Topics

- Approximating Cuts.
- Clustering.

Approximating Cuts

Remember the Sketch(\(S\)): Estimate the number of nodes that have an edge to a node in \(S\).
Define the following sketch, Sketch(\(S\)): Estimate the number of edges from \(S\) to \((V - S)\) i.e. size of cut(\(S\)) in undirected graph \(G = (V, E)\) such:

\[
\delta(S, V - S) = \{ e \in E : e \in S \times (V - S) \}
\]

Our sketch should be able to compute Sketch(\(S_1 \cup S_2\)) easily if \(S_1\) and \(S_2\) are disjoint. With above in mind we are looking for sketch: \(\sigma(v) \in R, v \in V\) and want to say \(\sigma(S) = \sum_{v \in S} \sigma(v)\).

If \((v, w)\) is an edge (assume nodes are integers) such:

\[
\sigma((v, m)) = \begin{cases} 
\delta(\text{sorted}(v, w)) & \text{if } (v \leq w) \\
-\delta(\text{sorted}(v, w)) & \text{if } (v > w) 
\end{cases}
\]

And now: \(\sigma(v) = \sum_{w:(v,w) \in E} \delta(v, w) \rightarrow \text{An edge } (v, w) \text{ will also be in } \sigma(w) \text{ as } (w, v)\).

If \(S_1\) and \(S_2\) are disjoint then \(\sigma(S_1 \cup S_2) = \sum_{v \in S_1 \cup S_2} \sigma(v) = \sum_{v \in S_1} \sigma(v) + \sum_{v \in S_2} \sigma(v) = \sigma(S_1) + \sigma(S_2)\)

\[
\sigma(S) = \sum_{v \in S} \sigma(v) = \sum_{v \in S} \sum_{w : (v,w) \in E} \sigma((v, w)) = \sum_{v \in S} \sum_{w : (v,w) \in E} \sigma((v, w)) + \sum_{v \in S} \sum_{w \notin S} \sigma((v, w))
\]

As we see from the above equation the elements in the second summation will cancel themselves out.

Set \(\sigma(S) = \text{Normal variable with mean 0 and variance } |\delta(S)|\). In this case \(|\delta(S)|^2\) is expectation of \(|\delta(S)|\).
This sketch can be used in:

- Sparsifiers → Preserves all cuts simultaneously, also it stores small number of edges.
- Finding connected components in graphs.

**Clustering**

Clustering algorithm on $N$ given nodes ($V$) and distance metric $d(v, w)$.

**Facility Location**

Find $F \subseteq V$ to minimize $f_{\text{facility cost}} + \sum_{v \in V} \min_{w \in F} d(v, w)$

Goal is to build facilities at subset of $V$, with $F$ cost of building a facility. We will build an incremental algorithm for this problem:

Nodes arrive one at a time. At time $t$, node $v_t$ arrives. Also $F_t = \text{set of facilities after node } t$ arrives, with the following properties:

- $F_0 = \{\}$, $F_t \subseteq F_{t+1}$.
- At each step the algorithm chooses $v_t$ as facility with prob $\delta \frac{\delta}{f}$ knowing $\delta = \min_{w \in F_{t-1}} d(v_t, w)$.

$$F_t = \begin{cases} 
F_{t-1} \cup \{v_t\} & \text{with prob } \frac{\delta}{f} \\
F_{t-1} & \text{otherwise}
\end{cases}$$

**Some algorithm notes**

- Incremental (never revisit old decisions)
- Space and time per node depends on $|F|
- Use LSH $\rightarrow \delta = \min_{w \in F_{t-1}} d(v_t, w)$

- $d(., .)$ and $V$ are chosen by adversary, but we will assume that nodes of $V$ are presented in random order (random permutation model)!

**Approximation factor**

Prove that the algorithm is giving a close answer to optimal solution will be provided in the next lecture note.