A brief note on approximate optimization of submodular functions

Alen Alexanderian*

Abstract

We briefly discuss the greedy method and a couple of its more efficient variants for approximately maximizing monotone submodular functions.

1 Introduction

Let V be a finite set with n elements.[1] For $k \in \{1, \ldots, n\}$, we let

$$\mathcal{V}_k := \{A \in \mathcal{P}(V) : |A| = k\}, \tag{1.1}$$

denote the collection of subsets of V that have k elements.[2] In this brief note, we consider optimization problems of the form

$$\max_{S \in \mathcal{V}_k} f(S), \tag{1.2}$$

where $f : \mathcal{P}(V) \to \mathbb{R}$ is a non-negative monotone submodular function with the property that $f(\emptyset) = 0$. Solving such problems by an exhaustive search is extremely challenging. This would require $\binom{n}{k}$ evaluations of f, which is prohibitive even for modest values of n and k.[3] In this note, we discuss approximate solution of such problems using the greedy method and some of its variants.

We start our discussion in Section 2 where we outline the requisite background concepts and notations. The greedy method is discussed in Section 3. In that section, we also discuss a well-known theoretical guaranty for the greedy algorithm, in the case of monotone submodular functions. Next, in Section 4, we discuss a couple of more efficient variants of the greedy algorithm.

2 Preliminaries

Let V be a finite set with $|V| = n$. Consider a set function $f : \mathcal{P}(V) \to \mathbb{R}$. We will always assume $f(\emptyset) = 0$. Before we define the notion of modularity, we define the notion of the *marginal gain*, which is also called the *discrete derivative* [1]. For $A \subset V$ and $v \in V$, we define the marginal gain of f at A with respect to s by

$$\Delta_f(v \,|\, A) := f(A \cup \{v\}) - f(A).$$

We mention the following useful relation, verifying which is a straightforward exercise. Let A and H be in $\mathcal{P}(V)$ and assume $|H| = k$ with $k \le |V|$. Letting $H = \{v_1, \ldots, v_k\}$, we can write

$$f(A \cup H) = f(A) + \sum_{j=1}^{k} \Delta_f(v_j \,|\, A \cup \{v_1, \ldots, v_{j-1}\}). \tag{2.1}$$

Note that this can be thought of a discrete analogue of the Fundamental Theorem of Calculus.[4]

*North Carolina State University, Raleigh, NC, USA. E-mail alexanderian@ncsu.edu

Last revised: August 24, 2025

[1] *In the present context, V is typically referred to as the ground set.*

[2] *Here, $\mathcal{P}(V)$ denotes the power set of V. Also, for $A \subseteq V$, $|A|$ denotes its cardinality.*

[3] *For example, we note $\binom{80}{20} = \mathcal{O}(10^{18})$.*

[4] *Recall from calculus that for $f \in C^1([a, a+h])$ where $a \in \mathbb{R}$ and $h > 0$,*

$$f(a+h) = f(a) + \int_a^{a+h} f'(t)\,dt.$$

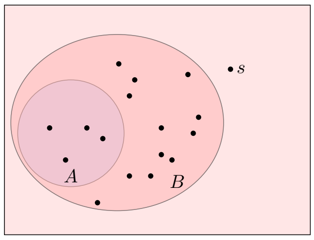
We next state the definition of a *submodular function*.

Definition 2.1. *Consider a set function $f : \mathcal{P}(V) \to \mathbb{R}$. We say f is submodular, if for every $A \subseteq V$ and $B \subseteq V$ such that $A \subseteq B$,*

$$\Delta_f(v \,|\, A) \geq \Delta_f(v \,|\, B), \quad \text{whenever } v \in V \setminus B. \tag{2.2}$$

This definition has an intuitive interpretation—this is a diminishing return property.[5] To make matters concrete, suppose the elements of V correspond to a set of experiments and suppose $f : \mathcal{P}(V) \to \mathbb{R}$ assigns a utility to each subset of V. Then, the above definition states that if the experiments in A are conducted, the marginal utility of performing the experiment $v \in V \setminus B$ does not increase if we also perform the experiments in $B \setminus A$.

[5] *The following figure provides an illustration of submodularity.*



$\Delta_f(s|A) \geq \Delta_f(s|B)$

An equivalent characterization of submodularity is provided by the following result [1]:

Proposition 2.2. *A set function $f : \mathcal{P}(V) \to \mathbb{R}$ is submodular if and only if*

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B), \quad \text{for all } A, B \in \mathcal{P}(V). \tag{2.3}$$

We next consider a basic example of a submodular function.

Example 2.3. *Let $V = \{1, \ldots, n\}$ and consider a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$. Define the function $f : \mathcal{P}(V) \to \mathbb{R}$ by*

$$f(A) = \begin{cases} \sum_{i=1}^{m} \left(\max_{j \in A} M_{ij} \right) & \text{if } A \in \mathcal{P}(V) \text{ and } A \neq \emptyset, \\ 0 & \text{if } A = \emptyset. \end{cases} \tag{2.4}$$

The function f in (2.4) is known as the *facility function* and is a well-known example of a submodular function. This function is associated with the scenario where one seeks to open facilities in n candidate locations to serve m customers. In this case, M_{ij} quantifies the value provided by the facility at the jth location to the ith customer. Assuming each customer chooses the facility that provides the highest value to them, the function f quantifies the total value provided by a given configuration of facilities.

Another key property of interest for the set functions under study is monotonicity as defined below.

Definition 2.4. *Consider a set function $f : \mathcal{P}(V) \to \mathbb{R}$. We say f is* monotone, *if for every A and B in $\mathcal{P}(V)$ such that $A \subseteq B$, $f(A) \leq f(B)$.*

An example of a monotone function is the facility function considered above, if we assume $M_{ij} \geq 0$ for all $i \in \{1, \ldots, m\}$ and $j \in V$.

3 The greedy method

Consider the optimization problem (1.2) and assume f is a monotone submodular function. Note that, in general, one considers optimization of f over subsets of V that have less than or equal to k elements. However, since f is assumed monotone, we consider the setup in (1.2) to keep the discussion simple.

A simple approach to approximately solving (1.2) is the greedy method. In this approach, we begin with the empty set $S = \emptyset$, and in each iteration pick the element in $V \setminus S$ that provides the largest marginal gain. Specifically, the greedy algorithm applied to (1.2) produces a finite sequence of sets $\{S_l\}_{l=1}^{k}$ as follows:

$$\begin{cases} S_0 = \emptyset, \\ S_l = S_{l-1} \cup \{\arg\max_{v \in V \setminus S_{l-1}} \Delta_f(v \,|\, S_{l-1})\}, \quad l = 1, \ldots, k. \end{cases} \tag{3.1}$$

The output of the greedy algorithm is the set $S = S_k$, which provides an approximate solution to (1.2).

For each $l \in \{1,\dots,k\}$, let $v_l \in \arg\max_v \Delta_f(v \,|\, S_l)$. That is v_l is the element of V that is selected in the step l of the greedy algorithm so that $S_l = S_{l-1} \cup \{v_l\}$. Note that there might be more than one element in $V \setminus S_{l-1}$ that maximize the marginal gain at that step. In such cases some form of tie-break rule must be used. The manner in which this is done does not impact the analysis that follows. Next, note that for every $l \in \{0,\dots,k-1\}$ and $v \in V$,

$$f(S_{l+1}) - f(S_l) = f(S_l \cup \{v_{l+1}\}) - f(S_l) = \Delta_f(v_{l+1} \,|\, S_l) \geq \Delta_f(v \,|\, S_l). \tag{3.2}$$

This observation will be revisited shortly.

The greedy algorithm is popular due to its simplicity. This is useful for example in the context of sensor placement, where one can place sensors in a greedy manner. In practice, this approach often provides near optimal sensor placements. In the case of monotone submodular functions, this approach admits a theoretical guaranty. This is made precise in Theorem 3.1 below. This important result was proven in [5]. We provide a proof of this result for completeness. The idea behind the proof belongs to [5]. The present proof was adapted from the presentation in [1].

Theorem 3.1. *Consider a finite set* $V = \{v_1,\dots,v_n\}$ *and assume* $f : \mathcal{P}(V) \to [0,\infty)$ *is a monotone submodular function with* $f(\emptyset) = 0$. *Let* $\{S_l\}_{l=1}^k$ *be produced by the greedy procedure* (3.1), *for a given* $k \in \{1,\dots,n\}$. *Then,*

$$f(S_k) \geq (1 - 1/e) \max_{S \in \mathcal{V}_k} f(S),$$

where $\mathcal{V}_k$ *is the collection of subsets of* V *with* k *elements as defined in* (1.1).

Proof. The result holds trivially for $k = 1$. Thus, we assume $k > 1$. Let $S^* \in \arg\max_{S \in \mathcal{V}_k} f(S)$. Enumerate elements of S^* as $S^* = \{s_1^*, s_2^*, \dots, s_k^*\}$. We note that for each $l \in \{1,\dots,k-1\}$,

$$\begin{aligned}
f(S^*) &\leq f(S^* \cup S_l) && \text{(by monotonicity of } f\text{)} \\
&= f(S_l) + \sum_{j=1}^{k} \Delta_f(s_j^* \,|\, S_l \cup \{s_1^*,\dots,s_{j-1}^*\}) && \text{(cf. (2.1))} \\
&\leq f(S_l) + \sum_{j=1}^{k} \Delta_f(s_j^* \,|\, S_l) && \text{(by submodularity of } f\text{)} \\
&\leq f(S_l) + \sum_{j=1}^{k} [f(S_{l+1}) - f(S_l)] && \text{(cf. (3.2))} \\
&= f(S_l) + k[f(S_{l+1}) - f(S_l)].
\end{aligned}$$

Hence, $f(S^*) - f(S_l) \leq k[f(S_{l+1}) - f(S_l)]$. Let $\delta_l := f(S^*) - f(S_l)$ and note

$$\delta_l \leq k(\delta_l - \delta_{l+1}). \tag{3.3}$$

This can be restated as $\delta_{l+1} \leq (1 - 1/k)\delta_l$. Note also that $\delta_0 = f(S^*) - f(\emptyset) = f(S^*)$. Therefore, we have

$$\delta_k \leq (1 - 1/k)^k \delta_0 = (1 - 1/k)^k f(S^*) \leq e^{-1} f(S^*). \tag{3.4}$$

In the last step, we have used the fact that $1 - x \leq e^{-x}$ for every $x \in \mathbb{R}$. Substituting $\delta_k = f(S^*) - f(S_k)$ in (3.4), yields $f(S^*) - f(S_k) \leq e^{-1} f(S^*)$. That is,

$$f(S_k) \geq (1 - 1/e) f(S^*),$$

which is the desired result. □

Remark 3.2. *Consider the relation* (3.3)*. The quantity* δ_l *measures the gap between* $f(S_l)$ *and the optimal objective value. An interpretation of* (3.3) *is that the improvement in optimality gap, at the step* l *of the greedy algorithm, is at least* $\big(f(S^*) - f(S_{l-1})\big)/k$*.*

4 Two variants of the greedy method

The greedy approach, while simple, can still become prohibitive as the cardinality of V grows. This is especially the case in problems of optimal sensor placement where each evaluation of f might be expensive. Over the years, several variants of the greedy approach have been proposed to accelerate computations. In this section, we discuss two such approaches: the lazy greedy and the stochastic greedy.

4.1 Lazy greedy

The idea behind the lazy greedy algorithm [3] is to make maximum use of submodularity to reduce the number of function evaluations. Recall that at the step l of the greedy method, we find an element $v \in V$ with maximum marginal gain $\Delta_f(v \,|\, S_{l-1})$; see (3.1). Moreover, by submodularity of f, we have that

$$\Delta_f(v \,|\, S_{l-1}) \geq \Delta_f(v \,|\, S_l).$$

Thus, instead of naively computing all the requisite marginal gains at each step of the greedy algorithm, we can use the already computed marginal gains as an upper bound for the subsequent ones.

Let us briefly outline the lazy greedy process. Starting with $S_0 = \emptyset$, the first iteration of the lazy greedy method is the same as standard greedy. We compute $\rho(v) := \Delta_f(v^* \,|\, S_0) = f(\{v\})$ for every $v \in V$. Subsequently, we select $s_1 \in V$ that maximizes $\rho(v)$ and let $S_1 = \{s_1\}$. Then, we sort the values of $\{\rho(v)\}_{v \in V \setminus S_1}$ in descending order. This sorted set of marginal gains will be maintained and updated in the subsequent iterations of the lazy greedy method. At the lth step of this method, we perform the following steps:

(i) take an entry $v^* \in V \setminus S_{l-1}$ that maximizes $\rho(v)$;

(ii) compute $\Delta_f(v^* \,|\, S_{l-1})$ and let $\rho(v^*) = \Delta_f(v^* \,|\, S_{l-1})$;

(iii) if v^* still maximizes $\{\rho(v)\}_{v \in V \setminus S_{l-1}}$ then let $S_l = S_{l-1} \cup \{v^*\}$ and go to step $l+1$ of the lazy greedy procedure. Otherwise, go back to (i).

While preserving the approximation guarantee of the standard greedy, the lazy greedy procedure often provides massive improvements over the standard greedy [2].

4.2 Stochastic greedy

The stochastic greedy [4] provides further improvements to the standard greedy procedure and its lazy counterpart. Recall that in step l of the standard greedy method, we need to compute the marginal gains corresponding to all elements in $V \setminus S_l$, which requires $n - l$ function evaluations. The idea of stochastic greedy is to select a random sample from this set of $n - l$ elements and evaluate the marginal gain for this randomly chosen sample. For clarity, we outline the stochastic greedy procedure in Algorithm 1.

Algorithm 1 Stochastic greedy algorithm.

Input: monotone submodular function f, ground set V of size n, size k of the desired subset of V, and sample size s.
Output: A near optimal set A with $|S| = k$.
$S = \emptyset$
for $l = 1$ **to** k **do**
  Draw a random sample set R of size s from $V \setminus S$
  $v_l = \arg\max_{v \in R} \Delta_f(v \,|\, S)$
  $S = S \cup \{v_l\}$.
end for

Note that due to its randomized nature, the output of Algorithm 1 is not deterministic. Hence, the approximation guarantee of the method is stated in expectation. The following result from [4] makes matters precise.

Theorem 4.1. *Let V be finite set with n elements and assume $f : \mathcal{P}(V) \to [0, \infty)$ is a monotone submodular function with $f(\emptyset) = 0$. Let $\varepsilon \in (0, 1)$ be given and S be a set of cardinality k obtained by performing k steps of the stochastic greedy algorithm with a sample size of $s = \lceil \frac{n}{k} \log \frac{1}{\varepsilon} \rceil$. Then,*

$$\mathbb{E}\{f(S_k)\} \geq (1 - 1/e - \varepsilon) \max_{S \in \mathcal{V}_k} f(S),$$

Note that to achieve the requisite approximation guarantee, the size s of the sample set needed in the stochastic greedy procedure is required to be $s = \lceil \frac{n}{k} \log \frac{1}{\varepsilon} \rceil$, as stated in Theorem 4.1 above.[6] Thus, the cost of performing k steps of the algorithm is $\lceil n \log \frac{1}{\varepsilon} \rceil$ function evaluations. The stochastic greedy approach provides enormous computational savings over the greedy approach and its lazy variant. Furthermore, stochastic greedy often provides solutions whose performance is close to those obtained from standard greedy approach. Additionally, as discussed in [4] the stochastic greedy can be made more efficient by incorporating lazy evaluations.

[6] *In this context n is typically large and the ratio n/k is small.*

References

[1] A. Krause and D. Golovin. Submodular function maximization. *Tractability*, 3(71-104):3, 2014.

[2] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

[3] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In J. Stoer, editor, *Optimization Techniques*, pages 234–243, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

[4] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

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