

1 Lezione del 19-05-25

1.1 Calcolo di autovalori e autovettori

Vediamo quindi il problema di, data una matrice $A \in \mathbb{C}^{n \times n}$, trovare autocopie (λ, v) , con $\lambda \in \mathbb{C}$, $v \in \mathbb{C}^n \setminus \{0\}$ che verificano:

$$Av = \lambda v$$

Vedremo 2 metodi iterativi che approssimano 2 versioni di questo problema:

1. Calcolo di una singola autocoppia (λ, v) , ad esempio una coppia in cui λ è l'autovalore di modulo massimo;
2. Calcolo di tutte le autocopie (λ, v) , ovvero una autocoppia per ogni autovalore distinto. Non approfondiremo questo metodo per questioni di complessità.

Osserviamo che gli autovalori sono necessariamente n contate le loro molteplicità μ_i , in quanto rappresentano le soluzioni di un polinomio di grado n , e altro non possono fare per via del teorema fondamentale dell'algebra.

Osserviamo poi che gli autovettori sono definiti a meno di moltiplicazione per scalare, cioè sono sempre infiniti gli autovettori associati ad un singolo autovalore λ (tutti quelli nell'autospazio).

1.1.1 Metodo delle potenze

Vediamo il metodo 1) per ottenere una singola autocoppia, e in particolare quella di modulo (autovalore) massimo. Questo può essere utile per:

- Trovare il raggio spettrale di una matrice;
- Trovare la norma 2 di una matrice come:

$$\|A\|_2 = \sqrt{\rho(A^H A)}$$

Supponiamo che A sia diagonalizzabile. Questa è l'ipotesi che ci assicura che ogni autovalore ha un autovettore associato linearmente indipendente. Prendiamo quindi gli autovalori $\lambda_1, \dots, \lambda_n$ posti che soddisfano:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$$

dove il primo è in maggioranza stretta (altrimenti non si distinguerebbe un unico autovalore di modulo massimo).

Dato che A è diagonalizzabile, abbiamo dai fondamenti di algebra lineare che esiste una base di \mathbb{C}^n fatta da autovettori di A , ovvero $\exists v^{(1)}, \dots, v^{(n)} \in \mathbb{C}^n$ tali che:

$$Av^{(j)} = \lambda_j v^{(j)}, \quad j = 1, \dots, n$$

e per ogni $z \in \mathbb{C}^n$ vale che $\exists! c_1, \dots, c_n \in \mathbb{C}$ tali che:

$$z = \sum_{j=1}^n c_j v^{(j)} = c_1 v^{(1)} + \dots + c_n v^{(n)}$$

L'idea fondamentale del metodo è che, preso $z \in \mathbb{C}^n$ a caso, abbiamo:

$$Az = A(c_1 v^{(1)} + \dots + c_n v^{(n)}) = c_1 A v^{(1)} + \dots + c_n A v^{(n)} = c_1 \lambda_1 v^{(1)} + \dots + c_n \lambda_n v^{(n)}$$

Continuando ad iterare si ha:

$$\begin{aligned} A^2 z &= A(Az) = A(c_1 \lambda_1 v^{(1)} + \dots + c_n \lambda_n v^{(n)}) \\ &= c_1 \lambda_1 A v^{(1)} + \dots + c_n \lambda_n A v^{(n)} = c_1 \lambda_1^2 v^{(1)} + \dots + c_n \lambda_n^2 v^{(n)} \end{aligned}$$

e quindi continuando ad iterare si ha che:

$$A^k z = c_1 \lambda_1^k v^{(1)} + \dots + c_n \lambda_n^k v^{(n)}$$

Il vantaggio è che l'autovalore di modulo massimo compare chiaramente con contribuzioni sempre più grandi, in quanto:

$$= \lambda_1^k \left(c_1 v^{(1)} + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v^{(2)} + \dots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v^{(n)} \right) \sim \lambda_1^k, \quad k \rightarrow +\infty$$

Più formalmente, abbiamo che vale il teorema:

Teorema 1.1: Convergenza del metodo delle potenze

Sia considerata la successione:

$$\begin{cases} z^{(0)} = z \\ z^{(k+1)} = A z^{(k)} = A^{k+1} z^{(0)} = A^{k+1} z \end{cases}$$

con $A \in \mathbb{C}^{n \times n}$ hermitiana ($A = A^H$), con autovalori:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| > 0$$

e sia $h \in \{1, \dots, n\}$ tale per cui $v_h^{(1)} \neq 0$. Se $z^{(0)} \in \mathbb{C}^n$ è tale che:

$$[V^{(1)}]^H z^{(0)} \neq 0$$

allora la successione è tale che:

$$\lim_{k \rightarrow +\infty} \frac{z^{(k)}}{z_h^{(k)}} = \tilde{v}^{(1)}$$

multiplo scalare di $v^{(1)}$ (in particolare $\tilde{v}^{(1)} = \frac{v^{(1)}}{v_h^{(1)}}$). Inoltre, per l'autovalore vale:

$$\lim_{k \rightarrow +\infty} \frac{[z^{(k)}]^H A z^{(k)}}{[z^{(k)}]^H z^{(k)}} = \lambda_1$$

Osserviamo che, dato $x \in \mathbb{C}^n$ e considerato il *quoziente di Rayleigh*:

$$R(x) = \frac{x^H A x}{x^H x}$$

Se x verifica $Ax = \lambda x$ (cioè è autovettore) allora:

$$R(x) = \frac{x^H Ax}{x^H x} = \frac{x^H \lambda x}{x^H x} = \lambda \frac{x^H x}{x^H x} = \lambda$$

Notiamo la motivazione dell'assunto di matrice hermitiana. Di base, per il teorema spettrale si ha che:

$$A = A^H \implies A \text{ diagonalizzabile}$$

e in particolare si può scegliere una base ortonormale di autovettori, cioè trovare $v^{(1)}, \dots, v^{(n)} \in \mathbb{C}^n$ tali che:

$$Av^{(j)} = \lambda_j v^{(j)}, \quad [v^{(i)}]^H v^{(j)} = \langle v^{(i)}, v^{(j)} \rangle = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

cioè equivalentemente:

$$A = VDV^H$$

con D diagonalizzabile e V normale data dagli autovettori in colonna.

Dimostriamo quindi il teorema. Dato che $A = A^H$ hermitiana si ha che:

$$z^{(0)} = \sum_{j=1}^n c_j v^{(j)}$$

ed inoltre abbiamo:

$$0 \neq [v^{(1)}]^H z^{(0)} = \sum_{j=1}^n c_j [v^{(1)}]^H v^{(j)} = c_1 [v^{(1)}]^H v^{(1)} = c_1$$

e quindi $c_1 \neq 0$.

Dimostriamo allora le due tesi, su autovettore (1) a autovalore massimo (2):

1. Come fatto in precedenza prendiamo:

$$\begin{aligned} z^{(k)} &= A^k z^{(0)} = \lambda_1^k \left(c_1 v^{(1)} + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k v^{(2)} + \dots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^k v^{(n)} \right) \\ &= \lambda_1^k \left(c_1 v^{(1)} + \sum_{j=2}^n \left(\frac{\lambda_j}{\lambda_1} \right)^k v^{(j)} c_j \right) \end{aligned}$$

dove vorremo dimostrare che il termine a destra scompare per $k \rightarrow +\infty$.

Prendiamo quindi il limite:

$$\begin{aligned} \lim_{k \rightarrow +\infty} \frac{\lambda_1^k \left(c_1 v^{(1)} + \sum_{j=2}^n \left(\frac{\lambda_j}{\lambda_1} \right)^k v^{(j)} c_j \right)}{\lambda_1^k \left(c_1 v_h^{(1)} + \sum_{j=2}^n \left(\frac{\lambda_j}{\lambda_1} \right)^k v_h^{(j)} c_j \right)} \\ \lim_{k \rightarrow +\infty} \frac{c_1 v^{(1)} + \sum_{j=2}^n \left(\frac{\lambda_j}{\lambda_1} \right)^k v^{(j)} c_j}{c_1 v_h^{(1)} + \sum_{j=2}^n \left(\frac{\lambda_j}{\lambda_1} \right)^k v_h^{(j)} c_j} = \lim_{n \rightarrow +\infty} \frac{c_1 v^{(1)}}{c_1 v_h^{(1)}} = \frac{v^{(1)}}{v_h^{(1)}} \end{aligned}$$

che è la prima tesi.

2. Prendiamo quindi:

$$\lim_{k \rightarrow +\infty} \frac{[z^{(k)}]^H A z^{(k)}}{[z^{(k)}]^H z^{(k)}} = \lim_{k \rightarrow +\infty} \frac{\left[\frac{z^{(k)}}{z_h^{(k)}}\right]^H A \frac{z^{(k)}}{z_h^{(k)}}}{\left[\frac{z^{(k)}}{z_h^{(k)}}\right]^H \frac{z^{(k)}}{z_h^{(k)}}} = \frac{\left[\frac{v^{(1)}}{v_h^{(1)}}\right]^H A \frac{v^{(1)}}{v_h^{(1)}}}{\left[\frac{v^{(1)}}{v_h^{(1)}}\right]^H \frac{v^{(1)}}{v_h^{(1)}}}$$

che moltiplicando sopra e sotto per $(v_h^{(1)})^2$ dà:

$$= \frac{[v^{(1)}]^H A v^{(1)}}{[v^{(1)}]^H v^{(1)}} = \lambda_1$$

da cui la seconda tesi. □

Osserviamo che h è stato usato perché dividiamo per $z_h^{(k)}$ che per k abbastanza grandi è $\neq 0$, in quanto vicino a $v_h^{(1)}$.

Osserviamo poi che nell'implementazione pratica il metodo delle potenze $z^{(k)}$ viene diviso per la sua norma, così da ottenere solo vettori di norma 1 ed evitare problemi di overflow o underflow.

1.1.2 Criterio di stop

Vediamo la condizione di stop del metodo delle potenze. In genere si usa:

$$|R(z^{(k)}) - R(z^{(k-1)})| < \epsilon$$

con ad esempio $\epsilon = 10^{-8}$.

Testiamo gli autovalori con $R()$ e non gli autovettori con:

$$|z^{(k)} - z^{(k+1)}|_2 < \epsilon$$

in quanto non sappiamo verso quale multiplo scalare degli autovettori ci stiamo dirigendo, per cui la differenza ad ogni passaggio potrebbe essere molto grande.

1.1.3 Implementazione MATLAB del metodo delle potenze

Vediamo quindi, da quello che abbiamo detto, come si può implementare una funzione per il calcolo dell'autocoppia massima in MATLAB:

```

1 function [z0, e] = max_eigen(A, z0, k, tol)
2     if nargin < 4
3         tol = 0.01;
4     end
5
6     if nargin < 3
7         k = 100;
8     end
9
10    % eigenvector thru power method
11    for i = 1:k
12        z1 = A * z0;
13        z1 = z1 / norm(z1);
14
15        if norm(z1 - z0) < tol
16            break;

```

```

17         end
18
19         z0 = z1;
20     end
21
22     % eigenvalue
23     e = norm(A * z0);
24 end

```

Il costo è quello di una norma ($O(n)$) e di un prodotto matrice vettore ($O(n^2)$) ad ogni iterazione, cioè $O(k \cdot n^2)$.

1.1.4 Commenti sull'ipotesi di convergenza

Possiamo fare un commento sulle ipotesi del teorema 22.1.

- Non è strettamente necessario che $A = A^H$ hermitiana, in quanto si può dimostrare che il metodo funziona anche con A solo diagonalizzabile;
- L'assunzione $[v^{(1)}]^H z^{(0)} \neq 0$ è virtualmente vera (o meglio vera con probabilità 1) per qualsiasi $z^{(0)} \in \mathbb{C}^n$ scelto a caso. Inoltre, nel caso di calcolo a macchina si ha che gli errori di arrotondamento stessi aiutano ad uscire dall'ortogonalità, anche se si parte da tale condizione.
- L'assunzione $|\lambda_1| > |\lambda_2|$ non è invece rinunciabile, in quanto si possono costruire esempi di successioni non convergenti nel caso di più autovalori dominanti.

Prendiamo l'esempio:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

con 2 autovalori di modulo 1. Prendiamo:

$$z^{(0)} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Si trova che l'aggiornamento scambia semplicemente le componenti:

$$Az^{(0)} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

e così via all'infinito.

Gli autovettori, calcolati a mano, sono semplicemente:

$$v^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad v^{(2)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Osserviamo quindi che per eseguire il metodo delle potenze, serve soltanto essere in grado di valutare il prodotto matrice vettore Az . Questo può essere utile nel caso sopra citato, quando si ha una matrice definita come risultato di operazioni aritmetiche, magari costose.

Ad esempio, se si vuole stimare:

$$|A|_2 = \sqrt{\rho(A^H A)}$$

si può pensare di applicare il metodo delle potenze su $A^H A$, senza nemmeno dover calcolare $A^H A$ ma applicando direttamente i prodotti matrice vettore $A^H Az$.

1.2 Metodi per più autovalori

1.2.1 Autovalori minimi

Supponiamo di essere interessati all'autocoppia $(\lambda_n, v^{(n)})$, associata all'autovalore λ_n di modulo minimo, e supponiamo:

$$|\lambda_1| \geq \dots \geq |\lambda_{n-2}| \geq |\lambda_{n-1}| > |\lambda_n|$$

L'idea è di applicare il metodo delle potenze ad A^{-1} , infatti A^{-1} ha autovalori:

$$\left| \frac{1}{\lambda_n} \right| > \left| \frac{1}{\lambda_{n-1}} \right| \geq \left| \frac{1}{\lambda_{n-2}} \right| \geq \dots \geq \left| \frac{1}{\lambda_1} \right|$$

ed inoltre ha come autovettori $v^{(1)}, \dots, v^{(n)}$. Quindi le successioni generate dal metodo delle potenze su A^{-1} convergono all'autocoppia:

$$\left(\frac{1}{\lambda_n}, v^{(n)} \right)$$

Chiamiamo questo metodo anche **metodo delle potenze inverse**.

Dove chiaramente non vale la pena di calcolare l'inversa ma piuttosto conviene risolvere il sistema:

$$Ay^{(k)} = z^{(k-1)}$$

ad esempio sfruttando l'eliminazione di Gauss con pivoting. Per rendere tutto più efficiente, conviene anzi precalcolare una fattorizzazione LU di A prima di iniziare il metodo.

In questo caso avremo:

$$A = \Pi LU$$

cioè si fa una fattorizzazione LU con pivoting dato dalla matrice di permutazione Π , così che ad ogni passaggio il prodotto $A^{-1}z$ diventa equivalente a risolvere un sistema lineare triangolare. In questo caso il costo diventa $O(n^3)$ per la fattorizzazione più $O(k \cdot n^2)$ per la risoluzione dei sistemi triangolari.

1.2.2 Implementazione MATLAB del metodo delle potenze inverse

Vediamo quindi come si può implementare una funzione per il calcolo dell'autocoppia minima in MATLAB:

```

1 function [z0, e] = min_eigen(A, z0, k, tol)
2     if nargin < 4
3         tol = 0.01;
4     end
5
6     if nargin < 3
7         k = 100;
8     end
9
10    % eigenvector thru power metod
11    for i = 1:k
12        z1 = A \ z0;
13        z1 = z1 / norm(z1);
14
15        if norm(z1 - z0) < tol
16            break;

```

```

17         end
18
19         z0 = z1;
20     end
21
22     % eigenvalue
23     e = norm(A * z0);
24 end

```

1.2.3 Autovalori intermedi

Vediamo quindi come modificare il metodo delle potenze per trovare un autovalore intermedio, cioè preso $\sigma \in \mathbb{C}$ scalare arbitrario calcolare l'autocoppia (λ, v) con λ più vicino a σ .

In questo caso si applica il metodo delle potenze alla matrice:

$$A \sim (A - \sigma \cdot I)^{-1}$$

che ha gli stessi autovettori di A e come autovalori ha quantità della forma:

$$\lambda'_j = \frac{1}{\lambda_j - \sigma}$$

Questo si dimostra prendendo assumendo (λ, v) autocoppia, quindi $Av = \lambda v$, e calcolando $(A - \sigma I)v$:

$$(A - \sigma I)x = Ax - \sigma Ix = Ax - \sigma x = \lambda x - \sigma x = (\lambda - \sigma)x$$

guardando agli estremi, e invertendo, si ottiene:

In questo caso l'autovalore di modulo più grande diventa quello più vicino a σ , per cui basta applicare il metodo delle potenze.