Optimal approximation for submodular and supermodular optimization with bounded curvature

Maxim Sviridenko[1], Jan Vondrák[2], and Justin Ward[3]

[1]Yahoo! Labs, New York, NY, USA, sviri@yahoo-inc.com
[2]Stanford University, Stanford, CA, USA, jvondrak@stanford.edu
[3]EPFL, Lausanne, Switzerland, justin.ward@epfl.ch

16 May 2017

Abstract

We design new approximation algorithms for the problems of optimizing submodular and supermodular functions subject to a single matroid constraint. Specifically, we consider the case in which we wish to maximize a monotone increasing submodular function or minimize a monotone decreasing supermodular function with a bounded total curvature c. Intuitively, the parameter $0 \leq c \leq 1$ represents how non-linear a function f is: when $c = 0$, f is linear, while for $c = 1$, f may be an arbitrary monotone increasing submodular function. For the case of submodular maximization with total curvature c, we obtain a $(1 - c/e)$-approximation — the first improvement over the greedy $(1 - e^{-c})/c$-approximation of Conforti and Cornuéjols from 1984, which holds for a cardinality constraint, as well as a recent analogous result for an arbitrary matroid constraint.

Our approach is based on modifications of the continuous greedy algorithm and non-oblivious local search, and allows us to approximately maximize the sum of a nonnegative, monotone increasing submodular function and a (possibly negative) linear function. We show how to reduce both submodular maximization and supermodular minimization to this general problem when the objective function has bounded total curvature. We prove that the approximation results we obtain are the best possible in the value oracle model, even in the case of a cardinality constraint.

We define an extension of the notion of curvature to general monotone set functions and show a $(1 - c)$-approximation for maximization and a $1/(1 - c)$-approximation for minimization cases. Finally, we give two concrete applications of our results in the settings of maximum entropy sampling, and the column-subset selection problem.

1 Introduction

The problem of maximizing a submodular function subject to various constraints is a meta-problem that appears in various settings, from combinatorial auctions [32, 14, 40] and viral marketing in social networks [25] to optimal sensor placement in machine learning [28, 29, 30, 27]. A classic result by Nemhauser, Wolsey and Fisher [35] is that the greedy algorithm provides a $(1-1/e)$-approximation for maximizing a monotone increasing submodular function subject to a cardinality constraint. The factor of $1-1/e$ cannot be improved, under the assumption that the algorithm queries the objective function a polynomial number of times [34]. While this result rules out improved approximation algorithms for arbitrary monotone increasing submodular functions, it is nonetheless possible to obtain improvements for restricted classes of submodular functions. One natural such class is based on the following notion of *curvature*, introduced by Conforti and Cornuéjols [11]:

Consider a set function $f : 2^X \to \mathbb{R}_{\geq 0}$, and for any $A \subseteq X$, $j \notin A$ let $f_A(j) = f(A \cup \{j\}) - f(A)$ be the marginal contribution of element j with respect to set A. Then, f is monotone increasing and submodular if and only if $f_A(j) \geq 0$ for all A and $j \notin A$ and $f_A(j) \geq f_B(j)$ for all $A \subseteq B$ and $j \notin B$, respectively. In this case, the marginal contribution $f_A(j)$ of element j may diminish as the set A grows, although it always remains non-negative. Intuitively, the *curvature* of a monotone increasing submodular function measures *how much* any element's marginal may decrease in the worst case. Formally, the total curvature $c \in [0,1]$ is defined as:

$$c = \max_{j \in X^*} \frac{f_\emptyset(j) - f_{X-j}(j)}{f_{\emptyset(j)}} = 1 - \min_{j \in X^*} \frac{f_{X-j}(j)}{f_\emptyset(j)}, \tag{1}$$

where $X^* = \{i \in X : f_\emptyset(i) > 0\}$. Note that when $c = 0$, all marginals of f must remain constant and so f is linear. Thus, the parameter c is one measure of how far from linear a submodular function f is. It was shown in [11] that the greedy algorithm for maximizing a monotone increasing submodular function has an approximation ratio of $(1-e^{-c})/c$ in the case of a cardinality constraint and $\frac{1}{1+c}$ for a single matroid constraint. Note that the ratios converge to 1 as $c \to 0$, and $1-1/e$ and $1/2$, respectively, as $c \to 1$, corresponding to known results for the greedy algorithm on linear and submodular functions, respectively.

Recently, various applications have motivated the study of submodular optimization under more general constraints. In particular, the $(1-1/e)$-approximation under a cardinality constraint has been generalized to any matroid constraint in [6]. This captures various applications such as welfare maximization in combinatorial auctions [40], generalized assignment problems [5] and variants of sensor placement [30]. Assuming that a monotone submodular function f has total curvature c, Vondrák [41] generalized the $(1-e^{-c})/c$-approximation of Conforti and Cornuéjols [11] to any matroid constraint, and hypothesized that this is the optimal approximation factor. Indeed, Vondrák [41] showed that this factor is optimal for any algorithm making a polynomial number of value queries to f, under a slightly generalized notion of curvature. Specifically, the lower bound requires that f have curvature c *with respect to the optimum solution.*[1] This is a generalization of the notion of total curvature,

[1] A function has a curvature c with respect to some set S if $f(S \cup T) - f(S) + \sum_{j \in S \cap T} f_{S \cup T \setminus \{j\}}(j) \geq$

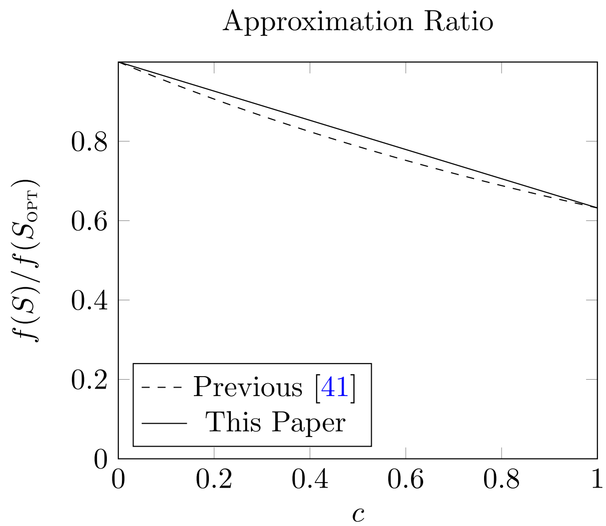

Figure 1: Comparison of Approximation Ratios for Submodular Maximization

in the sense that if f has total curvature c, it must also have total curvature at most c with respect to every set $S \subseteq X$.

1.1 Our Contribution

Our main result is that given total curvature $c \in [0, 1]$, the $\frac{1-e^{-c}}{c}$-approximation of Conforti and Cornuéjols for monotone submodular maximization subject to a cardinality constraint [11] is suboptimal and can be improved to a $(1 - c/e - O(\epsilon))$-approximation. We prove that this guarantee holds for the maximization of a monotone increasing submodular function subject to any matroid constraint, thus improving the result of [41] as well. We give two techniques that achieve this result: a modification of the continuous greedy algorithm of [6], and a variant of the local search algorithm of [19].

Using the same techniques, we obtain an approximation factor of $1 + \frac{c}{1-c}e^{-1} + \frac{1}{1-c}O(\epsilon)$ for minimizing a monotone decreasing supermodular function subject to a matroid constraint. Our approximation guarantees are strictly better than existing algorithms [22] for every value of c except $c = 0$ and $c = 1$. The relevant ratios are plotted in Figures 1 and 2. In the case of minimization, we have also plotted the inverse approximation ratio to aid in comparison. We also derive complementary negative results, showing that no algorithm that evaluates f on only a polynomial number of sets can have an approximation performance better than the algorithms we give. Thus, we resolve the question of optimal approximation as a function of total curvature in both the submodular and supermodular case. Our hardness results hold even in the special case of a uniform matroid (i.e. a cardinality constraint).

Further, we show that the assumption of bounded total curvature alone is sufficient to achieve certain approximations, even without assuming submodularity or supermodularity.

$(1 - c)f(T)$ for all sets $T \subseteq X$.

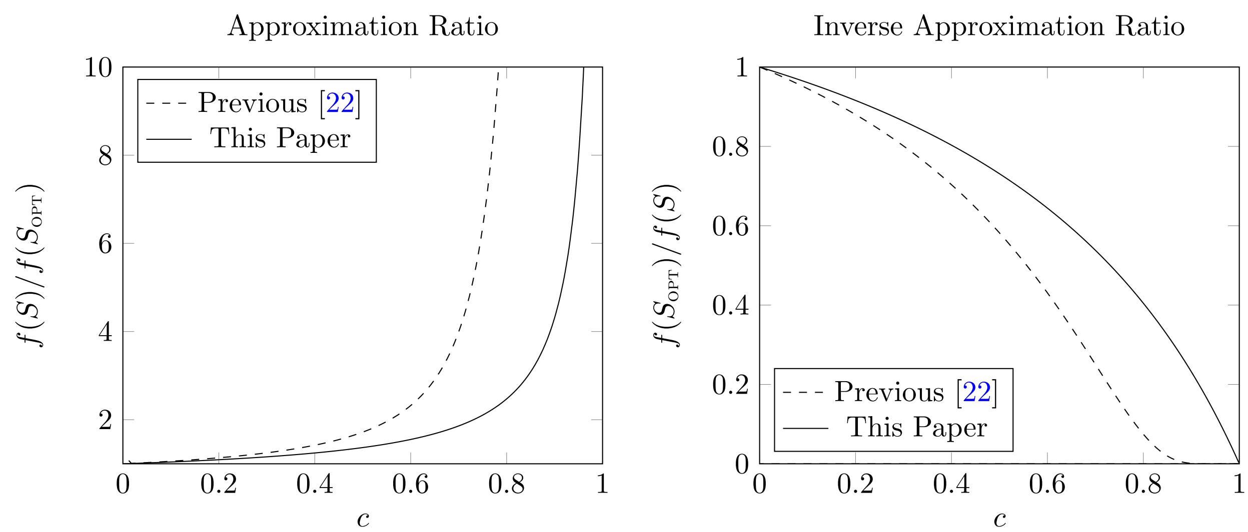

Figure 2: Comparison of Approximation Ratios for Supermodular Minimization

Specifically, there is a (simple) algorithm that achieves a $(1-c)$-approximation for the maximization of any monotone increasing function of total curvature at most c, subject to a matroid constraint. (In contrast, we achieve a $(1-c/e-O(\epsilon))$-approximation with the additional assumption of submodularity.) Also, there is a $\frac{1}{1-c}$-approximation for the minimization of any monotone decreasing function of total curvature at most c subject to a matroid constraint, compared with a $(1+\frac{c}{1-c}e^{-1}+\frac{1}{1-c}O(\epsilon))$-approximation for supermodular functions.

1.2 Applications

We also present two concrete applications of our results. Our first application is related to the Maximum Entropy Sampling Problem. Here, we are given a distribution over n random variables, with known covariance matrix M, and the goal is to select a subset of the variables that is most informative. One natural way to do this is to select some subset with maximum differential entropy. In the special case that the variables have a joint Guassian distribution, this is equivalent to finding a principle submatrix of M, corresponding to some feasible subset of variables, with maximum determinant. The Maximum Entropy Sampling Problem is NP-hard, and previous work has focused largely on obtaining exact solutions via branch and bound methods [26, 31]. Here, we consider the general problem of finding a principle submatrix $M[S,S]$ of some given matrix M with maximum determinant, subject to a matroid constraint on the set S of columns and rows that may be selected. Even for the case of a cardinality constraint k, it is impossible to approximate the maximum subdeterminant to factor better than c^k for some constant $c>1$ [7, 39]. Recently, Nikolov [36] gave an $e^{k+o(k)}$-approximation algorithm for this problem, and Nikolov and Singh later gave an $e^{k+o(k)}$-approximation algorithm for maximum subdeterminant problem even under partition matroid constraints of rank k. Here, we allow for an arbitrary matroid constraint on which columns and rows may be selected, but consider the special case in which the smallest eigenvalue λ_n of M is 1. In this restricted setting, we provide a $\big(1 - (1 - \frac{\ln \lambda_n}{\ln \lambda_1})e^{-1} - O(\epsilon)\big)$-approximation algorithm for maximizing $\ln \det(M[S,S])$ (note that our approximation results hold with respect to the natural logarithm of the determinant).

Our second application is the Column-Subset Selection Problem, which arises in various machine learning settings. Here, we are given a matrix $A \in \mathbb{R}^{m \times n}$, and the goal is to select a subset of k columns such that the matrix is well-approximated (say in squared Frobenius norm) by a matrix whose columns are in the span of the selected k columns. This is a variant of *feature selection*, since the rows might correspond to examples and the columns to features. The problem is to select a subset of k features such that the remaining features can be approximated by linear combinations of the selected features. This is related but not identical to Principal Component Analysis (PCA) where we want to select a subspace of rank k (not necessarily generated by a subset of columns) such that the matrix is well approximated by its projection to this subspace. While PCA can be solved optimally by spectral methods, the Column-Subset Selection Problem is less well understood. Here we take the point of view of approximation algorithms: given a matrix A, we want to find a subset of k columns such that the squared Frobenius distance of A from its projection on the span of these k columns is minimized. To the best of our knowledge, this problem is not known to be NP-hard; on the other hand, the approximation factors of known algorithms are quite large. The best known algorithm for the problem as stated is a $(k+1)$-approximation algorithm given by Deshpande and Rademacher [12]. For the related problem in which we may select any set of $r \geq k$ columns that contain a rank k submatrix of A, Deshpande and Vempala [13] showed that there exist matrices for which $\Omega(k/\epsilon)$ columns must be chosen to obtain a $(1+\epsilon)$-approximation. Boutsidis et al. [3] give a matching algorithm, which obtains a set of $O(k/\epsilon)$ columns that give a $(1+\epsilon)$ approximation. We refer the reader to [3] for further background on the history of this and related problems.

Here, we return to the setting in which only k columns of A may be chosen and show that this is a special case of monotone decreasing function minimization with bounded total curvature. We show a relationship between curvature and the condition number κ of A, which allows us to obtain approximation factor of κ^2. We define the problem and the related notions more precisely in Section 9.

1.3 Related Work

The problem of maximizing a monotone increasing submodular function subject to a cardinality constraint (i.e., a uniform matroid) was studied by Nemhauser, Wolsey, and Fisher [35], who showed that the standard greedy algorithm gives a $(1 - e^{-1})$-approximation. However, they later showed that the greedy algorithm has an approximation guarantee of only 1/2 for maximizing a monotone increasing submodular function subject to an arbitrary matroid constraint [20]. More recently, Calinescu et al. [6] obtained a $(1 - e^{-1})$ approximation for an arbitrary matroid constraint. In their approach, the *continuous greedy algorithm* first maximizes approximately a multilinear extension of the given submodular function and then applies a *pipage rounding* technique inspired by [1] to obtain an integral solution. The running time of this algorithm is dominated by the pipage rounding phase. Chekuri, Vondrák, and Zenklusen [8] later showed that pipage rounding can be replaced by an alternative rounding procedure called *swap rounding* based on the exchange properties of the underlying constraint. In later work [10, 9], they developed the notion of a *contention resolution scheme*, which gives a unified treatment for a variety of constraints, and allows rounding approaches for the continuous greedy algorithm to be composed in order to solve submodular maximization problems under combinations of constraints. Later, Filmus and Ward [19] obtained a $(1 - e^{-1})$-approximation for submodular maximization in an arbitrary matroid by using a non-oblivious local search algorithm that does not require rounding.

On the negative side, Nemhauser and Wolsey [34] showed that it is impossible to improve upon the bound of $(1 - e^{-1})$ in the value oracle model, even under a single cardinality constraint. In this model, f is given as a value oracle and an algorithm can evaluate f on only a polynomial number of sets. Feige [17] showed that $(1 - e^{-1})$ is the best possible approximation even when the function is given explicitly, unless $P = NP$.

In the special case of a uniform matroid, Nemhauser and Wolsey showed that the greedy algorithm is a $\frac{1-e^{-c}}{c}$-approximation algorithm whenever the curvature of f is at most c. Later, Vondrák [41] considered the continuous greedy algorithm in the setting of bounded curvature. He introduced the notion of *curvature with respect to the optimum*, which is a slightly weaker notion than total curvature, and showed that the continuous greedy algorithm is a $\frac{1-e^{-c}}{c}$-approximation for maximizing a monotone increasing submodular function f subject to an arbitrary matroid constraint whenever f has curvature at most c with respect to the optimum. He also showed that it is impossible to obtain a $\frac{1-e^{-c}}{c}$-approximation in this setting when evaluating f on only a polynomial number of sets. Unfortunately, unlike total curvature, it is in general not possible to compute the curvature of a function with respect to the optimum, as it requires knowledge of an optimal solution.

We shall also consider the problem of minimizing monotone decreasing *supermodular* functions $f : 2^X \to \mathbb{R}_{\geq 0}$. By analogy with total curvature, Il'ev [22] defines the *steepness* s of a monotone decreasing supermodular function. His definition, which is stated in terms of the marginal *decreases* of the function, is equivalent to (1) when reformulated in terms of marginal gains. He showed that, in contrast to submodular maximization, the simple greedy heuristic does not give a constant factor approximation algorithm in the general case. However, when the supermodular function f has total curvature at most c, he shows that the reverse greedy algorithm is an $\frac{e^p-1}{p}$-approximation algorithm where $p = \frac{c}{1-c}$.

2 Preliminaries

We now fix some of our notation and give two lemmas pertaining to functions with bounded total curvature. For brevity, note that we now refer to total curvature as simply *curvature*. From this point forth, we use the shorthand notation $A + i$ and $A - i$ to denote the sets $A \cup \{i\}$ and $A \setminus \{i\}$, respectively. Additionally, for any element $j \in X$, and set function $f : 2^X \to \mathbb{R}$, we write $f(j)$ as a shorthand for $f(\{j\})$.

2.1 Submodularity and Supermodularity

A set function $f : 2^X \to \mathbb{R}_{\geq 0}$ is *submodular* if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for all $A, B \subseteq X$. As noted in the introduction, submodularity can equivalently be characterized in terms of marginal values, defined by $f_A(i) = f(A + i) - f(A)$ for $i \in X$ and $A \subseteq X - i$. Then, f is submodular if and only if $f_A(i) \geq f_B(i)$ for all $A \subseteq B \subseteq X$ and $i \notin B$. Similarly, f is *supermodular* if and only if $-f$ is submodular. That is, f is supermodular if and only if $f_A(i) \leq f_B(i)$ for all $A \subseteq B \subseteq X$ and $i \notin B$.

We say that a function f is *monotone increasing*, if $f_A(i) \geq 0$ for all $i \in X$ and $A \subseteq X - i$, and *monotone decreasing* if $f_A(i) \leq 0$ for all $i \in X$ and $A \subseteq X - i$. We say that a monotone increasing function f is *normalized* if $f(\emptyset) = 0$, and similarly, that a monotone decreasing function is *normalized* if $f(X) = 0$. Note that in both cases, a normalized function is always non-negative.

Finally, suppose that f is monotone increasing and submodular and $f_\emptyset(j) = 0$ for some $j \in X$. Then we have $0 = f_\emptyset(j) \geq f_A(j) \geq 0$ for all sets $A \subseteq X$. Thus, $f_A(j) = 0$ for all $A \subseteq X$ and so j cannot contribute to any set's value. In this case, we simply remove j from X. Similarly, if f is monotone decreasing and supermodular, then $f_\emptyset(j) = 0$ implies that $0 = f_\emptyset(j) \leq f_A(j) \leq 0$ for all $A \subseteq X$, and so again we can remove j from X without affecting the optimal value of f. Henceforth, we shall thus assume that our problem's given objective function f satisfies $f_\emptyset(j) \neq 0$ for every $j \in X$. In particular, this means that we can simply set $X^* = X$ in the definition of curvature (1).

2.2 Matroids

We now present the definitions and notations that we shall require when dealing with matroids. We refer the reader to Schrijver [38] for a detailed introduction to basic matroid theory. Let $\mathcal{M} = (X, \mathcal{I})$ be a matroid defined on ground set X with independent sets given by $\mathcal{I}$. We denote by $\mathcal{B}(\mathcal{M})$ the set of all bases (inclusion-wise maximal sets in $\mathcal{I}$) of $\mathcal{M}$. We denote by $P(\mathcal{M})$ the matroid polytope for $\mathcal{M}$, given by:

$$P(\mathcal{M}) = \mathrm{conv}\{\mathbf{1}_I : I \in \mathcal{I}\} = \{x \geq 0 : \sum_{j \in S} x_j \leq r_{\mathcal{M}}(S),\ \forall S \subseteq X\},$$

where $r_{\mathcal{M}}$ denotes the rank function associated with $\mathcal{M}$. The second equality above is due to Edmonds [16]. Similarly, we denote by $B(\mathcal{M})$ the base polytope associated with $\mathcal{M}$:

$$B(\mathcal{M}) = \mathrm{conv}\{\mathbf{1}_I : I \in \mathcal{B}(\mathcal{M})\} = \{x \in P(\mathcal{M}) : \sum_{j \in X} x_j = r_{\mathcal{M}}(X)\}.$$

For a matroid $\mathcal{M} = (X, \mathcal{I})$, we denote by $\mathcal{M}^*$ the dual system $(X, \mathcal{I}^*)$ whose independent sets $\mathcal{I}^*$ are defined as those subsets $A \subseteq X$ that satisfy $A \cap B = \emptyset$ for some $B \in \mathcal{B}(\mathcal{M})$ (i.e., those subsets that are disjoint from some base of $\mathcal{M}$). Then, a standard result of matroid theory shows that $\mathcal{M}^*$ is a matroid whenever $\mathcal{M}$ is a matroid, and, moreover, $\mathcal{B}(\mathcal{M}^*)$ is precisely the set $\{X \setminus B : B \in \mathcal{B}(\mathcal{M})\}$ of complements of bases of $\mathcal{M}$.

Finally, given a set of elements $D \subseteq X$, we denote by $\mathcal{M} \,|\, D$ the matroid $(X \cap D, \mathcal{I}')$ obtained by restricting to $\mathcal{M}$ to D. The independent sets $\mathcal{I}'$ of $\mathcal{M} \,|\, D$ are simply those independent sets of $\mathcal{M}$ that contain only elements from D. That is, $I' = \{A \in \mathcal{I} : A \cap D = A\}$.

2.3 Lemmas for Functions with Bounded Curvature

We now give two general lemmas pertaining to functions of bounded curvature that will be useful in our analysis. The proofs, which follow directly from (1), are given in the Appendix.

Lemma 2.1. *If $f : 2^X \to \mathbb{R}_{\geq 0}$ is a normalized, monotone increasing submodular function with total curvature at most c, then $\sum_{j \in A} f_{X-j}(j) \geq (1-c)f(A)$ for all $A \subseteq X$.*

Lemma 2.2. *If $f : 2^X \to \mathbb{R}_{\geq 0}$ is a normalized, monotone decreasing supermodular function with total curvature at most c, then $(1-c)\sum_{j \in A} f_{\emptyset}(j) \geq -f(X \setminus A)$ for all $A \subseteq X$.*

3 Submodular + Linear Maximization

Our new results for both submodular maximization and supermodular minimization with bounded curvature make use of an algorithm for the following meta-problem: we are given a monotone increasing, normalized, submodular function $g : 2^X \to \mathbb{R}_{\geq 0}$, a linear function $\ell : 2^X \to \mathbb{R}$, and a matroid $\mathcal{M} = (X, \mathcal{I})$ and must find a base $S \in \mathcal{B}(\mathcal{M})$ maximizing $g(S) + \ell(S)$. Note that we do not require ℓ to be nonnegative. Indeed, in the case of supermodular minimization (discussed in Section 6.2), our approach shall require that ℓ be a negative, monotone decreasing function. We note that because ℓ is linear, we have $\ell(A) = \sum_{j \in A} \ell(j)$ for all $A \subseteq X$.

Let $\hat{v}_g = \max_{j \in X} g(j)$, $\hat{v}_\ell = \max_{j \in X} |\ell(j)|$, and $\hat{v} = \max(\hat{v}_g, \hat{v}_\ell)$. Then, because g is submodular and ℓ is linear, we have both $g(A) \leq n\hat{v}$ and $|\ell(A)| \leq n\hat{v}$ for every set $A \subseteq X$. Moreover, given ℓ and g, we can easily compute $\hat{v}$ in time $O(n)$. Our main technical result is the following, which gives a joint approximation for g and ℓ.

Theorem 3.1. *For every $\epsilon > 0$, there is an algorithm that, given a normalized, monotone increasing submodular function $g : 2^X \to \mathbb{R}_{\geq 0}$, a linear function $\ell : 2^X \to \mathbb{R}$ and a matroid $\mathcal{M}$, produces a set $S \in \mathcal{B}(\mathcal{M})$ in polynomial time satisfying with high probability*

$$g(S) + \ell(S) \geq \left(1 - e^{-1}\right) g(S_{\text{OPT}}) + \ell(S_{\text{OPT}}) - O(\epsilon) \cdot \hat{v},$$

for every $S_{\text{OPT}} \in \mathcal{B}(\mathcal{M})$.

In the next two sections, we give two different algorithms satisfying the conditions of Theorem 3.1.

4 A Modified Continuous Greedy Algorithm

The first algorithm we consider is a modification of the continuous greedy algorithm of [6]. Here, we describe the algorithm conceptually in the continuous setting, ignoring certain technicalities, which we shall address formally in Appendix B.

Consider $x \in [0,1]^X$. For any function $f : 2^X \to \mathbb{R}$, the *multilinear extension* of f is a function $F : [0,1]^X \to \mathbb{R}$ given by $F(x) = \mathbb{E}[f(R(x))]$, where $R(x)$ is a random subset of X in which each element e appears independently with probability x_e. Given two vectors $x, y \in [0,1]^X$, we denote by $x \vee y$ and $x \wedge y$ the vectors obtained by taking the coordinate-wise maximum and minimum, respectively, of x and y. The multilinear extension F satisfies the following properties, which follow from the submodularity of f [6, 18]:

1. $\frac{\partial F(x)}{\partial x_e} = F(x \vee \mathbf{1}_e) - F(x \wedge \mathbf{1}_{X-e}) = \frac{F(x \vee \mathbf{1}_e) - F(x)}{1 - x_e}$ for all $x \in [0,1]^X$ and $e \in X$.
2. $F(x) + F(y) \geq F(x \vee y) + F(x \wedge y)$ for all $x, y \in [0,1]^X$.

Now, we let G denote the multilinear extension of the given, monotone increasing submodular function g, and L denote the multilinear extension of the given linear function ℓ. Note that due to the linearity of expectation, $L(x) = \mathbb{E}[\ell(R(x))] = \sum_{j \in X} x_j \ell(j)$. That is, the multilinear extension L corresponds to the natural, linear extension of ℓ. Let $P(\mathcal{M})$ and $B(\mathcal{M})$ be the matroid polytope and matroid base polytope associated with $\mathcal{M}$, and let S_{OPT} be the arbitrary base in $\mathcal{B}(\mathcal{M})$ to which we shall compare our solution in Theorem 3.1. Our algorithm is shown in Figure 3. Note that in contrast to the standard continuous greedy algorithm, here we maximize ∇G over the polytope P_λ obtained from $B(\mathcal{M})$ by including the additional linear constraint $L(x) \geq \lambda$ to the matroid polytope. As we shall show, this ensures that at each time we obtain a direction that is larger than *both* the value of $\lambda = \ell(S_{\text{OPT}})$ and the residual value $g(S_{\text{OPT}}) - G(x)$. Applying the standard continuous greedy algorithm the polytope $B(\mathcal{M})$ and the function $(g + \ell)$ would give a direction that is larger than the *sum* of these two values, but this is insufficient for our purposes.

Our analysis proceeds separately for $L(x)$ and $G(x)$. First, because L is linear, and $v(x) \in P_\lambda$, we have:

$$\frac{dL}{dt} = \sum_{e \in X} v_e(t)\ell(e) = L(v(t)) \geq \lambda,$$

and therefore

$$L(x(1)) = \int_0^1 \frac{dL}{dt} dt \geq \int_0^1 \lambda dt = \lambda = \ell(S_{\text{OPT}}).$$

For the submodular component, we note that $\mathbf{1}_{S_{\text{OPT}}} \in P_\lambda$, and thus at each time step t, we must have:

$$\frac{dG}{dt} = v(t) \cdot \nabla G(x(t)) \geq \mathbf{1}_{S_{\text{OPT}}} \cdot \nabla G(x(t)) = \sum_{e \in S_{\text{OPT}}} \frac{\partial G(x(t))}{\partial x_e} = \sum_{e \in S_{\text{OPT}}} \frac{G(x(t) \vee \mathbf{1}_e) - G(x(t))}{1 - x_e(t)}$$

$$\geq \sum_{e \in S_{\text{OPT}}} G(x(t) \vee \mathbf{1}_e) - G(x(t)) \geq G(x(t) \vee \mathbf{1}_{S_{\text{OPT}}}) - G(x(t)) \geq G(\mathbf{1}_{S_{\text{OPT}}}) - G(x(t)).$$

Modified Continuous Greedy

- Guess the value of $\lambda = \ell(S_{\text{OPT}})$.
- Let $P_\lambda = B(\mathcal{M}) \cap \{x : L(x) \geq \lambda\}$.
- Initialize $x(0) = \mathbf{0}$.
- For time running from $t = 0$ to $t = 1$, update $x(t)$ according to

$$\frac{dx}{dt} = v(t),$$

where $v(t) = \arg\max_{v \in P_\lambda}(v \cdot \nabla G(x(t)))$.

- Apply randomized pipage rounding to the point $x(1)$ independently $N = \Theta(\epsilon^{-2} n^2 \log n)$ times to obtain $S_1, \ldots, S_N$.
- Return $\arg\max_{i \in [N]} f(S_i)$.

Figure 3: The modified continuous greedy algorithm

Thus, $G(x(t))$ dominates the solution of differential equation $\frac{d\phi}{dt} = g(S_{\text{OPT}}) - \phi(t)$, $\phi(0) = 0$, which is given by $(1 - e^{-t})g(S_{\text{OPT}})$. Combining the bounds on the linear and submodular components we obtain $F(x(1)) = G(x(1)) + L(x(1)) \geq (1 - e^{-1})g(S_{\text{OPT}}) + \ell(S_{\text{OPT}})$. Moreover, note that $x(1)$ is a convex combination $\int_0^1 v(t)dt$ and each $v(t)$ lies in the polytope $B(\mathcal{M})$. Thus, $x(1) \in B(\mathcal{M})$. In Appendix B we show how to implement the guessing of λ, as well as how to discretize time and efficiently find v at each step. Both of these details can be addressed while losing at most an additive term of $O(\epsilon) \cdot \hat{v}$ from the guarantees presented here.

In the last step, we run pipage rounding on $x(1)$ independently $N = \Theta(\epsilon^{-2} n^2 \log n)$ times to obtain N solutions $S_1, \ldots, S_N$ in $B(\mathcal{M})$, and return the best solution obtained. Then, as shown in [6], because $f = g + \ell$ is submodular, we have $\mathbb{E}[f(S_i)] \geq F(x(1))$ for each S_i. Consider the random variables $Y_i = \frac{f(S_i) - F(x(1))}{2n\hat{v}}$, and note that $\mathbb{E}[Y_i] \geq 0$. For any set $A \subseteq X$, $g(A) \leq n\hat{v}$ and $|\ell(A)| \leq n\hat{v}$. Hence, we have $|Y_i| \leq 1$ for all i. The algorithm returns $S = \arg\max_{i \in N} f(S_i)$. Thus, $f(S) \geq \frac{1}{N} \sum_{i \in [N]} f(S_i)$ and $\Pr[f(S) \leq F(x(1)) - 2\epsilon\hat{v}]$ is at most $\Pr[|\sum_{i \in [N]} Y_i| \geq \frac{N\epsilon}{n}]$. By a standard, symmetric variant of the Chernoff bound (see e.g. [2, Theorem A.1.16]) this probability is at most $e^{-N\epsilon^2/2n^2} = e^{-\Theta(\log n)}$. Thus, with high probability:

$$f(S) \geq F(x(1)) - O(\epsilon) \cdot \hat{v} \geq (1 - e^{-1})g(S_{\text{OPT}}) + \ell(S_{\text{OPT}}) - O(\epsilon) \cdot \hat{v}.$$

5 Non-Oblivious Local Search

We now give another proof of Theorem 3.1, using a modification of the local search algorithm of Filmus and Ward [19]. In contrast to the modified continuous greedy algorithm, our modified local search algorithm does not need to guess the optimal value of $\ell(S_{\text{OPT}})$, and also does not need to solve the associated continuous optimization problem over P_λ. However, here the convergence time of the algorithm becomes an issue that must be dealt with. We give a high-level overview of the algorithm here, ignoring the issue of convergence time. We present a full analysis considering convergence time in Appendix C

We begin by presenting a few necessary lemmas and definitions from the analysis of [19]. We shall require the following general property of matroid bases, first proved by Brualdi [4], which can also be found in, e.g. [38, Corollary 39.12a].

Lemma 5.1. *Let $\mathcal{M}$ be a matroid and A and B be two bases in $\mathcal{B}(\mathcal{M})$. Then, there exists a bijection $\pi : A \to B$ such that $A - x + \pi(x) \in \mathcal{B}(\mathcal{M})$ for all $x \in A$.*

We can restate Lemma 5.1 as follows: let $A = \{a_1, \dots, a_k\}$ and B be bases of a matroid $\mathcal{M}$ of rank k. Then we can index the elements $b_i \in B$ so that $b_i = \pi(a_i)$, and then we have that $A - a_i + b_i \in \mathcal{B}(\mathcal{M})$ for all $1 \le i \le k$. The resulting collection of sets $\{A - a_i + b_i\}_{i \in [k]}$ will define a set of feasible swaps between the bases A and B that we consider when analyzing our local search algorithm.

The local search algorithm of [19] maximizes a monotone submodular function g using a simple local search routine that evaluates the quality of the current solution using an auxiliary potential h, derived from g as follows:

$$h(A) = \sum_{B \subseteq A} g(B) \cdot \int_0^1 \frac{e^p}{e-1} \cdot p^{|B|-1}(1-p)^{|A|-|B|} dp.$$

We shall make use of the following fact, proved in [19, Lemma 4.4, p. 524-5]: for all A,

$$g(A) \le h(A) \le C \cdot g(A) \ln n$$

for some constant C.

In order to jointly maximize $g(S) + \ell(S)$, we employ a modified local search algorithm that is guided by the potential ψ, given by:

$$\psi(A) = (1 - e^{-1})h(A) + \ell(A),$$

where h is derived from g as above. Our final algorithm is shown in Figure 4. We defer a discussion of issues related to estimating h efficiently to Appendix C. Here, we present the main ideas of our modified algorithm, assuming that h can be computed exactly. As in our discussion of the continuous greedy algorithm, we can address the remaining technicalities while losing only an additive $O(\epsilon) \cdot \hat{v}$ term from our guarantees.

The following Lemma shows that if it is impossible to significantly improve $\psi(S)$ by exchanging a single element, then both $g(S)$ and $\ell(S)$ must have relatively high values.

Non-Oblivious Local Search

- Let $\delta = \frac{\epsilon}{n} \cdot \hat{v}$.
- $S \leftarrow$ an arbitrary base $S_0 \in \mathcal{B}(\mathcal{M})$.
- While there exists $a \in S$ and $b \in X \setminus S$ such that $S - a + b \in \mathcal{B}(\mathcal{M})$ and
$$\psi(S - a + b) \geq \psi(S) + \delta,$$
set $S \leftarrow S - a + b$.
- Return S.

Figure 4: The non-oblivious local search algorithm

Lemma 5.2. *Let $A = \{a_1, \dots, a_k\}$ and $B = \{b_1, \dots, b_k\}$ be any two bases of a matroid $\mathcal{M}$, and suppose that the elements of B are indexed according to Lemma 5.1 so that $A - a_i + b_i \in \mathcal{B}(\mathcal{M})$ for all $1 \leq i \leq k$. Then,*

$$g(A) + \ell(A) \geq (1 - e^{-1})g(B) + \ell(B) + \sum_{i=1}^{k} \left[\psi(A) - \psi(A - a_i + b_i)\right].$$

Proof. Filmus and Ward [19, Theorem 5.1, p. 526] show that for any submodular function g, the associated function h satisfies

$$\frac{e}{e-1} g(A) \geq g(B) + \sum_{i=1}^{k} \left[h(A) - h(A - a_i + b_i)\right]. \tag{2}$$

We note that since ℓ is linear, we have:

$$\ell(A) = \ell(B) + \sum_{i=1}^{k} \left[\ell(a_i) - \ell(b_i)\right] = \ell(B) + \sum_{i=1}^{k} \left[\ell(A) - \ell(A - a_i + b_i)\right] \tag{3}$$

Adding $(1 - e^{-1})$ times (2) to (3) then completes the proof. □

Suppose that $S \in \mathcal{B}(\mathcal{M})$ is locally optimal for ψ under single-element exchanges, and let S_{OPT} be an arbitrary base of $\mathcal{M}$. Then, local optimality of S implies that $\psi(S) - \psi(S - s_i + o_i) \geq 0$ for all $i \in [k]$, where the elements s_i of S and o_i of S_{OPT} have been indexed according to Lemma 5.1. Then, Lemma 5.2 gives $g(S) + \ell(S) \geq (1 - e^{-1})\, g(S_{\text{OPT}}) + \ell(S_{\text{OPT}})$, as required by Theorem 3.1.

6 Submodular Maximization and Supermodular Minimization

We now return to the problems of submodular maximization and supermodular minimization with bounded curvature. We reduce both problems to the general setting introduced in Section 3. In both cases, we suppose that we are seeking to optimize a function $f : 2^X \to \mathbb{R}_{\geq 0}$ over a given matroid $\mathcal{M} = (X, \mathcal{I})$ and we let S_{OPT} denote any optimal base of $\mathcal{M}$ (i.e., a base of $\mathcal{M}$ that either maximizes or minimizes f, according to the setting).

6.1 Submodular Maximization

Suppose that f is a monotone increasing submodular function with curvature at most $c \in [0, 1]$, and we seek to maximize f over a matroid $\mathcal{M}$.

Theorem 6.1. *For every $\epsilon > 0$ and $c \in [0, 1]$, there is a randomized algorithm that given a monotone increasing submodular function $f : 2^X \to \mathbb{R}_{\geq 0}$ of curvature c and a matroid $\mathcal{M} = (X, \mathcal{I})$, produces a set $S \in \mathcal{I}$ in polynomial time satisfying*

$$f(S) \geq (1 - c/e - O(\epsilon))f(S_{\text{OPT}})$$

for every $S_{\text{OPT}} \in \mathcal{I}$, with high probability.

Proof. Define the functions:

$$\ell(A) = \sum_{j \in A} f_{X-j}(j)$$
$$g(A) = f(A) - \ell(A).$$

Then, ℓ is linear and g is submodular, monotone increasing, and nonnegative (as verified in Lemma A.1 in Appendix A). Moreover, because f has curvature at most c, Lemma 2.1 implies that for any set $A \subseteq X$,

$$\ell(A) = \textstyle\sum_{j \in A} f_{X-j}(j) \geq (1 - c)f(A).$$

In order to apply Theorem 3.1 we must bound the term $\hat{v}$. By optimality of S_{OPT} and non-negativity of ℓ and g, we have $\hat{v} \leq g(S_{\text{OPT}}) + \ell(S_{\text{OPT}}) = f(S_{\text{OPT}})$. Then, from Theorem 3.1, with high probability we can find a solution S satisfying:

$$\begin{aligned} f(S) &= g(S) + \ell(S) \\ &\geq \left(1 - e^{-1}\right) g(S_{\text{OPT}}) + \ell(S_{\text{OPT}}) - O(\epsilon) \cdot f(S_{\text{OPT}}) \\ &= \left(1 - e^{-1}\right) f(S_{\text{OPT}}) + e^{-1}\ell(S_{\text{OPT}}) - O(\epsilon) \cdot f(S_{\text{OPT}}) \\ &\geq \left(1 - e^{-1}\right) f(S_{\text{OPT}}) + (1 - c)e^{-1}f(S_{\text{OPT}}) - O(\epsilon) \cdot f(S_{\text{OPT}}) \\ &= \left(1 - ce^{-1} - O(\epsilon)\right) f(S_{\text{OPT}}). \end{aligned}$$

□

6.2 Supermodular Minimization

Suppose that f is a monotone decreasing supermodular function with curvature at most $c \in [0, 1)$ and we seek to minimize f over a matroid $\mathcal{M}$.

Theorem 6.2. *For every $\epsilon > 0$ and $c \in [0, 1)$, there is a randomized algorithm that given a monotone decreasing supermodular function $f : 2^X \to \mathbb{R}_{\geq 0}$ of curvature c and a matroid $\mathcal{M} = (X, \mathcal{I})$, produces a set $S \in \mathcal{I}$ in polynomial time satisfying*

$$f(S) \leq \left(1 + \frac{c}{1-c} \cdot e^{-1} + \frac{1}{1-c} \cdot O(\epsilon)\right) f(S_{\text{OPT}})$$

for every $S_{\text{OPT}} \in \mathcal{I}$, with high probability.

Proof. Define the linear and submodular functions:

$$\ell(A) = \sum_{j \in A} f_{\emptyset}(j)$$
$$g(A) = -\ell(A) - f(X \setminus A).$$

Let us provide some intuition for the definitions of ℓ and g, beginning with the following naïve reduction. Finding an exact minimizer S of $f(S)$ is equivalent to finding a maximizer S of $f(\emptyset) - f(S)$. Because f is monotone decreasing, normalized, and supermodular, the latter objective is monotone increasing, normalized, and submodular. Unfortunately, $f(\emptyset)$ may be arbitrarily large, and so an approximate solution for the latter problem may be an arbitrarily bad solution of the original problem. In order to remove this dependency on $f(\emptyset)$, we consider instead the problem of finding some S that maximizes $-f(X \setminus S)$ in the *dual* matroid $\mathcal{M}^*$, whose bases correspond to complements of bases of $\mathcal{M}$. Thus, our definitions of ℓ and g give $-f(X \setminus S) = \ell(S) + g(S)$. Because f is monotone decreasing, we have $f_{\emptyset}(j) \leq 0$ and so $\ell(A) \leq 0$ for all $A \subseteq X$. Thus, ℓ is a non-positive, decreasing linear function. However, as we verify in Lemma A.2 of the appendix, g is submodular, monotone increasing, and nonnegative.

Now, let us turn to the problem of finding an S that maximizes $g(S) + \ell(S) = -f(X \setminus S)$ in the dual matroid $\mathcal{M}^*$. We compare our solution S to this problem to the base $S^*_{\text{OPT}} = X \setminus S_{\text{OPT}}$ of $\mathcal{M}^*$ corresponding to the optimal solution S_{OPT} of the original supermodular minimization problem. Again, in order to apply Theorem 3.1, we must bound the term $\hat{v}$. Here, because $\ell(A)$ is non-positive, we cannot bound $\hat{v}$ directly as in the previous section. Rather, we proceed by partial enumeration. Let $\hat{e} = \arg\max_{j \in S^*_{\text{OPT}}} \max(g(j), |\ell(j)|)$. We iterate through all possible guesses $e \in X$ for $\hat{e}$, and for each such e consider $\hat{v}_e = \max(g(e), |\ell(e)|)$. We set X_e to be the set $\{j \in X : g(j) \leq \hat{v}_e \wedge |\ell(j)| \leq \hat{v}_e\}$, and consider the matroid $\mathcal{M}^*_e = \mathcal{M}^* \mid X_e$, obtained by restricting $\mathcal{M}^*$ to the ground set X_e. For each e satisfying $r_{\mathcal{M}^*_e}(X_e) = r_{\mathcal{M}^*}(X)$, we apply our algorithm to the problem $\max\{g(A) + \ell(A) : A \in \mathcal{M}^*_e\}$, and return the best solution S obtained. Note since $r_{\mathcal{M}^*_e}(X_e) = r_{\mathcal{M}^*}(X)$, the set S is also a base of $\mathcal{M}^*$ and so $X \setminus S$ is a base of $\mathcal{M}$.

Consider the iteration in which we correctly guess $e = \hat{e}$. In the corresponding restricted instance we have $g(j) \leq \hat{v}_e = \hat{v}$ and $|\ell(j)| \leq \hat{v}_e = \hat{v}$ for all $j \in X_e$. Additionally, the base $S^*_{\text{OPT}} \subseteq X_e$. Thus, $r_{\mathcal{M}^*_e}(X_e) = |S^*_{\text{OPT}}| = r_{\mathcal{M}^*}(X)$ and $S^*_{\text{OPT}} \in \mathcal{B}(\mathcal{M}^*_e)$, as required by our analysis. Finally, from the definition of g and ℓ, we have $f(S_{\text{OPT}}) = f(X \setminus S^*_{\text{OPT}}) = -\ell(S^*_{\text{OPT}}) - g(S^*_{\text{OPT}})$. Since $\hat{e} \in S^*_{\text{OPT}}$, and ℓ is nonpositive while f is nonnegative,

$$\hat{v} \leq g(S^*_{\text{OPT}}) + |\ell(S^*_{\text{OPT}})| = -\ell(S^*_{\text{OPT}}) - f(S_{\text{OPT}}) - \ell(S^*_{\text{OPT}}) \leq -2\ell(S^*_{\text{OPT}}).$$

Therefore, by Theorem 3.1, the base S of $\mathcal{M}^*$ returned by the algorithm satisfies:

$$g(S) + \ell(S) \geq (1 - e^{-1})g(S^*_{\text{OPT}}) + \ell(S^*_{\text{OPT}}) + O(\epsilon) \cdot \ell(S^*_{\text{OPT}}),$$

with high probability. Finally, since f is supermodular with curvature at most c, Lemma 2.2 implies that for all $A \subseteq X$,

$$-\ell(A) = -\textstyle\sum_{j \in A} f_\emptyset(j) \leq \frac{1}{1-c} f(X \setminus A).$$

Thus, with high probability, we have

$$\begin{aligned}
f(X \setminus S) &= -g(S) - \ell(S) \\
&\leq -\left(1 - e^{-1}\right) g(S^*_{\text{OPT}}) - \ell(S^*_{\text{OPT}}) - O(\epsilon) \cdot \ell(S^*_{\text{OPT}}) \\
&= \left(1 - e^{-1}\right) f(S_{\text{OPT}}) - \left(e^{-1} + O(\epsilon)\right) \ell(S^*_{\text{OPT}}) \\
&\leq \left(1 - e^{-1}\right) f(S_{\text{OPT}}) + \left(e^{-1} + O(\epsilon)\right) \cdot \tfrac{1}{1-c} \cdot f(S_{\text{OPT}}) \\
&= \left(1 + \tfrac{c}{1-c} \cdot e^{-1} + \tfrac{1}{1-c} \cdot O(\epsilon)\right) f(S_{\text{OPT}}).
\end{aligned}$$ □

Note that because the error term depends on $\frac{1}{1-c}$, our result requires that c is bounded away from 1 by a constant.

7 Inapproximability Results

We now show that our approximation guarantees are the best achievable in the value oracle model, even in the special case that $\mathcal{M}$ is a uniform matroid (i.e., a cardinality constraint). Specifically, we show that if f is given by a value oracle then no algorithm that makes only a polynomial number of queries to f can attain a constant factor approximation better than those presented in the previous sections. Our inapproximability results are obtained by considering the problem:

$$\max\{f(S) : |S| \leq k\}, \tag{4}$$

where f is a submodular function that additionally satisfies the following property: let S_{OPT} be an optimal solution to (4), and let $p = \max_{e \in X} f_\emptyset(e)$; then, $f(S_{\text{OPT}}) = kp$. Let f be a function from this restricted class, and $\delta > 0$ be any given constant. We show that any algorithm $\mathcal{A}$ approximating $\max\{\hat{f}(S) : |S| \leq k\}$ to a factor of $(1 - ce^{-1} + \delta)$, where $\hat{f}$ is an arbitrary monotone submodular function of curvature at most $c \in (0, 1)$, can be used to approximate (4) to a factor of $(1 - e^{-1} + O(\delta))$. Moreover, if the $\mathcal{A}$ makes only a polynomial number of value queries to $\hat{f}$, then we can achieve this approximation ratio for (4) using only a polynomial number value queries to f. Although our reduction holds only under the assumption that $f(S_{\text{OPT}}) = kp$, we show in Appendix D that this property is in fact satisfied by the hard functions constructed by Nemhauser and Wolsey [34]. Specifically, they show[2] that there is a function satisfying $f(S_{\text{OPT}}) = kp$ for which no algorithm that makes only a polynomial number of value queries can obtain any constant-factor approximation ratio better than $(1 - e^{-1})$. This, combined with our reduction then shows that there is no $(1 - e^{-1} + \delta)$-approximation algorithm for maximizing a monotone increasing submodular function $\hat{f}$ of curvature at most c under a cardinality constraint that uses only a polynomial number of value queries to $\hat{f}$. We now give a full description and analysis of our reduction, as well as an analogous reduction in the case supermodular minimization.

Theorem 7.1. *For any constant $\delta > 0$ and $c \in (0, 1)$, there is no $(1-ce^{-1}+\delta)$-approximation algorithm for the problem $\max\{\hat{f}(S) : |S| \leq k\}$, where $\hat{f}$ is a monotone increasing submodular function with curvature at most c, that evaluates $\hat{f}$ on only a polynomial number of sets.*

Proof. Let $\alpha = (1 - ce^{-1} + \delta)$. Suppose for the sake of contradiction that for any monotone increasing submodular function $\hat{f}$ with curvature at most c, we could obtain a set S with $|S| \leq k$ satisfying $\hat{f}(S) \geq \alpha \hat{f}(S_{\text{OPT}})$ with constant probability for all S_{OPT} with $|S_{\text{OPT}}| \leq k$ by using only a polynomial number of value queries to $\hat{f}$. We shall show that this contradicts the negative result of Nemhauser and Wolsey [34]. Let f be a function from the family given by Nemhauser and Wolsey [34] for the cardinality constraint k, and let S_{OPT} be a set of size k on which f takes its maximum value. We define the function

$$\hat{f}(A) = f(A) + \frac{1-c}{c} \cdot |A| \cdot p,$$

where $p = \max_{i \in X} f_{\emptyset}(i)$. Note that $\hat{f}$ can be constructed by using only n initial queries to f, and each subsequent query to $\hat{f}$ can be accomplished using only a single query to f. Moreover, in Lemma A.3 in Appendix A, we show that $\hat{f}$ is monotone increasing, submodular, and nonnegative with curvature at most c. Thus, by assumption, we can obtain a set S satisfying $\hat{f}(S) \geq (1 - ce^{-1} + \delta)\hat{f}(S_{\text{OPT}})$ using only a polynomial number of value queries to $\hat{f}$, and hence to f. Because f is monotone increasing, we can assume without loss of generality that $|S| = k$. Then, from the definition of $\hat{f}$ and our assumption,

$$f(S) + \frac{1-c}{c} \cdot kp \geq \alpha \cdot f(S_{\text{OPT}}) + \alpha \cdot \frac{1-c}{c} \cdot kp = \alpha \cdot f(S_{\text{OPT}}) + \alpha \cdot \frac{1-c}{c} \cdot f(S_{\text{OPT}}) = \frac{1}{c} \cdot \alpha \cdot f(S_{\text{OPT}}),$$

[2]The analysis of Nemhauser and Wolsey [34] applies to deterministic algorithms only. In Appendix D we show how to extend it to randomized algorithms succeeding with any constant probability.

with constant probability, and so

$$\begin{aligned} f(S) &\geq \frac{1}{c} \cdot \alpha \cdot f(S_{\text{OPT}}) - \frac{1-c}{c} \cdot kp \\ &= \left(\frac{1}{c} \cdot \alpha - \frac{1-c}{c} \right) \cdot f(S_{\text{OPT}}) \\ &= \left(\frac{1}{c} - e^{-1} + \frac{\delta}{c} - \frac{1-c}{c} \right) \cdot f(S_{\text{OPT}}) \\ &= \left(1 - e^{-1} + \frac{\delta}{c} \right) \cdot f(S_{\text{OPT}}), \end{aligned}$$

with constant probability. This contradicts the information-theoretic hardness for maximizing the function f. □

Theorem 7.2. *For any constant $\delta > 0$ and $c \in (0,1)$, there is no $(1 + \frac{c}{1-c}e^{-1} - \delta)$-approximation algorithm for the problem $\min\{\hat{f}(S) : |S| \leq k\}$, where $\hat{f}$ is a monotone decreasing supermodular function with curvature at most c, that evaluates $\hat{f}$ on only a polynomial number of sets.*

Proof. Our argument proceeds similarly to the proof of Theorem 7.1. Again, let f be a function in the family given by Nemhauser and Wolsey [34] for the cardinality constraint k, and let S_{OPT} be a set of size k on which f takes its maximum value. We now construct the function

$$\hat{f}(A) = \frac{p}{c} \cdot |X \setminus A| - f(X \setminus A),$$

where again $p = \max_{i \in X} f_{\emptyset}(i)$. In Lemma A.4 in Appendix A, we show that $\hat{f}$ is monotone decreasing, supermodular, and nonnegative with curvature at most c. Again, note that the construction of $\hat{f}$ requires n initial queries to f, and each subsequent query to $\hat{f}$ can be accomplished using only a single query to f.

We consider the problem $\min\{\hat{f}(A) : |A| \leq n - k\}$. Let $\alpha = (1 + \frac{c}{1-c}e^{-1} - \delta)$, and assume that some algorithm returns a solution A to this problem, satisfying $\hat{f}(A) \leq \alpha \cdot \hat{f}(X \setminus S_{\text{OPT}})$ with constant probability, evaluating $\hat{f}$ on only a polynomial number of sets. We run this algorithm and then return the set $S = X \setminus A$. Because $\hat{f}$ is monotone decreasing, we assume without loss of generality that $|A| = n - k$ and so $|S| = k$. Then, from the definition of $\hat{f}$ and our assumption, we have (with constant probability)

$$\frac{kp}{c} - f(S) \leq \alpha \left(\frac{kp}{c} - f(S_{\text{OPT}}) \right) = \alpha \left(\frac{f(S_{\text{OPT}})}{c} - f(S_{\text{OPT}}) \right) = \alpha \cdot \frac{1-c}{c} \cdot f(S_{\text{OPT}}),$$

and so

$$\begin{aligned}
f(S) &\geq \frac{kp}{c} - \alpha \cdot \frac{1-c}{c} \cdot f(S_{\text{OPT}}) \\
&= \left(\frac{1}{c} - \alpha \cdot \frac{1-c}{c}\right) \cdot f(S_{\text{OPT}}) \\
&= \left(\frac{1}{c} - \frac{1-c}{c} - e^{-1} + \frac{1-c}{c}\delta\right) \cdot f(S_{\text{OPT}}) \\
&= \left(1 - e^{-1} + \frac{1-c}{c} \cdot \delta\right) \cdot f(S_{\text{OPT}}).
\end{aligned}$$

Again, we have obtained S using only a polynomial number of value queries to $\hat{f}$, and hence only a polynomial number of queries to f, contradicting the information-theoretic hardness result of Nemhauser and Wolsey [34]. □

8 Optimizing Monotone Nonnegative Functions with Bounded Curvature

Now we consider the problem of maximizing (respectively, minimizing) an arbitrary monotone increasing (respectively, monotone decreasing) nonnegative function f of bounded curvature subject to a single matroid constraint. We do not require f to be supermodular or submodular, but only that it have bounded curvature, in the following generalized sense.

Let f be an arbitrary monotone increasing or monotone decreasing function. We define the curvature c of f as

$$c = 1 - \min_{j \in X} \min_{S,T \subseteq X-j} \frac{f_S(j)}{f_T(j)}. \tag{5}$$

Note that in the case that f is either monotone increasing and submodular or monotone decreasing and supermodular, the minimum of $\frac{f_S(j)}{f_T(j)}$ over S and T is attained when $S = X - j$ and $T = \emptyset$. Then (5) agrees with the standard definition of curvature given in (1). Moreover, if a monotone increasing f has curvature at most c for some $c \in [0,1]$, then for any $j \in X$, and $A, B \subseteq X - j$, we have

$$(1-c) f_B(j) \leq f_A(j). \tag{6}$$

Analogously, if a monotone decreasing function f has curvature at most c, then for any $j \in X$ and $A, B \subseteq X - j$, we have

$$(1-c) f_B(j) \geq f_A(j). \tag{7}$$

Note that when $c = 0$, (6) and (7) require f to be a linear function, while when $c = 1$, they require only that f is monotone increasing or monotone decreasing, respectively.

First, we consider the case in which we wish to maximize a monotone increasing function f subject to a matroid constraint $\mathcal{M} = (X, \mathcal{I})$. Suppose that we run the standard greedy algorithm, which at each step adds to the current solution S the element e yielding the largest marginal gain in f, subject to the constraint $S + e \in \mathcal{I}$.

Theorem 8.1. *Suppose that $f : 2^X \to \mathbb{R}_{\geq 0}$ is a nonnegative, monotone increasing function with curvature at most $c \in [0, 1]$, and $\mathcal{M}$ is a matroid. Let $S \in \mathcal{B}(\mathcal{M})$ be the base produced by the standard greedy maximization algorithm on f and $\mathcal{M}$, and let $S_{\text{OPT}} \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$. Then,*

$$f(S) \geq (1 - c)f(S_{\text{OPT}}).$$

Proof. Let k be rank of $\mathcal{M}$. Let s_i be the ith element picked by the greedy algorithm, and let S_i be the set containing the first i elements picked by the greedy algorithm. We use the bijection guaranteed by Lemma 5.1 to order the elements o_i of S_{OPT} so that $S - s_i + o_i \in \mathcal{I}$ for all $i \in [k]$, and let $O_i = \{o_j : j \leq i\}$ be the set containing the first i elements of S_{OPT} in this ordering. Then,

$$\begin{aligned}
(1 - c)f(S_{\text{OPT}}) &= (1 - c)f(\emptyset) + (1 - c)\sum_{i=1}^{k} f_{O_{i-1}}(o_i) \\
&\leq f(\emptyset) + \sum_{i=1}^{k} f_{S_{i-1}}(o_i) \\
&\leq f(\emptyset) + \sum_{i=1}^{k} f_{S_{i-1}}(s_i) \\
&= f(S).
\end{aligned}$$

The first inequality follows from (6) and $f(\emptyset) \geq 0$. The last inequality is due to the fact that $S_{i-1} + o_i \in \mathcal{I}$ but s_i was chosen by the greedy maximization algorithm in the ith round. □

Similarly, we can consider the problem of finding a base of $\mathcal{M}$ that minimizes f. In this setting, we again employ a greedy algorithm, but at each step choose the element e yielding the *smallest* marginal gain in f, terminating only when no element can be added to the current solution. We call this algorithm the standard greedy minimization algorithm.

Theorem 8.2. *Suppose that $f : 2^X \to \mathbb{R}_{\geq 0}$ is a nonnegative, monotone increasing function with curvature at most $c \in [0, 1]$ and $\mathcal{M}$ is a matroid. Let $S \in \mathcal{B}(\mathcal{M})$ be the base produced by the standard greedy minimization algorithm on f and $\mathcal{M}$, and let $S_{\text{OPT}} \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$. Then,*

$$f(S) \leq \frac{1}{1 - c} f(S_{\text{OPT}}).$$

Proof. Let k, S_i, s_i, O_i, and o_i be defined as in the proof of Theorem 8.1. Then,

$$\begin{aligned} f(S_{\text{OPT}}) &= f(\emptyset) + \sum_{i=1}^{k} f_{O_{i-1}}(o_i) \\ &\geq (1-c)f(\emptyset) + (1-c)\sum_{i=1}^{k} f_{S_{i-1}}(o_i) \\ &\geq (1-c)f(\emptyset) + (1-c)\sum_{i=1}^{k} f_{S_{i-1}}(s_i) \\ &= (1-c)f(S). \end{aligned}$$

As in the proof of Theorem 8.1, the first inequality follows from (6) and $f(\emptyset) \geq 0$. The last inequality is due to the fact that $S_{i-1} + o_i \in \mathcal{I}$ but s_i was chosen by the greedy minimization algorithm in the ith round. □

Now, we consider the case in which f is a monotone decreasing function. For any function $f : 2^X \to \mathbb{R}_{\geq 0}$, we define the function $f^* : 2^X \to \mathbb{R}_{\geq 0}$ by $f^*(S) = f(X \setminus S)$ for all $S \subseteq X$. Then, since f is monotone decreasing, f^* is monotone increasing. Moreover, the next lemma shows that the curvature of f^* is the same as that of f.

Lemma 8.3. *Let $f : 2^X \to \mathbb{R}_{\geq 0}$ be a nonnegative, monotone decreasing function with curvature at most $c \in [0,1]$, and define $f^*(S) = f(X \setminus S)$ for all $S \subseteq X$. Then, f^* is nonnegative and increasing, and has curvature at most c.*

Proof. The nonnegativity and monotonicity of f^* follow immediately from that of f. Let us consider the curvature of f^*. From the definition of f^*, we have:

$$f^*_A(j) = f(X \setminus (A + j)) - f(X \setminus A) = -f_{X \setminus (A+j)}(j),$$

for any $A \subseteq X$ and $j \in X$. Consider any $j \in X$ and $S, T \subseteq X - j$. Since f is monotone decreasing with curvature at most c, (7) implies

$$f^*_S(j) = -f_{X \setminus (S+j)}(j) \geq -(1-c)f_{X \setminus (T+j)}(j) = (1-c)f^*_T(j).$$

Thus, $\frac{f^*_S(j)}{f^*_T(j)} \geq (1-c)$ for all $j \in X$ and $S, T \subseteq X - j$. □

Given a matroid $\mathcal{M}$, we consider the problem of finding a base of $\mathcal{M}$ minimizing f. This problem is equivalent to finding a base of the dual matroid $\mathcal{M}^*$ that minimizes f^*. Similarly, the problem of finding a base of $\mathcal{M}$ that maximizes f can be reduced to that of finding a base of $\mathcal{M}^*$ that maximizes f^*. Since f^* is monotone increasing with curvature no more than that of f, we obtain the following corollaries of Theorems 8.1 and 8.2, show how to employ the standard greedy algorithm to optimize monotone decreasing functions.

Corollary 8.4. *Suppose that f is a monotone decreasing function with curvature at most $c \in [0,1]$ and $\mathcal{M}$ is a matroid. Let $S^* \in \mathcal{B}(\mathcal{M}^*)$ be the base of $\mathcal{M}^*$ produced by running the standard greedy maximization algorithm on f^* and $\mathcal{M}^*$. Let $S_{\text{OPT}} \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$, $S^*_{\text{OPT}} = X \setminus S_{\text{OPT}}$, and $S = X \setminus S^* \in \mathcal{B}(\mathcal{M})$. Then,*

$$f(S) = f^*(S^*) \geq (1-c)f^*(S^*_{\text{OPT}}) = (1-c)f(S_{\text{OPT}}).$$

Corollary 8.5. *Suppose that f is a monotone decreasing function with curvature at most $c \in [0,1]$ and $\mathcal{M}$ is a matroid. Let $S^* \in \mathcal{B}(\mathcal{M}^*)$ be the base of $\mathcal{M}^*$ produced by running the standard greedy minimization algorithm on f^* and $\mathcal{M}^*$. Let $S_{\text{OPT}} \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$, $S^*_{\text{OPT}} = X \setminus S_{\text{OPT}}$, and $S = X \setminus S^* \in \mathcal{B}(\mathcal{M})$. Then,*

$$f(S) = f^*(S^*) \leq \frac{1}{1-c} f^*(S^*_{\text{OPT}}) = \frac{1}{1-c} f(S_{\text{OPT}}).$$

The approximation factors of $1-c$ and $1/(1-c)$ respectively are best possible, given curvature c. The hardness result for minimization follows from [23], where it is shown that no algorithm using polynomially many value queries can achieve an approximation factor of $\rho(n,\epsilon) = \frac{n^{1/2-\epsilon}}{1+(n^{1/2-\epsilon}-1)(1-c)}$ for the problem $\min\{f(S) : |S| \geq k\}$, where f is monotone increasing (even submodular) of curvature c. This implies that for any constant $\delta > 0$, there is no $\big(1/(1-c)+\delta\big)$-approximation algorithm using polynomially many value queries for this problem. Next, we prove the hardness result for maximization; this proof is based on known hardness constructions for maximization of XOS functions [15, 33]. Similar techniques have also been used to derive hardness results for minimization [21].

Theorem 8.6. *For any constant $c \in (0,1)$ and $\delta > 0$, there is no $(1-c+\delta)$-approximation using polynomially many queries for the problem $\max\{f(S) : |S| \leq k\}$ where f is monotone increasing of curvature c.*

Proof. Fix $c \in (0,1)$, $|X| = n$ and let $S_{\text{OPT}} \subseteq X$ be a random subset of size $k = n^{1/2}$ (assume k is an integer). We define the following functions:

$$f(S) = (1-c)|S| + c \cdot \max\{n^{1/3}, |S| \cdot n^{-1/6}, |S \cap S_{\text{OPT}}|\}$$
$$g(S) = (1-c)|S| + c \cdot \max\{n^{1/3}, |S| \cdot n^{-1/6}\}$$

The marginal values of f and g are always between $1-c$ and 1; therefore, each has curvature c. We now argue that with high probability $f(Q) = g(Q)$ for any given query Q, and so no deterministic algorithm can distinguish between f and g.

Formally, consider any fixed query Q. If $|Q| \leq n^{1/3}$, we have $f(Q) = (1-c)|Q| + cn^{1/3} = g(Q)$. We now show that if $|Q| > n^{1/3}$, then with high probability $|Q \cap S_{\text{OPT}}| \leq |Q| \cdot n^{-1/6}$ and so again $f(Q) = g(Q)$. Indeed since S_{OPT} is a random $n^{1/2}$-fraction of the ground set and $|Q| > n^{1/3}$, we have $\mu := \mathbb{E}[|Q \cap S_{\text{OPT}}|] = |Q|/n^{1/2} > n^{-1/6}$. Because Q is a random set of size *exactly* $n^{1/2}$, the events $\{e \in S_{\text{OPT}}\}_{e \in Q}$ are not independent. However, these events are negatively correlated and so we can still apply standard concentration results given by the Chernoff bound (see e.g. [37, Section 3.2]). Specifically, we have

$$\Pr[|Q \cap S_{\text{OPT}}| > |Q|n^{-1/6}] = \Pr[|Q \cap S_{\text{OPT}}| > n^{1/3}\mu] \leq e^{-\Omega\left(n^{1/3}\mu\right)} \leq e^{-\Omega\left(n^{1/6}\right)}.$$

Now, consider any deterministic algorithm and suppose that it makes a sequence of polynomially many queries $Q_1, Q_2, \ldots$ when applied to g. We also suppose, without loss of generality, that it returns some queried set Q_i with $|Q_i| \leq k$. For all sufficiently large n, with high probability we have $g(Q_i) = f(Q_i)$ for all i, by the above argument and a union bound. Thus, the algorithm will make the same sequence of queries, when applied to f. Moreover, for any queried set Q_i with $|Q_i| \leq k$ we have $f(Q_i) = g(Q_i) \leq (1-c)n^{1/2} + cn^{1/3}$. On the other hand, $f(S_{\text{OPT}}) = |S_{\text{OPT}}| = n^{1/2}$. Therefore with high probability over the choice of S_{OPT} the algorithm does not achieve better than a $(1 - c + o(1))$-approximation. For randomized algorithms, applying Yao's minimax principle shows that no randomized algorithm can achieve a better than $(1 - c + o(1))$-approximation with constant probability. □

Therefore, the approximation factors in Theorems 8.1 and 8.2 are optimal. Combining these inapproximability results with Lemma 8.3 we obtain similar inapproximability results showing the optimality of Corollary 8.4 and 8.5.

9 Applications

We now present two application of our algorithms.

9.1 Maximizing Subdeterminants and Maximum Entropy Sampling

In this application, we are given a positive semidefinite matrix $M \in \mathbb{R}^{n \times n}$. Let $M[S, S]$ be a principal minor defined by the columns and rows indexed by the set $S \subseteq \{1, \ldots, n\}$. In the Maximum Entropy Sampling Problem (or, more precisely, in a generalization of that problem) we would like to find a set $|S| = k$ maximizing $f(S) = \ln \det M[S, S]$. It is well-known that this set function $f(S)$ is submodular.[3]

We consider the special case in which M has eigenvalues $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_n \geq 1$. We show that in this case, the curvature of f is at most $\frac{\ln \lambda_n}{\ln \lambda_1}$. Consider some index j, and let μ_i denotes the ith largest eigenvalue of the submatrix $M[X - j, X - j]$. By the Cauchy Interlacing Theorem, $\mu_i \leq \lambda_i$ for all $1 \leq i \leq n-1$. Then, since the determinant of any matrix is just the product of its eigenvalues, we have:

$$f_{X-j}(j) = \ln \prod_{i=1}^{n} \lambda_i - \ln \prod_{i=1}^{n-1} \mu_i \geq \ln \prod_{i=1}^{n} \lambda_i - \ln \prod_{i=1}^{n-1} \lambda_i = \ln \lambda_n.$$

This, together with submodularity of f, implies that f is non-decreasing. Since $f(\emptyset) = \ln 1 = 0$, f is also normalized, and non-negative. Now, let μ_1 be the single eigenvalue of $M[\{j\}, \{j\}]$. Then again by the Cauchy Interlacing Theorem $\mu_1 \leq \lambda_1$, and we have

$$f_{\emptyset}(j) = \ln \mu_1 - \ln 1 = \ln \mu_1 \leq \ln \lambda_1.$$

[3]See, for example, [24]; many earlier and alternative proofs of that fact are known, as well.

Thus, f has curvature at most $\frac{\ln \lambda_1 - \ln \lambda_n}{\ln \lambda_1} = 1 - \frac{\ln \lambda_n}{\ln \lambda_1}$. It follows from Theorem 6.1 that we obtain a $\left(1 - \left(1 - \frac{\ln \lambda_n}{\ln \lambda_1}\right)\frac{1}{e} - O(\epsilon)\right)$-approximation for the problem $\max_{S \in \mathcal{I}} \ln \det M[S, S]$, even in the case that $\mathcal{I}$ is a general matroid constraint.

9.2 The Column-Subset Selection Problem

Let A be an $m \times n$ real matrix. We denote the columns of A by $\mathbf{c}_1, \ldots, \mathbf{c}_n$. I.e., for $\mathbf{x} \in \mathbb{R}^n$, $A\mathbf{x} = \sum x_i \mathbf{c}_i$. The (squared) Frobenius norm of A is defined as

$$\|A\|_F^2 = \sum_{i,j} a_{ij}^2 = \sum_{i=1}^{n} \|\mathbf{c}_i\|^2,$$

where here, and throughout this section, we use $\|\cdot\|$ to denote the standard, ℓ_2 vector norm.

For a matrix A with independent columns, the *condition number* is defined as

$$\kappa(A) = \frac{\sup_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|}{\inf_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|}.$$

If the columns of A are dependent, then $\kappa(A) = \infty$ (there is a nonzero vector $\mathbf{x}$ such that $A\mathbf{x} = 0$).

Given a matrix A with columns $\mathbf{c}_1, \ldots, \mathbf{c}_n$, and a subset $S \subseteq [n]$, we denote by

$$\mathsf{proj}_S(\mathbf{x}) = \operatorname{argmin}_{\mathbf{y} \in \mathsf{span}(\{\mathbf{c}_i : i \in S\})} \|\mathbf{x} - \mathbf{y}\|$$

the projection of $\mathbf{x}$ onto the subspace spanned by the respective columns of A. Given $S \subseteq [n]$, it is easy to see that the matrix $A(S)$ with columns spanned by $\{\mathbf{c}_i : i \in S\}$ that is closest to A in squared Frobenius norm is $A(S) = (\mathsf{proj}_S(\mathbf{c}_1), \mathsf{proj}_S(\mathbf{c}_2), \ldots, \mathsf{proj}_S(\mathbf{c}_n))$. The distance between the two matrices is thus

$$\|A - A(S)\|_F^2 = \sum_{i=1}^{n} \|\mathbf{c}_i - \mathsf{proj}_S(\mathbf{c}_i)\|^2.$$

We define $f^A : 2^{[n]} \to \mathbb{R}_{\geq 0}$ to be this quantity as a function of S:

$$f^A(S) = \sum_{i=1}^{n} \|\mathbf{c}_i - \mathsf{proj}_S(\mathbf{c}_i)\|^2 = \sum_{i=1}^{n} (\|\mathbf{c}_i\|^2 - \|\mathsf{proj}_S(\mathbf{c}_i)\|^2),$$

where the final equality follows from the fact that $\mathsf{proj}_S(\mathbf{c}_i)$ and $\mathbf{c}_i - \mathsf{proj}_S(\mathbf{c}_i)$ are orthogonal.

Given a matrix $A \in \mathbb{R}^{m \times n}$ and an integer k, the Column-Subset Selection Problem (CSSP) is to select a subset S of k columns of A so as to minimize $f^A(S)$. It follows from standard properties of projection that f^A is non-increasing, and so CSSP is a special case of non-increasing minimization subject to a cardinality constraint. We now show that the curvature of f^A is related to the condition number of A.

Lemma 9.1. *For any non-singular matrix A, the curvature $c = c(f^A)$ of the associated set function f_A satisfies*

$$\frac{1}{1-c} \leq \kappa^2(A).$$

Proof. We want to prove that for any S and $i \notin S$,

$$\min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2 \leq |f_S^A(i)| \leq \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2. \tag{8}$$

This implies that by varying the set S, a marginal value can change by at most a factor of $\kappa^2(A)$. Recall that the marginal values of f^A are negative, but only the ratio matters so we can consider the respective absolute values. The inequalities (8) imply that

$$\frac{1}{1-c} = \max_{j \in X} \max_{S,T \subseteq X-j} \frac{|f_S^A(j)|}{|f_T^A(j)|} \leq \kappa^2(A).$$

We now prove the inequalities (8). Let $\mathbf{v}_{i,S} = \mathbf{c}_i - \mathsf{proj}_S(\mathbf{c}_i)$ denote the component of $\mathbf{c}_i$ orthogonal to $\mathsf{span}(S)$. We have

$$\begin{aligned}
|f_S^A(i)| &= f^A(S) - f^A(S+i) \\
&= \sum_{j=1}^{n} \left(\|\mathbf{c}_j - \mathsf{proj}_S(\mathbf{c}_j)\|^2 - \|\mathbf{c}_j - \mathsf{proj}_{S+i}(\mathbf{c}_j)\|^2\right) \\
&= \sum_{j=1}^{n} \left(\|\mathbf{c}_j - \mathsf{proj}_S(\mathbf{c}_j)\|^2 - \|\mathbf{c}_j - \mathsf{proj}_S(\mathbf{c}_j) - \mathsf{proj}_{\mathbf{v}_{i,S}}(\mathbf{c}_j)\|^2\right) \\
&= \sum_{j=1}^{n} \|\mathsf{proj}_{\mathbf{v}_{i,S}}(\mathbf{c}_j)\|^2
\end{aligned}$$

because $\mathsf{proj}_S(\mathbf{c}_j)$, $\mathsf{proj}_{\mathbf{v}_{i,S}}(\mathbf{c}_j)$ and $\mathbf{c}_j - \mathsf{proj}_S(\mathbf{c}_j) - \mathsf{proj}_{\mathbf{v}_{i,S}}(\mathbf{c}_j)$ are orthogonal.

Our first goal is to show that if $|f_S^A(i)|$ is large, then there is a unit vector $\mathbf{x}$ such that $\|A\mathbf{x}\|$ is large. In particular, let us define $p_j = (\mathbf{v}_{i,S} \cdot \mathbf{c}_j)/\|\mathbf{v}_{i,S}\|$ and $x_j = p_j/\sqrt{\sum_{\ell=1}^{n} p_\ell^2}$. We have $\|\mathbf{x}\|^2 = \sum_{j=1}^{n} x_j^2 = 1$. Multiplying by matrix A, we obtain $A\mathbf{x} = \sum_{j=1}^{n} x_j \mathbf{c}_j$. We can estimate $\|A\mathbf{x}\|$ as follows:

$$\begin{aligned}
\mathbf{v}_{i,S} \cdot (A\mathbf{x}) &= \mathbf{v}_{i,S} \cdot \sum_{j=1}^{n} x_j \mathbf{c}_j \\
&= \sum_{j=1}^{n} \frac{p_j}{\sqrt{\sum_{\ell=1}^{n} p_\ell^2}} (\mathbf{v}_{i,S} \cdot \mathbf{c}_j) \\
&= \sum_{j=1}^{n} \frac{p_j^2}{\sqrt{\sum_{\ell=1}^{n} p_\ell^2}} \|\mathbf{v}_{i,S}\| \\
&= \|\mathbf{v}_{i,S}\| \sqrt{\sum_{j=1}^{n} p_j^2}.
\end{aligned}$$

By the Cauchy-Schwartz inequality, this implies that $\|A\mathbf{x}\| \geq \sqrt{\sum_{j=1}^n p_j^2} = \sqrt{|f_S^A(i)|}$.

On the other hand, using the expression above, we have

$$|f_S^A(i)| = \sum_{j=1}^n \|\mathsf{proj}_{\mathbf{v}_{i,S}}(\mathbf{c}_j)\|^2 \geq \|\mathsf{proj}_{\mathbf{v}_{i,S}}(\mathbf{c}_i)\|^2 = \|\mathbf{v}_{i,S}\|^2$$

since $\mathbf{v}_{i,S} = \mathbf{c}_i - \mathsf{proj}_S(\mathbf{c}_i)$ is the component of $\mathbf{c}_i$ orthogonal to $\mathsf{span}(S)$. We claim that if $\|\mathbf{v}_{i,S}\|$ is small, then there is a unit vector $\mathbf{x}'$ such that $\|A\mathbf{x}'\|$ is small. To this purpose, write $\mathsf{proj}_S(\mathbf{c}_i)$ as a linear combination of the vectors $\{\mathbf{c}_j : j \in S\}$: $\mathsf{proj}_S(\mathbf{c}_i) = \sum_{j \in S} y_j \mathbf{c}_j$. Finally, we define $y_i = -1$, and normalize to obtain $\mathbf{x}' = \mathbf{y}/\|\mathbf{y}\|$. We get the following:

$$\begin{aligned} \|A\mathbf{x}'\| = \frac{1}{\|\mathbf{y}\|}\|A\mathbf{y}\| &= \frac{1}{\|\mathbf{y}\|}\|\sum_{j=1}^n y_j \mathbf{c}_j\| \\ &= \frac{1}{\|\mathbf{y}\|}\|\mathsf{proj}_S(\mathbf{c}_i) - \mathbf{c}_i\| \\ &= \frac{1}{\|\mathbf{y}\|}\|\mathbf{v}_{i,S}\|. \end{aligned}$$

Since $\|\mathbf{y}\| \geq 1$, and $\|\mathbf{v}_{i,S}\| \leq \sqrt{|f_S^A(i)|}$, we obtain $\|A\mathbf{x}'\| \leq \sqrt{|f_S^A(i)|}$.

In summary, we have given two unit vectors $\mathbf{x}, \mathbf{x}'$ with $\|A\mathbf{x}\| \geq \sqrt{|f_S^A(i)|}$ and $\|A\mathbf{x}'\| \leq \sqrt{|f_S^A(i)|}$. This proves that $\min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2 \leq |f_S^A(i)| \leq \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2$, as required. ☐

By Corollary 8.5, the standard greedy minimization algorithm is then a $\kappa^2(A)$-approximation for the column-subset selection problem. The following lemma shows that Lemma 9.1 is asymptotically tight.

Lemma 9.2. *There exists a matrix A with condition number κ for which the associated function f^A has curvature $1 - O(1/\kappa^2)$.*

Let us denote by $\mathsf{dist}_S(\mathbf{x})$ the distance from $\mathbf{x}$ to the subspace spanned by the columns corresponding to S.

$$\mathsf{dist}_S(\mathbf{x}) = \|\mathbf{x} - \mathsf{proj}_S(\mathbf{x})\| = \min_{\mathbf{y} \in \mathsf{span}(\{\mathbf{c}_i : i \in S\})} \|\mathbf{x} - \mathbf{y}\|.$$

For some $\epsilon > 0$, consider $A = (\mathbf{c}_1, \ldots, \mathbf{c}_n)$ where $\mathbf{c}_1 = \mathbf{e}_1$ and $\mathbf{c}_j = \epsilon \mathbf{e}_1 + \mathbf{e}_j$ for $j \geq 2$. A similar example was used in [3] for a lower bound on column-subset approximation. Here, $\mathbf{e}_i$ is the i-th canonical basis vector in R^n. We claim that the condition number of A is $\kappa = O(\max\{1, \epsilon^2(n-1)\})$, while the curvature of f^A is $1 - O(\frac{1}{\max\{1, \epsilon^4(n-1)^2\}}) = 1 - O(\frac{1}{\kappa^2})$.

To bound the condition number, consider a unit vector $\mathbf{x}$. We have

$$A\mathbf{x} = \left(\left(x_1 + \epsilon \sum_{i=2}^n x_i\right), x_2, x_3, \ldots, x_n\right)$$

and

$$\|A\mathbf{x}\|^2 = (x_1 + \epsilon \sum_{i=2}^{n} x_i)^2 + \sum_{i=2}^{n} x_i^2$$

We need a lower bound and an upper bound on $\|A\mathbf{x}\|$, assuming that $\|\mathbf{x}\| = 1$. On the one hand, by the above identity and the Cauchy-Schwartz inequality, we have

$$\|A\mathbf{x}\|^2 \leq 1 + (x_1 + \epsilon \sum_{i=2}^{n} x_i)^2 \leq 1 + (1 + \epsilon^2(n-1)) = O(\max\{1, \epsilon^2(n-1)\}).$$

On the other hand, to get a lower bound: if $x_1 \leq 1/2$, then $\|A\mathbf{x}\|^2 \geq \sum_{i=2}^{n} x_i^2 = 1 - x_1^2 \geq \frac{3}{4}$. If $x_1 > 1/2$, then either $\sum_{i=2}^{n} |x_i| \leq \frac{1}{4\epsilon}$, in which case

$$\|A\mathbf{x}\|^2 \geq (x_1 + \epsilon \sum_{i=2}^{n} x_i)^2 \geq \frac{1}{16},$$

or $\sum_{i=2}^{n} |x_i| > \frac{1}{4\epsilon}$ in which case by convexity we get

$$\sum_{i=2}^{n} x_i^2 \geq \frac{1}{16\epsilon^2(n-1)}.$$

So, in all cases $\|A\mathbf{x}\|^2 = \Omega(1/\max\{1, \epsilon^2(n-1)\})$. This means that the condition number of A is $\kappa = O(\max\{1, \epsilon^2(n-1)\})$.

To lower-bound the curvature of f^A, consider the first column $\mathbf{c}_1$ and let us estimate the magnitudes of $f^A_\emptyset(1)$ and $f^A_{[n]\setminus\{1\}}(1)$. We have

$$|f^A_\emptyset(1)| = \|\mathbf{c}_1\|^2 + \sum_{j=2}^{n} \|\mathsf{proj}_1(\mathbf{c}_j)\|^2 = 1 + \epsilon^2(n-1).$$

On the other side,

$$|f^A_{[n]\setminus\{1\}}(1)| = \|\mathbf{c}_1 - \mathsf{proj}_{[n]\setminus\{1\}}\mathbf{c}_1\|^2 = (\mathsf{dist}_{[n]\setminus\{1\}}(\mathbf{c}_1))^2.$$

We exhibit a linear combination of the columns $\mathbf{c}_2, \ldots, \mathbf{c}_n$ which is close to $\mathbf{c}_1$. Let $\mathbf{y} = \frac{1}{\epsilon(n-1)} \sum_{j=2}^{n} \mathbf{c}_j$. We obtain

$$\mathsf{dist}_{[n]\setminus\{1\}}(\mathbf{c}_1) \leq \|\mathbf{c}_1 - \mathbf{y}\| = \frac{1}{\epsilon(n-1)} \|(0, -1, -1, \ldots, -1)\| = \frac{1}{\epsilon\sqrt{n-1}}.$$

Alternatively, we can also pick $\mathbf{y} = 0$ which shows that $\mathsf{dist}_{[n]\setminus\{1\}}(\mathbf{c}_1) \leq \|\mathbf{c}_1\| = 1$. So we have

$$|f^A_{[n]\setminus\{1\}}(1)| = (\mathsf{dist}_{[n]\setminus\{1\}}(\mathbf{c}_1))^2 \leq \min\left\{1, \frac{1}{\epsilon^2(n-1)}\right\} = \frac{1}{\max\{1, \epsilon^2(n-1)\}}.$$

We conclude that the curvature of f^A is at least

$$1 - \frac{1}{\max\{1, \epsilon^4(n-1)^2\}} = 1 - O\left(\frac{1}{\kappa^2}\right).$$

Acknowledgments.

The authors thank Christos Boutsidis for suggesting a connection between curvature and condition number and Wenxin Li for pointing out an issue in an earlier version of our proof of Theorem 8. Justin Ward was supported by ERC Starting Grant 335288-OptApprox and EPSRC grant EP/J021814/1.

References

[1] Alexander Ageev and Maxim Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Combinatorial Optimization*, 8(3):307–328, 2004.

[2] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. John WIley & Sons, 2nd edition, 2000.

[3] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM J. Comput.*, 43(2):687–717, 2014.

[4] Richard A Brualdi. Comments on bases in dependence structures. *Bull. of the Australian Math. Soc.*, 1(02):161–167, 1969.

[5] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint (extended abstract). In *Proc. 12th IPCO*, pages 182–196, 2007.

[6] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011.

[7] Ali Çivril and Malik Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, November 2009.

[8] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proc. 51st FOCS*, pages 575–584, 2010.

[9] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proc. 22nd SODA*, pages 1080–1097, 2011.

[10] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proc. 43rd STOC*, pages 783–792, 2011.

[11] Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7(3):251–274, 1984.

[12] Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *Proc. 51st FOCS*, pages 329–338, 2010.

[13] Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *Proc. 9th APPROX*, pages 292–303. Springer, 2006.

[14] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proc. 37th STOC*, pages 610–618, 2005.

[15] Shahar Dobzinski and Michael Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 1064–1073, 2006.

[16] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[17] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45:634–652, 1998.

[18] Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Proc. 52nd FOCS*, pages 570–579, 2011.

[19] Yuval Filmus and Justin Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM J. Comput.*, 43(2):514–542, 2014.

[20] M.L. Fisher, G.L. Nemhauser, and L.A. Wolsey. An analysis of approximations for maximizing submodular set functions—II. *Mathematical Programming Studies*, 8:73–87, 1978.

[21] Gagan Goel, Chinmay Karande, Pushkar Tripathi, and Lei Wang. Approximability of combinatorial problems with multi-agent submodular cost functions. In *Proc. 50th FOCS*, pages 755–764. IEEE, 2009.

[22] Victor P. Il'ev. An approximation guarantee of the greedy descent algorithm for minimizing a supermodular set function. *Discrete Applied Mathematics*, 114(1-3):131–146, October 2001.

[23] R. Iyer, S. Jegelka, and J. Bilmes. Curvature and optimal algorithms for learning and minimizing submodular functions. In *In Neural Information Processing Society (NIPS), Lake Tahoe, CA,*, 2013.

[24] A.K. Kelmans. Multiplicative submodularity of a matrix's principal minor as a function of the set of its rows. *Discrete Mathematics*, 44(1):113–116, 1983.

[25] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th KDD*, pages 137–146, 2003.

[26] C.W. Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1996.

[27] A. Krause and C. Guestrin. Submodularity and its applications in optimized information gathering. *ACM Trans. on Intelligent Systems and Technology*, 2(4):32, 2011.

[28] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *Proc. 5th IPSN*, pages 2–10, 2006.

[29] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Machine Learning Research*, 9:235–284, 2008.

[30] Andreas Krause, Ram Rajagopal, Anupam Gupta, and Carlos Guestrin. Simultaneous placement and scheduling of sensors. In *Proc. 8th IPSN*, pages 181–192, 2009.

[31] Jon Lee. Maximum entropy sampling. *Encyclopedia of Environmetrics*, 3:1229–1234, 2002.

[32] B. Lehmann, D. J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55:1884–1899, 2006.

[33] Vahab S. Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings 9th ACM Conference on Electronic Commerce (EC-2008), Chicago, IL, USA, June 8-12, 2008*, pages 70–77, 2008.

[34] G.L. Nemhauser and L.A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

[35] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

[36] Aleksandar Nikolov. Randomized rounding for the largest simplex problem. In *Proc. 47th STOC*, pages 861–870. ACM, 2015.

[37] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997.

[38] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[39] Marco Di Summa, Friedrich Eisenbrand, Yuri Faenza, and Carsten Moldenhauer. On largest volume simplices and sub-determinants. In *Proc. 26th SODA*, SODA '15, pages 315–323. Society for Industrial and Applied Mathematics, 2015.

[40] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. 40th STOC*, pages 67–74, 2008.

[41] Jan Vondrák. Submodularity and curvature: the optimal algorithm. In *RIMS Kokyuroku Bessatsu*, volume B23, pages 253–266, Kyoto, 2010.

A Proofs and Claims Omitted from the Main Body

Lemma 2.1. *If $f : 2^X \to \mathbb{R}_{\geq 0}$ is a normalized, monotone increasing submodular function with total curvature at most c, then $\sum_{j \in A} f_{X-j}(j) \geq (1-c)f(A)$ for all $A \subseteq X$.*

Proof. We order the elements of X arbitrarily, and let A_j be the set containing all those elements of A that precede the element j. Then, $\sum_{j \in A} f_{A_j}(j) = f(A) - f(\emptyset)$. From the definition of curvature, we have

$$c \geq 1 - \frac{f_{X-j}(j)}{f_\emptyset(j)}$$

which, since $f_\emptyset(j) \geq 0$, is equivalent to

$$f_{X-j}(j) \geq (1-c)f_\emptyset(j), \text{ for each } j \in A.$$

Because f is submodular, we have $f_\emptyset(j) \geq f_{A_j}(j)$ for all j, and so

$$\sum_{j \in A} f_{X-j}(j) \geq (1-c)\sum_{j \in A} f_\emptyset(j) \geq (1-c)\sum_{j \in A} f_{A_j}(j) = (1-c)[f(A) - f(\emptyset)] = (1-c)f(A),$$

where the final equality follows from the assumption that f is normalized. □

Lemma 2.2. *If $f : 2^X \to \mathbb{R}_{\geq 0}$ is a normalized, monotone decreasing supermodular function with total curvature at most c, then $(1-c)\sum_{j \in A} f_\emptyset(j) \geq -f(X \setminus A)$ for all $A \subseteq X$.*

Proof. Order A arbitrarily, and let A_j be the set of all elements in A that precede element j, including j itself. Then, $\sum_{j \in A} f_{X \setminus A_j}(j) = f(X) - f(X \setminus A)$. From the definition of curvature, we have

$$c \geq 1 - \frac{f_{X-j}(j)}{f_\emptyset(j)},$$

which, since $f_\emptyset(j) \leq 0$, is equivalent to

$$f_{X-j}(j) \leq (1-c)f_\emptyset(j).$$

Then, since f is supermodular, we have $f_{X \setminus A_j}(j) \leq f_{X-j}(j)$ for all $j \in A$, and so

$$(1-c)\sum_{j \in A} f_\emptyset(j) \geq \sum_{j \in A} f_{X-j}(j) \geq \sum_{j \in A} f_{X \setminus A_j}(j) = f(X) - f(X \setminus A) = -f(X \setminus A),$$

where the final equality follows from the assumption that f is normalized. □

Lemma A.1. *Let $f : 2^X \to \mathbb{R}_{\geq 0}$ be a normalized, monotone increasing submodular function and define $\ell(A) = \sum_{j \in A} f_{X-j}(j)$ and $g(A) = f(A) - \ell(A)$. Then, g is normalized, monotone increasing and submodular.*

Proof. The function g is the sum of a submodular function f and a linear function $-\ell$, and so must be submodular. Moreover, since f is normalized $g(\emptyset) = f(\emptyset) - \ell(\emptyset) = 0$, so g is normalized. For any $j \in X$ and $A \subseteq X - j$,

$$g_A(j) = f_A(j) - f_{X-j}(j) \geq 0$$

since f is submodular. Thus, g is monotone increasing. $\square$

Lemma A.2. *Let $f : 2^X \to \mathbb{R}_{\geq 0}$ be a normalized, monotone-decreasing supermodular function and define $\ell(A) = \sum_{j \in A} f_\emptyset(j)$ and $g(A) = -\ell(A) - f(X \setminus A)$. Then, g is normalized, monotone increasing, and submodular.*

Proof. We first show that g is monotone-increasing. Consider an arbitrary $j \in X$ and $A \subseteq X - j$. Then,

$$\begin{aligned} g_A(j) &= g(A + j) - g(A) \\ &= -\ell(A + j) - f((X \setminus A) - j) + \ell(A) + f(X \setminus A) \\ &= -\ell(j) + f_{(X \setminus A) - j}(j) \\ &= -f_\emptyset(j) + f_{(X \setminus A) - j}(j) \\ &\geq 0, \end{aligned}$$

where the final inequality holds because f is supermodular. Moreover, since f is normalized, we have $g(\emptyset) = -f(X) = 0$.

Finally, we show that g is submodular. Suppose $A \subseteq B$ and $j \notin B$. Then, $(X \setminus B) - j \subseteq (X \setminus A) - j$ and so, since f is supermodular, $f_{(X \setminus B) - j}(j) \leq f_{(X \setminus A) - j}(j)$. Thus,

$$g_A(j) = -f_\emptyset(j) + f_{(X \setminus A) - j}(j) \geq -f_\emptyset(j) + f_{(X \setminus B) - j}(j) = g_B(j).$$

$\square$

Lemma A.3. *Let f be a normalized, monotone increasing submodular function, satisfying $f_A(j) \leq p$ for all $j \in X$ and $A \subseteq X - j$, and let $c \in [0, 1]$. Define*

$$\hat{f}(A) = f(A) + \frac{1 - c}{c} \cdot |A| \cdot p.$$

Then, $\hat{f}$ is normalized, monotone increasing, and submodular, and has curvature at most c.

Proof. Because $\hat{f}$ is the sum of a normalized, monotone increasing, submodular function and a nonnegative linear function, must be normalized, monotone increasing, and submodular. Furthermore, for any $A \subseteq X$ and $j \notin A$, we have $\hat{f}_A(j) = f_A(j) + \frac{1-c}{c} p$. Thus,

$$\frac{\hat{f}_{X-j}(j)}{\hat{f}_\emptyset(j)} = \frac{f_{X-j}(j) + \frac{1-c}{c} \cdot p}{f_\emptyset(j) + \frac{1-c}{c} \cdot p} \geq \frac{\frac{1-c}{c} \cdot p}{p + \frac{1-c}{c} \cdot p} = \frac{\frac{1-c}{c}}{\frac{1}{c}} = 1 - c,$$

and so $\hat{f}$ has curvature at most c. $\square$

Lemma A.4. *Let f be a normalized, monotone increasing submodular function, satisfying $f_A(j) \leq p$ for all j, A, and let $c \in [0,1]$. Define:*

$$\hat{f}(A) = \frac{p}{c} \cdot |X \setminus A| - f(X \setminus A).$$

Then, $\hat{f}$ is normalized, monotone decreasing, and supermodular, and has curvature at most c.

Proof. Because f is submodular, so is $f(X \setminus A)$, and hence $-f(X \setminus A)$ is supermodular. Thus, $\hat{f}$ is the sum of a supermodular function and a linear function and so is supermodular. Moreover, $\hat{f}(X) = -f(\emptyset) = 0$, and so $\hat{f}$ is normalized. In order to see that $\hat{f}$ is decreasing, we consider the marginal $\hat{f}_A(j)$, which is equal to

$$\frac{p}{c} \cdot |X \setminus (A+j)| - f(X \setminus (A+j)) - \frac{p}{c} \cdot |X \setminus A| + f(X \setminus A) = -\frac{p}{c} + f_{X \setminus (A+j)}(j) \leq -\frac{p}{c} + p \leq 0.$$

Finally, we show that $\hat{f}$ has curvature at most c. We have:

$$\hat{f}_{\emptyset}(j) = -\frac{p}{c} + f_{X-j}(j) \geq -\frac{p}{c}$$
$$\hat{f}_{X-j}(j) = -\frac{p}{c} + f_{\emptyset}(j) \leq -\frac{p}{c} + p = -\frac{1-c}{c}p,$$

and therefore $\hat{f}_{X-j}(j)/\hat{f}_{\emptyset}(j) \leq 1 - c$. □

B Implementation of the Modified Continuous Greedy Algorithm

Here we discuss the technical details of how the continuous greedy algorithm can be implemented efficiently. There are two main issues that we ignored in our previous discussion: (1) How do we "guess" the value of $\ell(S_{\text{OPT}})$; (2) How discretize time efficiently and find a suitable direction $v(t)$ in each step of the algorithm. We address each in turn.

Guessing the value $\lambda = \ell(S_{\mathbf{opt}})$: Recall that $|\ell(S_{\text{OPT}})| \leq n\hat{v}$. We discretize the interval[4] $[-n\hat{v}, n\hat{v}]$ with $O(\epsilon^{-1})$ points of the form $i\epsilon \cdot \hat{v}$ for $-\epsilon^{-1} \leq i \leq \epsilon^{-1}$, filling the interval $[-\hat{v}, \hat{v}]$, together with $O(\epsilon^{-1} n \log n)$ points of the form $(1+\epsilon/n)^i \cdot \hat{v}$ and $-(1+\epsilon/n)^i \cdot \hat{v}$ for $0 \leq i \leq \log_{1+\epsilon/n} n$, filling the intervals $[\hat{v}, n\hat{v}]$ and $[-n\hat{v}, -\hat{v}]$, respectively. We then run the following algorithm using each point as a guess for λ, and return the best solution found. Then, if $|\ell(S_{\text{OPT}})| < \hat{v}$, we must have

$$\ell(S_{\text{OPT}}) \geq \lambda \geq \ell(S_{\text{OPT}}) - \epsilon \cdot \hat{v},$$

[4]In the applications we consider ℓ is either nonnegative or non-positive, and so we need only consider half of the given interval. For simplicity, here we give a general approach that does not depend on the sign of ℓ. In general, we have favored, whenever possible, simplicity in the analysis over obtaining the best runtime bounds.

for some iteration (using one of the guesses in $[-\hat{v}, \hat{v}]$). If $|\ell(S_{\text{OPT}})| \geq \hat{v}$, consider the largest guess λ satisfying $\ell(S_{\text{OPT}}) \geq \lambda$. If $\ell(S_{\text{OPT}}) \geq 0$, we have $\lambda \geq \ell(S_{\text{OPT}})(1+\frac{\epsilon}{n})^{-1} \geq \ell(S_{\text{OPT}})(1-\frac{\epsilon}{n}) = \ell(S_{\text{OPT}}) - \frac{\epsilon}{n}|\ell(S_{\text{OPT}})|$. Similarly, if $\ell(S_{\text{OPT}}) < 0$, we have $\lambda \geq \ell(S_{\text{OPT}})(1+\frac{\epsilon}{n})\ell(S_{\text{OPT}}) = \ell(S_{\text{OPT}}) - \frac{\epsilon}{n}|\ell(S_{\text{OPT}})|$. In both cases, we have

$$\ell(S_{\text{OPT}}) \geq \lambda \geq \ell(S_{\text{OPT}}) - \frac{\epsilon}{n}|\ell(S_{\text{OPT}})| \geq \ell(S_{\text{OPT}}) - \epsilon \cdot \hat{v},$$

where the last inequality follows from the fact that $|\ell(S_{\text{OPT}})| \leq n\hat{v}$. For the remainder of our analysis we consider the iteration of the algorithm corresponding to this guess for λ.

Discretizing time efficiently and finding a suitable direction in each step: These details are addressed by using the same approach given Calinescu et al. [6]. That is, we discretize time into $1/\delta$ steps (for some appropriate small δ) in exactly the same fashion as [6]. Here, we discuss only the required modifications to their general analysis. To simplify our notation, for any $j \in A \subseteq X$, define $g_A(j) = 0$ (recall that $g_A(j)$ was previously defined only when $j \notin A$). Given our guess of λ and a current solution $x(t)$, we find an appropriate direction $v(t)$ in each time step t, and update $x(t+\delta) = x(t) + \delta v(t)$. At some time t, suppose we set $w_e = \mathbb{E}[g_{R(x)}(e)]$ and then choose $v(t) = \max_{v \in P_\lambda} \sum_{e \in X} v_e w_e$. Note that this is simply a linear maximization problem over $B(\mathcal{M})$ subject to an additional linear constraint $L(v) \geq \lambda$, and can be solved by the ellipsoid method, for example (or more efficiently using other methods). Also, for our chosen guess of λ, we have $\mathbf{1}_{S_{\text{OPT}}} \in P_\lambda$, so $\sum_{e \in X} v_e w_e \geq \sum_{e \in S_{\text{OPT}}} w_e$. Let $\text{OPT} = g(S_{\text{OPT}})$. By submodularity and monotonicity of g, we have $\text{OPT} \leq g(R) + \sum_{e \in S_{\text{OPT}}} g_R(e)$ for any set $R \subseteq X$. Taking the expectation over a random set $R = R(x)$, we then obtain:

$$\text{OPT} \leq \mathbb{E}[g(R(x)) + \sum_{e \in S_{\text{OPT}}} g_{R(x)}(e)] = G(x) + \sum_{e \in S_{\text{OPT}}} \mathbb{E}[g_{R(x)}(e)] \leq G(x) + \sum_{e \in X} v_e w_e.$$

This is precisely the guarantee given in Lemma 3.1 of [6]. Let k be the rank of $\mathcal{M}$. By carrying out the remainder of the analysis exactly as in [6] (see Lemmas 3.2, 3.3, and following remarks on pp. 12-13), we obtain a polynomial-time algorithm that produces a fractional solution $x(1)$ satisfying $G(x(1)) \geq (1 - \frac{1}{e} - \frac{\epsilon}{k})S_{\text{OPT}} \geq (1-1/e)g(S_{\text{OPT}}) - O(\epsilon)\hat{v}$. Additionally, since L is linear and each $v(t) \in P_\lambda$, we have:

$$L(x(1)) = \sum_{i=1}^{1/\delta} \delta L(v(\delta i)) \geq \lambda \geq \ell(O) - \epsilon\hat{v}.$$

Combining our bounds on L and G we obtain

$$G(x(1)) + L(x(1)) \geq (1-e^{-1})g(S_{\text{OPT}}) - \ell(S_{\text{OPT}}) - O(\epsilon) \cdot \hat{v}.$$

C Implementation of the Local Search Algorithm

Here we discuss the technical details of how the non-oblivious local search algorithm can be implemented efficiently. We must address two remaining concerns: (1) how do we compute

ψ efficiently in polynomial time; and (2) how do we ensure that the search for improvements converges to a local optimum in polynomial time? As in the case of the continuous greedy algorithm, we can address these issues by using standard techniques, but we must be careful since ℓ may take negative values. As in that case, we have not attempted to obtain the most efficient possible running time analysis here, focusing instead on simplifying the arguments.

Estimating ψ efficiently: Although the definition of h requires evaluating g on a potentially exponential number of sets, Filmus and Ward show that h can be estimated efficiently using a sampling procedure:

Lemma C.1 ([19, Lemma 5.1, p. 525])**.** *Let $\tilde{h}(A)$ be an estimate of $h(A)$ computed from $N = \Theta(\epsilon^{-2} \ln^2 n \ln M)$ samples of g. Then,*

$$\Pr[|\tilde{h}(A) - h(A)| \geq \epsilon \cdot h(A)] = O(M^{-1}).$$

We let $\tilde{\psi}(A) = \tilde{h}(A) + \ell(A)$ be an estimate of ψ. Set $\delta = \frac{\epsilon}{n} \cdot \hat{v}$. We shall ensure that $\tilde{\psi}(A)$ differs from $\psi(A)$ by at most

$$\delta = \frac{\epsilon}{n} \cdot \hat{v} \geq \frac{\epsilon}{n^2} \cdot g(A) = \frac{\epsilon}{C \cdot n^2 \ln n} \cdot C \cdot g(A) \ln n \geq \frac{\epsilon}{C \cdot n^2 \ln n} \cdot h(A).$$

Applying Lemma C.1, we can then ensure that

$$\Pr[|\tilde{\psi}(A) - \psi(A)| \geq \delta] = O(M^{-1}),$$

by using $\Theta(\epsilon^{-2} n^4 \ln^4 n \ln M)$ samples for each computation of ψ. By the union bound, we can ensure that $|\tilde{\psi}(A) - \psi(A)| \leq \delta$ holds with high probability for all sets A considered by the algorithm, by setting M appropriately. In particular, if we evaluate $\tilde{\psi}$ on any polynomial number of distinct sets A, it suffices to make M polynomially small, which requires only a polynomial number of samples for each evaluation.

Bounding the convergence time of the algorithm: We initialize our search with an arbitrary base $S_0 \in \mathcal{B}(\mathcal{M})$, and at each step of the algorithm, we restrict our search to those improvements that yield a significant increase in the value of ψ. Specifically, we require that each improvement increases the current value of ψ by at least an additive term $\delta = \frac{\epsilon}{n} \cdot \hat{v}$. We now bound the total number of improvements made by the algorithm.

We suppose that all values $\tilde{\psi}(A)$ computed by the algorithm satisfy

$$\psi(A) - \delta \leq \tilde{\psi}(A) \leq \psi(A) + \delta.$$

From the previous discussion, we can ensure that this is indeed the case with high probability.

Let $S_\psi = \arg\max_{A \in \mathcal{B}(\mathcal{M})} \psi(A)$. Then, by assumption, we must have $\tilde{\psi}(S) \leq \psi(S) + \delta \leq \psi(S_\psi) + \delta$ for every current solution S in the algorithm. The total number of improvements applied by the algorithm is at most:

$$\begin{aligned}
\tfrac{1}{\delta}(\psi(S_\psi) + \delta - \tilde{\psi}(S_0)) &\leq \tfrac{1}{\delta}(\psi(S_\psi) - \psi(S_0) + 2\delta) \\
&= \tfrac{1}{\delta}((1-e^{-1})\cdot(h(S_\psi) - h(S_0)) + \ell(S_\psi) - \ell(S_0) + 2\delta) \\
&\leq \tfrac{1}{\delta}((1-e^{-1})\cdot h(S_\psi) + |\ell(S_\psi)| + |\ell(S_0)| + 2\delta) \\
&\leq \tfrac{1}{\delta}((1-e^{-1})\cdot C\cdot g(S_\psi)\ln n + |\ell(S_\psi)| + |\ell(S_0)| + 2\delta) \\
&\leq \tfrac{1}{\delta}((1-e^{-1})\cdot C\cdot \hat{v}\cdot n\ln n + n\cdot\hat{v} + n\cdot\hat{v} + 2\delta) \\
&= O(\epsilon^{-1}n^2\ln n).
\end{aligned}$$

Each improvement step requires $O(n^2)$ evaluations of ψ. From the discussion in the previous section, setting M sufficiently high will ensures that all of the estimates made for the first $\Theta(\epsilon^{-1}n^2\ln n)$ iterations will satisfy our assumptions with high probability, and so the algorithm will converge in polynomial time with high probability.

In order to obtain a deterministic bound on the running time of the algorithm we simply terminate our search if it has not converged in $\Theta(\epsilon^{-1}n^2\ln n)$ steps and return the current solution. With high probability the resulting algorithm will converge before this, in which case we will have $\tilde{\psi}(S) - \tilde{\psi}(S - s_i + o_i) \leq \delta$ for every $i \in [k]$. Then,

$$\sum_{i=1}^{k}[\psi(S) - \psi(S - s_i + o_i)] \leq k(\delta + 2\delta) \leq 3\epsilon\cdot\hat{v}.$$

From Lemma 5.2, the set S produced by the algorithm then satisfies

$$g(S) + \ell(S) \geq (1-e^{-1})g(S_{\text{OPT}}) + \ell(S_{\text{OPT}}) - O(\epsilon)\cdot\hat{v},$$

as required by Theorem 3.1.

D Hardness Construction for Randomized Algorithms

Here we review the value-query hardness construction of Nemhauser and Wolsey [34] that is used in our inapproximability results from Section 7, and show how their result can easily be extended to randomized algorithms. Rather than repeating the full construction, we shall refer the reader to specific relevant properties and lemmas wherever possible.

Consider the problem of finding a set S (approximately) maximizing a normalized, monotone increasing, submodular function $f : 2^X \to \mathbb{R}_{\geq 0}$ subject to the constraint that $|S| \leq k$. For each $k \geq 2$, $r \leq k-1$, and $n \geq 3(k-r) + r - 2$, Nemhauser and Wolsey [34] show how to construct a submodular, monotone increasing function $v_r^k : 2^X \to \mathbb{R}_{\geq 0}$, where X is a set of n elements. For all S, the value of $v_r^k(S)$ depends only on $|S|$ and $|S \cap M|$, where M is some fixed set of "special" elements. Consider any k and r satisfying $2 \leq r < k$, and let $p = (k-r+1)^{k-r}$. Then, for all n, the associated function v_r^k satisfies the following properties, given in [34]:

- $v_r^k(\emptyset) = 0$ [34, eq. 3.7, p. 181]

- $\max\{v_r^k(S) : |S| \leq k\} = v_r(M) = kp$ [34, Property 3, p. 180]
- $v_r^k(\{e\}) - v_r^k(\emptyset) = p$ for every $e \in X$ [34, eq. 3.8, p. 181]

Thus, every v_r^k is normalized, and satisfies the additional property $v_r^k(S_{\text{OPT}}) = kp$ required in Section 7. Intuitively, the construction of [34] is designed so that in order to find a good approximate maximizer for v_r^k, we need to find a set S such that $|S \cap M|$ is large. However, they show [34, p. 180] that (by construction) the value of v_r^k does not reveal any information about M unless $r \leq |S \cap M| \leq 3(k-r) + r - 2$. Hence, if we are given only a value oracle to v_r^k, determining M requires making a large number of value queries.

Formally, they define $\alpha_k^r = 1 - \left(\frac{k-r}{k}\right)\left(\frac{k-r-1}{k-r}\right)^{k-r}$ to be some desired approximation ratio. Then, they show that, because of the above properties of v_r^k, the number of function values required to approximate $\max\{v_r^k(S) : |S| \leq k\}$ to within a factor of α_k^{r-1} is at least the number of queries required to solve the following simple, combinatorial problem [34, Lemma 4.1, p. 182]:

Find a set $S \subseteq X$ with $|X| \leq 3(k-r) + r - 2$, such that $|S \cap M| \geq r + 1$, where M is unknown, and if a set S is proposed, we are informed whether S is a solution of the problem or not. (9)

By combinatorial arguments, Nemhauser and Wolsey then show that for any polynomial number of fixed queries in (9), there exists some M so that $|Q \cap M| \leq r$ for every query Q. Here, we proceed by choosing M randomly and arguing that any deterministic algorithm making a polynomial number of queries in (9) has $|Q \cap M| \leq r$ for every query Q with high probability.

To this end, we fix the parameters $k = n^{3/7}$ and $r = n^{2/7} - 1$, and let M be a random set of k elements. We have $r + 1 = n^{2/7}$ and $3(k-r) + r - 2 < 3n^{3/7}$. Note that for this choice of parameters, we have $\lim_{n\to\infty} \alpha_k^{r-1} = 1 - 1/e$. Thus, suppose that we have chosen n sufficiently large so that $\alpha_k^{r-1} < 1 - 1/e + \delta$. Fix some constant $q > 0$. We now show that with high probability any deterministically chosen sequence of n^q queries in (9) will have $|Q \cap M| \leq r$ for *every* query Q. Thus (by [34, Lemma 4.1]) no deterministic algorithm can attain a α_k^{r-1} approximation for $\max\{v_r^k(S) : |S| \leq k\}$ with constant probability. Applying Yao's principle, we then have that no randomized algorithm can attain an α_k^{r-1}-approximation with constant probability in the worst case.

In order to prove our claim, we consider some queried set Q in problem (9). For each $e \in Q$, let $Y_e \in \{0, 1\}$ be an indicator variable for the random event $e \in Q \cap M$. Then, for all $e \in Q$ we have $\mathbb{E}[Y_e] = n^{-4/7}$ since M is a set of $n^{3/7}$ elements chosen uniformly at random from X. If $|Q| < r + 1$ or $|Q| > 3(k-r) + r - 2$, then Q is never a solution to (9). For any other proposed set Q, let $\mu = \mathbb{E}[|Q \cap M|] = \mathbb{E}[\sum_{e \in Q} Y_e]$. Then, we have $n^{-2/7} < \mu < 3n^{-1/7}$. Moreover, $|Q \cap M| \geq r + 1$ only if $|Q \cap M| \geq \mu(1 + \delta)$, where $\delta = \frac{r+1}{3n^{-1/7}} - 1 \geq \frac{n^{3/7}}{4}$. Note that because M is a uniformly random set of size *exactly* r, the variables Y_e are not independent. However, we observe that they are negatively correlated. Thus, we can still apply standard concentration results given by the Chernoff bound (see e.g. [37, Section 3.2]). Specifically, we have that $\Pr[|Q \cap M| \geq r+1] \leq \Pr[\sum_{e \in Q} Y_e \geq \mu(1+\delta)]$ is at most

$$e^{-\frac{\mu\delta}{3}} \leq e^{-\frac{n^{-2/7}n^{3/7}}{12}} = e^{-\frac{n^{1/7}}{12}} < 1/n^{-(q+1)}$$

for any constant q and all sufficiently large n.

Now, consider some deterministic algorithm for problem (9) that queries some polynomial number n^q of sets in problem (9). The sequence of sets that the algorithm queries depends only whether each set is a solution of (9) or not. With probability at least $n^{-(q+1)}$ this is the case for any given query Q. Thus, by the union bound, with probability at least $1 - 1/n$ the algorithm will receive a "no" answer for every queried set, and so will always query the same sequence of sets. In particular, it never queries a set that is a solution to (9), and so cannot attain an α_k^{r-1}-approximation.