2022-03-18

1 The Big Picture

We now start with our discussion of Krylov subspace methods in general, and the famous method of conjugate gradients (CG) [1] in particular. Though this is the iterative method of choice for most positive definite systems, it may be as famously confusing as it is famous[2]. In order to avoid getting lost in the weeds, it seems worthwhile to start with a roadmap:

- We begin with the observation that if vectors $x^{(0)}, \ldots, x^{(k)}$ are increasingly good approximations to x, then some linear combination of these vectors may produce an even better approximation. If the original sequence is produced by a stationary iteration, these vectors span a *Krylov subspace.*

- One can generally show that a big enough Krylov subspace will contain a good approximation to x. Alas, this does not tell us how to find which vector in the space is best (or even good)! Attempting to minimize the norm of the error is usually impossible, but it is possible to minimize the residual (leading to GMRES or MINRES), an energy function (CG), or some other error-related quantity.

- The basic framework of a Krylov subspace plus a method of choosing approximations from the space allows us to describe some theoretical properties of several iterations without telling us why (or if) we can implement them efficiently and stably. A key practical point is the computation of well-conditioned bases for the Krylov subspaces, e.g., using the *Lanczos* algorithm (symmetric case) or the *Arnoldi* algorithm (nonsymmetric case).

[1] In lecture, we analyzed the GMRES method instead of CG.

[2] See, e.g., "Introduction to the Conjugate Gradient Method Without the Agonizing Pain" by Jonathan Shewchuk.

2 From Stationary Methods to Krylov Subspaces

Earlier in these notes, we tried to motivate the idea that we can improve the convergence of a stationary method by replacing the sequence of guesses

$$x^0, x^1, \ldots \to x$$

with *linear combinations*

$$\tilde{x}^k = \sum_{j=1}^{k} \alpha_{kj} x^j.$$

We could always choose α_{kj} to be one for $k = j$ and zero otherwise, in which case we have the original stationary method; but by choosing the coefficients more carefully, we might do better.

We've so far written stationary methods as

$$Mx^{j+1} = Nx^j + b.$$

This is equivalent to

$$x^{j+1} = x^j + M^{-1} r^j, \quad r^j \equiv b - Ax^j,$$

or

$$x^{j+1} = Rx^j + M^{-1} b$$

where $R = I - M^{-1}A$ is the iteration matrix we've seen in our previous analysis. If $x^0 = M^{-1}b$, this gives

$$x^j = \sum_{i=1}^{j} R^i M^{-1} b.$$

If we look at this expression closely, we might notice that the space spanned by the first k iterates of the stationary method is all vectors of the form

$$\sum_{i=0}^{j} c_i R^i M^{-1} b.$$

If we look a little harder, we might observe that this is equivalent to the space of all vectors of the form

$$\sum_{i=0}^{j} c_i (M^{-1}A)^i M^{-1} b = p(M^{-1}A) M^{-1} b$$

where $p(z) = \sum_{i=1}^{j} c_i z^i$ is a polynomial of degree at most j.

In general, the d-dimensional Krylov subspace generated by a matrix A and vector B is

$$\begin{aligned} \mathcal{K}_d(A, b) &= \operatorname{span}\{b, Ab, A^2 b, \ldots, A^{d-1} b\} \\ &= \{p(A)b : p \in \mathcal{P}_{d-1}\}. \end{aligned}$$

As we have just observed, the iterates of a stationary method form a basis for nested Krylov subspaces generated by $M^{-1}A$ and $M^{-1}b$. If the stationary method converges, we know the Krylov subspaces will eventually contain pretty good approximations to $A^{-1}b$. Let's now spell this out a little more carefully.

3 The Power of Polynomials

We showed a moment ago that the first m iterates of a stationary method form a basis for the space

$$\mathcal{K}_{m+1}(R, M^{-1}b) = \mathcal{K}_{m+1}(M^{-1}A, M^{-1}b)$$

What can we say about the quality of this space for approximation? As it turns out, this turns into a question about polynomial approximations. We will not spell out all the details (nor will this appear on any exams or homework problems for this class), but it's worth spending a few moments giving an idea of what happens.

We have seen that the iterates of the stationary method are

$$x^{(k)} = x + e^{(k)} = x + R^k e^{(0)}$$

We would like to take a linear combination

$$\tilde{x}^{(m)} = \sum_{k=0}^{m} \gamma_{mk} x^{(k)} = p_m(1)x + p_m(R)e^{(0)}$$

where $p_m(z) = \sum_{k=0}^{d} \gamma_{mk} z^k$. Moreover, if R is diagonalizable with $R = V \Lambda V^{-1}$, then

$$p_m(R) = V p(\Lambda) V^{-1}.$$

For any p_m with $p_m(1) = 1$, we have

$$\begin{aligned}
\|\tilde{e}^{(m)}\| &= \|\tilde{x}^{(m)} - x\| \\
&= \|p_m(R)e^{(0)}\| = \|Vp(\Lambda)V^{-1}e^{(0)}\| \\
&\leq \kappa(V) \max_{\lambda_j} |p(\lambda_j)| \|e^{(0)}\|.
\end{aligned}$$

Hence, we would really like to choose the polynomial that is one at 1 and as small as possible on each of the eigenvalues.

If all eigenvalues λ_j of R are real, then we have

$$\max_{\lambda_j} |p(\lambda_j)| \leq \max_{|z|<\rho(R)} |p(z)|,$$

and a reasonable way to choose polynomials is to minimize $|p_m(z)|$ on $[-\rho(R), \rho(R)]$ subject to the constraint $p_m(1) = 1$. The solution to this problem is the *scaled Chebyshev polynomials*, with which we can show that the optimal p_m satisfies

$$\begin{aligned}
p_m(z) &\leq \frac{2}{1 + m\sqrt{2/(1-\rho(R))}} \\
&= 2(1 - m\sqrt{2(1-\rho(R))}) + O(m^2(1-\rho(R)).
\end{aligned}$$

While the number of steps for the basic stationary iteration to reduce the error by a fixed amount scales roughly like $(1-\rho(R))^{-1}$, the number of steps to reduce the bound on the optimal error scales like $(1-\rho(R))^{-1/2}$.

While the Chebyshev bounds are correct, and involve a beautiful bit of approximation theory, they are limited. For one thing, they fall apart on non-normal matrices with complex spectra. Even in the SPD case, these bounds are often highly pessimistic in practice. When the eigenvalues of R come in clusters, a relatively low degree polynomial can do a good job of approximating λ^{-1} at each of the clusters, and hence a relatively low-dimensional Krylov subspace may provide excellent solutions.

All of this is to say that the detailed convergence theory for Krylov subspace methods can be quite complicated, but understanding a little about the eigenvalues of the problem can provide at least a modicum of insight.

For theoretical work, we are fine writing Krylov subspaces as polynomials in R applied to $M^{-1}b$. In practical computations, though, we need a basis. Because the iterates of the stationary method are converge to x, they tend to form a very ill-conditioned basis. We would like to keep the same Krylov subspace, but have a different basis – say, for instance, an orthonormal basis. We turn to this task next.

4 The Lanczos Idea

What is good about the “power basis” for a Krylov subspace? That is, why might we like to write

$$\mathcal{K}_m(A,b) = \operatorname{span}\{b, Ab, A^2b, \ldots, A^{m-1}b\}$$

rather than choosing a different basis? Though it’s terrible for numerical stability, there are two features of the basis that are attractive:

- The power bases span nested Krylov subspaces. Given the vectors $b, \ldots, A^{m-1}b$ spanning $\mathcal{K}_m(A,b)$, we only need one more vector ($A^d b$) to span $\mathcal{K}_{m+1}(A,b)$.

- Each successive vector can be computed from the previous vector with a single multiplication by A. There is no other overhead.

While we dislike the power basis from the perspective of stability, we would like to keep these attractive features for any alternative basis.

We’ve already described one approach to converting the vectors in a power basis into a more convenient set of vectors. Define the matrix

$$X^{(m)} = \begin{bmatrix} b & Ab & A^2b & \ldots & A^{m-1}b \end{bmatrix}$$

and consider the economy QR decomposition

$$X^{(m)} = Q^{(m)}R^{(m)}, \quad Q^{(m)} = \begin{bmatrix} q_1 & q_2 & \ldots & q_m \end{bmatrix}.$$

The columns of $Q^{(m)}$ are orthonormal and for any $k \leq m$, the first k columns of $Q^{(m)}$ span the same space as the first k columns of $X^{(m)}$. But forming $X^{(m)}$ and running QR is unattractive for two reasons. First, a dense QR decomposition may involve a nontrivial amount of extra work, particularly as m gets large; and second, simply forming the rounded version of $X^{(m)}$ is enough to get us into numerical trouble, even if we were able to run QR with no additional rounding errors. We need a better approach.

One helpful observation is that

$$\mathcal{R}(AQ^{(k)}) = \mathcal{R}(AX^{(k)}) \subseteq \mathcal{R}(X^{(k+1)}) = \mathcal{R}(Q^{(k+1)}).$$

That is, Aq_k can always be written as a linear combination of $q_1, \ldots, q_{k+1}$. In matrix terms, this means we can write

$$AQ^{(k)} = Q^{(k+1)}\bar{H}^{(k)}$$

where $\bar{H}^{(k)}$ is a rectangular upper Hessenberg matrix

$$\bar{H}^{(m)} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \dots & h_{1,k-1} & h_{1k} \\ h_{21} & h_{22} & h_{23} & & h_{2,k-1} & h_{2k} \\ 0 & h_{32} & h_{33} & & h_{3,k-1} & h_{3k} \\ & 0 & h_{43} & & h_{4,k-1} & h_{4k} \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & 0 & h_{k,k-1} & h_{kk} \\ & & & & 0 & h_{k+1,k} \end{bmatrix}.$$

Alternately, we write

$$AQ^{(k)} = Q^{(k)}H^{(k)} + q_{k+1}h_{k+1,k}$$

where $H^{(k)}$ is the square matrix consisting of all but the last row of $\bar{H}^{(k)}$. This formula is sometimes called an *Arnoldi decomposition*; it turns out to be crucial in the development of GMRES, one of the most popular iterative solvers for nonsymmetric linear systems. For the moment, though, we want to focus on the symmetric case, and so we will move on.

When A is symmetric, we have that

$$(Q^{(k)})^T AQ^{(k)} = H^{(k)}$$

is also symmetric, as well as being upper Hessenberg; in this case, the matrix $H^{(k)}$ is actually *tridiagonal*, and we write $H^{(k)}$ as $T^{(k)}$ to emphasize this fact. Conventionally, $T^{(k)}$ is written as

$$T^{(k)} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{bmatrix}.$$

Converting from matrix notation back into vector notation, we have

$$Aq_k = \beta_{k-1}q_{k-1} + \alpha_k q_k + \beta_k q_{k+1},$$

which we can rearrange as

$$\beta_k q_{k+1} = Aq_k - \alpha_k q_k - \beta_{k-1}q_{k-1}.$$

where

$$\begin{aligned} \alpha_k &= q_k^T A q_k \\ \beta_{k-1} &= q_k^T A q_{k-1} = q_{k-1}^T A q_k. \end{aligned}$$

Putting everything together, we have the *Lanczos iteration*, in which we obtain each successive vector q_{k+1} by forming Aq_k, orthogonalizing against the q_k and q_{k-1} by Gram-Schmidt, and normalizing. We saw this iteration earlier in the semester when we talked about eigenvalue problems and compared different methods of matrix tridiagonalization.

Presented as a *fait accompli*, the Lanczos iteration looks like magic. It seems even more like magic when one realizes that despite an instability in the iteration (because of the use of Gram-Schmidt for orthonormalization at each step), the iteration still produces useful information in floating point. The Lanczos iteration is the basis for one of the most popular iterative methods for solving eigenvalue problems, and in that setting it is important to acknowledge and deal with the instability in the method. For the moment, though, we are still interested in solving linear systems, and the method of Conjugate Gradients (also built on Lanczos) turns out to still work great.

4.1 Addendum: Three-Term Recurrences

The Lanczos iteration allows us to generate a sequence of orthonormal vectors using a three-term recurrence. As it turns out, the same approach leads to three-term recurrences that generate families of orthogonal polynomials, including the Chebyshev polynomials mentioned in passing earlier and the Legendre polynomials that play a significant role in the development of Gaussian quadrature. I consider the details beyond of these connections to be beyond the scope of the current class. But the connections are too beautiful and numerous to not mention that they exist. I would hate for you to walk away from this class with the impression that the mathematical development of the Lanczos iteration is only some quirky trick in numerical linear algebra that gets you part of the way to CG.

5 From Lanczos to CG

We developed the Lanczos iteration, which for a symmetric matrix A implicitly generates the decomposition

$$AQ^{(k)} = Q^{(k)}T^{(k)} + \beta_k q_{k+1}$$

where $T^{(k)}$ is a tridiagonal matrix with $\alpha_1, \ldots, \alpha_k$ on the diagonal and $\beta_1, \ldots, \beta_{k-1}$ on the subdiagonal and superdiagonal. The columns of $Q^{(k)}$ form an orthonormal basis for the Krylov subspace $\mathcal{K}_k(A, b)$, and are a numerically superior alternative to the power basis. We now turn to using this decomposition to solve linear systems.

The *conjugate gradient* algorithm can be characterized as a method that chooses an approximation $\tilde{x}^{(k)} \in \mathcal{K}_k(A, b)$ by minimizing the energy function

$$\phi(z) = \frac{1}{2} z^T A z - z^T b$$

over the subspace. Writing $\tilde{x}^{(k)} = Q^{(k)}u$, and using the fact that

$$\begin{aligned} (Q^{(k)})^T A Q^{(k)} &= T^{(k)} \\ (Q^{(k)})^T b &= \|b\| e_1 \end{aligned}$$

we have

$$\phi(Q^{(k)}u) = \frac{1}{2} u^T T^{(k)} u - \|b\| u^T e_1.$$

The stationary equations in terms of u are then

$$T^{(k)} u = \|b\| e_1.$$

In principle, we could solve find the CG solution by forming and solving this tridiagonal system at each step, then taking an appropriate linear combination of the Lanczos basis vectors. Unfortunately, this would require that we keep around the Lanczos vectors, which eventually may take quite a bit of storage. This is essentially what happens in methods like GMRES, but for the method of conjugate gradients, we have not yet exhausted our supply of cleverness. It turns out that we can derive a short recurrence relating the solutions at consecutive steps and their residuals. There are several different ways to this recurrence: one can work from a factorization of the nested tridiagonal matrices $T^{(k)}$, or work out the recurrence based on the optimization interpretation of the problem (this leads to the name "conjugate gradients"). For a detailed discussion, my preferred reference is *Templates for the Solution of Linear Systems*, a short book available from SIAM which is also freely available online. The paper "Introduction to the Conjugate Gradient Method Without the Agonizing Pain" provides a longer and possibly gentler introduction.