Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains

Yatao Bian
ETH Zurich
ybian@inf.ethz.ch

Baharan Mirzasoleiman
ETH Zurich
baharanm@inf.ethz.ch

Joachim M. Buhmann
ETH Zurich
jbuhmann@inf.ethz.ch

Andreas Krause
ETH Zurich
krausea@ethz.ch

Abstract

Submodular continuous functions are a category of (generally) non-convex/non-concave functions with a wide spectrum of applications. We characterize these functions and demonstrate that they can be maximized efficiently with approximation guarantees. Specifically, I) for monotone submodular continuous functions with an additional diminishing returns property, we propose a `Frank-Wolfe` style algorithm with $(1 - 1/e)$-approximation, and sub-linear convergence rate; II) for general non-monotone submodular continuous functions, we propose a `DoubleGreedy` algorithm with $1/3$-approximation. Submodular continuous functions naturally find applications in various real-world settings, including influence and revenue maximization with continuous assignments, sensor energy management, multi-resolution data summarization, facility location, etc. Experimental results show that the proposed algorithms efficiently generate superior solutions in terms of empirical objectives compared to baseline algorithms.

1 Introduction

Non-convex optimization delineates the new frontier in machine learning, arising in numerous learning tasks from training deep neural networks to latent variable models. Understanding, which classes of objectives can be tractably optimized, remains a central challenge both for empirical and expected objectives. In this paper, we investigate a class of generally non-convex and non-concave functions – *submodular continuous functions*, and derive algorithms for approximately optimizing them with strong approximation guarantees. Submodular objectives with separable regularizers for complexity control fall into this class and enable us to validate models with a submodular structure.

Submodularity is a structural property usually associated with *set functions*, with important implications for optimization. Optimizing submodular set functions has found numerous applications in machine learning, including variable selection [Krause and Guestrin, 2005], dictionary learning [Krause and Cevher, 2010, Das and Kempe, 2011], sparsity inducing regularizers [Bach, 2010], summarization [Lin and Bilmes, 2011, Mirzasoleiman et al., 2013] and variational inference [Djolonga and Krause, 2014]. Submodular set functions can be efficiently minimized [Iwata et al., 2001], and there are strong guarantees for approximate maximization [Nemhauser et al., 1978, Krause and Golovin, 2012].

Even though submodularity is most widely considered in the discrete realm, the notion can be generalized to arbitrary lattices [Fujishige, 2005]. Recently, Bach [2015] showed how results from *submodular set function minimization* can be lifted to the continuous domain. In this paper, we further pursue this line of investigation, and demonstrate that results from *submodular set function maxi-*

mization can be generalized as well. Note that the underlying concepts associated with submodular function minimization and maximization are quite distinct, and both require different algorithmic treatment and analysis techniques.

As motivation for our inquiry, we will illustrate how submodular continuous maximization captures various applications, ranging from multi-resolution summarization, to influence and revenue maximization, to sensor energy management, and non-convex/non-concave quadratic programming. We then present two algorithms for maximizing submodular continuous functions with guarantees. The first approach, based on the Frank-Wolfe algorithm [Frank and Wolfe, 1956], applies to *monotone* functions with an additional diminishing returns property. It is also inspired by the continuous greedy algorithm from submodular set functions maximization [Vondrák, 2008], and provides a $(1 - 1/e)$ approximation guarantee under a variety of constraints. We also provide a second, coordinate-ascent-style algorithm which applies to arbitrary submodular continuous functions and provides a $1/3$ approximation guarantee. This algorithm is based on the double-greedy algorithm from submodular set functions [Buchbinder et al., 2012]. Lastly, we experimentally demonstrate the effectiveness of our algorithms on several problem instances.

2 Background and related work

Submodularity is often viewed as a discrete analogue of convexity, and it provides computationally effective structure so that many discrete problems with this property are efficiently solvable or approximable. Of particular interest is $(1 - 1/e)$-approximation for maximizing a monotone submodular function subject to a cardinality, a matroid, or a knapsack constraint [Nemhauser et al., 1978, Vondrák, 2008, Sviridenko, 2004]. Another result relevant to this work is unconstrained maximization of (generally) non-monotone submodular functions, for which the deterministic double greedy algorithm of Buchbinder et al. [2012] provides a 1/3 approximation guarantee.

Although most commonly associated with set functions, in many practical scenarios, it is natural to consider generalizations of submodular set functions, including *bisubmodular* functions, k-*submodular* functions, *tree-submodular* functions, *adaptive submodular* functions, as well as submodular functions defined over integer lattices and continuous domains. Golovin and Krause [2011] introduce the notion of adaptive submodularity to generalize submodular set functions to adaptive policies. Kolmogorov [2011] studies tree-submodular functions and presents a polynomial algorithm for minimizing them. For distributive lattices, it is well-known that the combinatorial polynomial-time algorithms for minimizing a submodular set function can be adopted to minimize a submodular function over a bounded integer lattice [Fujishige, 2005]. Recently, maximizing a submodular function over integer lattices has attracted considerable attention. In particular, Soma et al. [2014] develop a $(1-1/e)$-approximation algorithm for maximizing a monotone submodular lattice function under a knapsack constraint. For non-monotone submodular functions over the bounded integer lattice, Gottschalk and Peis [2015] provide a 1/3-approximation. Approximation algorithms for maximizing bisubmodular functions and k-submodular functions have also been proposed by Singh et al. [2012], Ward and Zivny [2014].

Maximizing a submodular continuous function over a knapsack polytope is first considered by Wolsey [1982]. Submodular set functions can be associated with various continuous extensions. For example, Vondrák [2008] studies the multilinear extension of submodular set functions and uses it to improve the approximation guarantee for maximizing a submodular set function. Very recently, Bach [2015] considers the minimization of a submodular continuous function, and proves that efficient techniques from convex optimization may be used for minimization.

Optimizing non-convex continuous functions has received renewed interest in the last decades, we only cover representatives of the related literature here. Recently, tensor methods have been used for various non-convex tasks, e.g., learning latent variable models [Anandkumar et al., 2014] and training neural networks [Janzamin et al., 2015]. A certain part of the work on non-convex optimization [Sra, 2012, Li and Lin, 2015, Reddi et al., 2016a,b, Allen-Zhu and Hazan, 2016] mainly focus on obtaining local convergence by assuming smoothness of the objectives. With extra assumptions, certain global convergence results can be obtained. For example, for functions with Lipschitz continuous Hessians, the regularized Newton scheme of Nesterov and Polyak [2006] achieves global convergence results for functions with an additional star-convexity property or with an additional gradient-dominance property [Polyak, 1963]. Hazan et al. [2015] introduce the family of

Condition	Convex function $g(\cdot), \lambda \in [0,1]$	Submodular continuous function $f(\cdot)$
0^{th} order	$\lambda g(x) + (1-\lambda)g(y) \geq g(\lambda x + (1-\lambda)y)$	$f(x) + f(y) \geq f(x \vee y) + f(x \wedge y)$
1^{st} order	$g(y) - g(x) \geq \langle \nabla g(x), y - x \rangle$	`support DR` (this work)
2^{nd} order	$\nabla^2 g(x) \succeq 0$ (positive semi-definite)	$\frac{\partial^2 f}{\partial x_i \partial x_j} \leq 0, \forall i \neq j$

Table 1: Comparison of properties of convex and submodular continuous functions

σ-nice functions and propose a graduated optimization-based algorithm, that provably converges to a global optimum for this family of (generally) non-convex functions. However, it is typically difficult to verify whether these assumptions capture objective functions encountered in practice.

To the best of our knowledge, this work addresses the general problem of monotone and non-monotone submodular maximization over continuous domains for the first time, by proposing efficient algorithms with strong approximation guarantees. We further show that this submodularity property is sufficiently prevalent to capture objectives arising in various real-world problems.

3 Properties of submodular continuous functions

Submodular continuous functions are defined on subsets of $\mathbb{R}^n$: $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$, where each $\mathcal{X}_i$ is a compact subset of $\mathbb{R}$ [Topkis, 1978, McCormick, 2005, Bach, 2015]. A function $f : \mathcal{X} \to \mathbb{R}$ is submodular *iff* for all $(x, y) \in \mathcal{X} \times \mathcal{X}$,

$$f(x) + f(y) \geq f(x \vee y) + f(x \wedge y), \quad \texttt{(submodularity)} \tag{1}$$

where $\wedge$ and $\vee$ are the coordinate-wise minimum and maximum operations, respectively. When twice-differentiable, $f(\cdot)$ is submodular iff all off-diagonal entries of the Hessian are non-positive,

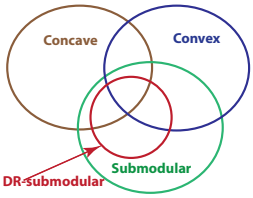


Figure 1: Concavity, convexity, submodularity and DR-submodularity.

$$\forall x \in \mathcal{X}, \quad \frac{\partial^2 f}{\partial x_i \partial x_j} \leq 0, \quad \forall i \neq j. \tag{2}$$

The class of submodular continuous functions contains a subset of both convex and concave functions, and shares some useful properties with them (illustrated in Fig. 1). Examples include submodular and convex functions of the form $\phi_{ij}(x_i - x_j)$ for ϕ_{ij} convex; submodular and concave functions of the form $x \mapsto g(\sum_{i=1}^n \lambda_i x_i)$ for g concave and λ_i non-negative (see Sec. 6 for example applications). Lastly, indefinite quadratic functions of the form $f(x) = \frac{1}{2} x^T H x + h^T x$ with all off-diagonal entries of H non-positive are examples of submodular but non-convex/non-concave functions. Continuous submodularity is preserved under various operations, e.g., the sum of two submodular continuous functions is submodular, a submodular continuous function multiplied by a positive scalar is still submodular.

Interestingly, characterizations of submodular continuous functions are in correspondence to those of convex functions. This relation is summarized in Table 1. In the remainder of this section, we introduce some useful properties of submodular continuous functions. We begin by generalizing the diminishing returns property to continuous functions.

Definition 3.1 (`support DR`)**.** *A function $f(\cdot)$ has the support diminishing returns property if $\forall a \leq b \in \mathbb{R}_+^E$, $\forall i \in E \backslash supp^+(b), \forall k \geq l \in \mathbb{R}_+$, the following inequality is satisfied,*

$$f(k\chi_i \vee a) - f(l\chi_i \vee a) \geq f(k\chi_i \vee b) - f(l\chi_i \vee b), \quad \textit{(Formulation I)} \tag{3}$$

or equivalently, $\forall a \leq b \in \mathbb{R}_+^E$, $\forall i \in E \backslash supp^+(b-a) = \{i \in E \mid a(i) = b(i)\}, \forall k \in \mathbb{R}_+$, it holds,

$$f(k\chi_i \vee a) - f(a) \geq f(k\chi_i \vee b) - f(b), \quad \textit{(Formulation II)} \tag{4}$$

where $E := \{e_1, e_2, \cdots, e_n\}$ is the ground set of n elements, χ_i is the characteristic vector for element e_i, $supp^+(x) := \{i \in E \mid x(i) > 0\}$ is the positive support of vector $x \in \mathbb{R}_+^E$.

The following lemma shows that for all set functions, as well as lattice and continuous functions, submodularity is equivalent to the `support DR` property.

Lemma 3.1 (`submodular`) $\Leftrightarrow$ (`support DR`)**.** *A function $f(\cdot)$ is submodular iff it satisfies the support DR property.*

All of the proofs can be found in the Appendix. Lemma 3.1 allows to characterize submodularity using the `support DR` property for all set, lattice and continuous functions. Furthermore, `support DR` can be considered as the first order condition of submodularity: Intuitively, for any $a \leq b \in \mathbb{R}_+^E$, $\forall i \in E \backslash \text{supp}^+(b-a)$, $a(i) = b(i)$, `support DR` (Formulation II) implies that the directional derivative (if it exists) of $f(\cdot)$ at a is no less than the directional derivative of $f(\cdot)$ at b in directions χ_i . Formally, we have $\lim_{k \to a(i)} \frac{f((k-a(i))\chi_i + a) - f(a)}{k - a(i)} = \lim_{k \to b(i)} \frac{f((k-b(i))\chi_i + a) - f(a)}{k - b(i)} \geq \lim_{k \to b(i)} \frac{f((k-b(i))\chi_i + b) - f(b)}{k - b(i)} \Leftrightarrow \nabla_{\chi_i} f(a) \geq \nabla_{\chi_i} f(b)$.

We further use the `support DR` property to generalize the `DR` properties defined by Soma et al. [2014], Soma and Yoshida [2015b,a] for lattice functions to continuous domains.

Definition 3.2 (`DR`). *$f(\cdot)$ satisfies the diminishing returns property if* $\forall a \leq b \in \mathbb{R}_+^E$, $\forall i \in E$,

$$f(a + \chi_i) - f(a) \geq f(b + \chi_i) - f(b)$$

Lemma 3.2 (`submodular` + `coordinate-wise concave`) $\Leftrightarrow$ (`DR`). *A function $f(\cdot)$ satisfies the DR property (DR-submodular) iff $f(\cdot)$ is submodular and coordinate-wise concave, where the `coordinate-wise concave` property is defined as*

$$f(b + \chi_i) - f(b) \geq f(b + 2\chi_i) - f(b + \chi_i) \qquad \forall b \in \mathbb{R}_+^E, \ \forall i \in E$$

or equivalently (if twice differentiable) $\frac{\partial^2 f}{\partial x_i^2} \leq 0, \forall i \in E$.

Lemma 3.2 shows that a twice differentiable function $f(\cdot)$ is DR-submodular iff $\forall x \in \mathcal{X}$, $\frac{\partial^2 f}{\partial x_i \partial x_j} \leq 0, \forall i, j \in E$, which in general does not imply concavity of $f(\cdot)$.

4 Maximizing monotone DR-submodular continuous functions

In this section, we present an algorithm for maximizing a monotone submodular continuous function with the `DR` property. The idea is based on the continuous greedy algorithm of Vondrák [2008] for maximizing the multilinear extension of a submodular set function, and the Frank-Wolfe algorithm [Frank and Wolfe, 1956, Jaggi, 2013] for minimizing a convex function.

Consider a monotone DR-submodular function $f(\cdot)$, and any number of linear constraints resulting in a down-closed polytope $\mathcal{P} = \{0 \leq x \leq \bar{u}, Ax \leq b\}, \forall A_{ij} \geq 0, b_j \geq 0$. In general, the algorithm can deal with any down-closed convex constraints as long as a linear maximization oracle is available. For simplicity, we only consider a down-closed polytope here. The problem setting of maximizing a monotone DR-submodular function over a down-closed polytope captures various real-world applications, e.g., the influence maximization with continuous assignments, sensor energy management, etc. It can be proved that this problem in general is NP-hard.

Proposition 4.1. *The problem of maximizing a monotone DR-submodular continuous function subject to general down-closed polytope constraints is NP-hard. The optimal approximation ratio is $(1 - 1/e)$ (up to low-order terms), unless P = NP.*

Algorithm 1: `Frank-Wolfe` for monotone DR-submodular function maximization

Input: $\max_{x \in \mathcal{P}} f(x)$, stepsize $\gamma \in (0, 1]$

$x^0 \leftarrow 0, t \leftarrow 0$;

while $t \leq 1$ **do**

  find v_m^t s.t. $\langle v_m^t, \nabla f(x^t) \rangle \geq \alpha \max_{v \in \mathcal{P}} \langle v, \nabla f(x^t) \rangle - \frac{1}{2}\delta L$; `//` $L > 0$, $\alpha \in (0, 1]$ `is the mulplicative error level,` $\delta \in [0, \bar{\delta}]$ `is the additive error level`

  $x^{t+\gamma} \leftarrow x^t + \gamma v_m^t, t \leftarrow t + \gamma$;

Return x^1;

We summarize the `Frank-Wolfe` style method in Alg. 1. On a high level, the idea is to start at $x^0 = 0$, and iteratively move toward a point $v \in \mathcal{P}$. In each iteration the algorithm uses the linearization of the objective function as a surrogate, and move towards a maximizer of this surrogate objective. The maximizer, i.e., $v^t = \arg\max_{v \in \mathcal{P}} \langle v, \nabla f(x^t) \rangle$ is used as the update direction in iteration t. Finding such a direction requires solving a linear program at each iteration[1]. At the

[1]It can easily be solved using standard LP solvers, e.g., simplex method, interior-point method, etc.

same time, it eliminates the need for projecting back to the feasible set in each iteration, which is an essential step for methods such as projected gradient descent. Different properties of submodular and convex functions deter us from using adaptive step sizes, which are commonly used in convex optimization. Instead, we utilize the constant step size γ to achieve the worst-case guaranteee. It is worth noting that the `Frank-Wolfe` algorithm can tolerate both multiplicative error α and additive error δ when solving the linear subproblem (Step 3 of Alg. 1). Setting $\alpha = 1$ and $\delta = 0$, we recover the error-free case.

DR-submodular functions are non-convex/non-concave in general. However, the following proposition gives us some intuition about the strategy of the algorithm, by making clear connections between DR-submodularity and concavity.

Proposition 4.2. *A DR-submodular function $f(\cdot)$ is concave along any non-negative direction.*

Proposition 4.2 implies that the univariate *auxiliary* function $g_{x,v}(\xi) := f(x+\xi v), \xi \in \mathbb{R}_+, v \in \mathbb{R}_+^E$ is concave. As a result, the `Frank-Wolfe` algorithm can follow a concave direction at each step, which is the main reason it can provide the approximation guarantee.

To derive the convergence rate, we need assumptions on the non-linearity of $f(\cdot)$ over the domain $\mathcal{P}$, which closely corresponds to a Lipschitz assumption on the derivative of $g_{x,v}(\cdot)$. For a $g_{x,v}(\cdot)$ with Lipschitz continuous derivative in $[0, 1]$ with parameter $L > 0$, we have,

$$-\frac{L}{2}\xi^2 \leq g_{x,v}(\xi) - g_{x,v}(0) - \xi \nabla g_{x,v}(0) = f(x + \xi v) - f(x) - \langle \xi v, \nabla f(x) \rangle, \forall \xi \in [0, 1] \quad (5)$$

Now, to prove the approximation guarantee of $(1 - 1/e^{\alpha})$, we show that at each iteration we close the gap to the optimum solution by a factor of $\alpha(f(x^*) - f(x^t))$.

Lemma 4.3. *$x^1 \in \mathcal{P}$. Assuming x^* to be the optimal solution, one has,*

$$\langle v_m^t, \nabla f(x^t) \rangle \geq \alpha(f(x^*) - f(x^t)) - \frac{1}{2}\delta L, \ \ \forall t \in [0, 1]. \quad (6)$$

Theorem 4.4 (Convergence rate)**.** *For error levels $\alpha \in (0, 1], \delta \in [0, \bar{\delta}]$, step size $\gamma \in (0, 1]$, with $K := 1/\gamma$ iterations, Alg. 1 outputs x^1 s.t.*

$$f(x^1) - (1 - 1/e^{\alpha}) f(x^*) \geq \frac{-L}{2K} + \frac{-L\delta}{2}$$

Theorem 4.4 implies that: 1) when $\gamma \to 0$ ($K \to \infty$), Algorithm 1 will output the solution with the optimal worst-case bound $(1 - 1/e)f(x^*)$ in the error-free case; and 2) `Frank-Wolfe` has a sub-linear convergence rate for monotone DR-submodular maximization.

5 Maximizing non-monotone submodular continuous functions

The problem of maximizing a general, potentially non-monotone, submodular continuous function under box constraints, i.e., $\max_{x \in [0, \bar{u}]} f(x)$, captures various real-world applications, including revenue maximization with continuous assignments, multi-resolution summarization, etc. First of all, it can be proved that the problem is NP-hard.

Proposition 5.1. *The problem of maximizing a generally non-monotone submodular continuous function subject to box constraint is NP-hard. And there is no $(1/2 + \epsilon)$-approximation $\forall \epsilon > 0$, unless RP = NP.*

We now describe our algorithm for maximizing a potentially non-monotone submodular continuous function subject to box constraints. It provides a 1/3-approximation using ideas from the double-greedy algorithm of Buchbinder et al. [2012] and Gottschalk and Peis [2015]. It can be viewed as a procedure performing coordinate-ascent on two solutions.

We view the process as two particles starting from $x^0 = 0$ and $y^0 = \bar{u}$, and following a certain flow continuously toward each other. The pseudo-code is given in Alg. 2. We proceed in n rounds that correspond to some arbitrary order of the coordinates. At iteration k, we consider solving a one-dimensional (1-D) subproblem over coordinate e_k for each particle, and moving the particles based on the calculated local gains toward each other. Formally, for a given coordinate e_k, we solve a 1-D subproblem to find the new value of the solution x along coordinate e_k,

Algorithm 2: `DoubleGreedy` for maximizing non-monotone submodular continuous functions

Input: $\max_{x\in[0,\bar{u}]} f(x)$, $f(0)+f(\bar{u})\geq 0$

1 $x^0 \leftarrow 0$, $y^0 \leftarrow \bar{u}$;

2 **for** $k = 1 \rightarrow n$ **do**

3 find $\hat{u}_a$ s.t. $f(x^{k-1}\vee \hat{u}_a\chi_{e_k}) \geq \max_{u_a\in[0,\bar{u}_{e_k}]} f(x^{k-1}\vee u_a\chi_{e_k}) - \delta$,
$\delta_a \leftarrow f(x^{k-1}\vee \hat{u}_a\chi_{e_k}) - f(x^{k-1})$; //$\delta \in [0,\bar{\delta}]$ is the additive error level

4 find $\hat{u}_b$ s.t. $f(y^{k-1}\wedge \hat{u}_b\chi_{e_k}) \geq \max_{u_b\in[0,\bar{u}_{e_k}]} f(y^{k-1}\wedge u_b\chi_{e_k}) - \delta$,
$\delta_b \leftarrow f(y^{k-1}\wedge \hat{u}_b\chi_{e_k}) - f(y^{k-1})$;

5 **If** $\delta_a \geq \delta_b$: $x^k \leftarrow x^{k-1}\vee \hat{u}_a\chi_{e_k}$, $y^k \leftarrow y^{k-1}\wedge \hat{u}_a\chi_{e_k}$;

6 **Else**: $y^k \leftarrow y^{k-1}\wedge \hat{u}_b\chi_{e_k}$, $x^k \leftarrow x^{k-1}\vee \hat{u}_b\chi_{e_k}$;

7 Return x^n (or y^n);

i.e., $\hat{u}_a = \arg\max_{u_a} f(x^{k-1}\vee u_a\chi_{e_k}) - f(x^{k-1})$, and calculate its marginal gain δ_a. We then solve another 1-D subproblem to find the new value of the solution y along coordinate e_k, i.e., $\hat{u}_b = \arg\max_{u_b} f(y^{k-1}\wedge u_b\chi_{e_k}) - f(y^{k-1})$, and calculate the second marginal gain δ_b. We then decide by comparing the two marginal gains. If changing $x(e_k)$ to be $\hat{u}_a$ has a larger local benefit, we consider changing *both* $x(e_k)$ and $y(e_k)$ to be $\hat{u}_a$. Otherwise, we change *both* of them to be $\hat{u}_b$. After n iterations the particles should meet at point $x^n = y^n$, which is the solution returned by the algorithm. Note that Alg. 2 can tolerate additive error in solving each 1-D subproblem (Steps 3, 4).

We would like to emphasize that the assumptions required by `DoubleGreedy` to provide the approximation guarantee of 1/3, are submodularity of f, $f(0)+f(\bar{u})\geq 0$ and the (approximate) solvability of the 1-D subproblem at each iteration. For providing the approximation guarantee, the idea is to bound the loss in the objective value from the assumed optimal objective value between every two consecutive steps, which is then used to bound the maximum loss after n iterations.

Theorem 5.2. *Assuming the optimal solution to be OPT, the output of Alg. 2 has function value no less than $\frac{1}{3}f(OPT) - \frac{4n}{3}\delta$, where $\delta \in [0,\bar{\delta}]$ is the additive error level for each 1-D subproblem.*

6 Applications of submodular continuous objective functions

In this section, we discuss several applications with submodular continuous objectives.

Non-convex/non-concave quadratic programming (NQP). NQP of the form $f(x) = \frac{1}{2}x^THx + h^Tx$ under linear constraints has recently been studied by Chen and Burer [2012]. Such programs naturally arise in many applications, including scheduling, inventory theory, and free boundary problems [Skutella, 2001]. A specific class of NQP is the submodular NQP, which has all off-diagonal entries of H to be non-positive. In this work we mainly use submodular NQP as synthetic functions for both monotone DR-submodular maximization and non-monotone submodular maximization.

Optimal budget allocation with continuous assignments. As a special case of influence maximization, the problem of allocating budget among ad sources trying to maximize their influence on the customers can be modeled as a bipartite graph $(S,T;E)$, where S and T are collections of advertising channels and customers, respectively. The edge weights, p_{st}, represent the influence probabilities. The goal is to distribute the budget (e.g. time for a TV advertisement, or space of an inline ad) among the source nodes, and to maximize the expected influence on the potential customers [Soma et al., 2014, Hatano et al., 2015]. The total influence of customer t from all channels can be modeled by a proper monotone DR-submodular function $I_t(x)$, e.g., $I_t(x) = 1 - \prod_{(s,t)\in E}[1-p_{st}]^{x(s)}$ where $x \in \mathbb{R}_+^S$ is the budget assignment among the advertising channels. For a set of k advertisers, let $x_i \in \mathbb{R}_+^S$ to be the budget assignment for advertiser i, let $x := [x_1,\cdots,x_k]$ denote the overall assignments. The overall objective is,

$$g(x) = \sum_{i=1}^{k}\alpha_i f(x_i) \text{ with } f(x_i) := \sum_{t\in T} I_t(x_i),\ 0\leq x_i\leq \bar{u}_i, \forall i = 1,\cdots,k \tag{7}$$

which is monotone DR-submodular. A concrete application is the search marketing advertiser bidding problem, in which vendors bid for the right to appear alongside the results of different search

keywords. Here, $x_i(s)$ is the volume of advertising space allocated to the advertiser i to show his ad alongside query keyword s. The search engine company needs to distribute budgets (ad space) to all vendors to maximize the influence on the customers, and maximize their profit, while respecting various constraints. For example, each vendor has a specified budget limit for advertising, and the ad space associated with each search keyword can not be too large, otherwise the customer would be unhappy with so many ads. All of these constraints can be arranged to be a down-closed polytope $\mathcal{P}$, so the problem of $\max_{x\in\mathcal{P}} g(x)$ can be approximately solved by the `Frank-Wolfe` algorithm. Note that one can flexibly add regularizers in designing $I_t(x_i)$ as long as it is still monotone DR-submodular. There is no restriction to make $I_t(x_i)$ concave. For example, separable regularizer in the form $\sum_s \phi(x_i(s))$ will not change the off-diagonal entries of the Hessian, thus will maintain submodularity. By bounding the second-order derivative of $\phi(x_i(s))$, the DR-submodularity can also be maintained.

Revenue maximization with continuous assignments. In viral marketing, sellers choose a small subset of buyers to give them some product for free, to trigger a cascade of further adoptions through "word-of-mouth" effects, in order to maximize the total revenue [Hartline et al., 2008]. For some products (e.g. software), the seller usually gives away the product in the form of a trial, to be used for free for a limited time period. Except for deciding whether to choose a user or not, the sellers also need to decide how much the free assignment should be. We call this problem *revenue maximization with continuous assignments*. Assume there are q products and n buyers/users, let $x_i \in \mathbb{R}_+^n$ to be the assignments of product i to the n users, let $x := [x_1, \cdots, x_q]$ denote the assignments for the q products. The revenue can be modelled as $g(x) = \sum_{i=1}^q f(x_i)$ with

$$f(x_i) = \alpha_i \sum_{s:x_i(s)=0} R_s(x_i) + \beta_i \sum_{t:x_i(t)\neq 0} \phi(x_i(t)) + \gamma_i \sum_{t:x_i(t)\neq 0} \bar{R}_t(x_i), \quad 0 \leq x_i \leq \bar{u} \tag{8}$$

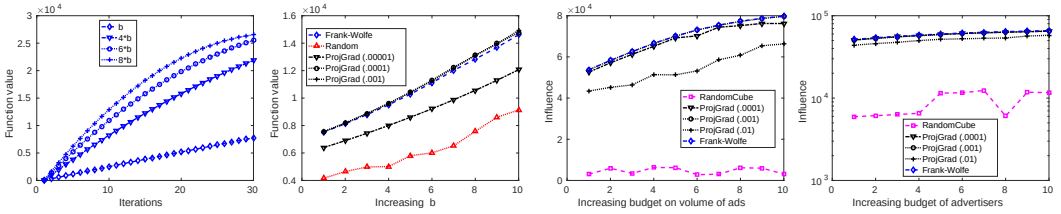
where $x_i(t)$ is the assignment of product i to user t for free, e.g., the amount of free trial time or the amount of the product itself. $R_s(x_i)$ models revenue gain from user s who did not receive the free assignment, it can be some non-negative, non-decreasing submodular function; $\phi(x_i(t))$ models revenue gain from user t who received the free assignment, since the more one user tries the product, the more likely he/she will buy it after the trial period; $\bar{R}_t(x_i)$ models the revenue loss from user t (in the free trial time period the seller cannot get profits), it can be some non-positive, non-increasing submodular function. It is reasonable to set $\beta_i \sum_{t:x_i(t)\neq 0} \phi(x_i(t)) + \gamma_i \sum_{t:x_i(t)\neq 0} \bar{R}_t(x_i) \geq 0, \forall 0 \leq x_i \leq \bar{u}$ because giving users reasonable amount of free trials will result in revenue gain, and, thereby, ensures $f(x_i)$ to be non-negative. With $\beta = \gamma = 0$, it recovers the classical model of Hartline et al. [2008] (See Appendix D for more details on the objective). For products with continuous assignments, usually the cost of the product does not increase with its amount, e.g., the product as a software, so we only have the box constraint on each assignment. The objective in Eq. 8 is generally *non-concave/non-convex*, and non-monotone submodular, thus can be efficiently approximately maximized by the proposed `DoubleGreedy` algorithm.

Lemma 6.1. *If $R_s(x_i)$ is non-decreasing submodular and $\bar{R}_t(x_i)$ is non-increasing submodular, then $f(x_i)$ in Eq. 8 is submodular.*

Sensor energy management. For cost-sensitive outbreak detection in sensor networks [Leskovec et al., 2007], one needs to place sensors in a subset of locations selected from all the possible locations S, to quickly detect a set of contamination/events E, while respecting the cost constraints of the sensors. For each location $s \in S$ and each event $e \in E$, a value $t(s, e)$ is provided as the time it takes for the placed sensor in s to detect event e. Soma and Yoshida [2015a] considered the sensors with discrete energy levels, it is also natural to model the energy levels of sensors to be a continuous variable $x \in \mathbb{R}_+^S$. For a sensor with energy level $x(s)$, the success probability it detects the event is $1 - (1-p)^{x(s)}$, which models that by spending one unit of energy one has an extra chance of detecting the event with probability p. In this model, except for deciding whether to place a sensor or not, one also needs to decide the optimal energy levels. Let $t_\infty = \max_{s\in S, e\in E} t(s, e)$, let s_e be the first sensor that detects event e (s_e is a random variable). One can define the objective as the expected detection time that could be *saved*,

$$f(x) = \mathbb{E}_{e\in E} \mathbb{E}_{s_e}[t_\infty - t(s_e, e)]. \tag{9}$$

Maximizing $f(x)$ w.r.t. the cost constraints pursues the goal to find the optimal energy levels of the sensors, to maximize the expected detection time that could be saved. It can be proved that $f(x)$ is monotone DR-submodular.



(a) Monotone NQP (b) Monotone NQP (c) Budget allocation (d) Budget allocation

Figure 2: Monotone experiments: a) `Frank-Wolfe` function value w.r.t. iterations for 4 instances with different b; b) NQP function value returned w.r.t. different b; c) Influence returned w.r.t. different budgets on volume of ads; d) Influence returned w.r.t. different budgets of advertisers;

Multi-resolution summarization. Suppose we have a collection of items, e.g., images $V = \{e_1, \cdots, e_n\}$. Our goal is to extract a representative summary, where representativeness is usually defined w.r.t. a submodular set function $F : 2^V \to \mathbb{R}$. However, instead of returning a single set, our goal is to obtain summaries at multiple levels of detail/resolution. One way to achieve this goal is to assign each item e_i a nonnegative score $x(i)$. Given a user-tunable threshold τ, the resulting summary $S_\tau = \{e_i : x(i) \geq \tau\}$ is the set of items with scores exceeding τ. Thus, instead of solving the discrete problem of selecting a fixed set S, we pursue the goal to optimize over the scores, e.g., to use the following submodular continuous function

$$f(x) = \sum_{i \in V} \sum_{j \in V} \phi(x(j)) s_{i,j} - \sum_{i \in V} \sum_{j \in V} x(i) x(j) s_{i,j}, \tag{10}$$

where $s_{i,j} \geq 0$ is the similarity between items i, j, and $\phi(\cdot)$ is a non-decreasing concave function.

Facility location. The classical discrete facility location problem can be naturally generalized where the scale of a facility is represented as a continuous value in interval $[0, \bar{u}]$. Assume there is a set E of facilities, let $x \in \mathbb{R}_+^E$ be the scale of all facilities. The goal is to decide how large each facility should be in order to optimally serve a set T of customers. For a facility s of scale $x(s)$, let $p_{st}(x(s))$ be the amount of service it can provide to customer $t \in T$, where $p_{st}(x(s))$ is a normalized monotone function ($p_{st}(0) = 0$). Given all of the facilities, assume that each customer chooses the facility with highest service value, then the total service provided to all customers is $f(x) = \sum_{t \in T} \max_{s \in E} p_{st}(x(s))$. It is not difficult to prove that f is monotone and submodular.

Other applications. Many discrete submodular problems can be naturally generalized to the continuous setting with submodular continuous objectives. For example, the maximum coverage problem and the problem of text summarization with submodular objectives [Filatova, 2004, Lin and Bilmes, 2010]. We defer further details to Appendix E.

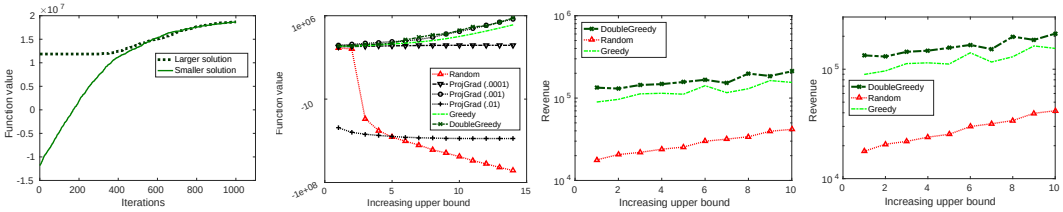
7 Experiments

We compare the performance of our proposed algorithms, `Frank-Wolfe` and `DoubleGreedy`, with the following baselines: a) `Random`: uniformly sample k_s solutions from the constraint set using the hit-and-run sampler [Kroese et al., 2013], and select the best one. For the constraint set as a very high-dimensional polytope ($\{x|Ax \leq b\}$), this approach is computationally very expensive; To accelerate sampling from a high-dimensional polytope, we also use b) `RandomCube`: randomly sample k_s solutions from the hypercube, and decrease their elements until they are inside the polytope; c) `ProjGrad`: projected gradient ascent with an empirically tuned step size; d) `Greedy`: for non-monotone submodular functions, we greedily increase each coordinate, as long as it remains feasible. This approach is similar to the coordinate ascent method.

7.1 Monotone maximization

The performance of the methods are evaluated for the following tasks:

Monotone DR-submodular NQP. We randomly generated monotone DR-submodular NQP functions of the form $f(x) = \frac{1}{2} x^T H x + h^T x$, where $H \in \mathbb{R}^{n \times n}$ is a random matrix with non-positive entries. In our experiments, we considered $n = 100$. We further generated a set of $m = 50$ linear



(a) `DoubleGreedy` (b) Non-monotone NQP (c) $\alpha = 10, \beta = \gamma = 1$ (d) $\gamma = 10, \alpha = \beta = 1$

Figure 3: Non-monotone experiments. a) Function values of the two intermediate solutions of `DoubleGreedy` in each iteration; b) Non-monotone NQP function value w.r.t. different upper bounds; c, d) Revenue returned with different upper bounds $\bar{u}$ on the Youtube dataset.

constraints to construct the polytope $\mathcal{P} = \{Ax \leq b, 0 \leq x \leq \bar{u}\}$. To make the gradient non-negative, we set $h = -H^T\bar{u}$. We empirically tuned step size α_p for `ProjGrad` and ran it for n iterations. Fig. 2a shows the utility obtained by the `Frank-Wolfe` algorithm v.s. the number of iterations for 4 function instances with different values of b. Fig. 2b shows the utility obtained by different algorithms with increasing values of b. For the `ProjGrad` method, we plotted the curves for three different values of α_p. Note that the performance of the `Random` method is reasonable, which indicates that this is not a difficult problem instance. One can observe that the performance of `ProjGrad` fluctuates with different step sizes, and with the best-tuned step size, `ProjGrad` performs close to the `Frank-Wolfe` algorithm.

Optimal budget allocation. As our real-world experiments, we used the Yahoo! Search Marketing Advertiser Bidding Data[2], which consists of 1,000 advertisers, 10,475 customers and 52,567 edges. We considered the frequency of (keyword, customer) pairs to estimate the influence probabilities, and used the average of the bidding prices to put a limit on the budget of each advertiser. Since the `Random` sampling was too slow, we compared with the `RandomCube` method. Fig. 2c and 2d show the value of the utility function (influence) when varying the budget on volume of ads and on budget of advertisers, respectively. Again, we observe that `Frank-Wolfe` outperforms the other baselines, and the performance of the `ProjGrad` highly depends on the choice of the step size.

7.2 Non-monotone maximization

Non-monotone NQP. We randomly generated non-monotone submodular NQP functions of the form $f(x) = \frac{1}{2}x^T Hx + h^T x + c$, where $H \in \mathbb{R}^{n \times n}$ is a n by n matrix with non-positive off-diagonal entries. We considered a matrix for which 50% of the eigenvalues are positive and the other 50% are negative. We set n to be 1,000. We then randomly select a value for c such that $f(0) + f(\bar{u}) \geq 0$. `ProjGrad` was executed for n iterations, with empirically tuned step sizes. For the `Random` method we set $k_s = 1,000$. Fig. 3a shows the utility of the two intermediate solutions maintained by `DoubleGreedy`. One can observe that they both increase in each iteration. Fig. 3b shows the values of the utility function for varying upper bound $\bar{u}$. We can see that `DoubleGreedy` outperforms `Greedy` and `ProjGrad`, while `ProjGrad`'s performance depends on the choice of the step size. With carefully hand-tuned step size, its performance is comparable to `DoubleGreedy`.

Revenue maximization. Without loss of generality, we considered maximizing the revenue from selling one product. Since the objective is non-smooth and non-monotone (see Eq. 8 for the objective), we did not compare the performance of `DoubleGreedy` with `ProjGrad`. We performed our experiments on the top 500 largest communities of the YouTube social network[3] consisting of 39,841 nodes and 224,235 edges. The edge weights were assigned according to a uniform distribution $U(0, 1)$. See Fig. 3c, 3d for an illustration of revenue for varying upper bound ($\bar{u}$) and different combinations of the parameters (α, β, γ) in the model (Eq. 8). For different values of the upper bound, the `DoubleGreedy` algorithm outperforms the other baselines, while `Greedy` maintaining only one intermediate solution obtained a lower utility.

[2] `https://webscope.sandbox.yahoo.com/catalog.php?datatype=a`

[3] `http://snap.stanford.edu/data/com-Youtube.html`

8 Conclusion

In this paper, we characterized submodular continuous functions, and proposed two approximation algorithms to efficiently maximize them. In particular, for maximizing monotone DR-submodular continuous functions subject to general down-closed polytope constraints, we proposed a $(1-1/e)$-approximation algorithm, and for maximizing non-monotone submodular continuous functions subject to a box constraint, we proposed a 1/3-approximation algorithm. We demonstrate the effectiveness of our algorithms through a set of experiments on real-world applications, including budget allocation, revenue maximization, and non-convex/non-concave quadratic programming, and show that our proposed methods outperform the baselines in the experiments. This work demonstrates that the submodularity structure can ensure guaranteed optimization in the continuous setting, thus allowing to model problems with this category of (generally) non-convex/non-concave objectives.

Acknowledgments

The authors would like to thank Martin Jaggi for valuable discussions. This research was partially supported by ERC StG 307036.

References

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

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. *arXiv preprint arXiv:1603.05643*, 2016.

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *JMLR*, 15(1):2773–2832, 2014.

Francis Bach. Submodular functions: from discrete to continous domains. *arXiv:1511.00394*, 2015.

Francis R Bach. Structured sparsity-inducing norms through submodular functions. In *NIPS*, pages 118–126, 2010.

Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *FOCS*, pages 649–658. IEEE, 2012.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Integer programming and combinatorial optimization*, pages 182–196. Springer, 2007.

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

Jieqiu Chen and Samuel Burer. Globally solving nonconvex quadratic programming problems via completely positive programming. *Math. Program. Comput.*, 4(1):33–52, 2012.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

Josip Djolonga and Andreas Krause. From map to marginals: Variational inference in bayesian submodular models. In *NIPS*, pages 244–252, 2014.

Shahar Dobzinski and Jan Vondrák. From query complexity to computational complexity. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1107–1116. ACM, 2012.

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

Uriel Feige, Vahab S Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

Elena Filatova. Event-based extractive summarization. In *Proceedings of ACL Workshop on Summarization*, pages 104–111, 2004.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Satoru Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

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

Corinna Gottschalk and Britta Peis. Submodular function maximization on the bounded integer lattice. In *Approximation and Online Algorithms*, pages 133–144. Springer, 2015.

Jason Hartline, Vahab Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 189–198. ACM, 2008.

Daisuke Hatano, Takuro Fukunaga, Takanori Maehara, and Ken-ichi Kawarabayashi. Lagrangian decomposition algorithm for allocating marketing channels. In *AAAI*, pages 1144–1150, 2015.

Elad Hazan, Kfir Y Levy, and Shai Shalev-Swartz. On graduated optimization for stochastic non-convex problems. *arXiv preprint arXiv:1503.03712*, 2015.

Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML 2013*, pages 427–435, 2013.

Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *CoRR abs/1506.08473*, 2015.

Vladimir Kolmogorov. Submodularity on a tree: Unifying $l^\natural$-convex and bisubmodular functions. In *Mathematical Foundations of Computer Science*, pages 400–411. Springer, 2011.

Andreas Krause and Volkan Cevher. Submodular dictionary selection for sparse representation. In *ICML*, pages 567–574, 2010.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.

Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *UAI*, pages 324–331, 2005.

Dirk P Kroese, Thomas Taimre, and Zdravko I Botev. *Handbook of Monte Carlo Methods*, volume 706. John Wiley & Sons, 2013.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. In *NIPS*, pages 379–387, 2015.

Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920, 2010.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *HLT*, 2011.

S Thomas McCormick. Submodular function minimization. *Handbooks in operations research and management science*, 12:321–391, 2005.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *NIPS*, pages 2049–2057, 2013.

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

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

BT Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alex Smola. Fast stochastic methods for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1605.06900*, 2016a.

Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alex Smola. Fast incremental method for nonconvex optimization. *arXiv preprint arXiv:1603.06159*, 2016b.

Ajit P Singh, Andrew Guillory, and Jeff Bilmes. On bisubmodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 1055–1063, 2012.

Martin Skutella. Convex quadratic and semidefinite programming relaxations in scheduling. *J. ACM*, 2001.

Tasuku Soma and Yuichi Yoshida. A generalization of submodular cover via the diminishing return property on the integer lattice. In *NIPS*, pages 847–855, 2015a.

Tasuku Soma and Yuichi Yoshida. Maximizing submodular functions with the diminishing return property over the integer lattice. *arXiv preprint arXiv:1503.01218*, 2015b.

Tasuku Soma, Naonori Kakimura, Kazuhiro Inaba, and Ken-ichi Kawarabayashi. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In *ICML*, pages 351–359, 2014.

Suvrit Sra. Scalable nonconvex inexact proximal splitting. In *NIPS*, pages 530–538, 2012.

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

Donald M Topkis. Minimizing a submodular function on a lattice. *Operations research*, 26(2):305–321, 1978.

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

Justin Ward and Stanislav Zivny. Maximizing bisubmodular and k-submodular functions. In *SIAM*, 2014.

Laurence A. Wolsey. Maximizing real-valued submodular functions: Primal and dual heuristics for location problems. *Math. Oper. Res.*, 7(3):410–425, 1982.

Supplementary Material of the Paper "Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains"

A Proof of properties of submodular continuous functions

For convenience of notation, we use the uppercase letter to denote the positive support of the corresponding lowercase letter, e.g., $B := \text{supp}^+(b)$. Also let $b(A) := \sum_{i\in A} b(i)\chi_i$ to be the vector only containing the entries of b falling inside the set A.

A.1 Equivalence of two formulations of the `support DR` property

First of all, let us prove that `support DR` has the following two equivalent formulations:

Corollary A.1 (Equivalence of two formulations of `support DR`)**.** *The `Formulation I` (Eq. 3) and `Formulation II` (Eq. 4) of the `support DR` property are equivalent.*

Proof. 1) `Formulation I` $\Rightarrow$ `Formulation II`

Let $D_1 = \{i|a(i) = b(i) = 0\} = E\backslash\text{supp}^+(b)$, $D_2 = \{i|a(i) = b(i) > 0\}$, one can see that $D = D_1 \cup D_2$.

When $i \in D_1$, set $l = 0$ in `Formulation I` one can get $f(k\chi_i \vee a) - f(a) \geq f(k\chi_i \vee b) - f(b)$.

When $i \in D_2$, let $l = a(i) = b(i) > 0$. If $k \leq l$, Eq. 4 holds trivially. If $k > l$, let $\bar{a} = a \wedge 0\chi_i, \bar{b} = b \wedge 0\chi_i$. It is easy to see that $\bar{a} \leq \bar{b}$, and $i \in E\backslash\text{supp}^+(\bar{b})$. Then from `Formulation I`,

$$\begin{aligned} &f(k\chi_i \vee \bar{a}) - f(l\chi_i \vee \bar{a}) = f(k\chi_i \vee a) - f(a) \\ &\geq f(k\chi_i \vee \bar{b}) - f(l\chi_i \vee \bar{b}) = f(k\chi_i \vee b) - f(b) \end{aligned}$$

which proves the `Formulation II`.

2) `Formulation I` $\Leftarrow$ `Formulation II`

$\forall a \leq b, \forall i \in E\backslash\text{supp}^+(b)$, one has $a(i) = b(i) = 0$. $\forall k \geq l \geq 0$, let $\hat{a} = l\chi_i \vee a, \hat{b} = l\chi_i \vee b$, it can be verified that $\hat{a} \leq \hat{b}$ and $i \in E\backslash\text{supp}^+(\hat{b} - \hat{a})$, from `Formulation II`,

$$\begin{aligned} &f(k\chi_i \vee \hat{a}) - f(\hat{a}) = f(k\chi_i \vee a) - f(l\chi_i \vee a) \\ \geq &f(k\chi_i \vee \hat{b}) - f(\hat{b}) = f(k\chi_i \vee b) - f(l\chi_i \vee b) \end{aligned}$$

which proves the `Formulation I`.

□

A.2 Proof of Lemma 3.1

Proof. 1) `submodularity` $\Rightarrow$ `support DR`:

Let us prove the `Formulation I` (Eq. 3) of `support DR`: $\forall a \leq b$, $\forall i \in E\backslash\text{supp}^+(b), \forall k \geq l \in \mathbb{R}_+$, the following inequality holds

$$f(k\chi_i \vee a) - f(l\chi_i \vee a) \geq f(k\chi_i \vee b) - f(l\chi_i \vee b).$$

And f is a submodular function iff $\forall x, y \in \mathbb{R}_+^E$, $f(x) + f(y) \geq f(x \vee y) + f(x \wedge y)$, so $f(y) - f(x \wedge y) \geq f(x \vee y) - f(x)$.

Now $\forall a \leq b \in \mathbb{R}_+^E, \forall i \in E\backslash\text{supp}^+(b), \forall k \geq l \in \mathbb{R}_+$, one can always pick x, y s.t. $x = l\chi_i \vee b$ (note that $b(i) = a(i) = 0$), and $y(X \cap Y\backslash\{i\}) = (x \wedge y)(X \cap Y\backslash\{i\}) = a$, it is easy to see that $a \leq b$ is ensured. Set $y = k\chi_i \vee y(X \cap Y\backslash\{i\})$, and one can see that $k\chi_i \vee a = k\chi_i \vee y(X \cap Y\backslash\{i\}) = y$, $l\chi_i \vee a = l\chi_i \vee ((x \wedge y)(X \cap Y\backslash\{i\})) = x \wedge y$, and $k\chi_i \vee b = x \vee y$. Substituting all the above equalities into $f(y) - f(x \wedge y) \geq f(x \vee y) - f(x)$ one can get $f(k\chi_i \vee a) - f(l\chi_i \vee a) \geq f(k\chi_i \vee b) - f(l\chi_i \vee b)$.

2) `submodularity` $\Leftarrow$ `support DR`:

Let us use `Formulation II` (Eq. 4) of `support DR`.

$\forall x, y \in \mathbb{R}_+^E$, let $D := \{e_1, \cdots, e_d\}$ to be the set of elements for which $y(e) > x(e)$. Now set $a^0 := x \wedge y, b^0 := x$ and $a^i = y(e_i)\chi_{e_i} \vee a^{i-1}, b^i = y(e_i)\chi_{e_i} \vee b^{i-1}$, for $i = 1 \cdots d$. One can verify that $a^i \leq b^i, a^i(e_{i'}) = b^i(e_{i'})$ for all $i' \in D, i = 0 \cdots d$, and that $a^d = y, b^d = x \vee y$.

Applying `Formulation II` of the `support DR` property for $i = 1, \cdots, d$ one can get

$$
\begin{aligned}
&f(y(e_1)\chi_{e_1} \vee a^0) - f(a^0) \geq f(y(e_1)\chi_{e_1} \vee b^0) - f(b^0) \\
&f(y(e_2)\chi_{e_2} \vee a^1) - f(a^1) \geq f(y(e_2)\chi_{e_2} \vee b^1) - f(b^1) \\
&\cdots \\
&f(y(e_d)\chi_{e_d} \vee a^{d-1}) - f(a^{d-1}) \geq f(y(e_d)\chi_{e_d} \vee b^{d-1}) - f(b^{d-1})
\end{aligned}
$$

Taking a sum over all the above d inequalities, one can get

$$
\begin{aligned}
&f(y(e_d)\chi_{e_d} \vee a^{d-1}) - f(a^0) \geq f(y(e_d)\chi_{e_d} \vee b^{d-1}) - f(b^0) \Leftrightarrow \\
&f(y) - f(x \wedge y) \geq f(x \vee y) - f(x) \Leftrightarrow \\
&f(x) + f(y) \geq f(x \vee y) + f(x \wedge y)
\end{aligned}
$$

which proves the submodularity. □

A.3 Proof of Lemma 3.2

Proof. 1) `submodular` + `coordinate-wise concave` $\Rightarrow$ `DR`:

When $i \in E \backslash B$, the result follows from Lemma 3.1 by setting $l = 0$. Thus, one can only consider the situation when $i \in B$. From coordinate-wise concavity we have $f(a + \chi_i) - f(a) \geq f(a + (b(i) - a(i) + 1)\chi_i) - f(a + (b(i) - a(i))\chi_i) = f(a \vee (b(i) + 1)\chi_i) - f(a \vee b(i)\chi_i)$. Therefore, it suffices to show that

$$
f(a \vee (b(i) + 1)\chi_i) - f(a \vee b(i)\chi_i) \geq f(b + \chi_i) - f(b) = f(b \vee (b(i) + 1)\chi_i) - f(b) \quad (11)
$$

Let $x := b, y := a \vee (b(i)+1)\chi_i$, so $y \wedge x = a \vee b(i)\chi_i, x \vee y = b \vee (b(i)+1)\chi_i$. From submodularity, one can see that inequality 11 holds.

2) `submodular` + `coordinate-wise concave` $\Leftarrow$ `DR`:

To prove *submodularity*, one just needs to prove the `support DR` since it is equivalent to submodularity. From `DR` property, one can easily prove that $\forall a \leq b, \forall i \in E, \forall \bar{k} \in \mathbb{R}_+, f(\bar{k}\chi_i + a) - f(a) \geq f(\bar{k}\chi_i + b) - f(b)$.

Now $\forall i \in D := E \backslash \text{supp}^+(b - a) = \{i \in E | a(i) = b(i)\}$, if $k < a(i) = b(i)$, then $f(k\chi_i \vee a) - f(a) \geq f(k\chi_i \vee b) - f(b)$ holds trivially. If $k \geq a(i) = b(i)$, set $\bar{k} = (k - a(i)) \in \mathbb{R}_+$, one gets $f((k-a(i))\chi_i + a) - f(a) \geq f((k-b(i))\chi_i + b) - f(b) \Leftrightarrow f(k\chi_i \vee a) - f(a) \geq f(k\chi_i \vee b) - f(b)$. Thus the `support DR` property holds, and submodularity holds as well.

To prove *coordinate-wise concavity*, one just need to set $b := a + \chi_i$, then it reads $f(a+\chi_i) - f(a) \geq f(a + 2\chi_i) - f(a + \chi_i)$. □

B Proofs for the monotone DR-submodular continuous functions maximization

B.1 Proof of Proposition 4.1

Proof. On a high level, the proof idea follows from the reduction from the problem of maximizing a monotone submodular set function subject to cardinality constraints.

Let us denote Π_1 as the problem of maximizing a monotone submodular set function subject to cardinality constraints, and Π_2 as the problem of maximizing a monotone DR-submodular continuous function under general down-closed polytope constraints. Following Călinescu et al. [2011], there exist an algorithm $\mathscr{A}$ for Π_1 that consists of a polynomial time computation in addition to polynomial number of subroutine calls to an algorithm for Π_2. For details see the following.

First of all, the multilinear extension [Calinescu et al., 2007] of a monotone submodular set function is a monotone submodular continuous function, and it is coordinate-wise linear, thus falls into a special case of monotone-DR submodular continuous functions.

So the algorithm $\mathscr{A}$ could be: 1) Maximize the multilinear extension of the submodular set function over the matroid polytope associated with the cardinality constraint, which can be achieved by solving an instance of Π_2. Get the fractional solution; 2) Rounding the fractional solution to be the feasible integeral solution using polynomial time rounding technique, e.g., the pipage rounding technique [Ageev and Sviridenko, 2004]. Thus we prove the reduction from Π_1 to Π_2.

And the NP-hardness of Π_2 follows from the NP-hardness of problem Π_1.

This reduction also implies the inapproximability result, coming from the optimal approximation ratio of the max-k-cover problem assuming P $\neq$ NP [Feige, 1998]. Associated with Theorem 4.4, we can conclude that the optimal approximation ratio for maximizing a monotone DR-submodular continuous function under general down-closed polytope constraints is $(1 - 1/e)$ (up to low-order terms). □

B.2 Proof of Proposition 4.2

Proof. Consider a function $g(\xi) := f(x + \xi v^*), \xi \geq 0, v^* \geq 0$. $\frac{dg(\xi)}{d\xi} = \langle v^*, \nabla f(x + \xi v^*)\rangle$.

$g(\xi)$ is concave $\Leftrightarrow$

$$\frac{d^2 g(\xi)}{d\xi^2} = (v^*)^T \nabla^2 f(x + \xi v^*) v^* = \sum_{i \neq j} v_i^* v_j^* \nabla_{ij}^2 f + \sum_i (v_i^*)^2 \nabla_{ii}^2 f \leq 0$$

The non-positiveness of $\nabla_{ij}^2 f$ is ensured by submodularity of $f(\cdot)$, and the non-positiveness of $\nabla_{ii}^2 f$ results from the coordinate-wise concavity of $f(\cdot)$. □

B.3 Proof of Lemma 4.3

Proof. It is easy to see that x^1 is a convex linear combination of points in $\mathcal{P}$, so $x^1 \in \mathcal{P}$.

Consider the point $v^* := (x^* \vee x) - x = (x^* - x) \vee 0 \geq 0$. Because $v^* \leq x^*$, we get $v^* \in \mathcal{P}$. By monotonicity, $f(x + v^*) = f(x^* \vee x) \geq f(x^*)$.

Consider the function $g(\xi) := f(x + \xi v^*), \xi \geq 0$. $\frac{dg(\xi)}{d\xi} = \langle v^*, \nabla f(x + \xi v^*)\rangle$. From Proposition 4.2, $g(\xi)$ is concave, hence

$$g(1) - g(0) = f(x + v^*) - f(x) \leq \frac{dg(\xi)}{d\xi}\Big|_{\xi=0} \times 1 = \langle v^*, \nabla f(x)\rangle$$

Then one can get

$$\begin{aligned}
&\langle v_m^t, \nabla f(x)\rangle \geq \alpha \langle v^*, \nabla f(x)\rangle - \frac{1}{2}\delta L \geq \\
&\alpha(f(x + v^*) - f(x)) - \frac{1}{2}\delta L \geq \alpha(f(x^*) - f(x)) - \frac{1}{2}\delta L
\end{aligned}$$

□

B.4 Proof of Theorem 4.4

Proof. From the Lipschitz continuous derivative assumption of $g(\cdot)$ (Eq. 5):

$$\begin{aligned}
f(x^t + \gamma v_m^t) - f(x^t) &= g(\gamma) - g(0) \\
&\geq \gamma \langle v_m^t, \nabla f(x)\rangle - \frac{L}{2}\gamma^2 \\
&\geq \gamma\alpha[f(x^*) - f(x^t)] - \frac{1}{2}\gamma\delta L - \frac{L}{2}\gamma^2 \quad \text{(Lemma 4.3)}
\end{aligned}$$

After rearrangement,

$$f(x^{t+1}) - f(x^*) \geq (1-\gamma\alpha)(f(x^t) - f(x^*)) - \frac{1}{2}\gamma\delta L - \frac{L}{2}\gamma^2$$

Therefore,

$$\begin{aligned}
&f(x^1) - f(x^*) \geq \\
&(1-\gamma\alpha)^{\frac{1}{\gamma}}(f(x^0) - f(x^*)) - \frac{L}{2}\gamma(\gamma+\delta)[1 + (1-\gamma\alpha) + (1-\gamma\alpha)^2 + \cdots + (1-\gamma\alpha)^{\frac{1}{\gamma}}] \\
&= (1-\gamma\alpha)^{\frac{1}{\gamma}}(0 - f(x^*)) - \frac{L}{2\alpha}(\gamma+\delta)[1 - (1-\gamma\alpha)^{\frac{1}{\gamma}}] \\
&\geq -e^{-\alpha}f(x^*) - \frac{L}{2}(\gamma+\delta)
\end{aligned}$$

where the last inequality results from the fact that $1-\alpha \leq (1-\gamma\alpha)^{\frac{1}{\gamma}} \leq e^{-\alpha}$ when $\gamma \in (0, 1]$. After rearrangement, we get,

$$(1 - 1/e^{\alpha})f(x^*) - f(x^1) \leq \frac{L}{2}\gamma + \frac{L}{2}\delta = \frac{L}{2K} + \frac{L}{2}\delta$$

So,

$$f(x^1) - (1 - 1/e^{\alpha})f(x^*) \geq \frac{-L}{2K} + \frac{-L}{2}\delta$$

□

C Proofs for the non-monotone submodular continuous functions maximization

C.1 Proof of Proposition 5.1

Proof. The main proof follows from the reduction from the problem of maximizing an unconstrained non-monotone submodular set function.

Let us denote Π_1 as the problem of maximizing an unconstrained non-monotone submodular set function, and Π_2 as the problem of maximizing a box constrained submodular continuous function. Following the Appendix A of Buchbinder et al. [2012], there exist an algorithm $\mathscr{A}$ for Π_1 that consists of a polynomial time computation in addition to polynomial number of subroutine calls to an algorithm for Π_2. For details see the following.

Given a submodular set function $F : E \to \mathbb{R}_+$, its multilinear extension [Calinescu et al., 2007] is a function $f : [0,1]^E \to \mathbb{R}_+$, whose value at a point $x \in [0,1]^E$ is the expected value of F over a random subset $R(x) \subseteq E$, where $R(x)$ contains each element $e \in E$ independently with probability $x(e)$. Formally, $f(x) := \mathbb{E}[R(x)] = \sum_{S \subseteq E} F(S) \prod_{e \in S} x(e) \prod_{e' \notin S} (1 - x(e'))$. It can be easily seen that $f(x)$ is a non-monotone submodular continuous function.

Then the algorithm $\mathscr{A}$ can be: 1) Maximize the multilinear extension $f(x)$ over the box constraint $[0,1]^E$, which can be achieved by solving an instance of Π_2. Obtain the fractional solution $\hat{x} \in [0,1]^n$; 2) Return the random set $R(\hat{x})$. According to the definition of multilinear extension, the expected value of $F(R(\hat{x}))$ is $f(\hat{x})$. Thus proving the reduction from Π_1 to Π_2.

Given the reduction, the hardness result follows from the hardness of unconstrained non-monotone submodular set function maximization.

The inapproximability result comes from that of the unconstrained non-monotone submodular set function maximization in Feige et al. [2011] and Dobzinski and Vondrák [2012]. □

C.2 Proof of Theorem 5.2

On a high level, the proof follows the proof idea of the `DoubleGreedy` algorithm for bounded integer lattice in Gottschalk and Peis [2015]. To better illustrate the proof, we reformulate Alg. 2 into its

equivalent form in Alg. 3, where we split the update into two steps: when $\delta_a \geq \delta_b$, update x first while keeping y fixed and then update y first while keeping x fixed ($x^i \leftarrow x^{i-1} \vee \hat{u}_a \chi_{e_i}, y^i \leftarrow y^{i-1}$; $x^{i+1} \leftarrow x^i, y^{i+1} \leftarrow y^i \wedge \hat{u}_a \chi_{e_i}$), when $\delta_a < \delta_b$, update y first. This iteration index change is only used to ease the analysis.

To prove the theorem, we first prove the following Lemmas.

Algorithm 3: `DoubleGreedy` (for analysis only)

Input: $\max f(x), x \in [0, \bar{u}]$, f is potentially non-monotone, $f(0) + f(\bar{u}) \geq 0$

$x^0 \leftarrow 0, y^0 \leftarrow \bar{u}$;
for $i = 1 \rightarrow 2n$ **do**
  $\delta_a \leftarrow f(x^{i-1} \vee \hat{u}_a \chi_{e_i}) - f(x^{i-1})$ s.t. $f(x^{i-1} \vee \hat{u}_a \chi_{e_i}) \geq \max_{u_a \in [0, \bar{u}_{e_i}]} f(x^{i-1} \vee u_a \chi_{e_i}) - \delta$;
  //$\delta \in [0, \bar{\delta}]$, let $u'_a = \arg\max_{u_a \in [0, \bar{u}_{e_i}]} f(x^{i-1} \vee u_a \chi_{e_i})$ be the optimal argument
  $\delta_b \leftarrow f(y^{i-1} \wedge \hat{u}_b \chi_{e_i}) - f(y^{i-1})$ s.t. $f(y^{i-1} \wedge \hat{u}_b \chi_{e_i}) \geq \max_{u_b \in [0, \bar{u}_{e_i}]} f(y^{i-1} \wedge u_b \chi_{e_i}) - \delta$;
  //let $u'_b = \arg\max_{u_b \in [0, \bar{u}_{e_i}]} f(y^{i-1} \wedge u_b \chi_{e_i})$ be the optimal argument
  if $\delta_a \geq \delta_b$ **then**
    $x^i \leftarrow x^{i-1} \vee \hat{u}_a \chi_{e_i}, y^i \leftarrow y^{i-1}$;
    $x^{i+1} \leftarrow x^i, y^{i+1} \leftarrow y^i \wedge \hat{u}_a \chi_{e_i}$;
  else
    $y^i \leftarrow y^{i-1} \wedge \hat{u}_b \chi_{e_i}, x^i \leftarrow x^{i-1}$;
    $y^{i+1} \leftarrow y^i, x^{i+1} \leftarrow x^i \vee \hat{u}_b \chi_{e_i}$;
Return x^{2n} (or y^{2n});

Lemma C.1 is used to demonstrate that the objective value of each intermediate solution is non-decreasing,

Lemma C.1.

$$f(x^i) \geq f(x^{i-1}) - \delta, f(y^i) \geq f(y^{i-1}) - \delta, \forall i \in [2n]. \tag{12}$$

Proof. Let $j := e_i$ be the coordinate that is going to be changed. From submodularity,

$$f(x^{i-1}|x_j^{i-1} \leftarrow \bar{u}_j) + f(y^{i-1}|y_j^{i-1} \leftarrow 0) \geq f(x^{i-1}) + f(y^{i-1})$$

where $x^{i-1}|x_j^{i-1} \leftarrow \bar{u}_j$ means only change the j-th element of x^{i-1} to be $\bar{u}_j$ while keeping all others unchanged. One can verify that $\delta_a + \delta_b \geq -2\delta$.

Assume x is changed first ($\delta_a \geq \delta_b$):

We can see that the Lemma holds for the first change ($x^{i-1} \rightarrow x^i, y^i = y^{i-1}$). Now we are left to prove $f(y^{i+1}) \geq f(y^i) - \delta$.

From submodularity:

$$f(y^{i-1}|y_j^{i-1} \leftarrow \hat{u}_a) + f(x^{i-1}|x_j^{i-1} \leftarrow \bar{u}_j) \geq f(x^{i-1}|x_j^{i-1} \leftarrow \hat{u}_a) + f(y^{i-1})$$

Therefore, $f(y^{i+1}) - f(y^i) \geq f(x^{i-1}|x_j^{i-1} \leftarrow \hat{u}_a) - f(x^{i-1}|x_j^{i-1} \leftarrow \bar{u}_j) \geq -\delta$, the last inequality comes from the selection rule of δ_a.

The situation when y is changed first is similar, the proof of which is omitted here. □

Let $OPT^i := (OPT \vee x^i) \wedge y^i$, it is easy to observe that $OPT^0 = OPT$ and $OPT^{2n} = x^{2n} = y^{2n}$.

Lemma C.2.

$$f(OPT^{i-1}) - f(OPT^i) \leq f(x^i) - f(x^{i-1}) + f(y^i) - f(y^{i-1}) + 2\delta, \forall i \in [2n] \tag{13}$$

Before proving Lemma C.2, we can see that when changing i from 0 to $2n$, the objective value changes from the optimal value $f(OPT)$ to the value returned by the algorithm: $f(x^{2n})$. Lemma C.2 is then used to bound the objective loss from the assumed optimal objective in each iteration.

Proof. Let $j := e_i$ be the coordinate that will be changed. Assume x is changed, y is kept unchanged ($x^i \neq x^{i-1}, y^i = y^{i-1}$), this could happen in two situations $\delta_a \geq \delta_b$ or $\delta_a < \delta_b$.

If $x_j^i \leq OPT_j$, $OPT^i = OPT^{i-1}$, the lemma holds.

Else $x_j^i > OPT_j$, $OPT_j^i = x_j^i$, all other coordinates of OPT^{i-1} remain unchanged. And since $x_j^{i-1} = 0$, so $OPT_j^{i-1} = OPT_j$.

From submodularity,

$$f(OPT^i) + f(y^{i-1}|y_j^{i-1} \leftarrow OPT_j) \geq f(OPT^{i-1}) + f(y^{i-1}|y_j^{i-1} \leftarrow x_j^i) \tag{14}$$

Suppose for the sake of contradiction that

$$f(OPT^{i-1}) - f(OPT^i) > f(x^i) - f(x^{i-1}) + 2\delta \tag{15}$$

Summing Eq. 14 and 15 we get:

$$0 > f(x^i) - f(x^{i-1}) + \delta + f(y^{i-1}|y_j^{i-1} \leftarrow x_j^i) - f(y^{i-1}|y_j^{i-1} \leftarrow OPT_j) + \delta$$

We know that either when $\delta_a \geq \delta_b$ or $\delta_a < \delta_b$, it holds:

$$f(x^i) - f(x^{i-1}) + \delta \geq f(y^{i-1}|y_j^{i-1} \leftarrow c) - f(y^{i-1}), \forall x_j^{i-1} \leq c \leq y_j^{i-1}$$

Set $c = OPT_j$ (notice that one can always do that in both situations $\delta_a \geq \delta_b$ or $\delta_a < \delta_b$), one can get,

$$0 > f(y^{i-1}|y_j^{i-1} \leftarrow x_j^i) - f(y^{i-1}) + \delta$$

which contradicts with Lemma C.1.

The situation when y is changed, x is kept unchanged is similar, the proof of which is omitted here. □

With Lemma C.2 at hand, one can prove Theorem 5.2: Taking a sum over i from 1 to $2n$, one can get,

$$\begin{aligned} f(OPT^0) - f(OPT^{2n}) &\leq f(x^{2n}) - f(x^0) + f(y^{2n}) - f(y^0) + 4n\delta \\ &= f(x^{2n}) + f(y^{2n}) - (f(0) + f(\bar{u})) + 4n\delta \\ &\leq f(x^{2n}) + f(y^{2n}) + 4n\delta \end{aligned}$$

Then $f(x^{2n}) = f(y^{2n}) \geq \frac{1}{3} f(OPT) - \frac{4n}{3}\delta$.

D Details of revenue maximization with continuous assignments

D.1 Details about the model

$R_s(x_i)$ should be some non-negative, non-decreasing, submodular function, we set $R_s(x_i) := \sqrt{\sum_{t:x_i(t)\neq 0} x_i(t) w_{st}}$, where w_{st} is the weight of edge connecting users s and t. The first part in R.H.S. of Eq. 8 models the revenue from users who have not received free assignments, while the second and third parts model the revenue from users who have gotten the free assignments. We use w_{tt} to denote the "self-activation rate" of user t: Given certain amount of free trail to user t, how probable is it that he/she will buy after the trial. The intuition of modelling the second part in R.H.S. of Eq. 8 is: Given the users more free assignments, they are more likely to buy the product after using it. Therefore, we model the expected revenue in this part by $\phi(x_i(t)) = w_{tt} x_i(t)$; The intuition of modelling the third part in R.H.S. of Eq. 8 is: Giving the users more free assignments, the revenue could decrease, since the users use the product for free for a longer period. As a simple example, the decrease in the revenue can be modelled as $\gamma \sum_{t:x_i(t)\neq 0} -x_i(t)$.

D.2 Proof of Lemma 6.1

Proof. First of all, we prove that $g(x) := \sum_{s:x(s)=0} R_s(x)$ is a non-negative submodular function.

It is easy to see that $g(x)$ is non-negative. To prove that $g(x)$ is submodular, one just need,

$$g(a) + g(b) \geq g(a \vee b) + g(a \wedge b), \forall a, b \tag{16}$$

Let $A := \text{supp}^+(a), B := \text{supp}^+(b)$. First, because $R_s(x)$ is non-decreasing,

$$\sum_{s\in A\setminus B} R_s(b) + \sum_{s\in B\setminus A} R_s(a) \geq \sum_{s\in A\setminus B} R_s(a \wedge b) + \sum_{s\in B\setminus A} R_s(a \wedge b) \tag{17}$$

By submodularity of $R_s(x)$ (sum over $s \in E\setminus(A \cup B)$),

$$\sum_{s\in E\setminus(A\cup B)} R_s(a) + \sum_{s\in E\setminus(A\cup B)} R_s(b) \geq \sum_{s\in E\setminus(A\cup B)} R_s(a \vee b) + \sum_{s\in E\setminus(A\cup B)} R_s(a \wedge b) \tag{18}$$

Summing Eq. 17 and 18 one can get

$$\sum_{s\in E\setminus A} R_s(a) + \sum_{s\in E\setminus B} R_s(b) \geq \sum_{s\in E\setminus(A\cup B)} R_s(a \vee b) + \sum_{s\in E\setminus(A\cap B)} R_s(a \wedge b)$$

which is equivalent to Eq. 16.

Then we prove that $h(x) := \sum_{t:x(t)\neq 0} \bar{R}_t(x)$ is submodular. Because $\bar{R}_t(x)$ is non-increasing,

$$\sum_{t\in A\setminus B} \bar{R}_t(a) + \sum_{t\in B\setminus A} \bar{R}_t(b) \geq \sum_{t\in A\setminus B} \bar{R}_t(a \vee b) + \sum_{t\in B\setminus A} \bar{R}_t(a \vee b) \tag{19}$$

By submodularity of $\bar{R}_t(x)$ (sum over $t \in A \cap B$),

$$\sum_{t\in A\cap B} \bar{R}_t(a) + \sum_{t\in A\cap B} \bar{R}_t(b) \geq \sum_{t\in A\cap B} \bar{R}_t(a \vee b) + \sum_{t\in A\cap B} \bar{R}_t(a \wedge b) \tag{20}$$

Summing Eq. 19,20 we get,

$$\sum_{t\in A} \bar{R}_t(a) + \sum_{t\in B} \bar{R}_t(b) \geq \sum_{t\in A\cup B} \bar{R}_t(a \vee b) + \sum_{t\in A\cap B} \bar{R}_t(a \wedge b)$$

which is equivalent to $h(a) + h(b) \geq h(a \vee b) + h(a \wedge b)$.

Since $f(x)$ is the sum of two submodular functions and one modular function, it is submodular. □

D.3 Solving 1-D subproblem when applying the `DoubleGreedy` algorithm

Suppose we are varying $x(i) \in [0, \bar{u}(i)]$ to maximize $f(x)$. First of all, let us leave $x(i) = 0$ out, one can see that $f(x)$ is concave and smooth along χ_i when $x(i) \in (0, 1]$,

$$\frac{\partial f(x)}{\partial x(i)} = \alpha \sum_{s\neq i:x(s)=0} \frac{w_{si}}{2\sqrt{\sum_{t:x(t)\neq 0} x(t)w_{st}}} - \gamma + \beta w_{ii}$$

$$\frac{\partial^2 f(x)}{\partial x^2(i)} = -\frac{1}{4}\alpha \sum_{s\neq i:x(s)=0} \frac{w_{si}^2}{\left(\sqrt{\sum_{t:x(t)\neq 0} x(t)w_{st}}\right)^3} \leq 0$$

Let $\bar{f}(z)$ be the univariate function when $x(i) \in (0, 1]$, then we extend the domain of $\bar{f}(z)$ to be $z \in [0, 1]$,

$$\bar{f}(z) = \bar{f}(x(i)) := \alpha \sum_{s\neq i:x(s)=0} R_s(x) + \beta \sum_{t\neq i:x(t)\neq 0} \phi(x(i)) + \gamma \sum_{t\neq i:x(t)\neq 0} \bar{R}_t(x(i)), z \in [0, 1]$$

To solve the 1-D subproblem, one can use the following method: a) start with any point $z^0 \in [0, \bar{u}(i)]$; b) $z^k \leftarrow z^{k-1} - \frac{\bar{f}'(z^{k-1})}{\bar{f}''(z^{k-1})}$; c) If $z^k \geq \bar{u}(i)$, compare $\bar{f}(\bar{u}(i))$ with $f(x|x(i) \leftarrow 0)$[4] and return argument of the larger one; Else if $z^k \leq 0$, compare $\bar{f}(0)$ with $f(x|x(i) \leftarrow 0)$, if $\bar{f}(0) \geq f(x|x(i) \leftarrow 0)$, set $x(i)$ to be some very small positive number ν, otherwise set $x(i)$ to be 0; Else, go to Step b); d) After K steps, compare $\bar{f}(z^K)$ with $f(x|x(i) \leftarrow 0)$ and return argument of the larger one. This method can solve the 1-D problem with arbitrary precision if ν is sufficiently small, since $\bar{f}(z)$ is a smooth concave function. In the experiments we set $\nu = 10^{-4}$.

[4] $x|x(i) \leftarrow 0$ means only set the i-th coordinate of x to be zero while keeping all others unchanged.

E More applications

Maximum coverage. In the maximum coverage problem, there are n subsets $C_1, \cdots, C_n$ from the ground set E. One subset C_i can be chosen with "confidence" level $x(i) \in [0, 1]$, the set of covered elements when choosing subset C_i with confidence $x(i)$ can be modelled with the following monotone normalized covering function: $p_i : \mathbb{R}_+ \to 2^E, i = 1, \cdots, n$. The target is to choose subsets from $C_1, \cdots, C_n$ with confidence level to maximize the number of covered elements $|\cup_{i=1}^{n} p_i(x(i))|$, at the same time respecting the budget constraint $\sum_i c(i)x(i) \leq b$ (where $c(i)$ is the cost of choosing subset C_i). This problem generalizes the classical maximum coverage problem. It is easy to see that the objective function is monotone submodular with down-closed polytope constraints.

Text summarization. Submodularity-based objective functions for text summarization perform well in practice [Filatova, 2004, Lin and Bilmes, 2010]. Let C to be the set of all concepts, and E to be the set of all sentences. As a typical example, the concept-based summarization aims to find a subset S of the sentences to maximize the total credit of concepts covered by S. Soma et al. [2014] discussed extending the submodular text summarization model to the one that incorporates "confidence" of a sentence, which has discrete value, and modelled the objective to be a monotone submodular function over integer lattice. It is also natural to model the confidence level of sentence i to be a continuous value $x(i) \in [0, 1]$. Let us use $p_i(x(i))$ to denote the set of covered concepts when selecting sentence i with confidence $x(i)$, it can be a monotone covering function $p_i : \mathbb{R}_+ \to 2^C, \forall i \in E$. Then the objective function of the extended model is $f(x) = \sum_{j \in \cup_i p_i(x(i))} c_j$, where $c_j \in \mathbb{R}_+$ is the credit of concept j. It can be verified that this objective is a monotone submodular continuous function.