CS 369P: Polyhedral techniques in combinatorial optimization
Instructor: Jan Vondrák
Lecture date: December 2, 2010

1 Maximizing a monotone submodular function subject to a matroid constraint

Consider now the problem $\max\{f(S) : S \in \mathcal{I}\}$ where f is monotone submodular and $\mathcal{M} = (N, \mathcal{I})$ is an arbitrary matroid. We saw that this captures the Submodular Welfare Problem and this special case admits a $(1 - 1/e)$-approximation. In today's lecture, we want to extend this result to the more general problem $\max\{f(S) : S \in \mathcal{I}\}$. We replace this problem by its multilinear relaxation

$$\max\{F(\mathbf{x}) : \mathbf{x} \in P(\mathcal{M})\}.$$

Here, $P(\mathcal{M})$ is the *matroid polytope*,

$$P(\mathcal{M}) = \mathrm{conv}\{\chi_I : I \in \mathcal{I}\}.$$

As we explained in the last lecture, there is an efficient algorithm which solves this continuous problem within a factor of $1 - 1/e$ for any solvable polytope P. The *matroid polytope* $P(\mathcal{M})$ is solvable - in fact the linear optimization problem over $P(\mathcal{M})$ can be solved very efficiently by the greedy algorithm. Furthermore, since the linear function we need to optimize is determined by ∇F, a nonnegative vector for a non-decreasing function, we can assume that the direction $\mathbf{v}_{max}(\mathbf{x})$ found by the continuous greedy algorithm at each point is in the *matroid base polytope* $P_{base}(\mathcal{M})$. By convexity, the final point $\mathbf{x}(1)$ is also in the base polytope $P_{base}(\mathcal{M})$.

The problem that remains is how to round the fractional solution. How can we convert $\mathbf{x}(1)$ into an integral solution of comparable value? Observe that if f were a linear function (and hence F as well), this would be a trivial problem: Every point $\mathbf{x} \in P(\mathcal{M})$ can be decomposed as a convex combination $\mathbf{x} = \sum_{I \in \mathcal{I}} \alpha_I \chi_I$ where $\alpha_I \geq 0$ and $\sum \alpha_I = 1$. If F is a linear function, then $F(\mathbf{x}) = \sum_{I \in \mathcal{I}} \alpha_I f(I)$ and hence at least one of the sets I satisfies $f(I) \geq F(\mathbf{x})$. Nevertheless, this is not true for submodular functions.

We will see that there is a rounding procedure which does not lose in terms of the objective value even if the objective function is submodular. In other words, the multilinear optimization problem $\max\{F(\mathbf{x}) : \mathbf{x} \in P(\mathcal{M})\}$ has integrality gap 1 for any submodular function and any matroid polytope. More precisely, we show the following.

Theorem 1 *There is an (efficient) rounding procedure such that given any submodular function $f : 2^N \to \mathbb{R}$, matroid $\mathcal{M} = (N, \mathcal{I})$ and a point $\mathbf{x} \in P(\mathcal{M})$, it outputs an independent set $S \in \mathcal{I}$ such that $f(S) \geq F(\mathbf{x})$.*

We remark that monotonicity does not play any role in this theorem. If we prove this theorem, we obtain together with the continuous greedy algorithm a $(1-1/e)$-approximation for maximizing any monotone submodular function subject to a matroid constraint.

Corollary 2 *For any monotone submodular function $f : 2^N \to \mathbb{R}_+$ and a matroid $\mathcal{M} = (N, \mathcal{I})$, there is a $(1 - 1/e)$-approximation for the problem $\max\{f(S) : S \in \mathcal{I}\}$.*

2 Rounding in the matroid polytope

To prove Theorem 1, we proceed as follows. We would like to round a fractional solution $\mathbf{x} \in P(\mathcal{M})$ into an integer solution $S \in \mathcal{I}$ in such a way that we only move along very special directions: $\mathbf{d} = \mathbf{e}_i - \mathbf{e}_j$. As we learned, the function $F(\mathbf{x})$ is *convex* along such directions. This will be useful in the analysis.

For simplicity, we present the procedure for the base polytope $P_{base}(\mathcal{M})$. This is without loss of generality, since the objective function is monotone, and so any point $\mathbf{x} \in P(\mathcal{M})$ can be replaced by a point $\mathbf{x}' \in P_{base}(\mathcal{M})$ such that $\mathbf{x}' \geq \mathbf{x}$ and $F(\mathbf{x}') \geq F(\mathbf{x})$. As we mentioned, the continuous greedy algorithm can be actually assumed to return a point in $P_{base}(\mathcal{M})$.

So let us start with a point $\mathbf{x} \in P_{base}(\mathcal{M})$. As we know, there is a chain of tight sets $\emptyset = C_0 \subset C_1 \subset C_2 \subset \ldots \subset C_k = N$ such that $\chi_T \in \mathrm{conv}(\{\chi_{C_i} : C_i \in \mathcal{C}\})$ for every tight set T. Note that the ground set N is always tight for $\mathbf{x} \in P_{base}(\mathcal{M})$ and hence present in the chain. We also consider $\emptyset$ as a tight set and include it in the chain. We maintain a chain of tight sets throughout the procedure.

Assume that there is a fractional variable $0 < x_i < 1$ (otherwise we are done). Let C_ℓ be the smallest set in $\mathcal{C}$ containing i. (We know some set in $\mathcal{C}$ contains i because the ground set N is always in $\mathcal{C}$.) In other words, $i \in C_\ell \setminus C_{\ell-1}$. Observe that $x(C_\ell \setminus C_{\ell-1}) = x(C_\ell) - x(C_{\ell-1}) = r(C_\ell) - r(C_{\ell-1})$ is an integer, therefore $C_\ell \setminus C_{\ell-1}$ in fact contains two fractional variables x_i, x_j.

In the following step, we will move in the direction $\mathbf{d} = \mathbf{e}_i - \mathbf{e}_j$, or $\mathbf{e}_j - \mathbf{e}_i$, depending on the objective function. We can do this in a way such that $F(\mathbf{x})$ never decreases.

Lemma 3 *For any $i, j \in N$ and $\mathbf{x} \in [0,1]^N$, $F(\mathbf{x} + \xi(\mathbf{e}_i - \mathbf{e}_j))$ is either non-decreasing for $\xi \geq 0$ or non-increasing for $\xi \leq 0$.*

Proof: We know that $\phi(\xi) = F(\mathbf{x} + \xi(\mathbf{e}_i - \mathbf{e}_j))$ is a convex function of ξ. If $\phi'(0) \geq 0$, then $\phi'(\xi) \geq 0$ for all $\xi \geq 0$, by convexity. Similarly, if $\phi'(0) \leq 0$, then $\phi'(\xi) \leq 0$ for all $\xi \leq 0$. In the first case, $F(\mathbf{x} + \xi(\mathbf{e}_i - \mathbf{e}_j))$ is non-decreasing for $\xi \geq 0$, and in the second case, it is non-increasing for $\xi \leq 0$. □

Therefore, we can choose one of the two directions $\mathbf{d} = \pm(\mathbf{e}_i - \mathbf{e}_j)$ and move along $\mathbf{d}$ without decreasing $F(\mathbf{x})$. Observe the following: since every currently tight set T is such that $\chi_T = \sum \alpha_\ell \chi_{C_\ell}$, we have $\mathbf{d} \cdot \chi_T = \sum \alpha_\ell \mathbf{d} \cdot \chi_{C_\ell} = 0$. Therefore, moving along direction $\mathbf{d}$ does not violate any currently tight set. We can move along $\mathbf{d}$ until we create a *new tight constraint*. The maximum distance that we can move can be determined by solving the following LP:

$$\max\{\xi : \mathbf{x} + \xi\mathbf{d} \in P(\mathcal{M})\}.$$

Since we can solve the separation problem over $P(\mathcal{M})$, we can also solve this LP. (A more efficient algorithm for this problem was given by Cunningham. Another way to solve this problem is to use a submodular minimization algorithm.)

Now assume that we determine the maximum value of ξ such that $\mathbf{x} + \xi\mathbf{d} \in P(\mathcal{M})$, and the new tight constraint is $x_j = 0$. Then we just remove element j from consideration. The other possibility is that we find a new tight set T. Note that $i \in T$ and $j \notin T$, otherwise the set would not become tight. By the uncrossing trick that we used in the lecture about matroid intersection, we can extend the chain as follows: Let $T' = (T \cap C_\ell) \cup C_{\ell-1}$. This is a tight set again, because tight sets are closed under taking unions and intersections. Moreover $C_{\ell-1} \subset T \subset C_\ell$ because T

contains some elements that $C_{\ell-1}$ does not, but not all elements of C_ℓ. Therefore we have a new set T' that we can insert in the chain and increase its size.

We summarize the algorithm as follows. It uses a subroutine **HitConstraint** which moves $\mathbf{x}$ as far as possible along the direction $\mathbf{d} = \mathbf{e}_i - \mathbf{e}_j$ and finds the next violated constraint, which could be either $x_j = 0$ or $x(A) = r(A)$ for some set A.

Subroutine **HitConstraint**($\mathbf{x}$, i, j):
  Denote $\mathcal{A} = \{A \subseteq X : i \in A, j \notin A\}$;
  Find $\delta = \min_{A \in \mathcal{A}}(r_{\mathcal{M}}(A) - x(A))$ and $A \in \mathcal{A}$ achieving this;
  If $x_j < \delta$ then $\{x_i \leftarrow x_i + x_j,\ x_j \leftarrow 0,\ A' \leftarrow \{j\}\}$
  Else $\{x_i \leftarrow x_i + \delta,\ x_j \leftarrow x_j - \delta,\ A' \leftarrow A\}$;
  Return $(\mathbf{x}, A')$.

Algorithm **MatroidRound**($\mathbf{x}$):
  $\mathcal{C} \leftarrow \{\emptyset, E\}$;
  While ($\mathbf{x}$ contains a fractional variable) do
    Pick i, j fractional in some $C_\ell \setminus C_{\ell-1}$ for some ℓ;
    If $F(\mathbf{x} + \xi(\mathbf{e}_i - \mathbf{e}_j))$ is increasing in ξ
      then $(\mathbf{x}, T) \leftarrow$ **HitConstraint**$(\mathbf{x}, i, j)$;
      else $(\mathbf{x}, T) \leftarrow$ **HitConstraint**$(\mathbf{x}, j, i)$;
    If $x_j = 0$ then remove j from all tight sets and N;
    else add $T' = (T \cap C_\ell) \setminus C_{\ell-1}$ to $\mathcal{C}$.
  EndWhile
  Output $\mathbf{x}$.

By the discussion above, $F(\mathbf{x})$ never decreases throughout the algorithm, and we keep deleting elements or growing the chain $\mathcal{C}$. Therefore, the procedure terminates in $O(n)$ iterations and returns an integer point $\mathbf{x}' \in P(\mathcal{M})$ such that $F(\mathbf{x}') \geq F(\mathbf{x})$. This finishes the proof of Theorem 1.