CS 369P: Polyhedral techniques in combinatorial optimization
Instructor: Jan Vondrák
Lecture date: November 16, 2010

1 Submodular functions

We have already encountered submodular functions. Let's recall the definition.

Definition 1 *Let N be a finite ground set and $f : 2^N \to \mathbb{R}$. Then f is* submodular *if for all $A, B \subseteq N$,*

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

As we have seen, an equivalent definition is as follows. We denote by $f_A(i) = f(A+i) - f(A)$ the *marginal value* of i with respect to A. Then f is submodular if for all $A \subseteq B \subseteq N$ and $i \in N \setminus B$,

$$f_A(i) \geq f_B(i).$$

More generally, we define a function $f_A : 2^N \to \mathbb{R}$ by $f_A(X) = f(X \cup A) - f(A)$. From the second definition, it is easy to see that if f is submodular then f_A is also submodular. Another equivalent definition of submodularity that is even more "local" is that f is submodular if for all $A \subset N$ and $i, j \in N \setminus A$,

$$f(A) - f(A+i) - f(A+j) + f(A+i+j) \leq 0.$$

We further classify submodular functions as follows:

- *Monotone* functions: f is monotone if $f(A) \leq f(B)$ whenever $A \subseteq B$.
- *Non-monotone* functions: no requirement as above. An important subclass of non-monotone functions are *symmetric* functions that satisfy the property that $f(A) = f(\overline{A})$ for all $A \subseteq N$.

Throughout, unless we explicitly say otherwise, we will assume that f is available via a *value oracle* which given a set $S \subseteq N$, returns $f(S)$. We say that f is given *explicitly* if f has a representation of bit size polynomial in $|N|$ and the value of $f(S)$ for any $S \subseteq N$ can be computed from this representation in time polynomial in $|N|$.

1.1 Examples of submodular functions

A number of interesting functions arising in combinatorial optimization turn out to be submodular.

- *Linear functions:* A function $f : 2^N \to \mathbb{R}$ is linear if $f(A) = \sum_{i \in A} w_i$ for some weights $w : N \to \mathbb{R}$. Such functions are also referred to as *additive* or *modular*. If $w_i \geq 0$ for all $i \in N$, then f is also monotone.
- *Budget-additive functions:* A small generalization of the linear case, the function $f(A) = \min\{\sum_{i \in A} w_i, B\}$ for any $w_i \geq 0$ and $B \geq 0$, is monotone submodular.

- *Set systems and coverage:* Given a universe U and n subsets $A_1, A_2, \ldots, A_n \subset U$, we obtain several natural submodular functions on the set $N = \{1, 2, \ldots, n\}$. First, the coverage function f given by $f(S) = |\cup_{i \in S} A_i|$ is submodular. This naturally extends to the weighted coverage function; given a non-negative weight function $w : U \to \mathbb{R}_+$, $f(S) = w(\cup_{i \in S} A_i)$. A related function defined by $f(S) = \sum_{x \in U} \max_{i \in S} w(A_i, x)$ is also submodular, where $w(A_i, x)$ is a non-negative weight for A_i covering x. All these functions are monotone.

- *Rank functions of matroids:* The rank function of a matroid $\mathcal{M} = (N, \mathcal{I})$, $r_{\mathcal{M}}(A) = \max\{|S| : S \subseteq A, S \in \mathcal{I}\}$, is monotone submodular. More generally, given $w : N \to \mathbb{R}_+$, the weighted rank function defined by $r_{\mathcal{M},w}(A) = \max\{w(S) : S \subseteq A, S \in \mathcal{I}\}$ is a monotone submodular function.

- *Cut functions in graphs and hypergraphs:* Given an undirected graph $G = (V, E)$ and a non-negative capacity function $c : E \to \mathbb{R}_+$, the cut capacity function $f : 2^V \to \mathbb{R}_+$ defined by $f(S) = c(\delta(S))$ is a symmetric submodular function. Here $\delta(S)$ is the set of all edges in E with exactly one endpoint in S. This naturally extends to hypergraphs. If $G = (V, E)$ is a hypergraph then the function $f(S) = c(\delta(S))$ is symmetric submodular, where $\delta(S)$ is the set of all hyperedges that contain both a vertex in S and $V \setminus S$. If $G = (V, A)$ is a directed graph and $c : A \to \mathbb{R}_+$ then $f : 2^V \to \mathbb{R}_+$ defined as $f(S) = c(\delta^+(S))$ is submodular; here $\delta^+(S)$ is the set of arcs leaving S. This function is typically not symmetric.

- *Valuation functions with decreasing marginal values:* Sometimes we assume that a certain function is submodular not because it arises in a specific combinatorial way, but because it arises in a setting where it's natural to assume submodularity. An example is the setting of combinatorial auctions, where each player has a *valuation function* $w : 2^N \to \mathbb{R}$ on subsets of items. This might have a specific form, like $w(S) = \min\{\sum_{j \in S} w_j, B\}$, or it might be given by a black box. However, we might assume that the (unknown) function is submodular just because in some settings it is natural to expect that having more items can only decrease the benefit of acquiring another item.

1.2 Operations on submodular functions

If f, g are submodular functions on the same ground set N then it is easy to see that $f + g$ is submodular. More generally, $\alpha f + \beta g$ is submodular for any $\alpha, \beta \geq 0$.

If g is a linear function then $-g$ is also linear and hence $f - g$ is submodular if f is submodular and g is linear. A useful context in which this arises is the following. Suppose N represents a set of items and f is a submodular valuation function of some player for bundles of items. Then, given prices $p : N \to \mathbb{R}_+$ on the items, the function $h(S) = f(S) - \sum_{j \in S} p_j$ is the utility of the player given the prices, and is also submodular.

In general, if f, g are submodular, then $f - g$, $\min(f, g)$ and $\max(f, g)$ are not necessarily submodular. Nonetheless, it is known that if $f - g$ is monotone then $\min(f, g)$ is submodular.

2 Optimization of submodular functions

A number of combinatorial optimization problems can be viewed as optimization problems with a submodular objective function. This often simplifies the problem, replacing complicated constraints by pulling them inside the objective function. In general, having a submodular objective function is more general and captures linear objective functions as a special case. Concrete examples include the following:

- *Minimum Cut:* in a graph G with nonnegative edge weights w_e, and two fixed vertices $s, t \in V$, minimize $w(\delta(S))$ over all $S \subset V$, $s \in S$, $t \notin S$. The objective function $w(\delta(S))$ is submodular but not monotone.

- *Maximum Cut:* in a graph G with nonnegative edge weights w_e, maximizing $w(\delta(S))$ over all $S \subseteq V$.

- *Minimum/Maximum Hypergraph Cut:* same thing, in a hypergraph (a hyperedge e is cut by S if $\emptyset \neq S \cap e \neq S$). The cut function is still submodular.

- *Maximum Coverage:* given $A_1, \ldots, A_m \subset U$ and $f(S) = |\bigcup_{i \in S} A_i|$, maximize $f(S)$ over $|S| \leq k$. The objective function $f(S)$ here is monotone submodular.

- *Maximum Group Coverage:* given sets $A_1, \ldots, A_m \subset U$ and k agents with different weight functions $w_i : U \to \mathbb{R}_+$, allocate each set to some agent in order to maximize

  $$\sum_{i=1}^{k} w_i(\bigcup_{j \in S_i} A_j),$$

  where S_i are the indices of sets allocated to agent i.

- *Submodular Welfare* (generalization of the previous example): Given k agents with monotone submodular functions $w_i : 2^{[m]} \to \mathbb{R}_+$, allocate m items to the agents to maximize

  $$\sum_{i=1}^{k} w_i(S_i)$$

  where S_i are the items allocated to agent i.

  This problem can be written as $\max\{f(S) : S \in \mathcal{M}\}$ where $f(S) = \sum_{i=1}^{k} w_i(S_i)$, $S \subseteq [m] \times [k]$, and $\mathcal{M}$ is a partition matroid allowing each item to appear in at most one set S_i.

  The Budgeted Allocation Problem is a special case of this, since budget-additive functions are monotone submodular.

3 The greedy algorithm

The greedy algorithm (henceforth referred to as Greedy) is a natural heuristic for maximizing a monotone submodular function subject to certain constraints. In several settings it provides good approximation ratios, and until quite recently, the approximation ratios provided by Greedy were the best known in most cases. Moreover, the simplicity of Greedy makes it useful in various applications where other algorithms may not suitable. Although the algorithm can be used also for non-monotone submodular functions, it performs poorly (without additional tricks).

Let $\mathcal{I} \subseteq 2^N$ be a collection of feasible sets, which we consider to be down-monotone (w.l.o.g. for monotone objective functions). The Greedy algorithm for the problem $\max_{S \in \mathcal{I}} f(S)$ is formally described below. It requires two subroutines. The first is a membership oracle for $\mathcal{I}$: given a set $S \subseteq N$ the oracle should return if $S \in \mathcal{I}$ or not. The second is a value oracle for f: given a set $S \subseteq N$ the oracle should return the value $f(S)$.

Algorithm **Greedy:**
  $S \leftarrow \emptyset$; $A \leftarrow \emptyset$;
  Repeat
    $A \leftarrow \{e \mid S \cup \{e\} \in \mathcal{I}\}$;
    If ($A \neq \emptyset$) then
      $e \leftarrow \mathrm{argmax}_{e' \in A} f_S(e')$;
      $S \leftarrow S \cup \{e\}$;
    Endif
  Until ($A = \emptyset$);
  Output S;

Observe that for a linear function $f(S) = \sum_{i \in S} w_i$ and $\mathcal{I}$ being the independent sets in a matroid, this is exactly the greedy algorithm which finds a maximum-weight base in matroids. In more general settings the greedy solution is not optimal. However, one setting where the algorithm works quite well is the following.

3.1 Cardinality constraint

Theorem 2 (Nemhauser, Wolsey, Fisher '78) *Greedy gives a* $(1-1/e)$*-approximation for the problem of* $\max_{|S| \le k} f(S)$ *when* $f : 2^N \to \mathbb{R}_+$ *is a monotone submodular function.*

Proof: Let S_i denote the first i elements selected by the greedy algorithm and let C denote the actual optimum, $f(C) = \mathrm{OPT}$. Greedy will select exactly k elements, i.e. S_k is the set returned by the algorithm. We claim via induction that for $0 \le i \le k$,

$$f(C) - f(S_i) \le (1 - 1/k)^i f(C). \tag{1}$$

The base case of $i = 0$ is trivially true. Suppose that $i > 0$ and in the i-th step, Greedy selects element a_i, maximizing $f_{S_{i-1}}(a_i)$ among the remaining elements. Observe that the remaining elements include $C \setminus S_{i-1}$, a set of size at most k. By submodularity, we have

$$f(C) - f(S_{i-1}) \le \sum_{a \in C \setminus S_{i-1}} f_{S_{i-1}}(a)$$

and this implies that the element a_i has marginal value

$$f_{S_{i-1}}(a_i) \ge \frac{1}{|C \setminus S_{i-1}|} \sum_{a \in C \setminus S_{i-1}} f_{S_{i-1}}(a) \ge \frac{1}{k}(f(C) - f(S_{i-1})).$$

Assuming that (1) holds true for S_{i-1}, we have

$$\begin{aligned} f(C) - f(S_i) &= f(C) - f(S_{i-1}) - f_{S_{i-1}}(a_i) \\ &\leq f(C) - f(S_{i-1}) - \frac{1}{k}(f(C) - f(S_{i-1})) \\ &= (1 - 1/k)(f(C) - f(S_{i-1})) \\ &\leq (1 - 1/k)^i f(C) \end{aligned}$$

which proves (1). Using the claim for $i = k$, we get

$$f(C) - f(S_k) \leq (1 - 1/k)^k f(C) \leq e^{-1} f(C).$$

$\square$

Interestingly, it was proved by Feige that the approximation factor $1 - 1/e$ is *optimal*. More precisely, for any fixed $\epsilon > 0$ it is NP-hard to achieve a $(1 - 1/e + \epsilon)$-approximation for the Max k-cover problem, which is a special case of $\max\{f(S) : |S| \leq k\}$ for f monotone submodular. Therefore, for this problem the greedy algorithm is the best approximation algorithm we can possibly hope for.