15-750: Graduate Algorithms April 17, 2017

Lecture 33: Sparsest Cut 1

Lecturer: David Witmer *Scribe(s): Mu-Chu Lee*

1 Min s-t Cut

Recall that in the lecture introducing flows, we defined the **Min s-t Cut** problem:

> Given graph $G = (V, E)$, edge capacities c_e for $e \in E$, source vertex s and sink vertex t, we want to partition V into two sets, S and $\bar{S}$, where $\bar{S} = V \backslash S$, with $s \in S$, $t \in \bar{S}$ such that
>
> $$\sum_{\substack{(u,v)\in E\\ u\in S\\ v\notin S}} c_{uv} \text{ is minimized.}$$

2 Sparsest Cut

Now, to generalize the min s-t cut problem, let's define the *sparsest cut* problem. We are given a capacity graph $G = (V, E_G)$ with capacities c_e for $e \in E_G$ and a demand graph $H = (V, E_H)$ on the same set of vertices with capacities D_e for $e \in E_H$. The *sparsity* of a cut $S \subseteq V$ is then

$$\Phi(S) = \frac{\sum_{\substack{(u,v)\in E_G\\ u\in S, v\notin S}} c_{uv}}{\sum_{\substack{(u,v)\in E_H\\ u\in S, v\notin S}} D_{uv}}.$$

We wish to find the cut S minimizing $\Phi(S)$.

For example, in Fig.1, we have $\Phi(S_{blue}) = \frac{2}{3}$ and $\Phi(S_{brown}) = 1$.

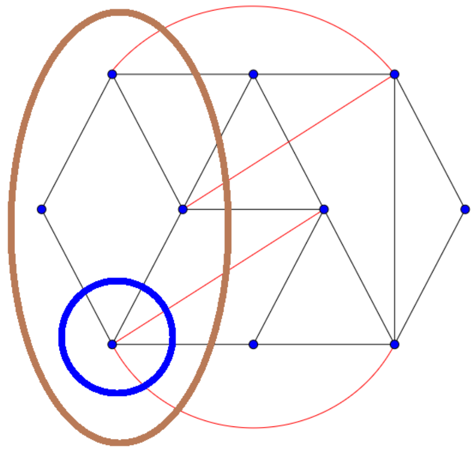

Figure 1: Red edges indicate demand, and black indicate capacity. Two cuts are shown by the brown and blue circle.

It is trivial to see that

1. If the only D_e existing is D_{st}, this is exactly the min s-t cut problem.
2. If D_e is complete (uniform sparsest cut), the problem has the form

$$\min_{S \subseteq V} \frac{\sum_{\substack{(u,v) \in E \\ u \in S, v \notin S}} c_{uv}}{|S||\bar{S}|}.$$

   In this uniform case, the intuition is to cut as little edge as possible, while keeping the two sets as balance as possible. For example, in Fig.2, we can see that the sparsest cut (by the red line) has $\Phi(S_{red}) = \frac{2}{4 \times 5} = \frac{1}{10}$, while the green line gives a bigger $\Phi(S_{green}) = \frac{1}{1 \times 8} = \frac{1}{8}$.

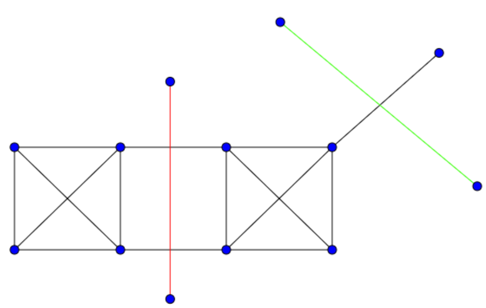

Figure 2: Two cuts are shown by the red line and the green line.

2.1 Motivations

There are some motivations behind sparsest cut. Take the uniform sparsest cut with unit demand between every pair of vertices (i.e. $D_{uv} = 1, \forall u \neq v \in V$). If we have sparse cuts, we are able to find good separators that enable us to do:

- Clustering: Edge capacities as a similarity measure (higher for more similar elements)
- Divide-and conquer algorithms: Recurse on 2 evenly-sized subproblems without cutting too many edges.

However, we know that this problem is *NP-Hard*. We can ask instead if we could find an **α-approximate sparsest cut**.

2.2 α-approximation Algorithm

An algorithm is an **α-approximation** for a minimization problem with optimum value OPT if it returns a value $\leq \alpha \cdot OPT$.
Given that a sparsest cut is a *NP-Hard* problem, we seek a poly-time α-approximation algorithm.

Theorem 2.1. *Sparsest cut has an poly-time $O(\log n)$-approximation algorithm.*

We obtain an $O(\log n)$-approximation algorithm for sparsest cut using a linear programming relaxation. We will lay the groundwork for stating this LP relaxation by giving an alternative formulation of sparsest cut in terms of optimizing over ℓ_1-metrics.

2.3 Metric Space

A metric space on a set X is defined as a distance measure $d : V \times V \to \mathbb{R}^{\geq 0}$ such that the following three criteria holds.

1. $d(x, y) = 0 \iff x = y$ (not necessary for **pseudometric**)
2. $d(x, y) = d(y, x)$
3. $d(x, z) \leq d(x, y) + d(y, z), \quad \forall x, y, z \in X$ (Triangle inequality).

We call (X, d) a **metric space**.

Example 2.2. A classic example is $\mathbb{R}^m$ equipped with ℓ_p-norms.

$$d(x, y) = (\sum_{i=1}^{m} |x_i - y_i|^p)^{\frac{1}{p}})$$

Example 2.3. Given a cut $S \subseteq V$, we associate a δ_S with this space such that:

$$\delta_S(x, y) = \begin{cases} 0, & \text{if } x, y \in S \text{ or } x, y \in \bar{S} \\ 1, & \text{otherwise} \end{cases}$$

We could prove this is a pseudometric easily. Clearly, $\delta_S(x, y) = \delta_S(y, x)$. For the triangle inequality, we have $d(x, z) \leq d(x, y) + d(y, z)$. It is trivial if $d(x, z) = 0$. If $d(x, z) = 1$, we must have one of the following:

- $x \in S, z \in \bar{S}$
- $x \in \bar{S}, z \in S$

Since y must exist in S or $\bar{S}$, so either $d(x, y) = 1$ or $d(y, z) = 1$, therefore RHS≥ 1.

2.4 Sparsest Cut and Metrics

Say that $c_e = 0$ if $e \notin E_G$ and $D_e = 0$ if $e \notin E_H$. Let's restate sparsest cut as

$$\min_{\text{cut metrics } \delta_S} \frac{\sum\limits_{u,v \in V} c_{uv} \delta_S(u,v)}{\sum\limits_{u,v \in V} D_{uv} \delta_S(u,v)} \tag{1}$$

and define a cut cone as combination of the metrics.

Definition 2.4. Cut cone:

$$CUT_n = \{d | d = \sum_{S \subseteq V} \alpha_S \delta_S, \alpha_S \geq 0 \quad \forall S\}$$

We claim that the following form associated with the cut cone is equivalent to eq.(1).

$$\min_{d \in CUT_n} \frac{\sum\limits_{u,v \in V} c_{uv} d(u,v)}{\sum\limits_{u,v \in V} D_{uv} d(u,v)} \tag{2}$$

Lemma 2.5. *Eq.(1) associated with cut metrics is equivalent to eq.(2) associated with cut cones.*

To prove that eq.(1) and eq.(2) is equivalent, we would need to show:

1. eq.(1) does no better than eq.(2)
2. eq.(2) does no better than eq.(1)

The first is trivial, since any cut metric $\delta_S \in CUT_n$, because we could just set the corresponding $\alpha_S = 1$, and the others 0.
To prove the second, we need to use the fact:

$$a_i, b_i \geq 0, \forall i \text{ then, } \exists i, \text{ s.t. } \frac{a_1 + a_2 + \cdots + a_t}{b_1 + b_2 + \cdots + b_t} \geq \frac{a_i}{b_i}.$$

We could verify this fact by some simple arithemic operations.
Now, let $d = \sum\limits_S \alpha_S \delta_S$. We would have:

$$\begin{aligned}
\frac{\sum\limits_{u,v \in V} c_{uv} d(u,v)}{\sum\limits_{u,v \in V} D_{uv} d(u,v)} &= \frac{\sum\limits_{u,v \in V} c_{uv} \sum\limits_S \alpha_S \delta_S}{\sum\limits_{u,v \in V} D_{uv} \sum\limits_S \alpha_S \delta_S} \\
&= \frac{\sum\limits_S \alpha_S \sum\limits_{(u,v)} c_{uv} \delta_S}{\sum\limits_S \alpha_S \sum\limits_{(u,v)} D_{uv} \delta_S} \\
&\geq \frac{\alpha_{S'} \sum\limits_{(u,v)} c_{uv} \delta_{S'}}{\alpha_{S'} \sum\limits_{(u,v)} D_{uv} \delta_{S'}}
\end{aligned}$$

The last comes from the fact, and since α can be arbitrary value ≥ 0, the second is proved.

2.5 CUT_n versus ℓ_1-metric

Recall that in example 2.2, we've showed the definition of ℓ_p-metrics. Here, we discuss a specific case, the ℓ_1-metrics. A metric d on n points X is an ℓ_1 metric if there exists a map $f : X \to \mathbb{R}^t$ for some t with $d(x, y) = ||f(x) - f(y)||_1 \quad \forall x, y \in X$.

Lemma 2.6. *$d \in CUT_n \iff d$ is an ℓ_1 metric.*

Any n-point metric can be associated with a vector in $\mathbb{R}^{\binom{n}{2}}$, each coordinate corresponds to some pair of points in V. We are now ready to prove $d \in CUT_n \iff d$ is an ℓ_1 metric.

1. $d \in CUT_n \Rightarrow d$ is an ℓ_1 metric.

   We want to have a mapping $f : V \to \mathbb{R}^m$ s.t. $d(u, v) = ||f(u) - f(v)||_1$.
   We will have one coordinate for every $S \subseteq V (m = 2^n)$.
   So we have, $f(v) = (f_{S_1}(v), f_{S_2}(v), \ldots, f_{S_{2^n}}(v))$.
   Where
   $$f_S(v) = \begin{cases} 0, & \text{if } v \in S \\ \alpha_S, & \text{otherwise.} \end{cases}$$
   We then get, $||f(u) - f(v)||_1 = \sum_S |f_S(u) - f_S(v)| = \sum_S \alpha_S \delta_S(u, v)$.

2. $d \in CUT_n \Leftarrow d$ is an ℓ_1 metric.

   Consider a set of n-point $(x_1, x_2, \ldots, x_n)$ in $\mathbb{R}^m$. We want to find α_S's such that:
   $$\sum_{i=1}^{m} |x_j(i) - x_k(i)| = \sum_S \alpha_S \delta_S(x_j, x_k)$$
   Suppose we take one dimension d, and sort the points in increasing value along d, we would get $v_1, v_2, \ldots, v_l$ distinct values. Define cut metric $S_i = \{x | x(d) \leq v_{i+1}\}$, and let $\alpha_i = v_{i+1} - v_i$, we have that $|x_j(d) - x_k(d)| = \sum_{i=1}^{l} \alpha_i \delta_{S_i}(x_j(d), x_k(d))$. We can construct S for all dimensions, and have a metric in CUT_n for every n-point metric in ℓ_1.

This lemma implies that (2) is equivalent to

$$\min_{\ell_1 \text{ metrics } d} \frac{\sum_{u,v \in V} c_{uv} d(u, v)}{\sum_{u,v \in V} D_{uv} d(u, v)}.$$