Introduction to Modern Combinatorial Optimization **POSTECH, Winter 2025**
Lecture #9: submodular function minimization January 16, 2025
Lecturer: Dabeen Lee

Outline

In this lecture, we consider the problem of minimizing a submodular function. We characterize the convex hull of the epigraph of a submodular function, based on the extended polymatroid. This gives rise to a separation-based algorithm for submodular function minimization. As an application, we propose a branch-and-cut framework for solving a chance-constrained program.

1 Submodular functions

Let E be a set of elements. We say that a set function $f : 2^E \to \mathbb{R}$ is **submodular** if it satisfies

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \quad \text{for all } A, B \subseteq E.$$

An equivalent definition of submodularity for set functions is the notion of **diminishing marginal returns** property. That is, a set function $f : 2^E \to \mathbb{R}$ is submodular if and only if it satisfies

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B) \quad \text{for all } A \subseteq B \subseteq E \text{ and } e \notin B.$$

Many functions that arise in discrete and combinatorial optimization problems turn out to be submodular. Let us provide a few representative examples below.

- **Linear function:** For any $w \in \mathbb{R}^{|E|}$, f with $f(S) = \sum_{e \in S} w_e$ for $S \subseteq E$ is submodular.
- **Concave utility:** For any concave function $g : \mathbb{R}_+ \to \mathbb{R}$ and $w \in \mathbb{R}_+^{|E|}$, f with $f(S) = g(\sum_{e \in S} w_e)$ for $S \subseteq E$ is a submodular function.
- **Coverage function:** Suppose that each element $e \in E$ corresponds to some area A_e. Then f with $f(S) = | \cup_{e \in S} A_e|$ for $S \subseteq E$ is submodular.
- **Success probability:** Let $p_e \in [0, 1]$ for $e \in E$. Then f with $f(S) = 1 - \prod_{e \in S}(1 - p_e)$ for $S \subseteq E$ is submodular.
- **Graph cuts:** Let $G = (V, E)$ be an undirected graph. Then f with $f(S) = |\delta(S)|$ for $S \subseteq V$ is submodular, where $\delta(S)$ is the set of edges crossing the partition $(S, V \setminus S)$ of the vertex set V.
- **Directed cuts:** Let $D = (N, A)$ be a directed graph. Then f with $f(S) = |\delta^+(S)|$ for $S \subseteq V$ is submodular, where $\delta^+(S)$ is the set of arcs leaving S.
- **Matroid rank functions:** Let $\mathcal{M} = (E, \mathcal{I})$ be a matroid. Then its rank function r given by $r(S) = \max\{|A| : A \in \mathcal{I}\}$ for $S \subseteq E$ is submodular.

As this wide range of examples suggests, submodular functions provide a useful framework for modeling discrete-valued decision variables. For utility, coverage, and success probability functions, the problem of maximizing a submodular function is relevant. For cut functions, submodular function minimization is relevant. As a first step, in this lecture, we consider the minimization problem.

2 Submodular function minimization

Let us consider the problem of minimizing a submodular function. Given a submodular function $f : 2^E \to \mathbb{R}$ over the element set E, we consider

$$\text{minimize} \quad f(S) \quad \text{subject to} \quad S \subseteq E. \tag{9.1}$$

Since f is a set function, we can interpret the function over the set of binary vectors $\{0,1\}^{|E|}$. To be more precise, any $S \subseteq E$ can be represented by its characteristic vector $\mathbf{1}_S \in \{0,1\}^{|E|}$ that takes 1 for the elements in S and 0 for the other elements. Similarly, any vector $z \in \{0,1\}^{|E|}$ corresponds to a subset $S_z = \{e \in E : z_e = 1\}$. Then, with a slight abuse of notation, we may define

$$f(z) := f(S_z).$$

In this case, (9.1) can be rewritten as the following binary optimization problem:

$$\text{minimize} \quad f(z) \quad \text{subject to} \quad z \in \{0,1\}^{|E|}. \tag{9.2}$$

Note that with an auxiliary variable y to make the objective linear, (9.2) is equivalent to

$$\text{minimize} \quad y \quad \text{subject to} \quad (y, z) \in Q_f \tag{9.3}$$

where Q_f is the **epigraph** of f given by

$$Q_f = \left\{ (y, z) \in \mathbb{R} \times \{0,1\}^{|E|} : \ y \geq f(z) \right\}.$$

Since y is a linear function, it follows that (9.3) is equivalent to

$$\text{minimize} \quad y \quad \text{subject to} \quad (y, z) \in \operatorname{conv}(Q_f) \tag{9.4}$$

where $\operatorname{conv}(Q_f)$ is the convex hull of Q_f. By the equivalence between optimization and separation, the optimization problem (9.4) is equivalent to separation over $\operatorname{conv}(Q_f)$.

Next we will characterize the convex hull of Q_f and provide a linear description of it. To do so, we need to define the **extended polymatroid** of f, given by

$$EP_f := \left\{ \pi \in \mathbb{R}^{|E|} : \ \sum_{e \in S} \pi_e \leq f(S) \quad \text{for all } S \subseteq E \right\}.$$

Note that the extended polymatroid is nonempty if and only if $f(\emptyset) \geq 0$. In general, a submodular function f does not have to satisfy $f(\emptyset) \geq 0$. Nevertheless, we may take $f - f(\emptyset)$, instead of f, which is submodular if f is submodular. Henceforth, we assume that $f(\emptyset) = 0$. Having defined the extended polymatroid, we are ready to characterize the convex hull of Q_f.

Theorem 9.1 (Edmonds [3], Lovász [6]). *Let $f : \{0,1\}^{|E|} \to \mathbb{R}$ be a submodular function with $f(\emptyset) = 0$, and let Q_f be its epigraph. Then*

$$\operatorname{conv}(Q_f) = \left\{ (y, z) \in \mathbb{R} \times [0,1]^{|E|} : \ y \geq \pi^\top z \quad \textit{for all } \pi \in EP_f \right\}.$$

Given $(y, z) \in \mathbb{R} \times [0,1]^{|E|}$, deciding whether $(y, z) \in \operatorname{conv}(Q_f)$ boils down to computing the maximum value of $z^\top \pi$ over all $\pi \in EP_f$. Edmonds [3] proved that there is a greedy algorithm for computing the maximum of a linear function over the extended polymatroid EP_f.

Theorem 9.2 (Edmonds [3]). *Let $z \in \mathbb{R}^{|E|}$. Then the linear program*

$$\max \left\{ \sum_{e \in E} z_e \pi_e : \ \pi \in EP_f \right\} \tag{P}$$

can be solved in $O(|E| \log |E|)$ time by a greedy algorithm.

Proof. We provide an algorithmic proof. If $z_e < 0$ for some $e \in E$, then the linear program is unbounded, as we can set $\pi_e = -\infty$. Thus we may assume that $z_e \geq 0$ for all $e \in E$. Let n denote the number of elements in E. Then we may enumerate the elements of E by $e_1, \dots, e_n$. Let $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$ denote a permutation so that

$$z_{\sigma(1)} \geq z_{\sigma(2)} \geq \cdots \geq z_{\sigma(n)}.$$

Then we define a sequence of sets $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_n$ given by

$$S_i := \{e_{\sigma(1)}, \dots, e_{\sigma(i)}\}.$$

Let $\bar{\pi} \in \mathbb{R}^E$ be the vector whose coordinates are given by

$$\bar{\pi}_{e_i} = \begin{cases} f(S_\sigma(1)) & \text{if } i = 1 \\ f(S_{\sigma(i)}) - f(S_{\sigma(i-1)}) & \text{if } i \geq 2. \end{cases}$$

Next, we take the dual of (P):

$$\min \left\{ \sum_{S \subseteq E} y_S f(S) : \begin{array}{rl} \displaystyle\sum_{S \subseteq E : e \in S} y_S = z_e & \text{for all } e \in E, \\ y_S \geq 0 & \text{for all } S \subseteq E \end{array} \right\}. \tag{D}$$

Let $\bar{y} \in \mathbb{R}^{2^E}$ be the vector whose coordinates are

$$\bar{y}_S = \begin{cases} z_{e_{\sigma(i)}} - z_{e_{\sigma(i+1)}} & \text{if } S = S_i, \ i \leq n-1 \\ z_{\sigma(n)} & \text{if } S = S_n \\ 0 & \text{otherwise} \end{cases}$$

We leave it as an exercise to show that $\bar{x}$ and $\bar{y}$ are optimal feasible solutions to (P) and (D), respectively. Note that the bottleneck of the algorithm is the ordering part, which can be done in $O(|V| \log |V|)$ time. ☐

Recall that the equivalence of optimization and separation is based on the ellipsoid method. Grötschel, Lovász, and Schrijver [5] showed that the algorithm can be turned into a strongly polynomial time algorithm.

Theorem 9.3 (Grötschel, Lovász, and Schrijver [5]). *Let $f : 2^E \to \mathbb{R}$ be submodular over the element set E. Then one can find $S \subseteq E$ minimizing f in strongly polynomial time.*

Later, Iwata, Fleischer, and Fujishige [8] and Schrijver [9] independently provided combinatorial algorithms for submodular function minimization.

3 Chance-constrained programs

We consider an inventory planning problem. A retail store prepares some inventory of items before the market opens. Therefore, the decision-maker has to prepare enough quantity of items before the market opens, based on the distribution of the stochastic demand.

- y: the amount of items that the retail store prepares before the market opens.
- h: the unit cost of preparing items before the market opens.
- b: the stochastic demand for items.

We assume that there are n possibilities, given by $b_1, \ldots, b_n$, for the stochastic demand b. Historically, the demand is equal to value b_i with probability p_i, i.e.,

$$\mathbb{P}[b = b_i] = p_i.$$

Here, $p_1, \ldots, p_n \geq 0$ and $\sum_{i=1}^n p_i = 1$. We assume that the probability distribution is known to the decision-maker.

The first attempt is to prepare again all possible scenarios. Basically, we target the largest possible demand by solving

$$\begin{aligned} \min \quad & hy \\ \text{s.t.} \quad & y \geq b_i, \quad i = 1, \ldots, n, \\ & y \in \mathbb{R}_+. \end{aligned}$$

However, targeting the largest possible demand may be a too conservative decision. Maybe the largest possible demand value occurs with probability less than 0.1% while we would face a moderate demand level with proability in most cases. How do we take this into account? Let us consider

$$\begin{aligned} \min \quad & hy \\ \text{s.t.} \quad & \mathbb{P}[y \geq b] \geq 0.95 \\ & y \in \mathbb{R}_+. \end{aligned}$$

This optimization model is called a **chance-constrained program**. Note that the constraint requires that we satisfy the stochastic demand with at least 95% chance. We might not satisfy the demand in some cases, but as long as the failure probability is at most 5%, we hare happy.

In fact, the chance-constrained program can be reformulated as an integer program. Note that

$$\mathbb{P}[y \geq b] \geq 0.95$$

is equivalent to

$$\mathbb{P}[y < b] \leq 0.05.$$

Moreover,

$$\mathbb{P}[y < b] = \sum_{i=1}^n p_i \cdot \mathbf{1}[y < b_i]$$

where

$$\mathbf{1}[y < b_i] = \begin{cases} 1, & \text{if } y < b_i, \\ 0, & \text{otherwise.} \end{cases}$$

To model $\mathbf{1}[y < b_i]$, we use a binary variable $z_i \in \{0,1\}$ with

$$z_i = \begin{cases} 0, & \text{if the demand for scenario } i \text{ is satisfied,} \\ 1, & \text{otherwise.} \end{cases}$$

Then the chance-constrained program can be reformulated as the following integer program.

$$\begin{aligned} \min \quad & hy \\ \text{s.t.} \quad & y + b_i z_i \geq b_i, \quad i = 1, \ldots, n, \\ & \sum_{i=1}^{n} p_i z_i \leq 0.05, \\ & y \in \mathbb{R}_+, z \in \{0,1\}^n. \end{aligned}$$

Note that any feasible solution (y, z) to the chance-constrained program belongs to

$$\{(y, z) \in \mathbb{R} \times \{0,1\}^n : \ y + b_i z_i \geq b_i, \quad i = 1, \ldots, n\}.$$

We refer to the set as the **binary mixing set** [7]. Let us define a function $f : \{0,1\}^n \to \mathbb{R}$ as

$$f(z) = \max\{b_i(1 - z_i) : i \in \{1, \ldots, n\}\}.$$

Note that the binary mixing set can be equivalently written as

$$Q_f = \{(y, z) \in \mathbb{R} \times \{0,1\}^n : \ y \geq f(z)\}.$$

Lemma 9.4. *The function $f : \{0,1\}^n \to \mathbb{R}$ with $f(z) = \max\{b_i(1 - z_i) : i \in \{1, \ldots, n\}\}$ is submodular.*

Proof. We may define the equivalent set function representation of f, given by $f(S) = f(\mathbf{1}_S)$. Then

$$f(S) = \max\{b_i : \ i \in \bar{S}\}$$

where $\bar{S} = \{1, \ldots, n\} \setminus S$. Let $S, T \subseteq \{1, \ldots, n\}$. Then we have

$$f(S \cup T) = \max\{b_i : \ i \in \bar{S} \cap \bar{T}\} \quad \text{and} \quad f(S \cap T) = \max\{b_i : \ i \in \bar{S} \cup \bar{T}\}.$$

We may observe that

$$\max\{b_i : \ i \in \bar{S} \cap \bar{T}\} + \max\{b_i : \ i \in \bar{S} \cup \bar{T}\} \geq \max\{b_i : \ i \in \bar{S}\} + \max\{b_i : \ i \in \bar{T}\},$$

which shows that $f(S \cup T) + f(S \cap T) \geq f(S) + f(T)$, establishing the submodularity of f. $\square$

Based on Lemma 9.4, we may deduce the following approach to solve the chance-constrained program.

1. We solve the LP relaxation of the integer programming formulation.
2. If the current solution $(y, z) \notin \mathrm{conv}(Q_f)$, then we separate an inequality based on the greedy algorithm of Theorem 9.2.

References

[1] Atamtürk, A. and Narayanan, V.: Polymatroids and mean-risk minimization in discrete optimization. Operations Reaearch Letters **89**(1), 618–622 (2008)

[2] Atamtürk, Alper and Nemhauser, George L. and Savelsbergh, Martin: The mixed vertex packing problem. Mathematical Programming **89**, 35–53 (2000)

[3] Edmonds, J.: Submodular functions, matroids, and certain polyhedra. In: Combinatorial Structures and Their Applications (Proceedings Calgary International Conference on Combinatorial Structures and Their Applications, Calgary, Alberta, 1969; R. Guy, H. Hanani, N. Sauer, J. Schönheim, eds.), Gordon and Breach, New York, 69–87 (1970) 9.1, 2, 9.2

[4] Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**, 169–197 (1981)

[5] Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1988) 2, 9.3

[6] Lovász, L.: Mathematical Programming The State of the Art: Bonn 1982. 235-257, Springer, Berlin (1983) 9.1

[7] Luedtke, James and Ahmed, Shabbir and Nemhauser, George L.: An integer programming approach for linear programs with probabilistic constraints. Mathematical Programming **122**, 247–272 (2010) 3

[8] Iwata, S., Fleischer, L., Fujishige, S.: A combinatorial strongly polynomial algorithm for minimizing submodular functions. J. of ACM **48**, 761–777 (2001) 2

[9] Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. J. Combin. Theory, Series B **80**, 346–355 (2000) 2