CS6220: Data-sparse Matrix Computations
Lecture 10: Krylov methods: MINRES and GMRES

Lecturer: Anil Damle
Scribers: Kun Dong and Eston Schweickart

October 2, 2017

1 Convergence of the Conjugate Gradient Method

In the previous lecture, the Conjugate Gradient (CG) method and its convergence properties were discussed. In particular, it was shown that, given a symmetric positive-definite matrix A and a right-hand side b, then at iteration k of CG, the following error bound holds:

$$\frac{\|e^{(k)}\|_A}{\|e^{(0)}\|_A} \leq 2\left(\frac{\sqrt{\kappa_2}-1}{\sqrt{\kappa_2}+1}\right)^k$$

where $e^{(k)} = x^{(k)} - A^{-1}b$ and κ_2 is the 2-norm condition number of A. This implies that, in order to obtain a solution to $Ax = b$ accurate to $O(\varepsilon)$, one would expect to perform $O(\log(\frac{1}{\varepsilon})\sqrt{\kappa_2})$ iterations of CG. However, this is an upper bound that assumes that the eigenvalues of A are not clustered; if clusters of eigenvalues appear, CG may converge faster.

2 An Introduction to MINRES and GMRES

MINRES and GMRES are iterative Krylov methods for solving $Ax = b$ in more general cases than CG. In particular, MINRES solves a linear system for symmetric A, whereas GMRES solves a linear system for nonsingular A. These work similarly to CG, but whereas CG minimized the objective $\min_{x\in\mathcal{K}_k(A;b)} \frac{1}{2}\|x - A^{-1}b\|_A^2$, these methods find a vector in the kth Krylov space that minimizes the residual, i.e., they solve $\min_{x\in\mathcal{K}_k(A;b)} \frac{1}{2}\|b - Ax\|_2^2$.

This constrained optimization problem is hard in general, but by making the substitution $x^{(k)} = V_k y^{(k)}$, we can lift the constraint and reframe the problem as $y^{(k)} = \text{argmin}_{y\in\mathbb{R}^k} \frac{1}{2}\|b - AV_k y\|_2^2$. This is now a least squares problem. This is fortunate because we have tools for solving least squares problems (e.g., QR or SVD). However, the matrix AV_k has N rows, which still makes this compuationally expensive for large A.

Fortunately, we can reframe the problem again in a way that significantly reduces the size of the least squares problem. The first vector of $\mathcal{K}_k(A;b)$ is $\frac{b}{\|b\|}$, so we can rewrite b as $V_k(\|b\|e_1)$ for any k. Let us assume momentarily that $A = A^T$; from the Lanczos algorithm, we can write AV_k as $V_{k+1}\tilde{T}_k$. Therefore, $\|b - AV_k y\|_2 = \left\|V_{k+1}(\|b\|e_1 - \tilde{T}_k y)\right\|_2$. In exact math, the columns of V_{k+1} are mutually orthogonal; this leaves us with the following least squares problem: $y^{(k)} = \text{argmin}_{y\in\mathbb{R}^k} \frac{1}{2}\left\|\|b\|e_1 - \tilde{T}_k y\right\|_2^2$. The size of this problem is $(k+1)\times k$, which is much more manageable. Note that in practice, the columns of V_{k+1} are not mutually orthogonal; even so, we know $\left\|V_{k+1}(\|b\|e_1 - \tilde{T}_k y)\right\|_2 \leq \|V_{k+1}\|_2 \left\|\|b\|e_1 - \tilde{T}_k y\right\|_2$, and that because the columns of V_{k+1} are normalized, $\|V_{k+1}\|_2$ is small (i.e., not much larger than 1, and never larger than $k+1$).

Now suppose $A \neq A^T$. In this case, the above discussion still applies if all instances of $\tilde{T}_k$ are replaced with $\tilde{H}_k$.

3 Concept of MINRES

In MINRES we start with a linear system (A, b) where A is symmetric but not necessarily positive definite. Here is the MINRES procedure [AG11],

MINRES Steps
Outline of components in a single MINRES step

1. Perform one Lanczos step for (A, b).
2. Update the QR factorization of $\widetilde{T}_k$.
3. Solve the least-square problem $y^{(k)} = \arg\min_y \left\| \, \|b\|e_1 - \widetilde{T}_k y \, \right\|_2^2$
4. Update solution $x^{(k)} = V_k y^{(k)}$ and check stopping criteria.

Now consider the cost of MINRES method. Step 1 takes one matrix-vector multiplication, same as CG. Step 2 requires us to maintain a QR factorization of the tri-diagonal $\widetilde{T}_k$, but it can be updated from previous step using Givens rotation in $O(1)$ time. We can take advantage of this decomposition to solve the least-square problem in step 3, and update $x^{(k+1)}$ from $x^{(k)}$ in $O(N)$ time with no dependency on k.

MINRES is slightly more expensive than CG due to those technical details, but one of its advantages is that we get residual for free. Let x^* be the solution to the linear system. Similar to CG where $\|x - x^*\|_A$ is monotone-decreasing, MINRES has its residual $\|r_k\|_2 = \|Ax^{(k)} - b\|_2$ is also monotone-decreasing.

4 Concept of GMRES

When A in the linear system is only nonsingular but not necessarily symmetric, we use GMRES method which is conceptually very similar to MINRES. Instead of the Lanczos iterations in symmetric case, we will apply Arnoldi iterations for the non-symmetric A. The outline remains almost identical,

GMRES Steps
Outline of components in a single GMRES step

1. Perform one Arnoldi step for (A, b).
2. Update the QR factorization of $\widetilde{H}_k$.
3. Solve the least-square problem $y^{(k)} = \arg\min_y \left\| \, \|b\|e_1 - \widetilde{T}_k y \, \right\|_2^2$
4. Update solution $x^{(k)} = V_k y^{(k)}$ and check stopping criteria.

Step 1 again costs one matrix-vector multiplication. However, we now need to store results of all previous steps (i.e. V_k) since we no longer have the recurrence relation for symmetric case. Hence the storage cost is $O(Nk)$ at k-th iteration. In step 2 we need to update the QR factorization for an upper Hessenberg matrix $\widetilde{H}_k$ instead of the tri-diagonal $\widetilde{T}_k$, resulting in $O(k)$ computation. Consequently, the computational complexity in going from $x^{(k)}$ to $x^{(k+1)}$ now depends on k as well for GMRES.

Notice that unlike CG and MINRES, GMRES has a significant memory cost because it needs to store all Krylov basis vectors. As the number of iterations grows for system that is either very large or has slow convergence, this cost can becomes too expensive. The most popular answer to this difficulty is using restart GMRES. In this case we would throw away the stored V_k after a small number of iterations, modify our initial guess (so far we have only considered using an initial guess of 0) and restart the GMRES steps. The maximum number of iterations m is usually taken to be very small (i.e. $m = 10$), thereby we can even use more accurate methods to maintain the QR decomposition. The price of the restart is slower convergence and the loss of optimality criterion.

However, restarted `GMRES` is almost always preferred in implementations as a practical method for large-scale problems.

5 QR Update for $\widetilde{H}_k$

A key ingredient that allows us to efficiently solve the least-square problems in step 3 of both `MINRES` and `GMRES` is the quick update on the QR factorization of the upper Hessenberg matrix $\widetilde{H}_k$ (tridiagonal for `MINRES`). Specifically, we leverage the existing decomposition for $\widetilde{H}_{k-1}$ and Givens rotation to compute the update in $O(1)$ for `MINRES` $O(k)$ for `GMRES`. To see this, we first write down the decomposition at step $k-1$.

$$\widetilde{H}_{k-1} = Q_{k-1} \begin{bmatrix} R_{k-1} \\ 0 \end{bmatrix}$$

where $Q_{k-1} \in \mathbb{R}^{k \times k}$ is orthogonal and $R_{k-1} \in \mathbb{R}^{(k-1)\times(k-1)}$ is upper triangular. When we proceed to the next Lanczos step,

$$\widetilde{H}_k = \begin{bmatrix} \widetilde{H}_{k-1} & h_k \\ 0 & h_{k+1,k} \end{bmatrix}, \quad h_k \in \mathbb{R}^{k \times 1}, \ h_{k+1,k} \in \mathbb{R}$$

Since $\widetilde{H}_{k-1}$ is a sub-matrix, we can apply the previous orthogonal transformation to get

$$\begin{bmatrix} Q_{k-1}^T & 0 \\ 0 & 1 \end{bmatrix} \widetilde{H}_k = \begin{bmatrix} R_{k-1} & \tilde{h}_{k-1} \\ 0 & \tilde{h}_{k,k} \\ 0 & h_{k+1,k} \end{bmatrix}, \quad Q_{k-1}^T h_k = \begin{bmatrix} \tilde{h}_{k-1} \\ \tilde{h}_{k,k} \end{bmatrix}$$

Notice that to get R_k, we only need to apply one Givens rotation to turn $h_{k+1,k}$ into 0. In particular, the rotation we need is

$$G_k = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & c_k & s_k \\ 0 & -s_k & c_k \end{bmatrix}, \quad c_k = \frac{\tilde{h}_{k,k}}{\sqrt{\tilde{h}_{k,k}^2 + h_{k+1,k}^2}}, s_k = \frac{h_{k+1,k}}{\sqrt{\tilde{h}_{k,k}^2 + h_{k+1,k}^2}}$$

and $Q_k = \begin{bmatrix} Q_{k-1}^T & 0 \\ 0 & 1 \end{bmatrix} G_k^T$.

During the update, we need to compute $Q_{k-1}^T h_k$ which is $O(k)$ for `GMRES`. However, in `MINRES` we can take advantage of the fact that $\widetilde{H}_{k-1}$ is tridiagonal. This implies $\tilde{h}_{k-1}$ is 0. To get $\tilde{h}_{k,k}$, we realize there is exactly one Givens rotation that involves k-th row, thereby it can be obtained in $O(1)$.

6 Iterative Refinement

We have been using $x^{(0)} = 0$ as the initial solution for both `MINRES` and `GMRES`. Here we explore the case when we have a good guess for the solution of the linear system.

For iterative refinement, we start with a initial solution $x^{(0)}$ which could be non-zero and compute its residual $r = b - Ax^{(0)}$. Then we solve the system $Ad = r$ with $d^{(0)} = 0$ and let $x = x^{(0)} + d$. It can be quickly observed that

$$Ax = A(x^{(0)} + d) = (b - r) + Ad = b$$

Of course this process can be repeated on the new residual, leading to more and more accurate solutions to the original system. For a description of the iterative refinement procedure,

Iterative Refinement

Starting with a initial solution x_0, for the m-th step

1. Calculate residual $r_m = b - Ax_m$.
2. Solve the linear system $Ad_m = r_m$.

3. Update $x_m = x_{m-1} + d_m$.

7 Convergence of MINRES and GMRES

To analyze the convergence of MINRES and GMRES, we again look at those methods from the perspective of polynomial approximation, like we did for CG.

Let $\mathcal{P}_k = \{\text{polynomial } p : \deg(p) \leq k,\ p(0) = 1\}$. We can say that both methods find $x^{(k)} = q(A)b$ for some polynomial q of degree $(k-1)$. We can write down the residual as

$$r_k = b - Ax^{(k)} = [I - Aq(A)]b$$

Let $p(z) = 1 - zq(z)$, then $p \in \mathcal{P}_k$ and $r_k = p(A)b$. In MINRES and GMRES, we look for $p \in \mathcal{P}_k$ such that $\|p(A)b\|_2$ is minimized. This implies

$$\frac{\|r_k\|_2}{\|b\|_2} \leq \inf_{p \in \mathcal{P}_k} \|p(A)\|_2 \tag{1}$$

If we assume A is diagonalizable, then let $A = X\Lambda X^{-1}$ be the eigen-decomposition,

$$\|p(A)\|_2 \leq \|X\|_2 \|X^{-1}\|_2 \|p(\Lambda)\|_2 = \kappa_2(X) \max_{\lambda \in \Lambda(A)} |p(\lambda)|$$

Here $\Lambda(A)$ stands for the set of eigenvalues of A. Combine this inequality with Eq. 1, we obtain the min-max formulation

$$\frac{\|r_k\|_2}{\|b\|_2} \leq \kappa_2(X) \inf_{p \in \mathcal{P}_k} \sup_{\lambda \in \Lambda(A)} |p(\lambda)|$$

If A is normal ($AA^T = A^TA$), in which case X is orthogonal, $\kappa_2(X) = 1$.

8 Krylov Methods for Eigenvalue Problems

8.1 Introduction

Suppose we have a real symmetric matrix A, and we would like to know its extremal eigenvalues. One option is to use the power method, but unless there is a significant gap between A's two highest eigenvalues, the method converges very slowly. One reason for this slow convergence is that we are throwing information away at each iteration; by keeping information from past iterations around, we can improve the method's performance. Fortunately, the Lanczos method allows us to use this information without explicitly storing all vectors (at least on paper; a practical method requires more bookkeeping than the exact arithmetic implies). The following is described in more detail in [GVL13].

Let $r_A(x)$ denote the Rayleigh quotient, defined as

$$r_A(x) = \frac{x^T A x}{x^T x}, \qquad x \neq 0.$$

It can be shown from the Courant-Fischer Minimax Theorem that the minimum eigenvalue of A is $\min_{x \neq 0} r_A(x)$, and the maximum is $\max_{x \neq 0} r_A(x)$, in which case x is the associated eigenvector. To attack this problem, we could restrict our search space to a Krylov subspace, that is, $\max_{0 \neq x \in \mathcal{K}_k(A;q_1)} r_A(x)$ for some arbitrary nonzero q_1. Once again, by making the substitution $x^{(k)} = V_k y^{(k)}$, we can simplify the optimization:

$$r_A(x^{(k)}) = r_A(V_k y^{(k)}) = \frac{(V_k y^{(k)})^T A (V_k y^{(k)})}{(V_k y^{(k)})^T (V_k y^{(k)})} = \frac{y^{(k)T} T_k y^{(k)}}{y^{(k)T} y^{(k)}} = r_{T_k}(y^{(k)}),$$

that is, the best approximating maximal or minimal eigenvalue incurred by a vector in the span of the kth Krylov subspace is the corresponding eigenvalue of the tridiagonal matrix T_k.

It can be shown that as k increases towards n, the maximum eigenvalue of T_k increases monotonically towards the maximum eigenvalue of A, reaching equality when $k = n$. Similarly, the minimum eigenvalue of T_k decreases monotonically towards the minimum eigenvalue of A.

In addition, this method converges faster than the power method. Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of A in decreasing order, and let $\theta_1, \ldots, \theta_k$ denote the eigenvalues of T_k. Then:

$$\theta_1 \geq \lambda_1 - (\lambda_1 - \lambda_n) \left(\frac{\tan(\phi_1)}{c_{k-1}(1 + 2\rho_1)} \right)^2$$

where $\cos(\phi) = |v_1^T x_1|$, x_1 is the eigenvector corresponding to λ_1, c_{k-1} is the Chebychev polynomial of degree $k-1$, and $\rho_1 = \frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}$. A similar (and clearly worse) bound can be found for the power method, which yields an estimate γ_1 for the largest eigenvalue at iteration k:

$$\gamma_1 \geq \lambda_1 - (\lambda_1 - \lambda_n) \tan^2(\phi_1) \left(\frac{\lambda_2}{\lambda_1} \right)^{2(k-1)}.$$

Similar bounds exist for the minimum eigenvalues.

Thus, finding the extremal eigenvalues of the small tridiagonal matrix T_k gives good estimates of the corresponding eigenvalues of A. In theory, one could run the Lanczos procedure for a number of iterations, and then reduce T_k to diagonal form, or use other means to obtain the eigenvalue estimates. In practice, this method is not as accurate as expected.

8.2 Loss of Orthogonality

As hinted in lecture, part of the problem with the Lanczos procedure is that the columns of V_k are not necessarily mutually orthogonal as the method progresses. It can be shown that, for $i = 1, \ldots, k$:

$$|\hat{v}_{k+1}^T \hat{v}_i| \approx \frac{|\hat{r}_k^T \hat{v}_i| + \varepsilon \|A\|_2}{|\hat{\beta}_k|}$$

where hats denote computed (not mathematically exact) values, $\hat{v}_i$ is the ith column of $\hat{V}_k$, ε is machine precision, and $\hat{r}_k = A\hat{v}_k - \hat{\alpha}_k \hat{v}_k - \hat{\beta}_{k-1} \hat{v}_{k-1}$ is the residual. Note that this loss of orthogonality is not an accumulation of error; when $|\hat{\beta}_k|$ is small, the columns of $\hat{V}_{k+1}$ may lose orthogonality even if $\hat{r}_k^T \hat{V}_k = 0$. If left unchecked, these errors may cause the eigenvalues of $\hat{T}_k$ to significantly differ from those of T_k.

8.3 Practical Considerations

One option is to fall back to the Arnoldi method; that is, each time we calculate a new basis vector for the Krylov subspace, we orthogonalize it against all basis vectors. This is known as *complete reorthogonalization*. While effective, this method requires explicit storage of V_k, and the cost of adding another basis vector is $O(nk)$; both of these considerations make this method expensive if n is very large.

An alternative method is known as *selective reorthogonalization* [PS79]. A brief sketch of the method follows: It is observed that the error in $\hat{T}_k$ can be characterized by a number of Ritz pairs, $\{\hat{\theta}, \hat{y}\}$, where each $\hat{\theta}$ is an eigenvalue of $\hat{T}_k$ and the corresponding $\hat{y} \in \mathbb{R}^n$ is the matching eigenvector multiplied by $\hat{V}_k$. We can form a set of so-called "good" Ritz vectors that satisfy $\|A\hat{y} - \hat{\theta}\hat{y}\|_2 \leq \sqrt{\varepsilon}\|A\|_2$. Typically, the number of good Ritz vectors needed to represent error due to loss of orthogonalization is much less than k. Therefore, each time a new Krylov basis vector v_{k+1} is computed, it is orthogonalized against each good Ritz vector. Of course, we need to know when to add a new Ritz vector to our set. We can do this by estimating $\|I_{k+1} - \hat{V}_{k+1}^T \hat{V}_{k+1}\|_2$ (which, with the right bookkeeping, can be done relatively cheaply); when this value grows too large, the set of good Ritz vectors can be expanded.

References

[AG11] Uri M Ascher and Chen Greif. *A first course on numerical methods*. SIAM, 2011.

[GVL13] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 4. JHU Press, 2013.

[PS79] Beresford N Parlett and David S Scott. The lanczos algorithm with selective orthogonalization. *Mathematics of computation*, 33(145):217–238, 1979.