

# Chapter 9 Approximating Eigenvalues

Baodong LIU  
baodong@sdu.edu.cn

November 19, 2019

## 9.1 Linear Algebra and Eigenvalues

- **Standard eigenvalue problem** : Given  $n \times n$  matrix **A**, find scalar  $\lambda$  and **nonzero vector** **x** such that

$$\mathbf{Ax} = \lambda \mathbf{x}$$

- $\lambda$  is eigenvalue;
- **x** is **corresponding eigenvector**.
- **Note that:**
  - ①  $\lambda$  may be complex even if **A** is real.
  - ② An  $n \times n$  matrix **A** has precisely  $n$  (not necessarily distinct )eigenvalues that are the roots of the polynomial

$$p(\lambda) = \det(\mathbf{A} - \lambda \mathbf{I}).$$

## 9.1 Linear Algebra and Eigenvalues

- **Standard eigenvalue problem** : Given  $n \times n$  matrix  $\mathbf{A}$ , find scalar  $\lambda$  and **nonzero vector**  $\mathbf{x}$  such that

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

- $\lambda$  is eigenvalue;
- $\mathbf{x}$  is **corresponding eigenvector**.

- **Note that:**

- ①  $\lambda$  may be complex even if  $\mathbf{A}$  is real.
- ② An  $n \times n$  matrix  $\mathbf{A}$  has precisely  $n$  (not necessarily distinct) eigenvalues that are the roots of the polynomial

$$p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}).$$

# Eigenvalues and Eigenvectors in Linear Algebra

- $\mathbf{x}$  can be viewed as **right eigenvector**, thus we can also define **left eigenvector**

$$\mathbf{y}^T \mathbf{A} = \lambda \mathbf{y}^T$$

- If  $\mathbf{y}$  is left eigenvector of  $\mathbf{A}$ , then it is right eigenvector of  $\mathbf{A}^T$ , since

$$\mathbf{A}^T \mathbf{y} = \lambda \mathbf{y}.$$

- **Spectrum** (谱) of  $\mathbf{A}$  = set of eigenvalues of  $\mathbf{A}$ , denoted by  $\lambda(\mathbf{A})$ .
- **Spectral radius**(谱半径) of  $\mathbf{A}$

$$\rho(\mathbf{A}) = \max\{|\lambda| : \lambda \in \lambda(\mathbf{A})\}.$$

- Matrix expands or shrinks any vector lying in direction of eigenvector by scalar factor.
- Expansion or contraction factor is given by corresponding eigenvalue  $\lambda$
- Eigenvalues and eigenvectors decompose complicated behavior of general linear transformation into simpler actions.

# Existence and Uniqueness

- Equation  $\mathbf{Ax} = \lambda\mathbf{x}$  is equivalent to

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$$

which has nonzero solution  $\mathbf{x}$ , if and only if, its matrix  $(\mathbf{A} - \lambda\mathbf{I})$  is singular.

- Eigenvalues of  $\mathbf{A}$  are roots  $\lambda_i$  of **characteristic polynomial**

$$p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

in  $\lambda$  of degree  $n$ .

- Fundamental Theorem of Algebra implies that  $n \times n$  matrix  $\mathbf{A}$  always has  $n$  eigenvalues, but they may not be real nor distinct
- Complex eigenvalues of **real matrix** occur in complex conjugate pairs: if  $\alpha + i\beta$  is eigenvalue of real matrix, then so is  $\alpha - i\beta$ , where  $i = \sqrt{-1}$ .

# Steps on Finding eigenvalues and eigenvectors of $\mathbf{A}$

- Solving  $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I}) = 0$ .
- Get its solution  $\lambda_j, j = 1, 2, \dots, n$ .
- Finding corresponding eigenvector  $\mathbf{v}_j$  for  $\lambda_j$ :

$$\mathbf{A}\mathbf{v}_j = \lambda_j\mathbf{v}_j, j = 1, 2, \dots, n$$

- **Note:**

- ① In practices, it is difficult to determine the root of an  $n$ th-degree polynomial.
- ② Approximation techniques are needed for finding eigenvalues and eigenvectors.

# Multiplicity (重根) and Diagonalizability

- Multiplicity is number of times root appears when polynomial is written as product of linear factors
- Eigenvalue of multiplicity 1 is simple(简根)
- **Defective matrix**(亏损矩阵) has eigenvalue of multiplicity  $k > 1$  with fewer than  $k$  linearly independent corresponding eigenvectors
- **Nondefective matrix**  $\mathbf{A}$  has  $n$  linearly independent eigenvectors, so it is diagonalizable

$$\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \mathbf{D}$$

where  $\mathbf{X}$  is nonsingular matrix of eigenvectors.



# Eigenspaces and Invariant Subspaces

- Eigenvectors can be scaled arbitrarily: if  $\mathbf{Ax} = \lambda\mathbf{x}$ , then

$$\mathbf{A}(\gamma\mathbf{x}) = \lambda(\gamma\mathbf{x})$$

for any scalar  $\gamma$ , so  $\gamma\mathbf{x}$  is also eigenvector corresponding to  $\lambda$ .

- Eigenvectors are usually normalized by requiring some norm of eigenvector to be 1.
- **Eigenspace:**  $\mathcal{S}_\lambda = \{\mathbf{x} : \mathbf{Ax} = \lambda\mathbf{x}\}$
- A subspace  $\mathcal{S}$  of  $\mathbb{R}^n$  (or  $\mathcal{C}^n$ ) is said to be **invariant subspace** if  $\mathbf{AS} \subseteq \mathcal{S}$
- For eigenvectors  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ ,  $\text{span}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  is invariant subspace

# Relevant Properties of Matrices

Property	Definition
diagonal	$a_{ij} = 0$ for $i \neq j$
tridiagonal	$a_{ij} = 0$ for $ i - j  > 1$
triangular	$a_{ij} = 0$ for $i > j$ (upper) $a_{ij} = 0$ for $i < j$ (lower)
Hessenberg	$a_{ij} = 0$ for $i > j + 1$ (upper) $a_{ij} = 0$ for $i < j - 1$ (lower)
orthogonal	$\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = \mathbf{I}$
unitary(酉矩阵)	$\mathbf{A}^H \mathbf{A} = \mathbf{A} \mathbf{A}^H = \mathbf{I}$
symmetric	$\mathbf{A} = \mathbf{A}^T$
Hermitian(厄密特矩阵)	$\mathbf{A} = \mathbf{A}^H$
normal(正规矩阵)	$\mathbf{A}^H \mathbf{A} = \mathbf{A} \mathbf{A}^H$

# Examples

- Transpose(转置):  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$
- Conjugate transpose(共轭转置):  
$$\begin{bmatrix} 1+i & 1+2i \\ 2-i & 2-2i \end{bmatrix}^H = \begin{bmatrix} 1-i & 2+i \\ 1-2i & 2+2i \end{bmatrix}$$
- Hermitian(厄密特矩阵):  $\begin{bmatrix} 1 & 1+i \\ 1-i & 2 \end{bmatrix},$
- nonHermitian:  $\begin{bmatrix} 1 & 1+i \\ 1+i & 2 \end{bmatrix}$
- Orthogonal:  $\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

- unitary(酉矩阵):  $\begin{bmatrix} \frac{\sqrt{2}}{2}i & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}i \end{bmatrix}$
- Normal(正规或正则矩阵):  $\begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \\ 2 & 0 & 1 \end{bmatrix}$
- Nonnormal:  $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

## Theorem 9.1

If  $\mathbf{A}$  is a matrix and  $\lambda_1, \dots, \lambda_k$  are distinct eigenvalues of  $\mathbf{A}$  with associated eigenvectors

$$\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(k)},$$

then

$$\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(k)}\}$$

is linearly independent.

## Definition 9.2

A set of vectors

$$\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \dots, \mathbf{v}^{(n)}$$

is called **orthogonal** if

$$(\mathbf{v}^{(i)})^T \mathbf{v}^{(j)} = 0, \quad \text{for all } i \neq j.$$

If, in addition ,

$$(\mathbf{v}^{(i)})^T \mathbf{v}^{(i)} = 1, \quad \text{for all } i = 1, 2, \dots, n,$$

then the set is **orthonormal**.

### Theorem 9.3

An orthogonal set of vectors that does not contain the zero vector is linearly independent.

### Definition 9.4

A matrix  $\mathbf{P}$  is said to be an orthogonal matrix if  $\mathbf{P}^{-1} = \mathbf{P}^T$ .

### Definition 9.5

Two matrices  $\mathbf{A}$  and  $\mathbf{B}$  are said to be **similar** if a nonsingular matrix  $\mathbf{S}$  exists with

$$\mathbf{A} = \mathbf{S}^{-1}\mathbf{B}\mathbf{S}.$$

## Theorem 9.6

- Suppose  $\mathbf{A}$  and  $\mathbf{B}$  are similar matrices and  $\lambda$  is an eigenvalue of  $\mathbf{A}$  with associated eigenvector  $\mathbf{x}$ .
- Then  $\lambda$  is also an eigenvalue of  $\mathbf{B}$ , and if  $\mathbf{A} = \mathbf{S}^{-1}\mathbf{B}\mathbf{S}$ , then  $\mathbf{S}\mathbf{x}$  is an eigenvector associated with  $\lambda$  for the matrix  $\mathbf{B}$ .

## Theorem 9.7 (Schur)

Let  $\mathbf{A}$  be an arbitrary matrix. A nonsingular matrix  $\mathbf{U}$  exists with the property that

$$\mathbf{T} = \mathbf{U}^{-1}\mathbf{A}\mathbf{U}$$

where  $\mathbf{T}$  is an upper-triangular matrix whose diagonal entries consist of the eigenvalues of  $\mathbf{A}$ .



## Theorem 9.8

If  $\mathbf{A}$  is a **symmetric matrix** and  $\mathbf{D}$  is a **diagonal matrix** whose diagonal entries are the eigenvalues of  $\mathbf{A}$ , then there exists an orthogonal matrix  $\mathbf{P}$  such that

$$\mathbf{D} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} = \mathbf{P}^T\mathbf{A}\mathbf{P}$$

## Corollary 9.9

If  $\mathbf{A}$  is a symmetric  $n \times n$  matrix, then the eigenvalues of  $\mathbf{A}$  are real numbers, and there exist  $n$  eigenvectors of  $\mathbf{A}$  that form an orthonormal set.

## 推论9.9证明

记  $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$  为  $\mathbf{A}$  的  $n$  个特征向量构成的矩阵； $\mathbf{D} = (d_{ii})$  为  $\mathbf{A}$  的  $n$  个特征值构成的对角矩阵；则

$$\mathbf{D} = \mathbf{v}^{-1} \mathbf{A} \mathbf{v} \text{ 或 } \mathbf{A} \mathbf{v} = \mathbf{D} \mathbf{v}.$$

对任一  $1 \leq i \leq n$  有

$$\mathbf{A} \mathbf{v}_i = d_{ii} \mathbf{v}_i,$$

其中  $d_{ii}$  是  $\mathbf{A}$  的特征值， $\mathbf{v}_i$  是其对应的特征向量.

等式两端同乘  $\mathbf{v}_i^T$ ，得

$$\mathbf{v}_i^T \mathbf{A} \mathbf{v}_i = d_{ii} \mathbf{v}_i^T \mathbf{v}_i.$$

由矩阵  $\mathbf{A}$  是对称的，则  $\mathbf{v}_i^T \mathbf{A} \mathbf{v}_i$  和  $\mathbf{v}_i^T \mathbf{v}_i$  都是实数，且  $\mathbf{v}_i^T \mathbf{v}_i = 1$ ，从而特征值  $d_{ii} = \mathbf{v}_i^T \mathbf{A} \mathbf{v}_i$  也是实数. ■

## Theorem 9.10

A symmetric matrix  $\mathbf{A}$  is positive definite if and only if all the eigenvalues of  $\mathbf{A}$  are positive.

## Theorem 4.11 (Gerschgorin Circle Theorem—圆盘定理)

- Let  $\mathbf{A}$  be an  $n \times n$  matrix
- $\mathbb{R}_i$  denote the circle in the complex plane with center  $a_{ii}$  and radius  $\sum_{j=1, j \neq i}^n |a_{ij}|$ ;
- that is

$$\mathbb{R}_i = \left\{ z \in \mathcal{C} \mid |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}| \right\}$$

where  $\mathcal{C}$  denotes the complex plane.

- The eigenvalues of  $\mathbf{A}$  are contained with  $\mathbb{R} = \cup_{i=1}^n \mathbb{R}_i$
- Moreover, the union (并集) of any  $k$  of these circles that do not intersect the remaining  $(n - k)$  contains precisely  $k$  (counting multiplicities) of the eigenvalues.

# 圆盘定理的证明

- Suppose that  $\lambda$  is an eigenvalue of  $\mathbf{A}$  with associated eigenvector  $\mathbf{x}$ , where  $\|\mathbf{x}\|_\infty = 1$ .
- Since  $\mathbf{Ax} = \lambda\mathbf{x}$ , the equivalent component representation is

$$\sum_{j=1}^n a_{ij} x_j = \lambda x_i, \text{ for each } i = 1, 2, \dots, n.$$

- If  $k$  is an integer with  $|x_k| = \|\mathbf{x}\|_\infty = 1$ , this equation, with  $i = k$ , implies that

$$\sum_{j=1}^n a_{kj} x_j = \lambda x_k.$$

- Thus

$$\sum_{j=1, j \neq k}^n a_{kj} x_j = \lambda x_k - a_{kk} x_k = (\lambda - a_{kk}) x_k,$$

- So

$$|\lambda - a_{kk}| \cdot |x_k| = \left| \sum_{j=1, j \neq k}^n a_{kj} x_j \right| \leq \sum_{j=1, j \neq k}^n |a_{kj}| |x_j|.$$

- Since  $|x_j| \leq |x_k| = 1$ , for all  $j = 1, 2, \dots, n$ ,

$$|\lambda - a_{kk}| \leq \sum_{j=1, j \neq k}^n |a_{kj}|$$

- Thus,  $\lambda \in R_k$ , which proves the first assertion in the theorem.
- 定理的第二部分证明需要连通性理论, 不再证明.

# Problem Transformations

- **Shift** : If  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$  and  $\sigma$  is any scalar, then

$$(\mathbf{A} - \sigma\mathbf{I})\mathbf{x} = (\lambda - \sigma)\mathbf{x},$$

so eigenvalues of **shifted matrix** (转移矩阵) are shifted eigenvalues of original matrix

- **Inversion** : If  $\mathbf{A}$  is nonsingular and  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$  with  $\mathbf{x} \neq 0$ , then  $\lambda \neq 0$  and  $\mathbf{A}^{-1}\mathbf{x} = \frac{1}{\lambda}\mathbf{x}$ , so eigenvalues of inverse are reciprocals of eigenvalues of original matrix
- **Powers** : If  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ , then  $\mathbf{A}^k\mathbf{x} = \lambda^k\mathbf{x}$ , so eigenvalues of power of matrix are same power of eigenvalues of original matrix
- **Polynomial** : If  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$  and  $p(t)$  is polynomial, then

$$p(\mathbf{A})\mathbf{x} = p(\lambda)\mathbf{x},$$

so eigenvalues of polynomial in matrix are values of polynomial evaluated at eigenvalues of original matrix

# Similarity Transformation

- $\mathbf{B}$  is similar to  $\mathbf{A}$  if there is nonsingular matrix  $\mathbf{P}$  such that

$$\mathbf{B} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}.$$

- Then

$$\mathbf{B}\mathbf{y} = \lambda\mathbf{y}, \Rightarrow \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{y} = \lambda\mathbf{y}, \Rightarrow \mathbf{A}(\mathbf{P}\mathbf{y}) = \lambda(\mathbf{P}\mathbf{y})$$

so  $\mathbf{A}$  and  $\mathbf{B}$  have same eigenvalues, and if  $\mathbf{y}$  is eigenvector of  $\mathbf{B}$ , then  $\mathbf{x} = \mathbf{P}\mathbf{y}$  is eigenvector of  $\mathbf{A}$ .

- Similarity transformations preserve eigenvalues and eigenvectors are easily recovered



## 9.2 Computing Eigenvalues and Eigenvectors: power method

### Iterative Power method

- assume that the  $n \times n$  matrix  $\mathbf{A}$  has  $n$  eigenvalues

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

with an associated collection of linearly independent eigenvectors

$$\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \dots, \mathbf{v}^{(n)}\}.$$

- If  $\mathbf{x}$  is any vector in  $\mathbb{R}^n$ , then constants  $\beta_1, \beta_2, \dots, \beta_n$  exist with

$$\mathbf{x} = \beta_1 \mathbf{v}^{(1)} + \beta_2 \mathbf{v}^{(2)} + \cdots + \beta_n \mathbf{v}^{(n)} = \sum_{j=1}^n \beta_j \mathbf{v}^{(j)}$$

- Multiplying both sides of this equation by  $\mathbf{A}, \mathbf{A}^2, \dots, \mathbf{A}^k$ , we obtain:

$$\begin{aligned}\mathbf{Ax} &= \sum_{j=1}^n \beta_j \mathbf{A} \mathbf{v}^{(j)} = \sum_{j=1}^n \beta_j \lambda_j \mathbf{v}^{(j)} \\ \mathbf{A}^2 \mathbf{x} &= \sum_{j=1}^n \beta_j \lambda_j \mathbf{A} \mathbf{v}^{(j)} = \sum_{j=1}^n \beta_j \lambda_j^2 \mathbf{v}^{(j)} \\ &\vdots \\ \mathbf{A}^k \mathbf{x} &= \sum_{j=1}^n \beta_j \lambda_j^k \mathbf{v}^{(j)} = \lambda_1^k \sum_{j=1}^n \beta_j \left( \frac{\lambda_j}{\lambda_1} \right)^k \mathbf{v}^{(j)} \\ &= \lambda_1^k \left( \beta_1 \mathbf{v}^{(1)} + \sum_{j=2}^n \beta_j \left( \frac{\lambda_j}{\lambda_1} \right)^k \mathbf{v}^{(j)} \right)\end{aligned}$$

- Since  $|\lambda_1| > |\lambda_j|$  for all  $j = 2, 3, \dots, n$ , we have

$$\lim_{k \rightarrow \infty} (\lambda_j / \lambda_1)^k = 0,$$

$$\lim_{k \rightarrow \infty} \mathbf{A}^k \mathbf{x} = \lim_{k \rightarrow \infty} \lambda_1^k \beta_1 \mathbf{v}^{(1)} \quad (1)$$

- This gives us the way to proceed to find  $\lambda_1$  and an associated eigenvector.
- but we can not use the sequence in (1) directly since it converges to zero if  $\lambda_1 < 1$  and diverges if  $\lambda_1 > 1$ , provided, of course, that  $\beta_1 \neq 0$ .
- Advantage can be made of the relationship expressed in Eq.(1) by scaling the powers of  $\mathbf{A}^k \mathbf{x}$  in an appropriate manner to ensure that the limit in Eq.(1) is finite and nonzero

# Scaling Method(比例方法):

## Step 1

- Choose an arbitrary unit vector  $\mathbf{x}^{(0)}$  relative to  $\|\cdot\|_\infty$ .
- Suppose a component  $x_{p_0}^{(0)}$  of  $\mathbf{x}^{(0)}$  with

$$x_{p_0}^{(0)} = 1 = \|\mathbf{x}^{(0)}\|_\infty$$

- Let  $\mathbf{y}^{(1)} = \mathbf{A}\mathbf{x}^{(0)}$ , and define:  $\mu^{(1)} = y_{p_0}^{(1)}$ .
- With this notation ,

$$\begin{aligned}\mu^{(1)} &= y_{p_0}^{(1)} = \frac{y_{p_0}^{(1)}}{x_{p_0}^{(0)}} = \frac{\beta_1 \lambda_1 v_{p_0}^{(1)} + \sum_{j=2}^n \beta_j \lambda_j v_{p_0}^{(j)}}{\beta_1 v_{p_0}^{(1)} + \sum_{j=2}^n \beta_j v_{p_0}^{(j)}} \\ &= \lambda_1 \left[ \frac{\beta_1 v_{p_0}^{(1)} + \sum_{j=2}^n \beta_j (\lambda_j / \lambda_1) v_{p_0}^{(j)}}{\beta_1 v_{p_0}^{(1)} + \sum_{j=2}^n \beta_j v_{p_0}^{(j)}} \right]\end{aligned}$$

- Then let  $p_1$  be the least integer such that

$$y_{p_1}^{(1)} = \| \mathbf{y}^{(1)} \|_{\infty}$$

- Define  $\mathbf{x}^{(1)}$  by

$$\mathbf{x}^{(1)} = \frac{1}{y_{p_1}^{(1)}} \mathbf{y}^{(1)} = \frac{1}{y_{p_1}^{(1)}} \mathbf{A} \mathbf{x}^{(0)}$$

- Then

$$x_{p_1}^{(1)} = 1 = \| \mathbf{x}^{(1)} \|_{\infty}$$

## Step 2

- Define

$$\mathbf{y}^{(2)} = \mathbf{A}\mathbf{x}^{(1)} = \frac{1}{y_{p_1}^{(1)}} \mathbf{A}^2 \mathbf{x}^{(0)}$$

- Let

$$\begin{aligned} \mu^{(2)} = y_{p_1}^{(2)} &= \frac{y_{p_1}^{(2)}}{x_{p_1}^{(1)}} = \frac{\left[ \beta_1 \lambda_1^2 v_{p_1}^{(1)} + \sum_{j=2}^n \beta_j \lambda_j^2 v_{p_1}^{(j)} \right] / y_{p_1}^{(1)}}{\left[ \beta_1 \lambda_1 v_{p_1}^{(1)} + \sum_{j=2}^n \beta_j \lambda_j v_{p_1}^{(j)} \right] / y_{p_1}^{(1)}} \\ &= \lambda_1 \left[ \frac{\beta_1 v_{p_1}^{(1)} + \sum_{j=2}^n \beta_j (\lambda_j / \lambda_1)^2 v_{p_1}^{(j)}}{\beta_1 v_{p_1}^{(1)} + \sum_{j=2}^n \beta_j (\lambda_j / \lambda_1) v_{p_1}^{(j)}} \right]. \end{aligned}$$

- Let  $p_2$  be the smallest integer with

$$|y_{p_2}^{(2)}| = \|\mathbf{y}^{(2)}\|_\infty$$

- Define:  $\mathbf{x}^{(2)} = \frac{1}{y_{p_2}^{(2)}} \mathbf{y}^{(2)} = \frac{1}{y_{p_2}^{(2)}} \mathbf{A}\mathbf{x}^{(1)} = \frac{1}{y_{p_2}^{(2)} y_{p_1}^{(1)}} \mathbf{A}^2 \mathbf{x}^{(0)}.$

- In a similar manner, define sequences of vectors  $\{\mathbf{x}^{(m)}\}_{m=1}^{\infty}$  and  $\{\mathbf{y}^{(m)}\}_{m=1}^{\infty}$ , and a sequence of scalars  $\{\mu^{(m)}\}_{m=1}^{\infty}$ .
- $\mathbf{y}^{(m)} = \mathbf{A}\mathbf{x}^{(m-1)}$
- $\mu^{(m)} = y_{p_{m-1}}^{(m)} = \lambda_1 \left[ \frac{\beta_1 v_{p_{m-1}}^{(1)} + \sum_{j=2}^n \beta_j (\lambda_j / \lambda_1)^m v_{p_{m-1}}^{(j)}}{\beta_1 v_{p_{m-1}}^{(1)} + \sum_{j=2}^n \beta_j (\lambda_j / \lambda_1)^{m-1} v_{p_{m-1}}^{(j)}} \right]$
- Let  $p_m$  be the smallest integer with

$$|y_{p_m}^{(m)}| = \|\mathbf{y}^{(m)}\|_{\infty}$$

- Let  $\mathbf{x}^{(m)} = \frac{\mathbf{y}^{(m)}}{y_{p_m}^{(m)}} = \frac{\mathbf{A}^m \mathbf{x}^{(0)}}{\prod_{k=1}^m y_{p_k}^{(k)}}$

- At each step,  $p_m$  is used to represent the smallest integer for which

$$|y_{p_m}^{(m)}| = \| \mathbf{y}^{(m)} \|_{\infty}$$

- Since  $|\lambda_j/\lambda_1| < 1$  for each  $j = 2, 3, \dots, n$ , then

$$\lim_{m \rightarrow \infty} \mu^{(m)} = \lambda_1,$$



# ALGORITHM: the Power method

- To approximate the dominant eigenvalue and an associated eigenvector of the  $n \times n$  matrix  $\mathbf{A}$  given a nonzero vector  $\mathbf{x}$
- **INPUT** dimension  $n$ ; matrix  $\mathbf{A}$ ; vector  $\mathbf{x}$ ; tolerance  $TOL$ ; maximum number of iterations  $N$ .
- **OUTPUT** approximate eigenvalue  $\mu$ ; approximate eigenvector  $\mathbf{x}$  (with  $\|\mathbf{x}\|_\infty = 1$ ) or a message that the maximum number of iterations was exceeded.
- **Step 1** Set  $k = 1$ .
- **Step 2** Find the smallest integer  $p$  with  $1 \leq p \leq n$  and  $|x_p| = \|\mathbf{x}\|_\infty$ .
- **Step 3** Set  $\mathbf{x} = \mathbf{x}/x_p$ .

- **Step 4** While ( $k \leq N$ ) do Steps 5-11.
  - **Step 5** Set  $\mathbf{y} = \mathbf{A}\mathbf{x}$ .
  - **Step 6** Set  $\mu = y_p$ .
  - **Step 7** Find the smallest integer  $p$  with  $1 \leq p \leq n$  and  $|y_p| = \|\mathbf{y}\|_\infty$ .
  - **Step 8** If  $y_p = 0$  then
    - OUTPUT ('Eigenvector',  $\mathbf{x}$ );
    - OUTPUT ('A has the eigenvalue 0, select a new vector  $\mathbf{x}$  and restart');
    - STOP.
  - **Step 9** Set  $ERR = \|\mathbf{x} - (\mathbf{y}/y_p)\|_\infty$ ;  $\mathbf{x} = \mathbf{y}/y_p$ .
  - **Step 10** If  $ERR < TOL$  then OUTPUT ( $\mu, \mathbf{x}$ ) ;(Procedure completed successfully.).STOP.
  - **Step 11** Set  $k = k + 1$ .
- **Step 12** OUTPUT ('Maximum number of iterations exceeded');(Procedure completed unsuccessfully.)
- STOP.

## Remarks:

- **Choosing the smallest integer of  $\|\cdot\|_\infty$ :** Choosing, in Step 7, the smallest integer  $p_m$  for which  $|p_m| = \|\mathbf{y}^{(m)}\|_\infty$  will generally ensure that this index eventually becomes invariant.
- **Rate of Convergence:** The rate at which  $\{\mu^{(m)}\}_{m=1}^\infty$  converges to  $\lambda_1$  is determined by the ratios  $|\lambda_j/\lambda_1|^m$ , for  $j = 2, 3, \dots, n$ , and in particular by  $|\lambda_2/\lambda_1|^m$ .
- The rate of convergence is  $O(|\lambda_2/\lambda_1|^m)$ , so there is a constant  $k$  such that for large  $m$ ,

$$|\mu^{(m)} - \lambda_1| \approx k \left| \frac{\lambda_2}{\lambda_1} \right|^m,$$

which implies that

$$\lim_{m \rightarrow \infty} \frac{|\mu^{(m+1)} - \lambda_1|}{|\mu^{(m)} - \lambda_1|} \approx \left| \frac{\lambda_2}{\lambda_1} \right| < 1.$$

## Inverse Power method– 求任一特征值

- The **Inverse Power method** is a modification of the Power method that gives faster convergence.
- It is used to determine the eigenvalue of  $\mathbf{A}$  that is closest to a specified number  $q$ .
- Assume that the matrix  $\mathbf{A}$  has eigenvalues

$$\lambda_1, \dots, \lambda_n$$

with linearly independent eigenvectors

$$\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}.$$

- Suppose that  $q \neq \lambda_i$  for  $i = 1, 2, 3, \dots, n$
- We can easily get that the eigenvalues of the matrix  $(\mathbf{A} - q\mathbf{I})^{-1}$  are

$$\frac{1}{\lambda_1 - q}, \frac{1}{\lambda_2 - q}, \dots, \frac{1}{\lambda_n - q}$$

with eigenvectors

$$\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$$

# Inverse Power Method

- Applying the Power method to  $(\mathbf{A} - q\mathbf{I})^{-1}$  gives

$$\mathbf{y}^{(m)} = (\mathbf{A} - q\mathbf{I})^{-1} \mathbf{x}^{(m-1)}$$

- Let

$$\mu^{(m)} = y_{p_{m-1}}^{(m)} = \frac{y_{p_{m-1}}^{(m)}}{x_{p_{m-1}}^{(m-1)}} = \frac{\sum_{j=1}^n \beta_j \frac{1}{(\lambda_j - q)^m} v_{p_{m-1}}^{(j)}}{\sum_{j=1}^n \beta_j \frac{1}{(\lambda_j - q)^{m-1}} v_{p_{m-1}}^{(j)}} \quad (2)$$

- Let  $p_m$  represents the smallest integer for which

$$|y_{p_m}^{(m)}| = \| \mathbf{y}^{(m)} \|_{\infty}$$

- Define

$$\mathbf{x}^{(m)} = \frac{\mathbf{y}^{(m)}}{y_{p_m}^{(m)}}$$

- The sequence  $\{\mu^{(m)}\}$  in Eq . (2) converges to

$$\frac{1}{|\lambda_k - q|} = \max_{1 \leq j \leq n} \frac{1}{|\lambda_j - q|}$$

and  $\lambda_k$  is the eigenvalue of  $\mathbf{A}$  closest to  $q$  .

# Convergence Rate

- With  $k$  known , Eq.(2) can be written as

$$\mu^{(m)} = \frac{1}{\lambda_k - q} \left[ \frac{\beta_k v_{p_{m-1}}^{(k)} + \sum_{j=1, j \neq k}^n \beta_j \left[ \frac{\lambda_k - q}{\lambda_j - q} \right]^m v_{p_{m-1}}^{(j)}}{\beta_k v_{p_{m-1}}^{(k)} + \sum_{j=1, j \neq k}^n \beta_j \left[ \frac{\lambda_k - q}{\lambda_j - q} \right]^{m-1} v_{p_{m-1}}^{(j)}} \right] \quad (3)$$

- Thus the choice of  $q$  determines the convergence.
- $\frac{1}{\lambda_k - q}$  is a unique dominant eigenvalue of  $(\mathbf{A} - q\mathbf{I})^{-1}$ .
- The closer  $q$  is to an eigenvalue  $\lambda_k$  of  $\mathbf{A}$  , the faster the convergence since the convergence is of order

$$O \left( \left| \frac{(\lambda - q)^{-1}}{(\lambda_k - q)^{-1}} \right|^m \right) = O \left( \left| \frac{(\lambda_k - q)}{(\lambda - q)} \right|^m \right)$$

where  $\lambda$  represents the eigenvalue of  $\mathbf{A}$  that is second closest to  $q$  .



- The determination of  $\mathbf{y}^{(m)}$  in iteration

$$\mathbf{y}^{(m)} = (\mathbf{A} - q\mathbf{I})^{-1}\mathbf{x}^{(m-1)}$$

can be obtained from the equation

$$(\mathbf{A} - q\mathbf{I})\mathbf{y}^{(m)} = \mathbf{x}^{(m)}$$

- In general , Gaussian elimination with pivoting is used to solve this system.
- Although the Inverse Power method requires the solution of an  $n \times n$  system at each step , the multipliers can be saved to reduce the computation .
- The selection of  $q$  can be based on the Gerschgorin Circle Theorem or on any other means of localizing an eigenvalue.

# ALGORITHM : the Inverse Power method

To approximate an eigenvalue and an associated eigenvector of the  $n \times n$  matrix  $\mathbf{A}$  given a nonzero vector  $\mathbf{x}$ :

- **INPUT:** Dimension  $n$ ; matrix  $\mathbf{A}$ ; vector  $\mathbf{x}$ ; tolerance  $TOL$ ; maximum number of iterations  $N$ .
- **OUTPUT:** Approximate eigenvalue  $\mu$ ; approximate eigenvector  $\mathbf{x}$  (with  $\|\mathbf{x}\|_\infty = 1$ ) or a message that the maximum number of iterations was exceeded.
- **Step 1:** Set  $q = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$ .
- **Step 2:** Set  $k = 1$ .
- **Step 3:** Find the smallest integer  $p$  with  $1 \leq p \leq n$  and  $|x_p| = \|\mathbf{x}\|_\infty$ .
- **Step 4** Set  $\mathbf{x} = \mathbf{x} / x_p$ .

- **Step 5:** While ( $k \leq N$ ) do Steps 6-12.
  - **Step 6:** Set the linear system  $(\mathbf{A} - q\mathbf{I})\mathbf{y} = \mathbf{x}$ .
  - **Step 7:** If the system doesn't have a unique solution, then OUTPUT (' $q$  is an eigenvalue',  $q$ );
  - **Step 8:** Set  $\mu = y_p$ .
  - **Step 9:** Find the smallest integer  $p$  with  $1 \leq p \leq n$  and  $|y_p| = \|\mathbf{y}\|_\infty$ .
  - **Step 10:** Set  $ERR = \|\mathbf{x} - (\mathbf{y}/y_p)\|_\infty$ ;  $\mathbf{x} = \mathbf{y}/y_p$ .
  - **Step 11:** If  $ERR < TOL$  then set  $\mu = (1/\mu) + q$ ;
    - OUTPUT ( $\mu, \mathbf{x}$ );
    - (Procedure was successfully.)
    - STOP.
  - **Step 12:** Set  $k = k + 1$ .
- **Step 13:** OUTPUT ('Maximum number of iterations exceeded'); (Procedure completed unsuccessfully.)
- STOP.

# Rayleigh Quotient Iteration–Rayleigh 商迭代方法

- If  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  with respect to the eigenvalue  $\lambda$ , then  $\mathbf{Ax} = \lambda\mathbf{x}$ . So,  $\mathbf{x}^T \mathbf{Ax} = \lambda \mathbf{x}^T \mathbf{x}$  and

$$\lambda = \frac{\mathbf{x}^T \mathbf{Ax}}{\mathbf{x}^T \mathbf{x}} = \frac{\mathbf{x}^T \mathbf{Ax}}{\|\mathbf{x}\|_2^2}.$$

- The quantity  $\frac{\mathbf{x}^T \mathbf{Ax}}{\mathbf{x}^T \mathbf{x}}$ , known as Rayleigh Quotient, has many useful properties.
- It can be used to accelerate the convergence of a method, such as power iteration or inverse power iteration method, since at the  $k$ th iteration, the Rayleigh quotient  $\frac{\mathbf{x}_k^T \mathbf{Ax}_k}{\mathbf{x}_k^T \mathbf{x}_k}$  gives a better approximation than the basic method alone.

# Symmetric Power Method(对称幂法)

- Suppose that the  $n \times n$  matrix  $\mathbf{A}$  is **symmetric**, thus  $\mathbf{A}$  has  $n$  eigenvalues

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$$

with real number, and a collection of eigenvectors

$$\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$$

which are orthonormal.

- For any vector  $\mathbf{x}_0$  in  $\mathbb{R}^n$ , there exists a set of constants  $\beta_1, \beta_2, \dots, \beta_n$ , such that:

$$x_0 = \beta_1 \mathbf{v}^{(1)} + \beta_2 \mathbf{v}^{(2)} + \cdots + \beta_n \mathbf{v}^{(n)}$$

- Then for the power of  $\mathbf{x}_k = \mathbf{A}^k \mathbf{x}_0$ , it can be rewritten as

$$\mathbf{x}_k = \mathbf{A}^k \mathbf{x} = \beta_1 \lambda_1^k \mathbf{v}^{(1)} + \beta_2 \lambda_2^k \mathbf{v}^{(2)} + \cdots + \beta_n \lambda_n^k \mathbf{v}^{(n)}$$

- Since the set of eigenvectors  $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$  are orthonormal, it can be seen that

$$\mathbf{x}_k^T \mathbf{x}_k = \sum_{j=1}^n \beta_j^2 \lambda_j^{2k} = \beta_1^2 \lambda_1^{2k} \left\{ 1 + \sum_{j=2}^n \left( \frac{\beta_j}{\beta_1} \right)^2 \left( \frac{\lambda_j}{\lambda_1} \right)^{2k} \right\},$$

and

$$\begin{aligned} \mathbf{x}_k^T \mathbf{A} \mathbf{x}_k &= \sum_{j=1}^n \beta_j^2 \lambda_j^{2k+1} \\ &= \beta_1^2 \lambda_1^{2k+1} \left\{ 1 + \sum_{j=2}^n \left( \frac{\beta_j}{\beta_1} \right)^2 \left( \frac{\lambda_j}{\lambda_1} \right)^{2k+1} \right\}. \end{aligned}$$

- Thus

$$\lim_{k \rightarrow \infty} \frac{\mathbf{x}_k^T \mathbf{A} \mathbf{x}_k}{\mathbf{x}_k^T \mathbf{x}_k} = \lambda_1$$

$$\lim_{k \rightarrow \infty} \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|_2} = \frac{\mathbf{v}^{(1)}}{\|\mathbf{v}^{(1)}\|_2}.$$

- The rate of convergence of the modified procedure given in Rayleigh Method for symmetric matrix is  $O(|\lambda_2/\lambda_1|^{2m})$ .
- The sequence  $\{\mu^{(m)}\}_{m=1}^{\infty}$  is still linearly convergent.

# ALGORITHM    Symmetric Power Method

- To approximate the dominant eigenvalue and an associated eigenvector of the  $n \times n$  symmetric matrix  $\mathbf{A}$ , given a nonzero vector  $\mathbf{x}$  :
- **INPUT** dimension  $n$ ; matrix  $\mathbf{A}$ ; vector  $\mathbf{x}$ ; tolerance  $TOL$ ; maximum number of iterations  $N$ .
- **OUTPUT** approximate eigenvalue  $\mu$ ; approximate eigenvector  $\mathbf{x}$  (with  $\|\mathbf{x}\|_2 = 1$ ) or a message that the maximum number of iterations was exceeded.
- **Step 1** Set  $k = 1$ ;

$$\mathbf{x} = \mathbf{x} / \|\mathbf{x}\|_2.$$



- **Step 2** While ( $k \leq N$ ) do Steps 3-8.
  - **Step 3** Set  $\mathbf{y} = \mathbf{A}\mathbf{x}$ .
  - **Step 4** Set  $\mu = \mathbf{x}^T \mathbf{y}$ .
  - **Step 5** If  $\|\mathbf{y}\|_2 = 0$ , then OUTPUT ('Eigenvector',  $\mathbf{x}$ ); OUTPUT (' $\mathbf{A}$  has eigenvalue 0, select new vector  $\mathbf{x}$  and restart'); STOP.
  - **Step 6** Set

$$ERR = \left\| \mathbf{x} - \frac{\mathbf{y}}{\|\mathbf{y}\|_2} \right\|_2$$

$$\mathbf{x} = \mathbf{y} / \|\mathbf{y}\|_2.$$

- **Step 7** If  $ERR < TOL$  then OUTPUT ( $\mu, \mathbf{x}$ ); (Procedure completed successfully.) STOP.
  - **Step 8** Set  $k = k + 1$ .
- **Step 9** OUTPUT ('Maximum number of iterations exceeded'); (Procedure completed unsuccessfully.) STOP.

- If  $\mathbf{A}$  is symmetric, then for any real number  $q$ ,  $(\mathbf{A} - q\mathbf{I})^{-1}$  is also symmetric.
- the Symmetric Power method, Algorithm can be applied to  $(\mathbf{A} - q\mathbf{I})^{-1}$  to speed the convergence to

$$O\left(\left|\frac{\lambda_k - q}{\lambda - q}\right|^{2m}\right)$$

- Numerous techniques are available for obtaining approximations to other eigenvalues are the same as those of  $\mathbf{A}$ , except that the dominant eigenvalue of  $\mathbf{A}$  is replaced by the eigenvalue 0.

# Deflation–求全部特征值的收缩算法

## Theorem 4.12

- Suppose  $\lambda_1, \lambda_2, \dots, \lambda_n$  are eigenvalues of  $\mathbf{A}$  with associated eigenvectors  $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(n)}$  and  $\lambda_1$  has multiplicity 1.
- Let  $\mathbf{x}$  be a vector with  $\mathbf{x}^T \mathbf{v}^{(1)} = 1$ .
- Then the matrix

$$\mathbf{B} = \mathbf{A} - \lambda_1 \mathbf{v}^{(1)} \mathbf{x}^T$$

has eigenvalues  $0, \lambda_2, \lambda_3, \dots, \lambda_n$  with associated eigenvectors  $\mathbf{v}^{(1)}, \mathbf{w}^{(2)}, \mathbf{w}^{(3)}, \dots, \mathbf{w}^{(n)}$ , where  $\mathbf{v}^{(i)}$  and  $\mathbf{w}^{(i)}$  are related by the equation

$$\mathbf{v}^{(i)} = (\lambda_i - \lambda_1) \mathbf{w}^{(i)} + \lambda_1 (\mathbf{x}^T \mathbf{w}^{(i)}) \mathbf{v}^{(1)} \quad (4)$$

for each  $i = 2, 3, \dots, n$ . ■

- After eigenvalue  $\lambda_1$  and corresponding eigenvector  $\mathbf{x}_1$  have been computed, then additional eigenvalues  $\lambda_2, \lambda_3, \dots, \lambda_n$  of  $\mathbf{A}$  can be computed by deflation, which effectively removes known eigenvalue.
- Let  $\mathbf{H}$  be any nonsingular matrix such that  $\mathbf{H}\mathbf{x}_1 = \alpha\mathbf{e}_1$  scalar multiple of first column of identity matrix (Householder transformation is good choice for  $\mathbf{H}$ )
- Then similarity transformation determined by  $\mathbf{H}$  transforms  $\mathbf{A}$  into form

$$\mathbf{H}\mathbf{A}\mathbf{H}^{-1} = \begin{bmatrix} \lambda_1 & \mathbf{b}^T \\ 0 & \mathbf{B} \end{bmatrix}$$

where  $\mathbf{B}$  is matrix of order  $n - 1$  having eigenvalues  $\lambda_2, \lambda_3, \dots, \lambda_n$ .

- Thus, we can work with  $\mathbf{B}$  to compute next eigenvalue  $\lambda_2$ .
- Moreover, if  $\mathbf{y}_2$  is eigenvector of  $\mathbf{B}$  corresponding to  $\lambda_2$ , then

$$\mathbf{x}_2 = \mathbf{H}^{-1} \begin{bmatrix} \gamma \\ \mathbf{y}_2 \end{bmatrix}, \text{ where } \gamma = \frac{\mathbf{b}^T \mathbf{y}_2}{\lambda_2 - \lambda_1}$$

is eigenvector corresponding to  $\lambda_2$  for original matrix  $\mathbf{A}$ , provided  $\lambda_1 \neq \lambda_2$ .

- Process can be repeated to find additional eigenvalues and eigenvectors

- Alternative approach to deflation is to let  $\mathbf{u}_1$  be any vector such that

$$\mathbf{u}_1^T \mathbf{x}_1 = \lambda_1$$

- Then the matrix  $\mathbf{A} - \mathbf{x}_1 \mathbf{u}_1^T$  has eigenvalues

$$0, \lambda_2, \dots, \lambda_n.$$

- Possible choices for  $\mathbf{u}_1$  include
  - $\mathbf{u}_1 = \lambda_1 \mathbf{x}_1$ , if  $\mathbf{A}$  is symmetric and  $\mathbf{x}_1$  is normalized so that  $\|\mathbf{x}_1\|_2 = 1$ .
  - $\mathbf{u}_1 = \lambda_1 \mathbf{y}_1$ , where  $\mathbf{y}_1$  is corresponding left eigenvector (i.e.,  $\mathbf{A}^T \mathbf{y}_1 = \lambda_1 \mathbf{y}_1$ ) normalized so that  $\mathbf{y}_1^T \mathbf{x}_1 = 1$ .
  - $\mathbf{u}_1 = \mathbf{A}^T \mathbf{e}_k$ , if  $\mathbf{x}_1$  is normalized so that  $\|\mathbf{x}_1\|_\infty = 1$  and  $k$ th component of  $\mathbf{x}_1$  is 1.

- **Wielandt deflation** proceeds from defining

$$\mathbf{x} = \frac{1}{\lambda_1 v_i^{(1)}} \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix} \quad (5)$$

where  $v_i^{(1)}$  is a coordinate of  $\mathbf{v}^{(1)}$  that is nonzero , and the values  $a_{i1}, a_{i2}, \dots, a_{in}$  are the entries in the  $i$ th row of  $\mathbf{A}$ .

- With this definition ,

$$\begin{aligned}\mathbf{x}^T \mathbf{v}^{(1)} &= \frac{1}{\lambda_1 v_i^{(1)}} [a_{i1}, a_{i2}, \dots, a_{in}] \left( v_1^{(1)}, v_2^{(1)}, \dots, v_n^{(1)} \right)^T \\ &= \frac{1}{\lambda_1 v_i^{(1)}} \sum_{j=1}^n a_{ij} v_j^{(1)}\end{aligned}$$

where the sum is the  $i$ th coordinate of the product  $A\mathbf{v}^{(1)}$ .

- Since  $A\mathbf{v}^{(1)} = \lambda_1 \mathbf{v}^{(1)}$ , we have

$$\sum_{j=1}^n a_{ij} v_j^{(1)} = \lambda_1 v_i^{(1)}$$

which implies that

$$\mathbf{x}^T \mathbf{v}^{(1)} = \frac{1}{\lambda_1 v_i^{(1)}} \left( \lambda_1 v_i^{(1)} \right) = 1.$$

- So  $\mathbf{x}$  satisfies the hypotheses of Theorem 4.12



- Moreover , the  $i$ th row of  $\mathbf{B} = \mathbf{A} - \lambda_1 \mathbf{v}^{(1)} \mathbf{x}^T$  consists entirely of zero entries.
- If  $\lambda \neq 0$  is an eigenvalue with associated eigenvector  $\mathbf{w}$  , the relation  $B\mathbf{w} = \lambda\mathbf{w}$  implies that the  $i$ th coordinate of  $\mathbf{w}$  must also be zero .
- Consequently the  $i$ th column of the matrix  $\mathbf{B}$  makes no contribution to the product  $\mathbf{B}\mathbf{w} = \lambda\mathbf{w}$  .
- Thus , the matrix  $B$  can be replaced by an  $(n - 1) \times (n - 1)$  matrix  $\mathbf{B}'$  has eigenvalues  $\lambda_2, \lambda_3, \dots, \lambda_n$  .

- If  $|\lambda_2| > |\lambda_3|$ , the Power method is reapplied to the matrix  $B'$  to determine this new dominant eigenvalue and an eigenvector,  $\mathbf{w}^{(2)'$ , associated with  $\lambda_2$ , with respect to the matrix  $B'$ .
- To find the associated eigenvector  $\mathbf{w}^{(2)}$  for the matrix  $B$ , insert a zero coordinate between the coordinates  $w_{i-1}^{(2)'} and  $w_i^{(2)'}$  of the  $(n-1)$ -dimensional vector  $\mathbf{w}^{(2)'}$  and then calculate  $\mathbf{v}^{(2)}$  by the use of Eq.(4).$

## ALGORITHM 9.4 Wielandt Deflation Technique

- To approximate the second most dominant eigenvalue and an associated eigenvector of the  $n \times n$  matrix  $\mathbf{A}$  given an approximation  $\lambda$  to the dominant eigenvalue, an approximation  $\mathbf{v}$  to a corresponding eigenvector, and a vector  $\mathbf{x} \in \mathbb{R}^{n-1}$  :
- **INPUT** dimension  $n$ ; matrix  $A$ ; approximate eigenvalue  $\lambda$  with eigenvector  $\mathbf{v} \in \mathbb{R}^n$ ; vector  $\mathbf{x} \in \mathbb{R}^{n-1}$ , tolerance  $TOL$ , maximum number of iterations  $N$ .
- **OUTPUT** approximate eigenvalue  $\mu$ ; approximate eigenvector  $\mathbf{u}$  or a message that the method fails.
- **Step 1** Let  $i$  be the smallest integer with  $1 \leq i \leq n$  and  $|v_i| = \max_{1 \leq j \leq n} |v_j|$ .

- **Step 2** If  $i \neq 1$  then
  - for  $k = 1, \dots, i - 1$ 
    - for  $j = 1, \dots, i - 1$
    - set

$$b_{kj} = a_{kj} - \frac{v_k}{v_i} a_{ij};$$

- **Step 3** If  $i \neq 1$  and  $i \neq n$  then
  - for  $k = i, \dots, n - 1$ 
    - for  $j = 1, \dots, i - 1$
    - set

$$b_{kj} = a_{k+1,j} - \frac{v_{k+1}}{v_i} a_{i,j};$$

$$b_{jk} = a_{j,k+1} - \frac{v_j}{v_i} a_{i,k+1};$$

- **Step 4** If  $i \neq n$  then
  - for  $k = i, \dots, n - 1$ 
    - for  $j = i, \dots, n - 1$
    - set  $b_{kj} = a_{k+1,j+1} - \frac{v_{k+1}}{v_i} a_{i,j+1};$

- **Step 5** Perform the power method on the  $(n - 1) \times (n - 1)$  matrix  $B' = (b_{kj})$  with  $\mathbf{x}$  as initial approximation.

- **Step 6** If the method fails, then OUTPUT ('Method fails'); STOP.  
Else let  $\mu$  be the approximate eigenvalue and  $\mathbf{w}' = (w'_1, w'_2, \dots, w'_{n-1})$  the approximate eigenvector.
- **Step 7** If  $i \neq 1$  then for  $k = 1, \dots, i - 1$  set  $w_k = w'_k$ .
- **Step 8** Set  $w_i = 0$ .
- **Step 9** If  $i \neq n$  then for  $k = i + 1, \dots, n$  set  $w_k = w'_{k-1}$ .
- **Step 10** For  $k = 1, \dots, n$  set

$$u_k = (\mu - \lambda)w_k + \left( \sum_{j=1}^n a_{ij} w_j \right) \frac{v_k}{v_i}.$$

(Compute the eigenvector using Eq. (4).)

- **Step 11** OUTPUT  $(\mu, \mathbf{u})$ ;  
(Procedure completed successfully.)  
STOP.

## 9.3 Orthogonalization Methods

- Possible methods include:
  - Householder transformations
  - Givens rotations
  - Gram-Schmidt orthogonalization

## 9.3.1 Householder transformation

- Householder transformation has form

$$\mathbf{H} = \mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$$

for nonzero vector  $\mathbf{v}$

- $\mathbf{H}$  is orthogonal and symmetric:

$$\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1}$$

- Notes:

$$\begin{aligned}\mathbf{H}\mathbf{H} &= \left(\mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) \left(\mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) \\ &= \mathbf{I} - 4 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} + 4 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \\ &= \mathbf{I}\end{aligned}$$

## 9.3.1 Householder transformation

- Householder transformation has form

$$\mathbf{H} = \mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$$

for nonzero vector  $\mathbf{v}$

- $\mathbf{H}$  is orthogonal and symmetric:

$$\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1}$$

- Notes:

$$\begin{aligned}\mathbf{H}\mathbf{H} &= \left(\mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) \left(\mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) \\ &= \mathbf{I} - 4 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} + 4 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \\ &= \mathbf{I}\end{aligned}$$



## 9.3.1 Householder transformation

- Householder transformation has form

$$\mathbf{H} = \mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$$

for nonzero vector  $\mathbf{v}$

- $\mathbf{H}$  is orthogonal and symmetric:

$$\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1}$$

- Notes:

$$\begin{aligned}\mathbf{H}\mathbf{H} &= \left(\mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) \left(\mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) \\ &= \mathbf{I} - 4 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} + 4 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}} \\ &= \mathbf{I}\end{aligned}$$

- Given vector  $\mathbf{a}$ , we want to choose  $\mathbf{v}$ , so that

$$\mathbf{H}\mathbf{a} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \mathbf{e}_1$$

- Substituting into formula for  $\mathbf{H}$ , we have

$$\alpha \mathbf{e}_1 = \mathbf{H}\mathbf{a} = \left( \mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{a} = \mathbf{a} - 2\mathbf{v} \frac{\mathbf{v}^T \mathbf{a}}{\mathbf{v}^T \mathbf{v}}$$

- then

$$\mathbf{v} = (\mathbf{a} - \alpha \mathbf{e}_1) \frac{\mathbf{v}^T \mathbf{v}}{2\mathbf{v}^T \mathbf{a}}$$

- Given vector  $\mathbf{a}$ , we want to choose  $\mathbf{v}$ , so that

$$\mathbf{H}\mathbf{a} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \mathbf{e}_1$$

- Substituting into formula for  $\mathbf{H}$ , we have

$$\alpha \mathbf{e}_1 = \mathbf{H}\mathbf{a} = \left( \mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{a} = \mathbf{a} - 2\mathbf{v} \frac{\mathbf{v}^T \mathbf{a}}{\mathbf{v}^T \mathbf{v}}$$

- then

$$\mathbf{v} = (\mathbf{a} - \alpha \mathbf{e}_1) \frac{\mathbf{v}^T \mathbf{v}}{2\mathbf{v}^T \mathbf{a}}$$

- Given vector  $\mathbf{a}$ , we want to choose  $\mathbf{v}$ , so that

$$\mathbf{H}\mathbf{a} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \mathbf{e}_1$$

- Substituting into formula for  $\mathbf{H}$ , we have

$$\alpha \mathbf{e}_1 = \mathbf{H}\mathbf{a} = \left( \mathbf{I} - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{a} = \mathbf{a} - 2\mathbf{v} \frac{\mathbf{v}^T \mathbf{a}}{\mathbf{v}^T \mathbf{v}}$$

- then

$$\mathbf{v} = (\mathbf{a} - \alpha \mathbf{e}_1) \frac{\mathbf{v}^T \mathbf{v}}{2\mathbf{v}^T \mathbf{a}}$$

- Let  $\mathbf{v} = \mathbf{a} - \alpha \mathbf{e}_1$
- To preserve the norm, we let

$$\alpha = \pm \|\mathbf{a}\|_2$$

i.e.,

$$\alpha = -\mathbf{sign}(a_1) \|\mathbf{a}\|_2$$

with sign chosen to avoid cancellation.

## Example: Householder Transformation

- If  $\mathbf{a} = [2 \ 1 \ 2]^T$ , then we take

$$\mathbf{v} = \mathbf{a} - \alpha \mathbf{e}_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix}$$

where  $\alpha = \pm \|\mathbf{a}\|_2 = \pm 3$ .

- Since  $a_1 > 0$ , we choose  $\alpha = -\|\mathbf{a}\|_2 = -3$ , so

$$\mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$$

- To confirm that transformation works,

$$\mathbf{H}\mathbf{a} = \mathbf{a} - 2 \frac{\mathbf{v}^T \mathbf{a}}{\mathbf{v}^T \mathbf{v}} \mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - 2 \frac{15}{30} \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$$

# Householder QR Factorization

- To compute QR factorization of  $\mathbf{A}$ , use Householder transformations to annihilate subdiagonal entries of each successive column
- Each Householder transformation is applied to entire matrix, but does not affect prior columns, so zeros are preserved
- In applying Householder transformation  $\mathbf{H}$  to arbitrary vector  $\mathbf{u}$ ,

$$\mathbf{H}\mathbf{u} = \left(\mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\mathbf{u} = \mathbf{u} - 2\frac{\mathbf{v}^T\mathbf{u}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

which is much cheaper than general matrix-vector multiplication and requires only vector  $\mathbf{v}$ , not full matrix  $\mathbf{H}$ .

# Householder QR Factorization

- To compute QR factorization of  $\mathbf{A}$ , use Householder transformations to annihilate subdiagonal entries of each successive column
- Each Householder transformation is applied to entire matrix, but does not affect prior columns, so zeros are preserved
- In applying Householder transformation  $\mathbf{H}$  to arbitrary vector  $\mathbf{u}$ ,

$$\mathbf{H}\mathbf{u} = \left(\mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\mathbf{u} = \mathbf{u} - 2\frac{\mathbf{v}^T\mathbf{u}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

which is much cheaper than general matrix-vector multiplication and requires only vector  $\mathbf{v}$ , not full matrix  $\mathbf{H}$ .



# Householder QR Factorization

- To compute QR factorization of  $\mathbf{A}$ , use Householder transformations to annihilate subdiagonal entries of each successive column
- Each Householder transformation is applied to entire matrix, but does not affect prior columns, so zeros are preserved
- In applying Householder transformation  $\mathbf{H}$  to arbitrary vector  $\mathbf{u}$ ,

$$\mathbf{H}\mathbf{u} = \left(\mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\mathbf{u} = \mathbf{u} - 2\frac{\mathbf{v}^T\mathbf{u}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

which is much cheaper than general matrix-vector multiplication and requires only vector  $\mathbf{v}$ , not full matrix  $\mathbf{H}$ .

- Process just described produces factorization

$$\mathbf{H}_n \cdots \mathbf{H}_1 \mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where  $\mathbf{R}$  is  $n \times n$  and upper triangular.

- If  $\mathbf{Q} = \mathbf{H}_1 \cdots \mathbf{H}_n$ , then  $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ .
- To preserve solution of linear least squares problem, right-hand side  $\mathbf{b}$  is transformed by same sequence of Householder transformations.
- Then solve triangular least squares problem

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \cong \mathbf{Q}^T \mathbf{b}$$

- Process just described produces factorization

$$\mathbf{H}_n \cdots \mathbf{H}_1 \mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where  $\mathbf{R}$  is  $n \times n$  and upper triangular.

- If  $\mathbf{Q} = \mathbf{H}_1 \cdots \mathbf{H}_n$ , then  $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ .
- To preserve solution of linear least squares problem, right-hand side  $\mathbf{b}$  is transformed by same sequence of Householder transformations.
- Then solve triangular least squares problem

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \cong \mathbf{Q}^T \mathbf{b}$$

- Process just described produces factorization

$$\mathbf{H}_n \cdots \mathbf{H}_1 \mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where  $\mathbf{R}$  is  $n \times n$  and upper triangular.

- If  $\mathbf{Q} = \mathbf{H}_1 \cdots \mathbf{H}_n$ , then  $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ .
- To preserve solution of linear least squares problem, right-hand side  $\mathbf{b}$  is transformed by same sequence of Householder transformations.
- Then solve triangular least squares problem

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \cong \mathbf{Q}^T \mathbf{b}$$

- Process just described produces factorization

$$\mathbf{H}_n \cdots \mathbf{H}_1 \mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

where  $\mathbf{R}$  is  $n \times n$  and upper triangular.

- If  $\mathbf{Q} = \mathbf{H}_1 \cdots \mathbf{H}_n$ , then  $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ .
- To preserve solution of linear least squares problem, right-hand side  $\mathbf{b}$  is transformed by same sequence of Householder transformations.
- Then solve triangular least squares problem

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \cong \mathbf{Q}^T \mathbf{b}$$

- For solving linear least squares problem, product  $Q$  of Householder transformations need not be formed explicitly.
- $R$  can be stored in upper triangle of array initially containing  $A$  .
- Householder vectors  $v$  can be stored in (now zero) lower triangular portion of  $A$  (almost)
- Householder transformations most easily applied in this form anyway

## Example: Householder QR Factorization

- For polynomial data-fitting example given previously, with

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0.0 & 0.0 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1.0 \\ 0.5 \\ 0.0 \\ 0.5 \\ 2.0 \end{bmatrix}$$

- Householder vector  $\mathbf{v}_1$  for annihilating subdiagonal entries of first column of  $\mathbf{A}$  is

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -2.236 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3.236 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- Applying resulting Householder transformation  $\mathbf{H}_1$  yields transformed matrix and right-hand side:

$$\mathbf{H}_1 \mathbf{A} = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & -0.191 & -0.405 \\ 0 & 0.309 & -0.655 \\ 0 & 0.809 & -0.405 \\ 0 & 1.309 & 0.345 \end{bmatrix}, \quad \mathbf{H}_1 \mathbf{b} = \begin{bmatrix} -1.789 \\ -0.362 \\ -0.862 \\ -0.362 \\ 1.138 \end{bmatrix}$$

- Householder vector  $\mathbf{v}_2$  for annihilating subdiagonal entries of second column of  $\mathbf{H}_1 \mathbf{A}$  is

$$\mathbf{v}_2 = \begin{bmatrix} 0 \\ -0.191 \\ 0.309 \\ 0.809 \\ 1.309 \end{bmatrix} - \begin{bmatrix} 0 \\ 1.581 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1.772 \\ 0.309 \\ 0.809 \\ 1.309 \end{bmatrix}$$



- Applying resulting Householder transformation  $\mathbf{H}_2$  yields

$$\mathbf{H}_2\mathbf{H}_1\mathbf{A} = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & 1.581 & 0 \\ 0 & 0 & -0.725 \\ 0 & 0 & -0.589 \\ 0 & 0 & 0.047 \end{bmatrix}, \quad \mathbf{H}_2\mathbf{H}_1\mathbf{b} = \begin{bmatrix} -1.789 \\ 0.632 \\ -1.035 \\ -0.816 \\ 0.404 \end{bmatrix}$$

- Householder vector  $\mathbf{v}_3$  for annihilating subdiagonal entries of third column of  $\mathbf{H}_2\mathbf{H}_1\mathbf{A}$  is

$$\mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ -0.725 \\ -0.589 \\ 0.047 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0.935 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1.660 \\ -0.589 \\ 0.047 \end{bmatrix}$$

- Applying resulting Householder transformation  $\mathbf{H}_3$  yields

$$\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A} = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & 1.581 & 0 \\ 0 & 0 & 0.935 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{b} = \begin{bmatrix} -1.789 \\ 0.632 \\ 1.336 \\ 0.026 \\ 0.337 \end{bmatrix}$$

- Now solve upper triangular system  $\mathbf{R}\mathbf{x} = \mathbf{c}_1$  by back-substitution to obtain  $\mathbf{x} = [0.086 \ 0.400 \ 1.429]^T$ .

# Givens Rotations

- Givens rotations introduce zeros one at a time
- Given vector  $[a_1 \ a_2]^T$ , choose scalars  $c$  and  $s$  so that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

with  $c^2 + s^2 = 1$ , or equivalently,  
 $\alpha = \sqrt{a_1^2 + a_2^2}$ .

- Previous equation can be rewritten

$$\begin{bmatrix} a_1 & a_2 \\ a_2 & -a_1 \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

- Gaussian elimination yields triangular system

$$\begin{bmatrix} a_1 & a_2 \\ 0 & -a_1 - a_2^2/a_1 \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} \alpha \\ -\alpha a_2/a_1 \end{bmatrix}$$

- Back-substitution then gives

$$s = \frac{\alpha a_2}{a_1^2 + a_2^2} \quad \text{and} \quad c = \frac{\alpha a_1}{a_1^2 + a_2^2}$$

- Finally,  $c^2 + s^2 = 1$ , or  $\alpha = \sqrt{a_1^2 + a_2^2}$ , implies

$$s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}} \quad \text{and} \quad c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}$$

# Givens QR Factorization

- More generally, to annihilate any desired component of vector in  $n$  dimensions, rotate target component with another component say  $(i, j)$ .
- For example, let  $n = 5, i = 4, j = 2$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -s & 0 & c & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ ca_{21} + sa_{41} & ca_{22} + sa_{42} & ca_{23} + sa_{43} & ca_{24} + sa_{44} & ca_{25} + sa_{45} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ -sa_{21} + ca_{41} & -sa_{22} + ca_{42} & -sa_{23} + ca_{43} & -sa_{24} + ca_{44} & -sa_{25} + ca_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \hat{a}_{21} & \hat{a}_{22} & \hat{a}_{23} & \hat{a}_{24} & \hat{a}_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ \hat{a}_{41} & 0 & \hat{a}_{43} & \hat{a}_{44} & \hat{a}_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

**Note that:** Let

$$c = \frac{a_{22}}{\sqrt{a_{22}^2 + a_{42}^2}}, s = \frac{a_{42}}{\sqrt{a_{22}^2 + a_{42}^2}}$$

- By systematically annihilating successive entries, we can reduce matrix to upper triangular form using sequence of Givens rotations.
- **Each rotation is orthogonal**, so **their product is orthogonal**, producing **QR** factorization.
- Straightforward implementation of Givens method requires about 50% more work than Householder method, and also requires more storage, since each rotation requires two numbers,  $c$  and  $s$ , to define it.
- These disadvantages can be overcome, but requires more complicated implementation.
- Givens can be advantageous for computing **QR** factorization when many entries of matrix are already zero, since those annihilations can then be skipped.

## Example: Givens QR Factorization

To solve least square problem

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \approx \begin{bmatrix} 1237 \\ 1941 \\ 2417 \\ 711 \\ 1177 \\ 475 \end{bmatrix} = \mathbf{b}$$



- First, to eliminate the entry in the position (4,1) of  $\mathbf{G}_1\mathbf{A}$ , since  $\sqrt{1^2 + (-1)^2} = \sqrt{2}$ , so  $c = 1/\sqrt{2}$ ,  $s = -1/\sqrt{2}$ , and the first Givens matrix is

$$\mathbf{G}_1 = \begin{bmatrix} 0.7071 & 0 & 0 & 0 & -0.7071 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0.7071 & 0 & 0 & 0 & 0.7071 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Applying this rotation to  $\mathbf{A}$  and  $\mathbf{b}$ , yields

$$\mathbf{G}_1\mathbf{A} = \begin{bmatrix} 1.4142 & 0 & -0.7071 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 0.7071 \\ 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{G}_1\mathbf{b} = \begin{bmatrix} 42 \\ 1941 \\ 2417 \\ 711 \\ 1707 \\ 475 \end{bmatrix}$$

- Second, to eliminate the entry in the position (4,1), since  $\sqrt{1.4142^2 + (-1)^2} = \sqrt{3}$ , so  $c = \sqrt{2}/\sqrt{3}$ ,  $s = -1/\sqrt{3}$ , and the second Givens matrix is

$$\mathbf{G}_2 = \begin{bmatrix} 0.8165 & 0 & 0 & -0.5774 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0.5774 & 0 & 0 & 0.8165 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Applying this rotation to  $\mathbf{G}_1\mathbf{A}$  and  $\mathbf{G}_1\mathbf{b}$ , yields

$$\mathbf{G}_2\mathbf{G}_1\mathbf{A} = \begin{bmatrix} 1.7321 & -0.5774 & -0.5774 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0.8165 & -0.4082 \\ 0 & 0 & 0.7071 \\ 0 & -1 & 1 \end{bmatrix}, \mathbf{G}_2\mathbf{G}_1\mathbf{b} = \begin{bmatrix} -376 \\ 1941 \\ 2417 \\ 605 \\ 1707 \\ 475 \end{bmatrix}$$

- Third working for the bottom of the other column of  $G_2 G_1 A$ , to eliminate the entry in the position (6,2),(4,2) and (6,3),(5,3),(4,3) with Givens rotation matrix.
- Finally yields

$$Q^T A = \begin{bmatrix} 1.7321 & -0.5774 & -0.5774 \\ 0 & 1.6330 & -0.8165 \\ 0 & 0 & 1.4142 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad Q^T b = \begin{bmatrix} -376 \\ 1200 \\ 2417 \\ 5.66 \\ -1.63 \\ -0.56 \end{bmatrix}$$

- We can now solve the upper triangular system by backward-substitution to obtain  $x = [1236 \quad 1943 \quad 2416]^T$

# Gram-Schmidt Orthogonalization

- Given vectors  $\mathbf{a}_1$  and  $\mathbf{a}_2$ , we seek orthonormal vectors  $\mathbf{q}_1$  and  $\mathbf{q}_2$  having same span.
- We first normalize  $\mathbf{a}_1$  to obtain

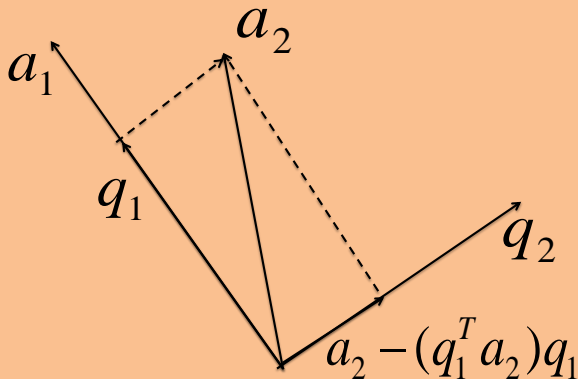
$$\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|_2}.$$

- Next we want to subtract from  $\mathbf{a}_2$  its component in  $\mathbf{q}_1$ , as shown in diagram.

$$\mathbf{q}_2 = \mathbf{a}_2 - (\mathbf{q}_1^T \mathbf{a}_2) \mathbf{q}_1$$

$$\mathbf{q}_2 = \frac{\mathbf{q}_2}{\|\mathbf{q}_2\|}.$$

# Gram-Schmidt Orthogonalization



### THEOREM: The Orthogonal Decomposition Theorem — 正交分解定理

Let  $W$  be a subspace of  $\mathbb{R}^n$ . Then each  $\mathbf{y}$  in  $\mathbb{R}^n$  can be written uniquely in the form

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{z} \quad (6)$$

where  $\hat{\mathbf{y}}$  is in  $W$  and  $\mathbf{z}$  is in  $W^\perp$ .

## 如何求 $\hat{\mathbf{y}}$ 和 $\mathbf{z}$ ?

事实上, 若  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$  是子空间  $W$  的正交基, 则

$$\hat{\mathbf{y}} = \frac{\mathbf{y} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 + \frac{\mathbf{y} \cdot \mathbf{u}_2}{\mathbf{u}_2 \cdot \mathbf{u}_2} \mathbf{u}_2 + \dots + \frac{\mathbf{y} \cdot \mathbf{u}_p}{\mathbf{u}_p \cdot \mathbf{u}_p} \mathbf{u}_p \quad (7)$$

进而, 可以很容易地得出:

$$\mathbf{z} = \mathbf{y} - \hat{\mathbf{y}}.$$

## 定理证明:

- 若  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$  是  $\mathbb{R}^n$  空间的子空间  $W$  的一个正交基, 由于  $\hat{\mathbf{y}} \in W$ , 则  $\hat{\mathbf{y}} \in W$  可以写成如下关于正交基  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$  的线性组合:

$$\hat{\mathbf{y}} = \frac{\mathbf{y} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 + \frac{\mathbf{y} \cdot \mathbf{u}_2}{\mathbf{u}_2 \cdot \mathbf{u}_2} \mathbf{u}_2 + \dots + \frac{\mathbf{y} \cdot \mathbf{u}_p}{\mathbf{u}_p \cdot \mathbf{u}_p} \mathbf{u}_p$$

- 令:  $\mathbf{z} = \mathbf{y} - \hat{\mathbf{y}}$ .
- 则可以证明:  $\mathbf{z} \in W^\perp$ .

$$\begin{aligned} \mathbf{z} \cdot \mathbf{u}_1 &= (\mathbf{y} - \hat{\mathbf{y}}) \cdot \mathbf{u}_1 \\ &= \mathbf{y} \cdot \mathbf{u}_1 - \frac{\mathbf{y} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 \cdot \mathbf{u}_1 - 0 - \dots - 0 \\ &= \mathbf{y} \cdot \mathbf{u}_1 - \mathbf{y} \cdot \mathbf{u}_1 = 0 \end{aligned}$$



- 类似地, 可以证明  $\mathbf{z}$  与  $W$  中每一个基向量  $\mathbf{u}_i, i = 1, 2, \dots, p$  都正交, 而

$$W = \text{Span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$$

所以,  $\mathbf{z}$  与  $W$  中任意向量都正交, 即  $\mathbf{z} \in W^\perp$ .

- 再证正交分解的唯一性: 设  $\mathbf{y} = \hat{\mathbf{y}}_1 + \mathbf{z}_1$  ( $\mathbf{y}_1 \in W, \mathbf{z}_1 \in W^\perp$ ) 是另一个正交分解. 则

$$\hat{\mathbf{y}} + \mathbf{z} = \hat{\mathbf{y}}_1 + \mathbf{z}_1$$

或写成

$$\hat{\mathbf{y}} - \hat{\mathbf{y}}_1 = \mathbf{z}_1 - \mathbf{z}.$$

- 注意到上式左端向量  $(\hat{\mathbf{y}} - \hat{\mathbf{y}}_1) \in W$ , 而右端向量  $(\mathbf{z}_1 - \mathbf{z}) \in W^\perp$ . 此类情况当且仅当两端同时为零向量时方可成立. 于是  $\hat{\mathbf{y}} = \hat{\mathbf{y}}_1, \mathbf{z}_1 = \mathbf{z}$ , 即正交分解是唯一的. ■■■

# Properties of Orthogonal Projections

- If  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$  is any orthogonal basis of  $W$ , and  $\mathbf{y} \in W$ , then

$$\hat{\mathbf{y}} = \text{Proj}_W \mathbf{y} = \frac{\mathbf{y} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 + \frac{\mathbf{y} \cdot \mathbf{u}_2}{\mathbf{u}_2 \cdot \mathbf{u}_2} \mathbf{u}_2 + \dots + \frac{\mathbf{y} \cdot \mathbf{u}_p}{\mathbf{u}_p \cdot \mathbf{u}_p} \mathbf{u}_p$$

- **THEOREM—The Best Approximation Theorem(最优逼近定理)**

Let  $W$  be a subspace of  $\mathbb{R}^n$ ,  $\mathbf{y}$  any vector in  $\mathbb{R}^n$ , and  $\hat{\mathbf{y}}$  the any orthogonal projection of  $\mathbf{y}$  onto  $W$ . Then  $\hat{\mathbf{y}}$  is the closest point in  $W$  to  $\mathbf{y}$ . In the sense that

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \leq \|\mathbf{y} - \mathbf{v}\|$$

for all  $\mathbf{v} \in W$  distinct from  $\hat{\mathbf{y}}$ .

## Proof of Theorem:

- Suppose  $\mathbf{v} \in W$ , then  $\hat{\mathbf{y}} - \mathbf{v}$  also in  $W$ , by the orthogonal decomposition theorem,  $\mathbf{y} - \hat{\mathbf{y}}$  is orthogonal to  $W$ , that is  $\mathbf{y} - \hat{\mathbf{y}}$  is orthogonal to  $\hat{\mathbf{y}} - \mathbf{v}$ .
- Since  $\mathbf{y} - \mathbf{v} = (\mathbf{y} - \hat{\mathbf{y}}) + (\hat{\mathbf{y}} - \mathbf{v})$
- By the Pythagorean Theorem, gives

$$\|\mathbf{y} - \mathbf{v}\|^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 + \|\hat{\mathbf{y}} - \mathbf{v}\|^2.$$

- If  $\hat{\mathbf{y}} \neq \mathbf{v}$ , then we have  $\|\hat{\mathbf{y}} - \mathbf{v}\|^2 > 0$ , so the inequality

$$\|\mathbf{y} - \hat{\mathbf{y}}\| \leq \|\mathbf{y} - \mathbf{v}\|$$

holds immediately.

## THEOREM:

If  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$  is an orthonormal basis for a subspace  $W$  of  $\mathbb{R}^n$ , then

$$\text{proj}_W \mathbf{y} = (\mathbf{y} \cdot \mathbf{u}_1)\mathbf{u}_1 + (\mathbf{y} \cdot \mathbf{u}_2)\mathbf{u}_2 + \dots + (\mathbf{y} \cdot \mathbf{u}_p)\mathbf{u}_p$$

If  $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_p]$ , then

$$\text{proj}_W \mathbf{y} = \mathbf{U}^T \mathbf{U} \mathbf{y}, \quad \forall \mathbf{y} \in \mathbb{R}^n$$

证明: 首先由正交投影定理, 可得

$$\text{proj}_W \mathbf{y} = \frac{\mathbf{y} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1} \mathbf{u}_1 + \frac{\mathbf{y} \cdot \mathbf{u}_2}{\mathbf{u}_2 \cdot \mathbf{u}_2} \mathbf{u}_2 + \dots + \frac{\mathbf{y} \cdot \mathbf{u}_p}{\mathbf{u}_p \cdot \mathbf{u}_p} \mathbf{u}_p$$

进一步地, 由于 $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$ 为子空间 $W$ 的标准正交基, 故有

$$\mathbf{u}_1 \cdot \mathbf{u}_1 = \mathbf{u}_2 \cdot \mathbf{u}_2 = \dots = \mathbf{u}_p \cdot \mathbf{u}_p = 1.$$

从而有

$$\text{proj}_W \mathbf{y} = (\mathbf{y} \cdot \mathbf{u}_1) \mathbf{u}_1 + (\mathbf{y} \cdot \mathbf{u}_2) \mathbf{u}_2 + \dots + (\mathbf{y} \cdot \mathbf{u}_p) \mathbf{u}_p$$

成立. 又由

$$\mathbf{y} \cdot \mathbf{u}_1 = \mathbf{u}_1^T \mathbf{y}, \mathbf{y} \cdot \mathbf{u}_2 = \mathbf{u}_2^T \mathbf{y}, \dots, \mathbf{y} \cdot \mathbf{u}_p = \mathbf{u}_p^T \mathbf{y}$$

所以

$$\text{proj}_W \mathbf{y} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_p] \begin{bmatrix} \mathbf{u}_1^T \mathbf{y} \\ \mathbf{u}_2^T \mathbf{y} \\ \vdots \\ \mathbf{u}_p^T \mathbf{y} \end{bmatrix} = \mathbf{U}(\mathbf{U}^T \mathbf{y}). \blacksquare \blacksquare$$

# The Gram-Schmidt process 格莱姆-施密特过程

若  $W$  是  $\mathbb{R}^n$  的子空间,  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$  是  $W$  的基, 记

$$\mathbf{v}_1 = \mathbf{x}_1$$

$$\mathbf{v}_2 = \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1} \mathbf{v}_1$$

$$\mathbf{v}_3 = \mathbf{x}_3 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1} \mathbf{v}_1 - \frac{\mathbf{x}_3 \cdot \mathbf{v}_2}{\mathbf{v}_2 \cdot \mathbf{v}_2} \mathbf{v}_2$$

$$\vdots$$

$$\mathbf{v}_p = \mathbf{x}_p - \frac{\mathbf{x}_p \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1} \mathbf{v}_1 - \frac{\mathbf{x}_p \cdot \mathbf{v}_2}{\mathbf{v}_2 \cdot \mathbf{v}_2} \mathbf{v}_2 - \dots - \frac{\mathbf{x}_p \cdot \mathbf{v}_{p-1}}{\mathbf{v}_{p-1} \cdot \mathbf{v}_{p-1}} \mathbf{v}_{p-1}.$$

则  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$  是  $W$  的正交基., 且

$$\text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\} = \text{Span}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}.$$

**证明:** 易证按Gram-Schmidt正交化过程产生的 $p$  个向量

$$\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p$$

两两正交. 先证

$$\mathbf{v}_2 \cdot \mathbf{v}_1 = \left( \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1} \mathbf{v}_1 \right) \cdot \mathbf{v}_1 = 0.$$

类似地, 可证对 $i = 1, 2, \cdots, p-1$ , 有

$$\mathbf{v}_i \cdot \mathbf{v}_j = 0, j = i+1, \cdots, p.$$

即 $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p\}$  两两正交, 故该向量组为线性无关组.  
由于其无关向量的个数为 $p$  个, 故该向量组为子空间 $W$  的一个正交基.

由Gram-Schmidt 向量的正交化过程可知:

$$\text{Span}\{\mathbf{x}_1, \cdots, \mathbf{x}_k\} = \text{Span}\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}, k = 1, 2, \cdots, p.$$

即向量组 $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k\}$  中任一向量都可以由向量组 $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$ 线性表出, 反之亦然. 因此

$$\text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p\} = \text{Span}\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p\}.$$

- Let  $\mathbf{x}_1 = \begin{bmatrix} 3 \\ 6 \\ 0 \end{bmatrix}$ ,  $\mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$ ,  $W = \text{Span}\{\mathbf{x}_1, \mathbf{x}_2\}$ ,  
construct an orthogonal basis  $\{\mathbf{v}_1, \mathbf{v}_2\}$  for  $W$

- Let  $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ ,  $\mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ ,  $\mathbf{x}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$ . Then

$W = \text{Span}\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$  is a subspace of  $\mathbb{R}^4$ .  
Construct an orthogonal basis for  $W$ .



- If  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$  is an orthogonal basis for  $W$ , then let

$$\mathbf{u}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}, \mathbf{u}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|}, \dots, \mathbf{u}_p = \frac{\mathbf{v}_p}{\|\mathbf{v}_p\|}$$

Then  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$  is an orthonormal basis for  $W$ .

- If an orthogonal basis  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$  and  $\text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\} = \text{Span}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$ . Then is orthonormal sets  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$  also forms an basis for  $W$ , and

$$\begin{aligned}\text{Span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\} &= \text{Span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\} \\ &= \text{Span}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}.\end{aligned}$$

## THEOREM 12: The QR Factorization

If  $\mathbf{A}$  is an  $m \times n$  matrix with **linearly independent columns**, then  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{QR}$ , where  $\mathbf{Q}$  is an  $m \times n$  matrix whose columns form an **orthonormal basis** for  $\text{Col}\mathbf{A}$  and  $\mathbf{R}$  is an  $n \times n$  **upper triangular invertible matrix** with **positive entries on its diagonal**.

证明: 记  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  为  $\mathbf{A}$  的  $n$  个线性无关列向量,  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$  为其按照 Gram-Schmidt 方法构造的正交向量组, 而  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$  为由线性无关的正交向量组  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$  标准化后形成的标准正交基.

则有

$$\text{Span}\{\mathbf{x}_1, \dots, \mathbf{x}_k\} = \text{Span}\{\mathbf{u}_1, \dots, \mathbf{u}_k\}, k = 1, 2, \dots, p.$$

即

$$\mathbf{x}_1 = r_{11}\mathbf{u}_1 + 0 \cdot \mathbf{u}_2 + \dots + 0 \cdot \mathbf{u}_n$$

$$\mathbf{x}_2 = r_{12}\mathbf{u}_1 + r_{22}\mathbf{u}_2 + \dots + 0 \cdot \mathbf{u}_n$$

...

$$\mathbf{x}_n = r_{1n}\mathbf{u}_1 + r_{2n}\mathbf{u}_2 + \dots + r_{nn}\mathbf{u}_n.$$

其中,  $r_{i,j}, i, j = 1, 2, \dots, n$  为组合系数, 且易证  $r_{kk}, k = 1, 2, \dots, n$  均为非负常数.

$$[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

记

$$\mathbf{Q} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n], \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

则  $m \times n$  矩阵  $\mathbf{A}$  可以分解为一个  $m \times n$  阶标准正交矩阵  $\mathbf{Q}$  和一个  $n \times n$  阶上三角矩阵  $\mathbf{R}$  的乘积的形式. 即

$$\mathbf{A} = \mathbf{QR}.$$

# Steps or Algorithm for computing $QR$ factorization for an $m \times n$ matrix $\mathbf{A}$

- Using Gram-Schmidt process, find its corresponding orthogonal set.
- Normalize the orthogonal set, and form  $\mathbf{Q}$
- Find  $\mathbf{R} = \mathbf{Q}^T \mathbf{A}$ .

Example: Find a **QR** factorization of **A**

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

# Classical Gram-Schmidt procedure:

For any number of vectors  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ , orthogonalizing each successive vector against all preceding ones, giving

## Gram-Schmidt procedure

```
for  $k = 1$  to  $n$   
     $\mathbf{q}_k = \mathbf{a}_k$   
    for  $j = 1$  to  $k - 1$   
         $r_{jk} = \mathbf{q}_j^T \mathbf{a}_k, \quad \mathbf{q}_k = \mathbf{q}_k - r_{jk} \mathbf{q}_j$   
    end  
     $r_{kk} = \|\mathbf{q}_k\|_2, \quad \mathbf{q}_k = \mathbf{q}_k / r_{kk}$   
end
```

Resulting  $\mathbf{q}_k$  and  $r_{jk}$  form reduced **QR** factorization of **A**.

# Modified Gram-Schmidt

- Classical Gram-Schmidt procedure often suffers loss of orthogonality in finite-precision.
- Also, separate storage is required for  $\mathbf{A}$ ,  $\mathbf{Q}$ , and  $\mathbf{R}$ , since original  $\mathbf{a}_k$  are needed in inner loop, so  $\mathbf{q}_k$  cannot overwrite columns of  $\mathbf{A}$ .
- Both deficiencies are improved by modified Gram-Schmidt procedure, with each vector orthogonalized in turn against all subsequent vectors, so  $\mathbf{q}_k$  can overwrite  $\mathbf{a}_k$ .



# Modified Gram-Schmidt QR Factorization Algorithm

## Modified Gram-Schmidt QR Factorization Algorithm

for  $k = 1$  to  $n$

$$r_{kk} = \|\mathbf{a}_k\|_2, \quad \mathbf{q}_k = \mathbf{a}_k / r_{kk}$$

for  $j = k + 1$  to  $n$

$$r_{kj} = \mathbf{q}_k^T \mathbf{a}_j, \quad \mathbf{a}_j = \mathbf{a}_j - r_{kj} \mathbf{q}_k$$

end

end

# Rank Deficiency

- If  $\text{rank}(\mathbf{A}) < n$ , then **QR** factorization still exists, but yields singular upper triangular factor **R**, and multiple vectors  $\mathbf{x}$  give minimum residual norm.
- Common practice selects minimum residual solution  $\mathbf{x}$  having smallest norm.
- Can be computed by **QR** factorization with column pivoting or by **singular value decomposition (SVD)**?
- Rank of matrix is often not clear cut in practice, so relative tolerance is used to determine rank

## Example: Near Rank Deficiency

- Consider  $3 \times 2$  matrix

$$\mathbf{A} = \begin{bmatrix} 0.641 & 0.242 \\ 0.321 & 0.121 \\ 0.962 & 0.363 \end{bmatrix}$$

- Computing  $\mathbf{QR}$  factorization,

$$\mathbf{R} = \begin{bmatrix} 1.1997 & 0.4527 \\ 0 & 0.0002 \end{bmatrix}$$

$\mathbf{R}$  is extremely close to singular (exactly singular to 3-digit accuracy of problem statement).

## Example: Near Rank Deficiency

- Consider  $3 \times 2$  matrix

$$\mathbf{A} = \begin{bmatrix} 0.641 & 0.242 \\ 0.321 & 0.121 \\ 0.962 & 0.363 \end{bmatrix}$$

- Computing **QR** factorization,

$$\mathbf{R} = \begin{bmatrix} 1.1997 & 0.4527 \\ 0 & 0.0002 \end{bmatrix}$$

**R** is extremely close to singular (exactly singular to 3-digit accuracy of problem statement).

- If  $\mathbf{R}$  is used to solve linear least squares problem, result is highly sensitive to perturbations in right-hand side.
- For practical purposes,  $\text{rank}(\mathbf{A}) = 1$  rather than 2, because columns are nearly linearly dependent.

# QR with Column Pivoting

- Instead of processing columns in natural order, select for reduction at each stage column of remaining unreduced submatrix having maximum Euclidean norm.
- If  $\text{rank}(\mathbf{A}) = k < n$ , then after  $k$  steps, norms of remaining unreduced columns will be zero (or “negligible” in finite-precision arithmetic) below row  $k$ .
- Yields orthogonal factorization of form

$$\mathbf{Q}^T \mathbf{A} \mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{S} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where  $\mathbf{R}$  is  $k \times k$ , upper triangular, and nonsingular, and permutation matrix  $\mathbf{P}$  performs column interchanges.

# QR with Column Pivoting

- Instead of processing columns in natural order, select for reduction at each stage column of remaining unreduced submatrix having maximum Euclidean norm.
- If  $\text{rank}(\mathbf{A}) = k < n$ , then after  $k$  steps, norms of remaining unreduced columns will be zero (or “negligible” in finite-precision arithmetic) below row  $k$ .
- Yields orthogonal factorization of form

$$\mathbf{Q}^T \mathbf{A} \mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{S} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where  $\mathbf{R}$  is  $k \times k$ , upper triangular, and nonsingular, and permutation matrix  $\mathbf{P}$  performs column interchanges.

# QR with Column Pivoting

- Instead of processing columns in natural order, select for reduction at each stage column of remaining unreduced submatrix having maximum Euclidean norm.
- If  $\text{rank}(\mathbf{A}) = k < n$ , then after  $k$  steps, norms of remaining unreduced columns will be zero (or “negligible” in finite-precision arithmetic) below row  $k$ .
- Yields orthogonal factorization of form

$$\mathbf{Q}^T \mathbf{A} \mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{S} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where  $\mathbf{R}$  is  $k \times k$ , upper triangular, and nonsingular, and permutation matrix  $\mathbf{P}$  performs column interchanges.



Basic solution to least squares problem  $\mathbf{Ax} \cong \mathbf{b}$  can now be computed by solving triangular system  $\mathbf{Rz} = \mathbf{c}_1$ , where  $\mathbf{c}_1$  contains first  $k$  components of  $\mathbf{Q}^T \mathbf{b}$ , and then taking

$$\mathbf{x} = \mathbf{P} \begin{bmatrix} \mathbf{z} \\ \mathbf{0} \end{bmatrix}$$

- Minimum-norm solution can be computed, if desired, at expense of additional processing to annihilate  $S$ .
- $\text{rank}(\mathbf{A})$  is usually unknown, so rank is determined by monitoring norms of remaining unreduced columns and terminating factorization when maximum value falls below chosen tolerance .

## 9.4 Singular Value Decomposition

- **Singular Value Decomposition (SVD)** of  $m \times n$  matrix  $\mathbf{A}$  has form

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

- Where
  - $\mathbf{U}$  is  $m \times m$  orthogonal matrix,  $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ .
  - $\mathbf{V}$  is  $n \times n$  orthogonal matrix,  $\mathbf{V}\mathbf{V}^T = \mathbf{I}$
  - $\mathbf{\Sigma} = (\sigma_{ij})_{m \times n}$  is  $m \times n$  diagonal matrix, with

$$\sigma_{ij} = \begin{cases} 0, & i \neq j; \\ \sigma_i > 0, & i = j. \end{cases}$$

- Diagonal entries  $\sigma_i$ , called **Singular Values** of  $\mathbf{A}$ , are usually ordered so that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$$

思考：为什么要排序？如何实现排序？

- Columns  $\mathbf{u}_i$  of  $\mathbf{U}$  are called **left singular vectors**.
- Columns  $\mathbf{v}_i$  of  $\mathbf{V}$  are called **right singular vectors**.

- If  $\mathbf{A}$  is  $m \times n$  with  $\text{rank}(\mathbf{A}) = n$ , then

$$\begin{aligned}\mathbf{A} &= \mathbf{U}\Sigma\mathbf{V}^T \\ &= [\mathbf{U}_1 \ \mathbf{U}_2] \begin{bmatrix} \Sigma_1 \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T \\ &= \mathbf{U}_1 \Sigma_1 \mathbf{V}^T\end{aligned}$$

where  $\mathbf{U}_1$  is  $m \times n$  matrix with **orthonormal** columns, and  $\Sigma_1$  is an  $n \times n$  **diagonal** matrix and nonsingular,  $\mathbf{V}$  is an  $n \times n$  **orthonormal** matrix.

- $\mathbf{A} = \mathbf{U}_1 \Sigma_1 \mathbf{V}^T$  is called reduced **SVD** of  $\mathbf{A}$ .

- For least square problem  $\mathbf{Ax} = \mathbf{b}$ , it can be solved with normal equations with the form

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$$

- If  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$  has a SVD form, then

$$(\mathbf{U}\Sigma\mathbf{V}^T)^T \mathbf{U}\Sigma\mathbf{V}^T \mathbf{x} = (\mathbf{U}\Sigma\mathbf{V}^T)^T \mathbf{b}$$

$$\mathbf{V}\Sigma^T \mathbf{U}^T \mathbf{U}\Sigma\mathbf{V}^T \mathbf{x} = \mathbf{V}\Sigma^T \mathbf{U}^T \mathbf{b}$$

$$\mathbf{V}\Sigma^T \Sigma \mathbf{V}^T \mathbf{x} = \mathbf{V}\Sigma^T \mathbf{U}^T \mathbf{b}$$

$$\mathbf{V}^T \mathbf{V}\Sigma^T \Sigma \mathbf{V}^T \mathbf{x} = \mathbf{V}^T \mathbf{V}\Sigma^T \mathbf{U}^T \mathbf{b}$$

$$\Sigma^T \Sigma \mathbf{V}^T \mathbf{x} = \Sigma^T \mathbf{U}^T \mathbf{b}$$

$$\Sigma \mathbf{V}^T \mathbf{x} = \mathbf{U}^T \mathbf{b}$$

$$\mathbf{x} = \mathbf{V}\Sigma^{-1} \mathbf{U}^T \mathbf{b}.$$

- More generally, for  $\mathbf{A}$  of any shape or rank, the least squares solution to  $\mathbf{A}\mathbf{x} \cong \mathbf{b}$  of minimum norm is given by

$$\mathbf{x} = \sum_{\sigma_i \neq 0} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

- For ill-conditioned or rank deficient problems, “small” singular values can be omitted from summation to stabilize solution.
- Euclidean matrix norm:  $\|\mathbf{A}\|_2 = \sigma_{\max}$ .
- Euclidean condition number of matrix :  
 $\text{cond}(\mathbf{A}) = \frac{\sigma_{\max}}{\sigma_{\min}}$ .
- Rank of matrix: number of nonzero singular values.

# Pseudoinverse (广义逆)

- Define pseudoinverse of scalar  $\sigma$  to be  $1/\sigma$  if  $\sigma \neq 0$ , zero otherwise.
- Define **pseudoinverse of (possibly rectangular) diagonal matrix** by transposing and taking scalar pseudoinverse of each entry.
- Then pseudoinverse of general real  $m \times n$  matrix  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  is given by

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T.$$



- Pseudoinverse always exists whether or not matrix is square or has full rank
- If  $\mathbf{A}$  is square and nonsingular, then  $\mathbf{A}^+ = \mathbf{A}^{-1}$ .
- In all cases, minimum-norm solution to  $\mathbf{Ax} \cong \mathbf{b}$  is given by

$$\mathbf{x} = \mathbf{A}^+\mathbf{b}.$$

# Orthogonal Bases

- SVD of matrix,  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ , provides orthogonal bases for subspaces relevant to  $\mathbf{A}$ .
- Columns of  $\mathbf{U}$  corresponding to nonzero singular values form orthonormal basis for  $\text{span}(\mathbf{A})$ .
- Remaining columns of  $\mathbf{U}$  form orthonormal basis for orthogonal complement  $\text{span}(\mathbf{A})^\perp$ .
- Columns of  $\mathbf{V}$  corresponding to zero singular values form orthonormal basis for null space of  $\mathbf{A}$ .
- Remaining columns of  $\mathbf{V}$  form orthonormal basis for orthogonal complement of null space of  $\mathbf{A}$ .

# Lower-Rank Matrix Approximation

- Another way to write **SVD** is

$$\mathbf{A} = \mathbf{U} \sum \mathbf{V}^T = \sigma_1 \mathbf{E}_1 + \sigma_2 \mathbf{E}_2 + \cdots + \sigma_n \mathbf{E}_n$$

with  $\mathbf{E}_i = \mathbf{u}_i \mathbf{v}_i^T$ .

- $\mathbf{E}_i$  has rank 1 and can be stored using only  $m + n$  storage locations.
- Product  $\mathbf{E}_i \mathbf{x}$  can be computed using only  $m + n$  multiplications.

- Condensed approximation to  $\mathbf{A}$  is obtained by omitting from summation terms corresponding to small singular values.
- Approximation using  $k$  largest singular values is closest matrix of rank  $k$  to  $\mathbf{A}$ .
- Approximation is useful in image processing(图像处理), data compression(数据压缩), information retrieval(信息检索), cryptography(加密), etc.

# Total Least Squares

- Ordinary least squares is applicable when right-hand side  $\mathbf{b}$  is subject to random error but matrix  $\mathbf{A}$  is known accurately.
- When all data, including  $\mathbf{A}$ , are subject to error, then total least squares is more appropriate.
- Total least squares minimizes orthogonal distances, rather than vertical distances, between model and data.
- Total least squares solution can be computed from **SVD** of  $[\mathbf{A}, \mathbf{b}]$ .



