Algoritmo di estrazione della forma canonica di Jordan

1 Algoritmo di estrazione della forma canonica di Jordan

Lo scopo di questo notebook è illustrare un breve e semplice algoritmo di estrazione della forma canonica di Jordan di una matrice $A \in M(n, \mathbb{C})$ qualsiasi. Si ricorda che la forma canonica di Jordan è un invariante completo per similitudine di una matrice e che è una matrice a blocchi diagonale composta da più blocchi, detti per l'appunto di Jordan, della seguente forma:

$$J_{\lambda,m} = \begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix} \in M(m, \mathbb{C}).$$

Innanzitutto si sceglie un $n \in \mathbb{N}^+$ che rappresenta la taglia della matrice quadrata data in esame all'algoritmo e si crea la matrice $I = I_n$, ossia la matrice identità di taglia n.

[1]:
```
%display latex
from IPython.display import display, Markdown, Latex

n = 10
I = matrix.identity(n)
```

Si definisce adesso una funzione jordan(A) che prende in ingresso una matrice $A \in M(n, \mathbb{C})$ senza restituire alcun risultato. Tale funzione illustra passo passo in output l'algoritmo di estrazione della forma canonica di Jordan della matrice A nei seguenti passi:

- Calcola il polinomio caratteristico $p_A(x)$ di A e lo fattorizza;
- Ne deduce lo spettro $\mathrm{sp}(A)$ e si riduce a studiare i blocchi relativi a ciascun autovalore;
- Per ogni autovalore λ, calcola le dimensioni dei kernel delle potenze di $B = A - \lambda I$ fino a che non viene raggiunta la molteplicità algebrica di λ;
- Infine, per l'autovalore λ, calcola il numero di blocchi b_i di taglia i secondo la formula $b_i = 2 \dim \ker B^i - \dim \ker B^{i-1} - \dim \ker B^{i+1}$;
- Compone tutti i blocchi in una matrice a blocchi diagonale, ossia la forma canonica di Jordan.

[2]:
```
def jordan(A):
    display(Latex(r"Innanzitutto, si calcola il polinomio caratteristico di␣
↪$A$, che risulta essere $p_A(x) = " + latex(factor(charpoly(A)(SR('x')))) +␣
↪"$."))
    eig = set(A.eigenvalues())
```

```python
    sp = ", ".join(map(latex, eig))

    display(Latex(r"Allora, calcolandone le radici, si ricava $\operatorname{sp}(A) = \{" + sp + r"\}$."))
    display(Markdown("---"))

    Js = []

    for i, t in enumerate(eig, start=1):
        B = A - t*I
        r = rank(B)
        k = n - r

        display(Latex(f"({i}) " + r"Consideriamo $\lambda = " + latex(t) + r"$ e $B=A-\lambda I=" + latex(B) + r".$"))
        display(Latex(r"$B$ ha rango $" + str(r) + r"$ e quindi $\mu_g(\lambda) = " + str(k) + r"$."))

        if k == 1:
            display(Latex(r"Allora a $\lambda$ sarà dedicato esattamente un blocco nella forma canonica di Jordan."))
        else:
            display(Latex(r"Allora a $\lambda$ saranno dedicati esattamente $" + str(k) + "$ blocchi nella forma canonica di Jordan."))

        K = [k]
        k_1 = k
        k_2 = n-rank(B^2)

        i = 3
        while k_1 != k_2:
            K.append(k_2)
            k_1 = k_2
            k_2 = n-rank(B^i)
            i += 1

        display(Latex(r"Si calcolano adesso le dimensioni dei kernel fino a quando non viene raggiunta la molteplicità algebrica $\mu_a(\lambda) = " + str(K[-1]) + "$:"))

        for i, k in enumerate(K, start=1):
            if i == 1:
                if i == len(K):
                    display(Latex(r"    • $\dim \ker B = " + str(k) + "$."))
                else:
                    display(Latex(r"    • $\dim \ker B = " + str(k) + "$;"))
```

```
            else:
                if i == len(K):
                    display(Latex(r"    • $\dim \ker B^" + str(i) + r" = " + str(k) + "$."))
                else:
                    display(Latex(r"    • $\dim \ker B^" + str(i) + r" = " + str(k) + "$;"))

        display(Latex(r"Pertanto a $\lambda$ sono assegnati i seguenti blocchi (dove $b_n$ indica il numero di blocchi di taglia $n$):"))

        b = {}

        for i, k in enumerate(K, start=1):
            if i == len(K):
                if i == 2:
                    b[i] = k - K[0]
                    display(Latex(r"    • $b_2 = \dim \ker B^2 - \dim \ker B = " + str(b[i]) + "$."))
                elif i == 1:
                    b[i] = k
                    display(Latex(r"    • $b_1 = \dim \ker B = " + str(b[i]) + "$."))
                else:
                    b[i] = k - K[i-2]
                    display(Latex(r"    • $b_" + str(i) + r" = \dim \ker B^" + str(i) + r" - \dim \ker B^" + str(i-1) + " = " + str(b[i]) + "$."))
            elif i == 1:
                b[i] = 2*k - K[1]
                display(Latex(r"    • $b_1 = 2 \dim \ker B - \dim \ker B^2 = " + str(b[i]) + "$;"))
            else:
                if i != 2:
                    b[i] = 2*k - K[i-2] - K[i]
                    display(Latex(r"    • $b_" + str(i) + r" = 2 \dim \ker B^" + str(i) + r" - \dim \ker B^" + str(i-1) + r" - \dim \ker B^" + str(i+1) + " = " + str(b[i]) + "$;"))
                else:
                    b[i] = 2*k - K[0] - K[2]
                    display(Latex(r"    • $b_" + str(i) + r" = 2 \dim \ker B^2 - \dim \ker B - \dim \ker B^3 = " + str(b[i]) + "$;"))

        JBs = [jordan_block(t, N) for N, j in b.items() for _ in range(j)]
        Js.extend(JBs)
```

```
        display(Latex(r"Allora l'insieme di tutti i blocchi relativi a $\lambda$ sarà rappresentato dalla seguente matrice: $J_{\lambda} = " + latex(block_diagonal_matrix(JBs)) + "$."))
        display(Markdown("---"))

    JNF = block_diagonal_matrix(Js)

    display(Latex(r"La forma canonica di Jordan di $A$ sarà allora $J = " + latex(JNF) + "$."))
```

Si sceglie adesso una matrice $A \in M(n, \mathbb{C})$ di cui si vuole calcolare la forma canonica di Jordan.

[3]:
```
# A deve avere esattamente n^2 elementi, essendo n x n...
A = matrix(SR, n, [1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 2, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3])
A
```

[3]:
$$\left(\begin{array}{cccccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3
\end{array}\right)$$

Si calcola infine la forma canonica di Jordan della matrice A mediante la funzione jordan.

[4]:
```
jordan(A)
```

Innanzitutto, si calcola il polinomio caratteristico di A, che risulta essere $p_A(x) = (x-1)^4 (x-2)^3 (x-3)^3$.

Allora, calcolandone le radici, si ricava $\mathrm{sp}(A) = \{1, 2, 3\}$.

---

(1) Consideriamo $\lambda = 1$ e $B = A - \lambda I = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$.

B ha rango 9 e quindi $\mu_g(\lambda) = 1$.

Allora a λ sarà dedicato esattamente un blocco nella forma canonica di Jordan.

Si calcolano adesso le dimensioni dei kernel fino a quando non viene raggiunta la molteplicità algebrica $\mu_a(\lambda) = 4$:

- $\dim \ker B = 1$;
- $\dim \ker B^2 = 2$;
- $\dim \ker B^3 = 3$;
- $\dim \ker B^4 = 4$.

Pertanto a λ sono assegnati i seguenti blocchi (dove b_n indica il numero di blocchi di taglia n):

- $b_1 = 2 \dim \ker B - \dim \ker B^2 = 0$;
- $b_2 = 2 \dim \ker B^2 - \dim \ker B - \dim \ker B^3 = 0$;
- $b_3 = 2 \dim \ker B^3 - \dim \ker B^2 - \dim \ker B^4 = 0$;
- $b_4 = \dim \ker B^4 - \dim \ker B^3 = 1$.

Allora l'insieme di tutti i blocchi relativi a λ sarà rappresentato dalla seguente matrice: $J_\lambda = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

---

(2) Consideriamo $\lambda = 2$ e $B = A - \lambda I = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$.

B ha rango 8 e quindi $\mu_g(\lambda) = 2$.

Allora a λ saranno dedicati esattamente 2 blocchi nella forma canonica di Jordan.

Si calcolano adesso le dimensioni dei kernel fino a quando non viene raggiunta la molteplicità algebrica $\mu_a(\lambda) = 3$:

- $\dim \ker B = 2$;
- $\dim \ker B^2 = 3$.

Pertanto a λ sono assegnati i seguenti blocchi (dove b_n indica il numero di blocchi di taglia n):

- $b_1 = 2 \dim \ker B - \dim \ker B^2 = 1$;
- $b_2 = \dim \ker B^2 - \dim \ker B = 1$.

Allora l'insieme di tutti i blocchi relativi a λ sarà rappresentato dalla seguente matrice: $J_\lambda = \left(\begin{array}{c|cc} 2 & 0 & 0 \\ \hline 0 & 2 & 1 \\ 0 & 0 & 2 \end{array}\right)$.

---

(3) Consideriamo $\lambda = 3$ e $B = A - \lambda I = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$.

B ha rango 9 e quindi $\mu_g(\lambda) = 1$.

Allora a λ sarà dedicato esattamente un blocco nella forma canonica di Jordan.

Si calcolano adesso le dimensioni dei kernel fino a quando non viene raggiunta la molteplicità algebrica $\mu_a(\lambda) = 3$:

- $\dim \ker B = 1$;
- $\dim \ker B^2 = 2$;
- $\dim \ker B^3 = 3$.

Pertanto a λ sono assegnati i seguenti blocchi (dove b_n indica il numero di blocchi di taglia n):

- $b_1 = 2 \dim \ker B - \dim \ker B^2 = 0$;
- $b_2 = 2 \dim \ker B^2 - \dim \ker B - \dim \ker B^3 = 0$;
- $b_3 = \dim \ker B^3 - \dim \ker B^2 = 1$.

Allora l'insieme di tutti i blocchi relativi a λ sarà rappresentato dalla seguente matrice: $J_\lambda = \begin{pmatrix} 3 & 1 & 0 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{pmatrix}$.

La forma canonica di Jordan di A sarà allora $J = \left(\begin{array}{cccc|c|cc|ccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{array}\right)$.