CMPSCI 711: More Advanced Algorithms

Graphs 3: Linear Sketching for Graph Sparsification

Andrew McGregor

Overview:

- Probabilistic algorithm for constructing a cut sparsifier.
- $O(\epsilon^{-2} n \operatorname{polylog} n)$ space algorithm for constructing cut sparsifier in insert/delete model.

[Guha, McGregor, Tench SODA 16]

Last Compiled: February 13, 2018

Sparsification

Fact (Karger)

G has at most $cn^{2t/\lambda}$ cuts of size t where λ is the size of the min-cut and c is some large constant.

Lemma

Let G be an unweighted graph G with minimum cut of size

$$\lambda > \lambda^* = 24\epsilon^{-2}\ln(2n^2c) \ .$$

Construct G' by sampling each edge with probability $1/2$. Then,

$$\lambda_A(G') = (1 \pm \epsilon)\frac{\lambda_A(G)}{2} \quad \forall A \subset V$$

where $\lambda_A(\cdot)$ is the number of edges between A and $V \setminus A$ in the graph.

Proof of Lemma

- Consider A with $\lambda_A(G) = t$ and let $X = \lambda_A(G')$.
- Then $\mathbb{E}[X] = t/2$ and by an application of the Chernoff Bound,

$$P(|X - \mathbb{E}[X]| \geq \epsilon \mathbb{E}[X]) \leq 2\exp(-\epsilon^2 t/6)$$

- Taking the union bound over all cuts gives,

$$\begin{aligned}
& \mathbb{P}[\lambda_A(G') \neq (1 \pm \epsilon)\lambda_A(G)/2 \text{ for some } A] \\
\leq \quad & \sum_{t \geq \lambda} \mathbb{P}[\lambda_A(G') \neq (1 \pm \epsilon)\lambda_A(G)/2 \text{ for some } A \text{ with } \lambda_A(G) = t] \\
\leq \quad & \sum_{t \geq \lambda} 2\exp(-\epsilon^2 t/6) \cdot cn^{2t/\lambda} \\
= \quad & \sum_{t \geq \lambda} 2c\exp\left(\frac{2t\ln n}{\lambda} - \frac{\epsilon^2 t}{6}\right) \\
\leq \quad & \sum_{t \geq \lambda} 2c\exp\left(-\frac{\epsilon^2 t}{12}\right) \leq 2cn^2\exp\left(-\frac{\epsilon^2 \lambda}{12}\right) \leq 1/n
\end{aligned}$$

Sparsification Algorithm

- Find "light" edges L_0 in G where a set of edges is light if it's removal leaves components with min-cut $\geq \lambda^*$. Let G_1 be formed by removing L_0 and sampling each remaining edge with probability $1/2$.

$$\lambda_A(G) =_{(1+\epsilon)} 2\lambda_A(G_1) + \lambda_A(L_0)$$

- Find light edges L_1 in G_1. Let G_2 be formed by removing L_1 and sampling each remaining edge with probability $1/2$.

$$\lambda_A(G_1) =_{(1+\epsilon)} 2\lambda_A(G_2) + \lambda_A(L_1)$$

and so

$$\lambda_A(G) =_{(1+\epsilon)^2} 4\lambda_A(G_2) + 2\lambda_A(L_1) + \lambda_A(L_0)$$

- Next iteration,

$$\lambda_A(G) =_{(1+\epsilon)^3} 8\lambda_A(G_3) + 4\lambda_A(L_2) + 2\lambda_A(L_1) + \lambda_A(L_0)$$

- Repeat $t = 2\log n$ times: With high probability $G_t = \emptyset$ and so

$$\lambda_A(G) =_{(1+\epsilon)^t} 2^t\lambda_A(L_t) + \ldots + 2\lambda_A(L_1) + \lambda_A(L_0)$$

k-Edge Connectivity via Sketches

- We designed a sketch $\mathcal{A}$ such that for any graph G, we can find a spanning forest F from $\mathcal{A}(G)$ with high probability.
- Construct k independent spanning sketches $\mathcal{A}_1(G), \ldots, \mathcal{A}_k(G)$:
  - $\mathcal{A}_1(G)$ gives a spanning forest F_1 of G.
  - $\mathcal{A}_2(G) - \mathcal{A}_2(F_1) = \mathcal{A}_2(G - F_1)$ gives a spanning forest F_2 of $G - F_1$.
  - $\mathcal{A}_3(G) - \mathcal{A}_3(F_1) - \mathcal{A}_3(F_2) = \mathcal{A}_3(G - F_1 - F_2)$ gives a spanning forest F_3 of $G - F_1 - F_2$.
  - Continue until we've found spanning forests $F_1, \ldots, F_k$.
- Note that $F_1 \cup \ldots \cup F_k$ is k-connected iff G is k-connected.
- Furthermore, an edge e is in a cut of size $\leq k - 1$ in $F_1 \cup \ldots \cup F_k$ iff it is in a cut of size $\leq k - 1$ in G.
- Let's call the overall sketch $\mathcal{B}$.

Finding Light Edges via k connectivity sketch

- Define sets of edges $E_1, E_2, \ldots$ where

$$E_1 = \text{ all edges in } G \text{ in a cut of size at most } \lambda^* - 1$$

$$E_i = \text{ all edges in } G - E_1 - E_2 - \ldots - E_i \text{ in a cut of size at most } \lambda^* - 1$$

  When the process terminates, $L = E_1 + E_2 + \ldots$ is set of light edges.
- We can find $E_1, E_2, \ldots$ from a λ^* edge connectivity sketch $\mathcal{B}(G)$:
  - $\mathcal{B}(G)$ gives you E_1
  - $\mathcal{B}(G) - \mathcal{B}(E_1) = \mathcal{B}(G - E_1)$ gives you E_2.
  - $\mathcal{B}(G) - \mathcal{B}(E_1) - \mathcal{B}(E_2) = \mathcal{B}(G - E_1 - E_2)$ gives you E_3 etc.
  - Continue until you've found L.

Putting it all together

- Let $\mathcal{S}_i$ be a sketches that samples each edge with probability $1/2^i$ where an edge is sampled using $\mathcal{S}_i$ only if it is sampled using $\mathcal{S}_{i-1}$.
- Sketch the data:

$$\mathcal{BS}_0(G), \mathcal{BS}_1(G), \ldots, \mathcal{BS}_{2\log n}(G)$$

- Post-processing:
  1. $\mathcal{BS}_0(G)$ gives L_0
  2. $\mathcal{BS}_1(G)$ gives L_1 (ignore any edges already in L_0)
  3. $\mathcal{BS}_2(G)$ gives L_2 (ignore any edges already in L_0, L_1)
  4. ...... gives L_t for $t = 2\log n$
- Return

$$L_0 + 2L_1 + 4L_2 + \ldots + 2^t L_t$$

- This is a $(1+\epsilon)^{2\log n}$ sparsifier and the size of the sketches is $O(\epsilon^{-2} n \,\text{polylog}\, n)$. Setting $\epsilon = \frac{\gamma}{2\log n}$ gives a $1+\gamma$ sparsifier.