Preconditioning by Trees

CS 292 F
Lecture 15
May 26, 2021

G : connected weighted grf $c(a,b)$

G : Laplacian matrix $0 = \lambda_1 < \lambda_2$

$$G x = b \quad (1^T b = 0). \qquad G \succeq 0$$

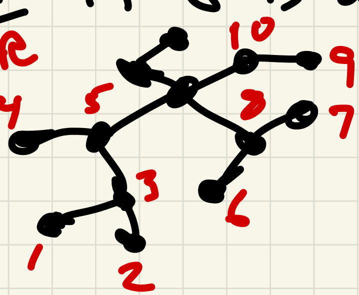
Cholesky factorization

1. Factor $G = R^T R = \Delta \cdot \nabla$
2. Solve $R^T y = b$ for y
3. Solve $Rx = y$ for x

In general, even if G has $O(n)$ edges factorization can take $\Theta(n^3)$ time and solves can take $\Theta(n^2)$ time.

__BUT__ if G is a tree it has an elim order with no extra fill

PARTER ~60



so $O(n)$ memory and $O(n)$ time

Conjugate gradient: $Gx = b$

THM CG finds x_t with $\dfrac{\|x_t - x\|_G}{\|x\|_G} \le \varepsilon$

in at most t iterations,

where $t = O\left(\sqrt{\kappa_f(G)} \cdot \log \frac{1}{\varepsilon}\right)$.

$$\kappa_f(G) = \frac{\lambda_n(G)}{\lambda_2(G)}$$

At each iteration ~~of~~ CG

does $O(m+n)$ work.

$\Rightarrow$ total work $O\left((m+n)\sqrt{\kappa_f} \log \frac{1}{\varepsilon}\right)$

Thm Suppose that all but k of the $n-1$ n^2 eigenvalues of G are in the interval $[a, b]$ $a \le \lambda_i \le b$ for all but k i's.

Then CG gets

$$\frac{\|x_t - x\|_G}{\|x\|_G} \le \varepsilon$$

in $t = k + O\left(\sqrt{\frac{b}{a}} \log \frac{1}{\varepsilon}\right)$

Theorem: If G is any connected, undirected weighted graph with n vertices + m edges, and $\mathbb{1}^T b = 0$, and $\varepsilon > 0$, preconditioned CG can find x_t with

$$\frac{\|x_t - x\|_G}{\|x\|_G} < \varepsilon$$

in $O\left(m^{4/3} \cdot \log \frac{1}{\varepsilon} \cdot (\log n)^c\right)$ time

(# iterations $t \approx m^{1/3}$)

<u>Precond</u> CG: Given G, $Gx=b$.

Have a preconditioner $H \approx G$

and solve $H^{+}Gx = H^{+}b$ for x

(or $H^{+/2}GH^{+/2}x = H^{+}b$)

same eigenvalues

At each iteration:

- multiply a vector by $H^{+}G$

$\equiv$ multiply by G and then H^{+}

$\equiv$ multiply by G and then solve with H.

$\equiv$ solve $Hz=c$ for z.

Idea (Vaidya ~1990):

Take H as a subgraph of G.

Take H to be a tree, a spanning tree.

(so solving $Hz = c$ for z is $\Theta(n)$)

PCG # iterations depends on

$$\kappa_f(H^+G) = \frac{\lambda_n(H^+G)}{\lambda_2(H^+G)}$$

So, how do we pick a tree T for which

$\kappa_f(T^+G)$ is small?

$$T \approx G$$

Let T be a spanning tree of G.

$$G = \sum_{ab \in G} c(a,b) (1_a - 1_b)(1_a - 1_b)^T$$

Consider the <u>trace</u> $Tr(T^+ G)$.

$$\left(\text{Trace} \overset{def}{\equiv} \sum \text{diagonal elts} \overset{thm}{=} \sum \text{eigenvalues} \right)$$

why? T is a sg of $G \Rightarrow T \preceq G$

$\Rightarrow$ (easy) $\lambda_2 (T^+ G) \geq 1$.

$$\Rightarrow \kappa_f (T^+ G) = \frac{\lambda_n}{\lambda_2} \leq \lambda_n \leq Tr(T^+ G)$$

$$\mathrm{Tr}(T^{+}G) = \sum_{ab \in G} c(ab)\,\mathrm{Tr}\left(T^{+}(1_a - 1_b)(1_a - 1_b)^{T}\right)$$

$$= \sum_{ab \in G} c(ab)\,\mathrm{Tr}\left((1_a - 1_b)^{T}\, T^{+}(1_a - 1_b)\right)$$

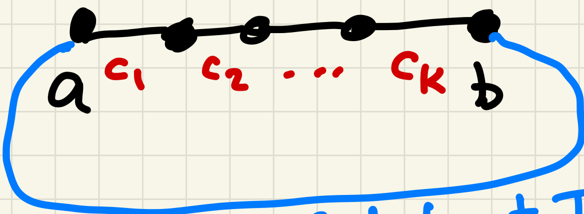
$$= \sum_{ab \in G} c(ab)\,\underline{(1_a - 1_b)^{T}\, T^{+}(1_a - 1_b)}$$

$$= \sum_{ab \in G} c(ab)\,\underbrace{R^{\mathrm{eff}}_{T}(a,b)}_{\text{leverage score of } ab \text{ in } T.}$$

$$\mathrm{Tr}(Avv^{T}) = vAv^{T}$$

$$\sum_{ab \in G} c(ab) \underbrace{R_T^{\text{eff}}(a,b)}_{\text{leverage score of } ab \text{ in } T.}$$

one path a to b in T.



edge $ab \in G$ but not T.

$$R_T^{\text{eff}}(a,b) = \frac{1}{c_1} + \frac{1}{c_2} + \cdots + \frac{1}{c_k}.$$

DEF: Stretch of edge $ab = c(ab) \sum_{i=1}^{k} \frac{1}{c_i}$

(sum along tree path)

DEF: Stretch of a spanning tree T

$$= \sum_{ab \in G} \text{stretch}(ab) = \mathrm{Tr}(T^+ G) \geq K_f$$

THM: Every weighted graph has a spanning tree of stretch $O(m \log n \log\log n)$ that can be computed in time

$$O(m \log n \log\log n) \text{ or so.}$$

$\Rightarrow$ Can compute preconditioner T and get $\kappa_f(T^+G) = \tilde{O}(m)$ so PCG converges in $O(\sqrt{\kappa_f} \log \frac{1}{\varepsilon})$ iterations (at time $O(m)$ each)

$\Rightarrow$ PCG takes time $\tilde{O}(m^{1.5} \log \frac{1}{\varepsilon})$.

[Boman, Hendrickson] ~2000.

Spielman improved the ~~data~~ analysis to show time $\tilde{O}(m^{4/3} \log \frac{1}{\varepsilon})$

HISTORY:

1. Vaidya ~1990
   used maximum-weight spanning tree.
   (plus a few extra edges)
   $\sim m^{1.75}$

2. Boman + Hendrickson ~2000
   used low-stretch trees
   $\sim m^{1.5}$

3. Sp + (?) ~ 2005 improved
   analysis to $\sim m^{1.33}$

   ⋮

   SPIELMAN + TENG

   ⋮

4. Koutis/Miller/Peng ~ 2011
   got $\tilde{O}(m)$.
   → use the few extra edges.
   → do everything recursively.