15-859N

Preconditioned Iterative Methods 10/7/16

Basic Idea

Goal: Solve $Ax=b$ A spd or spsd

Pick a method, say Richardson

Idea find a matrix B and solve

$B^{-1}Ax = B^{-1}b$ (B is called a preconditioner)

note $Ax=b$ iff $B^{-1}Ax=B^{-1}b$

This gives $x^{(n+1)} = x^n + (B^{-1}b - B^{-1}Ax^{(n)})$

$= x^n + B^{-1}(b - Ax^{(n)})$

Alg

1) $r = b - Ax^{(n)}$

2) solve $By = r$

(preprocess $B = LU$ or solve recursively)

3) return $x^{(n+1)} = x^{(n)} + y$

Examples of preconditions

1) Extrapolated method

$$B = \frac{2}{\lambda_{Max}(A) + \lambda_{min}(A)} I$$

2) Jacobi $B = Diagonal(A)$

3) Gauss-Seidel $B = LowerTri(A)$

4) SSOR

5) Graph theoretic

Convergence Rates

P-Richardson Rate $\approx \frac{1}{\kappa(B^{-1}A)}$

PCG Rate $\approx \frac{1}{\sqrt{\kappa(B^{-1}A)}}$

Goal: Find B st we get better convergence rate but solves in step 2 are not too expensive.

Estimating $\kappa(B^{-1}A)$

Note $B^{-1}Ax = \lambda x$ iff $Ax = \lambda Bx$

Def λ & x are called generalized eigenvalues & vectors

Let $\lambda_f(A,B) = \{\lambda : \exists x \; Ax = \lambda Bx \; \& \; x \notin Null(A) = Null(B)\}$

Note $Ax = \lambda Bx$ iff $Bx = (1/\lambda)Ax$

If B spd then iff $B^{-1/2} A B^{-1/2} x = \lambda x$

$$\kappa(B^{-1}A) = \frac{Max\, \lambda_f(A,B)}{min\, \lambda_f(A,B)} = Max\, \lambda_f(A,B)\, Max_f(B,A)$$

$$\tau(A/B) = \underset{\gamma}{\operatorname{argmin}}\ \gamma B \succeq A$$

Claim $\kappa(B^{-1}A) = \tau(A/B)\,\tau(B/A)$

Subgraph Preconditioners

Let $H \subseteq G$ be a subgraph of G.

$B = L_H$ & $A = L_G$

$B \preceq A$ then eigens of $B^{-1}A \geq 1$

Thus $\kappa(B^{-1}A) \leq \tau(A/B) = \arg\min_{\tau} \tau B \succeq A$

for any subgraph Precon!

Spanning Tree Preconditioners

Vaidya: H to be max weight spanning tree of G

say T

$\varphi : G \xrightarrow{\text{path}} T$

Congestion $\leq m$ # edge in G

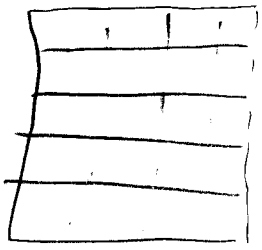
Dialation $\leq n$

$\nabla(T/G) \leq 1 \quad (T \subseteq G)$

$\nabla(G/T) \leq m \cdot n$

$K(G,T) \leq m \cdot n$

EG

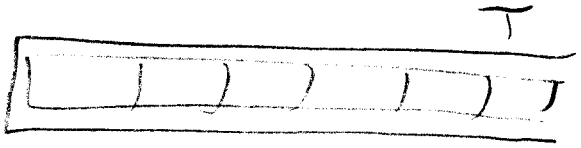


The E tree for square mesh.

$C = \sqrt{n}$

$D = \sqrt{n}$

$C \cdot D = n$



$C = n$

$D = n$

$C \cdot D = n^2$

Thm Max Spanning Tree CG in $O(m^{3/2} \cdot n^{1/2})$ time per bit.

---

1) $O(m)$ time to find T
2) $O(m)$ time per iteration
3) $O((m \cdot n)^{1/2})$ iteration per bit using PCG.

Finding Better Trees

EG C_n be the cycle & P_n a spanning tree.

Claim $\kappa(C_n, P_n) = \Theta(n)$

1) $\tau(P_n / C_n) = 1$ & $\tau(C_n / P_n) \leq C \cdot D = O(n)$

$x = (x_1, \ldots, x_n)$ $\quad x_i = i$

$$\frac{x^T C_n x}{x^T P_n x} = \frac{(n-1)^2 + (n-1)}{(n-1)} = n$$

$\lambda_{max}(P_n^{-1}C_n) \approx n \quad \lambda_{min}(P_n^{-1}C_n) \approx 1$

What is the average? $\sum \lambda_i$?

Fact: $Tr(A) = \sum \lambda_i$

$Tr(P_n^{-1}C_n)$? Not $Tr(AB) = Tr(BA)$

$A = L_G$ & $B = L_H$

Claim $Tr(B^{-1}A) = \sum_{e \in A} w_A(e)\, ER_B(e)$

pf Let $L_{u,v} = (e_u - e_v)(e_u - e_v)^T$ $\quad p_u = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \} u$

$$Tr(B^{\dagger}A) = Tr\left(\sum_{e=(u,v)\in A} B^{\dagger} w_A(u,v) L_{u,v}\right)$$

$$= \sum w_A(u,v) \sum Tr(B^{\dagger} L_{uv})$$

$$= \sum w_A(u,v) \underbrace{(e_u - e_v)^T B^{\dagger} (e_u - e_v)}_{(*)}$$

$$\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = B^{-1} \begin{pmatrix} 0 \\ \vdots \\ +1 \\ 0 \\ \vdots \\ -1 \\ 0 \\ \vdots \end{pmatrix} \text{ if } B \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ +1 \\ 0 \\ \vdots \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\therefore (*) = ER_B(u,v)$$

Def Given G, spanning tree T, and $e \in G$

Let $w_1, \ldots, w_k$ edge weights on path from u to v. $e = (u,v)$

$$\text{stretch}(e) = w_A(e) \sum_{i=1}^{k} \frac{1}{w_i}$$

Thm $\forall G \; \exists$ spanning tree T s.t.

$$\sum_{e \in G} \text{stretch}(e) = O(m \log n (\ell\ell n)^2)$$

m = # edges in G.

Back to C_n & P_n

$$\sum_{\lambda_i \in \lambda(C_n, P_n)} \lambda_i = Tr\left(P_n^{-1} C_n\right) = \sum_{e \in C_n} ER_T(e)$$

note: $\forall e \in P_n \quad ER_T(e) = 1$

For $e \notin P_n \quad ER_T(e) = n-1$

$$\sum \lambda_i = 2(n-1)$$

Homework: 1 of size n 1 of size 0.
$n-2$ of size 1

Consider $P(x) = (1-x)(1-x/n)$

After 3 iterations PCG on (C_n, P_n) will converge.

Thm1 Suppose we have

$$\lambda_1 \geq \cdots \geq \lambda_k \geq \beta \geq \lambda_{k+1} \geq \cdots \geq \lambda_n \geq \beta$$

then $\forall t \geq k$ $\exists P(x)$ of degree t s.t.

1) $P(0)=1$

2) $\forall i \quad |P(\lambda_i)| \leq 2\left(\frac{\sqrt{\beta/\alpha}-1}{\sqrt{\beta/\alpha}+1}\right)^{t-k}$

pf Let $r(x)$ be the Chebychev poly

of deg $= t-k$ st $\forall \alpha \leq x \leq \beta$

$$|r(x)| \leq 2\left(\frac{\sqrt{\beta/\alpha}-1}{\sqrt{\beta/\alpha}+1}\right)^{t-k}$$

Set $P(x) = r(x) \prod_{\lambda_i > \beta} \left(1 - x/\lambda_i\right)$

to show: $\forall i\ [1 \le i \le n]\quad |P(\lambda_i)| \le 2\left(\frac{\sqrt{\beta/2}-1}{\sqrt{\beta/2}+1}\right)^{t-k}$.

For $\lambda_i > \beta \quad P(\lambda_i) = 0$

For $\lambda_i < \beta$

Note: $(1 - x/\lambda_i) \le 1$ for $0 \le x \le \lambda_i$

Thus $|P(\lambda_i)| \le |r(\lambda_i)|$ □

Consider PCG using LSST.

Let $\lambda_1 \le \cdots \le \lambda_n = \lambda(L_T^{-1} L_G)$

Note $\lambda_1 \ge 1$

$Tr(L_T^{-1} L_G) = O(n \log n \log^2 n)$

We claim most eigens must small!

Thm PCG with LSST precon
$O(m^{1/3} \log n)$ iteration per bit.

Pf

to show: $\exists P(3)$, $\deg P = O(m^{1/3} \log n)$
s.t. λ_i $|P(\lambda_i)| \le 1/2$.

Set $\beta = \mathrm{Tr}(L_T^{-1} L_G)^{2/3}$

$K = \mathrm{Tr}(L_T^{-1} L_G)^{1/3}$

We will find $\deg(P) = 2K = 2\sqrt{\beta}$

$$|P(\lambda_i)| \le 2\left(1 - \frac{1}{\sqrt{\beta}}\right)^{2\sqrt{\beta} - \sqrt{\beta}} = 2\left(1 - \frac{1}{\sqrt{\beta}}\right)^{\sqrt{\beta}} \approx \frac{1}{e}$$

Thus $O(\mathrm{Tr}(L_T^{-1} L_G)^{1/3})$ iter per bit
or $O(m^{1/3} \log n)$ iter per bit.