Symmetric Preconditioner Refinement using Low Rank Approximations

Tyler Chen

University of Washington

Feburary 7, 2019

joint work with Anne Greenbaum and Kelly Liu

Introduction

- Given $A \in \mathbb{R}^{n \times n}$ full rank and b, want to find x so that,

$$Ax = b$$

Introduction

- Given $A \in \mathbb{R}^{n \times n}$ full rank and b, want to find x so that,

$$Ax = b$$

- Direct methods:
  - exist matrix multiplication time algorithms $(\mathcal{O}(n^{2.37..}))$
  - impractical if n is very large

Introduction

- Given $A \in \mathbb{R}^{n \times n}$ full rank and b, want to find x so that,

$$Ax = b$$

- Direct methods:
  - exist matrix multiplication time algorithms ($\mathcal{O}(n^{2.37..})$)
  - impractical if n is very large
- Iterative methods:
  - produces sequence of approximations to the true solution
  - good if A has additional structure (sparse, positive definite)

Krylov Subspace Methods

- Iteratively find approximations x_k to the solution x

Krylov Subspace Methods

- Iteratively find approximations x_k to the solution x
  - Construct (and store) orthonormal basis for k-th Krylov subspace, $\mathcal{K}_k(A, v)$ using Arnoldi algorithm

$$\mathcal{K}_k(A, v) = \text{span}\{v, Av, \ldots, A^{k-1}v\}$$

Krylov Subspace Methods

- Iteratively find approximations x_k to the solution x
  - Construct (and store) orthonormal basis for k-th Krylov subspace, $\mathcal{K}_k(A, v)$ using Arnoldi algorithm

$$\mathcal{K}_k(A, v) = \text{span}\{v, Av, \ldots, A^{k-1}v\}$$

  - Pick x_k from $\mathcal{K}_k(A, v)$ to optimize some quantity
    - 2-norm of residual: GMRES ($v = r_0$)
    - A-norm of error: conjugate gradient ($v = e_0$)

Krylov Subspace Methods cont.

- Cayley–Hamilton theorem: convergence in at most n steps
- Matrix free: only need to be able to evaluate $x \mapsto Ax$
- If A is symmetric positive definite, Arnoldi algorithm reduces to a 3-term recurrence
  - Don't need to save whole basis for $\mathcal{K}_k(A, e_0)$

Convergence of conjugate gradient

- Error satisifes,

$$\|e_k\|_A = \min_{p_k} \|p_k(A)\,e_0\|_A \,, \qquad \deg p_k \leq k$$

Convergence of conjugate gradient

- Error satisifes,

$$\|e_k\|_A = \min_{p_k} \|p_k(A)\, e_0\|_A \,, \qquad\qquad \deg p_k \leq k$$

- This gives rate of convergence

$$\frac{\|e_k\|_A}{\|e_i\|_A} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \qquad\qquad \kappa = \lambda_{\mathsf{max}}/\lambda_{\mathsf{min}}$$

Convergence of conjugate gradient

- Error satisifes,

$$\|e_k\|_A = \min_{p_k} \|p_k(A)\, e_0\|_A \,, \qquad \deg p_k \leq k$$

- This gives rate of convergence

$$\frac{\|e_k\|_A}{\|e_i\|_A} \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \qquad \kappa = \lambda_{\mathsf{max}} / \lambda_{\mathsf{min}}$$

- Convergence is roughly governed by condition number
  - In practice, not very simple how convergence depends on $\lambda(A)$
  - in finite precision even less is known

Preconditioners

- Basic idea: solve equivalent but “easier” system

Preconditioners

- Basic idea: solve equivalent but "easier" system
- For any full rank M, we can find x by solving,

$$M^{-1}Ax = M^{-1}b$$

  - Extreme case $M^{-1}A = I$

Preconditioners

- Basic idea: solve equivalent but "easier" system
- For any full rank M, we can find x by solving,

$$M^{-1}Ax = M^{-1}b$$

  - Extreme case $M^{-1}A = I$
- To keep system SPD, solve,

$$R^{-1}AR^{-T}y = R^{-1}b, \qquad x = Ry$$

  - Again want $R^{-1}AR^{-T}$ to be close to the identity

Preconditioner Refinement [2]

- Start with some standard symmetric preconditioner R^{-1}

Preconditioner Refinement [2]

- Start with some standard symmetric preconditioner R^{-1}
- Define $E = R^{-1}AR^{-T} - I$

Preconditioner Refinement [2]

- Start with some standard symmetric preconditioner R^{-1}
- Define $E = R^{-1}AR^{-T} - I$
- Approximate E by low rank matrix E_k
  - Approximate SVD: $\mathcal{O}(n^2 \log k)$ [1]

Preconditioner Refinement [2]

- Start with some standard symmetric preconditioner R^{-1}
- Define $E = R^{-1}AR^{-T} - I$
- Approximate E by low rank matrix E_k
  - Approximate SVD: $\mathcal{O}(n^2 \log k)$ [1]
- New preconditioner: $(I + E_k)^{-1/2}R^{-1}$
  - Efficiently compute $(I + E_k)^{-1}$ by Woodby formula:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VAU)^{-1}VA^{-1}$$
$$(I + U\Lambda U^T)^{-1} = I - U(\Lambda^{-1} + I_k)^{-1}U^T$$

Preconditioner Refinement cont.

- Storage for E_k is $\mathcal{O}(nk)$
- Multiplying by $(I + E_k)^{-1}$ is additional $\mathcal{O}(nk)$ per iteration
- This has potential in high performance regime, where limiting costs are multiplications with A

Example : 1138 BUS (Power systems admittance matrices)

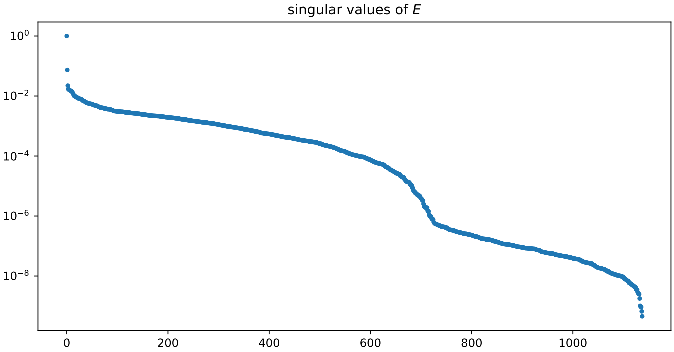

Figure: Singular value spectrum of E

Example : 1138 BUS (Power systems admittance matrices)

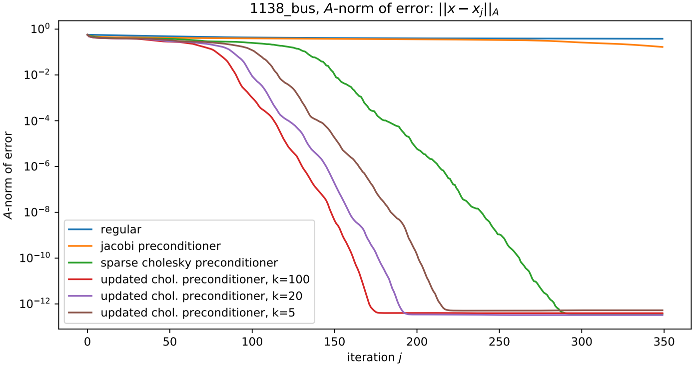

Figure: Convergence of conjugate gradient with specified preconditioners

Further work

- We really only need that $E = R^{-1}AR^{-T} - X$ is low rank for some X and that the inverse of $E + X$ is easy to compute.
- This means we can search for new classes of preconditioners that don't necessarily approximate A^{-1}

References

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM review **53** (2011), no. 2, 217–288.

N. Higham and T. Mary, *A new preconditioner that exploits low-rank approximations to factorization error*, SIAM Journal on Scientific Computing **41** (2019), no. 1, A59–A82.