COMS E6998-5: Algorithms through Geometric Lens — October 11, 2017

Lecture 13 – Max-Cut Problem

Instructors: *Alex Andoni and Ilya Razenshteyn* — Scribe: *Srikar Varadaraj*

1 Introduction

Today's topics:

- Max-Cut Problem
- Crash course on Mathematical Programming (MP)
- Best possible known approximation

2 Max-Cut Problem

Definition 1. *Given a Graph G on n nodes, the max-cut problem is to find a partition $S \cup \bar{S}$ which maximizes the cut $(S, \bar{S})$ = # of edges from S to $\bar{S}$.*

Solving this problem exactly is $NP-hard$. We would like to find an approximate algorithm that does well.

Goal 1. *Find $(S, \bar{S})$ such that:*

$$cut(S, \bar{S}) \geq \alpha cut(S^*, \bar{S}^*) \tag{1}$$

Here $*$ denotes the optimal configuration. We would like α to be as large as possible and we would like to find the approximate cut in polynomial time.

Claim 2. *We can achieve $\alpha = \frac{1}{2}$.*

Proof. Choose S to be a random subset of V, the set of vertices in G. Below I use the notation $i\&j$ to denote that vertices i and j are separated.
Analysis:

$$\mathbb{E}_S cut(S, \bar{S}) = \mathbb{E}_S[\sum_{(i,j)\in E} \chi[i\&j]] = \sum_{(i,j)\in E} \frac{1}{2} = \frac{m}{2} \tag{2}$$

Since $cut(S^*, \bar{S}^*) \leq m$, we can conclude that $\alpha = \frac{1}{2}$ is achievable. □

By repeating the procedure above many times, we can get the approximation $\alpha = \frac{1}{2}$ with high probability (say $\geq 90\%$).

3 Crash course on MP

If we have $x_1, ..., x_n \in \mathbb{R}$, a sample MP problem would be to find **max** $f(x)$ such that $x \in P$, where P is some set of constraints.
In our problem, we can consider:

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{if } i \notin S \end{cases}$$

So, our max-cut problem boils down to finding:

$$\max \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2}$$

The general LP problem above is NP-hard. MP can be solved in two broad cases:

- Linear Programs
- Convex Programs

Our technique will be to "relax" our MP into more tractable MP(such as LP/CP). Essentially, we will find the solution to an MP' problem instead of our current MP problem and then "massage" the solution to yield an approximate solution for our original problem. We will "relax" and then "round".

3.1 Relaxation

Relaxation gives us semidefinite programming (SP). The condition $x_i \in \{1, -1\}$ is reinterpreted as $x_i^2 = 1$ and now instead of considering x_i as a binary variable, we can think of it as a vector such that $||x_i|| = 1$. Hence, we expand our class of solutions. The Max-Cut SDP is now:

$$max \sum_{(i,j)\in E} (1 - x_i \cdot x_j)/2$$

where

$$||x_i|| = 1$$

Note that we now consider the dot products $x_i \cdot x_j$ instead of simple products. This SDP problem can now be solved in poly(description complexity * $\log(1/\epsilon)$) using the interior point method, for example and so it is polytime as opposed to exponential time.
The general SDP problem looks like:

$$max \sum_{i,j} w_{ij} G_{ij} \text{ such that } \sum_{i,j} A_{ij}^k G_{ij} \geq b^k$$

where

$$G = v^T v$$

for some matrix v. G is known as the Gram matrix.

Theorem 3. *The following are equivalent:*

- *$\exists v$ such that $G = v^T v$*
- *$\forall x \in \mathbb{R}^n$, $x^T G x \geq 0$*
- *All eigenvalues of G are ≥ 0, or G is positive semi-definite.*

Proof. The Proof that (1) $\implies$ (2): $x^T G x = x^T v^T v x = ||vx||_2^2 \geq 0$, as required. □

We also have the claim that $\forall G$ satisfying (2), they form a convex subset in $\mathbb{R}^{n^2}$, although we won't prove it here.
So far, we have $\phi =$ max obtained by MC, SDP. $x_1, ..., x_n$ are vectors and $G = v^T v$. Our claim is that ϕ is a relaxation of the original max-cut. We then have

$$\phi \geq \text{cut}(S^*, \bar{S}^*)$$

We would like $x_i \cdot x_j \sim 1$ if x_i and x_j are roughly in the same direction. If they are in opposite directions, we would like $x_i \cdot x_j \sim -1$. The algorithm we use makes use of a random hyperplane algorithm first suggested by Goemans and Williamson (1994). We choose the random hyperplane from a gaussian distribution g.
We have:

$$\mathbb{E}_g[\text{cut}(S, \bar{S})] = E_g[\sum_{(i,j)} w_{ij} \chi(S_i \neq S_j)] = \sum_{(i,j)} w_{ij} \Pr[i \& j]$$

Now note that $\Pr[i\&j] = \frac{\text{angle between } x_i \text{ and } x_j}{\pi}$. So, we have

$$\mathbb{E}_g[\text{cut}(S, \bar{S})] = \sum_{(i,j)} w_{ij} \frac{\arccos(x_i \cdot x_j)}{\pi} \geq \alpha\phi \geq \alpha \text{cut}(S^*, \bar{S}^*)$$

If we find the point where $\frac{\arccos(x_i \cdot x_j)}{\pi}$ is farthest from the line joining points $(-1, 1)$ and $(1, 0)$, then we see that at the worst point, we have:

$$\alpha_{GW} = \min \frac{arccos(r/\pi)}{(1-r)/2} = 0.878$$

In conclusion, we can always find S using our random hyperplane method such that:

$$\mathbb{E}_S cut(S, \bar{S}) \geq \alpha_{GW} cut(S^*, \bar{S}^*)$$

By repeating the procedure above many times, we can guarantee with high probability an α_{GW} approximation to the optimal max-cut. Under the unique games conjecture proposed by Subhash Khot, $\alpha_{GW} = 0.878$ turns out to be the best possible approximation factor.

Next time: We will talk about the 3-coloring problem.