Eigensystems and SVD

Eigenvalue problem:

$$A\mathbf{x} = \lambda\mathbf{x},$$

λ: eigenvalue
$\mathbf{x}$: right eigenvector. $\mathbf{y}^{\mathrm{H}}A = \lambda\mathbf{y}^{\mathrm{H}}$, $\mathbf{y}$ left eigenvector.

1 Nonsymmetric Eigenproblem

We first consider the case when A is a general nonsymmetric matrix.

1.1 Two Most Common Canonical Forms

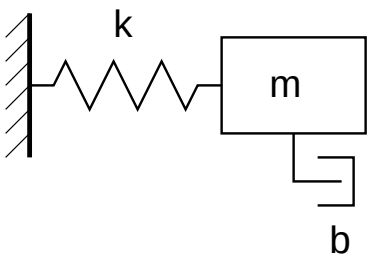
Jordan form

$$A = SJS^{-1}, \quad J = \mathrm{diag}(J_{n_1}(\lambda_1), ..., J_{n_k}(\lambda_k))$$

$$J_{n_i}(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix}.$$

- The algebraic multiplicity of λ_i is n_i.
- A Jordan block has one right eigenvector $[1, 0, ..., 0]^{\mathrm{T}}$ and one left eigenvector $[0, ..., 0, 1]^{\mathrm{T}}$.
- If all $n_i = 1$, then J is diagonal, A is called diagonalizable; otherwise, A is called defective.
- An n-by-n defective matrix has fewer than n eigenvectors.

Example. Spring-mass mechanical vibration



Newton's law $F = ma$ implies

$$m\ddot{x}(t) = -kx(t) - b\dot{x}(t).$$

Let

$$\mathbf{y}(t) = \begin{bmatrix} \dot{x}(t) \\ x(t) \end{bmatrix},$$

we transform the second order ODE into a system of the first order ODEs

$$\dot{\mathbf{y}}(t) = \begin{bmatrix} -\frac{b}{m} & -\frac{k}{m} \\ 1 & 0 \end{bmatrix} \mathbf{y}(t) =: A\mathbf{y}(t).$$

If A is diagonalizable and $A = S\Lambda S^{-1}$, let $\mathbf{z}(t) = S^{-1}\mathbf{y}(t)$, then $\dot{\mathbf{z}}(t) = \Lambda\mathbf{z}(t)$, whose solution is

$$z_i(t) = e^{\lambda_i t} z_i(0) \quad \text{or} \quad \mathbf{z}(t) = e^{\Lambda t}\mathbf{z}(0).$$

Thus

$$\mathbf{y}(t) = Se^{\Lambda t}S^{-1}\mathbf{y}(0).$$

The characteristic polynomial of A is

$$\lambda^2 + \frac{b}{m}\lambda + \frac{k}{m}$$

and the eigenvalues are

$$\lambda_{\pm} = \frac{-\frac{b}{m} \pm \sqrt{\left(\frac{b}{m}\right)^2 - 4\frac{k}{m}}}{2} = \frac{b}{2m}\left(-1 \pm \sqrt{1 - \frac{4km}{b^2}}\right).$$

- When $4km/b^2 < 1$, overdamped, two negative eigenvalues, the solution eventually decays monotonically to zero.
- When $4km/b^2 > 1$, underdamped, two complex conjugate eigenvalues with real part $-b/(2m)$, the solution oscillates while decaying to zero.
- When $4km/b^2 = 1$, critically damped, two equal eigenvalues, A is not diagonalizable.

In practice, confronting defective matrices is a fundamental fact.

It is undesirable to compute Jordan form, because

- Jordan block is discontinuous

$$J(0) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad J(\epsilon) = \begin{bmatrix} \epsilon & 1 \\ 0 & 2\epsilon \end{bmatrix},$$

  while $J(0)$ has an eigenvalue of multiplicity two, $J(\epsilon)$ has two simple eigenvalues.

- In general, computing Jordan form is unstable, that is there is no guarantee that $\widehat{S}\widehat{J}\widehat{S}^{-1} = A + \delta A$ for a small δA.

Schur form

$$A = QTQ^{\mathrm{H}}$$

Q: unitary
T: upper triangular
The eigenvalues of A are the diagonal elements of T.

Real case

$$A = QTQ^{\mathrm{T}}$$

Q: orthogonal
T: quasi-upper triangular, 1-by-1 or 2-by-2 blocks on the diagonal.

1.2 Perturbation Theory

- The condition number for eigenvalues can be infinite.
  Example

$$A = \begin{bmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ \epsilon & & & 0 \end{bmatrix}$$

  has eigenvalues $\lambda = \sqrt[n]{\epsilon}$. The condition number is infinite, since $d\lambda/d\epsilon = \epsilon^{1/n-1}/n = \infty$, when $n \geq 2$ and $\epsilon \to 0$. This example shows that a small change in A can change multiple eigenvalues into simple eigenvalues. A close to multiple eigenvalue can have a large condition number.

- Let λ be a simple eigenvalue of A with unit right eigenvector $\mathbf{x}$ and left eigenvector $\mathbf{y}$ and $\lambda + \Delta\lambda$ be the corresponding eigenvalue of $A + \Delta A$, then
$$\Delta\lambda = \frac{\mathbf{y}^{\mathrm{H}} \Delta A \mathbf{x}}{\mathbf{y}^{\mathrm{H}} \mathbf{x}} + O(\|\Delta A\|^2)$$
or
$$|\Delta\lambda| \leq \frac{1}{|\mathbf{y}^{\mathrm{H}} \mathbf{x}|} \|\Delta A\| + O(\|\Delta A\|^2).$$
Thus, $1/|\mathbf{y}^{\mathrm{H}}\mathbf{x}|$ is the condition number for finding the eigenvalue λ. For a Jordan block, $\mathbf{x} = [1, 0, ..., 0]^{\mathrm{T}}$ and $\mathbf{y} = [0, ..., 0, 1]^{\mathrm{T}}$, therefore, as we have seen, the condition number $1/|\mathbf{y}^{\mathrm{H}}\mathbf{x}| = \infty$. For symmetric A, $\mathbf{x} = \mathbf{y}$ and $1/|\mathbf{y}^{\mathrm{H}}\mathbf{x}| = 1$.

1.3 Algorithms

Computing eigenvalues cannot be finished in finite steps. An algorithm for computing eigenvalues must be iterative. Still, we categorize the eigenvalue algorithms into two groups: direct and iterative. Like solving linear systems, computing eigenvalues usually involves two stages. In the first stage, we reduce A into a simpler matrix with the same eigenvalues. In the second stage, we compute the eigenvalues of the simpler matrix. The algorithms which reduce A into a simpler matrix in finite steps are called direct methods. The algorithms which iteratively reduce A into a simpler matrix are called iterative methods.

We first study direct algorithms for nonsymmetric eigenproblem. For simplicity, we consider the case where A is a real matrix. Two orthogonal transformations: Givens rotation and Householder reflection, are commonly used to reduce A into upper Hessenberg form

Givens rotation
$$G = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Introducing a zero into a 2-vector:
$$G \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$$

i.e., rotate $\mathbf{x}$ onto x_1-axis.

Computing Givens rotation.

$$\cos\theta = \frac{x_1}{\sqrt{x_1^2 + x_2^2}} \quad \sin\theta = \frac{x_2}{\sqrt{x_1^2 + x_2^2}}$$

Algorithm. (Givens rotation)

```
if x(2) = 0
   c =1.0; s = 0.0;
elseif abs(x(2)) >= abs(x(1))
   ct = x(1)/x(2);
   s = 1/sqrt(1 + ct*ct); c = s*ct;
else
   t = x(2)/x(1);
   c = 1/sqrt(1 + t*t); s = c*t;
end
```

In general, select a pair (x_i, x_j), find a rotation G_{ij} to eliminate x_j. For example,

$$G_{13} = \begin{bmatrix} c & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G_{13} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \times \\ x_2 \\ 0 \\ x_4 \end{bmatrix}.$$

Reduction to upper Hessenberg form

$$\begin{array}{ccccccccccccccccc}
 & & \downarrow & \downarrow & & & & & \downarrow & & \downarrow & & & & & \downarrow & \downarrow \\
 & \times & \times & \times & \times & & & \times & \times & \times & \times & & & \times & \times & \times & \times \\
\rightarrow & \times & \times & \times & \times & \longrightarrow & \rightarrow & \times & \times & \times & \times & \longrightarrow & & \times & \times & \times & \times \\
\rightarrow & \otimes & \times & \times & \times & & & 0 & \times & \times & \times & & \rightarrow & 0 & \times & \times & \times \\
 & \times & \times & \times & \times & & \rightarrow & \otimes & \times & \times & \times & & \rightarrow & 0 & \otimes & \times & \times
\end{array}$$

$$G_{34}G_{24}G_{23}AG_{23}^{\mathrm{T}}G_{24}^{\mathrm{T}}G_{34}^{\mathrm{T}} = H,$$

$$A = QHQ^{\mathrm{T}}, \quad Q = G_{23}^{\mathrm{T}} G_{24}^{\mathrm{T}} G_{34}^{\mathrm{T}}.$$

Applying rotations on side, we can reduce a matrix to upper triangular. The QR factorization

$$\begin{array}{l} \rightarrow \\ \rightarrow \\ \\ \end{array}\!\!\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \longrightarrow \begin{array}{l} \rightarrow \\ \\ \rightarrow \\ \\ \end{array}\!\!\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \longrightarrow \begin{array}{l} \rightarrow \\ \\ \\ \rightarrow \end{array}\!\!\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ \times & \times & \times \end{bmatrix}$$

$$\longrightarrow \begin{array}{l} \rightarrow \\ \rightarrow \\ \\ \end{array}\!\!\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} \longrightarrow \begin{array}{l} \rightarrow \\ \\ \\ \rightarrow \end{array}\!\!\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & \times & \times \end{bmatrix} \longrightarrow \begin{array}{l} \\ \rightarrow \\ \rightarrow \\ \end{array}\!\!\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix}$$

$$G_{34} G_{24} G_{23} G_{14} G_{13} G_{12} A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$$Q = G_{12}^{\mathrm{T}} G_{13}^{\mathrm{T}} G_{14}^{\mathrm{T}} G_{23}^{\mathrm{T}} G_{24}^{\mathrm{T}} G_{34}^{\mathrm{T}}$$

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

Basically, in the QR decomposition, we introduce zeros below the main diagonal of A using orthogonal transformations. Here is another orthogonal transformation.

Householder transformation

$$H = I - 2\mathbf{u}\mathbf{u}^{\mathrm{T}} \quad \text{with } \mathbf{u}^{\mathrm{T}}\mathbf{u} = 1$$

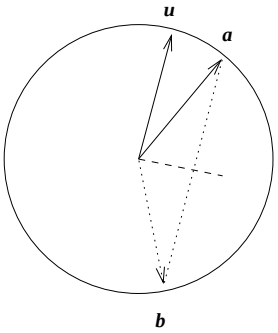
H is symmetric and orthogonal ($H^2 = I$).

Goal: $H\mathbf{a} = \alpha \mathbf{e}_1$.

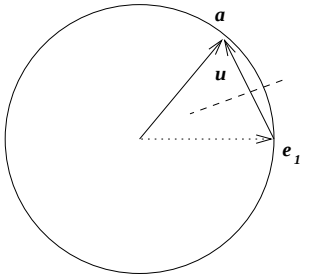
Choose

$$\mathbf{u} = \mathbf{a} \pm \|\mathbf{a}\|_2 \, \mathbf{e}_1$$

A geometric interpretation:



(a)



(b)

Figure (a) shows the image $\mathbf{b} = (I - 2\mathbf{u}\mathbf{u}^{\mathrm{T}})\mathbf{a}$ for an arbitrary $\mathbf{u}$, in figure (b), $\mathbf{u} = \mathbf{a} - \|\mathbf{a}\|_2\,\mathbf{e}_1$.

Normalize u using

$$\|u\|_2^2 = 2(\|a\|_2^2 \pm a_1\|a\|_2)$$

Algorithm. (Householder reflection) Given an n-vector x, this algorithm returns σ, α, and u such that $(I - \sigma^{-1}uu^{\mathrm{T}})x = -\alpha e_1$.

```
m = max(abs(x));
u = x/m;

alpha = sign(u(1))*norm(u);
u(1)= u(1) + alpha;
sigma = alpha*u(1);

alpha = m*alpha;
```

After A is reduced to the upper Hessenberg form, the next stage is the Schur decomposition of the upper Hessenberg matrix. Our ultimate practical algorithm for the Schur decomposition of an upper Hessenberg matrix is the QR method. First, we introduce some techniques used in the QR method.

Power method

Suppose A has distinct eigenvalues and $\mathbf{x}_i$ $(i = 1, ..., n)$ are the eigenvectors. An arbitrary vector $\mathbf{u}$ can be expressed as

$$\mathbf{u} = \mu_1\mathbf{x}_1 + \mu_2\mathbf{x}_2 + \cdots + \mu_n\mathbf{x}_n$$

If $\mu_1 \neq 0$, $A^k\mathbf{u}$ has almost the same direction as $\mathbf{x}_1$ when $(\lambda_2/\lambda_1)^k$ is small. Thus, the Rayleigh quotient

$$\frac{(A^k\mathbf{u})^{\mathrm{T}}A(A^k\mathbf{u})}{(A^k\mathbf{u})^{\mathrm{T}}(A^k\mathbf{u})} \approx \lambda_1.$$

Algorithm (Power method)

Initial $\mathbf{u}_0$;
$i = 0$;
repeat
$\mathbf{v}_{i+1} = A\mathbf{u}_i$;
$\mathbf{u}_{i+1} = \mathbf{v}_{i+1}/\|\mathbf{v}_{i+1}\|_2$;
$\tilde{\lambda}_{i+1} = \mathbf{u}_{i+1}^{\mathrm{T}}A\mathbf{u}_{i+1}$;
$i = i + 1$
until convergence

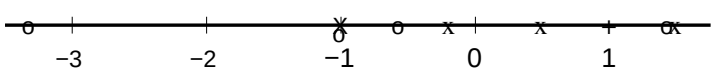
Problems

- Computes only (λ_1, x_1)
- Converges slowly when $|\lambda_1| \approx |\lambda_2|$
- Does not work when $|\lambda_1| = |\lambda_2|$

Inverse power method

Suppose that μ is an estimate of λ_k, then $(\lambda_k - \mu)^{-1}$ is the dominant eigenvalue of $(A - \mu I)^{-1}$. Applying the power method to $(A - \mu I)^{-1}$, we can compute $\mathbf{x}_k$ and λ_k.

Example.
Eigenvalues of A: -1, -0.2, 0.5, 1.5
Shift μ: 0.8
Eigenvalues of $(A - \mu I)^{-1}$: -3.3, -1, -0.6, 1.4



- Very effective when we have a good estimate for an eigenvalue.

QR method

Goal: Generate a sequence

$$A_0 = A, A_1, ..., A_{k+1}$$

$$A_{i+1} = Q_i^{\mathrm{T}} A Q_i = \begin{bmatrix} B & \mathbf{u} \\ \mathbf{s}^{\mathrm{T}} & \mu \end{bmatrix}$$

where $\mathbf{s}$ is small and Q_i is orthogonal, i.e., $Q_i^{\mathrm{T}} = Q_i^{-1}$ (so A_{k+1} and A have same eigenvalues).

- Since $\mathbf{s}$ is small, μ is an approximation of an eigenvalue of A_{k+1};
- Since A_{k+1} is similar to A, μ is an approximation of an eigenvalue of A;
- Deflate A_{k+1} and repeat the procedure on B. Size is reduced by one.

What does Q_k look like?
If the last column of Q_k is a left eigenvector $\mathbf{y}$ of A, then

$$\begin{aligned} Q_k^{\mathrm{T}} A Q_k &= \begin{bmatrix} P_k^{\mathrm{T}} \\ \mathbf{y}^{\mathrm{T}} \end{bmatrix} A \, [P_k \ \mathbf{y}] \\ &= \begin{bmatrix} P_k^{\mathrm{T}} \\ \mathbf{y}^{\mathrm{T}} \end{bmatrix} [AP_k \ A\mathbf{y}] \\ &= \begin{bmatrix} B & \mathbf{u} \\ 0^{\mathrm{T}} & \lambda \end{bmatrix} \end{aligned}$$

How do we get an approximation of a left eigenvector $\mathbf{y}$ of A ($\mathbf{y}^{\mathrm{T}} A = \lambda \mathbf{y}^{\mathrm{T}}$)?
One step of the inverse power method: Solve for $\mathbf{q}$ in $(A - \mu I)^{\mathrm{T}} \mathbf{q} = \mathbf{e}_n$, where μ is an estimate for an eigenvalue of A. (How? Later.)

How do we construct an orthogonal Q whose last column is $\mathbf{q}$?
If $(A - \mu I) = QR$ is the QR decomposition and $\mathbf{q}$ is the last column of Q, then

$$\mathbf{q}^{\mathrm{T}}(A - \mu I) = \mathbf{q}^{\mathrm{T}} QR = r_{n,n} \mathbf{e}_n^{\mathrm{T}}.$$

Thus, after normalizing,

$$(A - \mu I)^{\mathrm{T}} \mathbf{q} = \mathbf{e}_n.$$

This is one iteration of the inverse power method with initial vector $\mathbf{e}_n$.

But, we want

- similarity transformations of A, not $A - \mu I$;
- to carry on and improve accuracy.

$A - \mu I = QR$.
$RQ = Q^{\mathrm{T}}(A - \mu I)Q = Q^{\mathrm{T}}AQ - \mu I$.
$RQ + \mu I = Q^{\mathrm{T}}AQ$ is similar to A.

Algorithm. (One step of QR method)

```
repeat
   choose a shift mu;
   QR decomposition A - mu*I = QR;
   A = RQ + mu*I;
until convergence (A(n,1:n-1) small)
```

Problem:
If A is full, it is very expensive (one QR decomposition each iteration).

Solution:
Preprocess A to $Q^{\mathrm{T}}AQ = H$ upper Hessenberg using Householder transformations or Givens rotations.

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \rightarrow \cdots$$

Can we maintain the upper Hessenberg structure during the QR iterations?
H_0 is upper Hessenberg;
$H_0 - \mu I$ is upper Hessenberg;
$H_0 - \mu I = QR$, R is upper triangular and Q is upper Hessenberg;
$H_1 = RQ + \mu I$ is upper Hessenberg;

Implicit method

Compute H_{k+1} from H_k without explicitly computing the QR decomposition and the product RQ.

Find a rotation G such that

$$G \begin{bmatrix} h_{1,1} - \mu \\ h_{2,1} \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$$

Note that this is the first rotation in the QR decomposition of $H - \mu I$. We know that the new H is $Q^{\mathrm{T}}HQ$. Apply G to both sides of H:

$$\begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} \\ 0 & 0 & h_{4,3} & h_{4,4} \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & h_{3,3} & h_{3,4} \\ 0 & 0 & h_{4,3} & h_{4,4} \end{bmatrix}$$

We also know that $Q^{\mathrm{T}}HQ$ is upper Hessenberg. Restore upper Hessenberg structure:

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \otimes & \times & h_{3,3} & h_{3,4} \\ 0 & 0 & h_{4,3} & h_{4,4} \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \otimes & \times & \times \end{bmatrix} \rightarrow \cdots$$

Q is determined by its first column.

Choosing the shift

Since the last element converges to an eigenvalue, it is reasonable to choose $h_{n,n}$ as the shift. But, it doesn't always work. A more general method is to choose the eigenvalue of the trailing 2-by-2 submatrix

$$\begin{bmatrix} h_{n-1,n-1} & h_{n-1,n} \\ h_{n,n-1} & h_{n,n} \end{bmatrix}$$

that is close to $h_{n,n}$. Heuristically, it is more effective than choosing $h_{n,n}$ especially in the beginning.

What if the trailing 2-by-2 submatrix has a complex conjugate pair of eigenvalues? The double shift strategy can be used to overcome the difficulty. In general, the Francis QR method using double implicit shift strategy can reduce an real Hessenberg matrix into the real Schur form.

2 Symmetric Eigenproblem

A symmetric matrix is diagonalizable: $A = Q\Lambda Q^{\mathrm{T}}$, $\Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_n)$.

Example. In the mass-spring example, if there are n masses and n springs but no dampers, the differential equation becomes

$$M\ddot{\mathbf{x}}(t) = -K\mathbf{x}(t),$$

where $M = \text{diag}(m_1, ..., m_n)$ is a diagonal matrix of masses and

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & & & & \\ -k_2 & k_2 + k_3 & -k_3 & & & \\ & \ddots & \ddots & & & \\ & & -k_n & k_{n-1} + k_n & -k_n & \\ & & & -k_n & k_n & \end{bmatrix}.$$

Although K is symmetric, $M^{-1}K$ is not. Fortunately, in this case, we can transform it into a symmetric system. Define $M^{1/2} = \text{diag}(m_1^{1/2}, ..., m_n^{1/2})$ and $\mathbf{y}(t) = M^{1/2}\mathbf{x}(t)$, we have

$$\ddot{\mathbf{y}}(t) = -\widehat{K}\mathbf{y}(t),$$

where $\widehat{K} = M^{-1/2}KM^{-1/2}$ is symmetric, which is diagonalizable. Thus we have the solution of the form

$$y_i(t) = e^{\lambda_i t} y_i(0),$$

which means $-\widehat{K}e^{\lambda_i t}y_i(0) = \lambda_i^2 e^{\lambda_i t}y_i(0)$ or $-\lambda_i^2$ and $y_i(0)$ are the eigenvalues and eigenvectors of $\widehat{K}$.

Perturbation theory

- Eigenvalue: Let A and E be symmetric and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ the eigenvalues of A and $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_n$ the eigenvalues of $\hat{A} := A + E$, then $|\lambda_i - \hat{\lambda}_i| \leq \|E\|_2$. Suppose we have a stable algorithm, the computed eigenvalues are the exact eigenvalues of $A + E$ for a small E, then the computed eigenvalues are accurate in absolute sense. Relatively, the eigenvalues of the same size as the entries of A are accurate, but very small eigenvalues, if they exist, may have large relative errors.

- Eigenvector: Define $\text{gap}(i, A) = \min_{j \neq i} |\lambda_j - \lambda_i|$. Let $A = Q\Lambda Q^{\mathrm{T}}$ and $\hat{A} = A + E = \hat{Q}\hat{\Lambda}\hat{Q}^{\mathrm{T}}$. Let θ denote the acute angle between $\mathbf{q}_i$ and $\hat{\mathbf{q}}_i$, then

  $$\frac{1}{2}\sin(2\theta) \leq \frac{\|E\|_2}{\text{gap}(i, A)},$$

  provided that $\text{gap}(i, A) > 0$. Similarly,

  $$\frac{1}{2}\sin(2\theta) \leq \frac{\|E\|_2}{\text{gap}(i, A + E)},$$

provided that $\text{gap}(i, A + E) > 0$. When the eigenvalues are well separated, the computed eigenvectors are accurate. If we have a cluster of eigenvalues, then we cannot expect to get the individual eigenvectors accurately. However, we can expect accurate invariant subspace spanned by these eigenvectors.

Algorithms

- QR method. This method is very efficient if only all eigenvalues are desired or all eigenvalues and eigenvectors are desired and the matrix is small ($n \leq 25$). Analogous to the nonsymmetric case, it has two stages:

  1. Reduce A to symmetric tridiagonal. This costs $\frac{4}{3}n^3$ or $\frac{8}{3}n^3$ if eigenvectors are also desired.
  2. Apply the QR iteration to the tridiagonal. On average, it takes two QR steps per eigenvalue. Finding all eigenvalues takes $6n^2$. Finding all eigenvalues and eigenvectors requires $6n^3$.

  Example

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

  After tridiagonalization

$$\begin{bmatrix} 1.0000 & -5.3852 & 0 & 0 \\ -5.3852 & 5.1379 & -1.9952 & 0 \\ 0 & -1.9952 & -1.3745 & 0.2895 \\ 0 & 0 & 0.2895 & -0.7634 \end{bmatrix}$$

	μ	β_1	β_2	β_3
1	-0.6480	3.8161	0.2222	-0.0494
2	-0.5859	1.2271	0.0385	10^{-5}
3	-0.5858	0.3615	0.0070	converge
4	-1.0990	0.0821	10^{-10}	
5	-1.0990	0.0186	converge	

- Divide-and-conquer. This method is fast for large ($n > 25$) matrices if all eigenvalues and eigenvectors are desired.

  Given a symmetric tridiagonal

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & 0 \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_1 \\ 0 & & \beta_{n-1} & \alpha_n \end{bmatrix},$$

  we divide it into

$$T = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} + \beta_m \mathbf{v}\mathbf{v}^{\mathrm{T}}, \quad \text{where} \quad \mathbf{v} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

  If the eigendecompositions $T_1 = Q_1 \Lambda_1 Q_1^{\mathrm{T}}$ and $T_2 = Q_2 \Lambda_2 Q_2^{\mathrm{T}}$ are available, let

$$\mathbf{u} = \begin{bmatrix} Q_1^{\mathrm{T}} & 0 \\ 0 & Q_2^{\mathrm{T}} \end{bmatrix} \mathbf{v} = \begin{bmatrix} \text{last column of } Q_1^{\mathrm{T}} \\ \text{first column of } Q_2^{\mathrm{T}} \end{bmatrix},$$

  then the eigenvalues of T are the same as those of $D + \beta_m \mathbf{u}\mathbf{u}^{\mathrm{T}}$, where $D = \mathrm{diag}(\Lambda_1, \Lambda_2)$, a diagonal plus a rank-1 matrix. Those eigenvalues can be efficiently computed using Newton's method. If λ is an eigenvalue of $D + \beta_m \mathbf{u}\mathbf{u}^{\mathrm{T}}$, then its eigenvector is $(D - \lambda I)^{-1}\mathbf{u}$, which can be obtained in $O(n)$ since $D - \lambda I$ is diagonal.

3 Singular Value Decomposition (SVD)

3.1 Introduction

$$A = U \Sigma V^{\mathrm{T}}$$

A: m-by-n real matrix ($m \geq n$)
U: m-by-m orthogonal
V: n-by-n orthogonal
Σ: diagonal,diag(σ_i), $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$
Singular values: σ_i
Left singular vectors: columns of U
Right singular vectors : columns of V

SVD reveals many important properties of a matrix A. For example,

- The number of nonzero singular values is the rank of A. Suppose $\sigma_k > 0$ and $\sigma_{k+1} = 0$, then rank(A) $= k$. If $k < n$, the columns of A are linearly dependent. (A is rank deficient.)
- If $\sigma_n > 0$ (A is of full rank),

$$\text{cond}(A) = \frac{\sigma_1}{\sigma_n}$$

Example

$$A = \begin{bmatrix} 1 & 6 & 11 \\ 2 & 7 & 12 \\ 3 & 8 & 13 \\ 4 & 9 & 14 \\ 5 & 10 & 15 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.355 & -0.689 & 0.541 & 0.193 & 0.265 \\ 0.399 & -0.376 & -0.802 & -0.113 & 0.210 \\ 0.443 & -0.062 & 0.160 & -0.587 & -0.656 \\ 0.487 & 0.251 & -0.079 & 0.742 & -0.378 \\ 0.531 & 0.564 & 0.180 & -0.235 & 0.559 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 35.127 & 0 & 0 \\ 0 & 2.465 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.202 & 0.890 & 0.408 \\ 0.517 & 0.257 & -0.816 \\ 0.832 & -0.376 & 0.408 \end{bmatrix}$$

A compact form

$$U = \begin{bmatrix} 0.355 & -0.689 \\ 0.399 & -0.376 \\ 0.443 & -0.062 \\ 0.487 & 0.251 \\ 0.531 & 0.564 \end{bmatrix}$$

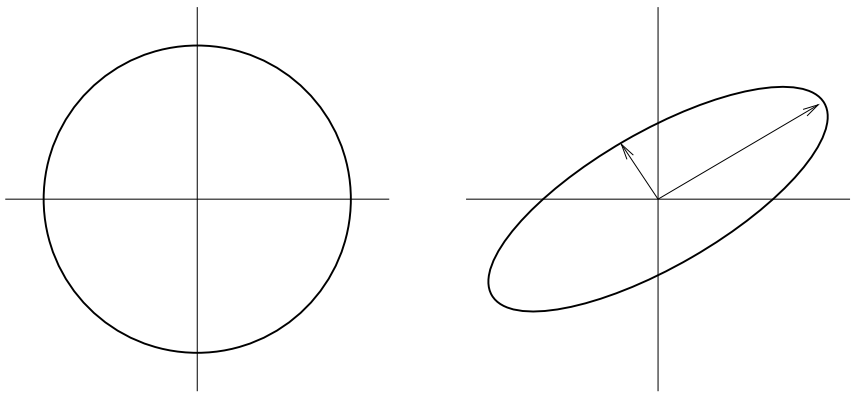
$$\Sigma = \begin{bmatrix} 35.127 & 0 \\ 0 & 2.465 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.202 & 0.890 \\ 0.517 & 0.257 \\ 0.832 & -0.376 \end{bmatrix}$$

Geometric interpretation

Transformation A: $\mathbf{x} \rightarrow A\mathbf{x}$

$$\sigma_1 \geq \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \geq \sigma_n$$



3.2 Applications

Mass-spring problem

It can be verified that the stiffness matrix

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & & & \\ -k_2 & k_2 + k_3 & -k_3 & & \\ & \ddots & \ddots & & \\ & & -k_n & k_{n-1} + k_n & -k_n \\ & & & -k_n & k_n \end{bmatrix}$$

in the mass-spring system can be factorized as

$$K = BK_D B^{\mathrm{T}},$$

where $K_D = \mathrm{diag}(k_1, ..., k_n)$ and

$$B = \begin{bmatrix} 1 & -1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & -1 \\ 0 & & & 1 \end{bmatrix}.$$

Then

$$\widehat{K} = M^{-1/2} K M^{-1/2} = (M^{-1/2} B K_D^{1/2})(M^{-1/2} B K_D^{1/2})^{\mathrm{T}} := GG^{\mathrm{T}}.$$

This shows that $\widehat{K}$ is symmetric and positive definite. Let σ_i be the singular values of G, then σ_i^2 are the eigenvalues of $\widehat{K}$. The left singular vectors of G are the eigenvectors of $\widehat{K}$. This example shows that we can use SVD to solve symmetric and positive definite eigenproblem.

Linear least-squares problem

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2^2$$

also called linear regression problem in statistics.

SVD: $A = U\Sigma V^{\mathrm{T}}$

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 = \|\Sigma\mathbf{z} - \mathbf{d}\|_2^2$$

where

$$\mathbf{d} = U^{\mathrm{T}}\mathbf{b} \quad \mathbf{z} = V^{\mathrm{T}}\mathbf{x}$$

Solution

$$\begin{aligned} z_j &= \frac{d_j}{\sigma_j} \qquad \text{if } \sigma_j \neq 0 \\ z_j &= \text{anything} \quad \text{if } \sigma_j = 0 \end{aligned}$$

Usually, we set

$$z_j = 0 \qquad \text{if } \sigma_j = 0$$

for minimum norm solution.

This allows us to solve the linear least-squares problems with singular A.

Example For the matrix A in the previous example.

$$\mathbf{b} = [4\ 5\ 5\ 5\ 5]^{\mathrm{T}}$$

$$\begin{aligned} 35.127z_1 &\approx 10.716 \\ 2.465z_2 &\approx -0.872 \\ 0z_3 &\approx -0.541 \\ 0 &\approx -0.193 \\ 0 &\approx -0.265 \end{aligned}$$

Setting $z_3 = 0$, we get

$$\mathbf{x} = \begin{bmatrix} -0.253 \\ 0.067 \\ 0.387 \end{bmatrix}$$

Check

$$A\mathbf{x} = \begin{bmatrix} 4.406 \\ 4.607 \\ 4.808 \\ 5.009 \\ 5.210 \end{bmatrix}$$

Principal component analysis (PCA)

Suppose that A is a data matrix. For example, each column contains samples of a variable. It is frequently standardized by subtracting the means of the columns and dividing by their standard deviations. If A is standardized, then $A^{\mathrm{T}}A$ is the correlation matrix. The most common description of PCA is in terms of eigenvalues and eigenvectors of the correlation matrix. In terms of SVD, let

$$A = U\Sigma V^{\mathrm{T}}$$

be the SVD of A. If the variables are strongly correlated, there are few components, fewer than the number of variables, can predict all the variables. In terms of SVD, if A is m-by-n ($m \geq n$), we partition $U = [\mathbf{u}_1, ..., \mathbf{u}_m]$ and $V = [\mathbf{v}_1, ..., \mathbf{v}_n]$. Then we can write

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^{\mathrm{T}} + \cdots + \sigma_n \mathbf{u}_n \mathbf{v}_n^{\mathrm{T}}.$$

If the variables are strongly correlated, there are few singular values that are significantly larger than the others. In other words, we can find an r such that $\sigma_r \gg \sigma_{r+1}$. We can use the rank r matrix

$$A_r = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^{\mathrm{T}} + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^{\mathrm{T}}$$

to approximate A. In fact, A_r is the closest (in Frobenius norm) rank r approximation to A. Usually, we can find $r \ll n$.

Principal component analysis is used in a wide range of fields. In image processing, for example, A is a 2D image.

3.3 Algorithm

Note that the columns of U are the eigenvectors of AA^{T} (symmetric and positive semi-definite); the columns of V are eigenvectors of $A^{\mathrm{T}}A$. The algorithm is parallel to the QR method for symmetric eigenvalue decompostion. We work on A instead of $A^{\mathrm{T}}A$.

1. Bidiagonalize A using Householder transformations ($A \to B$ is upper bidiagonal and $B^{\mathrm{T}}B$ tridiagonal);
2. QR iteration
   (a) Find a Givens rotation G_1 from the first column of $B^{\mathrm{T}}B - \mu I$;
   (b) Apply G_1 to B;
   (c) Apply a sequence of rotations, left and right, to restore the bidiagonal structure of B;

The shift μ is obtained by calculating the eigenvalues of the 2-by-2 trailing submatrix of $B^{\mathrm{T}}B$.

4 Lanczos Method

In this section, we study Lanczos algorithm for symmetric eigensystem. This method is suited for large and structured matrices.

- The matrix is too large to use direct methods and we only want to compute a few eigenvalues and eigenvectors.

- The matrix is structured so that the matrix can be stored in far less than $O(n^2)$ storage and matrix-vector multiplication can be computed in less than $O(n^2)$ flops. A "black-box" is available for matrix-vector multiplication.

Recall that Lanczos algorithm can reduce a symmetric matrix into symmetric tridiagonal form. Suppose we carry out k Lanczos iterations on an n-by-n symmetric matrix A, then

$$T_k = Q_k^{\mathrm{T}} A Q_k,$$

where T_k is symmetric tridiagonal and Q_k has orthonormal columns computed by Lanczos algorithm. Let $Q = [Q_k \; Q_c]$ be orthogonal, so Q_c is the complement to Q_k. Denote

$$T = Q^{\mathrm{T}} A Q = \begin{bmatrix} Q_k^{\mathrm{T}} A Q_k & Q_k^{\mathrm{T}} A Q_c \\ Q_c^{\mathrm{T}} A Q_k & Q_c^{\mathrm{T}} A Q_c \end{bmatrix} := \begin{bmatrix} T_k & T_{kc} \\ T_{ck} & T_c \end{bmatrix}.$$

- The eigenvalues of T_k are called Ritz values. They are the "best" approximations to the k (not necessary the largest k) eigenvalues of A. Specifically, for the k eigenvalues θ_i of T_k, there are k eigenvalues λ_i of A such that $|\theta_i - \lambda_i| \leq \|T_{kc}\|_2$. As k grows, the largest and smallest ones converge first and the convergence is monotonic.
- If $T_k = V \Lambda V^{\mathrm{T}}$ is the eigendecomposition of T_k, then the columns of $Q_k V$ are the "best" approximations to the corresponding k eigenvectors of A. They are called Ritz vectors.
- The computed Lanczos vectors $\mathbf{q}_i$ in floating-point arithmetic tend to loose orthogonality quickly. A practical Lanczos algorithm must incorporate reorthogonalization techniques.

Algorithm. Lanczos algorithm for symmetric eigensystem (no reorthogonalization).

```
b(0) = 0; Q(:,0) = 0;
for j = 1 to k
   r = A*Q(:,j);
   a(j) = Q(:,j)'*r;
```

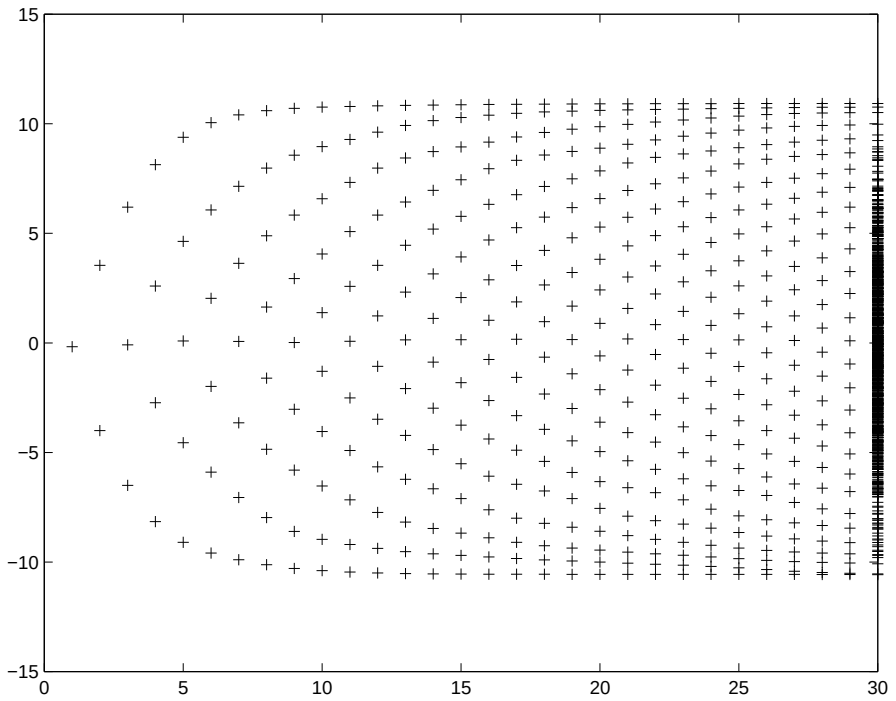


Figure 1: The horizontal axis shows the iteration number k and each column shows the eigenvalues θ_i of T_k. The last column plots the eigenvalues of A.

```
    r = r - a(j)*Q(:,j) - b(j-1)*Q(:,j-1);
    b(j) = norm2(r);
    if b(j) = 0, quit;
    Q(:,j+1) = r/b(j);
    Compute eigenvalues and eigenvectors of Tk
end
```

Summary

Nonsymmetric

- Canonical forms: Jordan form, Schur form, and real Schur form.
- Application to system of ODEs: $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + f(t)$.

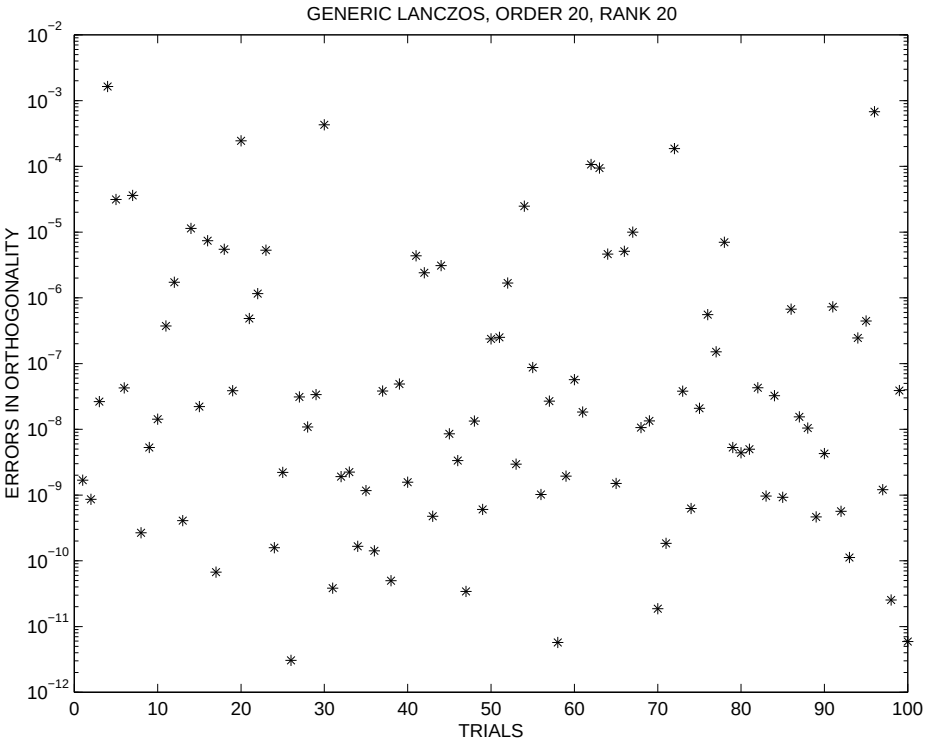


Figure 2: Each column plots $\|Q^{\mathrm{T}}Q - I\|_2$ at a trial for a 20-by-20 matrix.

- QR method
  1. Hessenberg reduction;
  2. QR iteration to get Schur form.
- Cost: $10n^3$ for eigenvalues only and $27n^3$ if eigenvectors are also desired.

Symmetric

- Tridiagonalization, direct methods.
- QR iteration.
- Perturbation theory.
- Lanczos tridiagonalization.

SVD and applications.

Software packages

LAPACK sgees (Schur form), sgeev (eigenvalues and eigenvectors), ssyev (symmetric and dense), sstev (symmetric and tridiagonal), sgesvd (SVD), sbdsqr (small and dense SVD)

IMSL evcrg, evcsf, lsvrr

MATLAB schur, eig, svd

NAG f02agf, f02abf, f02wef

Reference

Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, 2000.

James W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd Ed., The Johns Hopkins University Press, Baltimore, MD, 1996.