arXiv:1311.4728v3 [cs.DS] 12 Dec 2014

Optimal approximation for submodular and supermodular optimization with bounded curvature

Maxim Sviridenko* Jan Vondrák† Justin Ward‡

December 15, 2014

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 nondecreasing submodular function or minimize a nonincreasing supermodular function in the setting of bounded total curvature c. In the case of submodular maximization with curvature c, we obtain a $(1-c/e)$-approximation — the first improvement over the greedy $(1-e^{-c})/c$-approximation of Conforti and Cornuejols from 1984, which holds for a cardinality constraint, as well as recent approaches that hold 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, nondecreasing 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 $(1-c)$-approximation for maximization and $1/(1-c)$-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 [30, 12, 35] and viral marketing in social networks [23] to optimal sensor placement in machine learning [26, 27, 28, 25]. A classic result by Nemhauser, Wolsey and Fisher [33] is that the greedy algorithm provides a $(1-1/e)$-approximation for maximizing a nondecreasing 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 [32].

The greedy algorithm has been applied in numerous settings in practice. Although it is useful to know that it never performs worse than $1-1/e$ compared to the optimum, in practice its performance is often even better than this, in fact very close to the optimum. To get a quantitative handle on this phenomenon, various assumptions can be made about the input. One such assumption is the notion of *curvature*, introduced by Conforti and Cornuéjols [9]: A function $f : 2^X \to \mathbb{R}_+$ has curvature $c \in [0,1]$, if $f(S+j) - f(S)$ does not change by a factor larger than $1-c$ when varying S. A function with $c = 0$ is linear, so the parameter measures in some sense how far f is from linear. It was shown in [9] that the greedy algorithm for nondecreasing submodular functions provides a $(1-e^{-c})/c$-approximation, which tends to 1 as $c \to 0$.

*Yahoo! Labs, New York, NY, USA. sviri@yahoo-inc.com

†IBM Almaden Research Center, San Jose, CA, USA. jvondrak@us.ibm.com

‡Department of Computer Science, University of Warwick, Coventry, United Kingdom. J.D.Ward@warwick.ac.uk Work supported by EPSRC grant EP/J021814/1.

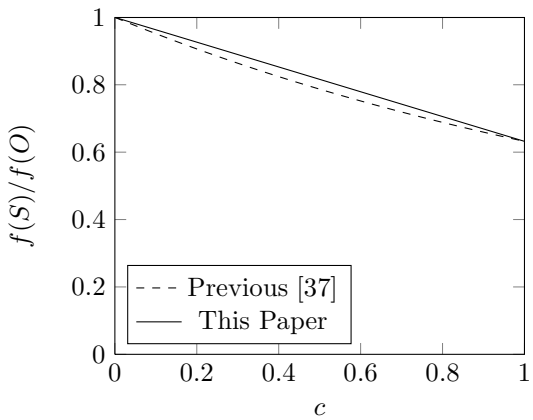

Figure 1: Comparison of Approximation Ratios for Submodular Maximization

Recently, various applications have motivated the study of submodular optimization under various more general constraints. In particular, the $(1-1/e)$-approximation under a cardinality constraint has been generalized to any matroid constraint in [5]. This captures various applications such as welfare maximization in combinatorial auctions [35], generalized assignment problems [4] and variants of sensor placement [28]. Assuming curvature c, [37] generalized the $(1-e^{-c})/c$-approximation of [9] to any matroid constraint, and hypothesized that this is the optimal approximation factor. It was proved in [37] that this factor is indeed optimal for instances of curvature c *with respect to the optimum* (a technical variation of the definition, which depends on how values change when measured on top of the optimal solution). In the following, we use *total curvature* to refer to the original definition of [9], to distinguish from curvature w.r.t. the optimum [37].

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 [9] 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 nondecreasing submodular function subject to any matroid constraint, thus improving the result of [37] as well. We give two techniques that achieve this result: a modification of the continuous greedy algorithm of [5], and a variant of the local search algorithm of [18].

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 nonincreasing supermodular function subject to a matroid constraint. Our approximation guarantees are strictly better than existing algorithms 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 approximation performance better than the algorithms we give. Thus, we resolve the question of optimal approximation as a function of curvature in both the submodular and supermodular case.

Further, we show that the assumption of bounded curvature alone is sufficient to achieve certain approximations, even without assuming submodularity or supermodularity. Specifically, there is a (simple) algorithm that achieves a $(1-c)$-approximation for the maximization of any nondecreasing 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 nonincreasing 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.

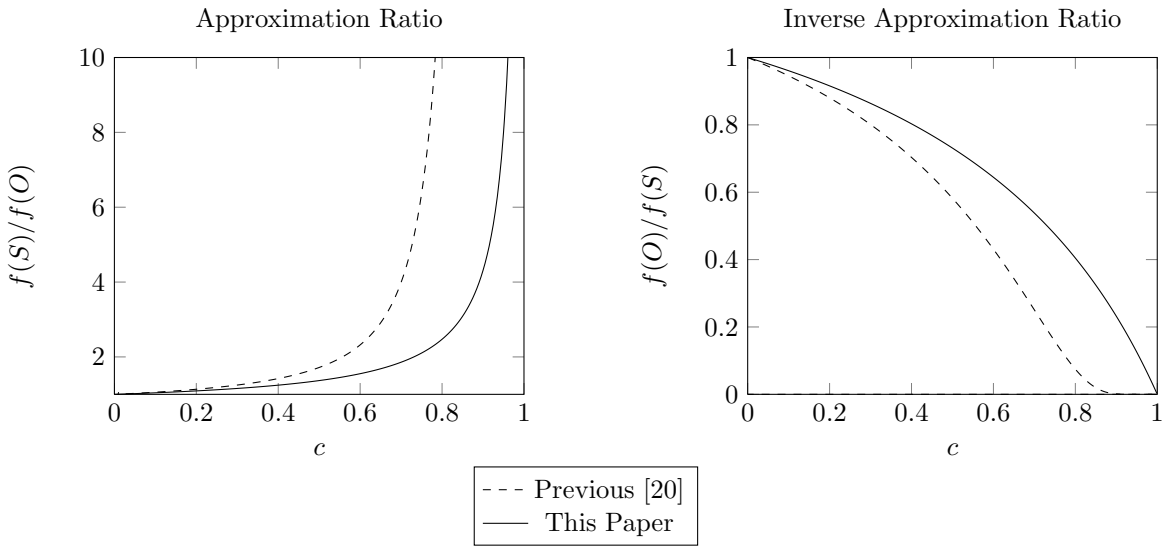

Figure 2: Comparison of Approximation Ratios for Supermodular Minimization

1.2 Applications

We provide two applications of our results. In the first application, we are given a positive semidefinite matrix M. 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 the 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 [22] (many earlier and alternative proofs of that fact are known). In addition, we know that solving this problem exactly is NP-hard [24] (see also Lee [29] for a survey on known optimization techniques for the problem).

We consider the maximum entropy sampling problem when the matrix M has eigenvalues $1 \leq \lambda_1 \leq \cdots \leq \lambda_n$. Since the determinant of any matrix is just a product of its eigenvalues, the Cauchy Interlacing Theorem implies that the the submodular function $f(S) = \ln\det M[S,S]$ is nondecreasing. In addition we can easily derive a bound on its curvature $c \leq 1 - 1/\lambda_n$ (see the formal definition of curvature in Section 1.3). This immediately implies that our new algorithms for submodular maximization have an approximation guarantee of $1 - \left(1 - \frac{1}{\lambda_n}\right)\frac{1}{e}$ for the maximum entropy sampling problem.

Our second application is the Column-Subset Selection Problem arising in various machine learning settings. The goal is, given a matrix $A \in \mathbb{R}^{m\times n}$, 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 [10]. For the related problem in which we may select any set of $r \geq k$ columns that form a rank k submatrix of A, Deshpande

and Vempala [11] showed that there exist matrices for which $\Omega(k/\epsilon)$ columns must be chosen to obtain a $(1+\epsilon)$-approximation. Boutsidis et al. [2] give a matching algorithm, which obtains a set of $O(k/\epsilon)$ columns that give a $(1+\epsilon)$ approximation. We refer the reader to [2] 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 nonincreasing function minimization with bounded 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 nondecreasing submodular function subject to a cardinality constraint (i.e., a uniform matroid) was studied by Nemhauser, Wolsey, and Fisher [33], who showed that the standard greedy algorithm gives a $(1-e^{-1})$-approximation. However, in [19], they show that the greedy algorithm is only 1/2-approximation for maximizing a nondecreasing submodular function subject to an arbitrary matroid constraint. More recently, Calinescu et al. [5] 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 [6] 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 [8, 7], 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 [17] 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 [32] 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 [16] showeds that $(1-e^{-1})$ is the best possible approximation even when the function is given explicitly, unless $P = NP$. In later work, Vondrák [36] introduced the notion of the *symmetry gap* of a submodular function f, which unifies many inapproximability results in the value oracle model, and proved new inapproximability results for some specific constrained settings. Later, Dobzinski and Vondrák [14] showed how these inapproximability bounds may be converted to matching complexity-theoretic bounds, which hold when f is given explicitly, under the assumption that $RP \neq NP$.

Conforti and Cornuéjols [9] defined the *total curvature* c of non-decreasing submodular function f as

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

They showed that greedy algorithm has an approximation ratio of $\frac{1}{1+c}$ for the problem of maximizing a nondecreasing submodular function with curvature at most c subject to a single matroid constraint. In the special case of a uniform matroid, they were able to show that the greedy is a $\frac{1-e^{-c}}{c}$-approximation algorithm. Later, Vondrák [37] 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 weaker notion than total curvature, and showed that the continuous greedy algorithm is a $\frac{1-e^{-c}}{c}$-approximation for maximizing a nondecreasing 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 nonincreasing *supermodular* functions $f : 2^X \to \mathbb{R}_{\geq 0}$. By analogy with total curvature, Il'ev [20] defines the *steepness* s of a nonincreasing 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 where $p = \frac{c}{1-c}$.

2 Preliminaries

We now fix some of our notation and give two lemmas pertaining to functions with bounded curvature.

2.1 Set Functions

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 $A, B \subseteq X$. 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 $i \notin B$. That is, submodular functions are characterized by decreasing marginal values.

Intuitively, 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 $i \notin B$.

Finally, we say that a function is *non-decreasing*, or *monotone increasing*, if $f_A(i) \geq 0$ for all $i \in X$ and $A \subseteq X$, and *non-increasing*, or *monotone decreasing*, if $f_A(i) \leq 0$ for all $i \in X$ and $A \subseteq X$.

2.2 Matroids

We now present the definitions and notations that we shall require when dealing with matroids. We refer the reader [34] 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:

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

where $r_{\mathcal{M}}$ denotes the rank function associated with $\mathcal{M}$. The second equality above is due to Edmonds [15]. Similarly, we denote by $B(\mathcal{M})$ the base polytope associated with $\mathcal{M}$:

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

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 $B \in \mathcal{I}$ (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 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 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, nonnegative, 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. For $j \in X$, we shall write $\ell(j)$ and $g(j)$ as a shorthand for $\ell(\{j\})$ and $g(\{j\})$. 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 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 $S \in \mathcal{B}(M)$ in polynomial time satisfying*

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

for every $O \in \mathcal{B}(M)$, with high probability.

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 [5] We first sketch the algorithm conceptually in the continuous setting, ignoring certain technicalities.

4.1 Overview of the Algorithm

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 $F(x) = \mathbb{E}\left[f(R_x)\right]$, where R_x is a random subset of X in which each element e appears independently with probability x_e. 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, $L(x) = \mathbb{E}\left[\ell(R_x)\right] = \sum_{j \in X} x_j \ell(j)$. 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 $\mathcal{M}$, and let O 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. In contrast to the standard continuous greedy algorithm, in the third step we require a direction that is larger than *both* the value of $\ell(O)$ and the residual value $\gamma - G(x(t))$. Applying the standard continuous greedy algorithm to $(g + \ell)$ gives 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, we obtain

$$\frac{dL}{dt}(x(t)) = L\left(\frac{dx}{dt}\right) \geq \lambda,$$

Modified Continuous Greedy

- Guess the values of $\lambda = \ell(O)$ and $\gamma = g(O)$.
- Initialize $x = \mathbf{0}$.
- For time running from $t = 0$ to $t = 1$, update x according to $\frac{dx}{dt} = v(t)$, where $v(t) \in P(\mathcal{M})$ is a vector satisfying both:

$$L(v(t)) \geq \lambda$$
$$v(t) \cdot \nabla G(x(t)) \geq \gamma - G(x(t)).$$

  Such a vector exists because $\mathbf{1}_O$ is one possible candidate (as in the analysis of [5]).
- Apply pipage rounding to the point $x(1)$ and return the resulting solution.

Figure 3: The modified continuous greedy algorithm

for every time t, and hence $L(x(1)) \geq \lambda = \ell(O)$. For the submodular component, we obtain

$$\frac{dG}{dt}(x(t)) = \frac{dx}{dt} \cdot \nabla G(x(t)) \geq \gamma - G(x(t))$$

similar to the analysis in [5]. This leads to a differential equation that gives

$$G(x(t)) \geq (1 - e^{-t})\gamma.$$

The final pipage rounding phase is oblivious to the value of the objective function and so is not affected by the potential negativity of ℓ. We can view $G + L$ as the multilinear extension of $g + \ell$ and so, as in the standard continuous greedy analysis, pipage rounding produces an integral solution S satisfying $g(S) + \ell(S) \geq G(x(1)) + L(x(1)) \geq (1 - e^{-t})g(O) + \ell(O)$.

4.2 Implementation of the Modified Continuous Greedy Algorithm

Now we discuss the technical details of how the continuous greedy algorithm can be implemented efficiently. There are three main issues that we ignored in our previous discussion: (1) How do we "guess" the values of $\ell(O)$ and $g(O)$; (2) How do we find a suitable direction $v(t)$ in each step of the algorithm; and (3) How do we discretize time efficiently? Let us now address them one by one.

Guessing the optimal values: In fact, it is enough to guess the value of $\ell(O)$; we will optimize over $v(t) \cdot \nabla G$ later. Recall that $|\ell(O)| \leq n\hat{v}$. We discretize the interval[1] $[-n\hat{v}, n\hat{v}]$ 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 $|\ell(O)| < \hat{v}$, we must have

$$\ell(O) \geq \lambda \geq \ell(O) - \epsilon \cdot \hat{v},$$

for some iteration (using one of the guesses in $[-\hat{v}, \hat{v}]$). Similarly, if $|\ell(O)| \geq \hat{v}$, then for some iteration we have

$$\ell(O) \geq \lambda \geq \ell(O) - (\epsilon/n)|\ell(O)| \geq \ell(O) - \epsilon \cdot \hat{v}$$

(using one of the guesses in $[\hat{v}, n\hat{v}]$ or $[-n\hat{v}, -\hat{v}]$). For the remainder of our analysis we consider this particular iteration.

[1] 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.

Finding a suitable direction: Given our guess of λ and a current solution $x(t)$, our goal is to find a direction $v(t) \in P(\mathcal{M})$ such that $L(v(t)) \geq \lambda$ and $v(t) \cdot \nabla G(x(t)) \geq \gamma - G(x(t))$. As in [5], we must estimate $\nabla G(x(t))$ by random sampling. Then, given an estimate $\tilde{\nabla} G$, we solve the linear program:

$$\max\{v \cdot \tilde{\nabla} G : v \in B(\mathcal{M}), L(v) \geq \lambda\}. \tag{2}$$

We can do this by the ellipsoid method, for example (or more efficiently using other methods).

Following the analysis of [5], we can obtain, in polynomial time, an estimate satisfying

$$|v(t) \cdot \tilde{\nabla} G(x(t)) - v(t) \cdot \nabla G(x(t))| \leq O(\epsilon) \cdot \hat{v},$$

with high probability. Since $L(O) = \ell(O) \geq \lambda$, the base O is a feasible solution of (2). Because $G(x(t))$ is concave along any nonnegative direction, we then have:

$$v \cdot \nabla G(x(t)) \geq g(O) - G(x(t)) - O(\epsilon) \cdot \hat{v}, \tag{3}$$

just as in the analysis of [5].

Discretizing the algorithm: We discretize time into steps of length $\delta = \epsilon/n^2$; let us assume for simplicity that $1/\delta$ is an integer. In each step, we find a direction $v(t)$ as described above and we update

$$x(t+\delta) = x(t) + \delta \cdot v(t).$$

Clearly, after $1/\delta$ steps we obtain a solution $x(1)$ which is a convex combination of points $v(t) \in P(\mathcal{M})$, and therefore a feasible solution. In each step, we had $L(v(t)) \geq \lambda$ and so

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

The analysis of the submodular component follows along the lines of [5]. In one time step, we gain

$$\begin{aligned}
&G(x(t+\delta)) - G(x(t)) \\
&\quad \geq (1 - n\delta) \cdot (x(t+\delta) - x(t)) \cdot \nabla G(x(t)) \\
&\quad \geq (1 - n\delta) \cdot \delta \cdot [g(O) - G(x(t)) - O(\epsilon) \cdot \hat{v}] \\
&\quad \geq \delta \cdot [g(O) - G(x(t)) - O(\epsilon) \cdot \hat{v} - (\epsilon/n) \cdot g(O)] \\
&\quad = \delta \cdot [g(O) - G(x(t)) - O(\epsilon) \cdot \hat{v}]
\end{aligned}$$

using the bound (3). By induction (as in [5]), we obtain

$$G(x(t)) \geq (1 - e^{-t}) g(O) - O(\epsilon) \cdot \hat{v},$$

and so

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

as required.

5 Non-Oblivious Local Search

We now give another proof of Theorem 3.1, using a modification of the local search algorithm of [18]. In contrast to the modified continuous greedy algorithm, our modified local search algorithm does not need to guess the optimal value of $\ell(O)$, and also does not need to solve the associated continuous optimization problem given in (2). However, here the convergence time of the algorithm becomes an issue that must be dealt with.

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

5.1 Overview of the Algorithm

We begin by presenting a few necessary lemmas and definitions from the analysis of [18]. We shall require the following general property of matroid bases, first proved by Brualdi [3], which can also be found in e.g. [34, 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 $A = \{a_1, \ldots, a_r\}$ and B be bases of a matroid $\mathcal{M}$ of r. Then we can index the elements of $b_i \in B$ so that $b_i = \pi(a_i)$, and then we have that $A - a_i + b_i \in \mathcal{B}(\mathcal{M})$ for all $1 \leq i \leq r$. The resulting collection of sets $\{A - a_i + b_i\}_{i \in A}$ 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 [18] 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 defer a discussion of issues related to convergence and computing h until the next subsection, and first sketch the main idea of our modified algorithm. We shall make use of the following fact, proved in [18, Lemma 4.4, p. 524-5]: for all A,

$$g(A) \leq h(A) \leq C \cdot g(A) \ln n,$$

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:

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

Our final algorithm is shown in Figure 4.

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.

Lemma 5.2. *Let $A = \{a_1, \ldots, a_r\}$ and $B = \{b_1, \ldots, b_r\}$ 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 $1 \leq i \leq r$. Then,*

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

Proof. Filmus and Ward [18, Theorem 5.1, p. 526] show that for any submodular function g, the associated function h satisfies

$$\frac{e}{e-1} g(A) \geq g(B) + \sum_{i=1}^{r} \left[h(A) - h(A - a_i + b_i) \right]. \tag{4}$$

We note that since ℓ is linear, we have:

$$\begin{aligned} \ell(A) &= \ell(B) + \sum_{i=1}^{r} \left[\ell(a_i) - \ell(b_i) \right] \\ &= \ell(B) + \sum_{i=1}^{r} \left[\ell(A) - \ell(A - a_i + b_i) \right] \end{aligned} \tag{5}$$

Adding $(1 - e^{-1})$ times (4) to (5) then completes the proof. □

Suppose that $S \in \mathcal{B}(\mathcal{M})$ is locally optimal for ψ under single-element exchanges, and let O 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 $i \in [r]$, where the elements s_i of S and o_i of O have been indexed according to Lemma 5.1. Then, Lemma 5.2 gives $g(S) + \ell(S) \geq \left(1 - e^{-1}\right) g(O) + \ell(O)$, as required by Theorem 3.1.

5.2 Implementation of the Non-Oblivious Local Search Algorithm

We now show how to obtain a polynomial-time algorithm from Lemma 5.2. We face two technical difficulties: (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 5.3 ([18, Lemma 5.1, p. 525])**.** *Let $\tilde{h}(A)$ be an estimate of $h(A)$ computed from $N = \Omega(\epsilon^{-2} \ln^2 n \ln M)$ samples of g. Then,*

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

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

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

Applying Lemma 5.3, we can then ensure that

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

by using $\Omega(\epsilon^{-2} n^4 \ln^4 n \ln M)$ 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 $O_\psi = \arg\max_{A \in \mathcal{B}(\mathcal{M})} \psi(A)$. Then, the total number of improvements applied by the algorithm is at most:

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

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 $\Omega(\epsilon^{-1} n^2 \ln n)$ iterations will satisfy our assumptions with high probability, and so the algorithm will converge in polynomial time.

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 $\Omega(\epsilon^{-1} n^2 \ln n)$ steps and return the current solution. Then when the resulting algorithm terminates, with high probability, we indeed have $\tilde{\psi}(S) - \tilde{\psi}(S - s_i + o_i) \leq \delta$ for every $i \in [r]$ and so

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

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

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

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 optimize a function $f : 2^X \to \mathbb{R}_{\geq 0}$ over a given matroid $\mathcal{M} = (X, \mathcal{I})$ and we let O 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 an 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*

$$f(S) \geq (1 - c/e - O(\epsilon))f(O)$$

for every $O \in \mathcal{I}$, 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 of the appendix). 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 O and non-negativity of ℓ and g, we have $\hat{v} \leq g(O) + \ell(O) \leq f(O)$. From Theorem 3.1, we can find a solution S satisfying:

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

□

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 an 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*

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

for every $O \in \mathcal{I}$, with high probability.

Proof. Define the linear and submodular functions:

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

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.

We shall consider the problem of *maximizing* $g(S) + \ell(S) = -f(X \setminus S)$ in the *dual* matroid $\mathcal{M}^*$, whose bases correspond to complements of bases of $\mathcal{M}$. We compare our solution S to this problem to the base $O^* = X \setminus O$ of $\mathcal{M}^*$. 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 $\hat{e} = \arg\max_{j \in O^*} \max(g(j), |\ell(j)|)$. 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 $\mathcal{M}_e^* = \mathcal{M}^* \,|\, X_e$, 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, $O^* \subseteq X_e$ and so $O^* \in \mathcal{B}(\mathcal{M}_e)$, as required by our analysis. Finally, from the definition of g and ℓ, we have $f(O) = -\ell(O^*) - g(O^*)$. Since $\hat{e} \in O^*$, and ℓ is nonpositive while f is nonnegative,

$$\begin{aligned}\hat{v} \leq g(O^*) + |\ell(O^*)| &= -\ell(O^*) - f(O) - \ell(O^*) \\ &\leq -2\ell(O^*).\end{aligned}$$

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

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

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, we have

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

□

We 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 using only a polynomial number of function evaluations, even in the special case that $\mathcal{M}$ is a uniform matroid (i.e., a cardinality constraint). Nemhauser and Wolsey [32] considered the problem of finding a set S of cardinality at most r that maximizes a monotone submodular function. They give a class of functions for which obtaining a $(1 - e^{-1} + \epsilon)$-approximation for any constant $\epsilon > 0$ requires a super-polynomial number of function evaluations. Our analysis uses the following additional property, satisfied by every function f in their class: let $p = \max_{i \in X} f_\emptyset(i)$, and let O be a set of size r on which f takes its maximum value. Then, $f(O) = rp$.

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 $\max\{\hat{f}(S) : |S| \leq r\}$, 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 f be a function in the family given by [32] for the cardinality constraint r, and let O be a set of size r on which f takes its maximum value. Consider the function:

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

In Lemma A.3 of the appendix, we show that $\hat{f}$ is monotone increasing, submodular, and nonnegative with curvature at most c.

We consider the problem $\max\{\hat{f}(S) : |S| \leq r\}$. Let $\alpha = (1 - ce^{-1} + \delta)$, and suppose that some algorithm returns a solution S satisfying $\hat{f}(S) \geq \alpha \cdot \hat{f}(O)$, evaluating $\hat{f}$ on only a polynomial number of sets. Because f is monotone increasing, we assume without loss of generality that $|S| = r$. Then,

$$f(S) + \frac{1-c}{c} \cdot rp \geq \alpha \cdot f(O) + \alpha \cdot \frac{1-c}{c} \cdot rp = \alpha \cdot f(O) + \alpha \cdot \frac{1-c}{c} \cdot f(O) = \frac{1}{c} \cdot \alpha \cdot f(O),$$

and so

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

Because each evaluation of $\hat{f}$ requires only a single evaluation of f, this contradicts the negative result of [32]. □

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 $\min\{\hat{f}(S) : |S| \leq r\}$, 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. Again, let f be a function in the family given by [32] for the cardinality constraint r. Let O be a set of size r on which f takes its maximum value, and recall that $f(O) = rp$, where $p = \max_{i \in X} f_{\emptyset}(i)$. Consider the function:

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

In Lemma A.4 of the appendix we show that $\hat{f}$ is monotone decreasing, supermodular, and nonnegative with curvature at most c.

We consider the problem $\min\{\hat{f}(A) : |A| \leq n - r\}$. Let $\alpha = (1 + \frac{c}{1-c}e^{-1} - \delta)$, and suppose that some algorithm returns a solution A, satisfying $\hat{f}(A) \leq \alpha \cdot \hat{f}(X \setminus O)$, evaluating $\hat{f}$ on only a polynomial number of sets.

Suppose we run the algorithm for minimizing $\hat{f}$ and return the set $S = X \setminus A$. Because $\hat{f}$ is monotone decreasing, we assume without loss of generality that $|A| = n - r$ and so $|S| = r$. Then,

$$\frac{rp}{c} - f(S) \leq \alpha \left(\frac{rp}{c} - f(O)\right) = \alpha \left(\frac{f(O)}{c} - f(O)\right) = \alpha \cdot \frac{1-c}{c} \cdot f(O),$$

and so

$$\begin{aligned} f(S) &\geq \frac{rp}{c} - \alpha \cdot \frac{1-c}{c} \cdot f(O) \\ &= \left(\frac{1}{c} - \alpha \cdot \frac{1-c}{c}\right) \cdot f(O) \\ &= \frac{1-(1-c)-ce^{-1}+(1-c)\delta}{c} \cdot f(O) \\ &= \left(1 - e^{-1} + \frac{1-c}{c} \cdot \delta\right) \cdot f(O). \end{aligned}$$

Again, since each evaluation of $\hat{f}$ requires only one evaluation of f, this contradicts the negative result of [32]. □

8 Optimizing Monotone Nonnegative Functions of 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 that f be supermodular or submodular, but only that there is a bound on the following generalized notion of curvature.

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

$$c = 1 - \min_{j \in X} \min_{S,T \subseteq X \setminus \{j\}} \frac{f_S(j)}{f_T(j)}. \tag{6}$$

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 $T = \emptyset$. Thus, (6) agrees with the standard definition of curvature given in (1). Moreover, if monotonically increasing f has curvature at most c for some $c \in [0,1]$, then for all $j \in X$ we have

$$(1-c)f_B(j) \leq f_A(j), \tag{7}$$

for any $j \in X$, and $A, B \subseteq X \setminus \{j\}$. Analogously, for monotonically decreasing function f we have

$$(1-c)f_B(j) \geq f_A(j), \tag{8}$$

for any $j \in X$, and $A, B \subseteq X \setminus \{j\}$. As with the standard notion of curvature for submodular and supermodular functions, when $c = 0$, we find that (7) and (8) 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 is a 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 $O \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$. Then,*

$$f(S) \geq (1-c)f(O).$$

Proof. Let r 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 of o_i of O so that $S - s_i + o_i \in \mathcal{I}$ for all $i \in [r]$, and let $O_i = \{o_j : j \leq i\}$ be the set containing the first i elements of O in this ordering. Then,

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

The first inequality follows from (7) 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 is a 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 $O \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$. Then,*

$$f(S) \leq \frac{1}{1-c} f(O).$$

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

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

As in the proof of Theorem 8.1, the first inequality follows from (7) 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 be a 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^* also has curvature at most c.*

Proof. 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 $S, T \subseteq X \setminus \{j\}$. Since f is monotone decreasing with curvature at most c, (8) 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 $S, T \subseteq X \setminus \{j\}$. □

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 f, we obtain the following corollaries, which 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 $O \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$, $O^* = X \setminus O$, and $S = X \setminus S^* \in \mathcal{B}(\mathcal{M})$. Then,*

$$f(S) = f^*(S^*) \geq (1-c)f^*(O^*) = (1-c)f(O).$$

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 $O \in \mathcal{B}(\mathcal{M})$ be any base of $\mathcal{M}$, $O^* = X \setminus O$, and $S = X \setminus S^* \in \mathcal{B}(\mathcal{M})$. Then,*

$$f(S) = f^*(S^*) \leq \frac{1}{1-c} f^*(O^*) = \frac{1}{1-c} f(O).$$

The approximation factors of $1-c$ and $1/(1-c)$ for maximization and minimization respectively are best possible, given curvature c. The hardness result for minimization follows from [21], 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 $\min\{f(S) : |S| \geq r\}$, where f is monotone increasing (even submodular) of curvature c. This implies that no $1/(1-c+\epsilon)$-approximation for a constant $\epsilon > 0$ is possible for this problem. Next, we prove the hardness result for maximization; this proof is based on known hardness constructions for maximization of XOS functions [13, 31].

Theorem 8.6. *For any constant $c \in (0,1)$ and $\epsilon > 0$, there is no $(1-c+\epsilon)$-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 $O \subseteq X$ be a random subset of size $k = n^{2/3}$ (assume k is an integer). We define the following function:

- $f(S) = \max\{|S \cap O|, (1-c)|S| + cn^{1/3}, \min\{|S|, n^{1/3}\}\}$.

The marginal values of f are always between $1-c$ and 1; therefore, its curvature is c. Consider a query Q. For $|Q| \leq n^{1/3}$, we have $f(Q) = |Q|$ in any case. For $|Q| > n^{1/3}$, we have $f(Q) = (1-c)|Q| + cn^{1/3}$, unless $|Q \cap O| > (1-c)|Q| + cn^{1/3}$. Since O is a random $\frac{1}{n^{1/3}}$-fraction of the ground set, we have $\mathbb{E}[|Q \cap O|] = \frac{1}{n^{1/3}}|Q| \geq 1$. Therefore, by the Chernoff bound, $\Pr[|Q \cap O| > (1-c)|Q| + cn^{1/3}]$ is exponentially small in $n^{1/3}$ (with respect to the choice of O). Furthermore, as long as $|Q \cap O| \leq (1-c)|Q| + cn^{1/3}$, $f(Q)$ depends only on $|Q|$ and hence the algorithm does not learn anything about the identity of the optimal set O. Unless the algorithm queries an exponential number of sets, it will never find a set S satisfying $|S \cap O| > (1-c)|S| + cn^{1/3}$ and hence the value of the returned solution is $f(S) \leq (1-c)|S| + cn^{1/3} \leq (1-c)n^{2/3} + cn^{1/3}$ with high probability. On the other hand, the optimum is $f(O) = |O| = n^{2/3}$. Therefore, the algorithm cannot achieve a better than $(1-c+o(1))$-approximation. □

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

9 Application: 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 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 $f^A : 2^{[n]} \to \mathbb{R}$ to be this quantity as a function of S:

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

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 $i \notin S$,

$$\min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2 \leq |f_S^A(i)| \leq \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2. \tag{9}$$

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 (9) imply that

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

We now prove the inequalities (9). 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 $\min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2 \leq |f_S^A(i)| \leq \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|^2$. □

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 dist_S 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 [2] 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

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

and

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

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

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

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 $f^A_\emptyset(1)$ 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

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

Acknowledgment We thank Christos Boutsidis for suggesting a connection between curvature and condition number.

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

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

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

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

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

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

[13] 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.

[14] Shahar Dobzinski and Jan Vondrák. From query complexity to computational complexity. In *Proc. 44th STOC*, pages 1107–1116, 2012.

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

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

[17] Yuval Filmus and Justin Ward. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *Proc. 53nd FOCS*, 2012.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

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

[26] 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.

[27] 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.

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

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

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

[31] 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.

[32] 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.

[33] 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.

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

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

[36] Jan Vondrák. Symmetry and approximability of submodular maximization problems. In *Proc. 50th FOCS*, pages 651–670, 2009.

[37] 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 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 (1), 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

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

□

Lemma 2.2. *If $f : 2^X \to \mathbb{R}_{\geq 0}$ is a 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 (1), 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

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

□

Lemma A.1. *Let $f : 2^X \to \mathbb{R}_{\geq 0}$ be a 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 submodular, monotone increasing, and nonnegative.*

Proof. The function g is the sum of a submodular function f and a linear function ℓ, and so must be submodular. For any set $A \subseteq X$ and element $j \notin A$,

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

since f is submodular. Thus, g is monotone increasing. Finally, we note that $g(\emptyset) = f(\emptyset) - \ell(\emptyset) = f(\emptyset) \geq 0$ and so g must be nonnegative. □

Lemma A.2. *Let $f : 2^X \to \mathbb{R}_{\geq 0}$ be a 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 submodular, monotone increasing, and nonnegative.*

Proof. We first show that g is monotone-increasing. Consider an arbitrary $A \subseteq X$ and $j \notin A$. 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 last line holds because f is supermodular. Moreover, note that $g(\emptyset) = -\ell(\emptyset) - f(X) = 0$, so g is nonnegative.

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,

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

□

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

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

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

Proof. Because $\hat{f}$ is the sum of a monotone increasing, nonnegative submodular function and a nonnegative linear function, and hence must be monotone increasing, nonnegative, 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. □

Lemma A.4. *Let f be a 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 submodular, monotone increasing, and nonnegative, 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. In order to see that $\hat{f}$ is decreasing, we consider the marginal $\hat{f}_A(j)$, which is equal to

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

Additionally, we note that $\hat{f}(X) = 0$, and so $\hat{f}$ must be nonnegative. 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 $\hat{f}_{X-j}(j)/\hat{f}_\emptyset(j) \leq 1 - c$. □