IE 511: Integer Programming, Spring 2021 **9 Mar, 2021**

Lecture 13: Submodular Functions, Polymatroids, TDI

Lecturer: Karthik Chandrasekaran *Scribe: Karthik*

Recall that we were interested in the maximum weight independent set problem over matroids. We formulated it as an IP using the rank function of matroids. We were in the process of understanding if the LP relaxation of the IP has an integral optimal solution. If so, then this would allow us to solve several discrete optimization problems (that can be formulated as a combinatorial optimization problem where the feasible sets correspond to independent set of matroids) by solving a suitable LP. Recall the rank function and its properties that we proved in the previous lecture:

Recap

Definition 1. The *rank function* of a matroid $(N, \mathcal{I})$ is the function $r : 2^N \to \mathbb{Z}_{\geq 0}$ given by

$$r(A) := \max\{|I| : I \subseteq A, I \in \mathcal{I}\} \ \forall A \subseteq N.$$

Lemma 1.1. *(i)* $r(A) \leq r(B) \ \forall A \subset B \subseteq N$.

(ii) $r(A) + r(B) \geq r(A \cup B) + r(A \cap B) \ \forall A, B \subseteq N$.

In this lecture, we will study a broader family functions which satisfy the two conditions of Lemma 1.1. Functions which satisfy the second condition are commonly encountered in economics and game theory where utilities are involved. These functions are called submodular functions. We will see certain properties of submodular functions which will help us solve the max weight independent set problem in matroids by solving a LP.

13.1 Submodular Functions

Definition 2. Let N be a finite set, 2^N denote the collection of subsets of N, and $f : 2^N \to \mathbb{R}$ be a function.

(i) f is *non-decreasing* if $f(A) \leq f(B) \ \forall A \subseteq B \subseteq N$.

(ii) f is *submodular* if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \ \forall A, B \subseteq N$.

Example:

(1) **Linear function.** Let $w : N \to \mathbb{R}$ denote the weights of the elements of a finite set N. Consider the function $f : 2^n \to \mathbb{R}$ defined by

$$f(S) := \sum_{i \in S} w_i \quad \forall S \subseteq N. \tag{13.1}$$

The function f is known as the *linear function.*

Proposition 3. *The function f in (13.1) is submodular.*

Proof. Let $A, B \subseteq N$. Then $f(A)+f(B) = \sum_{i\in A} w_i + \sum_{i\in B} w_i = \sum_{i\in A\cup B} w_i + \sum_{i\in A\cap B} w_i = f(A\cap B) + f(A\cup B)$. □

Observation. If $w_i \geq 0\ \forall i \in N$, then f is non-decreasing.

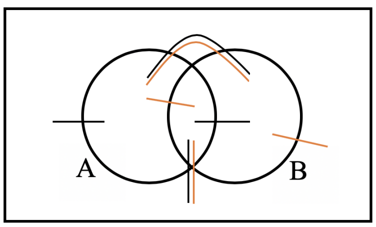
(2) **Graph cut function.** Let $G = (V, E)$ be an undirected graph. Consider $N := V$, and the function $f : 2^V \to \mathbb{R}$ defined by

$$f(S) := \text{ number of edges with exactly one end vertex in } S\ \forall S \subseteq V. \tag{13.2}$$

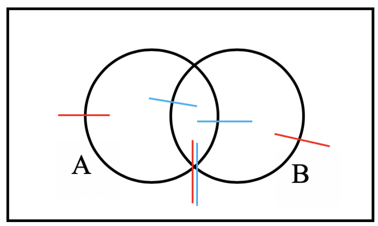
The function f is known as the *graph cut function*.

Proposition 4. *The cut function $f : 2^V \to \mathbb{R}_+$ in (13.2) is submodular.*

Proof. Let $A, B \subseteq V$. We need to show that $f(A) + f(B) \geq f(A\cup B) + f(A\cap B)$. See Figure 13.1 for a proof by picture.



(a) $f(A) + f(B)$



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

Figure 13.1: $f(A)$ counts the edges in the position of the black edges, $f(B)$ counts the edges in the position of the orange edges, $f(A\cup B)$ counts the edges in the position of the red edges, and $f(A\cap B)$ counts the edges in the position of the blue edges. Thus, every edge that is counted by $f(A\cup B) + f(A\cap B)$ is also counted by $f(A) + f(B)$.

□

Observation. f may be decreasing. See Figure 13.2.

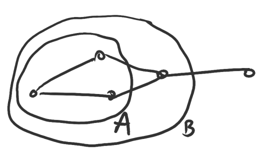


Figure 13.2: The cut function may be decreasing. Here, we have $f(A) = 2$ while $f(B) = 1$ with $A \subset B$.

Submodular function and the law of diminishing marginal returns. Submodular functions are commonly encountered in finance and economic contexts because they are exactly the functions that satisfy the law of *diminishing marginal returns*. The law of diminishing marginal returns is summarized by the inequality given in the following proposition:

Proposition 5 (Exercise)**.** *A function $f : 2^N \to \mathbb{R}$ is submodular iff $f(A+e) - f(A) \geq f(B+e) - f(B)$ $\forall A \subseteq B \subseteq N, \forall\ e \in N$.*

The inequality in Proposition 5 says that an element e brings at least as much *marginal improvement* when it is added to a set A as it does when added to a superset B of A. This property is known as the law of diminishing marginal returns. This property is common for *utility functions* that come up in economics—e.g., the increase in "happiness" by eating an apple is going to be larger when your stomach is not already full.

Submodular minimization problem

A fundamental optimization problem associated with submodular functions is the submodular minimization problem stated below.
Given: a finite set N, a submodular function $f : 2^N \to \mathbb{R}$ (given by the value function oracle)
Goal: $\min\{f(S) : S \subseteq N\}$
Submodular minimization gives a unified formulation for several discrete optimization problems. However, our focus is *NOT* on how to solve the submodular minimization problem. Instead, we will focus on a polyhedron associated with submodular functions. We will see that this polyhedron is integral and use the integrality of this polyhedron to solve the max weight independent set problem in matroids.

13.2 Optimizing over Polymatroid

Recap

Recall the polyhedron for matroid $(N, \mathcal{I})$ with rank function $r : 2^N \to \mathbb{Z}_{\geq 0}$ that we defined in the previous lecture:

$$P^{\text{ind}} := \left\{ x \in \mathbb{R}^N : \sum_{e \in S} x_e \leq r(S)\ \forall S \subseteq N, x_e \geq 0\ \forall e \in N \right\}.$$

For the purpose of solving the max weight independent set problem in matroids, we are interested in the integrality of the polyhedron P^{ind}. We have shown that the rank function is non-decreasing, submodular, and $\text{rank}(\emptyset) = 0$. More generally, we will study the following polyhedron associated with a submodular function.

Definition 6. Let $f : 2^N \to \mathbb{R}$ be a non-decreasing submodular function with $f(\emptyset) = 0$. The *submodular polyhedron* associated with f is

$$P_f := \left\{ x \in \mathbb{R}^n : \sum_{j \in S} x_j \leq f(S)\ \forall S \subseteq N, x_j \geq 0\ \forall j \in N \right\}.$$

P_f is also known as the *polymatroid* associated with f.

Note the similarity between P_f and P^{ind}. In fact, P^{ind} is equal to P_f, where f is the rank function of a matroid.

Observation. $P_f \neq \emptyset$. This is because $\bar{x} = 0 \in P_f$ since $f(\emptyset) = 0$ and f is non-decreasing.

We will see the following important result for the system describing a polymatroid.

Theorem 7 (Edmonds). *Let $f : 2^N \to \mathbb{R}$ be a non-decreasing submodular function with $f(\emptyset) = 0$.*

1. *The following system is TDI:*

$$\sum_{j \in S} x_j \leq f(S) \qquad \forall S \subseteq N$$
$$x_j \geq 0 \qquad \forall j \in N$$

2. *There exists a* greedy *algorithm to solve* $\max\{c^T x : x \in P_f\}$ *using $|N|$ queries to the value function oracle and in time poly($|N|$).*

Before proving the theorem, we discuss the significance of this result. Using the fact that a TDI-system with integral RHS describes an integral polyhedron (result from Lecture 11), we have the following corollary.

Corollary 7.1. *If $f : 2^N \to \mathbb{R}$ is an integer-valued non-decreasing submodular function with $f(\emptyset) = 0$, then the polyhedron P_f is integral.*

Note that if f is integer-valued, then P_f is integral and furthermore, we can optimize over P_f in time that is a polynomial in the size of the ground set N (as opposed to optimizing in time $2^{|N|}$). In particular, this allows us to solve the max weight independent set problem in matroids without having to explicitly write down the exponentially many constraints in the LP-relaxation of its BIP. Recall that the rank function of a matroid is non-decreasing and submodular with $\text{rank}(\emptyset) = 0$.

Corollary 7.2. *(i) P^{ind} is integral.*

(ii) Maximum weight independent set in a given matroid can be solved using $|N|$ queries to the rank function oracle and in time poly($|N|$).

This allows us to solve the maximum weight forest problem as well as the maximum weight linearly independent set problem. In fact, this result can be used to solve all discrete optimization problems which can be formulated as a max weight independent set problem over matroids, i.e., all combinatorial optimization problems whose feasible sets are the independent sets of some matroid. We now prove Theorem 7.

Proof of Theorem 7. We focus on proving the second conclusion. We will see that our proof also leads to the first conclusion. For the second conclusion, we would like to solve the LP

$$\max\left\{c^T x : \sum_{j \in S} x_j \leq f(S) \ \forall S \subseteq N, x_j \geq 0 \ \forall j \in N\right\}. \tag{13.3}$$

Note that problem (13.3) is an LP. In order to prove the second conclusion, we need an algorithm to solve this LP. Observe that the number of constraints in this LP is exponential.

Let $n := |N|$. We will solve this LP by giving a greedy algorithm to find an optimal solution. Note that *if the LP was integral* and the submodular function is the rank function of a matroid, then a solution would correspond to a solution to the matroid optimization problem (i.e., to find a maximum weight independent set in a given matroid). To start off, let us consider a natural approach to solving the matroid optimization problem, namely the greedy approach: We would like to find an independent set with max weight. For this, let us order the elements in non-increasing order of weights and process the elements in this order. We start from the empty set and add the i^{th} element to the current set if it preserves independence. We will see that this greedy algorithm finds an optimal solution for the matroid optimization problem. In fact, we will see that this greedy approach also generalizes to optimizing over polymatroids, i.e., to solve the LP given in (13.3). The generalization is given in Algorithm 1.

Algorithm 1: The Greedy Algorithm

Input: Finite set $N = [n]$, non-decreasing submodular function $f : 2^N \to \mathbb{R}$ satisfying $f(\emptyset) = 0$ given by the value function oracle

Output: $\max\{c^T x : x \in P_f\}$

Order the elements such that $c_1 \geq c_2 \geq c_r > 0 \geq c_{r+1} \geq \cdots \geq c_n$.

Let $S^i := \{1, \ldots, i\}$ for all $i \in [n]$ and $S^0 := \emptyset$

Set $x_i := \begin{cases} f(S^i) - f(S^{i-1}) & \text{for } i = 1, \ldots, r \text{ and} \\ 0 & \text{otherwise.} \end{cases}$

We will show that the solution x constructed by the greedy algorithm described in Algorithm 1 is feasible and optimal to (13.3).

Feasibility: We have that $x_j \geq 0\ \forall j \in N$ since f is non-decreasing. Let $T \subseteq N$. We need to show that $\sum_{j \in T} x_j \leq f(T)$. We have

$$\sum_{j \in T} x_j = \sum_{j \in T \cap S^r} x_j = \sum_{j \in T \cap S^r} \left(f(S^j) - f(S^{j-1}) \right).$$

For $j \in T$, we have

$$\begin{aligned} f(S^j \cap T) + f(S^{j-1}) &\geq f((S^j \cap T) \cap S^{j-1}) + f\left((S^j \cap T) \cup S^{j-1}\right) && \text{(Submodularity)} \\ &= f(S^{j-1} \cap T) + f(S^j). && \text{(See Figure 13.3)} \end{aligned}$$

Hence, $f(S^j \cap T) - f(S^{j-1} \cap T) \geq f(S^j) - f(S^{j-1})$ for every $j \in T$.

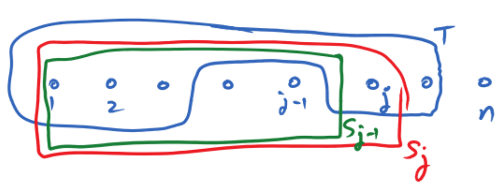


Figure 13.3: Setting of S^j and S^{j-1}

Thus, we have

$$\begin{aligned}
\sum_{j\in T} x_j &= \sum_{j\in T\cap S^r} \big(f(S^j) - f(S^{j-1})\big) \\
&\le \sum_{j\in T\cap S^r} \big(f(S^j\cap T) - f(S^{j-1}\cap T)\big) \\
&\le \sum_{j\in S^r} \big(f(S^j\cap T) - f(S^{j-1}\cap T)\big) && \text{(Since } f \text{ is non-decreasing)} \\
&= f(S^r\cap T) - f(\emptyset) \\
&\le f(T) - 0 && \text{(Since } f \text{ is non-decreasing and } f(\emptyset)=0) \\
&= f(T).
\end{aligned}$$

Thus, $x \in P_f$.

Optimality: Cost of the greedy solution that we constructed is $\sum_{i=1}^{r} c_i \big(f(S^i) - f(S^{i-1})\big) =: V$. We will show it is optimal by constructing a dual feasible solution with dual objective value V. The LP dual is

$$\begin{aligned}
\min \sum_{S\subseteq N} f(S) y_S & \\
\sum_{S: j\in S} y_S \ge c_j & \qquad \forall j \in N \\
y_S \ge 0 & \qquad \forall S \subseteq N.
\end{aligned}$$

Note that the dual has 2^N variables, i.e., a variable for each subset of N. Let

$$\begin{aligned}
y_{S^i} &:= c_i - c_{i+1} && \text{for } i = 1, \dots, r-1 \\
y_{S^r} &:= c_r \\
y_S &:= 0 && \text{for all other sets } S
\end{aligned}$$

Claim 7.1. *y is feasible.*

Proof. $y_S \ge 0 \ \forall S \subseteq N$ since we sorted the elements in non-increasing order of c. Let $j \in N$.

- *Case 1:* If $j > r$ then $\sum_{S:j\in S} y_S = 0 \ge c_j$.
- *Case 2:* If $j \le r$, then $\sum_{S:j\in S} y_S \ge \sum_{i=j}^{r} y_{S^i} = \sum_{i=j}^{r-1} (c_i - c_{i+1}) + c_r = c_j$

□

Claim 7.2. *Dual objective value of y is V.*

Proof. Dual objective value of y is

$$\begin{aligned}\sum_{i=1}^{r} f(S^i) y_{S^i} &= \sum_{i=1}^{r-1} f(S^i)(c_i - c_{i+1}) + f(S^r)c_r \\ &= \sum_{i=1}^{r} \left(f(S^i) - f(S^{i-1})\right) c_i \\ &= V.\end{aligned}$$

□

Therefore, by LP-duality, x is a primal optimum solution and y is a dual optimum solution, i.e., the greedy algorithm returns an optimal solution. The number of function evaluation queries made by Algorithm 1 is $|N|$ and the algorithm can be implemented to run in poly($|N|$) time. Thus, we have shown the second conclusion. We now show the first conclusion.

Recap

A rational linear system $Ax \leq b$ is *TDI* if for all integral c with $z^{LP} := \max\{c^T x : Ax \leq b\}$ finite, the dual $\min\{y^T b : y^T A = c^T, y \geq 0\}$ has an integral optimum solution.

If c is integral, then the dual optimal solution y that we constructed is also optimal. Thus, we have shown that the system defining the polymatroid is TDI. □