1. Introduction

Let A be a square n by n matrix
x a vector
$\lambda \in \mathbb{R}$ a scalar

$$Ax = \lambda x$$

eigensystem. λ is called an eigenvalue
x is the eigenvectors
usually (λ, x) is called an eigenpair

Here and later we assume that A is Hermitian, i.e. $A^* = A$
i.e. real, symmetric

2. ~~Determining~~ Eigenvalues and eigenvectors

$Ax = \lambda x \Rightarrow (A - I\lambda)x = 0$, where I is the identity matrix

This LSE has $x \neq 0$ solution $\Leftrightarrow \det(A - I\lambda) = 0$

$\Downarrow$

Since the degree of $p(\lambda)$ is n $\qquad p(\lambda) = \det(A - I\lambda)$: characteristic polynomial

$\Downarrow$

it has n roots

$\Downarrow$

A matrix has n eigenvalues and n corresponding x eigenvectors.

The n eigenvalues:

$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ $\qquad$ If all $\lambda_i > 0 \Rightarrow A$ is positive definite
$\lambda_i \geq 0 \Rightarrow A$ positive semidefinite

So each λ_i has an x_i eigenvector.

↳ $\|x_i\| = 1$ eigenvectors are normal

↳ $x_i \cdot x_j = \delta_{ij}$ the eigenvectors are orthogonal

3, Methods in general

Eigensolver
↙ ↘
direct solver $\qquad$ iterative solver

Direct solver : transformation method
↳ need to store the full matrix
↳ never fails to get the eigenvalues (99%)

Iterative solver : starting vector(s)
↳ don't need to store a full matrix
↳ can be very slow or fail to find eigenvalue

Important questions:

↳ How many eigenvalues do you need?

↳ all ⇒ direct method

↳ Size of the matrix

↳ if very large ⇒ iterative method

↳ small matrix, but ill-conditioned ⇒ direct better than iterative

4. Iterative methods

4.1. Power method

Algorithm:

y starting vector; $y \neq 0$

for $k = 1, 2, \dots$

$v = \frac{y}{\|y\|}$

$y = A \cdot v$

$\theta = v \cdot y$

if $\| y - \theta v \| < \varepsilon$, stop

end for

Results: $\theta = \lambda$ eigenvalues

$x = v$ eigenvector

But λ which eigenvalues?

$$y = c_1 \cdot x_1 + c_2 x_2 + \dots + c_n x_n \quad / A^k \cdot$$

$$A^k y = c_1 A^k x_1 + c_2 A^k x_2 + \dots + c_n A^k x_n = c_1 \lambda_1^k x_1 + c_2 \lambda_2^k x_2 + \dots =$$

$$= \lambda_1^k \left(c_1 x_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1} \right)^k c_i x_i \right)$$

If $\lambda_1 >> \lambda_i \Rightarrow \left(\frac{\lambda_i}{\lambda_1} \right)^k \to 0$, if $k \to \infty$

So power iteration converges the largest eigenvalue.

If A sparse $\Rightarrow$ $A \cdot v$ is very fast

If $\lambda_1 >> \lambda_1 \Rightarrow$ the convergence is very fast, otherwise slow.

4.2. Futher eigenvalues

If we know (λ_1, x_1) eigenpair: choose another a starting vector $a \perp x_1$:

$$b_1 = a - c_1 \cdot x_1 \quad , \text{where} \quad c_1 = a \cdot x_1$$

If we know the first m eigenpairs:

$$b = a - \sum^{m} c_i x_i \quad , \text{where} \quad c_i = a \cdot x_i$$

4.3 Interior eigenvalues · shift method.

$$(A - \mu I)\, v = (\lambda - \mu)\, v$$

↳ using power method to the shifted matrix, we get the closest λ eigenvalue to μ.

If μ is a good estimation ⇒ very fast.

4.4. Inverse (power) method.

y : starting vector

for $k = 1, 2, ..$

$v = \frac{y}{\|y\|}$

$y = A^{-1} \cdot v$

$\theta = v \cdot y$

if $\| y - \theta v \| < \varepsilon$, stop

end for

~~Result~~, $\lambda = 1/\theta$ and $x = \frac{y}{\theta}$

Computing A^{-1} is expensive ⇒ $A \cdot v = y$ LSO ⇒ fast (CGM)

λ_1 largest ⇒ $\frac{1}{\lambda_1}$ smallest, so the inverse method results in the smallest eigenvalue.

Usually very slow!

4.5. Rayleigh Quotient iteration

shift the matrix with $\rho(x) = \frac{x \cdot A \cdot x}{x \cdot x}$ Rayleigh quotient

y : starting vector; $\rho_1 = 0$

for $k = 1, 2, 3, ...$

$v = \frac{y}{\|y\|}$

$y = (A - \rho_k I)^{-1} v$

$\theta = \|y\|$

$\rho_{k+1} = \rho_k + \frac{y \cdot v}{\theta^2}$

$v = \frac{y}{\theta}$

if $\theta \geq \varepsilon$, stop.

end for

Results: $\lambda = \rho$ and $x = v$

4.6. Lanczos method

$$A \longrightarrow T \text{ tridiagonal matrix} \qquad T = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & \beta_2 & \alpha_3 & \beta_3 \\ & & \beta_3 & \ddots \\ & & & & \ddots \end{bmatrix}$$

eigenvalues of T are eigenvalues of A.

similarity transformation:

$$Q^T A Q = T \quad , \text{where } Q^T Q = I$$

$$\Downarrow$$

$$AQ = QT$$

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} \cdots & q_j & \cdots \end{bmatrix} = \begin{bmatrix} \cdot & q_{j-1} & q_j & q_{j+1} \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \beta_{j-1} & \\ & \alpha_j & \ddots \\ & \beta_j & \ddots \end{bmatrix}$$

$$Aq_j = \beta_{j-1} q_{j-1} + \alpha_j q_j + \beta_j q_{j+1}$$

a) determine α_j, q_j^T

$$q_j^T A q_j = \beta_{j-1} \underbrace{q_j^T q_{j-1}}_{0} + \alpha_j \underbrace{q_j^T q_j}_{1} + \beta_j \underbrace{q_j^T q_{j+1}}_{0} \Rightarrow \alpha_j = q_j^T A q_j$$

b) determine β_j

$$r_j = \beta_j q_{j+1} = Aq_j - \beta_{j-1} q_{j-1} - \alpha_j q_j$$

$$\| r_j \| = \beta_j \quad \Rightarrow q_{j+1} = \frac{r_j}{\beta_j}$$

Algorithm:

r_0 starting vector, $\beta_0 = \| r_0 \|$

$j = 1, 2, \dots$ ~~until~~

$q_j = \dfrac{r_{j-1}}{\beta_{j-1}}$

$a = A \cdot q_j$

$\alpha_j = q_j a$

$r_j = a - \beta_{j-1} q_{j-1} - \alpha_j q_j$

$\beta_j = \| r_j \|$

if $\beta_j < \varepsilon$ stop

Results $T = \begin{bmatrix} \alpha_1 & \beta_1 & \\ \beta_1 & \ddots & \\ & \ddots & \end{bmatrix} \rightarrow$ compute eigenvalue/eigenvector of T

A eigenvalues = T eigenvalues

A eigenvectors : $X = Q \cdot S$, where $Q = [q_1, q_2, q_3 \dots]$
S eigenvectors of T matrix.

If we use $a = A \cdot q_j \Rightarrow$ largest eigenvalue

If we use $a = A^{-1} q_j$ or $A \cdot a = q_j \Rightarrow$ smallest eigenvalues.

It is necessary $Q^T Q = 1 \Rightarrow$ we need to reorthogonalize the q vectors

$$r_j = r_j - \sum_{k=1}^{j-1} (q_k \cdot r_j \cdot q_k)$$

5, Direct methods

5.1. Jacobi method

Similarity transformation with P Jacobian rotation matrix

$$A_{k+1} = P_k^T A_k P_k \text{, where } P_k = \begin{array}{cc} \begin{array}{cccc} & p & q & \end{array} & \\ \begin{bmatrix} 1 & & & \\ & \cos\varphi & \sin\varphi & \\ & -\sin\varphi & \cos\varphi & \\ & & & 1 \end{bmatrix} & \begin{array}{c} \\ p \\ q \\ \\ \end{array} \end{array}$$

each step we modify p/q row and column

$$S_k = \sum_{i \neq j} (a_{ij}^k)^2 \rightarrow \text{minimize}$$

Very reliable

5.2. Householder method

$$A_{k+1} = P_k^T A_k P_k \text{, where } P_k = I - 2 w_k w_k^T \text{, where}$$

w_k has an explicit formula

$\Downarrow$

$A_k =$ $\rightarrow$ determine the eigenvalues and eigenvectors.

5.3. QR method

Each matrix $A = QR$, where Q orthogonal

R upper triangle matrix

for $k = 1, 2, 3 \dots$

$A_k = Q_k R_k$

$A_{k+1} = R_k Q_k$

Results: A_k : a_{ii} are the eigenvalues

Very good for triangle matrix $\Rightarrow$ QR used in Householder and Lanczos