapsack Constraints

Input: $\ell \geq 1$, monotone submodular function $g : 2^V \to \mathbb{R}_+$, knapsack constraint $\mathcal{K}$.
Output: sets $S_1, \ldots, S_\ell \in \mathcal{K}$.

for $\tau = 1, \ldots, \ell$ **do**
  $S_\tau \leftarrow \emptyset$
  while $V \neq \emptyset$ **do**
    Compute $e^* = \operatorname{argmax}_{e \in V} \frac{g(\cup_{j=1}^{\tau} S_j + e) - g(\cup_{j=1}^{\tau} S_j)}{c_e}$.
    if $\sum_{e \in S^\tau} c_e + c_{e^*} \leq 2$ **then**
      $S_\tau \leftarrow S_\tau + e^*$.
    $V \leftarrow V - e^*$
  Restart ground set V.

Given a monotone submodular function $g : 2^V \to \mathbb{R}_+$, Algorithm 3 produces a set $S^{\mathrm{ALG}} = S_1 \cup \cdots \cup S_\ell$ such that $g(S^{\mathrm{ALG}}) \geq \left(1 - \frac{1}{e^\ell}\right) \cdot \max_{S \in \mathcal{K}} g(S)$. Therefore, Corollary 1 can be easily proved by defining g in the same way as in Theorem 1, and running Algorithm 3 on g with $\ell = O(\ln \frac{k}{\epsilon})$.

4.2 Multiple matroid constraints

Consider a family of r matroids $\mathcal{M}_j = (V, \mathcal{I}_j)$ for $j \in [r]$. Our interest is to solve the following robust problem

$$\max_{S \in \bigcap_{j=1}^{r} \mathcal{I}_j} \min_{i \in [k]} f_i(S) \tag{17}$$

Corollary 2. *For Problem* (17)*, there is a polynomial time algorithm that returns a set S^{ALG}, such that for all $i \in [k]$, for a given $0 < \epsilon < 1$,*

$$f_i(S^{\mathrm{ALG}}) \geq (1-\epsilon) \cdot \max_{S \in \bigcap_{j=1}^{r} \mathcal{I}_j} \min_{i \in [k]} f_i(S),$$

where S^{ALG} is the union of $O(\log \frac{k}{\epsilon} / \log \frac{r+1}{r})$ independent sets in $\mathcal{I}$.

(Fisher et al., 1978) proved that the standard greedy algorithm gives a $1/(1+r)$ approximation for problem (17) when $k = 1$. Therefore, we can adapt Algorithm 1 to produce a set $S^{\mathrm{ALG}} = S_1 \cup \cdots \cup S_\ell$ such that

$$f(S^{\mathrm{ALG}}) \geq \left(1 - \left(\frac{r}{r+1}\right)^{\ell}\right) \cdot \max_{S \in \bigcap_{j=1}^{r} \mathcal{I}_j} f(S).$$

Then, Corollary 2 can be proved similarly to Theorem 1 by choosing $\ell = O(\log \frac{k}{\epsilon} / \log \frac{r+1}{r})$

4.3 Distributionally robust over polyhedral sets

Let $\mathcal{Q} \subseteq \Delta(k)$ be a polyhedral set, where $\Delta(k)$ is the probability simplex on k elements. For $q \in \mathcal{Q}$, denote $f_q \coloneqq q_1 f_1 + \cdots + q_k f_k$, which is also monotone and submodular. Given a matroid $\mathcal{M} = (V, \mathcal{I})$, our interest is to solve the following distributionally robust problem

$$\max_{S \in \mathcal{I}} \min_{q \in \mathcal{Q}} f_q(S) \tag{18}$$

Denote by $\mathrm{Vert}(\mathcal{Q})$ the set of extreme points of $\mathcal{Q}$, which is finite since $\mathcal{Q}$ is polyhedral. Then, problem (18) is equivalent to $\max_{S \in \mathcal{I}} \min_{q \in \mathrm{Vert}(\mathcal{Q})} f_q(S)$. Then, we can easily derive Corollary 3 (below) by applying Theorem 1 in the equivalent problem. Note that when $\mathcal{Q}$ is the simplex we get the original Theorem 1.

Corollary 3. *For Problem* (18)*, there is a polynomial time algorithm that returns a set S^{ALG}, such that for all $i \in [k]$, for a given $0 < \epsilon < 1$,*

$$f_i(S^{\mathrm{ALG}}) \geq (1-\epsilon) \cdot \max_{S \in \mathcal{I}} \min_{q \in \mathcal{Q}} f_q(S),$$

with $S^{\mathrm{ALG}} = S_1 \cup \cdots \cup S_\ell$ for $\ell = O(\log \frac{|\mathrm{Vert}(\mathcal{Q})|}{\epsilon})$ and $S_1, \ldots, S_\ell \in \mathcal{I}$.

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. We also thank Shabbir Ahmed for the discussions about the distributionally robust problem (18).

References

Niv Buchbinder and Moran Feldman. Deterministic algorithms for submodular maximization problems. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 392–403, 2016.

Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. Society for Industrial and Applied Mathematics, 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 *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 575–584. IEEE, 2010.

Robert Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. arXiv, 2017.

Wei Chen, Tian Lin, Zihan Tan, Mingfei Zhao, and Xuren Zhou. Robust influence maximization. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 795–804. ACM, 2016.

Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 45–54. ACM, 2008.

A. Ene and H. L. Nguyen. Constrained submodular maximization: Beyond 1/e. In *2016 IEEE 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 *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 570–579. IEEE, 2011.

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

D. Golovin, A. Krause, and M. 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 Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, December 2015. ISSN 2160-6455.

Xinran He and David Kempe. Robust influence maximization. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 885–894. ACM, 2016.

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

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 *21st Conference on Uncertainty in Artificial Intelligence (UAI)*, page 5, 2005.

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

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

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, June 2008c.

Andreas Krause, Ram Rajagopal, Anupam Gupta, and Carlos Guestrin. Simultaneous placement and scheduling of sensors. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, 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 forty-first annual ACM symposium on Theory of computing*, pages 323–332. ACM, 2009.

Hui Lin and Jeff A. Bilmes. How to select a good training-data subset for transcription: submodular active selection for sequences. In *INTERSPEECH*, 2009.

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 - Volume 9682*, IPCO 2016, pages 312–324. Springer-Verlag New York, Inc., 2016.

T. Powers, J. Bilmes, D. W. Krout, and L. Atlas. Constrained robust submodular sensor selection with applications to multistatic sonar arrays. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 2179–2185, July 2016a.

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

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

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

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, 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 fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 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.

A Proof of properties of the Soft-Min function

Proof of Lemma 4. We will just prove properties 1 and 4, since the rest is an straightforward calculation.

1. First, for all $i \in [k]$ we have $e^{-\alpha g_i(x)} \leq e^{-\alpha g_{min}(x)}$. Thus,

$$H(x) = -\frac{1}{\alpha} \ln \sum_{i \in [k]} e^{-\alpha g_i(x)} \geq -\frac{1}{\alpha} \ln \left(k e^{-\alpha g_{min}(x)} \right) = g_{min}(x) - \frac{\ln k}{\alpha}$$

   On the other hand, $\sum_{i \in [k]} e^{-\alpha g_i(x)} \geq e^{-\alpha g_{min}(x)}$. Hence,

$$H(x) \leq -\frac{1}{\alpha} \ln \left(e^{-\alpha g_{min}(x)} \right) = g_{min}(x).$$

4. Let us consider sets $A_1 = \{i \in [k] : \ g_i(x) \leq g_{min}(x) + (n + \ln T)/\alpha\}$ and $A_2 = \{i \in [k] : \ g_i(x) > g_{min}(x) + (n + \ln T)/\alpha\}$. Our intuitive argument is the following: when α is sufficiently large, those $p_i(x)$'s with $i \in A_2$ are exponentially small, and $p_i(x)$'s with $i \in A_1$ go to a uniform distribution over elements in A_1. First, observe that for each $i \in A_2$ we have

$$p_i(x) = \frac{e^{-\alpha g_i(x)}}{\sum_{i \in [k]} e^{-\alpha g_i(x)}} < \frac{e^{-\alpha [g_{min}(x) + (n + \ln T)/\alpha]}}{e^{-\alpha g_{min}(x)}} = \frac{e^{-n}}{T},$$

   so $\sum_{i \in A_2} p_i(x) g_i(x) \leq \frac{k e^{-n}}{T}$. On the other hand, for any $i \in A_1$ we have

$$\sum_{i \in A_1} p_i(x) g_i(x) \leq \left(g_{min}(x) + \frac{n + \ln T}{\alpha} \right) \sum_{i \in A_1} p_i(x) \leq H(x) + \frac{n + \ln T}{\alpha} + \frac{\ln k}{\alpha}$$

   where in the last inequality we used (9). Therefore,

$$\sum_{i \in [k]} p_i(x) g_i(x) \leq H(x) + \frac{n + \ln T}{\alpha} + \frac{\ln k}{\alpha} + \frac{k e^{-n}}{T}.$$

   Finally, the other inequality is clear since $\sum_{i \in [k]} p_i(x) g_i(x) \geq g_{min}(x) \geq H(x)$.

□

Proof of Lemma 5. For every $t \in [T]$ define a matroid $\mathcal{M}_t = (V \times \{t\}, \mathcal{I} \times \{t\}) = (V_t, \mathcal{I}_t)$. Given this, the union matroid is given by a ground set $V^{[T]} = \bigcup_{t=1}^{T} V_t$, and independent set family $\mathcal{I}^{[T]} = \{S \subseteq V^{1:T} : \ S \cap V_t \in \mathcal{I}_t\}$. Define $\mathbb{H}(X) := \sum_{t \in [T]} H^t(x^t)$ for any matrix $X \in \mathcal{P}(\mathcal{M})^T$, where x^t denotes the t-th column of X. Clearly, $\nabla_{(e,t)} \mathbb{H}(X) = \nabla_e H^t(x^t)$. Moreover, the Hessian corresponds to

$$\nabla^2_{(e_1,t),(e_2,s)} \mathbb{H}(X) = \begin{cases} 0 & \text{if } t \neq s \\ \nabla^2_{e_1,e_2} \mathbb{H}^t(x^t) & \text{if } t = s \end{cases}$$

Consider any $X, Y \in \mathcal{P}(\mathcal{M})^T$ with $|y_e^t - x_e^t| \leq \delta$. Therefore, a Taylor's expansion of $\mathbb{H}$ gives

$$\mathbb{H}(Y) = \mathbb{H}(X) + \nabla\mathbb{H}(X) \cdot (Y - X) + \frac{1}{2}(Y - X)^\top \nabla^2 \mathbb{H}(\xi) \cdot (Y - X)$$

where ξ is on the line between X and Y. If we expand the previous expression we obtain

$$\mathbb{H}(Y) - \mathbb{H}(X) = \sum_{e \in V} \sum_{t \in [T]} \nabla_e H^t(x^t)(y_e^t - x_e^t) + \frac{1}{2} \sum_{e_1, e_2 \in V} \sum_{t \in [T]} (y_{e_1}^t - x_{e_1}^t) \nabla^2_{e_1, e_2} H^t(\xi)(y_{e_2}^t - x_{e_2}^t)$$

Finally, by using property 3 in Lemma 4 and by bounding the Hessian we get

$$\mathbb{H}(Y) - \mathbb{H}(X) \geq \sum_{e \in V} \sum_{t=1}^{T} \nabla_e H^t(x^t)(y_e^t - x_e^t) - O(Tn^3\delta^2\alpha),$$

which is equivalent to

$$\sum_{t \in [T]} H^t(y^t) - \sum_{t \in [T]} H^t(x^t) \geq \sum_{e \in V} \sum_{t \in [T]} \nabla_e H^t(x^t)(y_e^t - x_e^t) - O(Tn^3\delta^2\alpha).$$

□

B Follow-the-Perturbed-Leader algorithm

In this section, we briefly recall the well-known Follow-the-Perturbed-Leader (FPL) algorithm introduced in (Kalai and Vempala, 2005) and used in many online optimization problems (see e.g., (Rakhlin and Tewari, 2009)). The classical online learning framework is as follows: Consider a dynamic process over T time steps. In each stage $t \in [T]$, a decision-maker has to choose a point $d_t \in \mathcal{D}$ from a fixed (possibly infinite) set of actions $\mathcal{D} \subseteq \mathbb{R}^n$, then an adversary chooses a vector s_t from a set $\mathcal{S}$. Finally, the player observes vector s_t and receives reward $s_t \cdot d_t$, and the process continues. The goal of the player is to maximize the total reward $\sum_{t \in [T]} s_t \cdot d_t$, and we compare her performance with respect to the best single action picked in hindsight, i.e., $\max_{d \in \mathcal{D}} \sum_{t=1}^{T} s_t \cdot d$. This performance with respect to the best single action in hindsight is called (expected) *regret*, formally:

$$\textbf{Regret}(T) = \max_{d \in \mathcal{D}} \sum_{t \in [T]} s_t \cdot d - \mathbb{E}\left[\sum_{t \in [T]} s_t \cdot d_t\right].$$

(Kalai and Vempala, 2005) showed that even if one has only access to a linear programming oracle for $\mathcal{D}$, i.e., we can efficiently solve $\max_{d \in \mathcal{D}} s \cdot d$ for any $s \in \mathcal{S}$, then the FPL algorithm 4 achieves sub-linear regret, specifically $O(\sqrt{T})$.

In order to state the main result in (Kalai and Vempala, 2005), we need the following. We assume that the decision set $\mathcal{D}$ has diameter at most D, i.e., for all $d, d' \in \mathcal{D}$, $\|d - d'\|_1 \leq D$. Further, for all $d \in \mathcal{D}$ and $s \in \mathcal{S}$ we assume that the absolute reward is bounded by L, i.e., $|d \cdot s| \leq L$ and that the ℓ_1-norm of the reward vectors is bounded by A, i.e., for all $s \in \mathcal{S}$, $\|s\|_1 \leq A$.

Theorem 7 ((Kalai and Vempala, 2005))**.** *Let $s_1, \ldots, s_T \in \mathcal{S}$ be a sequence of rewards. Running the FPL algorithm 4 with parameter $\eta \leq 1$ ensures regret*

$$\textbf{Regret}(T) \leq \eta LAT + \frac{D}{\eta}.$$

Moreover, if we choose $\eta = \sqrt{D/LAT}$, then $\textbf{Regret}(T) \leq 2\sqrt{DLAT} = O(\sqrt{T})$.

Algorithm 4 Follow-the-Perturbed-Leader (FPL), (Kalai and Vempala, 2005)

Input: Parameter $\eta > 0$

Output: Sequence of decisions $d_1, \ldots, d_T$

1: Sample $q \sim [0, 1/\eta]^n$.

2: **for** $t = 1$ to T **do**

3: **Play** $d_t = \text{argmax}_{d \in \mathcal{D}} \left(\sum_{j=1}^{t-1} s_j + q \right)^\top d$.