1 Submodular functions

Submodularity is a property of *set functions*, i.e., functions $f : 2^V \to \mathbb{R}$ that assign each subset $S \subseteq V$ a value $f(S)$. Hereby V is a finite set, commonly called the *ground set*. In our example, V may refer to the locations where sensors can be placed, and $f(S)$ the utility (e.g., detection performance) obtained when placing sensors at locations S. In the following, we will also assume that $f(\emptyset) = 0$, i.e., the empty set carries no value. Submodularity has two equivalent definitions, which we will now describe. The first definition relies on a notion of discrete derivative, often also called the marginal gain.

Definition 1.1 (Discrete derivative) For a set function $f : 2^V \to \mathbb{R}$, $S \subseteq V$, and $e \in V$, let $\Delta_f(e \mid S) := f(S \cup \{e\}) - f(S)$ be the *discrete derivative* of f at S with respect to e.

Where the function f is clear from the context, we drop the subscript and simply write $\Delta(e \mid S)$.

Definition 1.2 (Submodularity) A function $f : 2^V \to \mathbb{R}$ is *submodular* if for every $A \subseteq B \subseteq V$ and $e \in V \setminus B$ it holds that

$$\Delta(e \mid A) \geq \Delta(e \mid B).$$

Equivalently, a function $f : 2^V \to \mathbb{R}$ is *submodular* if for every $A, B \subseteq V$,

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B).$$

For submodular maximization, the intuition provided by the first definition is often helpful: Suppose we interpret $S \subset V$ as a set of actions which provide some benefit $f(S)$. Then the first definition says that for a submodular function f, after performing a set A of actions, the marginal benefit of any action e does not increase as we perform the actions in $B \setminus A$. Therefore, submodular set functions exhibit a natural diminishing returns property. Figure 1 illustrates this effect in our sensor placement application. In this example, the marginal benefit provided by placing a sensor at a fixed location s' given that we deployed sensors at locations s_1, s_2 does not increase as we deploy more sensors (s_3 and s_4).

An important subclass of submodular functions are those which are *monotone*, where enlarging the argument set cannot cause the function to decrease.

Definition 1.3 (Monotonicity) A function $f : 2^V \to \mathbb{R}$ is *monotone* if for every $A \subseteq B \subseteq V$, $f(A) \leq f(B)$.

Note that a function f is monotone iff all its discrete derivatives are nonnegative, i.e., iff for every $A \subseteq V$ and $e \in V$ it holds that $\Delta(e \mid A) \geq 0$. Further note that the important subclass of monotone submodular functions can be characterized by requiring that for all $A \subseteq B \subseteq V$ and $e \in V$ it holds that $\Delta(e \mid A) \geq \Delta(e \mid B)$. This is slightly different from Definition 1.2 in that we do not require $e \notin B$.

Typically, and in most of this chapter, we will assume that f is given in terms of a *value oracle*, a black box that computes[4] $f(S)$ on any input set S.

2 Greedy maximization of submodular functions

As argued in Section 1.1, submodular functions arise in many applications, and therefore it is natural to study submodular optimization. There is a large amount of work on minimizing submodular functions (*c.f.*, Fujishige 2005; Schrijver 2003). In this chapter, we will focus on the problem of maximizing submodular functions. That is, we are interested in solving problems of the form

$$\max_{S \subseteq V} f(S) \text{ subject to some constraints on } S. \tag{1}$$

The simplest example are *cardinality constraints*, where we require that $|S| \leq k$ for some k. In our example, we may wish to identify the k best locations to place sensors. Unfortunately, even this simple problem is NP-hard, for many classes of submodular functions, such as weighted coverage (Feige, 1998) or mutual information (Krause and Guestrin, 2005). While there are specialized branch and bound algorithms for maximizing submodular functions (Nemhauser and Wolsey, 1981; Goldengorin et al., 1999; Kawahara et al., 2009), ultimately their scalability is limited by the hardness of Problem 1. Therefore, in the remaining of this chapter we focus on efficient algorithms with theoretical approximation guarantees.

The greedy algorithm. In the following, we will consider the problem of approximately maximizing monotone submodular functions. A simple approach towards solving Problem 1 in the case of cardinality constraints is the *greedy algorithm*, which starts with the empty set S_0, and in iteration i, adds the element maximizing the discrete derivative $\Delta(e \mid S_{i-1})$ (ties broken arbitrarily):

$$S_i = S_{i-1} \cup \{\arg\max_{e} \Delta(e \mid S_{i-1})\}. \tag{2}$$

A celebrated result by Nemhauser et al. (1978) proves that the greedy algorithm provides a good approximation to the optimal solution of the NP-hard optimization problem.

Theorem 1.5 (Nemhauser et al. 1978) *Fix a nonnegative monotone submodular function* $f : 2^V \to \mathbb{R}_+$ *and let* $\{S_i\}_{i \geq 0}$ *be the greedily selected sets defined in Eq. (2). Then for all* *positive integers k and ℓ,*

$$f(S_\ell) \geq \left(1 - e^{-\ell/k}\right) \max_{S:|S|\leq k} f(S).$$

In particular, for $\ell = k$, $f(S_k) \geq (1 - 1/e) \max_{|S|\leq k} f(S)$.

Proof Nemhauser et al. only discussed the case $\ell = k$, however their very elegant argument easily yields the slight generalization above. It goes as follows. Fix ℓ and k. Let $S^* \in \arg\max \{f(S) : |S| \leq k\}$ be an optimal set of size k (due to monotonicity of f we can assume w.l.o.g. it is of size exactly k), and order the elements of S^* arbitrarily as $\{v_1^*, \ldots, v_k^*\}$. Then we have the following sequence of inequalities for all $i < \ell$, which we explain below.

$$\begin{aligned}
f(S^*) &\leq f(S^* \cup S_i) && (3)\\
&= f(S_i) + \sum_{j=1}^{k} \Delta\left(v_j^* \mid S_i \cup \left\{v_1^*, \ldots, v_{j-1}^*\right\}\right) && (4)\\
&\leq f(S_i) + \sum_{v \in S^*} \Delta(v \mid S_i) && (5)\\
&\leq f(S_i) + \sum_{v \in S^*} \left(f(S_{i+1}) - f(S_i)\right) && (6)\\
&\leq f(S_i) + k\left(f(S_{i+1}) - f(S_i)\right) && (7)
\end{aligned}$$

Eq. (3) follows from monotonicity of f, Eq. (4) is a straightforward telescoping sum, Eq. (5) follows from the submodularity of f, Eq. (6) holds because S_{i+1} is built greedily from S_i in order to maximize the marginal benefit $\Delta(v \mid S_i)$, and Eq. (7) merely reflects the fact that $|S^*| \leq k$. Hence

$$f(S^*) - f(S_i) \leq k\left(f(S_{i+1}) - f(S_i)\right). \tag{8}$$

Now define $\delta_i := f(S^*) - f(S_i)$, which allows us to rewrite Eq. (8) as $\delta_i \leq k(\delta_i - \delta_{i+1})$, which can be rearranged to yield

$$\delta_{i+1} \leq \left(1 - \frac{1}{k}\right)\delta_i \tag{9}$$

Hence $\delta_\ell \leq \left(1 - \frac{1}{k}\right)^\ell \delta_0$. Next note that $\delta_0 = f(S^*) - f(\emptyset) \leq f(S^*)$ since f is nonnegative by assumption, and by the well-known inequality $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$ we have

$$\delta_\ell \;\leq\; \left(1 - \frac{1}{k}\right)^\ell \delta_0 \;\leq\; e^{-\ell/k} f(S^*). \tag{10}$$

Substituting $\delta_\ell = f(S^*) - f(S_\ell)$ and rearranging then yields the claimed bound of $f(S_\ell) \geq \left(1 - e^{-\ell/k}\right) f(S^*)$. □

The slight generalization allowing $\ell \neq k$ is quite useful. For example, if we let the greedy algorithm pick $5k$ sensors, the approximation ratio (compared to the optimal set of size k) improves from $\approx .63$ to $\approx .99$.