15-859NN
4/2/21

Nearly $O(m \log n)$ Laplician Solver

Thm $\exists$ alg st $Alg(A,b) = x$ s.t.
If $A\bar{x} = b$ then $\|x - \bar{x}\|_A \leq \varepsilon \|\bar{x}\|_A$ in time
$\tilde{O}(m \log n \log(1/\varepsilon))$ expected time.

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

Recall: $\|x\|_A = \sqrt{x^T A x}$

We will need but not prove:

Thm $\exists$ LSST T of G st
$\text{stretch}_T(G) = O(m \log n (\log\log)^3 n)$
in time $O(m \log n + n \log n\, \ell\ell n)$

We Proved

(AW Thm) Let $Y_1, \cdots, Y_k$ be iid random SPSD $n \times n$ matrices st

1) $E(Y_i) = Z$ 2) $Y_i \leq \mu Z$ then

$$\Pr\left[(1-\varepsilon)Z \preceq \frac{1}{k}\sum_{i=1}^{k} Y_i \preceq (1+\varepsilon)Z\right]$$

$$\geq 1 - 2n\, e^{-\varepsilon^2 k/4\mu}$$

High Level View of Alg

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

Generate a solver chain of graphs:

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

such that (k is condition number)

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

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

2) Run

Recursive Preconditioned Chebyshev (RPC)

on seq $G_1, \ldots, G_\ell$.

That is, when doing iteration to solve

$G_i y_i = b_i$

We make calls to $G_{i+1} y_{i+1} = b_{i+1}$.

The solver has 2 distinct Phases.

In Phase 1 we reduce solving G to solving a spine-heavy graph.

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

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

Phase1(G):

1) Compute $T \Leftarrow LSST(G)$
2) Set $H \Leftarrow G + (t-1)T$ some $t = c\log^2 n$
3) $H' \Leftarrow \text{Sample}(H, tT)$ (sample nontree edges)
4) Run PCG(G, H') using call to Phase2(H', T, b').
5) Run 4) $O(\log n = \sqrt{t})$ times per bit of accuracy required.

Note: Here G_1 is H'.

All remaining calls will use Recursive Preconditioned Chebyshev.

<u>Note</u> AW is sampling with replacement.
In the past we combined identical samples.
Here we will not combine samples
Thus we will get a multgraph.

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

<u>Recall:</u> Given an $r_e \geq ER_e$ we can sample with prob $= \frac{r_e}{r}$ $(r = \sum r_e)$ $r \log n$ times

This gives a spectral sparsifier with $O(r \log n)$ edges with H.P.

Sampling nontree edges

Sample$(G=(V,E,w),T)$

1) **for each** $e \in G \setminus T$ **set** $p'_e = w_e \, ER_T(e)$
2) **set** $t = \sum_{e \in G\setminus T} p'_e = \text{Stretch}_T(E(G)\setminus E(T))$
3) **Set** $p_e = p'_e / t$
4) **set** $k = C_S \, t \log n \log(1/\varepsilon)$ (constant C_S)
5) Sample k edges from $G \setminus T$ with prob p_e setting weights to w_e/p_e.
   $(e_1, \dots, e_k)$
6) return $\left(T + \frac{w_{e_1}}{k p_{e_1}} e_1 + \dots + \frac{w_{e_k}}{k p_{e_k}} e_k\right)$

---

to apply AW a sample is formally

$$Y_i = \left(T + \frac{w_e}{p_e} e\right) \text{ with prob } p_e$$

This will return G).

In Phase 1 $\text{Stretch}_T(E(G)\setminus E(T)) = m/\log n$

$\#\text{samples} = k = \tilde{O}(m \log(1/\varepsilon))$

Graph Algorithms on Trees.

Almost all problems on trees is $O(n)$.

Problem: Given Tree T and nontree edges $e_1 \cdots e_m$

Output $ER_T(e_1) \cdots ER_T(e_m)$.

Let T be a root tree with root r.

Note $\forall a \in V(T)$ Using BFS we can label a with $l(a) = ER_T(r, a)$.

Def $LCA(a,b) \equiv$ Lowest common ancessor.

Note: Given $a, b \in V(T)$ and $LCA(a,b)$ Then $ER_T(a,b) = l(a) + l(b) - 2LCA(a,b)$.

Tarjan The LCA of $e_1, \cdots, e_m$ can be computed in $O(m\alpha(m))$ Time. where $\alpha(m)$ is inverse Accerman.

Timing Anaysis of Phase 1
Assuming Phase 2

Claim If Phase 2 correctly solves problems of form $L_{H'}X = b'$ then Phase 1 step 4 returns a constant error reduction to $L_G X = b$.

---

pf: $\sigma(G/H) = 1$ thus $k(G,H) \leq c \log^2 n$
$\sigma(H/G) \leq c \log^2 n$

$k(H, H') \leq 3$ (by AW with error prob $\leq 1/k$)

Thus $k(G, H') \leq 3 \cdot c \log^2 n = O(\log^2 n)$

Since PCG only requires $O(\sqrt{k(G,H')})$ iteration per bit of error.

Thus step only requires $O(\log n)$ iter.

The critical trick to get a one-log solver

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

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

---

pf: Let $e \in G \setminus T$

Recall:

We pick e with frequency $w_e ER_T(e)$.

Thus we pick with probability

$$P_e = w_e ER_T(e)/t \quad \text{where } t = \sum P_e'.$$

If e is picked then it's weight w_1 will be w_e/p_e

The normalized weight after k samples

$$w_\ell = \frac{w_e}{p_e}\frac{1}{k}\ ;\ k = C_s t \log n \log(1/\varepsilon)$$

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

$$= \frac{1}{ER_T(e)\cdot C_s \log n \log(1/\varepsilon)}$$

Finally the Stretch (ℓ)

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

$\square$

Phase2

While #nontree $\geq \sqrt{n}$ do

1) Set $H' \Leftarrow H + (k-1)T$
   (increase weight of T by constant k)
2) $H'' = \text{Sample}(H', kT)$ (note $kT \leq H'$)
3) Pre-Cheb(H, H'')
4) Using calls to Phase2(H'', T)

Analysis of Phase 2

<u>Claim</u> In Phase2 $\kappa(H, H'') \leq c$ fixed constant with h.p.

<u>pf</u> 1) $H \preceq H' \preceq k H$ since $H \subseteq H' \subseteq k H$

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

thus
$\frac{1}{2} H \preceq \frac{1}{2} H' \preceq H'' \preceq \frac{3}{2} H' \preceq \frac{3}{2} k H$ ie $\kappa(H, H'') \leq 3k$

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

<u>pf</u> note #samples $= S = C_s \, \mathrm{Str}_T(H' \setminus T) \lg n \lg(1/\varepsilon)$

Now $\mathrm{Str}_T(H' \setminus T) = \dfrac{\frac{1}{k} m}{C_s \lg n \lg(1/\varepsilon)}$

Since we dialated T by k.

$$S' = \cancel{C_s}\left(\frac{m}{k \cancel{C_s} \cancel{\lg n} \cancel{\lg(1/\varepsilon)}}\right) \cancel{\lg n} \cancel{\lg(1/\varepsilon)} = \frac{m}{k}$$

In Summary

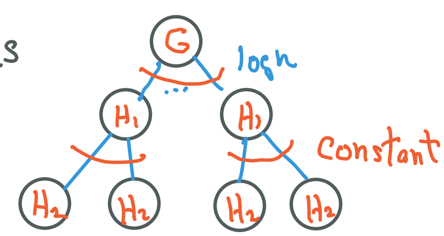
Phase 1: Reduces solving $L_G x = b$ to $\log n$ calls to $L_H x' = b'$. (same H) where H is spine heavy.

Phase 2: Generates a chain of graphs. $H_1, H_2, \dots, H_t$ s.t.

1) $|E(H_i) \setminus T| \geq 3 |E(H_{i+1}) \setminus T|$
2) $\kappa(H_i, H_{i+1}) \leq 9$ HP.
3) To solve $L_{H_i} x = b$ we make a constant # of calls to $L_{H_{i+1}} x' = b'$.

Tree of recursive calls

1) Vertex count is not decreasing!



2) We need that cost/call decreases faster than # calls.

Removing degree one & two node using Gaussian Elimination.

Thm If T ST of G, with m nontree edges, and G has no degree one or two nodes.

then $n \le 2m$

Pf Let 1) V_1 = # leaves in T.

2) V_2 = # nodes of degree 2

3) V_3 = # nodes of degree ≥ 3

Note 1) $V_3 + 2 \le V_1$

Thus $n = V_1 + V_2 + V_3 \le 2V_1 + V_2$

We next count non tree edges

1) Each V_1 node has at least 2 nontree attachments

2) Each V_2 node has at least 1 nontree attachments

Thus $2m \geq 2V_1 + V_2$

or $n \le 2m$ $\square$

Full Algorithm pivots out degree one & two vertices after each Phase2.