Computational methods and numerical algorithms: Lecture 7

Wangtao Lu
Zhejiang University

October 27, 2020

Contents

Wangtao Lu
Zhejiang University

This chapter, we will discuss numerical algorithms for computing eigenvalues and singular values of a matrix $A \in \mathbb{R}^{m \times n}$.

Eigenvalues and eigenvectors

Wangtao Lu
Zhejiang University

Let $A \in \mathbb{R}^{m\times m}$ be a given square matrix. If $Av = \lambda v$ for a **nonzero** $v \in \mathbb{R}^m$ and a number $\lambda \in \mathbb{C}$, then we call λ an **eigenvalue** of A and the vector v is the assoicated **eigenvector**.
Clearly, all m eigenvalues of A are roots of the following degree m polynomial

$$P(\lambda) = \det(A - \lambda I_m) = 0.$$

However, we cannot use the previous root-finding algorithms to find the eigenvalues based on the following two reasons:

- Evaluating $P(\lambda)$ is too expensive.
- Root-finding algorithm is unstable even for trivial cases.

Revisit Wilkinson Polynomial

Consider the following diagonal matrix:

$$A = \mathrm{diag}(1, 2, \ldots, 20).$$

What is $P(\lambda)$?
Why finding the roots of $P(\lambda)$ is so difficulty?

Special case: 2×2 matrices!

For 2×2 matrices,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Characteristic Polynomial is

$$P(\lambda) = \lambda^2 - \mathrm{tr}(A)\lambda + \det(A) = 0.$$

Eigenvalues:

$$\lambda_{1,2} = \frac{\mathrm{tr}(A) \pm \sqrt{\mathrm{tr}^2(A) - 4\det(A)}}{2}.$$

Power iteration

Wangtao Lu
Zhejiang University

Suppose A has a **dominant eigenvalue** λ, i.e., the magnitude of eigenvalue λ is greater than all other eigenvalues of A. The assoicated eigenvector v is the so-called **dominant eigenvector**.
We will see that repeatedly multiplying almost any random vector by A yields a squence of vectors that converges to the dominant eigenvector. This is the key step of **Power iteration**.

Example 1. Find the dominant eigenvector of

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

by the Power iteration.

Example 1. Find the dominant eigenvector of

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

by the Power iteration.
How to make the Power iteration really converges?

Normalization

To make sure that Power iterations really converge to a vector with finite numbers, we need to normalize the resulting vector after each multiplication.

Wangtao Lu
Zhejiang University

Rayleigh quotient

How to find the dominant eigenvalue once approximate dominant eigenvector is available?
Suppose x is an approximate eigenvector and λ is unknown. We need to solve

$$x\lambda = Ax,$$

for λ. How to find λ?

Wangtao Lu
Zhejiang University

Rayleigh quotient

How to find the dominant eigenvalue once approximate dominant eigenvector is available?

Suppose x is an approximate eigenvector and λ is unknown. We need to solve

$$x\lambda = Ax,$$

for λ. How to find λ?

We make use of method of normal equations. Specifically, normal equations become

$$x^T x\lambda = x^T Ax,$$

so that

$$\lambda = \frac{x^T Ax}{x^T x}.$$

Wangtao Lu
Zhejiang University

Power iteration

Wangtao Lu
Zhejiang University

Normalization plus Rayleigh Quotient constitute the method of Power iteration.

Power iteration

Given initial vector x_0

for $j = 1, 2, 3, \ldots$

$u_{j-1} = x_{j-1}/||x_{j-1}||_2$ (Normalization)

$x_j = Au_{j-1}$

$\lambda_j = u_{j-1}^T Au_{j-1}$ (Rayleigh Quotient)

end

$u_j = x_j/||x_j||_2.$

Power iteration

Normalization plus Rayleigh Quotient constitute the method of Power iteration.

Power iteration

Given initial vector x_0

for $j = 1, 2, 3, \ldots$

$u_{j-1} = x_{j-1}/||x_{j-1}||_2$ (Normalization)

$x_j = Au_{j-1}$

$\lambda_j = u_{j-1}^T x_j$ (Rayleigh Quotient)

end

$u_j = x_j/||x_j||_2$.

Convergence of Power iteration

Theorem. Let A be an $m \times m$ matrix with real eigenvalues $\lambda_1, \dots, \lambda_m$ satisfying $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_m|$. Assume that the eigenvectors of A span $\mathbb{R}^m$. For almost every initial vector, Power Iteration converges linearly to an eigenvector associated to λ_1 with convergence rate constant $S = |\lambda_2/\lambda_1|$.

Proof. Let $v_1, \dots, v_m$ be the eigenvectors that form a basis of $\mathbb{R}^m$. Suppose the initial vector $x_0 = c_1 v_1 + \ldots + c_m v_m$. Here, **we assume that** $c_1, c_2 \neq 0$. Applying Power Iteration yields

$$\begin{aligned} A x_0 &= c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + \ldots + c_m \lambda_m v_m \\ A^2 x_0 &= c_1 \lambda_1^2 v_1 + c_2 \lambda_2^2 v_2 + \ldots + c_m \lambda_m^2 v_m \\ A^3 x_0 &= c_1 \lambda_1^3 v_1 + c_2 \lambda_2^3 v_2 + \ldots + c_m \lambda_m^3 v_m \\ &\vdots \end{aligned}$$

Then, we see that at k-th step,

$$\frac{A^k x_0}{\lambda_1^k} = c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \ldots + c_m \left(\frac{\lambda_m}{\lambda_1}\right)^k v_m \to c_1 v_1, \quad k \to \infty.$$

The convergence rate depends exactly on the magnitude of λ_2/λ_1.

Inverse Power Iteration

Inverse Power Iteration finds an eigenvalue with the smallest magnitude.

Lemma. Let the eigenvalues of the $m \times m$ matrix A be denoted by $\lambda_1, \lambda_2, \ldots, \lambda_m$. (a) The eigenvalues of the inverse matrix A^{-1} are $\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}$, assuming that the inverse exists. The eigenvectors are the same as those of A. (b) The eigenvalues of the shifted matrix $A - sI$ are $\lambda_1 - s, \lambda_2 - s, \ldots, \lambda_m - s$. and the eigenvectors are the same as those of A.

This Lemma suggests:

Inverse Power Iteration

Inverse Power Iteration finds an eigenvalue with the smallest magnitude.

Lemma. Let the eigenvalues of the $m \times m$ matrix A be denoted by $\lambda_1, \lambda_2, \dots, \lambda_m$. (a) The eigenvalues of the inverse matrix A^{-1} are $\lambda_1^{-1}, \lambda_2^{-1}, \dots, \lambda_m^{-1}$, assuming that the inverse exists. The eigenvectors are the same as those of A. (b) The eigenvalues of the shifted matrix $A - sI$ are $\lambda_1 - s, \lambda_2 - s, \dots, \lambda_m - s$. and the eigenvectors are the same as those of A.

This Lemma suggests:

1. To find an eigenvalue of A with the smallest magnitude, we can find the dominant eigenvalue of A^{-1} with the largest magnitude so that the Power Iteration applies.

Inverse Power Iteration

Wangtao Lu
Zhejiang University

Inverse Power Iteration finds an eigenvalue with the smallest magnitude.

Lemma. Let the eigenvalues of the $m \times m$ matrix A be denoted by $\lambda_1, \lambda_2, \ldots, \lambda_m$. (a) The eigenvalues of the inverse matrix A^{-1} are $\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}$, assuming that the inverse exists. The eigenvectors are the same as those of A. (b) The eigenvalues of the shifted matrix $A - sI$ are $\lambda_1 - s, \lambda_2 - s, \ldots, \lambda_m - s$. and the eigenvectors are the same as those of A.

This Lemma suggests:

1. To find an eigenvalue of A with the smallest magnitude, we can find the dominant eigenvalue of A^{-1} with the largest magnitude so that the Power Iteration applies.
2. To find an eigenvalue of A closest to s, we can find the eigenvalue of $A - sI$ with the smallest magnitude, or we can find the dominant eigenvalue of $(A - sI)^{-1}$ with the largest magnitude so that the Power Iteration applies.

Inverse Power Iteration-Code

Wangtao Lu
Zhejiang University

Power iteration for dominant eigenvalue
Given initial vector x_0
for $j = 1, 2, 3, \ldots$
$u_{j-1} = x_{j-1}/||x_{j-1}||_2$
$x_j = Au_{j-1}$
$\lambda_j = u_{j-1}^T x_j$
end
$u_j = x_j/||x_j||_2$.

Inverse Power iteration for eigenvalue closest to s
Given initial vector x_0 and shift s
for $j = 1, 2, 3, \ldots$
$u_{j-1} = x_{j-1}/||x_{j-1}||_2$
$x_j = (A - sI)^{-1}u_{j-1}$
$\lambda_j = u_{j-1}^T x_j$
end
$u_j = x_j/||x_j||_2$.

Inverse Power Iteration-Code

Power iteration for dominant eigenvalue

Given initial vector x_0

for $j = 1, 2, 3, \ldots$

$u_{j-1} = x_{j-1}/||x_{j-1}||_2$

$x_j = Au_{j-1}$

$\lambda_j = u_{j-1}^T x_j$

end

$u_j = x_j/||x_j||_2$.

Inverse Power iteration for eigenvalue closest to s

Given initial vector x_0 and shift s

for $j = 1, 2, 3, \ldots$

$u_{j-1} = x_{j-1}/||x_{j-1}||_2$

solve $(A - sI)x_j = u_{j-1}$ for x_j

$\lambda_j = u_{j-1}^T x_j$

end

$u_j = x_j/||x_j||_2$.

Example 2. Assume that A is a 5×5 matrix with eigenvalues $-5, -2, 1/2, 3/2, 4$. Find the eigenvalues and convergence rate expected when applying (a) Power iteration. (b) Inverse Power iteration with shift $s = 0$. (c) Inverse Power iteartion with shift $s = 2$.

Rayleigh Quotient Iteration

Power iteration

Given initial vector x_0

for $j = 1, 2, 3, \ldots$

$u_{j-1} = x_{j-1}/||x_{j-1}||_2$

$x_j = Au_{j-1}$

$\lambda_j = u_{j-1}^T x_j$

end

$u_j = x_j/||x_j||_2.$

Rayleigh Quotient iteration

Given initial vector x_0

for $j = 1, 2, 3, \ldots$

$u_{j-1} = x_{j-1}/||x_{j-1}||_2$

$\lambda_j = u_{j-1}^T A u_{j-1}$

solve $(A - \lambda_j I)x_j = u_{j-1}$ for x_j

end

$u_j = x_j/||x_j||_2.$

QR Algorithm

The Power Iteration method only finds a single eigenvalue and its associated eigenvector at a time. Not very efficient. We hope to find all eigenvectors and all eigenvalues simultaneously. This section we will introduce the QR Algorithm to find all eigenvalues at once.

Simultaneous iteration

We begin with m pairwise orthogonal initial vectors $v_1, \ldots, v_m$.
$Av_1, \ldots, Av_m$ are no longer guaranteed to be orthogonal to one another.
$A^k v_1, \ldots, A^k v_m$ all would converge to the dominant eigenvector.
To avoid this, we use QR algorithm to orthogonalize the resulting matrix after each multiplication. That is
Set $\overline{Q_0} = I$.

1. QR factorize $A\overline{Q_0} = \overline{Q_1}R_1$.
2. QR factorize $A\overline{Q_1} = \overline{Q_2}R_2$.
3. QR factorize $A\overline{Q_2} = \overline{Q_3}R_3$. ...

Eventually, we expect that $\overline{Q_k}$ converges to a matrix of all eigenvectors of A, and the diagonal elements of $\overline{Q_k}^T A\overline{Q_k}$ converge to all eigenvalues!

Simultaneous iteration

We begin with m pairwise orthogonal initial vectors $v_1, \ldots, v_m$.
$Av_1, \ldots, Av_m$ are no longer guaranteed to be orthogonal to one another.
$A^k v_1, \ldots, A^k v_m$ all would converge to the dominant eigenvector.
To avoid this, we use QR algorithm to orthogonalize the resulting matrix after each multiplication. That is
Set $\overline{Q_0} = I$.

1. QR factorize $A\overline{Q_0} = \overline{Q_1}R_1$.
2. QR factorize $A\overline{Q_1} = \overline{Q_2}R_2$.
3. QR factorize $A\overline{Q_2} = \overline{Q_3}R_3$. ...

Eventually, we expect that $\overline{Q_k}$ converges to a matrix of all eigenvectors of A, and the diagonal elements of $\overline{Q_k}^T A\overline{Q_k}$ converge to all eigenvalues! Since,

$$\overline{Q_k}\mathrm{diag}\{\lambda_1, \cdots, \lambda_m\} = A\overline{Q_k} \to \mathrm{diag}\{\lambda_1, \cdots, \lambda_m\} = \mathrm{diag}\{\overline{Q_k}^T A\overline{Q_k}\}.$$

Unshifted QR Algorithm

Wangtao Lu
Zhejiang University

Set $Q_0 = I$ and $R_0' = A$.

1. QR factorize $R_0' Q_0 = Q_1 R_1'$. We find that

$$A_1 := R_1' Q_1 = Q_1^T A Q_0 Q_1 = Q_1^{-1} A Q_1,$$

   that is, $A_1 \sim A$ so that A_1 and A share the same eigenvalues.

2. Shall we need to perform QR Factorization of AQ_1?

Unshifted QR Algorithm

Set $Q_0 = I$ and $R_0' = A$.

1. QR factorize $R_0' Q_0 = Q_1 R_1'$. We find that
$$A_1 := R_1' Q_1 = Q_1^T A Q_0 Q_1 = Q_1^{-1} A Q_1,$$
that is, $A_1 \sim A$ so that A_1 and A share the same eigenvalues.
2. Shall we need to perform QR Factorization of AQ_1? No, we only need to perform
$$A_1 = R_1' Q_1 = Q_2 R_2'!$$
we find that
$$A_2 := R_2' Q_2 = Q_2^T A_1 Q_2 = Q_2^{-1} A_1 Q_2 \sim A_1 \sim A,$$
so that A_2 share the same eigenvalues with A.
3. Repeating the steps, we get a sequence of pairwise similar matrices $\{A_k\}_{k=1}^{\infty}$, and two auxilary matrices $\{Q_k, R_k'\}_{k=0}^{\infty}$

Unshifted QR Algorithm

Normalized Simultaneous Iteration:
Set $\overline{Q_0} = I$.

1. $A\overline{Q_0} = \overline{Q_1}R_1$.
2. $A\overline{Q_1} = \overline{Q_2}R_2$.
3. $A\overline{Q_2} = \overline{Q_3}R_3$.

...

Unshifted QR Algorithm:
Set $Q_0 = I$ and $R_0' = A$.

1. Set $A_0 = R_0'Q_0$, $A_0 = Q_1R_1'$.
2. Set $A_1 = R_1'Q_1$, $A_1 = Q_2R_2'$.
3. Set $A_2 = R_2'Q_2$, $A_2 = Q_3R_3'$.

...

Relations:

$$\overline{Q_k} = Q_0Q_1\cdots Q_k, \quad R_k = R_k',$$

and

$$A_k = Q_k^TA_{k-1}Q_k = \cdots = (Q_0Q_1\cdots Q_k)^TA_0(Q_0Q_1\cdots Q_k) = \overline{Q_k}^TA\overline{Q_k}$$

Eigenvectors: $\overline{Q_k} = Q_0Q_1\cdots Q_k$ as $k \to \infty$.
Eigenvalues: diagonal elements of $\overline{Q_k}^TA\overline{Q_k} = A_k$ as $k \to \infty$.
Theorem. Assume that A is a **symmetric** $m \times m$ matrix with eigenvalues $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_m|$. The unshifted QR algorithm converges linearly to the eigenvectors and eigenvalues of A. **As $k \to \infty$, A_k converges to a diagonal matrix containing the eigenvalues on the main diagonal and $\overline{Q_k} = Q_1\cdots Q_k$ converges to an othogonal matrix whose columns are the eigenvectors.**

A failure

Example 3.A tie for dominant eigenvector for a symmetric matrix:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Example 4.A tie for complex eigenvalues:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Real Schur form and the QR algorithm

A matrix T has **real Schur form** if it is upper triangular, except possibly for 2×2 blocks on the main diagonal.
For example,

$$\begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & x & x & x \\ & & & & x \end{bmatrix}$$

has real Schur form.

Theorem. Let A be an $n \times n$ matrix with real entries. Then there exists an orthogonal $n \times n$ matrix Q and an $n \times n$ matrix T in real Schur form such that $A = Q^T TQ$.

Shifted QR Algorithm–preliminary version

Wangtao Lu
Zhejiang University

Set $Q_0 = I$ and $A_0 = AQ_0$.

1. Choose a proper s_0. QR factorize $A_0 - s_0 I = Q_1 R_1$. We find that

$$A_1 = s_0 I + R_1 Q_1 = Q_1^T sIQ_1 + Q_1^T (A_0 - sI) Q_1 = Q_1^T (A_0) Q_1,$$

   that is, $A_1 \sim A_0$ so that A_1 and A_0 share the same eigenvalues.
2. Choose a proper s_1. QR factorize

$$A_1 - s_1 I = Q_2 R_2.$$

   we find that

$$A_2 = R_2 Q_2 + s_1 I = Q_2^T A_1 Q_2 \sim A_1 \sim A_0,$$

   so that $A_2 = R_2 Q_2$ share the same eigenvalues with A.
3. Repeat the steps, we get a sequence of pairwise similar matrices $\{A_k\}_{k=1}^{\infty}$.

Shifted QR Algorithm–preliminary version

Set $Q_0 = I$ and $A_0 = AQ_0$.

1. Choose a proper s_0. QR factorize $A_0 - s_0 I = Q_1 R_1$. We find that

$$A_1 = s_0 I + R_1 Q_1 = Q_1^T s I Q_1 + Q_1^T (A_0 - sI) Q_1 = Q_1^T (A_0) Q_1,$$

   that is, $A_1 \sim A_0$ so that A_1 and A_0 share the same eigenvalues.

2. Choose a proper s_1. QR factorize

$$A_1 - s_1 I = Q_2 R_2.$$

   we find that

$$A_2 = R_2 Q_2 + s_1 I = Q_2^T A_1 Q_2 \sim A_1 \sim A_0,$$

   so that $A_2 = R_2 Q_2$ share the same eigenvalues with A.

3. Repeat the steps, we get a sequence of pairwise similar matrices $\{A_k\}_{k=1}^{\infty}$.

How to choose s_k for $k = 1, 2, \ldots$ to speed up the shifted QR algorithm?

Deflation procedure

Suppose at k-th step,

$$A_k = \left[\begin{array}{c|c} A^k_{n-1,n-1} & * \\ \hline 0 & a^k_{nn} \end{array}\right]$$

If we choose $s_k = a_{nn}$, then we see

$$A_k - s_k I_n = \left[\begin{array}{c|c} A^k_{n-1,n-1} - s_k I & * \\ \hline 0 & 0 \end{array}\right]$$

Shall we QR factorize $A_k - s_k I$? No. If we get A_k with only one nonzero element in the last row, we in fact get an eigenvalue s_k. Next, we can remove the last row and the last column, and then QR factorize

$$A^k_{n-1,n-1} - s_k I_{n-1} = Q_{k+1} R_{k+1} \to A_{k+1} = R_{k+1} Q_{k+1} + s_k I_{n-1}.$$

Notice, A_{k+1} is of size one less than A_k!
Then, we hope that A_k converges and deflates to a constant!

Shifted QR Algorithm–revised

Set $Q_0 = I$ and $A_0 = AQ_0$.

1. Choose s_1 be the bottom-right entry of A_0.
   a. If $A_0 - s_1 I$ has a zero or near zero bottom row, mark s_1 as an eigenvalue and deflate A_0 by deleting the last row and the last column of A_0. QR factorize $A_0(1:n-1, 1:n-1) - s_1 I = Q_1 R_1$.
   b. Otherwise, QR factorize $A_0 - s_1 I = Q_1 R_1$.

   We find that,

$$A_1 = s_1 I + R_1 Q_1 \sim A_0(1:n-1, 1:n-1), \text{or} \quad A_1 \sim A_0.$$

2. Choose s_2 to be the bottom-right entry of A_1.
   a. If $A_1 - s_2 I$ has a zero or near zero bottom row, mark s_2 as an eigenvalue and deflate A_1 by deleting the last row and the last column of A_1. QR factorize $A_1(1:n-2, 1:n-2) - s_2 I = Q_2 R_2$.
   b. Otherwise, QR factorize $A_1 - s_2 I = Q_2 R_2$.

   We find that,

$$A_2 = s_2 I + R_2 Q_2 \sim A_1(1:n-2, 1:n-2), \text{or} \quad A_2 \sim A_1.$$

3. Repeat the steps until that $\{A_k\}$ finally is deflated to empty. The shifting parameters when deflation conditions hold are in fact eigenvalues.

A failure

The above procedure fails in the following example!

$$\begin{bmatrix} 0 & -1 & & \\ 1 & 0 & & \\ & & 0 & -1 \\ & & 1 & 0 \end{bmatrix}$$

Why?

Complex eigenvalues identification!

Suppose in the previous precedure, we cannot deflate A by Shifted QR algorithms after a number of QR Algorithms.
What should we expect? we expect that the following matrix is produced

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ & & & {\color{red}x} & {\color{red}x} \\ & & & {\color{red}x} & {\color{red}x} \end{bmatrix}$$

Thus, we deflat A twice by eleminate the last two rows and the last two columns, mark the two eigenvalues of the principal submatrix on the last two rows as eigenvalues of A.

Shifted QR algorithm–Setup Threshold for computing complex eigenvalues

We introduce a variable COUNT and a constant THRESHOLD(e.g., 10000).

0. Let COUNT$= 0$.
1. Do Shifted QR algorithsm by setting the shifting parameter s_1 as the bottom-right entry of A_0. Let COUNT increase by 1.
2. Repeating shifted QR algorithms k times until $A_k - s_k I$ has a zero bottom row. We encounter two cases:
3. If when $k =$THRESHOLD$+1$, $A_k - s_k I$ still doesn't have a zero bottom row, we deflat A twice and mark the two eigenvalues of the principal submatrix on the last two rwos as eigenvalues of A.

3'. If $k <$THRESHOLD, we deflat A_k and mark $s_k$as an eigenvalue of A.

4. Repeat the above procedures unitl A_k is deflated to empty.

A failure

The above procedure fails in the following example!

$$\begin{bmatrix} & & 0 & -1 \\ & & 1 & 0 \\ 0 & -1 & & \\ 1 & 0 & & \end{bmatrix}$$

Why?

Upper Hessenberg form

A matrix $A = (a_{ij})_{n\times n}$ has **Upper Hessenberg form** if $a_{ij} = 0$ for $i > j + 1$.
For example,

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{bmatrix}$$

has Upper Hessenberg form.
Theorem. Let A be an $n \times n$ matrix with real entries. Then there exists an orthogonal $n \times n$ matrix Q and an $n \times n$ matrix B in Upper Hessenberg form such that $A = Q^T BQ$.

Two examples

Example 1. Put

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 3 & 5 & -5 \\ 4 & 0 & 0 \end{bmatrix}$$

into Upper Hessenberg form.

Example 2. Put

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

into Upper Hessenberg form.

Hessenberg form for general matrix

Suppose

$$A = \left[\begin{array}{c|c} a_{11} & a^T \\ \hline b & A_{n-1,n-1} \end{array}\right] \in \mathbb{R}^{n\times n}.$$

First, we find the Householder reflector $H_1 \in$ s.t.,

$$H_1 b = \begin{bmatrix} \pm||b||_2 \\ 0_{n-2,1} \end{bmatrix}.$$

Then,

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & H_1 \end{array}\right] A = \left[\begin{array}{c|c} a_{11} & a^T \\ \hline \begin{bmatrix} ||b||_2 \\ 0 \end{bmatrix} & H_1 A_{n-1,n-1} \end{array}\right]$$

. Don't forget that we need to right multiply the above by the transpose to keep the eigenvalues unchanged.

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & H_1 \end{array}\right] A \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & H_1 \end{array}\right]^{-1} = \left[\begin{array}{c|c} a_{11} & a^T H_1^T \\ \hline \begin{bmatrix} ||b||_2 \\ 0 \end{bmatrix} & H_1 A_{n-1,n-1} H_1^T \end{array}\right]$$

Repeating the above procedure, we get the Hessenberg form.

Wangtao Lu
Zhejiang University

Hessenberg form for general matrix

Suppose

$$A = \left[\begin{array}{c|c} a_{11} & a^T \\ \hline b & A_{n-1,n-1} \end{array}\right] \in \mathbb{R}^{n\times n}.$$

First, we find the Householder reflector $H_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ s.t.,

$$H_1 b = \begin{bmatrix} \pm||b||_2 \\ 0_{n-2,1} \end{bmatrix}.$$

Then,

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & H_1 \end{array}\right] A = \left[\begin{array}{c|c} a_{11} & a^T \\ \hline \begin{bmatrix} ||b||_2 \\ 0 \end{bmatrix} & H_1 A_{n-1,n-1} \end{array}\right]$$

. Don't forget that we need to right multiply the above by the transpose to keep the eigenvalues unchanged.

$$\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & H_1 \end{array}\right] A \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & H_1 \end{array}\right]^{-1} = \left[\begin{array}{c|c} a_{11} & a^T H_1^T \\ \hline \begin{bmatrix} ||b||_2 \\ 0 \end{bmatrix} & H_1 A_{n-1,n-1} H_1^T \end{array}\right]$$

Repeating the above procedure, we get the Hessenberg form.

12.3 Singular Value Decomposition

Wangtao Lu
Zhejiang University

Theorem. For every $m \times n$ matrix A, there are orthonormal sets $\{u_1, \ldots, u_m\}$ and $\{v_1, \ldots, v_n\}$, together with nonnegative numbers $s_1 \geq \ldots \geq s_n \geq 0$, satisfying

$$Av_i = s_i u_i, \tag{1}$$

for $1 \leq i \leq \min(m, n)$.
Set $V = [v_1|v_2|\cdots|v_n]$, $S = \mathrm{diag}\{s_1, \cdots, s_n\}$, $U = [u_1|u_2|\cdots|u_m]$, then we get a **singular value decomposition** of A, i.e.,

$$A = USV^T,$$

where the v_i are called the **right singular vectors** of A, the u_i are called the **left singular vectors** of A, and the s_i are the **singular values** of A. Notice that U and V are **orthogonal matrices.**

How to numerically find the SVD decomposition?

Since
$$A^T A = VS^2 V^T,$$
we can find eigenvalues and eigenpairs of $A^T A$ to get S and V.
Similarly,
$$AA^T = US^2 U^T,$$
we can find eigenvalues and eigenpairs of AA^T to get S and U.
Numerically unstable!

A more stable way!

To find singular values/vectors of $A \in \mathbb{R}^{m \times n}$, we can find eigenvalues/eigenvectors of the $(m+n) \times (m+n)$ matrix

$$B = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}.$$

For example, suppose $[v, w]^T$ denotes a $(m+n)$-vector such that

$$B \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \lambda \begin{bmatrix} v \\ w \end{bmatrix}.$$

Then, we get

$$A^T A v = \lambda^2 v.$$

Thus, $|\lambda|$ gives a singular value of A, v gives the associated right singular vector of A.

Contents

Wangtao Lu
Zhejiang University

Assignment VII

Wangtao Lu
Zhejiang University

Assignment VI:
P539, 12.1: Computer Problems, 1.
P548, 12.2: Computer Problems, 1.
Due date: November 18, 2020.

Final exam

The final exam will be held **11/18/2020 from 8:00 to 11:30** at **XX XXXX**. Exam questions may include:

- Multiple choices.
- True or False
- Problems.

Answers to the above questions must be written in English. Don't worry about gramma mistakes.