Eigenvalue Solvers

Derivation of Methods

Similarity Transformation

Similarity transformation for A: BAB^{-1};
this can be done with any nonsingular B.

Let $Ax = \lambda x$, then $BAB^{-1}Bx = BAx = \lambda Bx$.

BAB^{-1} has the same eigenvalues as A, and eigenvectors Bx where x is an eigenvector of A.

Although any nonsingular B possible, most stable and accurate algorithms with orthogonal (unitary) matrix.

For example used in the QR algorithm.

Similarity Transformation

Orthogonal similarity transformation for A: $Q^* A Q$, where $Q^* Q = I$.

If $Q^* A Q = \begin{pmatrix} L_1 & F \\ 0 & L_2 \end{pmatrix}$, then $\mathcal{L}(A) = \mathcal{L}(L_1) \cup \mathcal{L}(L_2)$.

If we can find $Q \equiv \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ that yields such a decomposition we have reduced the problem to two smaller problems.

Moreover, $A Q_1 = Q_1 L_1$ and $\text{range}(Q_1)$ is invariant subspace.
Eigenpair $L_1 z = \lambda z$ gives eigenpair $A Q_1 z = Q_1 L_1 z = \lambda Q_1 z$.

Approximation over Search Space

For large matrices we cannot use full transformations. Often we do not need all eigenvalues/vectors. Look for proper basis Q_1 that captures relevant eigenpairs. We do not need Q_2.

Approximations over subspace $\mathrm{range}(Q_1)$: $L_1 = Q_1^* A Q_1$

When is an approximation good (enough)?

We will rarely find $AQ_1 - Q_1 L_1 = 0$ unless we do huge amount of work.
Not necessary. We are working with approximations and we must deal with numerical error anyway.

Approximation over Search Space

Let $AQ_1 - Q_1L_1 = R$ with $\|R\|$ small relative to $\|A\|$.

Now, $\left(A - RQ_1^*\right)Q_1 - Q_1L_1 = AQ_1 - R - Q_1L_1 = 0$.

$\mathrm{range}(Q_1)$ is exact invariant subspace of perturbed matrix, $\hat{A}$.
$\hat{A} = A - RQ_1^*$ and $\left\|\hat{A} - A\right\| / \|A\| = \|R\| / \|A\|$

If $\|R\| / \|A\|$ sufficiently small, then Q_1 is acceptable.
In fact, this is as good as we can expect (unless we're lucky).
Any numerical operation involves perturbed operands!
Note that we cannot always say that Q_1 is accurate.

Power method

So, what kind of methods lead to good Q_1?
In many cases matrices are sparse and/or it is possible to do the matrix-vector product, Ax, very cheap.

First consider very small subspace: 1 vector
Well-known method is the Power method.

$v^{(k+1)} = Av^{(k)} / \left\| Av^{(k)} \right\|$, $\lambda^{(k+1)} = v_j^{(k+1)} / v_j^{(k)}$ until convergence.

Converges to eigenpair with largest absolute eigenvalue, if unique. May converge very slowly if dominant eigenvalue not well-separated (next one is close).

Power method

Assume A diagonalizable: $A = X\Lambda X^{-1}$ and $|\lambda_1| > |\lambda_j|$

Decompose $v^{(0)}$ along eigenvectors: $v^{(0)} = \sum_j x_j \alpha_j$

$$v^{(k)} = A^k v^{(0)} = \sum_j x_j \lambda_j^k \alpha_j = \lambda_1^k \left(x_1 \alpha_1 + \sum_{j>1} x_j \left(\frac{\lambda_j}{\lambda_1} \right)^k \alpha_j \right)$$

So, $v^{(k)} \rightarrow x_1$ since $\lambda_j^k / \lambda_1^k \rightarrow 0$.

If $|\lambda_1| > |\lambda_2| \geq |\lambda_j|$, then $\frac{|\lambda_2|}{|\lambda_1|}$ determines rate of convergence.

Convergence may be slow if $\frac{|\lambda_2|}{|\lambda_1|} \approx 1$, and linearly at best.

Work with search space

Improve on the Power method by keeping all iterates.

Krylov space: $K^m(A,v) = \text{span}\left\{v, Av, A^2v, \ldots, A^{m-1}v\right\}$

Let $Q_m = \left[q_1\, q_2 \ldots q_m\right]$ give orthonormal basis for $K^m(A,v)$.

Approximate eigenpair $(\mu, Q_m w)$ is called Ritz pair if

$$AQ_m w - \mu Q_m w \perp Q_m \Leftrightarrow Q_m^* A Q_m w - \mu w = 0$$

We find Ritz pairs from eigenpairs (μ, w) of $H \equiv Q_m^* A Q_m$.

This is a small eigenvalue problem that we can solve using standard software (e.g., QR algorithm in LAPACK).

Arnoldi

Generate an orthogonal basis for the Krylov subspace:

```
$q_1 = v^{(0)} / \| v^{(0)} \|_2;$
for $k = 1 : m,$
    $\tilde{q} = A q_k;$
    for $j = 1 : k,$
        $h_{j,k} = q_k^* \tilde{q};$    $\tilde{q} = \tilde{q} - q_k h_{k,j};$
    $h_{k+1,k} = \| \tilde{q} \|_2;$    $q_{k+1} = \tilde{q} / h_{k+1,k};$
end
```

Accurate orthogonalization required: (partial) reorthogonalization

$AQ_m = Q_{m+1} \underline{H}_m,$

where $Q_{m+1} = \left[q_1 \;\; q_2 \;\; \cdots \;\; q_{m+1} \right]$ and $\underline{H}_m = \left[h_{j,k} \right]_{j=1:m+1, k=1:m}$

with $Q_{m+1}^* Q_{m+1} = I_{m+1}$ and $\mathrm{range}(Q_{m+1}) = K^{m+1}(A, v^{(0)})$.

Arnoldi

Compute m approximate eigenpairs from $H_m = Q_m^* A Q_m$:
$H_m y = \lambda y$ which gives approximate pairs: $(\lambda, Q_m y)$ of A.

To reduce costs we select after m steps k approximate eigenvectors, get new Q_k $(k < m)$, and discard the rest.
All Ritz vectors (Krylov space) have 'same' residual: q_{m+1}.

$$\begin{aligned} AQ_m y &= Q_{m+1}\underline{H}_m y = Q_m H_m y + q_{m+1} e_m^T y h_{m+1,m} \\ &= Q_m y_m \lambda + q_{m+1} y_m h_{m+1,m} \end{aligned}$$

We can use q_{m+1} to continue generating vectors from the Krylov space.

Lanczos

The Lanzcos algorithm is Arnoldi applied to a symmetric system using short recurrence (two previous vectors).

$AQ_m = Q_{m+1}\underline{T}_m$, where $\underline{T}_m = [\, t_{j,k} \,]_{j=1:m+1,k=j-1:j+1}$ is tridiagonal and symmetric (even cheaper)

Use $T_m = Q_m^* A Q_m$ to compute approximate eigenpairs.

$T_m y = \lambda y$ which gives approximate pairs: $(\lambda,\, Q_m y)$ of A.

If we only need eigenvalues (and do not re-orthogonalize) we can discard the vectors and restarting is not necessary.

Special care required to avoid spurious copies of eigenvalues.

Otherwise, techniques for Arnoldi can be used.

IRA

We need to restart from time to time to save memory and CPU time.

This also holds for Lanczos if we want to compute eigenvectors or reorthogonalize (part.) for accuracy.

Efficient and accurate implementation is so-called Implicitly Restarted Arnoldi (Lanczos).

This method discards unwanted Ritz pairs using polynomial filter applied to Hessenberg matrix (tridiagonal).

IRA

Discard unwanted eigenpairs with polynomial filter. (Sorensen'92)

The remaining vectors span new Krylov subspace with different starting vector 'closer' to wanted invariant subspace.

Method builds search space of dimension m, discards $m-k$ vectors, and extends again to m vectors using the residual of the Ritz pairs (same vector; w_{m+1} from $AW_m = W_{m+1}H_m$)

ARPACK from www.netlib.org, user guide from SIAM (Lehoucq, Sorensen, and Yang '98)

Inverse Iteration

What if we want the smallest (magnitude) eigenvalue or any other one that is not the largest in magnitude? Inverse iteration:

$$x_k = A^{-1}x_{k-1} \quad \Leftrightarrow \quad \text{solve } Ax_k = x_{k-1} \qquad \text{or}$$
$$x_k = (A - sI)^{-1}x_{k-1} \quad \Leftrightarrow \quad \text{solve } (A - sI)x_k = x_{k-1}$$

Inverse iteration with A converges to smallest (in magnitude) eigenvalue if unique, and inverse iteration with $(A - sI)$ converges to eigenvalue/vector closest to s.
By choosing s more accurate we can improve convergence speed.

$(A - sI)^{-1}$ has eigenvalues $(\lambda_j - s)^{-1}$ and so, assuming we want to approximate the eigenpair with (simple) eigenvalue λ_k the rate of convergence is determined by

$$\max_{j \neq k} \frac{|\lambda_k - s|}{|\lambda_j - s|}$$

This can be made arbitrarily small if we know λ_k sufficiently accurate.

Rayleigh Quotient Iteration

Let x be an approximate eigenvector for a real matrix A. Then approximate eigenvalue is given by $n \times 1$ linear least squares problem:

$$x\hat{\lambda} \approx Ax \Rightarrow x^T x\hat{\lambda} = x^T Ax \quad \Rightarrow \quad \hat{\lambda} = \frac{x^T Ax}{x^T x} \qquad \text{(Rayleigh quotient)}$$

Let $Av = \lambda v$ and A be symmetric then $\quad \|x - v\| = O(\varepsilon) \Rightarrow \left|\hat{\lambda} - \lambda\right| = O(\varepsilon^2)$

Let $x = v + \varepsilon p$, where $v \perp p$, $\|v\|_2 = \|p\|_2 = 1$

$$\frac{x^T Ax}{x^T x} = \frac{(v+\varepsilon p)^T A(v+\varepsilon p)}{(v+\varepsilon p)^T (v+\varepsilon p)} = \frac{v^T Av+2\varepsilon p^T Av+\varepsilon^2 p^T Ap}{v^T v+2\varepsilon p^T v+\varepsilon^2 p^T p} = \frac{\lambda+\varepsilon^2 p^T Ap}{1+\varepsilon^2} \approx$$

$$(\lambda + \varepsilon^2 p^T Ap)(1 - \varepsilon^2) = \lambda + O(\varepsilon^2)$$

Rayleigh Quotient Iteration

So given $O(\varepsilon)$ approximation to v we get $O(\varepsilon^2)$ approximation to λ.
Cunning plan: Improve eigenvalue estimate s every step using Rayleigh quotient.

$$x_k = (A - r_{k-1}I)^{-1} x_{k-1} \quad \Leftrightarrow \quad \text{solve } (A - r_{k-1}I) x_k = x_{k-1} \text{ with } r_{k-1} = \frac{x_{k-1}^T A x_{k-1}}{x_{k-1}^T x_{k-1}}$$

This is called Rayleigh Quotient Iteration
For symmetrix matrices, close to solution, we have cubic convergence.

Let $x = v_k + \varepsilon p = v_k + \Sigma_{j \neq k} v_j \alpha_j$, where $v \perp p$, $\| v \|_2 = \| p \|_2 = 1$

$x = (A - \hat{\lambda} I)^{-1} x = (\lambda_k - \hat{\lambda})^{-1} v_k + \Sigma_{j \neq 1} v_j \alpha_j (\lambda_j - \hat{\lambda})^{-1}$ normalization gives

$x(\lambda_k - \hat{\lambda}) = v_k + (\lambda_k - \hat{\lambda}) \Sigma_{j \neq 1} v_j \alpha_j (\lambda_j - \hat{\lambda})^{-1}$ and the norm of the error now gives

$$\| x(\lambda_k - \hat{\lambda}) - v_k \|_2 = \| (\lambda_k - \hat{\lambda}) \Sigma_{j \neq 1} v_j \alpha_j (\lambda_j - \hat{\lambda})^{-1} \|_2 \leq \| (\lambda_k - \hat{\lambda}) \rho \, \Sigma_{j \neq 1} v_j \alpha_j \| = O(\varepsilon^3)$$

Rayleigh Quotient Iteration

The Rayleigh Quotient iteration

$$r_{k-1} = \frac{x_{k-1}^T A x_{k-1}}{x_{k-1}^T x_{k-1}} \qquad \text{solve } (A - sI)x_k = x_{k-1}$$

requires repeatedly solving a (new) linear system.
We can make this cheaper by first reducing A to tridiagonal or Hessenberg form using unitary similarity transformations: $A' = Q^T A Q$

This gives $T = Q^T A Q$ is tridiagonal if A is symmetric, and $H = Q^T A Q$ is (upper) Hessenberg if A is nonsymmetric.

Solving a linear tridiagonal system (order n) takes only $O(n)$ operations.
Solving an upper Hessenberg system (order n) takes $O(n^2)$ operations, for n Givens rotations and backsubstitution $O(n^2)$.

(Approximate) Shift Invert Methods

- Inverse iteration with shift and Rayleigh quotient iteration are examples of shift-invert methods
- Very fast convergence, but require inverse of matrix
- Expensive for large matrix, especially if shift changes every iteration
- Hence, approximate shift invert methods (really, shift, approximate invert …)
- Approximate solution by one step approximation or few steps of iterative method
- Davidson’s method, Jacobi-Davidson method

Davidson

Davidson's method also uses a search space and Ritz values/vectors from that space, but extension of search space is different.

Let $Q_m \in \mathbb{R}^{n\times m}$, such that $Q_m^* Q_m = I$, $Q_m^* A Q_m = H_m$ and $H_m y = \lambda y$. We have residual $r = A Q_m y - \lambda Q_m y$.

Then extend search space as follows:
Solve $(D_A - \lambda I) t = r$ where D_A is diagonal of A.

$t = \left(I - Q_m Q_m^*\right) t$ (orthog) and set $Q_{m+1} = \left[Q_m \;\; t\right]$.
Compute H_{m+1} and its eigenpairs, continue.

Davidson

Very effective for outer most (esp.) dominant eigenvalues.

Various explanations for this have been given. In many cases the matrix $(D_A - \lambda I)^{-1}(A - \lambda I)$ has only few outlying eigenvalues (1). Another interprets Davidson as an improvement to an old method by Jacobi (2).

This method discards unwanted Ritz pairs using polynomial filter applied to Hessenberg matrix (tridiagonal).

Jacobi-Davidson

Linear eigenvalue problem: $Ax = \lambda x$

Orthon. basis for search space: $Q_k,\ H_k = Q_k^* A Q_k \to (\lambda, x)$,

Ritz pair: (λ, u) where $u = Q_k y$ and residual $r_k = Au - \lambda u$

Correction equation:

Solve: $\left(I - uu^*\right)(A - \lambda I)\left(I - uu^*\right)t = -r, \qquad t \perp u$

$$t = \left(I - Q_k Q_k^*\right)t \,/\, \left\|\left(I - Q_k Q_k^*\right)t\right\|$$

$$Q_{k+1} = \left[Q_k \ \ t\right],\ H_{k+1} = Q_{k+1}^* A Q_{k+1}$$

Continue

Discard unwanted eigenpairs with polynomial filter or using Schur decomposition of H_{k+1}.

(Sleijpen and van der Vorst '96)

Some literature

1. R.B. Morgan, Generalizations of Davidson's method for computing eigenvalues of large nonsymmetric matrices, J. Comput. Physics 101, pp. 287-291, 1992
2. Sleijpen and van der Vorst, A Jacobi-Davidson iteration method for linear eigenvalue problems, SIAM Review 42(2), pp. 267-293, 2000 (extension of SIMAX 1996).

(2) Provides many good references

Quick Comparison

- Implicitly restarted Arnoldi (IRA) and Davidson work well for eigenvalues on the boundary of the spectrum.
- Jacobi-Davidson also works for these, but may be less efficient.

- Jacobi-Davidson (JD) is superior for interior eigenvalues. Probably, because it is an approximate shift and invert technique and more effectively so than Davidson. To improve the 'invert' part preconditioning can be used in the linear solver. This may improve the effectiveness dramatically. (One can also use various linear solvers, e.g. multigrid to improve convergence)
- IRA has some advantage in looking for several eigenvalues at once.