**CPSC 665 : An Algorithmist's Toolkit** Lecture 8 : 4 Feb 2015

# Semidefinite Programming Part 2

*Lecturer: Sushant Sachdeva* *Scribe: Alex Reinking*

## 1. Introduction

In this lecture we continued our discussion of semidefinite programming, and extended our previous methods to the Balanced Separator problem. Unlike Max-Cut, Balanced Separator requires that the Semidefinite Relaxation be *strengthened* to reduce the space of possible answers.

## 2. Max-Cut Revisited

As in last week's lecture, we approximate solutions to Max-Cut using Goemans's and Williamson's $\alpha_{GW} = 0.878$-approximation. Specifically, we seek $\max \sum_{(i,j)\in E} \frac{1}{4} \left\|v_i - v_j\right\|^2$ subject to the constraint that $\forall i, \left\|v_i\right\|^2 = 1$. We can visualize this by drawing the vectors restricted to a unit circle, as seen in the figure to the left. There is an appealing geometric intuition here. When seen as assignments of unit vectors to vertices, it becomes clear that you want them somehow as far apart as possible.

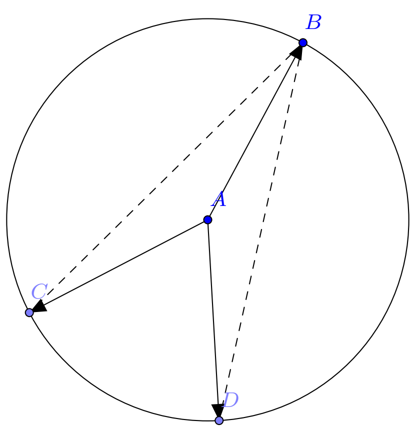


Figure 1. Illustration of SDP constraints.

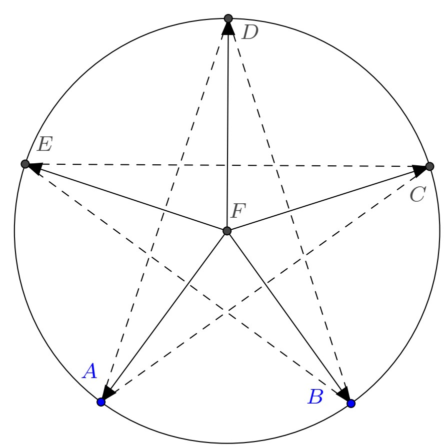


Figure 2. Illustration of $C_5$ situation.

So, for a concrete example, consider the 5-vertex cycle, $C_5$. The actual Max-Cut of this graph is 4. Drawn as a pentagram, we can see that the angles between adjacent vertices are far apart – note that this does not mean adjacent with respect to the points, but to their order in the cycle.

Specifically, the angle between these vectors is $2 \cdot \frac{2\pi}{5} = \frac{4\pi}{5}$. Thus, the semidefinite program gives $\sum\limits_{\text{edges}} \frac{1.5}{4}(1 + 1 - 2\cos\frac{4\pi}{5}) = \frac{5}{2}(1 - \cos\frac{4\pi}{5})$. The actual approximation ratio this particular instance gives is 0.885. Thus, the integrality gap cannot be greater than this, as remarked in the following lemma.

**Lemma 2.1.** *Integrality Gap* $\leq 0.885$

Now, given that [GW95] proved the integrality gap is at least 0.878 (using a rounding algorithm), we can see that there isn't much room left for improvement between 0.878 and 0.885.

2.1. **Higher Dimensions and Generalization.** Now, suppose that instead of $\mathbb{R}^2$, we had vectors in $\mathbb{R}^3$ or higher. Then, rather than cutting with a line, we cut with an $n$ dimensional hyperplane. We can actually construct these cuts reliably with a randomized algorithm:

(1) Sample a uniformly random Gaussian, $g \sim N(0, 1)$
(2) Return $\left\{ i \in V | v_i^T g \geq 0 \right\}$ where $v_i$ come from SDP solution.

**Theorem 2.2.** *The above algorithm produces a cut $S$ such that*

$$\mathbf{E}[\textit{no. edges cut}] \geq \alpha_{GW} \cdot \mathit{SDP}$$

*That is, there is a cut in $G = (V, E)$ so that the number of edges cut is greater than $\alpha_{GW} \cdot \mathit{SDP}$.*

But this is for any cut, thus this is at most the max cut. So

$$\text{Max-Cut} \geq \mathbf{E}[\text{no. edges cut}] \geq \alpha_{GW} \cdot \text{SDP}$$

which is the same as

$$\text{Max-Cut} \geq \sum_{(i,j) \in E} \mathbf{Pr}[(i, j) \text{is cut}] \geq \alpha_{GW} \cdot \text{SDP}$$

But this leaves us with the question. Given the algorithm above, what is $\mathbf{Pr}[(i, j)\text{is cut}]$? This is done geometrically, so we need only look at $v_i, v_j$. Suppose that their span is this page, as displayed in the figure below.

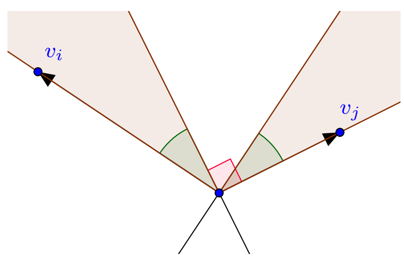


Figure 3. Illustration of vector projection

Then, let $r$ be the projection of $g$ to the plane. So then we have

$$\begin{cases} g^T v_i = r^T v_i \\ g^T v_j = r^T v_j \end{cases}$$

from which it is easy to see that the direction of $r$ is distributed identically to $g$ (ie. $\frac{r}{\|r\|}$ is distributed uniformly in the plane. So, consulting the figure, $r$ must land in the shaded region, which each of which are slivers of angle $\theta_{ij}$. Thus, the probability of landing in a shaded region is $\frac{2\theta_{ij}}{2\pi} = \frac{\theta ij}{\pi}$. Now, we can finish up our analysis:

$$\begin{aligned}
\mathbf{E}[\text{no. edges cut}] &= \sum_{(i,j) \in E} \frac{\theta_{ij}}{\pi} \\
\text{SDP} &= \sum_{(i,j) \in E} \frac{1}{2}(1 - \cos\theta_{ij}) \\
\frac{\mathbf{E}[\text{no. edges cut}]}{\text{SDP}} &\geq \min_{\theta_{ij}} \frac{\theta_{ij}/\pi}{\frac{1}{2}(1 - \cos\theta_{ij})} \\
&\geq \min_{\theta \in [0,\pi]} \frac{\theta/\pi}{\frac{1}{2}(1 - \cos\theta)}
\end{aligned}$$

and this happens to be

$$= \alpha_{GW}$$

which was the value we were interested in from the beginning.

2.2. **Can we do better?** [Kar99] showed that there are graphs for which the SDP approaches the Max-Cut asymptotically and so approach the value $\alpha_{GW}$. Additionally, [FS02] gave individual examples of graphs whose SDP solutions are very close to $\alpha_{GW}$.

You actually don't lose too much by looking at one edge. If you let $\theta^*$ be the minimizing term in the final inequality above that produces $\alpha GW$, then all one need do is search for graphs that have many angles close to $\theta^*$. The algorithm simply cannot do better on these types of graphs.

Some stronger results include [Has01], which shows that it is NP-Hard to approximate Max-Cut better than $\frac{16}{17} + \epsilon \approx 0.94$. [KKMO07] shows that any approximation better than GW for Max-Cut disproves the unique games conjecture (which most assume to be false, but has not yet been decided).

The main takeaway from this is that a random cut does pretty well even though it's really simple, and **it's still the best we have**.

## 3. Balanced Separator Problem

Now, we turn our attention to a different problem with an SDP solution. What if we want to do divide and conquer on graphs?

Ideally, each part in a decomposition would have roughly $\frac{1}{3}$ of the graph (and be roughly even). But it actually turns out that the size of these "halves" as cuts matters. For example, an electrical engineer laying out a PCB would want to have few edge crossings in each cut, and could conceivably use a divide and conquer approach for laying it out.

So our question is this: Given $G = (V, E)$, we want to find $S \subseteq V$ such that $S$ is $c$-balanced, ie. $|S|, |V \setminus S| \geq c|V|$, and which minimizes the edges cut. For the sake of notation, we will write:

$$|E(S, V \setminus S)| = |\left\{(i,j) \in E | i \in S, j \in V \setminus S\right\}|$$

This problem has applications in graph clustering and layout, as well as leading to theoretical results. It's also closely related to finding the conductance and mixing times of random walks.

As with all good problems, this problem is NP-Hard, so we'll try to approximate it.

$$\text{SDP} \begin{cases} \min \sum\limits_{(i,j)\in E} \frac{1}{4} \left\|v_i - v_j\right\|^2 \\ \forall i, \left\|v_i\right\|^2 = 1 \\ \sum\limits_{i,j \in V} \frac{1}{4} \left\|v_i - v_j\right\|^2 \geq c(1-c)n^2 \end{cases}$$

and our intended solution would be to take $S$ such that $v_i$ is the standard basis vector $e_i$ in the positive direction if it is contained in $S$, and in the negative direction otherwise. We can relax this,

$$\sum_{(i,j)\in E} \left\|v_i - v_j\right\|^2 = 2\alpha n(1-\alpha)n4 \geq 8c(1-c)n^2$$

but what we proved before (for GW) doesn't work for this, as it goes in the opposite direction. We claim that, in fact this SDP $\leq$ Bal-Sep optimum.

**Remark 3.1.** *A good exercise would to be to prove that for the n-vertex cycle, $C_n$, $\frac{\textit{Bal-Sep}}{\textit{SDP}} = \Omega(|V|) = \Omega(n)$.*

So what can be done when your relaxations fail?

3.1. **Strengthening Relaxations.** We can add any "vector constraint" we want as long as it is satisfied by the intended solution. So, because it is true that

$$\forall x_i, x_j, x_k \in \{\pm 1\} \quad (x_i - x_j)^2 + (x_j - x_k)^2 \geq (x_i - x_k)^2$$

we can add the following constraints to the Balanced Separator SDP:

$$\forall i, j, k : \left\|v_i - v_j\right\|^2 + \left\|v_j - v_k\right\|^2 \geq \left\|v_i - v_k\right\|^2 .$$

These inequalities are called as triangle inequalities. This leaves us with our final theorem for the day:

**Theorem 3.2** (Arora-Rao-Vazirani [ARV09])**.** *Using the above relaxation for c-Balanced Separator, we can find a $\frac{c}{100}$-balanced cut that cuts at most $O(\sqrt{\log n} \cdot \mathsf{OPT})$ edges, where $\mathsf{OPT}$ is the cost of the best c-balanced cut and $n = |V|$.*

## References


[ARV09] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, April 2009.

[FS02] Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for max cut. *Random Structures Algorithms*, 20(3):403–440, 2002.

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[Has01] Johan Hastad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.

[Kar99] Howard Karloff. How good is the goemans–williamson max cut algorithm? *SIAM Journal on Computing*, 29(1):336–350, 1999.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, April 2007.