E0 249: Approximation Algorithms	February 9, 2022

Week-6
Lecture 2: Max Cut

Instructor: Anand Louis *Scribe: V S S Aniketh*

1 Introduction

In this lecture, we will look at solving the Max-cut problem. First we will look into the basic formulation and a randomised , de-randomised algorithm to solve the problem using this formulation. Both the algorithms produce a 0.5- approximation. Then, we will look into Quadratic, semi-definite, and vector program based formulations. Subsequently, we will look into Max-Cut SDP rounding method to solve the problem using vector program based formulation which produces a 0.878- approximation to the Max-cut problem.

2 Max-Cut Formulations

Definition 1. ***(Max-Cut)***
Given a weighted graph $G = (V, E, w)$, the max-cut is given as $\text{argmax}_{S \subset V} w(E(S, V - S))$

2.1 Basic Formulation

An integer programming formulation of Max-cut is given as:

$$\max \sum_{(i,j)\in E} w_{(u,v)} |x_u - x_v| \tag{IP-1}$$
$$st.\ x_i \in \{0, 1\}, \forall i \in V$$

2.1.1 A Randomised Algorithm

For every $i \in V$, randomly assign it to $\{0, 1\}$, with probability $= \frac{1}{2}$.
An edge (i, j) would cross the cut (iff) $(x_i = 1 \wedge x_j = 0) \vee (x_i = 0 \wedge x_j = 1)$ and the probabilty of this event is $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$.
The capacity of edges crossing the cut is:

$$\sum_{(i,j)\in E} w_{(i,j)} \times I_{(i,j)\text{crosses cut}}$$

where $I_{(i,j)\text{crosses cut}}$ is the indicator r.v denoting the event that the edge (i, j) crosses the cut.
Hence the expected capacity of edges crossing the cut is:

$$\begin{aligned}
E(\sum_{(i,j)\in E} w_{(i,j)} \times I_{(i,j)\text{crosses cut}}) =& (\sum_{(i,j)\in E} w_{(i,j)} \times Pr(I_{(i,j)\text{crosses cut}})) \\
=& (\sum_{(i,j)\in E} w_{(i,j)}) \times \frac{1}{2} \\
=& \frac{\sum_{(i,j)\in E} w_{(i,j)}}{2} \\
\geq& \frac{|\text{Max-cut}|}{2}
\end{aligned}$$

2.1.2 A De-randomised Algorithm

The Algorithm above can be de-randomised by the method in which by induction, at every stage we make a choice for a variable st. the conditional expectation with this choice, along with the choices made up to previous stage, is at least the conditional expectation obtained up to the previous stage.

This, in the case of Max-Cut problem translates to the following Algorithm:
Starting from $i = 1$ up to $|V|$, place i on that side of the cut where less no. of neighbors of i, among $\{1, 2, ..i - 1\}$, are present.
This de-randomised Algorithm has an approximation Ratio = 0.5.

2.2 Quadratic, Semi-definite, Vector program based formulations

Max-Cut problem can be re-formulated in the following way:

$$\max \sum_{(i,j)\in E} w_{(i,j)}(x_i - x_j)^2 \tag{IP-2}$$
$$st.\ x_i \in \{0, 1\}, \forall i \in V$$

Another IP for the Max-Cut problem is:

$$\max \frac{1}{4} \times \sum_{(i,j)\in E} w_{(i,j)}(x_i - x_j)^2 \tag{IP-3}$$
$$st.\ x_i \in \{-1, 1\}, \forall i \in V$$

Consider vector $x = [x_i], i \in V$. Introduce a variable X_{ij} to denote $x_i x_j$. Consider $X = [X_{ij}], (i, j) \in V \times V$. Then,

$$X = xx^T$$

Substituting this into (IP-3) and setting X_{ii} to 1 and using the fact that $X \succcurlyeq 0$, we get the following semi-definite program:

$$\max \sum_{i,j\in V} w_{ij} \frac{1}{4}(X_{ii} + X_{jj} - 2X_{ij}) \tag{SDP}$$
$$st.\ X_{ii} = 1, \forall i \in V$$
$$X \succcurlyeq 0$$

Any feasible solution of (IP-3) has a feasible equivalent(in terms of cost) solution SDP.

Lemma 2. *For any $X \succcurlyeq 0, X \in \mathbb{R}^{n\times n}$ there exist vectors $v_1, v_2, ...v_n \in \mathbb{R}^n$, such that $X_{ij} = \langle v_i, v_j\rangle, \forall i, j \in [n]$*

Proof. It follows from spectral decomposition of X(being symmetric and positive semidefinite) that:

$$X = PDP^T = (PD^{\frac{1}{2}}P^T) \times (PD^{\frac{1}{2}}P^T)$$

where, D is a diagonal matrix with all entries(eigenvalues)≥ 0. P has orthonormal eigenvectors.

Setting $PD^{\frac{1}{2}}P^T = V$, where $V \in \mathbb{R}^{n\times n}$ and $V = [v_1\ v_2\ ...\ v_n]$, where $v_i \in \mathbb{R}^n$, we get $X_{ij} = \langle v_i, v_j\rangle \forall i, j \in [n]$ □

Using the above result, we get the following vector program based formulation which has the same set of feasible solutions as that of (SDP):

$$\max \sum_{i,j\in V} w_{ij} \frac{1}{4} \|v_i - v_j\|^2 \tag{VP-1}$$
$$st.\ \|v_i\|^2 = 1, \forall i \in V$$
$$v_i \in \mathbb{R}^n, \forall i \in V$$

3 Max-cut SDP rounding

3.1 Algorithm

An SDP based rounding method for Max-Cut is as below:

- Solve the Vector program (VP-1) for getting the vectors $v_i, i \in V$
- Sample a random vector $g \sim \mathcal{N}(0,1)^n$
- Output $S_g = \{i \in V : \langle v_i, g\rangle > 0\}$

3.2 Analysis

3.2.1 Algorithm is 0.878-approximate

Consider an edge $\{i,j\} \in E$

$$Pr(\{i,j\} \text{ is cut }) = Pr(\langle v_i, g\rangle > 0 \ and \ \langle v_j, g\rangle < 0) + Pr(\langle v_i, g\rangle < 0 \ and \ \langle v_j, g\rangle > 0) = \frac{\theta_{ij}}{\pi}$$

SDP contribution of $\{i,j\} \in E$ is:

$$\frac{1}{4}\left\|v_i - v_j\right\|^2 = \frac{1}{4}(\|v_i\|^2 + \|v_j\|^2 - 2\langle v_i, v_j\rangle) = \frac{1}{4}(1 + 1 - 2\cos\theta_{ij} = \frac{1}{2}(1 - \cos\theta_{ij})$$

Let $\alpha_{GW} = \inf_{\theta\in[0,\pi]} \frac{\frac{\theta}{\pi}}{\frac{1}{2}(1-\cos\theta}$). Then,

$$Pr[i,j \ is \ cut] = \frac{\theta_{ij}}{\pi} \geq \alpha_{GW}.\frac{1}{2}(1 - \cos\theta_{ij}) = \alpha_{GW}.\frac{1}{4}\left\|v_i - v_j\right\|^2$$

$$E[cut] = \sum_{\{i,j\}\in E} w_{ij} Pr[i,j \ is \ cut] \geq \alpha_{GW} \sum_{\{i,j\}\in E} .w_{ij}\frac{1}{4}\left\|v_i - v_j\right\|^2 = \alpha_{GW}|OPT_{SDP}|$$

where $|OPT_{SDP}|$ is the value of optimal solution obtained for the SDP.
Also $|OPT_{SDP}| \geq |\text{Max-Cut}|$. Hence the SDP rounding Algorithm for Max-Cut problem is α_{GW}-approximate.

$\boxed{\alpha_{GW} \approx 0.878}$

3.3 Lower bound on Fraction of edges cut on Average

Let $\frac{1}{W}\sum_{i,j\in V} w_{ij}\frac{1}{4}\left\|v_i - v_j\right\|^2 = 1 - \varepsilon$, where $W = \sum_{i,j\in V} w_{ij}$. Let $\frac{1}{4}\left\|v_i - v_j\right\|^2 = 1 - \varepsilon_{ij}$. Then,

$$\sum_{i,j\in V} \frac{w_{ij}}{W} \times \frac{1}{4}\left\|v_i - v_j\right\|^2 = 1 - \varepsilon$$

Lemma 3. *For a fixed $i,j \in V$, $Pr[(i,j)$ is not cut by Algorithm$] = O(\sqrt{\varepsilon_{ij}})$*

Proof.

$$Pr[(i,j) \text{ is not cut by Algorithm}] = 1 - \frac{\theta_{ij}}{\pi} = \frac{\pi - \theta_{ij}}{\pi}$$

$$\varepsilon_{ij} = 1 - \frac{1}{4}\left\|v_i - v_j\right\|^2 = 1 - (\frac{1 - \cos\theta_{ij}}{2}) = \frac{1 + \cos\theta_{ij}}{2} = \cos^2(\frac{\theta_{ij}}{2}) = \sin^2(\frac{\pi - \theta_{ij}}{2})$$

For $x \in [0, \frac{\pi}{2}], x = O(\sin x)$. Therefore,

$$Pr[(i,j) \text{ is not cut by Algorithm}] = \frac{\pi - \theta_{ij}}{2} = O(|\sin(\frac{\pi - \theta_{ij}}{2})|) = O(\sqrt{\varepsilon_{ij}})$$

□

Theorem 4. *Algorithm cuts at least $1 - O(\sqrt{\varepsilon})$ fraction of edges.*

Proof. Let the fraction of edges not cut be denoted by r.v F. Let I_E denote indicator rv for an event E. Then,

$$
\begin{aligned}
F &= \sum_{i,j \in V} \frac{w_{ij}}{W} \times I_{(i,j) \text{ is not cut by Algorithm}} \\
E(F) &= \sum_{i,j \in V} \frac{w_{ij}}{W} \times Pr[(i,j) \text{ is not cut by Algorithm}] \\
&= \sum_{i,j \in V} \frac{w_{ij}}{W} \times O(\sqrt{\varepsilon_{ij}})
\end{aligned}
$$

We have the following equality:

$$
\begin{aligned}
\sum_{i,j \in V} \frac{w_{ij}}{W} \times (1 - \varepsilon_{ij}) = 1 - \varepsilon \\
\sum_{i,j \in V} \frac{w_{ij}}{W} \times \varepsilon_{ij} = \varepsilon
\end{aligned}
$$

Further, observing that $f(x) = \sqrt{x}$ is a concave function, on applying Jensen's inequality for concave functions, to the last expression, we get:

$$
E(F) = \sum_{i,j \in V} \frac{w_{ij}}{W} \times O(\sqrt{\varepsilon_{ij}}) \leq O(\sqrt{\sum_{i,j \in V} \frac{w_{ij}}{W} \times \varepsilon_{ij}}) = O(\sqrt{\varepsilon})
$$

Hence, the expected fraction of edges cut by the Algorithm is $1 - E(F) = 1 - O(\sqrt{\varepsilon})$ □

4 $\{0, 1\}$-SDP

- In this SDP version, for each $i \in V$, introduce vectors $v_i^{(0)}, v_i^{(1)}$ to indicate 0 or 1 label.
- Since i can receive exactly one of these labels, add constraint $\langle v_i^{(0)}, v_i^{(1)} \rangle = 0$ and $\left\| v_i^{(0)} \right\|^2 + \left\| v_i^{(1)} \right\|^2 = 1$.
- For each $i, j \in V$, add constraints $\left\| v_i^{(0)} - v_j^{(0)} \right\|^2 = \left\| v_i^{(1)} - v_j^{(1)} \right\|^2 = \langle v_i^{(0)}, v_j^{(1)} \rangle + \langle v_i^{(1)}, v_j^{(0)} \rangle$

With the above constraints we consider the following vector program(VP-2) which corresponds to SDP-1:

$$
\begin{aligned}
\max \quad & \sum_{i,j \in V} w_{ij} \left\| v_i^{(0)} - v_j^{(0)} \right\|^2 \qquad \text{(VP-2)} \\
st. \quad & \langle v_i^{(0)}, v_i^{(1)} \rangle = 0 \\
& \left\| v_i^{(0)} \right\|^2 + \left\| v_i^{(1)} \right\|^2 = 1, \forall i \in V \\
& \left\| v_i^{(0)} - v_j^{(0)} \right\|^2 = \left\| v_i^{(1)} - v_j^{(1)} \right\|^2 \\
& = \langle v_i^{(0)}, v_j^{(1)} \rangle + \langle v_i^{(1)}, v_j^{(0)} \rangle, \forall i, j \in V
\end{aligned}
$$

Under the above constraints, the following claim holds

Claim 5. *For all $i, j \in V$, $v_i^{(0)} + v_i^{(1)} = v_j^{(0)} + v_j^{(1)}$.*

Proof.

$$\begin{aligned}
\langle v_i^{(0)} + v_i^{(1)}, v_j^{(0)} + v_j^{(1)} \rangle &= \langle v_i^{(0)}, v_j^{(0)} \rangle + \langle v_i^{(1)}, v_j^{(1)} \rangle + (\langle v_i^{(0)}, v_j^{(1)} \rangle + \langle v_i^{(1)}, v_j^{(0)} \rangle) \\
&= \langle v_i^{(0)}, v_j^{(0)} \rangle + \langle v_i^{(1)}, v_j^{(1)} \rangle + \frac{1}{2}(\left\| v_i^{(0)} - v_j^{(0)} \right\|^2 + \left\| v_i^{(1)} - v_j^{(1)} \right\|^2) \\
&= \frac{1}{2}(\left\| v_i^{(0)} \right\|^2 + \left\| v_j^{(0)} \right\|^2 + \left\| v_i^{(1)} \right\|^2 + \left\| v_j^{(1)} \right\|^2) \\
&= 1
\end{aligned}$$

Therefore,

$$\left\| (v_i^{(0)} + v_i^{(1)}) - (v_j^{(0)} + v_j^{(1)}) \right\|^2 = \left\| v_i^{(0)} + v_i^{(1)} \right\|^2 + \left\| v_j^{(0)} + v_j^{(1)} \right\|^2 - 2\langle v_i^{(0)} + v_i^{(1)}, v_j^{(0)} + v_j^{(1)} \rangle = 0$$

Hence, $v_i^{(0)} + v_i^{(1)} = v_j^{(0)} + v_j^{(1)}$. □

Let $v_0 = v_i^{(0)} + v_i^{(1)}$, and define $v_i = 2v_i^{(0)} - v_0$.

Lemma 6. $(v_i)_{i \in V}$ *form a feasible solution to VP-1 with equal cost as that in VP-2*

Proof.

$$\begin{aligned}
\|v_i\|^2 = \left\| 2v_i^{(0)} - v_0 \right\| = 4 \left\| v_i^{(0)} \right\|^2 + \|v_0\|^2 - 4\langle v_i^{(0)}, v_0 \rangle = 1 \\
\frac{1}{4} \|v_i - v_j\|^2 = \frac{1}{4} \left\| (2v_i^{(0)} - v_0) - (2v_j^{(0)} - v_0) \right\|^2 = \left\| v_i^{(0)} - v_j^{(0)} \right\|^2
\end{aligned}$$

□