MATH3322 - Matrix Computation

Taught by Jianfeng Cai

Notes by Aaron Wang

Contents

1 March 22nd, 2019

1.1 Eigenvalue Decomposition

Definition 1.1. Let $A \in \mathbb{R}^{n\times n}$ be a square matrix. A non-zero vector x is an **eigenvector** of A with $\lambda \in \mathbb{C}$ being the corresponding **eigenvalue** if:

$$Ax = \lambda x.$$

- Even if A is a real matrix, its eigenvalue and eigenvectors can be complex
- The set of eigenvalues of A is called the spectrum of A. The spectral radius $\rho(A)$ is the maximum value $|\lambda|$ over all eigenvalues of A.
- If (λ, x) is an eigenpair of A, then:

$$\begin{aligned}\left(\lambda^2, x\right) &\text{ is a eigenpair of } A^2\\ (\lambda-\sigma, x) &\text{ is a eigenpair of } A-\sigma I\\ \left(\frac{1}{\lambda-\sigma}, x\right) &\text{ is a eigenpair of } (A-\sigma I)^{-1}.\end{aligned}$$

Proof. Since (λ, x) is an eigenpair of A, $Ax = \lambda x$ Multiplying both sides by A from the left:

$$\begin{aligned}A\cdot Ax = \lambda Ax &\implies A^2x = \lambda Ax = \lambda\cdot\lambda x = \lambda^2 x.\\ Ax - \sigma x = \lambda x - \sigma x &\implies (A-\sigma I)x = (\lambda-\sigma)x\\ \implies x = (\lambda-\sigma)(A-\sigma I)^{-1}x &\implies (A-\sigma I)^{-1}x.\end{aligned}$$

□

Definition 1.2. Two matrices A and B are **similar** with each other if there exists a nonsingular matrix T such that

$$B = TAT^{-1}.$$

Theorem 1.3

If A and B are similar, then A and B have the same eigenvalues.

Proof. Since A, B are similar, $B = TAT^{-1}$, which implies $A = T^{-1}BT$. If (λ, x) is an eigenpair of A, then $Ax = \lambda x$, so that

$$T^{-1}BTx = \lambda x \implies B(Tx) = \lambda(Tx).$$

Thus, (λ, Tx) is an eigenpair of B. i.e. any eigenvalue of A is an eigenvalue of B. The reverse is similar. □

Definition 1.4. An **eigenvalue decomposition** of a square matrix $A \in \mathbb{R}^{n\times n}$ is a factorization

$$A = X\Lambda X^{-1},$$

where $X \in \mathbb{C}^{n\times n}$ is non-singular and $\Lambda \in \mathbb{C}^{n\times n}$ is diagonal.

- If $A \in \mathbb{R}^{n\times n}$ admits an eigenvalue decomposition, then

$$AX = X\Lambda.$$

If we rewrite $X = [x_1 x_2 \ldots x_n]$ with $x_i \in \mathbb{C}^n$ the i-th column of x, and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2 \ldots, \lambda_n) = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$ with $\lambda_i \in \mathbb{C}$ being the i-th diagonal of Λ, then

$$A[x_1 x_2 \ldots x_n] = [x_1 x_2 \ldots x_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}.$$

$$\implies [Ax_1 Ax_2 \ldots Ax_n] = [\lambda_1 x_1 \lambda_2 x_2 \ldots \lambda_n x_n].$$

$$\implies Ax_i = \lambda_i x_i, \quad i = 1, 2 \ldots, n.$$

In other words $(\lambda_i, x_i)\,, i = 1, 2, \ldots, n$ are eigenpairs of A.

- Since X is nonsingular, x_i are linearly independent. So, x_i are n independent eigenvectors, which span $\mathbb{C}^n$.

- Eigenvalue decomposition implies $X^{-1}AX = \Lambda$, so that we also say A is diagonalizable.

- Eigenvalue decomposition does not always exist, as a square matrix $A \in \mathbb{R}^{n\times n}$ does not always have n independent eigenvectors.

- Though $A \in \mathbb{R}^{n\times n}$ is real, the eigenvalue decomposition may be complex.

1.2 Characteristic Polynomial

Definition 1.5. The **characteristic polynomial** of $A \in \mathbb{R}^{n\times n}$ denoted P_A is a degree n polynomial defined by

$$P_A(z) = \det(zI - A), \quad \text{where } z \in \mathbb{C}.$$

Let (λ_1, x) be an eigenpair of A. Then $Ax = \lambda x$, which is equivalent to:

$$(\lambda I - A)\, x = 0.$$

Since x is non-zero, $\lambda I - A$ has a non-zero solution. Therefore, $\lambda I - A$ is singular. That is $\det(\lambda I - A) = P_A(\lambda) = 0$. Thus, λ is an eigenvalue of A iff $P_A(\lambda) = 0$, and the corresponding eigenvector x are non-zero solutions of $(\lambda I - A)\, x = 0$.

Example 1.6

$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. The characteristic polynomial is:

$$P_A(z) = \det\left(zI - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\right) = \det\left(\begin{bmatrix} z & -1 \\ 0 & z \end{bmatrix}\right) = z^2.$$

Therefore, $P_A(\lambda) = \lambda^2 = 0 \implies \lambda_1 = \lambda_2 = 0$ are the eigenvalues of A.

For eigenvectors, solve $(0I - A) = 0$, i.e.

$$\begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix} x = 0 \implies x = \begin{bmatrix} q \\ 0 \end{bmatrix}.$$

As there is only one independent eigenvector, A is not diagonalizable (i.e. no eigenvalue decomposition.

Example 1.7

$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. The characteristic polynomial is:

$$P_A(z) = \det\left(\begin{bmatrix} z & 1 \\ -1 & z \end{bmatrix}\right) = z^2 + 1.$$

Therefore, $P_A(\lambda) = \lambda^2 + 1 = 0 \implies \lambda_1 = i, \quad \lambda_2 = -i$ are the eigenvalues.
For eigenvector of $\lambda_1 = i$, solve $(iI - A)\,x = 0$, i.e.

$$\begin{bmatrix} i & 1 \\ -1 & i \end{bmatrix} x = 0 \implies x = \alpha \begin{bmatrix} i \\ 1 \end{bmatrix} \quad \alpha \in \mathbb{C}.$$

Therefore, a corresponding eigenvector is $x_i = \begin{bmatrix} i \\ 1 \end{bmatrix}$.

For eigenvector of $\lambda_2 = -i$:

$$\begin{bmatrix} -i & 1 \\ -1 & -i \end{bmatrix} x = 0 \implies x = \beta \begin{bmatrix} i \\ -1 \end{bmatrix} \quad \beta \in \mathbb{C}.$$

The corresponding eigenvector is $x_2 = \begin{bmatrix} i \\ -1 \end{bmatrix}$.

Define $X = \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} i & i \\ 1 & -1 \end{bmatrix}, \Lambda = \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} = \begin{bmatrix} i & \\ & -i \end{bmatrix}, X^{-1} = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2}i & -\frac{1}{2} \end{bmatrix}$,
Therefore $A = X\Lambda X^{-1}$

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} i & i \\ 1 & -1 \end{bmatrix} \begin{bmatrix} i & \\ & -i \end{bmatrix} \begin{bmatrix} -\frac{1}{2}i & \frac{1}{2} \\ -\frac{1}{2}i & -\frac{1}{2} \end{bmatrix}.$$

This shows that a real matrix may have a complex eigenvalue decomposition.

> **Remark 1.8** — However, we don't usually solve the characteristic equation, as polynomial root-finding is not numerically stable in general.

1.3 Special Case: Symmetric Matrix and SPD Matrix

Assume $A \in \mathbb{R}^{n\times n}$ is symmetric. Then

1. The eigenvalues of A are real.

   Proof. Let (λ, x) be an eigenpair of A. Then, $Ax = \lambda x$.
   Multiply both sides by $x^* \equiv \overline{x^T}$ (conjugate transpose) from the left:

   $$x^*Ax = \lambda x^*x \implies \lambda = \frac{x^*Ax}{x^*x}.$$

   - x^*Ax is real because $\overline{x^*Ax} = \overline{(x^*Ax)}^T = \overline{x^T A^T \overline{x}} = x^*Ax$
   - x^*x is also real, because $\overline{x^*x} = \overline{(x^*x)}^T = \overline{x^T\overline{x}} = x^*x$.
   - As such, $\lambda = \frac{x^*Ax}{x^*x}$ is real.

   □

2. Eigenvectors corresponding to distinct eigenvalues are orthogonal.
3. A is always diagonalizable, and the eigenvalue decomposition has a special form

   $$A = Q\Lambda Q^T$$

   where $Q \in \mathbb{R}^{n\times n}$ is orthonormal and $\Lambda \in \mathbb{R}^{n\times n}$ is diagonal.
4. If A is SPD, then all eigenvalues are positive.
5. If A is SPSD, then all eigenvalues are non-negative.

   Proof. Let (λ, x) a be an eigenpair of A. then $Ax = \lambda x$, and λ, x are real. So

   $$x^TAx = \lambda x^Tx \implies \lambda = \frac{x^TAx}{x^Tx} > 0.$$

   if A is SPD. If A is SPSD, then $\lambda = \frac{x^TAx}{x^Tx} \geq 0$, since $x^TAx \geq 0$. □

2 March 27th, 2019

2.1 Computation of Eigenvalue Decomposition

For simplicity, we assume that $A \in \mathbb{R}^{n\times n}$ is symmetric, so that all eigenvalues/eigenvectors are real. Let λ_i $i = 1, 2 \ldots, n$ be the eigenvalues of A, which are sorted in magnitude, i.e.

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|.$$

The corresponding eigenvectors are denoted by q_i. We have

$$Q = \begin{bmatrix} q_1 & q_2 & \ldots & q_n \end{bmatrix} \in \mathbb{R}^{n\times n}$$

satisfying $Q^TQ = Q^T = I$.

Definition 2.1. Let $A \in \mathbb{R}^{n\times n}$ be symmetric. For a given vector $x \in \mathbb{R}^n$, the **Rayleigh Quotient** is defined by

$$r(x) = \frac{x^T A x}{x^T x}.$$

If x is an eigenvector,

$$r(x) = \frac{x^T A x}{x^T x} = \frac{\lambda x^T x}{x^T x} = \lambda,$$

i.e. $r(x)$ is an eigenvalue.

The eigenvalues are critical points of $r(x)$, with $\nabla r(x) = 0$. It can be proven that

$$\min_i \lambda_i = \min_{x\neq 0} r(x).$$

> **Remark 2.2** — This can be extended to non-symmetric matrices/ matrices or eigenvalues that are complex.

2.2 Power Iteration

Purpose: Find λ_1 and its associated eigenvector x_1, with $\|x_1\|_2 = 1$.

Algorithm 2.3

```
1. Choose $y^{(0)} \in R^n$ s.t. $\|y^{(0)}\|_2 = 1$.
2. for $k = 1, 2, \dots, n$
       $z^{(k)} = Ay^{(k-1)}$
       $y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2}$
       $\mu^{(k)} = \frac{\left(y^{(k)}\right)^T Ay^{(k)}}{\left(y^{(k)}\right)^T y^{(k)}} = \left(y^{(k)}\right)^T Ay^{(k)}.$
```

> **Remark 2.4** — $y^{(k)}$ is an approximation to $\pm x_1$, $\mu^{(k)}$ is an approximation to λ_1.

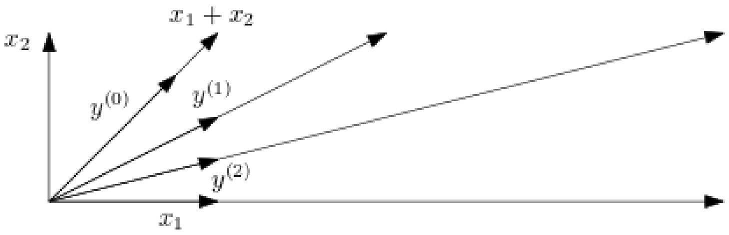

Figure 1

- Assume $(2, x_1)$, $(1, x_2)$ are two eigenpairs of $A \in \mathbb{R}^{2\times 2}$ (so that $x_1 \perp x_2$).
- Assume $y^{(0)} = \frac{1}{\sqrt{2}}(x_1 + x_2)$

- $k = 1$:

$$z^{(1)} = Ay^{(0)} = A\left(\frac{1}{\sqrt{2}}(x_1 + x_2)\right) = \frac{1}{\sqrt{2}}(Ax_1 + Ax_2) = \frac{1}{\sqrt{2}}(2x_1 + x_2).$$

$$y^{(1)} = \frac{1}{\sqrt{5}}(2x_1 + x_2).$$

Note that $y^{(k)}$ approaches x_1 more than x_2.

$\vdots$

- $k + 1$:

$$z^{(k+1)} = Ay^{(k)} = A\left(\frac{1}{\sqrt{2^{2k}+1}}\left(2^k x_1 + x_2\right)\right) = \frac{1}{\sqrt{2^{2k}+1}}\left(2^{k+1}x_1 + x_2\right).$$

If the component of x_1 is non-zero, then it will converge to x_1, i.e. as long as $y^{(0)}$ is not a multiple of x_2, it will converge to x_1.

Claim 2.5. Power iteration may not be convergent:

Example 2.6

Assume $(1, x_1)$, $(-1, x_2)$ are two eigenpairs of $A \in \mathbb{R}^{2\times 2}$. Assume $y^{(0)} = \frac{1}{\sqrt{2}}(x_1 + x_2)$.

$$k = 1 : z^{(1)} = Ay^{(0)} = \frac{1}{\sqrt{2}}(x_1 - x_2).$$

$$y^{(1)} = \frac{1}{\sqrt{2}}(x_1 - x_2).$$

$$k = 2 : z^{(2)} = \frac{1}{\sqrt{2}}(x_1 + x_2)$$

$$y^{(2)} = \frac{1}{\sqrt{2}}(x_1 + x_2).$$

which just repeats itself.

Remark 2.7 — Try with $(-2, x_1)$, $(1, x_2)$. Does not converge, but we can get the direction of x_1 since both x_1 and $-x_1$ are eigenvectors.

Remark 2.8 — Power iteration may not converge to (λ_1, x_1), e.g. $y^{(0)} = x_2$. This is because there is no x_1 component.

2.3 Analysis of Power Iteration

We will show $|\langle y^{(k)}, x\rangle| \to 1$. It is the same as $1 - \langle y^{(k)}, x_1\rangle^2 \to 0,\ k \to \infty$

Theorem 2.9

Assume $A \in \mathbb{R}^{n\times n}$ is symmetric and $|\lambda_1| > |\lambda_2|$ (otherwise they might be amplified at the same rate).

If $\langle y^{(0)}, x_1\rangle \neq 0$, then $\exists C_0 > 0$ depending on $y^{(0)}$ only such that

$$\left(1 - \langle y^{(k)}, x_1\rangle^2\right)^{\frac{1}{2}} \leq C_0 \left|\frac{\lambda_2}{\lambda_1}\right|^k \to 0, \text{ as } k \to \infty.$$

Consequently,

- $\min\{\|y^{(k)} - x_1\|_2, \|y^{(k)} + x_1\|_2\} \leq \sqrt{2}C_o \left|\frac{\lambda_2}{\lambda_1}\right|^k$, i.e. $y^{(k)} \to \pm x_1$
- $|\mu^{(k)} - \lambda_1| \leq 2\sqrt{2}C_o \left|\frac{\lambda_2}{\lambda_1}\right|^k \to 0$

Proof. Note that

$$y^{(k)} = \frac{A^k y^{(0)}}{\|A^k y^{(0)}\|_2}.$$

Let $A = X\Lambda X^T$ be the eigenvalue decomposition of A. Then

$$A^k = X\Lambda X^T X\Lambda X^T \ldots X\lambda X^T = X\Lambda^k X^T.$$

So

$$A^k y^{(0)} = X\Lambda^k X^T y^{(0)} = X\Lambda^k v$$

$$A^k y^{(0)} = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix} \begin{bmatrix} \lambda_1^k v_1 \\ \vdots \\ \lambda_n^k v_n \end{bmatrix} = \sum_{i=1}^n \lambda_i^k v_i x_i, \; v_i \in \mathbb{R}, \; x_i \in \mathbb{R}^n.$$

Because x_i are othronormal,

$$\|A^k y^{(0)}\|_2^2 = \sum_{i=1}^n \lambda_i^{2k} v_1^2 = \sum_{i=1}^n |\lambda_i|^{2k}|v_i|^2 = |\lambda_1|^{2k}|v_1|^2\left(1 + \ldots\right) \geq \left(|\lambda_1|^k|v_1|\right)^2$$

and

$$\langle y^{(k)}, x_1\rangle^2 = \frac{1}{\|A^k y^{(0)}\|_2^2}\langle A^k y^{(0)}, x_1\rangle^2 = \frac{1}{\|A^k y^{(0)}\|_2^2}\langle \sum_{i=1}^n \lambda_i^k v_i x_i, x_1\rangle^2 = \frac{1}{\|A^k y^{(0)}\|_2^2}\left(\lambda_1^k v_1\right)^2.$$

$$\leq \left|\frac{\lambda_2}{\lambda_1}^{2k}\right| \left(\left|\frac{v_2}{v_1}\right|^2 + \left|\frac{v_3}{v_1}\right|^2 + \left|\frac{v_4}{v_1}\right|^2 + \ldots + \left|\frac{v_n}{v_1}\right|^2\right) = C_0^2 \left|\frac{\lambda_2}{\lambda_1}^k\right|.$$

Thus

$$\sqrt{1 - \langle y^{(k)}, x_1\rangle^2} \leq C_0 \left|\frac{\lambda_2}{\lambda_1}^k\right|.$$

Because $C_0 < +\infty$, as $v_1 = \angle x_1, y^{(0)} \neq 0$ by assumption,

$$\sqrt{1 - \langle y^{(k)}, x_1\rangle^2} \leq C_0 \left|\frac{\lambda_2}{\lambda_1}^k\right| \to 0, \text{ as } k \to \infty.$$

$$\langle y^{(k)}, x_1\rangle^2 \geq 1 - C_0^2 \left|\frac{\lambda_2}{\lambda_1}^{2k}\right| \implies \langle y^{(k)}, x_1\rangle^2 \leq \|y^{(k)}\|_2^2 \|x_1\|_2^2 = 1.$$

So

$$1 - C_0^2 \left|\frac{\lambda_2}{\lambda_1}^{2k}\right| \leq 1.$$

If $\langle y^{(k)}, x_1\rangle \geq 0$, then

$$\|y^{(k)} - x_1\|_2 = \sqrt{\|y^{(k)}\|_2^2 + \|x_1\|_2^2 - 2\langle y^{(k)}, x_1\rangle} = \sqrt{2 - 2\langle y^{(k)}, x_1\rangle} \leq \left(2 - 2\sqrt{1 - C_0^2 \left|\frac{\lambda_2}{\lambda_1}^{2k}\right|}\right)^{\frac{1}{2}}.$$

I give up will do this later □

Remark 2.10 —

1. $\langle y^{(k)}, x_1\rangle = \cos\theta$. Genearlly,

$$\cos\angle(x, y) = \frac{\langle x, y\rangle}{\|x\|_2 \|y\|_2}.$$

2. The convergenec rate depends on $\left|\frac{\lambda_2}{\lambda_1}\right| < 1$, the smaller $\left|\frac{\lambda_2}{\lambda_1}\right|$, the faster the convergence. When $|\lambda_1| = |\lambda_2|$, the power iteration may not converge.

3. When $\langle y^{(0)}, x_1\rangle = 0$, then $C_0 = +\infty$, so y will not converge to λ_1.

4. In power iteration, only one matrix-product and several vector operations are used, the ocst per step is $O(n^2)$. If we want an approximate eigenvalue/eigenvector of error ϵ, we need to choose k, s.t.

$$C\left|\frac{\lambda_2}{\lambda_1}\right|^{\frac{k}{2}} \leq \epsilon \implies \left|\frac{\lambda_1}{\lambda_2}\right|^{\frac{k}{2}} \geq \frac{c}{\epsilon}.$$

$$\frac{k}{2}\log\left|\frac{\lambda_1}{\lambda_2}\right| \geq \log\left(\frac{c}{\epsilon}\right) \implies k \geq \frac{\log\left(\frac{c}{\epsilon}\right)}{\log\left(\left|\frac{\lambda_1}{\lambda_2}\right|\right)} \sim O\left(\log\left(\frac{1}{\epsilon}\right)\right).$$

Then the total computational cost is

$$O\left(\log\left(\frac{1}{\epsilon}\right) \cdot n^2\right).$$

5. Only the matrix-cector product involving A is needed. This means that A is not needed explicitly, only the subroutine to compute Ax is sufficient.

3 March 29th, 2019

3.1 Inverse Power Iteration

If $\lambda_i, \ i \in 1, \ldots, n$ with $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$ are eigenvalues of A, then $\frac{1}{\lambda_i}$ are eigenvalues of A^{-1} and

$$\frac{1}{|\lambda_1|} \leq \frac{1}{|\lambda_2|} \leq \ldots \leq \frac{1}{|\lambda_n|}.$$

Therefore, we can apply power iteration to A^{-1} to get λ_n and hence x_n. This is called the inverse power iteration.

Algorithm 3.1 1. Choose $y^{(0)} \in \mathbb{R}^n$ s.t. $\|y^{(0)}\|_2 = 1$

2. for $k = 1, 2, \ldots$

$$z^{(k)} = A^{-1} y^{(k-1)}$$
$$y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2}$$
$$\mu^{(k)} = \left(y^{(k)}\right)^T A y^{(k)}.$$

Remark 3.2 — 1. From the convergence of power iteration, if:

- $\langle y^{(0)}, x_n \rangle \neq 0$
- $\frac{1}{|\lambda_n|} > \frac{1}{|\lambda_{n-1}|}$ (i.e. $|\lambda_n| < |\lambda_{n-1}|$)
- A^{-1} is symmetric (always true because A is symmetric.

then the limit of the iteration is:

$$y^{(k)} \to \pm x_n, \quad \mu^{(k)} \to \lambda_n,$$

with a rate $\left(\frac{|\lambda_n|}{|\lambda_{n-1}|}\right)^{\frac{k}{2}}$

2. We need to solve $Az^{(k)} = y^{(k-1)}$ in each iteration, which can be done by Gaussian Elimination. But we only need to compute $A = LU$ before the iteration and then, in each iteration, we obtain:

$$z^{(k)} = U^{-1} L^{-1} y^{(k-1)},$$

which is just a forward and backward substitution.

- Thus the total computational cost is:

$$O(n^3) + O\left(n^2 \cdot \log\left(\frac{1}{\epsilon}\right)\right)$$

for an ϵ-solution, ($O(n^3)$ for the LU decomposition)

> 3. If $|\lambda_n|$ is very close to 0, then, A is very close to singular, meaning that the solution of $Az^{(k)} = y^{(k-1)}$ may have a large error. However, we can still get a very accurate solution.

4 April 3rd, 2019

4.1 Shifted Inverse Power Iteration Part 2

Let λ_i, $i = 1, 2, \ldots, n$ with $|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|$ be eigenvalues of A. Note that $(\lambda_i - \mu)^{-1}$ are eigenvalues of $(A - \mu I)^{-1}$.

Thus we can apply the power iteration to $(A - \mu I)^{-1}$:

Algorithm 4.1
Choose $y^{(0)} \in \mathbb{R}^n$ s.t. $\|y^{(0)}\|_2 = 1$
For $k = 1, 2, \ldots$

- $z^{(k)} = (A - \mu I)^{-1} y^{(k-1)}$ (Done by solving $(A - \mu I)z^{(k)} = y^{(k-1)}$.
- $y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2}$.
- $\mu^{(k)} = (y^{(k)})^T A y^{(k)}$ (the Rayleigh Quotient of A)

1. This iteration is the "shifted inverse power iteration"
2. To make the iteration converge to (λ_j, x_j), the following has to be satisfied:
   (a) μ is chosen s.t. $\frac{1}{|\lambda_j - \mu|}$ is the largest among $\frac{1}{|\lambda_i - \mu|}$

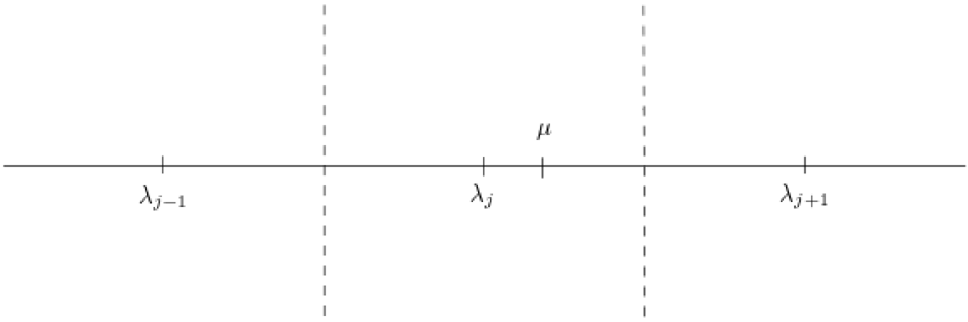

Figure 2: Choosing μ to guarantee convergence

   (b) $\langle y^{(0)}, x_j \rangle \neq 0$. (or else it will converge to another eigenvector.
   (c) The convergence rate depends on:

$$\frac{\frac{1}{|\lambda_j' - \mu|}}{\frac{1}{|\lambda_j - \mu|}} = \frac{|\lambda_j - \mu|}{|\lambda_j' - \mu}$$

   , where $\frac{1}{|\lambda_j' - \mu|}$ is the second largest among $\frac{1}{|\lambda_i - \mu|}$ and

$$|\lambda_j - \mu| < |\lambda_j' - \mu|.$$

(d) For an ϵ- precision solution, the computational complexity is

$$O(n^3) + O\left(\log \frac{1}{\epsilon} \cdot n^2\right),$$

since we only compute the LU decomposition of $(A - \mu I)$ once.

To accelerate the shifted power iteration, we can also use an adaptive shift (if we shift μ to be closer to the target eigenpair, it will converge faster).

Algorithm 4.2 (Rayleigh Quotient Iteration)

```
Choose $y^{(0)} \in \mathbb{R}^n$ s.t. $\|y^{(0)}\|_2 = 1$
$\mu^{(0)} = (y^{(0)})^T A y^{(0)}$
For $k = 1, 2, \dots$
    • $z^{(k)} = (A - \mu^{(k-1)} I)^{-1} y^{(k-1)}$ (Done by solving $(A - \mu I) z^{(k)} = y^{(k-1)}$.
    • $y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|_2}$.
    • $\mu^{(k)} = (y^{(k)})^T A y^{(k)}$ (the Rayleigh Quotient of A)
```

i.e. we choose μ to be close to the desired eigenvalue.

- This converges to some eigenpair (λ_i, x_i) such that λ_i is close to $\mu^{(0)}$.
- This Rayleigh Quotient iteration converges very fast (cubic).
- However since $(A - \mu^{(k-1)} I)$ is changing, we have to calcualte the LU decomposition each time.
- However, this will only converge to one eivenpair.

4.2 Simultaneous Iteration

To compute r eigenpairs:

Algorithm 4.3

```
Choose $Y^{(0)} \in R^{n \times r}$ s.t. $(Y^{(0)})^T (Y^{(0)}) = I$
For $k = 1, 2, \dots$
    • $Z^{(k)} = A Y^{(k-1}$
    • Set $Y^{(k)}$ to be the Q matrix in the QR decomposition of $Z^{(k)}$.
    • $\mu_i^{(k)} = (y_i^{(k)})^T A y_i^{(k)}$, where $y_i^{(k)}$ is the $i$-th column of $Y^{(k)}$
```

Under some assumption, we have:

$$\|y_i^{(k)} - \pm x\| \leq C\rho^k, i = 1, \dots, r,$$

where $\rho = \max\limits_{i=1,\dots,r} \frac{|\lambda_i+1}{\lambda_i} < 1$ and $|\mu_i^{(k)} - \lambda_i| \leq C\rho^k$

4.3 QR algorithm for Eigenvalue Decomposition

We set $r = n$ in the simultaneous power iteration

$$Z^{(k)} = AY^{(k-1)}$$
$$Y^{(k)}R^{(k)} = Z^{(k)},$$

i.e. let $Z^{(k)} = Y^{(k)}R^{(k)}$ be the QR decomposition of $Z^{(k)}$. Eliminating $Z^{(k)}$, we have

$$Y^{(k)}R^{(k)} = AY^{(k-1)} \iff (Y^{(k)})^T AY^{(k-1)} = R,$$

because $(Y^{(k)})^T Y^{(k)} = Y^{(k)}(Y^{(k)})^T = I$.

Let $A^{(k)} = (Y^{(k)})^T AY^{(k)}$, then

$$A^{(k-1)} = (Y^{(k-1)})^T AY^{(k-1)} = (Y^{(k-1)})^T Y^{(k)} R^{(k)}.$$

Since $(Y^{(k-1)})^T Y^{(k)}$ is orthogonal and $R^{(k)}$ is upper triangular, $A^{(k)}$ is just an orthogonal square matrix. Note that

$$A^{(k)} = (Y^{(k)})^T AY^{(k)} = (Y^{(k)})^T AY^{(k-1)}(Y^{(k-1)})^T Y^{(k)} = R^{(k)}(Y^{(k-1)})^T Y^{(k)}.$$

This means that after getting the QR decomposition of $A^{(k-1)}$, we swap the two matrices to get $A^{(k)}$.

Algorithm 4.4 (QR Algorithm)
Choose initial guess $A^{(0)} = (Q^{(0)})^T \forall Q^{(0)}$, e.g. $Q^{(0)} = I \implies A^{(0)} = A$.
For $k = 1, 2, \ldots$

- Compute the QR decomposition: $A^{(k-1)} = Q^{(k)}R^{(k)}$.
- Set $A^{(k)} = R^{(k)}Q^{(k)}$.

Remark 4.5 —

1. Note that $A^{(k)} = R^{(k)}Q^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$. By induction

$$A^{(k)} = (Q^{(k)})^T \cdots (Q^{(1)})^T (A^{(0)})^T A Q^{(0)} Q^{(1)} \cdots Q^{(k)} \implies A^{(k)} \text{ is similar to } A$$

   as $Q^{(0)}Q^{(1)} \cdots$ is an orthogonal matrix.

2. Since $Y^{(k)}$ is expected to converge to the eigenvectors of A,

$$A^{(k)} = (Y^{(k)})^T AY^{(k)} \text{ is expected to converge to } \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}.$$

   It can be proven that if the eigenvalues of A are well separated, then $A^{(k)} \to \Lambda$ and $Q^{(0)} \cdots Q^{(k)}$ converges to the eigenvectors of A.

3. Since QR decomposition and matrix-matrix product costs $O(n^3)$, the total computational cost is

$$O(kn^3), \text{ where } k \text{ is the number of iteration needed.}$$

> Note that if k iterations is done (or if $k \sim O(n)$), then it will be $O(n^4)$, which is expensive.

Note that even though the QR decomposition is not unique, it will still work since the QR have similar properties/structure.

4.4 Practical Implementation of QR Algorithm

The idea is to choose $Q^{(0)}$ such that $A^{(0)}$ is ”well structured”. This will allow QR decomposition to be done in $O(n^2)$. For our purpose, this ”structure” is to be tridiagonal.

Thus this implementation has two phases:

1. Find $Q^{(0)}$ such that $A^{(0)} = (Q^{(0)})^T A Q^{(0)}$ is tridiagonal.

2. QR decomposition of tridiagonal matrices is done in $O(n^2)$ and so that $A^{(k)}$ preserves the tridiagonal structure.

Algorithm 4.6 (Phase 1)
We will use the Householder transformation.

1. Let P_1 be the Householder transform s.t.

$$P_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & H_1 & \\ 0 & & & \end{bmatrix} : \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \\ \times & \times & & \times \end{bmatrix} \to \begin{bmatrix} \Delta & \Delta & \cdots & \Delta \\ \times & \times & \cdots & \times \\ 0 & & & \\ \vdots & \vdots & \ddots & \\ 0 & \times & & \times \end{bmatrix}.$$

Then $P_1 A P_1^T$

dsfont

5 April 12th, 2019

5.1 QR Algorithm for Non-Symmetric Matrix

- Phase 1: We can not reduce A to a tridiagonal matrix. Instead we reduce A to an upper Hessenberg Matrix

  Definition 5.1. A **Hessenberg matrix** is a special kind of square matrix, one that is ”almost” triangular. To be exact, an upper Hessenberg matrix has zero entries below the first subdiagonal, and a lower Hessenberg matrix has zero entries above the first superdiagonal.

$$\begin{bmatrix} 1 & 0 \\ 0 & H_1 \end{bmatrix} A = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \cdots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 0 \\ 0 & H_1 \end{bmatrix} A \begin{bmatrix} 1 & 0 \\ 0 & H_1 \end{bmatrix}^T = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \cdots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix}$$

$$\vdots$$

$$Q^{(0)} A \left(Q^{(0)}\right)^T = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \times \end{bmatrix}$$

which is a Hessenberg Matrix.

- Phase 2:

Algorithm 5.2

For $k = 1, 2, \ldots$

- Computer the QR Decomposition of $A^{(k)} = Q^{(k)} R^{(k)}$
- $A^{(k)} = R^{(k)} Q^{(k)}$.

- Note that $A^{(k)} = R^{(k)} Q^{(k)}$ is also a hessenberg Matrix:
  - For $A^{(k)} = R^{(k)} Q^{(k)} = R^{(k)} \left(G_k^{(1)}\right)^T \left(G_k^{(2)}\right)^T \cdots \left(G_k^{(n-1)}\right)^T$:

$$R^{(k)} \left(G_k^{(1)}\right)^T = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ & \times & \times & \cdots & \times \\ & & \times & \cdots & \times \\ & & & \ddots & \vdots \\ & & & & \times \end{bmatrix} \begin{bmatrix} c_1 & -s_1 & \\ s_1 & c_1 & \\ & & I \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ + & \times & \times & \cdots & \times \\ & & \times & \cdots & \times \\ & & & \ddots & \vdots \\ & & & & \times \end{bmatrix}.$$

$$R^{(k)} \left(G_k^{(1)}\right)^T \left(G_k^{(2)}\right)^T = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ + & \times & \times & \cdots & \times \\ & + & \times & \cdots & \times \\ & & & \ddots & \vdots \\ & & & & \times \end{bmatrix}.$$

- The idea is to first make it a Hessenberg Matrix, and then using Givens Rotation to remove the subdiagonal, giving us a upper triangular matrix.

The computational cost for the full QR algorithm is $O(n^3) + O(n^2 \cdot k)$, where k is the number of iterations (usually $O(n)$).

The algorithm generates $A^{(k)}$ satisfies:

$$A^{(k)} = \left(Q^{(k)}\right)^T \cdots \left(Q^{(0)}\right)^T AQ^{(0)} \cdots Q^{(k)}$$

which is similar to A, meaning that $A^{(k)}$ has the same eigenvalues of A.

However, we don't expect that this gives the eigenvalue decomposition of A. This is because:

1. the eigenvectors A are not necessarily orthogonal. (but $Q^{(i)}$ are orthogonal matrices), meaning that $X = Q^{(0)} \cdots Q^{(k)}$ is orthogonal.
2. the eigenvalue decomposition of A may not exist.

This QR algorithm converges to the Schur decomposition

Definition 5.3. For any matrix $A \in R^{n\times n}$, there exists $Q, S \in \mathbb{R}^{n\times n}$ such that

$$A = QSQ^T,$$

where

- $Q \in \mathbb{R}^{n\times n}$ is orthogonal, i.e. $Q^TQ = \mathbb{Q}^T = I$.
- $S \in \mathbb{R}^{n\times n}$ is a block upper triangular matrix, with 1 × 1 or 2 × 2 diagonal blocks. (i.e., there exists a partition of S:

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ & S_{22} & \cdots & S_{2p} \\ & & \ddots & \\ & & & S_{pp} \end{bmatrix} \qquad S_{ii} \text{ is either } 1\times 1 \text{ or } 2\times 2.$$

  Furthermore:

  - if $S_{ii} \in \mathbb{R}^{1\times 1}$, then it is an eigenvalue of A
  - if $S_{ii} \in \mathbb{R}^{2\times 2}$, then $S_{ii} = \begin{bmatrix} q & -b \\ b & a \end{bmatrix}$, with $a \pm bi$ being eigenvalues of A.
  - The blocks S_{ii} can be sorted s.t.

$$|\text{eig}(S_{11})| \geq |\text{eig}(S_{22})| \geq \ldots \geq |\text{eig}(S_{pp})|.$$

Theorem 5.4

Under mild assumption, the QR algorithm converges to the Schur decomposition, more precisely,

$$A^{(k)} \to S$$

and

$$Q^{(0)}Q^{(1)} \ldots Q^{(k)} \to Q.$$

We can get eigenvalues of A from S and the eigenvectors of A from Q.

> **Remark 5.5 —** Note that when A is symmetric, the Schur decomposition is the same as the eigenvalue decomposition, as S would by symmetric and upper triangular, i.e. S is diagonal. Because of this, all eigenvalues of A are real, since $S_{ii} \in \mathbb{R}^{1\times 1}$.

> **Remark 5.6 —** There is no direct formula for the Schur decomposition, since it is related to the eigenvalues of the matrix, which are the roots of a polynomial, which doesn't have a general closed form solution.

Example 5.7

Consider:

$$A = \begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{3}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Since A is non-symmetric, then we would not have a unitary eivenvalue decomposition.

The Schur decomposition of A is:

$$A = QSQ^T = \begin{bmatrix} 1 & & \\ & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & -1 & \\ 1 & 1 & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

So the eigenvalues of A are:

- $1 = \det(S_{22})$
- $1 \pm i = \det(S_{11})$

5.2 Case Studies C - Applications of Eigenvalue Decomposition

Case Study I: Find the roots of a polynomial $p(x)$. Let

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + x^n.$$

There are n solutions (roots) of $p(x) = 0$ in $\mathbb{C}$. We want to find all the roots of p.

Instead of using a root finding algorithm, we will use the eigenvalue decomposition to do so:

- Construct

$$A_p = \begin{bmatrix} 0 & & & & & a_0 \\ 1 & \ddots & & & & a_1 \\ & \ddots & \ddots & & & \vdots \\ & & \ddots & \ddots & & \vdots \\ & & & \ddots & 0 & a_{n-2} \\ & & & & 1 & a_{n-1} \end{bmatrix}.$$

Then:

$$\det(\lambda I - A_p = a_0 + a_1\lambda + a_2\lambda^2 + \ldots + a_{n-1}\lambda^{n-1} + \lambda^n.$$

This means that the eigenvalues of A_p are the roots of $p(x)$, meaning we can use the QR algorithm to find eigenvalues of A_p This is widely used in available software packages.

> **Remark 5.8 —** The reason why we might want to use the Schur decomposition to do so is because it is numerically stable due to the unitary transformation.

Case Study II: Ranking webpages. Once agian we have

$$\Pi = \frac{1-p}{n}1 + pA\Pi$$

$$(I - pA)\Pi = \frac{1-p}{n}1$$

$$(I - pA)\Pi = \frac{1-p}{n}11^T\Pi$$

$$(pA + \frac{1-p}{n}11^T)\Pi = \Pi.$$

This is in the form of an eigenvalue problem

6 April 24th, 2019

6.1 Case Studies 3

Let

- $W = [w_{ij}]$
- $D = \text{diag}(W_1)$

Define $L = D - W$. Computer the eigenvector x_{n-1} corresponding to λ_{n-1}, the 2nd smallest eigenvalue of L. Define:

$$S = \{i \mid x_{n-1}(i) > 0\}$$

and

$$S^C = \{i \mid x_{n-1}(i) \leq 0\}.$$

However, this method doesn't work well, as it tends to produce unbalanced groups, with one being huge and the other tiny. A better cut would be a "normalized cut".

To make the two groups have similar sizes, define the size of a group:

$$\text{Vol}(A) = \sum_{\substack{i\in A\\ j\in V}} w_{ij}.$$

Definition 6.1. A normalized cut of A and B is:

$$N_{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{Vol}(A)} + \frac{\text{cut}(A, B)}{\text{Vol}(B)}.$$

This is minimized if and only if $\text{cut}(A, B)$ is minimized and $\text{Vol}(A)$ and $\text{Vol}(B)$ are similar. However, this is not solvable, so we just solve:

$$\min_{S \subseteq V} N_{vut}(S, \overline{S}), \quad \text{where } \overline{S} = V \setminus S.$$

This again cannot be efficiently solved, so it can be solved approximately by an eigenvector/eigenvalue solver:

Define $z \in \mathbb{R}^m$ by:

$$z_i = \begin{cases} \frac{1}{\text{Vol}(S)} & \text{if } i \in S \\ -\frac{1}{\text{Vol}(\overline{S})} & \text{if } i \notin S \end{cases}.$$

Then

$$z^T L z = Z^T (D - W) z.$$

6.2 Singular Value Decomposition (SVD)

If $A \in \mathbb{R}^{n \times n}$ is symmetric, then

$$A = U \Lambda U^T,$$

where $U \in \mathbb{R}^{n \times n}$ satisfies $U^T U = U U^T = I$ and $\Lambda \in \mathbb{R}^{n \times n}$ is diagonal. This allows A to be it numerically stable.

However, this only only applies if A is a square and symmetric matrix. As such, we would like to extend the decomposition to any matrix, regardless of if they are square or symmetric. This is SVD.

Let $A \in R^{m \times n}$ with $m \geq n$ be an arbitrary real matrix (we can generalize this to $m < n$ and complex matrix). We will make use of the following facts:

- Since $A^T A \in R^{n \times n}$ is square, symmetric, and SPSD, there exists an eigenvector decomposition of $A^T A$.

  Let $(\lambda_i, v_i), i = 1, 2, \ldots, n$ be the eigenpairs of A^{TA}, with

  $$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0.$$

  So

  $$A^T A = V \Lambda V^T, V = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix}, \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ \lambda_n & & \end{bmatrix}.$$

- $A^T \in R^{m \times m}$ is also square, symmetric, and SPSD.

Theorem 6.2

Let $A \in R^{m\times n}$ and $B \in \mathbb{R}^{m\times m}$. Let $\lambda_1, \dots, \lambda_n$ be eigenvalues of BA. Then the eigenvalues of AB are

$$\lambda_1, \lambda_2, \dots, \lambda_n, \underbrace{0, \dots, 0}_{\text{m-n 0's}}.$$

By this theorem, the eigenvalues of

7 April 26th, 2019

7.1 SVD

Let $A \in \mathbb{R}^{m\times n}$ (with $m >= n$). Following from the previous lecture we have that the eigenvalues of AA^T are:

$$\lambda_1, \lambda_2, \dots, \lambda_n, 0, 0, \dots, 0$$

with corresponding eigenvectors:

$$u_1, u_2, \dots, u_n, u_{n+1}, \dots u_m.$$

Them:

$$AA^T = \begin{bmatrix} u_1 & u_2 & \dots & u_n & u_{n+1} & \dots \end{bmatrix} \begin{bmatrix} \lambda_1 \end{bmatrix} = U\Lambda U^T.$$

$$(AA^T)(Av_1) = A(A^TAv_i) = A(\lambda_i v_i) = \lambda_i(Av_i)$$
$$\implies Av_i \text{ is an eigenvector of } AA^T \text{ with eigenvalue } \lambda_i$$
$$\implies Av_i = \sigma_i u_i, \quad i = 1, 2, \dots, n \qquad (a).$$
$$(A^TA)(A^Tv_1) = A^T(AA^Tv_i) = A^T(\lambda_i v_i) = \lambda_i(A^Tv_i)$$
$$\implies A^Tu_i \text{ is an eigenvector of } A^TA \text{ with eigenvalue } \lambda_i$$
$$\implies A^Tv_i = \tilde{\sigma}_i u_i, \quad i = 1, 2, \dots, n \qquad (b).$$

Let's find σ_i and $tilde\sigma_i$:

$$(b) \implies v_i^TA^Tu_i = \tilde{\sigma}_i v_i^T v_i = \tilde{\sigma}_i.$$

$$(a) \implies u_i^TAv.$$

Index