Algorithms II December 1 and 2, 2025

Lecture 21 and 22: Submodularity and Minimization

Notes by Ola Svensson

Recall

Definition 1 *A set function f is submodular if*

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

holds for all $A, B \subseteq N$.

Examples of submodular functions

There are plentiful examples of useful and interesting submodular functions. Here we give some of our favorites. For the first three, we prove submodularity. For the remaining, we leave that as an exercise.

- Our first example is the function measuring the size of a cut in a graph $G = (V, E)$ induced by $(S, V \setminus S)$. Formally, we consider the ground set V and the define the function $\delta : 2^V \to \mathbb{R}$ by

  $$\delta(S) = |\{(u, v) : u \in S, v \in V \setminus S\}|$$

  for every subset of nodes $S \subseteq V$. To see that δ is submodular we want to measure the marginal contribution. Let $E(v, T)$ be the number of edges between some node v and a set of nodes T.

  $$\delta(v \mid S) = E(v, V \setminus (S \cup \{v\})) - E(v, S)$$

  Observe that the first term on the right is decreasing in S while the second term is increasing in S. Thus, the entire right hand side is decreasing in S. This proves the submodularity (via the "diminishing returns" definition given by Lemma **??**).

- Consider a finite collection of sets $T_1, T_2 \ldots, T_n$, where each $T_i \subseteq \mathbb{N}$ is a finite set. We consider the ground set $N = \{1, 2, \ldots, n\}$ and set function $c : 2^N \to \mathbb{R}$ by:

  $$c(S) = \left| \bigcup_{i \in S} T_i \right|,$$

  for every $S \subseteq N$. Intuitively c measures the number of elements from $\mathbb{N}$ contained in the union of the sets specified by (indices from) S. This kind of function is often referred to as a coverage function. To show that c is submodular we (again) analyze the marginal contribution of a set T_i:

  $$c(i \mid S) = \left| T_i \cup \bigcup_{j \in S} T_j \right| - \left| \bigcup_{j \in S} T_j \right| = \left| T_i \setminus \bigcup_{j \in S} T_j \right|$$

  The last term is decreasing in S which, as in the previous case, proves submodularity.

- Let $\mathcal{M} = (\mathcal{I}, X)$ be a matroid on ground set X. The rank function $r : 2^X \to \mathbb{R}$ of the matroid is defined by $r(A) = \max_{I \in \mathcal{I} \cap A} |I|$ for every $A \subseteq X$. That is, $r(A)$ is the size of a maximal independent containing only the elements in A. We will show that r is submodular. Consider any two subsets A and B of X. Let C be any maximal independent set of $\mathcal{M}$ contained in $A \cap B$. By the matroid augmentation axiom, we can extend C to a maximal independent set D contained in $A \cup B$. Then:

  $$r(A \cup B) + r(A \cap B) = |D| + |C| \leq |D \cap (A \cup B)| + |D \cap (A \cap B)| = |D \cap B| + |D \cap A| \leq r(B) + r(A).$$

  the penultimate equality follows from the inclusion-exclusion principle, and the last inequality follows since both $D \cap B$ and $D \cap A$ must be independent (as they are subsets of an independent set D) and are contained in B and A, respectively.

- When summarizing data the utility function is often submodular. This is for the same reason as in the previous coin collection example. Suppose e.g. you wish to summarize all the photos of animals in Switzerland. It is better to select one goat and one horse than two horses or two goats. This is exactly the diminishing returns property.

- Another example is influence maximization: suppose you wish to select k persons to give free samples to in order to launch your product. You wish to select k persons that have the maximum influence: they tell the most people about the product. As you will see in the exercise session, this can again be modeled as the problem of maximizing a submodular function subject to a cardinality constraint.

- Our final example is from information theory. Let $x_1, x_2, \ldots, x_n$ be discrete random variables. For any $A \subseteq [n]$, let $H(A)$ be the joint entropy of the variables $\{x_i\}_{i\in A}$. In other words, $H(A) = -\sum_a \mathbb{P}\left[\{x_i\}_{i\in A} = a\right] \log\left(\mathbb{P}\left[\{x_i\}_{i\in A} = a\right]\right)$. One can see that H is a submodular function.

1 Submodular function minimization

Next, we will start concerning ourselves with some algorithmic problems concerning submodular functions. Since we will want that our algorithms are efficient, a first issue is how the examined submodular function is given to us as part of the input. Clearly, providing the value of such a function for every possible subset of the ground set N might require a lot of space (exponential in $|N|$), while nearly every interesting problem concerning submodular functions suddenly becomes trivial, as we can go over all subsets of N in linear time and find the one that satisfies the requirements of the examined problem (e.g. if we are interested in minimizing a submodular function, we can find the subset of N with the minimum value). Also the space requirement $2^{|N|}$ is unreasonable in most settings.

Therefore, from now on we will assume that f is given in the form of access to an oracle, i.e., for every $S \subseteq N$ we can in constant time query the value $f(S)$. That is, the encoding of f is not considered part of the input.

We have already developed some intuition that submodularity is like a discrete form of concavity. Strangely enough, we now show that it is also connected to *convexity*. The first problem we will examine is called unconstrained submodular function minimization, which is the problem of finding some $S \subseteq N$ on which f takes its minimum value:

$$f(S) = \min_{S' \subseteq N} f(S') \,.$$

As we will see, this is a quite rare case of a problem of substantial interest that we can solve exactly in polynomial time. The way we will do this is by showing that it is equivalent to finding the global minimum of a certain convex continuous function, a problem we can solve using the so-called ellipsoid method in polynomial time. To be more precise, in order to minimize a (bounded) convex function g with the ellipsoid method[1], it is sufficient to solve the following problem: given an input x, either decide that x is an optimal solution or, if not, compute a (sub)gradient $\nabla g(x)$ in polynomial time. To use this framework, we need to extend f from a discrete function to a continuous function.

1.1 Lovász Extension

We want to use the general framework for minimizing a convex function to solve the problem of minimizing a submodular function $f : 2^N \to \mathbb{R}$. To do this, we will define an *extension* $\hat{f}$ of f. The definition itself makes sense for an arbitrary set function $f : 2^N \to \mathbb{R}$. However, $\hat{f}$ will be convex if and only if f is submodular.

First, we think of $f : 2^N \to \mathbb{R}$ as $f : \{0,1\}^n \to \mathbb{R}$ by associating a set S with its $0-1$ indicator from $\{0,1\}^n$ where $n = |N|$. We shall also let $N = [n] = \{1, 2, \ldots, n\}$ and interchangeably use $[n]$ and N for the ground set.

[1] Recall that a function g is convex if $g(\lambda x + (1-\lambda)y)) \leq \lambda g(x) + (1-\lambda)g(y)$ for all $\lambda \in [0,1]$ and inputs x and y.

Definition 2 *Let $f : \{0,1\}^n \to \mathbb{R}$. Define $\hat{f} : [0,1]^n \to \mathbb{R}$, the Lovász extension of f, by*

$$\hat{f}(z) = \mathop{\mathbb{E}}_{\lambda \sim [0,1]}[f(\{i : z_i \geq \lambda\})] \qquad \text{for every } z \in [0,1]^n,$$

where $\lambda \sim [0,1]$ denotes a uniformly random sample from the interval $[0,1]$.

Why is $\hat{f}$ an extension of f? Let $z \in \{0,1\}^n$. Then notice that for any $\lambda \in (0,1]$, $\{i \mid z_i \geq \lambda\} = S$ where $S = \{i \mid z_i = 1\}$. Thus $\hat{f}$ agrees with f over the hypercube (all 0-1 points) and is some kind of average of f at fractional points.

We can in fact explicitly find out this averaging representation of $\hat{f}(z)$. To do this, we define a "chain" of sets associated with any $z \in [0,1]^n$. For simplicity of notation, assume that $z_0 = 1 \geq z_1 \geq z_2 \geq \ldots \geq z_n \geq 0 = z_{n+1}$. Let S_i for any $i \in [n] \cup \{0\}$ equal $\{1, 2, \ldots, i\}$. Then, $S_0 = \emptyset \subseteq S_1 \subseteq S_2 \subseteq \cdots \subseteq S_n = [n]$.

Further notice that if $\lambda \in [z_i, z_{i+1})$ (the probability of which is equal to $z_i - z_{i+1}$) for any $i \in [n] \cup \{0\}$, then $\{j | z_j \geq \lambda\} = S_i$. Thus,

$$\hat{f}(z) = \sum_{i=0}^{n} (z_i - z_{i+1}) f(S_i)\,. \tag{1}$$

In particular, we can evaluate $\hat{f}$ at any z efficiently and we only need $n+1$ calls to the evaluation oracle.

1.2 Lovàsz Extension is Convex iff f is Submodular

The key result that allow us to minimize submodular functions in polynomial time is the following:

Theorem 3 *Let $\hat{f}$ be the Lovàsz extension of $f : \{0,1\}^n \to \mathbb{R}$. Then, $\hat{f}$ is convex iff f is submodular.*

Proof We will only show the "if" part – that is, if f is submodular then $\hat{f}$ is convex. This is what is needed for our minimization algorithm. The proof is a neat application of LP Duality and is a reminder that duality is not only useful in algorithm design but also a powerful method of arguing structural properties especially in situations where one can define the quantity of interest as the optimum value of a linear program.

We will assume that $f(\emptyset) = 0$. This is without loss of generality as otherwise we can define a new function $f' = f - f(\emptyset)$ which satisfies $f'(\emptyset) = 0$ and is just a shifted version of f.

In Definition 2, we have a definition of the Lovász extension via an explicit formula. We now give a second definition of the same function as an optimum of a linear program. Specifically, for $z \in [0,1]^n$ and f, the submodular function above, consider the following linear program (with exponentially many constraints):

$$\begin{aligned} g(z) = \max_x \; & z^\top x \\ \text{subject to: } & \sum_{i \in S} x_i \leq f(S) \qquad \forall S \subseteq N \\ & \sum_{i \in N} x_i = f(N)\,. \end{aligned}$$

We will call it the primal (P) program. Our main claim is that $g(z) = \hat{f}(z)$ for every $z \in [0,1]^n$. In other words, the LP optimum above is an alternate definition of $\hat{f}(z)$.

To see why this helps, we observe that g is convex. The idea is simple and general, the maximum of a linear function over a convex set is always convex. Concretely, for $0 \leq \lambda \leq 1$ and $z_1, z_2 \in [0,1]^n$, observe that $g(\lambda z_1 + (1-\lambda) z_2) = \max_x (\lambda z_1^\top x + (1-\lambda) z_2^\top x) \leq \lambda \max_x z_1^\top x + (1-\lambda) \max_x z_2^\top x = \lambda g(z_1) + (1-\lambda) g(z_2)$, where the maximum is taken over the x's that are feasible to the above linear program (P).

Thus, proving $\hat{f} = g$ completes the proof. To show this, we will use weak duality. Verify that the dual linear program (D) to (P) is given by:

$$\begin{aligned} \min \ & \sum_{S\subseteq N} y_S f(S) \\ \text{subject to:} \ & \sum_{S\ni i} y_S = z_i \qquad \forall i \in N \\ & y_S \geq 0 \qquad \forall S \subsetneq N\,. \end{aligned}$$

How does this help us? For a fixed $z \in [0,1]^n$, we want to prove that the optimum value of (P) is $\hat{f}(z)$. We will do this by "guessing" the optimal solutions to (P) and (D) (you'll see why it's intuitive to guess these solutions, especially when you know what we want to prove). Given such a guess, how do we verify that they are indeed optimal?

By weak duality, for any x feasible for (P) and y feasible for (D), $z^\top x \leq \sum_{S\subseteq N} y_S f(S)$. So if our guesses x^* for (P) and y^* for (D) it so happens that $z^\top x = \sum_{S\subseteq N} y_S f(S)$, then x^* and y^* have to be optimal solutions for the respective linear programs as any better solution will violate weak duality!

For notational convenience, assume (without loss of generality by relabeling coordinates) that $z_0 = 1 \geq z_1 \geq z_2 \geq \ldots \geq z_n \geq 0 = z_{n+1}$. Further, let S_i for any $i \in [n] \cup \{0\}$ equal $\{1, 2, \ldots, i\}$. We define x^* by: $x_i^* = f(S_i) - f(S_{i-1})$ for every $i \in [n]$ and y^* by:

$$y_S^* = \begin{cases} z_i - z_{i+1} & \text{for } S = S_i \text{ with } i \in [n] \\ 0 & \text{otherwise.} \end{cases}$$

We now make four claims about x^* and y^* which will complete the proof. First, we establish that $z^\top x^* = \hat{f}(z)$.

CLAIM 1: $z^\top x^* = \hat{f}(z)$.

Next, we claim that the weak duality condition is satisfied by equality for x^* and y^*.

CLAIM 2: $z^\top x^* = \sum_{S\subseteq N} y_S f(S)$.

And finally,

CLAIM 3: y^* is feasible for (D).

CLAIM 4: x^* is feasible for (P).

Using our discussion from above, the above claims complete the proof. Let us now verify them one by one. First,

$$\begin{aligned} z^\top x^* &= \sum_{i=1}^{n} z_i \cdot (f(S_i) - f(S_{i-1})) && (2) \\ &= \sum_{i=0}^{n} (z_i - z_{i+1}) f(S_i) && (3) \\ &= \sum_{S\subseteq N} y_S^* f(S)\,. && (4) \end{aligned}$$

(3) along with (1) establishes Claim 1. (4) establishes Claim 2. Let us now proceed to Claims 3 and 4. Observe that y^* has all non-negative entries because $z_i \geq z_{i+1}$ for every i. Further, for any i,

$$\sum_{S:S\ni i} y_S^* = \sum_{j\geq i} y_{S_j}^* = (z_i - z_{i+1}) + (z_{i+1} - z_{i+2}) + \cdots + (z_{n-1} - z_n) + (z_n - z_{n+1}) = z_i - z_{n+1} = z_i\,,$$

as required. This establishes that y^* is feasible for (D).

Let us now finally come to Claim 4. Observe that $\sum_{i\in[n]} x_i^* = \sum_{i\le n}(f(S_i) - f(S_{i-1})) = f([n]) - f(\emptyset) = f([n])$ using that $f(\emptyset) = 0$. Let $S \subsetneq [n]$. Then, we must show that $\sum_{i\in S} x_i^* \le f(S)$. We will argue this by induction on the size of the set S. The base case trivially holds as $f(\emptyset) = 0 \ge \sum_{i\in\emptyset} x_i^*$. Let us now prove the inductive case (observe that so far we haven't used the submodularity of f – this is the part of the argument it comes in).

Let i be the largest index in S. By definition of submodularity, $f(S)+f(S_{i-1}) \ge f(S\cup S_{i-1})+f(S\cap S_{i-1}) = f(S_i) + f(S\setminus\{i\})$ which can be rearranged to

$$f(S) \ge f(S_i) - f(S_{i-1}) + f(S\setminus\{i\}) = x_i^* + f(S\setminus\{i\}) \ge \sum_{i\in S} x_i^*\,,$$

where we used the inductive hypothesis in the final inequality. This completes the proof of Claim 4.

■

1.3 Some remarks

- The arguments presented in these notes present the "core" of why we can minimize submodular functions efficiently. To obtain faster algorithms than that given by the Ellipsoid method is a very active research area.

- Using the Ellipsoid method, one can in fact show that we can in polynomial time minimize the Lovász extension subject to (polynomially) many inequalities.

- In contrast, maximizing a submodular function generalizes the max cut problem and is thus NP-hard.

- In the proof of Theorem 3, we showed that the Lovász extension equals the optimum value to the linear program (P). By strong duality, we also have that it equals the optimum value of the dual (D). One can see that implies that the Lovász extension is the convex closure of f. The convex closure f^- of f is a function $f^- : [0,1]^n \to \mathbb{R}$ such that, if $D^-(z)$ for $z \in [0,1]^n$ is a distribution on subsets of N with marginal probabilities given by z (i.e. $\Pr_{S\sim D^-(z)}[u \in S] = z_u$) which minimizes

  $$\mathop{\mathbb{E}}_{S\sim D^-(z)}[f(S)]$$

  then $f^-(z) = \mathbb{E}_{S\sim D^-(z)}[f(S)]$. .