Krylov Space Methods

Nonstationary sounds good

Radu Trîmbițaș

"Babeș-Bolyai" University

Introduction

- These methods are used both to solve $Ax = b$ and to find eigenvalues of A
- They assume A is accessible via a black box subroutine that returns $y = Az$ (and perhaps $y = A^T z$ if A is nonsymmetric) — no direct access or manipulation of matrix entries is used.
- Reasons:
  - This is the cheapest operation on sparse matrices
  - A may not be represented as a matrix but as a subroutine for computing Ax
- There exists a variety of such methods depending on A and availability of A^T

Extracting information via matrix-vector multiplication

- Given a vector b and a subroutine for computing Ax what can we deduce about A?
- Compute $y_1 = b, \ldots, y_m = Ay_{m-1} = A^{m-1}y_1$. Let $K = [y_1, \ldots, y_m]$.

$$A \cdot K = [Ay_1, \ldots, Ay_{m-1}, Ay_m] = [y_2, \ldots, y_m, A^m y_1]. \qquad (1)$$

- Assume K nonsingular, let $c = -K^{-1}A^m y_1$;
  $A \cdot K = K \cdot [e_2, e_3, \ldots, e_m, -c] \equiv K \cdot C$, or

$$K^{-1}AK = C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_1 \\ 1 & 0 & \cdots & 0 & -c_2 \\ 0 & 1 & \cdots & \vdots & \vdots \\ \vdots & 0 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & 0 & \vdots \\ \vdots & \vdots & \cdots & 1 & -c_m \end{bmatrix}.$$

Extracting information via matrix-vector multiplication

- C upper Hessenberg; it is a *companion matrix* and its caracteristic polynomial is $p(x) = x^m + \sum_{i=1}^{m} c_i x^{i-1}$
- We can in principle find the eigenvalues of A by finding the zeros of $p(x)$
- This simple form is not very useful
  - Finding c requires $m-1$ matrix-vector multiplication by A and solving a linear system with K
  - K is ill-conditioned — equivalent to power method y_i converge to a dominant ev of A; column of K tend to get more and more parallel
- We fix this by replacing K with an orthogonal matrix Q such that the leading k columns of K and Q span the same space – called a Krylov subspace.
- We will compute only as many leading columns of Q as needed (for the solution of $Ax = b$ or $Ax = \lambda x$) – very few compare to matrix size

Extracting information via matrix-vector multiplication

- Let $K = QR$ the QR decomposition of K

$$K^{-1}AK = \left(R^{-1}Q^T\right)A(QR) = C \Longrightarrow Q^TAQ = RCR^{-1} \equiv H$$

- C and H are upper Hessenberg
- If A is symmetric, then $H = Q^TAQ$ is tridiagonal
- We compute columns of Q one at a time, rather than all of them, using the modified Gram-Schmidt algorithm (instead of Householder reflections).

Extracting information via matrix-vector multiplication

- Let Q_n be the $m \times n$ matrix with the first n columns of $AQ = QH$, or $AQ_n = Q_{n+1}\widetilde{H}_n$:

$$\left[\begin{array}{c} A \end{array}\right] \left[\begin{array}{c|c|c} q_1 & \cdots & q_n \end{array}\right] = \left[\begin{array}{c|c|c} q_1 & \cdots & q_{n+1} \end{array}\right] \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ h_{21} & & \vdots \\ & \ddots & \vdots \\ & & h_{n+1,n} \end{bmatrix}$$

- The nth column of $AQ_n = Q_{n+1}\tilde{H}_n$ gives

$$Aq_n = h_{1n}q_1 + \cdots + h_{nn}q_n + h_{n+1,n}q_{n+1}$$

- q_i are orthonormal, multiply both sides by q_j^T

$$q_j^T Aq_n = \sum_{i=1}^{n+1} h_{in} q_j^T q_n = h_{j,n} \Longrightarrow h_{n+1,n}q_{n+1} = Aq_n - \sum_{i=1}^{n} h_{in}q_i$$

The Arnoldi algorithm

Algorithm: Arnoldi Iteration for partial reduction to Hessenberg form

$b :=$ arbitrary; $q_1 := b/\|b\|$;
for $n := 1, 2, 3, \ldots$ **do**
  $v := Aq_n$;
  for $j := 1$ **to** n **do**
    $h_{jn} := q_j^* v$;
    $v := v - h_{jn} q_j$;
  $h_{n+1,n} := \|v\|$;
  if $h_{n+1,n} = 0$ **then**
    return
  $q_{n+1} := v/h_{n+1,n}$;

- Complexity n matrix-vector multiplications by $A + O(n^2 m)$ other work

The Arnoldi algorithm

- If we stop the algorithm here, what we have learned about A?
- Let $Q = [Q_n, Q_u]$, where $Q_n = [q_1, \ldots, q_k]$, $Q_u = [q_{n+1}, \ldots, q_m]$; colums of Q_u excepting q_{n+1} are unknown
- Then

$$H = Q^T A Q = [Q_n, Q_u]^T A [Q_n, Q_u] = \begin{bmatrix} Q_n^T A Q_n & Q_n^T A Q_u \\ Q_u^T A Q_n & Q_u^T A Q_u \end{bmatrix}$$

$$\equiv \begin{array}{c} \\ n \\ m-n \end{array} \begin{array}{c} \begin{array}{cc} n & m-n \end{array} \\ \begin{bmatrix} H_n & H_{un} \\ H_{nu} & H_u \end{bmatrix} \end{array}$$

- H_n is upper Hessenberg, H_{nu} has a single nonzero entry ($h_{n+1,n}$), H_u, $H_{u,n}$ unknown

Symmetric Matrices and the Lanczos Iteration

- For symmetric A, H_n reduces to tridiagonal T_n, and q_{n+1} can be computed by a three-term recurrence:

$$Aq_n = \beta_{n-1} q_{n-1} + \alpha_n q_n + \beta_n q_{n+1}$$

Algorithm: Lanczos Iteration

$\beta_0 := 0$; $q_0 := 0$; $b :=$arbitrary; $q_1 := b/\|b\|$;
for $n := 1, 2, 3, \ldots$ **do**
  $v := Aq_n$; {or $Aq_n - \beta_{n-1} q_{n-1}$ for greater stability}
  $\alpha_n := q_n^T v$;
  $v := v - \beta_{n-1} q_{n-1} - \alpha_n q_n$;
  $\beta_n := \|v\|$;
  if $\beta_n = 0$ **then**
    return
  $q_{n+1} := v/\beta_n$;

Symmetric Matrices and the Lanczos Iteration

- If we stop the algorithm here, what we have learned about A?
- We have

$$T = Q^T A Q = [Q_n, Q_u]^T A [Q_n, Q_u] = \begin{bmatrix} Q_n^T A Q_n & Q_n^T A Q_u \\ Q_u^T A Q_n & Q_u^T A Q_u \end{bmatrix}$$

$$\equiv \begin{array}{c} \\ n \\ m-n \end{array} \begin{array}{c} \begin{array}{cc} n & m-n \end{array} \\ \begin{bmatrix} T_n & T_{un} \\ T_{nu} & T_u \end{bmatrix} \end{array}$$

$$= \begin{bmatrix} T_n & T_{nu}^T \\ T_{nu} & T_u \end{bmatrix}$$

Krylov subspace

Definition

The *Krylov subspace* $\mathcal{K}_n(A, b)$ is $\left\langle b, Ab, A^2b, \ldots, A^{n-1}b \right\rangle$.

- We shall write $\mathcal{K}_n$ instead of $\mathcal{K}_n(A, b)$ if A and b are implicit from the context.
- H_n or T_n is the projection of A onto the Krylov subspace $\mathcal{K}_n$.
- One can show that $\mathcal{K}_n$ has dimension n iff the Arnoldi or Lanczos algorithm can compute q_n without quitting first.
- We shall solve $Ax = b$ and $Ax = \lambda x$ using only the information computed by n steps of Arnoldi or Lanczos algorithm
- We hope $n \ll m$, so the algorithms are efficient

Krylov subspace

- For eigenvalues, if $h_{n+1,n} = 0$, then H (or T) is block upper triangular and eigenvalues of H_n are eigenvalues of H, and therefore of A
- To obtain eigenvector of A we multiply eigenvectors of H_n by Q_n
- If $h_{n+1,n}$ is small ew and ev of H_n are good approximations to the ew and ev of A
- Roundoff errors cause a number of algorithms to behave entirely different from how they would in exact arithmetic.
- In particular Lanczos vectorscan loose their orthogonality and become almost linearly dependent.
- Researchers learned how to stabilize the algorithm or that the convergence occured despite instability.

Solving a system using Krylov subspace

- We express the solution as "the best" approximation in the Krylov subspace of the form

$$x_n = \sum_{k=1}^{n} z_k q_k = Q_n z, \qquad \text{where } z = [z_1, \ldots, z_n]^T$$

- We have to define "best". There are several natural but different definitions, leading to different algorithms. Let $x = A^{-1}b$ be the exact solution and $r_n = b - Ax_n$ be the residual

1. x_n minimizes $\|x_n - x\|_2$ — we do not have enough information

Solving a system using Krylov subspace

- We express the solution as "the best" approximation in the Krylov subspace of the form

$$x_n = \sum_{k=1}^{n} z_k q_k = Q_n z, \qquad \text{where } z = [z_1, \ldots, z_n]^T$$

- We have to define "best". There are several natural but different definitions, leading to different algorithms. Let $x = A^{-1}b$ be the exact solution and $r_n = b - Ax_n$ be the residual

1. x_n minimizes $\|x_n - x\|_2$ — we do not have enough information
2. x_n minimizes $\|r_n\|_2$ — A symmetric MINRES (minimum residual), A nonsymmetric (generalized minimum residual)

Solving a system using Krylov subspace

- We express the solution as "the best" approximation in the Krylov subspace of the form

$$x_n = \sum_{k=1}^{n} z_k q_k = Q_n z, \qquad \text{where } z = [z_1, \ldots, z_n]^T$$

- We have to define "best". There are several natural but different definitions, leading to different algorithms. Let $x = A^{-1}b$ be the exact solution and $r_n = b - Ax_n$ be the residual

1. x_n minimizes $\|x_n - x\|_2$ — we do not have enough information
2. x_n minimizes $\|r_n\|_2$ — A symmetric MINRES (minimum residual), A nonsymmetric (generalized minimum residual)
3. x_n makes $r_n \perp \mathcal{K}_n$, i.e. $Q_n^T r_n = 0$ — SYMMLQ and GMRES

Solving a system using Krylov subspace

- We express the solution as "the best" approximation in the Krylov subspace of the form

$$x_n = \sum_{k=1}^{n} z_k q_k = Q_n z, \qquad \text{where } z = [z_1, \ldots, z_n]^T$$

- We have to define "best". There are several natural but different definitions, leading to different algorithms. Let $x = A^{-1}b$ be the exact solution and $r_n = b - Ax_n$ be the residual

1. x_n minimizes $\|x_n - x\|_2$ — we do not have enough information
2. x_n minimizes $\|r_n\|_2$ — A symmetric MINRES (minimum residual), A nonsymmetric (generalized minimum residual)
3. x_n makes $r_n \perp \mathcal{K}_n$, i.e. $Q_n^T r_n = 0$ — SYMMLQ and GMRES
4. When A is SPD, it defines a norm $\|r\|_{A^{-1}} = \left(r^T A^{-1} r\right)^{1/2}$. The best x_n minimizes $\|r_n\|_{A^{-1}}$. This norm is the same as $\|x_n - x\|_A$ — conjugate gradient algorithm

Minimizing Residuals

- Generalized Minimal RESiduals – iterative method for solving $Ax = b$
- Find $x_n \in \mathcal{K}_n$ that minimizes $\|r_n\| = \|b - Ax_n\|$
- This is a least squares problem: Find a vector c such that

$$\|AK_n c - b\| = \text{minimum}$$

  where K_n is the $m \times n$ Krylov matrix
- QR factorization could be used to solve for c, and $x_n = K_n c$
- In practice the columns of K_n are ill-conditioned and an orthogonal basis is used instead, produced by Arnoldi iteration

Minimal Residual with Orthogonal Basis

- Instead of $x_n = K_n c$ set $x_n = Q_n y$, where the orthogonal columns of Q_n span $\mathcal{K}_n$, and solve

$$\|AQ_n y - b\| = \text{minimum}$$

- For the Arnoldi iteration we showed that $AQ_n = Q_{n+1}\tilde{H}_n$:

$$\left\|Q_{n+1}\tilde{H}_n y - b\right\| = \text{minimum}$$

- Left multiplication by Q_{n+1}^* does not change the norm (since both vectors are in the column space Q_{n+1}):

$$\left\|\tilde{H}_n y - Q_{n+1}^* b\right\| = \text{minimum}$$

- Finally, it is clear that $Q_{n+1}^* b = \|b\| e_1$:

$$\left\|\tilde{H}_n y - \|b\| e_1\right\| = \text{minimum}$$

The GMRES Algorithm

- High-level description of the algorithm:

Algorithm: GMRES

$q_1 := b/\|b\|$;
for $n := 1, 2, 3, \ldots$ **do**
  $<$ step n of Arnoldi iteration $>$
  Find y to minimize $\|\tilde{H}_n y - \|b\| e_1\| = \|r_n\|$;
  $x_n := Q_n y$

- The residual $\|r_n\|$ does not need to be computed explicitly from x_n
- Least squares problem has Hessenberg structure, solve with QR factorization of $\tilde{H}_n$ (computed by updating the factorization of $\tilde{H}_{n-1}$)
- Memory and cost grow with n restart the algorithm by clearing accumulated data (might stagnate the method)

Convergence of GMRES

- Two obvious observations based on the minimization in $\mathcal{K}_n$: GMRES converges monotonically and it converges after at most m steps,

$$\|r_{n+1}\| \leq \|r_n\| \wedge \|r_m\| = 0$$

- The residual $r_n = p_n(A)b$, where $p_n \in \mathbb{P}_n$ is a degree n polynomial with $p(0) = 1$, so GMRES also finds a minimizing polynomial:

$$\|p_n(A)b\| = \text{minimum}$$

- Based on this, diagonalizable $A = V\Lambda V^{-1}$ converges as:

$$\frac{\|r_n\|}{\|b\|} \leq \kappa(V) \inf_{p_n \in \mathbb{P}_n} \|p_n\|_{\Lambda(A)}$$

or in words: If A has well-conditioned eigenvectors, the convergence is based on how small polynomials p_n can be on the spectrum