Rounding Techniques in Approximation Algorithms

Lecture 13: Iterative Relaxation for Bounded Degree Matroids

Lecturer: Nathan Klein

1 Bounded Degree Matroids

1.1 Iterative Relaxation

Last class we started discussing the iterative relaxation framework, and used it to prove a discrepancy bound.

Iterative Relaxation

Consider any linear program and an extreme point solution x. Fix all integer coordinates of x, delete one of the constraints (in some problem-specific manner), and re-solve. Iterate until all coordinates are integral.

And we noted the following key fact:

Fact 1.1. *In every iteration, the cost of x can only improve. So, the cost of the resulting integer solution is no worse than integer OPT. Of course, it may not obey all the same constraints as integer OPT since we deleted some.*

In other words, we are relaxing the set of constraints instead of the requirement that we get a solution of optimal cost.

1.2 Uncrossing for Matroids

In Lecture 11, we showed that given a skew supermodular requirement function for the number of edges in each cut of a graph we can uncross the set of tight constraints to obtain a laminar family. In the past, we have also seen how uncrossing can be used to prove the integrality of a formulation of the spanning tree polytope and in general of matroid polytopes.

Here, we will prove a very similar theorem for matroids. A *chain* $\mathcal{L}$ over a ground set is a family such that for every $A, B \in \mathcal{L}$ we either have $A \subseteq B$ or $B \subseteq A$. If it is not the case that $A \subseteq B$ or $B \subseteq A$ we say A and B are incomparable.

Theorem 1.2 (Tight Sets form a Chain)**.** *Let r be the rank function of a matroid $M = (\mathcal{I}, E)$ and let $x \in P_M$. Then, the set of non-trivial tight constraints, i.e. sets $F \subseteq E$ with $x(F) = r(F)$, can be uncrossed to form a chain $\mathcal{L}$ so that $\{\chi(F) \mid F \in \mathcal{L}\}$ is linearly independent, consists of sets F with $x(F) = r(F)$, and spans the set of all non-trivial tight constraints.*

Proof. At this point we are quite familiar with these ideas. We start by proving that if A, B are tight then so are $A \cup B$ and $A \cap B$ in the usual way:

$$r(A) + r(B) = x(A) + x(B) = x(A \cap B) + x(A \cup B) \leq r(A \cap B) + r(A \cup B) \leq r(A) + r(B)$$

where we use the submodularity of the rank function in the last step. So, we obtain an equality. Now, take any maximal linearly independent chain of tight sets. We will prove this is the desired set.

Suppose by way of contradiction that there is a set A which is not in the span of $\mathcal{L}$ and among all such sets is incomparable with the fewest sets in $\mathcal{L}$. Let B be a set in $\mathcal{L}$ it is incomparable with. By the above, $A \cup B$ and $A \cap B$[1] are both tight, and notice that both these sets are incomparable with fewer sets in $\mathcal{L}$.

Let's prove this for $A \cup B$, as the other case is similar. Suppose $A \cup B$ is incomparable with a set C.

1. First suppose that $B \subseteq C$. If $A \subseteq C$, then $A \cup B \subseteq C$ which contradicts that $A \cup B$ is not a subset of C. If $C \subseteq A$, then $C \subseteq A \cup B$, again a contradiction.

2. Otherwise $C \subseteq B$. If $A \subseteq C$, then $C \subseteq A \cup B$, contradiction. If $C \subseteq A$ then $C \subseteq A \cup B$, contradiction.

So, if $A \cup B$ is incomparable with C then so is A. But $A \cup B$ is not incomparable with A. So, it is incomparable with strictly fewer sets as we set out to show.

Thus both $A \cap B$ and $A \cup B$ are in the span, implying that $\mathcal{L}$ spans A, since $\chi(A) + \chi(B) = \chi(A \cap B) + \chi(A \cup B)$. □

This gives us a new way to prove that P_M is integral for every matroid. In fact, we get something even stronger: the tight constraints of a matroid only require "half" of the tokens we can generate.

Corollary 1.3. *P_M has integral vertices. Furthermore, if there are k fractional variables at a non-vertex point $x \in P_M$, the rank of the tight constraints (after putting all integral elements in the basis) is at most $\lfloor \frac{k}{2} \rfloor$.*

Proof. Contract all elements set to 1 and delete all elements set to 0: the resulting object is a new matroid polytope.

Let $L_1 \subseteq L_2 \subseteq \cdots \subseteq L_m$ be the sets in $\mathcal{L}$ from the above theorem. Since $\mathcal{L}$ is linearly independent, $L_i \setminus L_{i-1} \neq \emptyset$ for $1 \leq i \leq m$ where $L_0 = \emptyset$. Thus, by the integrality of the rank function, $x(L_i \setminus L_{i-1}) \geq 1$. This implies that $|L_i \setminus L_{i-1}| \geq 2$ since all elements are fractional. Since every elements lies in at most one such set, $|E| \geq 2m$, giving the claim. □

In other words, there is a way to assign tokens from elements to tight constraints such that every constraint gets *two* tokens. This suggests that maybe we can handle two matroids at the same time and still maintain integrality. This turns out to be true!

1.3 Matroid Intersection

Let M_1, M_2 be any two matroids. Let $P_{M_1 \cap M_2}$ be the set of points x with $x \in P_{M_1}$ and $x \in P_{M_2}$. Then we can prove the following:

Lemma 1.4. *Every matroid intersection polytope has integral vertices.*

[1]Note $A \cap B$ may be $\emptyset$.

Proof. We break the constraints into those from M_1 and those from M_2. Suppose there are k fractional variables at a point $x \in P_{M_1 \cap M_2}$. Contract all elements set to 1 and delete all elements set to 0. By Corollary 1.3, the rank of the tight constraints for each matroid is at most $\lfloor \frac{k}{2} \rfloor$. So, we are done unless both have size $\frac{k}{2}$ and both chains contain the full set of elements. But if they do then there is a linear dependence as these must be the same constraint. $\square$

This allows us to, for example, find a minimum cost *arborescence* in polynomial time. An arborescence is a set of edges in a directed graph so that for a fixed root r with in-degree 0, there is a unique directed path from r to v for every $v \in V \setminus \{r\}$.

An arborescence is the intersection of a spanning tree matroid and a partition matroid, where in the partition matroid we constrain $|\delta^+(v)| = 1$ for all $v \in V \setminus \{r\}$.

1.4 Bounded Degree Matroids

Another way to try to exploit the fact that matroids take only "half" the tokens is to add additional constraints. We will make this work in this subsection, the main result of this lecture.

Consider a matroid $M = (\mathcal{I}, E)$ with weights $c_e \geq 0$ for all $e \in E$, and in addition suppose we have a hypergraph H with hyperedges $B_1, \ldots, B_k$ over the ground set E and integers $b_1, \ldots, b_k$. We now want to find a minimum cost basis B of M and ensure that $|B \cap B_i| \leq b_i$ for each $1 \leq i \leq k$. Note that if the hypergraph has maximum degree 1 then this can be modeled as a partition matroid and we can return a basis with no violation at all.

Otherwise, let's model this as a linear program: we want to minimize $\sum_{e \in E} c_e x_e$ subject to $x \in P_M^B$ defined as follows:

$$P_M^B = \begin{cases} x \in P_M \\ x(B_i) \leq b_i & \forall 1 \leq i \leq k \\ 0 \leq x_e \leq 1 & \forall e \in E \end{cases}$$

Theorem 1.5 ([KLS12]). *Suppose every element has degree at most Δ in H and that some basis exists that obeys all bounds. Let B^* be the cheapest basis obeying all bounds. Then there is a polynomial time algorithm which outputs a basis B of M with $|B \cap B_i| \leq b_i + \Delta - 1$ for all $1 \leq i \leq k$ and $c(B) \leq c(B^*)$.*

Proof. We will use the iterative relaxation framework. At each timestep, we will fix integral variables by deleting elements with $x_e = 0$, contracting elements with $x_e = 1$ (and updating the b_i values), and dropping constraints which have at most $b_i + \Delta - 1$ fractional elements. If the algorithm can always find a new variable set to 0 or 1 or drop a constraint, we will make progress and end with an integral solution with the desired bounds, as if there are $b_i + \Delta - 1$ fractional elements in a constraint we can clearly violate the bound by at most $\Delta - 1$.

Assign x_e tokens to the smallest set in $\mathcal{L}$ containing e. Assign $\frac{1-x_e}{\Delta}$ tokens to each set B_i containing e.

Every set $F \in \mathcal{L}$ gets at least one token. The reason is as follows. Let F' be the child of F (possibly, $F' = \emptyset$). Every element in $F \setminus F'$ contributes x_e tokens to F. But $x(F \setminus F') = x(F) - x(F') > 0$ as otherwise $F = F'$ and it must be an integer, so it is at least 1.

Secondly, every tight B_i gets at least this many tokens, using that $|B_i| \geq b_i + \Delta$ (as otherwise we would have dropped the constraint):

$$\frac{1}{\Delta}(|B_i| - x(B_i)) = \frac{1}{\Delta}(|B_i| - b_i) \geq \frac{1}{\Delta}(b_i + \Delta - b_i) = 1$$

Finally we need to argue that there is a token left over. We are done if any fractional element is not in exactly Δ tight constraints B_i since otherwise it has some leftover tokens. Furthermore, if the set of all elements is not in $\mathcal{L}$ we are done, as some elements will give no tokens to a set in $\mathcal{L}$. So, we may assume both exist. But now, summing over all constraints B_i gives us ΔE, which is a linear dependence. $\square$

1.5 Bounded Degree Spanning Trees

We obtain the following as a corollary, which was a major open question in combinatorial optimization until this beautiful method was discovered by Singh and Lau in 2007. A previous result by Goemans [Goe06] gave the same result with $k+2$ instead of $k+1$ based on matroid intersection.

Theorem 1.6 ([SL15])**.** *Let $G = (V, E)$ be a weighted graph and $k \in \mathbb{N}$. Assume there is a tree with maximum degree k, and let T^* be the cheapest such tree. Then, there exists a polynomial time algorithm which outputs a tree of cost at most $c(T^*)$ and maximum degree $k+1$.*

This allows us to consider a different metric for approximation: instead of losing on cost, we can output a solution that has slightly weaker properties than the OPT we compare against. Notice that the above theorem is tight (unless P=NP). Setting $k = 2$, this is the Hamiltonian path problem, so it cannot be solved without losing something on maximum degree (even without costs).

1.6 Extensions

It turns out that the same guarantee can be given for a spanning tree if we have lower bounds as well: we can return a tree with minimum degree $l_i - \Delta + 1$ and maximum degree $u_i + \Delta - 1$ given lower and upper bounds l_i, u_i for each set B_i.

However, for a general matroid the best known is $l_i - 2\Delta + 1$ and $u_i + 2\Delta - 1$. Finding an algorithm with an improved guarantee is an open problem.

References

[Goe06] Michel X. Goemans. "Minimum Bounded Degree Spanning Trees". In: *FOCS*. 2006, pp. 273–282 (cit. on p. 4).

[KLS12] Tamás Király, Lap Chi Lau, and Mohit Singh. "Degree Bounded Matroids and Submodular Flows". In: *Combinatorica* 32 (2012), pp. 703–720. DOI: 10.1007/s00493-012-2760-6 (cit. on p. 3).

[SL15] Mohit Singh and Lap Chi Lau. "Approximating minimum bounded degree spanning trees to within one of optimal". In: *Journal of the ACM* 62.1 (2015). DOI: 10.1145/2629366 (cit. on p. 4).