Continuous Submodular Maximization: Beyond DR-Submodularity

Moran Feldman
Department of Computer Science
University of Haifa
Haifa 3498838, Israel
moranfe@cs.haifa.ac.il

Amin Karbasi
School of Engineering and Applied Science
Yale University
New Haven, CT 06520
amin.karbasi@yale.edu

Abstract

In this paper, we propose the first continuous optimization algorithms that achieve a constant factor approximation guarantee for the problem of monotone continuous submodular maximization subject to a linear constraint. We first prove that a simple variant of the vanilla coordinate ascent, called COORDINATE-ASCENT+, achieves a $(\frac{e-1}{2e-1} - \varepsilon)$-approximation guarantee while performing $O(n/\varepsilon)$ iterations, where the computational complexity of each iteration is roughly $O(n/\sqrt{\varepsilon} + n \log n)$ (here, n denotes the dimension of the optimization problem). We then propose COORDINATE-ASCENT++, that achieves the tight $(1 - 1/e - \varepsilon)$-approximation guarantee while performing the same number of iterations, but at a higher computational complexity of roughly $O(n^3/\varepsilon^{2.5} + n^3 \log n/\varepsilon^2)$ per iteration. However, the computation of each round of COORDINATE-ASCENT++ can be easily parallelized so that the computational cost per machine scales as $O(n/\sqrt{\varepsilon} + n \log n)$.

1 Introduction

Submodularity is a fundamental concept in combinatorial optimization, usually associated with discrete set functions [Fujishige, 1991]. As submodular functions formalize the intuitive notion of diminishing returns, and thus provide a useful structure, they appear in a wide range of modern machine learning applications including various forms of data summarization [Lin and Bilmes, 2012, Mirzasoleiman et al., 2013], influence maximization [Kempe et al., 2003], sparse and deep representations [Balkanski et al., 2016, Oh Song et al., 2017], fairness Celis et al. [2016], Kazemi et al. [2018], experimental design [Harshaw et al., 2019], neural network interpretability Elenberg et al. [2017], human-brain mapping [Salehi et al., 2017], adversarial robustness [Lei et al., 2018], crowd teaching [Singla et al., 2014], to name a few. Moreover, submodularity ensures the tractability of the underlying combinatorial optimization problems as minimization of submodular functions can be done exactly and (constrained) maximization of submodular functions can be done approximately. For more details regarding the theory and applications of submodular functions in machine learning and signal processing, we refer the interested reader to the recent surveys by Buchbinder and Feldman [2018] and Tohidi et al. [2020].

To capture an even larger set of applications, while providing rigorous guarantees, the discrete notion of submodularity has been generalized in various directions, including adaptive and interactive submodualarity for sequential decision making problems [Golovin and Krause, 2011, Guillory and Bilmes, 2010], weak submodularity for general set functions with a bounded submodularity distance [Das and Kempe, 2011] and sequence submodularity for time series analysis [Tschiatschek et al., 2017, Mitrovic et al., 2019], among other variants.

Very recently, a surge of new applications in machine learning and statistics motivated researchers to study continuous submodular functions [Bach, 2015, Wolsey, 1982a], a large class of non-convex/non-

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

concave functions, which may be optimized efficiently. In particular, it has been shown that continuous submodular minimization can be done exactly Bach [2015]. In contrast, for continuous submodular maximization, it is usually assumed that the continuous function is not only submodular, but also has the extra condition of diminishing returns. Such functions are usually called continuous DR-submodular [Bian et al., 2017b]. We should highlight that even though in the discrete domain, submodularity and diminishing returns are equivalent; in the continuous domain, the diminishing returns condition implies continuous submodularity, but not vice versa.

In this paper, we propose the first algorithms that achieve constant factor approximation guarantees for the maximization of a monotone continuous submodular function subject to a linear constraint. More specifically, **our contributions** can be summarized as follows:

- We develop a variant of the coordinate ascent algorithm, called COORDINATE-ASCENT+, that achieves a $(\frac{e-1}{2e-1} - \varepsilon)$-approximation guarantee while performing $O(n/\epsilon)$ iterations, where the computational complexity of each iteration is $O(n\sqrt{B/\varepsilon} + n \log n)$. Here, n and B denote the dimension of the optimization problem and the ℓ_1 radius of the constraint set, respectively.
- We then develop COORDINATE-ASCENT++, that achieves the tight $(1 - 1/e - \varepsilon)$ approximation guarantee while performing $O(n/\epsilon)$ iterations, where the computational complexity of each iteration is $O(n^3\sqrt{B}/\varepsilon^{2.5} + n^3 \log n/\varepsilon^2)$. Moreover, COORDINATE-ASCENT++ can be easily parallelized so that the computational complexity per machine in each round scales as $O(n\sqrt{B/\epsilon} + n \log n)$.

Notably, to establish these results, we do not assume that the continuous submodular function satisfies the diminishing returns condition.

1.1 Related Work

Continuous submodular functions naturally arise in many machine learning applications such as Adwords for e-commerce and advertising [Mehta et al., 2007, Devanur and Jain, 2012], influence and revenue maximization [Bian et al., 2017b], robust budget allocation [Staib and Jegelka, 2017], multi-resolution data summarization [Bian et al., 2017b], learning assignments [Golovin et al., 2014], experimental design [Chen et al., 2018b], and MAP inference for determinantal point processes [Gillenwater et al., 2012, Hassani et al., 2019]. Continuous submodular functions have also been studied in statistics as negative log-densities of probability distributions. These distributions are referred to as multivariate totally positive of order 2 (MTP2) [Fallat et al., 2017] and classical examples are the multivariate logistic, Gamma and F distributions, as well as characteristic roots of random Wishart matrices [Karlin and Rinott, 1980].

The focus of the current work is to study continuous submodular maximization. Almost all the existing works in this area consider a proper subclass of continuous submodular functions, called continuous DR-submodular, which satisfy diminishing returns conditions. In particular, when first order information (i.e., exact or stochastic gradients) is available Hassani et al. [2017] showed that (stochastic) gradient ascent achieves $1/2$-approximation guarantee for monotone continuous DR-submodular functions subject to a general convex body constraint.[1] Interestingly, one can achieve the tight approximation guarantee of $1 - 1/e$ by using conditional gradient methods [Bian et al., 2017b] or its efficient stochastic variants [Mokhtari et al., 2018a, Karbasi et al., 2019, Zhang et al., 2020]. A simple variant of the conditional gradient methods can also be applied to non-monotone DR-submodular functions, which results in a $1/e$-approximation guarantee [Bian et al., 2017a, Mokhtari et al., 2018c, Hassani et al., 2019]. The only work, we are aware of, that goes beyond the above line of work, and considers also non-DR continuous submodular functions is a recent work by Niazadeh et al. [2018], which developed a polynomial time algorithm with a tight $1/2$-approximation guarantee for the problem of continuous submodular maximization subject to a box constraint.

Discrete and continuous submodular maximization problems are inherently related to one another through the multilinear extension [Calinescu et al., 2011a]. Indeed, maximization of the multilinear extension (along with a subsequent rounding) has led to the best theoretical results in many settings,

[1] We note that the result of Hassani et al. [2017] can be viewed as an adaptation of an algorithm designed by Chekuri et al. [2011] for multilinear extensions of discrete monotone submodular functions.

including submodular maximization subject to various complex constraints [Feldman et al., 2011, Chekuri et al., 2014, Buchbinder and Feldman, 2019], online and bandit submodular maximization [Zhang et al., 2019, Chen et al., 2018a], decentralized solution [Mokhtari et al., 2018b, Xie et al., 2019], and algorithms with low adaptivity complexity Chekuri and Quanrud [2019], Balkanski et al. [2019], Chen et al. [2019], Ene et al. [2019].

Remark: Soma and Yoshida [2018] considered the closely related problem of maximizing a non-DR submodular function subject to a cardinality constraint over the integer lattice. In principle, any algorithm for their setting can be applied to our setting using a reduction at the cost of a small decrease in the approximation guarantee and some increase in the time complexity.[2] However, an error was recently found in the analysis of the algorithm of [Soma and Yoshida, 2018], and therefore, this algorithm cannot currently be used to get an algorithm for our setting via the above mentioned reduction. It is likely that the algorithm of [Soma and Yoshida, 2018] can be fixed by introducing an enumeration step similar to the one we describe in Section 5, but such a modification will significantly increase the time complexity of their algorithm. In Appendix A in the supplementary material we give the reduction from our settings to the setting of [Soma and Yoshida, 2018] and also describe the error that was found in the algorithm of [Soma and Yoshida, 2018].

2 Preliminaries and Problem Formulation

We first recall a few standard definitions regarding submodular functions. Even though submodularity is mostly considered in the discrete domain, the notion can be naturally extended to arbitrary lattices [Fujishige, 1991]. To this end, let us consider a subset of $\mathbb{R}_+^d$ of the form $\mathcal{X} = \prod_{i=1}^d \mathcal{X}_i$ where each $\mathcal{X}_i$ is a compact subset of $\mathbb{R}_+$. A function $F\colon \mathcal{X} \to \mathbb{R}_+$ is *submodular* [Wolsey, 1982b] if for all $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{X}$, we have

$$F(\mathbf{x}) + F(\mathbf{y}) \geq F(\mathbf{x} \vee \mathbf{y}) + F(\mathbf{x} \wedge \mathbf{y}) \ ,$$

where $\mathbf{x} \vee \mathbf{y} \doteq \max(\mathbf{x}, \mathbf{y})$ (component-wise) and $\mathbf{x} \wedge \mathbf{y} \doteq \min(\mathbf{x}, \mathbf{y})$ (component-wise). A submodular function is monotone if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$, we have $F(\mathbf{x}) \leq F(\mathbf{y})$ (here, by $\mathbf{x} \leq \mathbf{y}$ we mean that every element of $\mathbf{x}$ is less than that of $\mathbf{y}$). The above definition includes the discrete notion of submodularity over a set by restricting each $\mathcal{X}_i$ to $\{0, 1\}$. In this paper, we mainly consider *continuous submodular* functions, where each $\mathcal{X}_i$ is a closed interval in $\mathbb{R}_+$. When F is twice differentiable, a continuous function is submodular if and only if all cross-second-derivatives are non-positive [Bach, 2015], i.e.,

$$\forall i \neq j, \forall \mathbf{x} \in \mathcal{X}, \ \frac{\partial^2 F(\mathbf{x})}{\partial x_i \partial x_j} \leq 0 \ .$$

Thus, continuous submodular functions can be convex (e.g., $F(\mathbf{x}) = \sum_{i,j} \phi_{i,j}(x_i - x_j)$ for $\phi_{i,j}$ convex), concave (e.g., $F(\mathbf{x}) = g(\sum_{i=1}^n \lambda_i x_i)$ for g concave and λ_i's non-negative), and neither (e.g., quadratic program $F(\mathbf{x}) = \mathbf{x}^T H \mathbf{x}$ where all off-diagonal elements of H are non-positive).

A proper subclass of continuous submodular functions are called *DR-submodular* [Bian et al., 2017b, Soma and Yoshida, 2015] if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$, standard basis vector $\mathbf{e}_i \in \mathbb{R}_+^n$ and a non-negative number $z \in \mathbb{R}_+$ such that $z\mathbf{e}_i + \mathbf{x} \in \mathcal{X}$ and $z\mathbf{e}_i + \mathbf{y} \in \mathcal{X}$, it holds that $F(z\mathbf{e}_i + \mathbf{x}) - F(\mathbf{x}) \geq F(z\mathbf{e}_i + \mathbf{y}) - F(\mathbf{y})$.[3] One can easily verify that for a differentiable DR-submodular function the gradient is an antitone mapping, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ such that $\mathbf{x} \leq \mathbf{y}$ we have $\nabla F(\mathbf{x}) \geq \nabla F(\mathbf{y})$ [Bian et al., 2017b]. An important example of a DR-submodular function is the multilinear extension [Calinescu et al., 2011b]. Maybe the simplest example of a continuous submodular function that is not DR-submodular is the quadratic function $F(x) = x^T H x + h^T x + b$ where only the off-diagonal entries of H are non-positive (and there is no restrictions on the diagonal entries). Moreover, as we mentioned in the introduction, continuous submodular functions naturally arise as the negative log-densities of probability distributions. For instance, a distribution p on $\mathcal{X}$ is called MTP2 if $p(x)p(y) \leq p(x \vee y)p(x \wedge y)$ for all $x, y \in \mathcal{X} \subseteq \mathbb{R}$. MTP2 implies positive association between random variables. As an example, a multivariat Gaussian distibution is MTP2

[2] The reduction between the two settings is a bit subtle, and we would like to thank an anonymous referee for suggesting its crux idea to us.

[3] It is well-known that (non-DR) continuous submodularity is equivalent to requiring this inequality to hold whenever the vectors $\mathbf{x}$ and $\mathbf{y}$ also obey $x_i = y_i$.

if an only if its inverse covariance matrix has non-positive off-diagonal entries. Therefore, finding the the most likely configuration in this setting amounts to maximizing a continuous submodular function.

The definition of DR-submodular functions implies that decreasing coordinates of a vector affects the contribution of these coordinates to the value of the vector in at most a proportional way. More formally, if $\mathbf{x}$ and $\mathbf{y}$ are two vectors such that $\mathbf{x}, \mathbf{x}+\mathbf{y} \in \mathcal{X}$, then for every real value $c \in [0,1]$ it holds that $F(\mathbf{x}+c\mathbf{y}) \geq F(\mathbf{x}) + c[F(\mathbf{x}+\mathbf{y}) - F(\mathbf{x})]$. This property is very helpful in algorithms designed for DR-submodular functions since it allows one to lower bound the increase in the value of a given solution $\mathbf{x}$ when some coordinate i of $\mathbf{x}$ is increased even when the increase does not make x_i as large as the value of coordinate i in some optimal solution. For (non-DR) continuous submodular functions this cannot be done, and thus, a coordinate whose value cannot be increased to match its value in some optimal solution (due to the constraint and incorrect decisions already made by the algorithm) is basically useless for the algorithm; rendering the design of algorithms for (non-DR) continuous submodular functions much more challenging than for DR-submodular functions.

In this paper, we consider the following fundamental optimization problem

$$\max_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}) \text{ subject to } \|\mathbf{x}\|_1 \leq B, \tag{1}$$

where F is a non-negative monotone continuous submodular function. Without loss of generality, we assume that each closed interval $\mathcal{X}_i$ is of the form $[0, u_i]$ since otherwise for $\mathcal{X}_i = [a_i, a_i + u_i]$ we can always define a corresponding continuous submodular function $G(\mathbf{x}) = F(\mathbf{x}+\mathbf{a})$, where $\mathbf{a} = [a_1, \ldots, a_n]$. Similarly, we assume w.l.o.g., that $u_i \leq B$ for every coordinate i. We also assume that F is L-smooth, meaning that $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\|_2 \leq L\|\mathbf{x}-\mathbf{y}\|_2$ for some $L \geq 0$ and for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Finally, note that replacing a linear constraint of the form $\sum_{i=1}^n w_i x_i \leq B$ (where $w_i > 0$) with $\|\mathbf{x}\|_1 \leq B$ does not change the nature of the problem. In this case, we can simply define a corresponding function $G(\mathbf{x}) = F\left(\sum_{i=1}^n x_i \mathbf{e}_i / w_i\right)$ and solve Problem (1). This change of course changes L by a factor of $W = \min_{1\leq i\leq n} w_i$. Prior to our work, no constant approximation guarantee was known for Problem (1).

3 Plain Coordinate Ascent

In this section we present our plain coordinate ascent algorithm and analyze its guarantee. Our algorithm uses as a black box an algorithm for a one dimensional optimization problem whose properties are summarized by the following proposition. We include the proof of this proposition in Appendix B in the supplementary material.

Proposition 3.1. *Given a point $\mathbf{x} \in [\mathbf{0}, \mathbf{u}]$, a coordinate $i \in [n]$, bounds $0 < a \leq b \leq u_i - x_i$ and a positive parameter $\varepsilon \in (0,1)$ there is a polynomial time algorithm that runs in $O(\sqrt{B/\varepsilon} + \log(\varepsilon/a))$ time and returns a value $y \in [a,b]$ maximizing the ratio $F(\mathbf{x}+y\mathbf{e}_i)/y$ up to an additive error of εL.*

Using the algorithm whose existence is guaranteed by the last proposition, we can now formally state our coordinate ascent algorithm as Algorithm 1. This algorithm gets a quality control parameter $\varepsilon \in (0, 1/4)$.

We begin the analysis of Algorithm 1 with the following observation that bounds its time complexity. The proof of this observation and some other proofs appearing later in this paper have been moved to Appendix D in the supplementary material due to space constraints. In a nutshell, the observation holds since each iteration of the main loop of Algorithm 1 producees progress in at least one of three ways: making $x_j = u_j$ for some coordinate j, making $\|\mathbf{x}\|_1$ as large as B, or increasing $\|\mathbf{x}\|_1$ by at least δ.

Observation 3.2. *The main loop of Algorithm 1 makes at most $O(n/\varepsilon)$ iterations, each running in $O(n\sqrt{B/\varepsilon} + n\log n)$ time. Thus, the entire algorithm runs in $O(n^2\sqrt{B}/\varepsilon^{1.5} + n^2 \log n/\varepsilon)$ time.*

Fix now some feasible solution $y \in [\mathbf{0}, \mathbf{u}]$. Intuitively, we say that an iteration of the main loop of Algorithm 1 is *good* (with respect to $\mathbf{y}$) if, at the beginning of the iteration, the algorithm still has the option to increase each coordinate of $\mathbf{x}$ to be equal to the corresponding coordinate of $\mathbf{y}$, and this does not violate the constraint (if a coordinate of $\mathbf{x}$ is already larger than the corresponding coordinate of $\mathbf{y}$ that is fine as well). Formally, an iteration is good if the inequality $y_i - x_i \leq d'_i$ was true in this iteration for every coordinate $i \in C$ (before the vector $\mathbf{x}$ was updated at the end of the iteration). Let

Algorithm 1: COORDINATE-ASCENT (ε)

Let $\mathbf{x} \leftarrow \mathbf{0}$ and $\delta \leftarrow \varepsilon B/n$.
while $\|\mathbf{x}\|_1 \leq B$ **do**
  Let $C \subseteq [n]$ be the set of coordinates $i \in [n]$ for which $x_i < u_i$ (i.e., these coordinates can be increased in $\mathbf{x}$ to some positive extent without violating feasibility).
  for *every* $i \in C$ **do**
    Let d'_i be the maximum amount by which x_i can be increased without violating feasibility. Formally, $d'_i = \min\{u_i - x_i, B - \|\mathbf{x}\|_1\}$.
    Use the algorithm suggested by Proposition 3.1 to find a value $d_i \in [\min\{d'_i, \delta\}, d'_i]$ maximizing $\frac{F(\mathbf{x}+d_i\mathbf{e}_i)-F(\mathbf{x})}{d_i}$ up to an additive error of εL.
  Let j be the coordinate of C maximizing $\frac{F(\mathbf{x}+d_j\mathbf{e}_j)-F(\mathbf{x})}{d_j}$, and update $\mathbf{x} \leftarrow \mathbf{x} + d_j\mathbf{e}_j$.
return $\mathbf{x}$.

ℓ denote the number of good iterations of the main loop of Algorithm 1, and let us denote by $\mathbf{x}^{(h)}$ the value of $\mathbf{x}$ after h iterations for every $0 \leq h \leq \ell$. Using this notation, we can now state and prove the following lemma, which provides a lower bound on the value of $\mathbf{x}$ after any number of (good) iterations of Algorithm 1.

Lemma 3.3. *For every vector* $\mathbf{y} \in [\mathbf{0}, \mathbf{u}]$ *and integer* $0 \leq h \leq \ell$, $F(\mathbf{x}^{(h)}) \geq (1 - e^{-\|\mathbf{x}^{(h)}\|_1/(\|y\|_1+\varepsilon B)}) \cdot F(\mathbf{y}) - \|\mathbf{x}^{(h)}\|_1 \cdot \varepsilon L$.

Proof. We prove the lemma by induction on h. For $h = 0$, $\|\mathbf{x}^{(h)}\|_1 = 0$, and the lemma follows from the non-negativity of F. Thus, it remains to prove the lemma for some $h > 0$ given that it holds for $h - 1$. From this point on we restrict our attention to iteration number h of Algorithm 1, and thus, when we refer to variables such as C and d_i, these variables should be understood as taking the values they are assigned in this iteration. Given this assumption, for every $i \in C$, let us now define a value o_i that is closest to $y_i - x_i^{(h-1)}$ among all the values in the range to which d_i can belong. Formally,

$$o_i = \min\{\max\{y_i - x_i^{(h-1)}, \min\{\delta, d'_i\}\}, d'_i\} = \max\{y_i - x_i^{(h-1)}, \min\{\delta, d'_i\}\} \quad \forall\, i \in C \ ,$$

where the equality holds since the fact that the iteration we consider is a good iteration implies $y_i - x_i^{(h-1)} \leq d'_i$. Since o_i is a valid choice for d_i, we get by the definition of d_i that

$$\frac{F(\mathbf{x}^{(h-1)} + d_i\mathbf{e}_i) - F(\mathbf{x}^{(h-1)})}{d_i} \geq \frac{F(\mathbf{x}^{(h-1)} + o_i\mathbf{e}_i) - F(\mathbf{x}^{(h-1)})}{o_i} - \varepsilon L \ .$$

Using the definition of j and the submodularity of F, the last inequality implies

$$\begin{aligned} &\frac{F(\mathbf{x}^{(h-1)} + d_j\mathbf{e}_j) - F(\mathbf{x}^{(h-1)})}{d_j} \geq \frac{\sum_{i\in C} o_i \cdot \frac{F(\mathbf{x}^{(h-1)}+d_i\mathbf{e}_i)-F(\mathbf{x}^{(h-1)})}{d_i}}{\sum_{i\in C} o_i} \\ &\geq \frac{\sum_{i\in C}[F(\mathbf{x}^{(h-1)} + o_i\mathbf{e}_i) - F(\mathbf{x}^{(h-1)})]}{\sum_{i\in C} o_i} - \varepsilon L \geq \frac{F(\mathbf{x}^{(h-1)} + \sum_{i\in C} o_i\mathbf{e}_i) - F(\mathbf{x}^{(h-1)})]}{\sum_{i\in C} o_i} - \varepsilon L \ . \end{aligned} \tag{2}$$

To understand the rightmost side of the last inequality, we need the following two bounds.

$$\sum_{i\in C} o_i \leq \sum_{i\in C} \max\{y_i, \delta\} \leq \sum_{i\in C} y_i + n\delta \leq \|\mathbf{y}\|_1 + \varepsilon B \ ,$$

and

$$\mathbf{x}^{(h-1)} + \sum_{i\in C} o_i\mathbf{e}_i \geq \mathbf{x}^{(h-1)} + \sum_{i\in C}(y_i - x_i^{(h-1)})\mathbf{e}_i \geq \mathbf{y} \ .$$

Plugging these bounds into Inequality (2), and using the monotonicity of F, we get

$$\frac{F(\mathbf{x}^{(h-1)} + d_j\mathbf{e}_j) - F(\mathbf{x}^{(h-1)})}{d_j} \geq \frac{F(\mathbf{y}) - F(\mathbf{x}^{(h-1)})}{\|\mathbf{y}\|_1 + \varepsilon B} - \varepsilon L \ .$$

Since $\mathbf{x}^{(h)} = \mathbf{x}^{(h-1)} + d_j \mathbf{e}_j$, the last inequality now yields the following lower bound on $F(\mathbf{x}^{(h)})$.

$$\begin{aligned} F(\mathbf{x}^{(h)}) &\geq F(\mathbf{x}^{(h-1)}) + \frac{d_j}{\|\mathbf{y}\|_1 + \varepsilon B} \cdot [F(\mathbf{y}) - F(\mathbf{x}^{(h-1)})] - \varepsilon L d_j \\ &\geq \left(1 - \frac{d_j}{\|\mathbf{y}\|_1 + \varepsilon B}\right) \cdot F(\mathbf{x}^{(h-1)}) + \frac{d_j}{\|\mathbf{y}\|_1 + \varepsilon B} \cdot F(\mathbf{y}) - \varepsilon L d_j \ . \end{aligned}$$

Finally, plugging into the last inequality the lower bound on $F(\mathbf{x}^{(h-1)})$ given by the induction hypothesis, we get

$$\begin{aligned} F(\mathbf{x}^{(h)}) \geq& \left(1 - \frac{d_j}{\|\mathbf{y}\|_1 + \varepsilon B}\right) \cdot \left\{(1 - e^{-\|\mathbf{x}^{(h-1)}\|_1/(\|\mathbf{y}\|_1 + \varepsilon B)}) \cdot F(\mathbf{y}) - \|\mathbf{x}^{(h-1)}\|_1 \cdot \varepsilon L\right\} \\ &+ \frac{d_j}{\|\mathbf{y}\|_1 + \varepsilon B} \cdot F(\mathbf{y}) - \varepsilon L d_j \\ \geq& \left(1 - e^{-(\|x^{(h-1)}\|_1 + d_j)/(\|\mathbf{y}\|_1 + \varepsilon B)}\right) \cdot F(\mathbf{y}) - (\|x^{(h-1)}\|_1 + d_j) \cdot \varepsilon L \\ =& \left(1 - e^{-\|x^{(h)}\|_1/(\|\mathbf{y}\|_1 + \varepsilon B)}\right) \cdot F(\mathbf{y}) - \|x^{(h)}\|_1 \cdot \varepsilon L \ . \end{aligned}$$

□

Our next objective is to get an approximation guarantee for Algorithm 1 based on the last lemma. Such a guarantee appears below as Corollary 3.5. However, to prove it we also need the following observation, which shows that lower bounding the value of $F(x)$ at some point during the execution of Algorithm 1 implies the same bound also for the value of the final solution of the algorithm.

Observation 3.4. *The value of $F(\mathbf{x})$ only increases during the execution of Algorithm 1.*

Proof. The observation follows from the monotonicity of F since d_j is always non-negative. □

Let opt be some optimal solution vector.

Corollary 3.5. *Let $\mathbf{x}_{\text{CA}}$ be the vector outputted by Algorithm 1, then $F(\mathbf{x}_{\text{CA}}) \geq (1 - 1/e - B^{-1} \cdot \max_{i \in [n]} u_i - \varepsilon) \cdot F(\mathsf{opt}) - \varepsilon BL$.*

The guarantee of Corollary 3.5 is close to an approximation ratio of $1 - 1/e$ when the upper bound u_i is small compared to B for every $i \in [n]$. In the next two sections we describe enhanced versions of our coordinate ascent algorithm that give an approximation guarantee which is independent of this assumption. We note that, formally, the analyses of these enhanced versions are independent of Corollary 3.5. However, the machinery used to prove this corollary is reused in these analyses.

4 Fast Enhanced Coordinate Ascent

In this section we describe one simple and fast way to enhance the plain coordinate ascent algorithm from Section 3, leading to the algorithm that we name COORDINATE-ASCENT+. Before describing COORDINATE-ASCENT+ itself, let us give a different formulation for the guarantee of Algorithm 1.

Lemma 4.1. *There is a coordinate $j \in [n]$ such that the output $\mathbf{x}_{\text{CA}}$ of Algorithm 1 has a value of at least $(1 - 1/e - 2\varepsilon) \cdot F(\mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j) - \varepsilon BL$.*

Proof. In this proof we use the notation from Section 3, and consider the last iteration ℓ' during this execution in which there is no coordinate $i \in [n]$ such that $\mathsf{opt}_i - x_i > d'_i$ (where x_i represents here its value at the beginning of the iteration). If ℓ' is the last iteration of Algorithm 1, then all the iterations of Algorithm 1 are good when we choose $\mathbf{y} = \mathsf{opt}$. Thus, for this choice of $\mathbf{y}$ we get $\|\mathbf{x}^{(\ell')}\|_1 = B$, and by Lemma 3.3 the value of the output $\mathbf{x}_{\text{CA}} = \mathbf{x}^{(\ell')}$ of Algorithm 1 is at least

$$(1 - e^{-B/(\|\mathsf{opt}\|_1 + \varepsilon B)}) \cdot F(\mathsf{opt}) - \varepsilon BL \geq (1 - e^{-1} - \varepsilon) \cdot F(\mathsf{opt}) - \varepsilon BL \ ,$$

where the inequality holds since $\|\mathsf{opt}\|_1 \leq B$. This guarantee is stronger than the guarantee of the lemma (because of the monotonicity of F), and thus, completes the proof for the current case.

Consider now the case in which iteration ℓ' is not the last iteration of Algorithm 1. In this case we set j to be some coordinate in $[n]$ for which the inequality $\mathsf{opt}_j - x_j^{(\ell')} > d'_j$ holds. Choosing

$\mathbf{y} = \mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j$, we get that Algorithm 1 has at least ℓ' good iterations. There are now two cases to consider based on the relationship between $\|\mathbf{y}\|_1$ and B. If $\|\mathbf{y}\|_1 \geq B/2$, then Lemma 3.3 and Observation 3.4 imply together that the value of $\mathbf{x}_{\text{CA}}$ is at least

$$\begin{aligned} F(\mathbf{x}_{\text{CA}}) &\geq F(\mathbf{x}^{(\ell')}) \geq (1 - e^{-(B-\mathsf{opt}_j)/(B+\varepsilon B-\mathsf{opt}_j)}) \cdot F(\mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j) - \varepsilon BL \\ &\geq (1 - e^{2\varepsilon - 1}) \cdot F(\mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j) - \varepsilon BL \geq (1 - e^{-1} - 2\varepsilon) \cdot F(\mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j) - \varepsilon BL \ , \end{aligned}$$

where the second inequality holds since $\|\mathbf{x}^{(\ell')}\|_1 \leq B$, but $\mathsf{opt}_j - x_j^{(\ell')} > d'_j = B - \|\mathbf{x}^{(\ell')}\|_1$ (the last equality holds since the inequalities $\mathsf{opt}_j - x_j^{(\ell')} > d'_j$ and $u_j \geq \mathsf{opt}_j$ exclude the possibility of $d'_j = u_j - x_j^{(\ell')}$). The penultimate inequality holds since, by our assumption, $B - \mathsf{opt}_j \geq \|\mathbf{y}\|_1 \geq B/2$.

It remains to consider the caes in which $\|\mathbf{y}\|_1 \leq B/2$. In this case, for every coordinate $i \in [n]$ we have $y_i - x_i \leq y_i \leq B/2 \leq B - \|\mathbf{x}\|_1$ as long as $\|\mathbf{x}\|_1 \leq B/2$. Thus, all the iterations of Algorithm 1 are good until $\|\mathbf{x}\|_1$ gets to a size lager than $B/2$; which implies $\|\mathbf{x}^{(\ell')}\|_1 \geq B/2$. Hence, Lemma 3.3 and Observation 3.4 allow us to lower bound $F(\mathbf{x}_{\text{CA}})$ also by

$$\begin{aligned} F(\mathbf{x}_{\text{CA}}) &\geq F(\mathbf{x}^{(\ell')}) \geq (1 - e^{-(B/2)/(B/2+\varepsilon B)}) \cdot F(\mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j) - \varepsilon BL \\ &\geq (1 - e^{2\varepsilon - 1}) \cdot F(\mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j) - \varepsilon BL \geq (1 - e^{-1} - 2\varepsilon) \cdot F(\mathsf{opt} - \mathsf{opt}_j \mathbf{e}_j) - \varepsilon BL \ , \end{aligned}$$

where the second inequality follows from the above discussion and the inequality $\|\mathbf{x}^{(\ell')}\|_1 \leq B$ which holds since $\mathbf{x}^{(\ell')}$ is a feasible solution. □

We are now ready to present the enhanced algorithm COORDINATE-ASCENT+, which appears as Algorithm 2. The enhancement done in this algorithm, and its analysis, is related to an algorithm of Cohen and Katzir [2008] obtaining the same approximation guarantee for the special case of discrete monotone submodular functions.

Algorithm 2: COORDINATE-ASCENT+ (ε)

Let $\mathbf{x}_{\text{CA}}$ be the solution produced by Algorithm 1 when run with ε.
Let $\mathbf{x}_{\text{CA+}}$ be the best solution among the $n+1$ solutions $\mathbf{x}_{\text{CA}}$ and $\{u_i \cdot \mathbf{e}_i\}_{i \in [n]}$.
return $\mathbf{x}_{\text{CA+}}$.

It is clear that the time complexity of Algorithm 2 is dominated by the time complexity of Algorithm 1. Thus, we only need to analyze the approximation ratio of Algorithm 2. This is done by the next theorem, whose proofs relies on the fact that one of the solutions checked by Algorithm 2 is $u_j e_j$ for the coordinate j whose existence is guaranteed by Lemma 4.1.

Theorem 4.2. *Algorithm 2 outputs a solution of value at least* $\left(\frac{e-1}{2e-1} - 2\varepsilon\right) \cdot F(OPT) - \varepsilon BL \geq (0.387 - 2\varepsilon) \cdot F(OPT) - \varepsilon BL$. *It has* $O(n/\varepsilon)$ *iterations, each running in* $O(n\sqrt{B/\varepsilon} + n \log n)$ *time, which yields a time complexity of* $O(n^2\sqrt{B}/\varepsilon^{1.5} + n^2 \log n/\varepsilon)$.

5 Optimal Approximation Ratio

In this section we describe a more involved way to enhance the plain coordinate ascent algorithm from Section 3, which leads to the algorithm that we name COORDINATE-ASCENT++ and achieves the optimal approximation ratio of 1 - 1/e (up to some error term). This enhancement uses as a black box an algorithm for a one dimensional optimization problem whose properties are summarized by the following proposition. We include the proof of this proposition in Appendix C in the supplementary material.

Proposition 5.1. *Given a point* $\mathbf{x} \in [\mathbf{0}, \mathbf{u}]$, *a coordinate* $i \in [n]$, *a target value* $F(\mathbf{x}) \leq v \leq F(\mathbf{x} \vee u_i \mathbf{e}_i)$ *and a positive parameter* $\varepsilon \in (0, 1)$, *there is a polynomial time algorithm that runs in* $O(\log(B/\varepsilon))$ *time and returns a value* $0 \leq y \leq u_i - x_i$ *such that*

- $F(\mathbf{x} + y\mathbf{e}_i) \geq v - \varepsilon L$.

- *There is no value* $0 \leq y' < y$ *such that* $F(\mathbf{x} + y'\mathbf{e}_i) \geq v$.

We can now give a simplified version of COORDINATE-ASCENT++, which appears as Algorithm 3. For simplicity, we assume in the description and analysis of this algorithm that $n \geq 3$. If this is not the case, one can simulate it by adding dummy coordinates that do not affect the value of the objective function. Algorithm 3 starts by guessing two coordinates h_1 and h_2 that contribute a lot of value to opt. Then it constructs a solution $\mathbf{x}$ with a small support using two executions of the algorithm whose existence is guaranteed by Proposition 5.1, one execution for each one of the coordinates h_1 and h_2. It then completes the solution $\mathbf{x}$ into a full solution by executing Algorithm 1 after "contracting" the coordinates h_1 and h_2, i.e., modifying the objective function so that it implicitly assumes that these coordinates take the values they take in $\mathbf{x}$.

Algorithm 3: COORDINATE-ASCENT++ (SIMPLIFIED) (ε)

Guess the coordinate $h_1 \in [n]$ maximizing $F(\mathsf{opt}_{h_1} \cdot \mathbf{e}_{h_1})$ and the coordinate $h_2 \in [n] \setminus \{h_1\}$ other than h_1 maximizing $F(\sum_{i \in \{h_1, h_2\}} \mathsf{opt}_i \cdot \mathbf{e}_i)$.
Let $\mathbf{x} \leftarrow \mathbf{0}$.
for $i = 1$ **to** 2 **do**
  Guess a value v_i obeying

$$\max\{F(\mathbf{x}), F(\mathbf{x} + \mathsf{opt}_{h_i}\mathbf{e}_{h_i}) - \varepsilon \cdot F(\mathsf{opt})\} \leq v_i \leq F(\mathbf{x} + \mathsf{opt}_{h_i}\mathbf{e}_{h_i}) \ .$$

  Let y_i be the value returned by the algorithm whose existence is guaranteed by Proposition 5.1 given $\mathbf{x}$ as the input vector, the coordinate h_i and the target value v_i.
  Update $\mathbf{x} \leftarrow \mathbf{x} + y_i\mathbf{e}_{h_i}$.
Execute Algorithm 1 on the instance obtained by removing the coordinates h_1 and h_2, replacing the objective function with $F'(\mathbf{x}') = F(\mathbf{x}' + \mathbf{x}) - F(\mathbf{x})$ and decreasing B by $\|\mathbf{x}\|_1$. Let $\mathbf{x}_{\text{CA}}$ be the output of Algorithm 1.
Return $\mathbf{x} + \mathbf{x}_{\text{CA}}$ (we denote this sum by $\mathbf{x}_{\text{CA++}}$ in the analysis).

We begin the analysis of Algorithm 3 by bounding its time complexity.

Observation 5.2. *Assuming the guesses made by Algorithm 3 do not require any time, Algorithm 3 has* $O(n/\varepsilon)$ *iterations, each running in* $O(n\sqrt{B/\varepsilon} + n \log n)$ *time, which yields a time complexity of* $O(n^2\sqrt{B}/\varepsilon^{1.5} + n^2 \log n/\varepsilon)$.

The next step in the analysis of Algorithm 3 is proving some properties of the vector $\mathbf{x} = \sum_{i \in \{h_1, h_2\}} y_i \cdot \mathbf{e}_i$ produced by the first part of the algorithm. In a nutshell, these properties holds since the definition of v_j and the properties of Proposition 5.1 show together that the value chosen for x_{h_j} by the algorithm of Proposition 5.1 gives almost as much value as choosing opt_{h_j}, but it never overestimates opt_{h_j}.

Lemma 5.3. $F(\mathbf{x}) \geq F(\sum_{j=1}^{2} \mathsf{opt}_{h_j} \cdot \mathbf{e}_{h_j}) - 2\varepsilon \cdot F(\mathsf{opt}) - 2\varepsilon L$ *and* $\mathbf{x} \leq \sum_{j=1}^{2} \mathsf{opt}_{h_j} \cdot \mathbf{e}_{h_j}$.

We now ready to prove the approximation guarantee of Algorithm 3. Intuitively, this proof is based on simply adding up the lower bound on $F(\mathbf{x})$ given by Lemma 5.3 and the lower bound on $F'(\mathbf{x}_{\text{CA}})$ given by Lemma 4.1. Some of the ideas used in the proof can be traced back to a recent result by Nutov [2020], who described an algorithm achieving $(1 - 1/e)$-approximation for the discrete version of the problem we consider (namely, maximizing a non-negative monontone discrete submodular function subject to a knapsack constraint) using $O(n^4)$ function evaluations.

Lemma 5.4. *Algorithm 3 outputs a vector* $\mathbf{x}_{\text{CA++}}$ *whose value is at least* $(1 - 1/e - 4\varepsilon) \cdot F(\mathsf{opt}) - \varepsilon(B + 2)L$.

To get our final COORDINATE-ASCENT++ algorithm, we need to explain how to implement the guesses of Algorithm 3. The coordinates h_1 and h_2 can be guessed by simply iterating over all the possible pairs of two coordinates. Similarly, by the next observation, to get v_i it suffices to try all the possible values in the set $\{F(\mathbf{x}) + \varepsilon j \cdot F(u_{h_i}\mathbf{e}_{h_i}) \mid j$ is a non-negative integer and $F(\mathbf{x}) + \varepsilon j \cdot F(u_{h_i}\mathbf{e}_{h_i}) \leq F(\mathbf{x} + u_{h_i}\mathbf{e}_{h_i})\}$. In the following, we refer to this set as $\mathcal{J}(\mathbf{x}, h_i)$.

Observation 5.5. *Consider the vector* $\mathbf{x}$ *at the point in which Algorithm 3 guesses the value* v_i*. Then, there exists a value in the set* $\mathcal{J}(\mathbf{x}, h_i)$ *obeying the requirements from* v_i*.*

Our final COORDINATE-ASCENT++ algorithm appears as Algorithm 4. By the above discussion, the number of iterations it makes exceeds the number of iterations given by Observation 5.2 only by a factor of

$$n^2 \cdot \prod_{i=1}^{2} |\mathcal{J}(\mathbf{x}, h_i)| \leq n^2 \cdot \prod_{i=1}^{2} \left(1 + \frac{F(\mathbf{x} + u_{h_i}\mathbf{e}_{h_i}) - F(\mathbf{x})}{\varepsilon \cdot F(u_{h_i}\mathbf{e}_{h_i})}\right) = O(\varepsilon^{-2} n^2) \ ,$$

where the equality holds since the submodulrity and non-negativity of f imply $F(\mathbf{x} + u_{h_i}\mathbf{e}_{h_i}) - F(\mathbf{x}) \leq F(u_{h_i}\mathbf{e}_{h_i})$.

Algorithm 4: COORDINATE-ASCENT++ (ε)

for *every pair of distinct coordinates* $h_1, h_2 \in [n]$ **do**
  Let $\mathbf{x}^{(0)} \leftarrow \mathbf{0}$.
  for *every* $v_1 \in \mathcal{J}(\mathbf{x}^{(0)}, h_1)$ **do**
    Let y_1 be the value returned by the algorithm guaranteed by Proposition 5.1 given $\mathbf{x}^{(0)}$ as the input vector, the coordinate h_1 and the target value v_1.
    Set $\mathbf{x}^{(1)} \leftarrow \mathbf{x}^{(0)} + y_1\mathbf{e}_{h_1}$.
    for *every* $v_2 \in \mathcal{J}(\mathbf{x}^{(1)}, h_2)$ **do**
      Let y_2 be the value returned by the algorithm guaranteed by Proposition 5.1 given $\mathbf{x}^{(1)}$ as the input vector, the coordinate h_2 and the target value v_2.
      Update $\mathbf{x} \leftarrow \mathbf{x}^{(1)} + y_2\mathbf{e}_{h_2}$.
      Execute Algorithm 1 on the instance obtained by removing the coordinates h_1 and h_2, replacing the objective function with $F'(\mathbf{x}') = F(\mathbf{x}' + \mathbf{x}) - F(\mathbf{x})$ and decreasing B by $\|\mathbf{x}\|_1$. Let $\mathbf{x}_{\text{CA}}$ be the output of Algorithm 1.
      Mark $\mathbf{x} + \mathbf{x}_{\text{CA}}$ as a candidate solution.
Return the solution maximizing F among all the solutions marked above as candidate solutions.

The next theorem summarizes the result we have proved in this section.

Theorem 5.6. *For every* $\varepsilon \in (0, 1)$*, Algorithm 4 is an algorithm for our problem which produces a solution of value at least* $(1-1/e-4\varepsilon)\cdot F(\mathsf{opt})-\varepsilon(B+2)L$*. It has* $O(n^3/\varepsilon^3)$ *iterations, each running in* $O(n\sqrt{B/\varepsilon} + n \log n)$ *time, which yields a time complexity of* $O(n^4\sqrt{B}/\varepsilon^{2.5} + n^4 \log n/\varepsilon^3)$*.*

In all the loops of Algorithm 4, the iterations are independent, and thus, can be done in parallel instead of sequentially. Thus, the parallel time required for Algorithm 4 is equal to the time complexity of Algorithm 3, which by Observation 5.2 is only $O(n^2\sqrt{B}/\varepsilon^{1.5} + n^2 \log n/\varepsilon)$.

6 Conclusion

In this paper, we provided the first constant factor approximation guarantees for the problem of maximizing a monotone continuous submodular function subject to a linear constraint. Crucially, our results did not rely on DR-submodularity.

A natural open problem is to extend our algorithms to more involved kinds constraints. To understand the difficult of such an extension, let us recall that our algorithms resemble algorithms for maximizing a submodular *set* function subject to a knapsack constraint even though all the coordinates have a coefficient of 1 in the constraint, which corresponds to a simpler cardinality constraint. Intuitively, this is the case because our algorithms treat monotone non-DR submodular continuous function as if they were monotone submodular *set* functions over an infinite ground set consisting of pairs (i, v), where i is a coordinate and v is the value assigned to this coordinate. Unfortunately, this intuitive reduction from continuous to set functions converts cardinality constraints into knapsack constraints, forcing us to employ knapsack techniques even when handling cardinality-like constraints. Similarly, other kinds of constraints will also become more involved if the above intuitive reduction is applied to them, often leading to constraints that are difficult to handle.

Broader Impact

In this paper, we provided the first constant factor approximation guarantees for a large class of non-convex functions with combinatorial structure. Since it is a theoretical result in nature, a broader impact discussion is not applicable.

Acknowledgments and Disclosure of Funding

We would like to thank the anonymous referees for pointing out the work of Soma and Yoshida [2018] to us.

The work of Moran Feldman was supported in part by ISF grants no. 1357/16 and 459/20. Amin Karbasi is partially supported by NSF (IIS-1845032), ONR (N00014-19-1-2406), AFOSR (FA9550-18-1-0160), and TATA Sons Private Limited.

References

Francis R. Bach. Submodular Functions: from Discrete to Continous Domains. *CoRR*, abs/1511.00394, 2015.

Eric Balkanski, Baharan Mirzasoleiman, and Yaron Singer. Learning Sparse Combinatorial Representations via Two-stage Submodular Maximization. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2207–2216, 2016.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An optimal approximation for submodular maximization under a matroid constraint in the adaptive complexity model. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

An Bian, Kfir Yehuda Levy, Andreas Krause, and Joachim M. Buhmann. Non-monotone continuous dr-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 486–496, 2017a.

Andrew An Bian, Baharan Mirzasoleiman, Joachim M. Buhmann, and Andreas Krause. Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017b.

Niv Buchbinder and Moran Feldman. Submodular functions maximization problems., 2018.

Niv Buchbinder and Moran Feldman. Constrained submodular maximization via a non-symmetric technique. *Math. Oper. Res.*, 44(3):988–1005, 2019.

Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrak. Maximizing a submodular set function subject to a matroid constraint. *SIAM Journal on Computing*, 2011a.

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

L. Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K. Vishnoi. How to be Fair and Diverse? *CoRR*, abs/1610.07183, 2016.

Chandra Chekuri and Kent Quanrud. Parallelizing greedy for submodular set function maximization in matroids and beyond. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*, 2011.

Chandra Sekhar Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 2014.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, 2018a.

Lin Chen, Hamed Hassani, and Amin Karbasi. Online Continuous Submodular Maximization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1896–1905, 2018b.

Lin Chen, Moran Feldman, and Amin Karbasi. Unconstrained submodular maximization with constant adaptive complexity. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 102–113, 2019.

Reuven Cohen and Liran Katzir. The generalized maximum coverage problem. *Inf. Process. Lett.*, 108(1):15–22, 2008. doi: 10.1016/j.ipl.2008.03.017. URL `https://doi.org/10.1016/j.ipl.2008.03.017`.

Abhimanyu Das and David Kempe. Submodular meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection. In *International Conference on Machine Learning*, pages 1057–1064, 2011.

Nikhil R Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 137–144. ACM, 2012.

Ethan R. Elenberg, Alexandros G. Dimakis, Moran Feldman, and Amin Karbasi. Streaming Weak Submodularity: Interpreting Neural Networks on the Fly. In *Advances in Neural Information Processing Systems*, 2017.

Alina Ene, Huy L Nguyen, and Adrian Vladu. Submodular maximization with matroid and packing constraints in parallel. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

Shaun Fallat, Steffen Lauritzen, Kayvan Sadeghi, Caroline Uhler, Nanny Wermuth, and Piotr Zwiernik. Total positivity in Markov structures. *The Annals of Statistics*, 45(3):1152–1184, 2017.

Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Foundations of Computer Science (FOCS)*, pages 570–579. IEEE, 2011.

Satoru Fujishige. *Submodular functions and optimization*, volume 58. Annals of Discrete. Mathematics, North Holland, Amsterdam, 2nd edition, 1991. ISBN 0-444-88556-0.

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Near-optimal map inference for determinantal point processes. In *Advances in Neural Information Processing Systems*, pages 2735–2743, 2012.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Daniel Golovin, Andreas Krause, and Matthew J. Streeter. Online Submodular Maximization under a Matroid Constraint with Application to Learning Assignments. *CoRR*, abs/1407.1082, 2014.

Andrew Guillory and Jeff Bilmes. Interactive Submodular Set Cover. In *International Conference on Machine Learning*, pages 415–422. Omnipress, 2010.

Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. In *International Conference on Machine Learning*, 2019.

Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Zebang Shen. Stochastic conditional gradient++: (non-)convex minimization and continuous submodular maximization, 2019.

S. Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient Methods for Submodular Maximization. In *Advances in Neural Information Processing Systems*, pages 5843–5853, 2017.

Amin Karbasi, Hamed Hassani, Aryan Mokhtari, and Zebang Shen. Stochastic continuous greedy ++: When upper and lower bounds match. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13066–13076, 2019.

Samuel Karlin and Yosef Rinott. Classes of orderings of measures and related correlation inequalities. i. multivariate totally positive distributions. *Journal of Multivariate Analysis*, 10(4):467–498, 1980.

Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *International conference on machine learning*, pages 2544–2553, 2018.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

Qi Lei, Lingfei Wu, Pin-Yu Chen, Alexandros G Dimakis, Inderjit S Dhillon, and Michael Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *arXiv preprint arXiv:1812.00151*, 2018.

Hui Lin and Jeff A. Bilmes. Learning Mixtures of Submodular Shells with Application to Document Summarization. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 479–490, 2012.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 2007.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed Submodular Maximization: Identifying Representative Elements in Massive Data. In *Advances in Neural Information Processing Systems*, pages 2049–2057, 2013.

Marko Mitrovic, Ehsan Kazemi, Moran Feldman, Andreas Krause, and Amin Karbasi. Adaptive sequence submodularity. In *Advances in Neural Information Processing Systems*, pages 5353–5364, 2019.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *International Conference on Artificial Intelligence and Statistics*, pages 1886–1895, 2018a.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Decentralized submodular maximization: Bridging discrete and continuous settings. In *International Conference on Machine Learning*, 2018b.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *arXiv preprint arXiv:1804.09554*, 2018c.

Rad Niazadeh, Tim Roughgarden, and Joshua R. Wang. Optimal algorithms for continuous non-monotone submodular and DR-submodular maximization. In *Advances in Neural Information Processing Systems 31*, pages 9617–9627, 2018.

Zeev Nutov, 2020. Personal communication.

Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5382–5390, 2017.

Mehraveh Salehi, Amin Karbasi, Dustin Scheinost, and R Todd Constable. A submodular approach to create individualized parcellations of the human brain. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 478–485. Springer, 2017.

Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *International Conference on Machine Learning*, 2014.

Tasuku Soma and Yuichi Yoshida. A generalization of submodular cover via the diminishing return property on the integer lattice. In *Advances in Neural Information Processing Systems*, pages 847–855, 2015.

Tasuku Soma and Yuichi Yoshida. Maximizing monotone submodular functions over the integer lattice. *Math. Program.*, 172(1-2):539–563, 2018. doi: 10.1007/s10107-018-1324-y. URL `https://doi.org/10.1007/s10107-018-1324-y`.

Matthew Staib and Stefanie Jegelka. Robust Budget Allocation via Continuous Submodular Functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3230–3240, 2017.

Ehsan Tohidi, Rouhollah Amiri, Mario Coutino, David Gesbert, Geert Leus, and Amin Karbasi. Submodularity in action: From machine learning to signal processing applications. *IEEE Signal Processing Magazine*, 37(5):120–133, 2020.

Sebastian Tschiatschek, Adish Singla, and Andreas Krause. Selecting sequences of items via submodular maximization. In *AAAI Conference on Artificial Intelligence*, pages 2667–2673, 2017.

Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 1982a.

Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 1982b.

Jiahao Xie, Chao Zhang, Zebang Shen, Chao Mi, and Hui Qian. Decentralized gradient tracking for continuous dr-submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, 2019.

Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In *Advances in Neural Information Processing Systems*, 2019.

Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, 2020.