6.883 Learning with Combinatorial Structure
Note for Lecture 8
Authors: Hongyi Zhang

A bit of history in combinatorial optimization. Last time we talked about the Lovász extension, which plays an important role in the optimization of submodular functions. Actually, despite the name Lovász, Jack Edmonds is another important figure who made huge contribution to this concept, and submodular optimization in general. In DISCML 2011[1], there was a brief introduction to Jack Edmonds' contributions to mathematics and computer science by Jeff Bilmes.

1 Base polytopes

1.1 Examples of base polytopes

Recall that in the last lecture we defined the base polytope $\mathcal{B}_F$ of a submodular function, and showed that there is a greedy algorithm to solve the optimization problem

$$\max_{y \in \mathcal{B}_F} y^\top x$$

and the solution is closely related to Lovász extension. Now we shall look at several other examples where interesting concepts turn out to be the base polytopes of some submodular (or supermodular) functions.

1.1.1 Probability simplex

Let the ground set be $\mathcal{V}$ such that $|\mathcal{V}| = n$. Define $F(S) = \min\{|S|, 1\}, S \subseteq \mathcal{V}$, in last lecture we already saw that $F(S)$ is submodular. It is easy to check that $\mathcal{B}_F$ is a probability simplex. In fact, by definition

$$\begin{aligned} \mathcal{B}_F &= \left\{ y : \sum_{i \in \mathcal{V}} y_i = F(\mathcal{V}) \text{ and } \sum_{i \in S} y_i \leq F(S) \ \ \forall S \subseteq \mathcal{V} \right\} \\ &= \left\{ y : \sum_{i=1}^{n} y_i = 1 \text{ and } y_i \geq 0, \ \ \forall i \right\} \end{aligned}$$

which is a *standard* $(n-1)$*-simplex* in $\mathbb{R}^n$.

[1] `http://las.ethz.ch/discml/discml11.html` NIPS Workshop on Discrete Optimization in Machine Learning 2011

1.1.2 Permutahedron

Let the ground set be $\mathcal{V}$ such that $|\mathcal{V}| = n$. Define $F(S) = \sum_{i=1}^{|S|}(n-i+1)$. It is easy to verify that $F(S)$ is submodular. Furthermore, the base polytope of $F(S)$ is

$$\mathcal{B}_F = \left\{ y : \sum_{i=1}^{n} y_i = \frac{n(n+1)}{2} \text{ and } \sum_{i \in S} y_i \leq \sum_{i=1}^{|S|}(n-i+1), \ \forall S \subseteq \mathcal{V} \right\}$$

One can verify that $\mathcal{B}_F$ is exactly $P_{n-1}(1, \ldots, n)$, where $P_{n-1}(x_1, \ldots, x_n)$ is the permutahedron defined as the convex hull of all permutations of the vector $(x_1, \ldots, x_n)$:

$$P_{n-1}(x_1, \ldots, x_n) := \text{conv}\{(x_{\pi(1)}, \ldots, x_{\pi(n)}) | \pi \in S_n\}$$

where S_n is the symmetric group.

1.1.3 Cores of convex games

Let the ground set $\mathcal{V}$ be a set of players and $|\mathcal{V}| = n$. A cooperative game is described by a characteristic function $v : 2^{\mathcal{V}} \to \mathbb{R}$ from the set of all possible coalitions of players to a set of payments that satisfies $v(\emptyset) = 0$. A cooperative game is called convex if v is supermodular, i.e.

$$v(S) + v(T) \leq v(S \cup T) + v(S \cap T)$$

A payoff vector x assigns to each player i payoff x_i. The core of a cooperative game is the set of payoff vectors such that no coalition has incentive to leave the grand coalition, i.e.

$$C(v) = \left\{ x \in \mathbb{R}^n : \sum_{i \in \mathcal{V}} x_i = v(\mathcal{V}) \text{ and } \sum_{i \in S} x_i \geq v(S), \ \forall S \subseteq \mathcal{V} \right\}$$

which is just the base polytope $\mathcal{B}_v$.

1.1.4 Encoding correlated sources

Let $\mathcal{V}$ be a set of (possibly correlated) sources that we would like to encode, with $|\mathcal{V}| = n$. A theorem due to Slepian and Wolf (1973) and Cover (1975) states that then sequences $\{X_k^{(i)}\}_{i=1}^{\infty}$ $(k = 1, \ldots, n)$ can be sent separately with rate $R(X_k)$ $(k = 1, \ldots, n)$ to a common receiver with arbitrarily small probability of error, if and only if

$$R(S) = \sum_{k \in S} R_k > H(X_S | X_{\mathcal{V} \setminus S}) = H(X_{\mathcal{V}}) - H(X_{\mathcal{V} \setminus S}), \ \forall S \subseteq \mathcal{V}$$

Note that $H(X_S|X_{\mathcal{V}\setminus S})$ is a supermodular function. Hence, the set of achievable rate region given by

$$\mathcal{P}_H = \left\{ R(X) \in \mathbb{R}^n : \sum_{k\in S} R_k \geq H(X_S|X_{\mathcal{V}\setminus S}),\ \forall S \subseteq \mathcal{V} \right\}$$

is a supermodular polyhedron, and the set of R achieving minimum total rate is the base polytope of $H(X_S|X_{\mathcal{V}\setminus S})$.

1.1.5 Max-weight spanning tree

Chow-Liu algorithm is a greedy algorithm to find the best approximation (more precisely, the M-projection) of a distribution in tree graphical models. The algorithm simply adds edges to a tree according to the pairwise mutual information in descending order, while avoiding loops. Essentially, it is a max-weight spanning tree algorithm. Interestingly, it can also be seen as maximizing some linear function over the base polytope of some submodular function. To see the connections, we shall now introduce the concept of a matroid.

2 Matroids

2.1 Definition

Definition 1. Matroid. *A finite matroid $\mathcal{M}$ is a pair $(\mathcal{V}, \mathcal{I})$, where $\mathcal{V}$ is called the* ground set, *and $\mathcal{I} \subseteq 2^{\mathcal{V}}$ is called the* independent sets, *that satisfies the following three axioms:*

A1. $\emptyset \in \mathcal{I}$.

A2. if $T \in \mathcal{I}$ and $S \subseteq T$ then $S \in \mathcal{I}$.

A3. if $S, T \in \mathcal{I}$ and $|S| < |T|$, then there exists an element $e \in T\setminus S$ such that $S \cup e \in \mathcal{I}$.

Definition 2. Bases of a matroid. *A* basis *B of a matroid is a maximal independent set, i.e. $B \in \mathcal{I}$ and $B \cup \{a\} \notin \mathcal{I},\ \ \forall a \in \mathcal{V}\setminus B$. The set of all basis $\mathcal{B}$ is called the bases of a matroid.*

From Axiom A3, it follows that all bases have the same cardinality, which we shall call the *matroid rank*, denoted as rank($\mathcal{M}$).

2.2 Examples

2.2.1 Uniform matroid

A matroid with independent sets defined by $\mathcal{I} = \{S \subseteq \mathcal{V} : \ |S| \leq k\}$ is called a uniform matroid. It is easy to verify the definition satisfies all three axioms, and rank$(\mathcal{M}) = \min\{|\mathcal{V}|, k\}$.

2.2.2 Partition constraints

Partition $\mathcal{V}$ by $\mathcal{V} = G_1 \cup G_2 \cup \cdots \cup G_m$ such that $G_i \cap G_j = \emptyset, \ \forall i \neq j$. A matroid with independent sets defined by $\mathcal{I} = \{S : \ |S \cap G_i| \leq k_i, \ \forall 1 \leq i \leq m\}$ satisfies all three axioms, and rank$(\mathcal{M}) = \sum_i \min\{|G_i|, k_i\}$

2.2.3 Cycle-free subgraphs

Let the ground set $\mathcal{V} = \{\mathcal{E} : \ \mathcal{G}(\mathcal{U}, \mathcal{E})$ is a graph$\}$ be the edge set of a graph. A matroid with independent sets defined by $\mathcal{I} = \{S : \ S \in 2^{\mathcal{V}}$ does not contain a cycle$\}$ satisfies all three axioms, and the bases are the set of spanning trees of $\mathcal{G}$, with rank$(\mathcal{M}) = |\mathcal{U}| - 1$. Now the connection to max-weight spanning tree should be clear – if we set w as the edge weights, then finding the max-weight spanning tree is equivalent to solving

$$\max_y w^\top y, \ \text{ s.t. } y \text{ is a basis of } \mathcal{M}$$

2.2.4 Linearly independent sets

Let the ground set $\mathcal{V} = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$ be a set of vectors. A matroid with independent sets defined by $\mathcal{I} = \{S : \ \{v_i\}_{i \in S}$ are linearly independent$\}$ satisfies all three axioms, and rank$(\mathcal{M})$ is the rank of matrix V with the set of column vectors $\mathcal{V}$.

2.3 Rank function

Similarly, we can define the rank of a matroid on a set $S \subseteq \mathcal{V}$ to be the size of a maximal independent subset of S, where a subset of S is independent if and only if it is an independent set in $\mathcal{M}$.

Definition 3. Rank function. *A rank function r maps $S \subseteq \mathcal{V}$ to its rank $r(S)$, i.e.*

$$r(S) = \{|T| : \ T \subseteq S \text{ and } T \in \mathcal{I}\}$$

Proposition 1. *r is a matroid rank function if and only if it satisfies the following properties:*

P1. $r(\emptyset) = 0$ and $r(S) \geq 0, \ \forall S \subseteq \mathcal{V}$.

P2. For any two subsets A and B of $\mathcal{V}$, $r(A \cup B) + r(A \cap B) \leq r(A) + r(B)$. That is, r is submodular.

P3. For any set S and element x, $r(S) \leq r(S \cup \{x\}) \leq r(S) + 1$.

P1 and P3 are easy to prove from the definition of rank function. We now show why P2 is and has to be true if r is the rank function of a matroid.

First we prove the "only if" part. Suppose r is a matroid rank function, now consider any two sets $X, Y \subseteq \mathcal{V}$. Denote $I_{X \cap Y}$ to be a maximal independent set of $X \cap Y$. Since $X \cap Y \subseteq X \subseteq X \cup Y$, by axiom A3 we can extend $I_{X \cap Y}$ to a maximal independent set I_X of X, then extend I_X to a maximal independent set $I_{X \cup Y}$ of $X \cup Y$. From the construction, we know $|I_{X \cup Y}| = |I_{X \cup Y} \cap Y| - |I_{X \cap Y}| + |I_X|$. Now consider the set $I_{X \cup Y} \cap Y$, by axiom A2 and $I_{X \cup Y} \cap Y \subseteq Y$ we know $r(Y) \geq |I_{X \cup Y} \cap Y|$. Therefore, $r(Y) \geq |I_{X \cup Y}| + |I_{X \cap Y}| - |I_X| = r(X \cup Y) + r(X \cap Y) - r(X)$. Thus r is submodular.

Since rank function r is submodular, it has a base polytope. We define the base polytope of a matroid $\mathcal{M}$ to be the base polytope of its rank function:

$$\mathcal{B}_{\mathcal{M}} = \mathcal{B}_r = \text{conv}\{1_S : \ S \text{ is a base of } \mathcal{M}\}$$

Recall that in the last lecture we introduced a greedy algorithm to solve the optimization problem of maximizing a linear function over the base polytope of a submodular function, i.e.

$$\max_{y \in \mathcal{B}_r} w^\top y$$

which is equivalent to

$$\max \sum_{e \in S} w(e) \quad \text{s.t. } S \text{ is a base of } \mathcal{M} \tag{1}$$

The greedy algorithm is the following:

1. Sort the elements of w, i.e. find a permutation $\pi(\cdot)$ such that

$$w_{\pi(1)} \geq w_{\pi(2)} \geq \cdots \geq w_{\pi(n)}.$$

2. Initialize the sets $S_0 = \emptyset$ and $S_i = \{\pi(1), \pi(2), \ldots, \pi(i)\}$ for $1 \leq i \leq n$.
3. For $1 \leq i \leq n$, loop until S_i is a maximal independent set.

The proof of the "if" part of the proposition P2 is implied by the following theorem.

Theorem 1. *$\mathcal{M} = (\mathcal{V}, \mathcal{I})$ is a matroid if and only if it satisfies A1, A2 and the greedy algorithm gives an optimal solution for (1):*

Proof. The "only if" part of the theorem follows by the correctness of the greedy algorithm to solve the optimization problem, which we have proved during the last lecture. We now prove the "if" part by contradiction.

Assume that $\mathcal{M}$ is not a matroid, i.e. axiom A3 is violated, then there exist two independent sets $I_1, I_2 \in \mathcal{I}$, such that $|I_1| < |I_2|$ but $\forall e \in I_2 \backslash I_1$ we have $I_1 \cup \{e\} \notin \mathcal{I}$. We now construct an optimization problem making the greedy algorithm to fail.

Consider the optimization problem (1) with w defined by

$$w(e) = \begin{cases} 2, & e \in I_1 \cap I_2 \\ 1, & e \in I_1 \backslash I_2 \\ \alpha, & e \in I_2 \backslash I_1 \\ 0, & \text{otherwise} \end{cases}$$

Since $|I_1| < |I_2|$, one has $|I_1 \backslash I_2| < |I_2 \backslash I_1|$. Therefore, we can choose α so that $\alpha < 1$ and $\alpha |I_2 \backslash I_1| > |I_1 \backslash I_2|$. As a result, the greedy algorithm will first select all elements in I_1. But by that time, according to our assumption, the algorithm will stop and output $\sum w(e) = w(I_1) < w(I_2)$. Hence the algorithm is not optimal. □

3 Summary

Now we take a step back and look at the big picture:

- our initial goal is to minimize a submodular function F:

$$\min_{S \subseteq \mathcal{V}} F(S) = \min_{x \in \{0,1\}^n} F(x)$$

- we construct the Lovász extension so that instead of minimizing $F(x)$ over a discrete set, we now minimize its extension $f(x)$ over the more amendable set $[0,1]^n$:

$$\min_{x \in [0,1]^n} f(x) = \min_{x \in [0,1]^n} \max_{y \in \mathcal{B}_F} y^\top x$$

- the Lovász extension $f(x)$ has several nice properties: it is convex, meaning that we can minimize it using subgradient methods; we have a greedy algorithm to compute $f(x)$ and its maximizing y, meaning that we have an efficient way to compute its subgradient; finally, the relaxation is exact, meaning that the minimum and minimizer of a submodular function agree with those of its Lovász extension.

- several algorithms can give exact solution to this problem in polynomial time.

3.1 Submodular minimization algorithms

There are in general two types of algorithms to minimize a submodular function. One type contains algorithms from convex optimization – including ellipsoid method, subgradient method and minimum-norm point method (a.k.a Fujishige-Wolfe algorithm). We have already talked about subgradient method – it is easy to implement, but won't give us an exact solution in finite time (need $O(\frac{1}{\epsilon^2})$ steps for ϵ-approximation). The most practical algorithm right now is the one solving a minimum-norm point problem, but more on that later. The other type contains combinatorial methods. Here we discuss one such algorithm using convex duality.

Note that since $f(x)$ is convex, we have the strong duality condition satisfied, thus

$$\begin{aligned}
\min_{S\subseteq\mathcal{V}} F(S) &= \min_{x\in[0,1]^n} f(x) \\
&= \min_{x\in[0,1]^n} \max_{y\in\mathcal{B}_F} y^\top x \\
&= \max_{y\in\mathcal{B}_F} \min_{x\in[0,1]^n} y^\top x \\
&= \max_{y\in\mathcal{B}_F} \left(\sum_{i=1}^{n} \min\{y_i, 0\} \right) \qquad (2)
\end{aligned}$$

So we could instead solve (2). Upon inspection, maximizing $\sum_{i=1}^{n} \min\{y_i, 0\}$ involves removing the negative entries in y while staying in the base polytope. However, we are not going to check whether some y is in the base polytope because it would require checking exponential number of constraints. Instead, what we can do is to start with a feasible y, then move mass among its entries so that we don't violate any constraints. This can be done using some network flow algorithm.