Week 13: Wednesday, Nov 18

Re-orthogonalization

As we discussed last time, the Lanczos procedure in floating point behaves differently from Lanczos in exact arithetic, leading to "ghost" eigenvalues. Because we are doing a (limited) version of Gram-Schmidt, we expect a loss of orthogonality as we go on. An alternative is to use *complete* orthogonalization: compute Aq_j and then orthogonalize against all previous vectors (e.g. by a sequence of Householder transformations). This is expensive in terms of storage (and data movement) and arithmetic, but it does work. A less expensive solution, *selective* orthogonalization, involves orthogonalizing against converged eigenvector estimates (Ritz vectors).

In the interest of time, we will not discuss reorthogonalization further. If you are interested, though — or if you are interested in any number of other aspects of Lanczos iteration that we skip over — I recommend looking at Parlett's book *The Symmetric Eigenvalue Problem*. Stewart's book *Matrix Algorithms, Vol 2: Eigensystems* also has a nice treatment.

Partial tridiagonalization and residual bounds

Suppose $AQ = QT$, where $T_{ii} = \alpha_i$, $T_{i,i+1} = \beta_i$. If we take m steps of Lanczos iteration, we generate $Q_{:,1:m} = \begin{bmatrix} q_1 & q_2 & \dots & q_m \end{bmatrix}$ as well as the first m coefficients $(\alpha_i)_{i=1}^m$ and $(\beta_i)_{i=1}^m$. Let us denote Q_m and T_m as the leading m columns of Q and the leading $m \times m$ submatrix of T respectively; then writing the first m columns of AQ and QT gives us

$$AQ_m = Q_m T_m + \beta_m q_{m+1} e_m^T. \tag{1}$$

Here T_m is a block Rayleigh quotient $T_m = Q_m^T A Q_m$, and the eigenvalues of T_m (the *Ritz values*) are used to approximate the eigenvalues of A. Now consider the eigendecomposition $T_m = Y \Theta Y^T$ where $Y^T Y = I$ and $\Theta = \mathrm{diag}(\theta_1, \dots, \theta_m)$. Then postmultiplying (1) by Y gives

$$AQ_m Y = Q_m Y \Theta + \beta_m q_{m+1} e_m^T Y.$$

The columns of $Z = Q_m Y = \begin{bmatrix} z_1 & \dots & z_m \end{bmatrix}$ are approximate eigenvalues corresponding to the approximate eigenvalues $\theta_1, \theta_2, \dots, \theta_m$. For each column,

we have

$$Az_k - z_k\theta_k = \beta_m q_{m+1} e_m^T y_k,$$

which means

$$\|Az_k - z_k\theta_k\|_2 = |\beta_m||e_m^T y_k|.$$

This is useful because, as we discussed before, in the symmetric case a small residual error implies a small distance to the closest eigenvalue. This is also useful because the residual error can be computed with no further matrix operations — we need only to look at quantities that we would already compute in the process of obtaining the tridiagonal coefficients and the corresponding Ritz values. In particular, note that we can compute the residual for the Ritz pair (approximate eigenpair) (z_k, θ_k) *without* explicitly computing z_k!

The polynomial connection

Suppose $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ are the eigenvalues of A, with corresponding eigenvectors w_1 through w_n. In matrix form, then,

$$A = W\Lambda W^T.$$

Now, suppose we run m steps of Lanczos iteration (in exact arithmetic), and let θ_1 be the largest eigenvector of T_m (the largest Ritz value). This is the same as saying that θ_1 maximizes the Rayleigh quotient over the Krylov subspace $\mathcal{K}_m(A, q_1)$:

$$\theta_1 = \max_{y \in \mathbb{R}^m} \frac{y^T T_m y}{y^T y} = \max_{y \in \mathbb{R}^m} \frac{(Q_m y)^T A (Q_m y)}{(Q_m y)^T (Q_m y)} = \max_{z \in \mathcal{K}_m(A,q_1)} \frac{z^T A z}{z^T z}.$$

Now, any vector $z \in \mathcal{K}_m(A, q_1)$ can be written as

$$z = c_0 q_1 + c_1 A q_1 + \ldots + c_{m-1} A^{m-1} q_1 = p(A) q_1,$$

where p is a polynomial of degree at most $m-1$. Thus,

$$\theta_1 = \max_{\deg(p)<m} \frac{q_1^T p(A) A p(A) q_1}{q_1^T p(A)^2 q_1}.$$

This is still somewhat awkward, so let us rewrite things in terms of the eigenvector basis. Define $d = Z^T q_1$; then

$$\theta_1 = \max_{\deg(p)<m} \frac{d^T p(\Lambda)\Lambda p(\Lambda) d}{d^T p(\Lambda)^2 d} = \max_{\deg(p)<m} \frac{\sum_{i=1}^n d_i^2 p(\lambda_i)^2 \lambda_i}{\sum_{i=1}^n d_i^2 p(\lambda_i)^2}.$$

Let's give a name to the function in this maximization:

$$\phi(p) = \frac{\sum_{i=1}^n d_i^2 p(\lambda_i)^2 \lambda_i}{\sum_{i=1}^n d_i^2 p(\lambda_i)^2} = \lambda_1 - \frac{\sum_{i=2}^n d_i^2 p(\lambda_i)^2 (\lambda_1 - \lambda_i)}{\sum_{i=1}^n d_i^2 p(\lambda_i)^2} \geq \lambda_1 - (\lambda_1 - \lambda_n) \frac{\sum_{i=2}^n d_i^2 p(\lambda_i)^2}{d_1^2 p(\lambda_1)^2}.$$

We know that $\theta_1 = \max_{\deg(p)<m} \phi(p) \leq \lambda_1$; we would also like a *lower* bound on θ_1. If we can show that this lower bound approaches λ_1 at some rate with increasing m, then we know θ_1 will converge to λ_1 at least as fast.

Note that if we found a polynomial that was *zero* at $\lambda_2, \ldots, \lambda_n$ and nonzero at λ_1, then we would recover λ_1 exactly. But such a polynomial usually has too high a degree. What we would like in order to get a bound, then, is a degree m polynomial which is not too big on $[\lambda_2, \lambda_n]$, but is relatively large at λ_2. A good candidate is a rescaled *Chebyshev polynomial*.

The Chebyshev polynomials appear frequently in approximation theory; just as linear combinations of trigonometric functions (Fourier series) are a natural tool in the approximation of periodic functions, linear combinations of Chebyshev polynomials (Chebyshev series) are a natural tool in the approximation of functions on bounded intervals. The Chebyshev polynomials are defined by the recurrence

$$c_0(x) = 1 \tag{2}$$

$$c_1(x) = x \tag{3}$$

$$c_{k+1}(x) = 2x c_k(x) - c_{k-1}(x). \tag{4}$$

Note that for any fixed x, the Chebyshev polynomials follow a constant coefficient second-order linear difference equation. We analyze such difference equations in the same way we analyze constant coefficient second-order differential equations: in terms of the roots of the characteristic equation

$$\xi^2 - 2x\xi + 1 = 0.$$

Any solution to (4) can be written as $a_+ \xi_+^k + a_- \xi_-^k$, where $\xi_\pm = x \pm \sqrt{x^2 - 1}$ are roots of the characteristic equation. To satisfy the boundary conditions (2) and (3), we find

$$c_k(x) = \frac{1}{2} \left(\xi_-^k + \xi_+^k \right).$$

Notice that for $|x| < 1$, the roots $\xi_\pm$ are a complex conjugate pair of unit magnitude, which we can write as $\exp(\pm \imath\theta)$, and so $c_k(x) = \cos(k\theta) \in [-1, 1]$. In fact, for $x \in [-1, 1]$, we can write $c_k(x) = \cos(k \arccos(x))$. On the other

hand, for $x > 1$, we have a pair of real roots, one of which is smaller than one in magnitude, the other of which is larger than one; and so $c_k(x)$ grows exponentially as a function of k.

Therefore, the values of $c_k(x)$ remain bounded for $x \in [-1, 1]$ and become large elsewhere. The mapping

$$x \mapsto -1 + 2\frac{x - \lambda_n}{\lambda_2 - \lambda_n}$$

has the property that $[\lambda_2, \lambda_n]$ maps to $[-1, 1]$ and

$$-1 + 2\frac{\lambda_1 - \lambda_n}{\lambda_2 - \lambda_n} = 1 + 2\left(\frac{\lambda_1 - \lambda_n}{\lambda_2 - \lambda_n} - 1\right) = 1 + 2\left(\frac{\lambda_1 - \lambda_2}{\lambda_2 - \lambda_n}\right) = 1 + 2\rho_1,$$

where $\rho_1 = (\lambda_1 - \lambda_2)/(\lambda_2 - \lambda_n)$. Now define

$$\hat{c}_k(x) = c_k\left(-1 + \frac{x - \lambda_n}{\lambda_2 - \lambda_n}\right),$$

so that $|\hat{c}_k(\lambda_j)| < 1$ for $j > 1$ and $|\hat{c}_k(\lambda_1)|$ grows exponentially with k.

Note that for any p, we have

$$\begin{aligned}\phi(p) &= \lambda_1 - \frac{\sum_{i=2}^n d_i^2 p(\lambda_i)^2(\lambda_1 - \lambda_i)}{\sum_{i=1}^n d_i^2 p(\lambda_i)^2} \\ &\geq \lambda_1 - (\lambda_1 - \lambda_n)\frac{\sum_{i=2}^n d_i^2 p(\lambda_i)^2}{d_1^2 p(\lambda_1)^2}.\end{aligned}$$

Now, note that $\sum_{i=1}^n d_i^2 = \|d\|_2^2 = \|q_1\|_2^2 = 1$, that $\hat{c}_{m-1}(\lambda_i)^2 \leq 1$ for $i = 2, \ldots, n$ and that $\hat{c}_{m-1}(\lambda_1) = c_{m-1}(1 + 2\rho_1)$. Thus,

$$\phi(\hat{c}_{m-1}) \geq \lambda_1 - (\lambda_1 - \lambda_n)\frac{\sum_{i=2}^n d_i^2 \hat{c}_{m-1}(\lambda_i)^2}{d_1^2 \hat{c}_{m-1}(\lambda_1)^2} \geq \lambda_1 - (\lambda_1 - \lambda_n)\frac{1 - d_1^2}{d_1^2 c_{m-1}(1 + 2\rho_1)^2}.$$

The entry $d_1 = w_1^T q_1$ is the cosine of the angle ϕ_1 between the first eigenvector w_1 and the starting vector q_1; so we can write $(1 - d_1^2)/d_1^2 = \tan(\phi_1)^2$, which gives us the final bound

$$\lambda_1 \geq \theta_1 = \max_{\deg p < m} \phi(p) \geq \phi(\hat{c}_{m-1})\lambda_1 - \frac{(\lambda_1 - \lambda_n)\tan(\phi_1)^2}{c_{m-1}(1 + 2\rho_1)^2}.$$

We notice a few things from this bound. The rate of convergence (determined by $c_{m-1}(1 + 2\rho_1)^2$) is strictly better than the rate of convergence for

power iteration, which makes sense since we are using strictly more information in the Lanczos iteration than we use in the power iteration (we maximize over an m-dimensional subspace rather than a 1-dimensional space). Also, the trick involved is a good one: instead of reasoning about matrices, we were able to reason about polynomials. However, notice that this bound may be very pessimistic, since it does not take into account the distribution of eigenvalues in $[\lambda_2, \lambda_n]$. For example, if many eigenvalues of A cluster near zero (as happens with certain discretizations of compact operators, for instance), then we might get much better convergence than indicated by a basic bound that works for any distribution of eigenvalues between $[\lambda_2, \lambda_n]$.

We will see the same basic picture — bounds in terms of Chebyshev polynomials that turn out to be pessimistic when the eigenvalues of A are clustered — when we touch on the analysis of the method of conjugate gradients in a couple lectures.