The Sparsest Cut[1]

- The next cut problem we will look at is the *sparsest cut* problem. Once again, the input is an undirected graph $G = (V, E)$ with non-negative costs $c(e)$ on edges. The objective is to find a subset of vertices $S \subseteq F$ such that the ***ratio*** $\Phi(S) := \frac{\sum_{e\in\partial S} c(e)}{|S|\cdot|V\setminus S|}$ is minimized. $\Phi(S)$ is called the sparsity of cut, and the above problem is to find the *sparsest cut* in G.

- ***Linear Programming Relaxation.*** As in the previous cut problems, we have "distance variables" d_{uv} which satisfy triangle inequality. The idea is for two vertices u and v in *different* parts (that is one in S and the other in $V \setminus S$), then we want $d_{uv} = 1$, and the others have $d_{uv} = 0$. With this semantic, note that (a) $\sum_{e\in\partial S} c(e) = \sum_{e=(u,v)\in R} c(e)d_{uv}$, and (b) $|S| \cdot |V \setminus S| = \sum_{u,v\in V} d_{uv}$. To see part (b), note that the RHS is counting the number of pairs $\{u, v\}$ with $u \in S$ and $v \notin S$.

  So, the LP relaxation would like to find d_{uv}'s for every pair satisfying triangle inequality, but the objective seems to be a *ratio* of two linear functions. How do we fix that? The main observation is that the triangle inequality (and the non-negativity inequality) is "scale-free", that is, the "b-side" is 0 in the LP. And therefore, multiplying the variables by any parameter doesn't change feasibility. Once we have that, then the ratio of two linear functions can be handled by simply asserting that the denominator equals 1, and minimizing the numerator. This is the LP for sparsest cut.

$$\mathsf{lp} := \min \sum_{e=(u,v)\in E} c(e)d_{uv} \qquad \text{(Sparsest Cut LP)}$$

$$d_{uw} \leq d_{uv} + d_{vw}, \quad \forall i \in F,\ \forall\{u, v, w\} \subseteq V \qquad (1)$$

$$d_{vv} = 0, \quad \forall v \in V \qquad (2)$$

$$\sum_{u\in V}\sum_{v\in V} d_{uv} = 1$$

  We now describe *two* algorithms for the sparsest cut problem. The first algorithm is similar to the *region growing* algorithm for multicut we saw in the previous lecture. This algorithm gives an $O(\log n)$-approximation **unless** a certain condition occurs. However, we show a second algorithm which, if that condition occurs, in fact gives an $O(1)$-approximation. Let's begin with region growing.

- ***Low Diameter Decomposition Algorithm.*** We begin with a lemma akin to the region growing lemma from the multicut lecture. It can be proved similarly and we defer the proof to the very end (the reader who has read the multicut notes may try it as an exercise). To state the lemma, we need the notion of the diameter of a subset $S \subseteq V$ given the "distances" d_{uv}. For any subset $S \subseteq V$, define $\mathsf{diam}(S) := \max_{u,v\in S} d_{uv}$.

  Lemma 1 (Low Diameter Decomposition)**.** Suppose we are given any undirected graph $G = (V, E)$ and a solution d_{uv} to (Sparsest Cut LP) with objective value lp. There is an efficient algorithm called LOW DIAMETER DECOMPOSITION which takes input $R > 0$ and finds a partition

[1] *Lecture notes by Deeparnab Chakrabarty. Last modified : 19th Feb, 2022*
These have not gone through scrutiny and may contain errors. If you find any, or have any other comments, please email me at deeparnab@dartmouth.edu. Highly appreciated!

$\Pi := (S_1, \ldots, S_k)$ of V such that

a. $\mathsf{diam}(S_i) \leq R$, for all $S_i \in \Pi$

b. $\sum_{e \in E(\Pi)} c(e) \leq \frac{4\ln(2n)}{R} \cdot \mathsf{lp} = O(\log n) \cdot \frac{\mathsf{lp}}{R}$

We can use the diameter decomposition algorithm to obtain an approximation of sparsest cut as follows. Obtain the partition $\Pi = (S_1, \ldots, S_k)$ with parameter $R = \frac{1}{n^2}$. If there exists some S_i with $|S_i| > \frac{n}{3}$, then abort and we move to the second algorithm for sparsest cut, and in fact in that case, as you will see, we would obtain a $O(1)$-approximation. Otherwise, all $|S_i| \leq n/3$ and therefore, arbitrarily picking sets till we cross $n/3$ would give a set T with $|T|$ and $|V \setminus T|$ both $\Theta(n)$. This is the set we return.

```
1: procedure LDD ALGORITHM(G = (V, E), c(e) ≥ 0 on edges):
2:     Solve (Sparsest Cut LP) to obtain d_uv's.
3:     Run LOW DIAMETER DECOMPOSITION with R = 1/n^2 to obtain Π = (S_1, ..., S_k).
4:     if |S_i| > n/3 for any i then:
5:         Abort, and run SWEEPCUT(S_i).
6:     else:
7:         Pick the smallest ℓ such that Σ_{i=1}^{ℓ} |S_i| > n/3.
8:         return T ← ⋃_{i=1}^{ℓ} S_i.
```

Theorem 1. If the above algorithm reaches Line 6 and returns a set T, then $\Phi(T) \leq O(\log n)\mathsf{lp}$

Proof. First we observe that $n/3 < |T| \leq 2n/3$. The first inequality is by design, and the latter is because $|S_\ell| \leq n/3$ and $\sum_{i=1}^{\ell-1} |S_i| \leq n/3$. Therefore, $|V \setminus T| \geq n/3$, and in turn, $|T| \cdot |V \setminus T| \geq \frac{n^2}{9}$. The second thing we observe is that $\sum_{e \in \partial T} c(e) \leq \sum_{e \in E(\Pi)} c(e)$ since $\partial T \subseteq E(\Pi)$. By Lemma 1(b), the RHS is $\leq O\left(n^2 \log n\right) \cdot \mathsf{lp}$. Therefore, $\Phi(T) = \frac{\sum_{e \in \partial T} c(e)}{|T| \cdot |V \setminus T|} \leq O(\log n)\mathsf{lp}$ □

- ***The Sweep Cut Algorithm.*** The LDD ALGORITHM aborted if it discovered some S_i with $|S_i| > \frac{n}{3}$ and $\mathsf{diam}(S_i) \leq \frac{1}{n^2}$. This seems to suggest we have a "lot" of vertices clustered around a "very small" region. In that case, we can just use a sweeping cut algorithm we have been using for all the other cut problems. Before we describe the algorithm, let's set a notation: for any subset T and any vertex u, we define $d(T, u) := \min_{v \in T} d_{vu}$. Note that for $u \in T$, $d(T, u) = 0$.

```
procedure SWEEP CUT(T):
    ▷ We assume G, c, d are given and diam(T) ≤ 1/n^2 and |T| > n/3
    Let's rename the vertices in V \ T as v_1, ..., v_k in increasing order of d(T, v).
    For any 0 ≤ i ≤ k, let T_i := T ∪ {v_1, ..., v_i}; note T_0 = T.
    return T_i with the smallest Φ(T_i).
```

Theorem 2. If $\mathsf{diam}(T) \le \frac{1}{n^2}$ and $|T| > \frac{n}{3}$, then the sparsity of the cut returned by SWEEP CUT is $\le 12 \cdot \mathsf{lp}$.

Proof. We prove this via a probabilistic argument. Define $R := \max_{v \in V \setminus T} d(T, v)$, or using the notation in the algorithm, $R = d(T, v_k)$. Consider the following random set: sample $r \in [0, R)$ and let $S_r := \{v \in V : d(T, v) \le r\}$ be a random set. First note that the support of S_r is $\{T_0, \ldots, T_k\}$. We now claim the following:

$$\frac{\mathbf{Exp}[\sum_{e \in \partial S_r} c(e)]}{\mathbf{Exp}[|S_r||V \setminus S_r|]} \le 12 \cdot \mathsf{lp} \quad \text{(Claim)}$$

For the moment suppose (Claim) is true. Therefore, $\mathbf{Exp}[\sum_{e \in \partial S_r} c(e)] - 12 \cdot \mathsf{lp} \cdot \mathbf{Exp}[|S_r||V \setminus S_r|] \le 0$, or by linearity of expectation

$$\mathbf{Exp}\left[\sum_{e \in \partial S_r} c(e) - 12 \cdot \mathsf{lp} \cdot |S_r| \cdot |V \setminus S_r|\right] \le 0$$

In particular, this means there is some S_r in the support, that is, some T_j for $0 \le j \le k$ such that

$$\sum_{e \in \partial T_j} c(e) - 12 \cdot \mathsf{lp} \cdot |T_j| \cdot |V \setminus T_j| \le 0 \quad \Rightarrow \Phi(T_j) \le 12 \cdot \mathsf{lp}$$

and this proves the theorem. We now prove (Claim)

We first upper bound $\mathbf{Exp}[\sum_{e \in \partial S_r} c(e)]$. Fix an edge $e = (u, v)$ and wlog, assume $d(T, u) \le d(T, v)$. This edge e is present in ∂S_r iff $d(T, u) \le r < d(T, v)$. Now let $z \in T$ be the vertex attaining $d(T, u) = d_{zu}$. Note that

$$d(T, v) \le d_{zv} \le d_{zu} + d_{uv} = d(T, u) + d_{uv}$$

That is, the $d(T, \cdot)$'s also satisfy a form of triangle inequality.

Therefore, the proabability $\mathbf{Pr}[e \in \partial S_r] \le \mathbf{Pr}[r \in [d(T, u), d(T, u) + d_{uv}]] \le \frac{d_{uv}}{R}$. And so,

$$\mathbf{Exp}[\sum_{e \in \partial S_r} c(e)] \le \sum_{e \in E} c(e) \cdot \frac{d_{uv}}{R} = \frac{\mathsf{lp}}{R} \quad (3)$$

Next, we *lower bound* $\mathbf{Exp}[|V \setminus S_r|]$. To this end, let $g(r)$ denote the number of vertices in $v \in V$ with $d(T, v) > r$. Then since r is drawn uniformly in $[0, R)$,

$$\mathbf{Exp}[|V \setminus S_r|] = \frac{1}{R} \int_{r=0}^{R} g(r) dr = \frac{1}{R} \int_{r=0}^{R} \left(\sum_{v \in V} \mathbf{1}_{d(T,v) > r}\right) dr = \frac{1}{R} \cdot \sum_{v \in V} \int_{r=0}^{R} \mathbf{1}_{d(T,v) > r} dr$$

where $\mathbf{1}_{d(T,v) > r}$ is 1 if $d(T, v) > r$ and 0 otherwise. Now observe that $\int_{r=0}^{R} \mathbf{1}_{d(T,v) > r} dr$ is *precisely* $d(T, v)$. Therefore,

$$\mathbf{Exp}[|V \setminus S_r|] = \frac{1}{R} \cdot \sum_{v \in V} d(T, v) \quad (4)$$

We now (finally) use that $\sum_{u,v\in V} d_{uv} = 1$ to show that $\sum_{v\in V} d(T,v)$ is large. First we note, by triangle inequality, that $d_{uv} \leq d(T,u) + \mathsf{diam}(T) + d(T,v)$. To see this, let $d(T,u) = d_{uz}$ and $d(T,v) = d_{vy}$, and use triangle inequality to assert $d_{uv} \leq d_{uz} + d_{zy} + d_{vy}$, and then use the definition of $\mathsf{diam}(T)$. Therefore,

$$1 = \sum_{u,v\in V} d_{uv} \leq \sum_{u,v\in V} \left(d(T,u) + \mathsf{diam}(T) + d(T,v)\right) = \binom{n}{2} \cdot \mathsf{diam}(T) + 2n \sum_{v\in V} d(T,v)$$

Using $\mathsf{diam}(T) \leq \frac{1}{n^2}$, we get $1 \leq \frac{1}{2} + 2n\sum_{v\in V} d(T,v)$, or $\sum_{v\in V} d(T,v) \geq \frac{1}{4n}$.
Substituting in (4), we get $\mathbf{Exp}[|V \setminus S_r|] \geq \frac{1}{4nR}$. Now since $T \subseteq S_r$, we get $|S_r| \geq |T| \geq \frac{n}{3}$. Therefore,

$$\mathbf{Exp}[|S_r| \cdot |V \setminus S_r|] \geq \frac{1}{12R} \tag{5}$$

Combining (3) and (5), we obtain the proof of (Claim). □

- ***Proof of Lemma 1.*** We start with a couple of definitions. Given a parameter $r \in \mathbb{R}$, a *subset* $U \subseteq V$, a vertex $a \in U$ let $S_a(r;U) := \{u \in U : d_{ua} \leq r\}$. Define $\partial S_a(r;U) := \{(u,v) \in E[U] : u \in S_a(r;U), v \notin S_a(r;U)\}$. Let us define $E[S_a(r;U)] = \{(u,v) \in E[U] : u,v \in S_a(r)\}$. Here $E[U]$ are the edges with both endpoints in U. The main claim is the following.

Claim 1. For every subset $U \subseteq V$, every vertex $a \in U$, there exists $r \in [0, R/2]$ such that

$$\sum_{e\in\partial S_a(r;U)} c(e) \leq \frac{2\ln(2n)}{R} \cdot \left(\frac{\mathsf{lp}}{n} + \sum_{(u,v)\in E[S_a(r;U)]} c(u,v)d_{uv} + \sum_{(u,v)\in\partial S_a(r;U)} c(u,v)d_{uv}\right)$$

Furthermore, this r can be found efficiently.

Before proving the claim, let us describe the algorithm assuming the claim.

```
procedure LOW DIAMETER DECOMPOSITION(G, c, d, R):
    Initialize U ← V; Π ← ∅; Ctrs ← ∅.
    while U ≠ ∅ do:
        Select an a ∈ U arbitrarily and add it to Ctrs.
        Find r_a ∈ [0, R/2] as in Claim 1 satisfying the conditions mentioned there.
        Add S_i := S_{r_a}(a; U) to Π
    return Π.
```

We claim that Π satisfies the conditions of Lemma 1. First, $\mathsf{diam}(S_i) \leq R$. This is because for any two $u, v \in S_{r_a}(a)$, we have $d_{uv} \leq d_{ua} + d_{va} \leq 2r_a \leq R$. Next, we note that

$$\begin{aligned} c(E(\Pi)) &= \sum_{a \in \mathsf{Ctrs}} \sum_{e \in \partial S_{r_a}(a;U)} c(e) && (6) \\ &\underbrace{\leq}_{\text{Claim 1}} \frac{2\ln(2n)}{R} \cdot \sum_{a \in \mathsf{Ctrs}} \left(\frac{\mathsf{lp}}{n} + \sum_{(u,v) \in E[S_a(r;U)]} c(u,v) d_{uv} + \sum_{(u,v) \in \partial S_a(r;U)} c(u,v) d_{uv} \right) \\ &\leq \frac{4\ln(2n)}{R} \cdot \mathsf{lp} && (7) \end{aligned}$$

In the equality (6), the set U is the one when a was added to Ctrs, and (7) follows since (a) $|\mathsf{Ctrs}| \leq n$, and (b) every edge $e \in E$ is at most one $E[S_a(r;U)]$ or $\partial S_a(r;U)$.

Proof of Claim 1. Define the "volume" of a ball of radius r around a center $a \in V$.

$$\mathtt{Vol}_a(r;U) := \frac{\mathsf{lp}}{n} + \sum_{(u,v) \in E[S_a(r;U)]} c(u,v) d_{uv} + \sum_{(u,v) \in \partial S_a(r;U), u \in S_a(r;U)} c(u,v) \cdot (r - d_{ua}) \quad (8)$$

Note that for (u,v) participating in the last summation in the definition, we have $d_{va} > r$ and so $r - d_{ua} < d_{va} - d_{ua} \leq d_{uv}$, where the last follows from triangle inequality. And therefore, $\mathtt{Vol}_a(r;U)$ is at most the parenthesized term in the RHS of the claim. So, it suffices to prove that there exists $r \in (0, R/2)$ such that $\sum_{e \in \partial S_a(r;U)} c(e) \leq \frac{2\ln(2n)}{R} \mathtt{Vol}_a(r;U)$. So, for the sake of contradiction this is not the case, and for all r, we have the inequality flipped.

Next, note that $\mathtt{Vol}_a(r;U)$ is a continuous, piece-wise linear function of r, and crucially observe that

$$\frac{d\,\mathtt{Vol}_a(r;U)}{dr} = \sum_{(u,v) \in \partial S_i(r;U)} c(u,v) > \frac{2\ln(2n)}{R} \cdot \mathtt{Vol}_a(r;U) \;\Rightarrow\; \frac{d\mathtt{Vol}_a(r;U)}{\mathtt{Vol}_a(r;U)} > \frac{2\ln(2n)}{R} \cdot dr$$

Therefore, if we integrate with r going from 0 to $R/2$, we get $\ln\left(\frac{\mathtt{Vol}_a(R/2;U)}{\mathtt{Vol}_a(0;U)}\right) > \ln(2n)$. By design, $\mathtt{Vol}_a(0;U) = \mathsf{lp}/n$. And, $\mathtt{Vol}_a(R/2;U) \leq 2\mathsf{lp}$ (being extremely generous). Therefore, the LHS is at most $\ln(2n)$, which is a contradiction. □

Notes

The sparsest cut is intimately connected to the balanced cut which asks to divide the graph into roughly equal pieces and minimize the number of crossing edges. This notion of partition is, in many applications, a much more robust notion of connectivity of a graph, and has numerous applications from image processing to network analysis. Sparsity is also connected to *expansion* of a graph which is a notion of "algebraic connectivity", and we point the reader to the excellent survey [3] by Hoory, Linial, and Wigderson to learn more about this. The algorithm described here is from the seminal paper [5] by Leighton and Rao. The current best known approximation for sparsest cut is an $O(\sqrt{\log n})$-approximation from another seminal paper [1] by Arora, Rao, and Vazirani. The sparsest cut is also extremely connected to *metric embeddings*, and we may touch on this in a later lecture. There is a $O(1)$-approximation known when the graph is a planar or a bounded-genus graph, and there is a 2-approximation known in graphs of low tree-width. The former result is in the paper [4] by Klein, Plotkin, and Rao, and the latter is in the paper [2] by Gupta, Talwar, and Witmer.

References

[1] S. Arora, S. Rao, and U. Vazirani. Expander Flows, Geometric Embeddings, and Graph Partitionings. *Proceedings of the* 36^{th} *Annual ACM Symposium on the Theory of Computing (STOC)*, pages 222–231, 2004.

[2] A. Gupta, K. Talwar, and D. Witmer. Sparsest cut on bounded treewidth graphs: algorithms and hardness results. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 281–290, 2013.

[3] S. Hoory, N. Linial, and A. Wigderson. Expander Graphs and their Applications. *Bull. of the Amer. Soc.*, 43(4):439–561, 2006.

[4] P. Klein, S. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proc., ACM Symposium on Theory of Computing (STOC)*, pages 682–690, 1993.

[5] Leighton and Rao. Multicommodity Max-Flow Min-Cut Theorems and Their Use in Designing Approximation Algorithms. *Journal of the ACM*, 46, 1999.