15-859N
11/9/16

Near $O(m \log n)$ Laplician solvers

<u>Thm</u> Can find x st if $A\bar{x} = b$ then

$\|x - \bar{x}\|_A < \varepsilon \|\bar{x}\|_A$ in time

$\tilde{O}(m \log n \log(1/\varepsilon))$ expected time.

Note $\tilde{O}$ hides $\log\log n$ terms.

<u>Thm</u> Can find LSST T st

$\text{stretch}_T(G) = O(m \log n\, ll^3 n)$ in time

$O(m \log n + n \log n\, ll n)$

(AW Thm) Let $Y_1, \dots, Y_k$ be iid random spsd $n \times n$ matrices st 1) $E(Y_i) = Z$

2) $Y_i \preceq \mu Z$

$$\Pr\left[(1-\varepsilon)Z \preceq \sum_{i=1}^{k} \frac{Y_i}{k} \preceq (1+\varepsilon)Z\right] \geq 1 - 2n e^{-\varepsilon^2 k / 4\mu}$$

High Level View of Alg

1) Given $G=(V,E)$

Generate a solver chain of graphs:

$$G=G_0, G_1, \ldots, G_{l=c\log n} = \text{constant}.$$

such that (k is condition number)

$$k(G_0,G_1) = O(\log^2 n)$$

$$k(G_i, G_{i+1}) = \text{constant for } 1 \le i \le l-1$$

2) Run

Recursive Preconditioned Chebyshev on seq $G_0, \ldots, G_l$.

Solver has 2 phases

phase 1 $\equiv$ Reduces G to a spine-heavy graph

Def G is spine-heavy if $\exists$ ST T of G st

$$\text{stretch}_T(G \setminus T) = O(m/\log n)$$

ReduceSpineHeavy(G) (Phase-1)

1) $T \Leftarrow LSST(G)$
2) $H \Leftarrow G + (t-1)T$ some constant $t = c\log^3 n$.
3) $H' \Leftarrow \text{Sample}(H, tT)$
4) Run $PCG(G, H')$ $O(\log n)$ iterations ($\sqrt{t} = \log n$) using calls to Phase 2(H', b', T).

Here H' is G_1.

<u>Note</u> AW is sampling with replacement.
Before we have combined identical samples.
Here we will not.
Thus we get a multigraph.

<u>Def</u> We call each edge in multigraph a <u>sample</u>.

<u>Note</u> in Spielman-Srivastava
we sampled with prob $\equiv \dfrac{ER_e}{n-1}$

Now given $r_e \geq ER_e$ we sample
with prob $= \dfrac{r_e}{r}$ $\quad r = \sum r_e$.

This gives a subgraph with $O(r \log n)$
edges.

<u>Sample</u>$(G=(V,E,w),T)$

1) $e \in G\setminus T \quad p'_e = w_e\, ER_T(e)$

2) $t \equiv \sum_{e \in G\setminus T} p'_e \equiv \text{stretch}_T(G\setminus T)$

3) $p_e = p'_e / t$

4) set $k = C_S\, t \log n \log(1/\varepsilon)$

5) Sample k edges $G\setminus T$ with prob p_e weight w_e/p_e $(e_1 \cdots e_k)$

6) return $\left(T + \frac{w_{e_1}}{k \cdot p_{e_1}} e_1 + \cdots + \frac{w_{e_k}}{k \cdot p_{e_k}} e_k\right)$

To apply (AW) a sample will formally be:

$$Y_i = \left(T + \frac{w_{e_i}}{p_{e_i}} e_i\right) \text{ with prob } p_{e_i}$$

This will return 6).

<u>Note</u> #samples $\equiv k = \tilde{O}(m \log^2 n \log(1/\varepsilon))$

Graph Algorithms on Trees

T is a rooted tree with edge weights
root r.

<u>Claim</u> $O(n)$ time find dist to each from root.

DFS or BFS.

<u>Def</u> LCA = lowest common ancestor.

<u>(Targan)</u> Given pairs (v_i, w_i) $v_i, w_i \in V$

Find $LCA(v_1, w_1), \ldots, LCA(v_m, w_m)$ in $O(m\,\alpha(m))$

Time $\alpha(m)$ is inverse Ackerman fcn. $*$

<u>Cor</u> Find $Dist(v_i, w_i)$ $\forall i$ in $O(m\,\alpha(m))$ time.

<u>Claim</u> If Phase 2 correctly solves $L_{H'}x = b'$ then Phase 1 returns an approx solution to $L_G x = b$

Pf $\sigma(G/H) = 1$
$\sigma(H/G) \le c \log^2 n$ $\Rightarrow \kappa(G,H) \le c \log^2 n$

$\kappa(H, H') \le 3$ (by AW) ($\varepsilon = 1/2$ in AW)

$\Rightarrow \kappa(G, H') \le 3c \log^2 n$

Since PCG only requires $O(\sqrt{\kappa(G,H')})$ iterations per bit. #iter $= O(\log n)$

Claim Let ℓ be a normalized sample returned by Sample(G,T) then

$$\mathrm{stretch}_T(\ell) = \frac{1}{C_S \log n \log(1/\varepsilon)}$$

---

pf

Let $e \in G \setminus T$

$P_e = P'_e / t \qquad t = \sum P'_e$

if picked weight $\frac{w_e}{P_e}$

normalized weight $\frac{w_e}{P_e} \frac{1}{k} = w_\ell$

$$w_\ell = \frac{\cancel{w_e}\, t}{\cancel{w_e}\, ER_T(e)} \cdot \frac{1}{C_S \cancel{t} \log n \log(1/\varepsilon)}$$

$$= \frac{1}{C_S\, ER_T(e) \log n \log(1/\varepsilon)}$$

$$\text{stretch}_T(\ell) = w_\ell \, ER_T(e) = \frac{1}{C_S \log n \log(1/\varepsilon)}$$

Phase2(H, T)

1) $H' \leftarrow H + (k-1)T$
2) $H'' = \text{Sample}(H', kT)$ note $kT \subseteq H'$
3) P-Chebyshev(H, H'')
4) Phase2(H'', T)

<u>Claim</u> In Phase 2 $k(H, H'') \leq c$ fixed constant with h.p.

1) $H \preceq H' \preceq kH$

2) (AW) $(1-\frac{1}{2})H' \preceq H'' \preceq (1+\frac{1}{2})H'$ h.p.

$\Rightarrow \frac{1}{2}H \preceq \frac{1}{2}H' \preceq H'' \preceq \frac{3}{2}H' \preceq \frac{3}{2}kH \Rightarrow k(H, H'') \leq 3k$

<u>Claim</u> If H has m samples (multi-edges) then Phase2(H'', T) returns $m' \leq m/k$ samples h.p.

<u>Pf</u>

$\#$samples $= S = C_S \, St_T(H' \backslash T) \log n \log(1/\varepsilon)$

$$St_T(H' \backslash T) = \frac{1/k \, m}{C_S \log n \log(1/\varepsilon)}$$

$$S' = \cancel{C_S}\left(\frac{m}{k \, \cancel{C_S} \cancel{\log n} \cancel{\log(1/\varepsilon)}}\right) \cancel{\log n} \cancel{\log(1/\varepsilon)} = m/k$$

In Summary

Phase 1: Reduces solving G to $\lg n$ solves of form $L_H x = b'$ H fixed.
H is spine heavy.

Phase 2: Generates a chain of graphs $H_1, H_2, \ldots, H_k$ st

1) $|E(H_i)| \geq 3|E(H_{i+1})|$

2) $\kappa(H_i, H_{i+1}) \leq 9$

To solve $L_{H_i} x = b'$ we make constant # of calls to $L_{H_{i+1}} = b''$.

Issues:

1) The vertex count is <u>not</u> decreasing!
2) We need that cost/call decreases faster #calls.