CS 6210: Matrix Computations

Large-scale eigensolvers

David Bindel

2025-12-01

Lanczos and Arnoldi eigensolvers

The standard ingredients in all the subspace methods we have described so far are a choice of an approximation subspace (usually a Krylov subspace) and a method for choosing an approximation from the space. In the most common methods for large-scale eigensolvers, one uses a Krylov subspace together with a Bubnov-Galerkin condition for choosing approximate eigenpairs; that is, we choose $v \in \mathcal{V}$ such that

$$r = (A - \hat{\lambda} I)\hat{v} \perp \mathcal{V}.$$

In the symmetric case, this is equivalent to finding a constrained stationary point of the Rayleigh quotient (i.e. a $\hat{v}$ such that directional derivatives of $\rho_A(\hat{v})$ are zero for any direction in the space). This approximation scheme is known as the *Rayleigh-Ritz* method, and approximate eigenvectors and eigenvalues obtained in this way are often called *Ritz vectors* and *Ritz values*.

In the Lanczos method for the symmetric eigenvalue problem, we compute the Lanczos decomposition

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

and use it to compute the residual relation for an approximate pair $(\mu, Q_m y)$ by

$$r = (A - \mu I)Q_m y = Q_m(T_m - \mu I)y + \beta_m q_{m+1} e_m^T y.$$

The condition $Q_m^T r = 0$ gives us the *projected* problem

$$(T_m - \mu I)y = 0;$$

if we satisfy this condition, we have

$$r = \beta_m q_{m+1} y_m$$

and the residual norm (in the 2-norm) is $|\beta_m y_m|$. Generalizing, if we compute the eigendecomposition

$$T_m = Y \Theta Y^T,$$

we have the collected approximations $Z = Q_m Y_m$ with residuals

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

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 tridiangonal coefficients and the corresponding Ritz values.

The Arnoldi method for computing approximate eigenpairs similarly uses the Galerkin condition together with the Arnoldi decomposition to express an approximate partial Schur form. From the decomposition

$$AQ_m = Q_m H_m + h_{m+1,m} q_{m+1} e_m^T,$$

we write a subspace residual for $(Q_m Y, T)$ as

$$R = AQ_m Y - Q_m YT = Q_m(H_m Y - YT) + h_{m+1,m} q_{m+1} e_m^T.$$

Forcing $Q_m^T R = 0$ gives the projected problem

$$H_m Y = YT,$$

i.e. we seek a Schur decomposition of the (already Hessenberg) matrix H_m.

There are three main issues with the Lanczos and Arnoldi methods that we need to address in practical situations.

1. We must deal with forward instability, particularly in the case of the Lanczos method. Unless we are careful to maintain orthogonality between the computed Lanczos basis vectors, the method derails. The result is not that we get bad approximate eigenpairs; indeed, the forward instability is intimately tied to the very thing we want, which is convergence of eigenpairs. The real problem is that we get the same eigenpairs over and over again, a phenomenon known as "ghost" eigenvalue approximations. We deal with this issue by careful re-orthogonalization (selective or complete).

2. Because of the cost of storing a Krylov basis and maintaining its orthogonality, we typically only want to approximate a few eigenpairs at a time.

3. The Krylov subspace generated by A and some random start vector contains iterates of the power method applied to any A (or to $A - \sigma I$ for any shift σ — the Krylov subspace is shift-invariant). This is at least as good as power iteration for approximating the extremal parts of the spectrum, and we can use the same Chebyshev-based games we discussed before to give concrete (though typically pessimistic) convergence bounds. But if eigenvalues cluster, or if we are interested in eigenvalues that are not at the edge of the spectrum, then the convergence in theory and in practice can be painfully slow.

We address these issues with three basic techniques, all of which we have already seen in other contexts: *reorthogonalization*, *spectral transformation* and *restarting*.

Reorthogonalization

Both Lanczos and Arnoldi involve a Gram-Schmidt-style orthogonalization step – and, alas, one has potential loss of orthogonality when adding a new vector that is close to the span of the previous vectors. However, we can always take the result of a Gram-Schmidt process and apply Gram-Schmidt again – this is known as *re-orthogonalization*.

Re-orthogonalization can be an expensive thing to do constantly, particularly in the case of Lanczos (where we usually only bother to orthogonalize against two previous vectors). Therefore, Lanczos-based eigensolvers often use *selective re-orthogonalization*, triggered when there is evidence of significant cancellation. Without selective re-orthogonalization, Lanczos-based eigensolvers are prone to the phenomenon of "ghost eigenvalues": the solver keeps effectively restarting after convergence leads to a small off-diagonal, and hence re-converging to the same eigenpairs (as opposed to continuing to get new ones).

Spectral transformation

We have dealt with the notion of spectral transformation before, when we discussed the power iteration. The idea of spectral transformation is to work not with A, but with some rational $f(A)$ where f maps the eigenvalues of interest to the outside of the spectrum. Usually f is a rational function; common examples include

- *Shift-invert*: $f(z) = (z - \sigma)^{-1}$. Favors eigenvalues close to the shift σ.
- *Cayley*: $f(z) = (\sigma - z)(\sigma + z)^{-1}$. This maps the left half plane to the interior of the unit circle and the right half plane to the exterior; it is commonly used in stability analysis.
- *Polynomial*: Just what it sounds like.

In general, the shifted linear solves needed to carry out rational spectral transformations (e.g. shift-invert and Cayley) must be computed to rather high accuracy. Hence, we favor sparse direct methods. An alternate approach, similar to what we say when we briefly considered flexible GMRES, is to break out of the confines of using a Krylov subspace; the most popular variant here is the *Jacobi-Davidson* method.

Jacobi-Davidson

The *Jacobi-Davidson* iteration is an alternative subspace-based large-scale eigenvalue solver that does not use Krylov subspaces. Instead, one builds a subspace via steps of an inexact Newton iteration on the eigenvalue equation. Given an approximate eigenpair (θ, u) where θ is the Rayleigh quotient, we seek a correction $s \perp u$ so that

$$A(u+s) = \lambda(u+s).$$

Rewriting this in terms of $r = (A - \theta I)u$, we have for any approximate $\tilde{\lambda}$ to the desired eigenvalue

$$(A - \tilde{\lambda} I)s = -r + (\lambda - \theta)u + (\lambda - \tilde{\lambda})s.$$

Using the desiderata that $u^* s = 0$ and the fact that $(I - uu^*)u = 0$, we obtain the correction equation

$$(I - uu^*)(A - \tilde{\lambda} I)(I - uu^*)s = -r, \quad \text{where } s \perp u.$$

The method proceeds by at each step solving the correction equation approximately and extending the subspace by a new direction s. One then seeks an approximate eigenpair from within the subspace.

In addition to a proper choice of subspaces, one needs a method to extract approximate eigenvectors and eigenvalues. This is particularly important for approximating interior eigenvalues, as the standard Rayleigh-Ritz approach may give poor results there. One possible method is to use the *refined Ritz* vector, which is obtained by minimizing the residual over all candidate eigenvectors associated with an approximate eigenvalue $\tilde{\lambda}$. The refined Ritz vector may then be plugged into the Rayleigh quotient to obtain a new eigenvector. Another method is the *harmonic Rayleigh-Ritz* approach, which for eigenvalues near a target τ employs the condition

$$(A - \tau I)^{-1}\tilde{u} - (\tilde{\theta} - \tau)^{-1}\tilde{u} \perp \mathcal{V}.$$

We again usually use the Rayleigh quotient from the harmonic Ritz vector rather than the harmonic Ritz value $\tilde{\theta}$.

Restarting

A major difficulty with Arnoldi (and with Lanczos) is the need to store the basis vectors. If we run out of storage before we converge to the vectors we care about, what are we to do? A natural approach is to restart the iteration with a new vector — but if we choose starting vectors at random, we are unlikely to make much additional progress. So what vector to choose?

If we want to take advantage of computations that we have already done, it is natural to choose a starting vector from our current subspace. Recall that all vectors in a m-dimensional Krylov

subspace with starting vector u_1 can be written as $p(A)u_1$ for some $p \in \mathcal{P}_{m-1}$, and that if A is diagonalizable we have

$$p(A)u_1 = Vp(\Lambda)V^{-1}u_1.$$

Ideally, we would like a starting vector that is "rich" in the eigendirections that we want and not in the other directions. That is, we want p to be nearly zero at eigenvalues that are no longer of interest — either because they were never of interest or because we have already computed them accurately. Because p serves to "filter out" the directions that are not of interest, it is called a *filter polynomial*. A common choice is to let p be the polynomial whose roots are the Ritz values for the current Krylov subspace (the "exact shifts" strategy), but other choices are possible.

Implicit restarting

While we could restart the Arnoldi process from scratch, it's generally useful to keep around a few vectors (e.g. any eigenvectors that have already converged). The *implicit restarting* technique applies a filter polynomial of degree $m-k$ to reduce from an m-dimensional Krylov subpace to a k-dimentional filtered space generated by $p(A)q_1$ directly, without re-running the Arnoldi process explicitly for k steps. Suppose $p(z)$ is a filter polynomial with roots $\kappa_1, \ldots \kappa_{m-k}$; we consider applying the filter "one root at a time." Let κ_1 be the first root, and recall that the Arnoldi decomposition gives us

$$AU_m = U_m H_m + \beta_m u_{m+1} e_m^T.$$

Therefore,

$$(A - \kappa_1 I)U_m = U_m(H_m - \kappa_1 I) + \beta_m u_{m+1} e_m^T$$

Let $Q_1 R_1 = H_m - \kappa_1 I$; then

$$(A - \kappa_1 I)(U_m Q_1) = (U_m Q_1)(R_1 Q_1) + \beta_m u_{m+1}(e^T Q_1).$$

Therefore

$$AU_m^{(1)} = U_m^{(1)} H_m^{(1)} + \beta_m u_{m+1} b_{m+1}^{(1)T},$$

where

$$U_m^{(1)} = U_m Q_1, \quad H_m^{(1)} = R_1 Q_1 + \kappa_1 I, \quad b_{m+1}^{(1)T} = e_m^T Q_1.$$

In effect, what we have done is to apply one step of QR iteration to H_m to get a new factorization. Each time we do this, we modify the remainder term in the factorization; but that modification changes from something that only affects the last column to something that only affects the last two columns, three columns, etc. Therefore, after applying k shifts, the first $m-k$ columns still correspond to the leading part of an Arnoldi factorization generated by $p(A)u_1$. In practice, of course, we do not do explicit QR iteration steps to apply the filter: we use the same implicit QR logic discussed earlier in the semester.

Krylov-Schur

An Arnoldi decomposition has the form

$$AU_m = U_m H_m + \beta_m u_{m+1} e_m^T.$$

A slightly more general form is the *Krylov decomposition*

$$AV_m = V_m S_m + \beta_m u_{m+1} b_m^*.$$

The Krylov decomposition still represents an orthonormal basis for a Krylov subspace, but though potentially no longer a nested orthonormal basis. A benefit of working with a Krylov decomposition is that we can apply arbitrary unitary similarities to S_m and still have a Krylov decomposition. In particular, we can compute a Schur form to get

$$A \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} V_1 & V_2 & u \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \\ b_1^* & b_2^* \end{bmatrix}$$

which can be truncated to get a new Krylov decomposition

$$AV_1 = V_1 T_{11} + u b_1^*.$$

This is the basis for the *Krylov-Schur* restarting method for applying exact shifts: compute a Schur decomposition of matrix in a Krylov decomposition, sort the desired eigenvalues to the "front" of the decomposition, and then truncate. Krylov-Schur restarting also has some advantages in terms of rate of progress: the implicitly restarted Arnoldi process suffers a forward instability that can slow things down near convergence (though it doesn't affect overall stability of the algorithm).