Nearly linear time Laplacian solvers.

Capstone: use many things from this semester.

Discuss const accuracy solves - because iterative refinement provides $\ln 1/\varepsilon$ dependence for ε-accuracy.

Fastest is $\tilde{O}(m\sqrt{\lg n})$ $\quad f(n) \leq \tilde{O}(g(n))$ if

$f(n) \leq O(g(n) \lg^c g(n))$ some const c.

Today do $\tilde{O}(m \lg^2 n)$ and sketch $\tilde{O}(m \lg n)$

Often divide into two parts:

precomputation (like factorization, low-stretch tree)

and application.

After precomputation can get time to $O(m)$.

Augmented Spanning Tree:

$\tilde{G}$ = Tree + k edges $k \leq n/5$

eliminate degree 1 & 2 vertices

elim deg 1 decreases degree of neighbor

elim deg 2 does not change degree of neighbors

So, time to do this is $\leq O(n)$.

Are left with a Schur complement on $\leq 4k$ vertices with $\leq 5k$ edges, H.

Algebraically is $L_{\tilde{G}} = U^T \begin{pmatrix} I & 0 \\ 0 & L_H \end{pmatrix} U$

for an upper Δ-ar U with $\leq 2n$ entries

Solve systems in L_G by solve in L_H, U and U^T

As U, U^T Δ-ar, solve in time $O(n)$.

Solve L_H recursively, and thus approximately.

Guarantee alg is linear, corresponds to a Z matrix s.t.

$$(1-\varepsilon) Z^+ \preccurlyeq L_H \preccurlyeq (1+\varepsilon) Z^+$$

$$\Rightarrow (1-\varepsilon)\begin{pmatrix} I & 0 \\ 0 & Z^+ \end{pmatrix} \preccurlyeq \begin{pmatrix} I & 0 \\ 0 & L_H \end{pmatrix} \preccurlyeq (1+\varepsilon)\begin{pmatrix} I & 0 \\ 0 & Z^+ \end{pmatrix}$$

and so $(1-\varepsilon) U^T \begin{pmatrix} I & 0 \\ 0 & Z^+ \end{pmatrix} U \preccurlyeq L_{\tilde{G}} \preccurlyeq (1+\varepsilon) U^T \begin{pmatrix} I & 0 \\ 0 & Z^+ \end{pmatrix} U$

ε- approx translates up in size.

let $M = U^T \begin{pmatrix} I & 0 \\ 0 & Z^+ \end{pmatrix} U$

We get $\kappa(L_G, M) \le \kappa(L_G, L_{\tilde{G}}) \cdot \kappa(L_{\tilde{G}}, M)$

$$\le \frac{(1+\varepsilon)}{(1-\varepsilon)} \kappa(L_G, L_{\tilde{G}})$$

So, loose little.

Lem More than half the vertices of a tree have degree 1 or 2.

proof Has $n-1$ edges. So, ave degree < 2

$\Rightarrow$ # degree 1 vertices > # vertices of degree ≥ 3.

Lem If $H = T + S$, T tree S has k edges

elim vertices of T of degree 1 or 2 that do not touch S until none left, wind up with $\le 4k$ vertices and $\le 5k$ edges.

proof. If $2k \le \frac{n}{2}$, is a degree 1 or 2 vertex not touching S. Eliminate it. Still have a tree + k edges.

Stops when $2k > \frac{n}{2}$.

How choose those extra edges?

Random Sampling

Let $G = (V, E, w)$, $\varepsilon > 0$, $p_{a,b} \in [0,1]$ satisfy

$$p_{a,b} \geq \min\left(1, \frac{4 \ln n}{\varepsilon^2} w_{a,b} \mathrm{Reff}_G(a,b)\right)$$

Form $\tilde{G} = (V, F, u)$ by

$$u_{a,b} = \begin{cases} w_{a,b}/p_{a,b} & \text{prob } p_{a,b} \\ 0 & \text{prob } 1 - p_{a,b} \end{cases}$$

Then $\exists$ $c > 0$ s.t. with prob $1 - n^{-c}$ $\tilde{G}$ is an ε-approx of G

and # edges in $\tilde{G} \leq 2 \sum_{a,b} p_{a,b}$

Want fewer edges, at cost of worse approximation.

And, don't want to compute Reff.

Let T be a low-stretch spanning tree of G.

$\mathrm{Reff}_G(a,b) \leq \mathrm{Reff}_T(a,b)$ (Rayleigh's Monotonicity)

$= \mathrm{Stretch}_T(a,b)$

Set $\hat{G} = G + (s-1)T$: mult edge wts of T by s

$\mathrm{Reff}_{\tilde{G}}(a,b) \leq \frac{1}{s} \mathrm{Stretch}_T(a,b)$

$(a,b) \in T \quad p_{a,b} = 1$

$(a,b) \notin T \quad p_{a,b} = \min\left(1, \frac{4 \ln n}{s \varepsilon^2} \text{stretch}_T(a,b)\right)$

And, construct $\tilde{G}$. Get an ε approx of $\hat{G}$, $\varepsilon = \frac{1}{2}$

with $\leq 2 \sum p_{a,b}$ edges

Let $\sigma = \frac{1}{m} \sum_{a,b} \text{stretch}_T(a,b) \leq \tilde{O}(\lg n)$

$$\sum p_{a,b} \leq \frac{16 (\ln n) \cdot m \cdot \sigma}{s}$$

so for $s \geq \text{const} \cdot \sigma \cdot \ln n$, # edges goes down.

But, want to solve systems in G.

Use $\kappa(G, \tilde{G}) \leq \kappa(G, \hat{G})\, \kappa(\hat{G}, \tilde{G}) \leq 3s$

Will solve in $\tilde{G}$ const $\cdot \sqrt{s}$ times

To make a win, want

off-tree edges in $\tilde{G} \leq \frac{m}{c \cdot \sqrt{s}}$

$\Rightarrow s \geq c(\sigma \cdot \ln n)^2$

Will show this works, and gives time $\tilde{O}(m \lg^2 n)$

Recursion: start with $G_0 = G$

Will solve systems in G_i by G_{i+1}.

Form $\widehat{G_i}$ - scale up tree

$\widetilde{G_i}$ - down sample

Let o_i = ~~# off~~ tree edges in G_i = $m_i - (n_i - 1)$

If $\widetilde{G_i}$ has few off-tree edges, eliminate degree 1 and 2 To set Schur complement, G_{i+1}.

Else, $G_{i+1} = \widetilde{G_i}$

Now, $o_{i+1} \leq \dfrac{o_i}{\text{const} \cdot \sigma \cdot \ln n}$

Will show by induction solve in G_i takes time

$$\leq \tilde{O}(o_i \lg^2 n)$$

By elim, should assume $o_i \geq m_i/5$, so essentially same.

Solve in G_i takes const $\sqrt{s}$ solves in G_{i+1},

mults by G_i

mults by U

Takes time

$$\tilde{O}\left(\sqrt{s} \cdot \left(o_{i+1} \lg^2 n + o_i + n_i\right)\right)$$

$$\leq \tilde{O}\left(\sqrt{s} \cdot o_i\right) \leq \tilde{O}\left(o_i \lg^2 n\right)$$

Saving a log (KMP2)

$G_0 = G + (s-1)T$ is the only big blow up.

Solve systems in G_0 in time $O(n)$

$s = \text{const} \cdot \sigma \cdot \ln n$, so # iters $\sim \sqrt{s} \le \tilde{O}(\ln n)$.

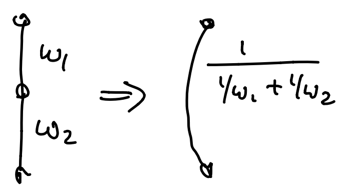
Because T remains dominant.

After sample, $\mathbb{E}\, \text{stretch}_T(\tilde{G} - T) = \text{Stretch}_T(G - T)$

So with high probability, $\text{Stretch}_T(\tilde{G} - T) \le 2\, \text{Stretch}_T(G - T)$

Means set $\hat{G}_i = G_i + 2T_i$ to compensate.

Elimination of degree 1 and 2 vertices in T
does not change stretch.



$l_1 = \frac{1}{w_1}$ $\quad$ $l_{new} = l_1 + l_2$

$l_2 = \frac{1}{w_2}$

Or, can see from Schur complement formula.

Still, need $\sigma_{i+1} \le \frac{\sigma_i}{\text{const}}$

really, $\hat{G}_i = G_i + \text{const}\, T_i$

and time to solve $G_i \le O(\sigma_i)$

Logistics: returning Homework?

Edit this book?

Will be free on line.

Long story: have been many simplifications and improvements.

And are many fast algorithms.

Beginning: many algs use as a primitive

Many generalizations of spectral analysis.

Good luck with the rest of the semester