Análisis Matemático

Clase 4

Autovalores y autovectores

Motivación

La idea de descomponer en autovalores (avas) y autovectores (aves), es poder recuperar propiedades "universales" de la matriz.

Por ejemplo, la descomposición de un número en números primos. No importa como represente el número 12 (en que base) siempre vale que 12 = 2 x 3 x 3 , y se pueden extraer propiedades

Definición

Def: Sea $A \in \mathbb{C}^{n\times n}$, diremos que $\lambda \in \mathbb{C}$ es un autovalor de A y $x \in \mathbb{C}^n$ es el autovector asociado a λ si

$$Ax = \lambda x$$

Interpretación geométrica: A cada vector que se encuentre en la dirección de x, la transformación $T(x) = Ax$ lo contrae (o expande) por un factor λ

Algunas aplicaciones

- Análisis de estabilidad (sistemas de control, ing. mecánica y civil)
- Cálculo de resonancias del sistema
- Análisis del comportamiento de ecuaciones diferenciales
- Reducción de dimensiones (ej PCA)
- PageRank

¿Cómo encuentro los autovalores?

Teorema: Dada $A \in \mathbb{C}^{n \times n}$, λ es un autovalor de A sii. λ es un cero del polinomio característico de A:

$$p(\lambda) := det(A - \lambda I)$$

Obs: Puede ocurrir que algún λ sea raíz múltiple de $p(\lambda)$. Se llama multiplicidad algebraica (m_A), a la cantidad de veces que λ aparece como raíz.

¿Hay alguna forma más fácil?

¡Si! En python:

```
s,V = np.linalg.eig(A)
```

s: es el vector de autovalores, ordenados de mayor a menor ($\lambda_1 \geq \ldots \geq \lambda_n$)

V: Son los autovectores asociados a cada λ_i, escalados para que tengan norma 1.

Propiedades de los autovectores

Def: Dada $A \in \mathbb{R}^{n\times n}$, el conjunto de todos los autovectores asociados al autovalor λ generan un subespacio de $\mathbb{R}^n$ (autoespacio de A respecto de λ). El conjunto de todos los autovectores de A se llama autoespectro.

Si λ es un autovalor de $A \in \mathbb{R}^{n\times n}$, el autoespacio E_λ asociado es la solución al sistema homogéneo $(A - \lambda I)x = 0$.

Más propiedades de los autovectores

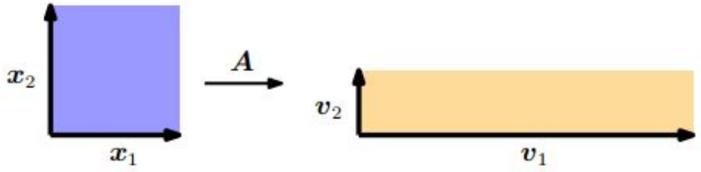
Def: Se llama multiplicidad geométrica (m_g) del autovalor λ a la cantidad de autovectores l.i. asociados a λ. **Obs:** $1 \leq m_g \leq m_a$.

Teorema: Los autovectores $x_1 \dots x_n$ de una matriz $A \in \mathbb{R}^{n \times n}$ con n autovalores distintos $\lambda_1, \dots, \lambda_n$ son l.i. (i.e. forman una base de $\mathbb{R}^n$)

Teorema: Si $A \in \mathbb{R}^{n \times n}$ es simétrica, entonces existe una base ortonormal correspondiente a un espacio V, formado por autovectores de A, todos los $\lambda_i \in \mathbb{R}$

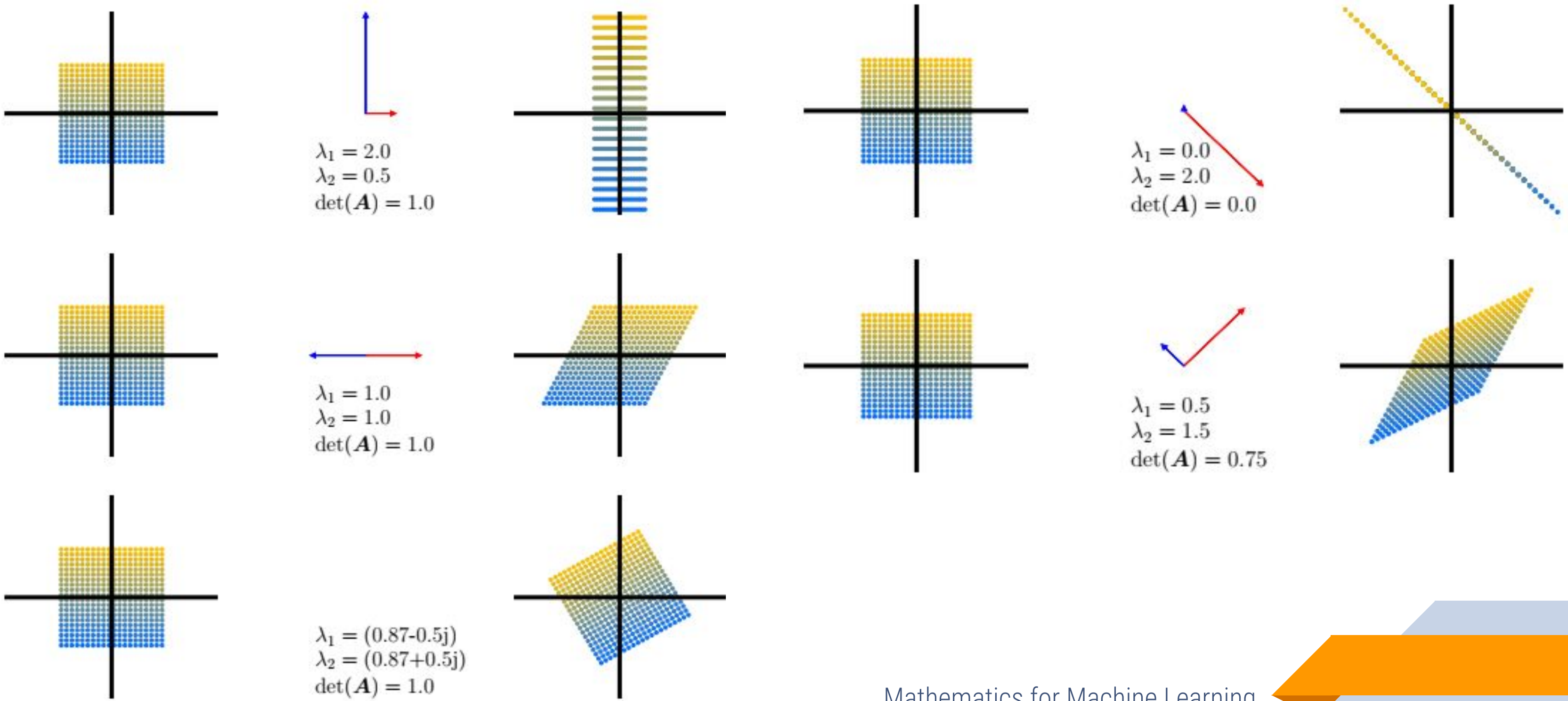
Propiedades asociadas

Teorema: $det(A) = \prod_i \lambda_i$. (área)



Teorema: $Tr(A) = \sum_i \lambda_i$ (perímetro)

Interpretación gráfica



Diagonalización de una matriz

Def: Una matriz $A \in \mathbb{C}^{n \times n}$ es digonalizable si es similar a una matriz diagonal D. Es decir si existe $Q \in \mathbb{C}^{n \times n}$ invertible tal que $D = Q^{-1} AP$

Teorema: Una matriz cuadrada $A \in \mathbb{C}^{n \times n}$ puede ser factorizada como

$$A = QDQ^{-1},$$

con $Q \in \mathbb{C}^{n \times n}$ y D diagonal tal que $D_{ii} = \lambda_i$, **sii**. los autovectores de A forman una base de $\mathbb{C}^{n \times n}$.

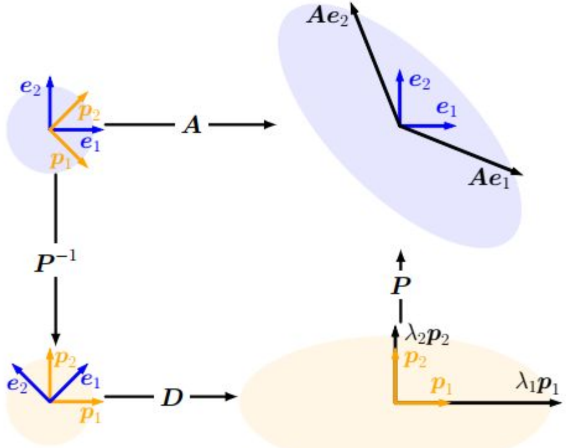
Diagonalización (continuación)

Obs 1: las columnas de Q se van a corresponder con los autovectores de A.

Obs 2: Toda matriz hermítica puede ser diagonalizable.

Obs 3: Más aún, si A es hermítica, las columnas forman un BON y

$$A = QDQ^T$$



Mathematics for Machine Learning

Descomposición en valores singulares

¿Qué pasa si las matrices no son cuadradas?

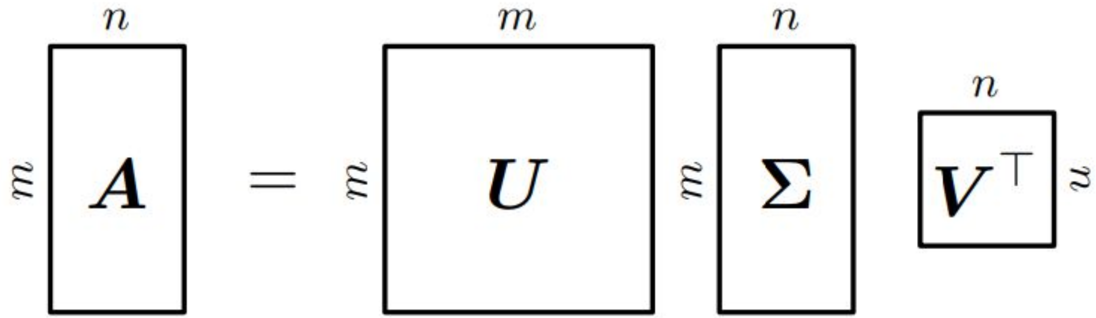
Teorema: Dada $A \in \mathbb{C}^{m\times n}$, siempre se puede obtener una matriz hermítica, semidefinida positiva $S \in \mathbb{C}^{n\times n}$ como:

$$S := A^H A$$

Obs: $\tilde{S} \in \mathbb{C}^{m\times m}$ definida como $\tilde{S} := AA^H$ también resulta hermítica y semidefinida positiva

Descomposición en valores singulares

Teorema: Sea $A \in \mathbb{C}^{m \times n}$ una matriz rectangular de rango $r \in [0, \min(m, n)]$. La descomposición en valores singulares (SVD) de A es una descomposición de la forma:



Descomposición en valores singulares (continuación)

donde $U \in \mathbb{C}^{m\times m}$ es la matriz ortogonal con columnas $u_i,\ i = 1, \ldots m$, $V \in \mathbb{C}^{n\times n}$ con columnas $v_i,\ i = 1, \ldots, n$, y $\Sigma \in \mathbb{C}^{m\times n}$ es una matriz "diagonal"($\Sigma_{ii} = \sigma_i \leq 0$, y $\Sigma_{ij} = 0, i \neq j$)

Las columnas de U son los autovectores de AA^H

Las columnas de V son los autovectores de $A^H A$

$\sigma_i = \sqrt{\lambda_i}$, y λ_i son los autovalores distintos de cero de $A^H A$ y AA^H.

SVD Truncada

Sea $A \in \mathbb{C}^{m \times n}$

- Si $m > n$, A se puede descomponer como

$$A = U_m \Sigma_m V_m^T$$

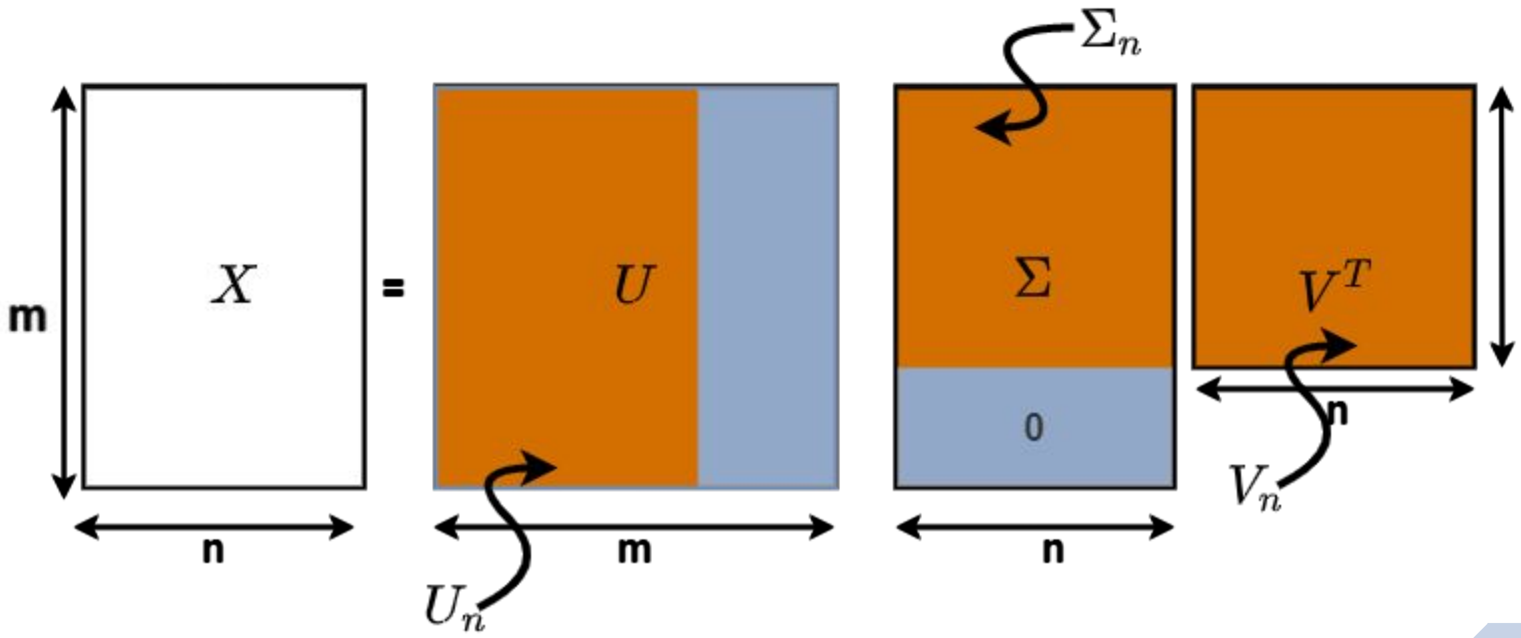
- Si $m < n$ A se puede descomponer como

$$A = U_n \Sigma_n V_n^T$$

- En general, si A tiene r autovalores distintos de cero vale que

$$A = U_r \Sigma_r V_r^T$$

SVD Truncada



Python

Para calcular la SVD:

```
U,s,Vt = np.linalg.svd(A)
```

donde $U \in \mathbb{C}^{\mathrm{m}\times\mathrm{m}}$, $s = [\sigma_1, \ldots, \sigma_r]$, $V \in \mathbb{C}^{n\times n}$.

```
U,s,Vt = np.linalg.svd(A,full_matrices=False)
```

devuelve la SVD reducida.

Pseudoinversa de Moore-Penrose

Sean $A \in \mathbb{C}^{m\times n}$, y el sistema lineal $Ax = y$. La inversa de A no está definida, pero habíamos visto que una posible solución era

$$\hat{x} = \left(A^T A\right)^{-1} A^T y.$$

¿Qué ocurre si $\left(A^T A\right)$ no es invertible? Se define la pseudoinversa de Moore-Penrose como

$$A^\dagger = V_r \Sigma_r^{-1} U_r^T$$

y $\hat{x} = A^\dagger y$ es la solución de norma mínima (euclídea) $(m > n)$, o la solución de mínimo error (distancia euclídea) $(m < n)$.

Para calcular la pseudoinversa

```
Ainv = np.linalg.pinv(A)
```

Caso de uso:

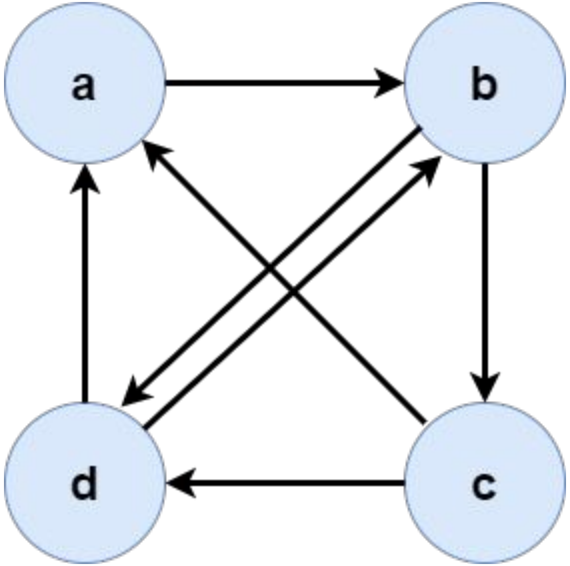
Algoritmo de PageRank

Idea del algoritmo

El ranking de de cada página depende de la cantidad (y calidad) de links que apuntan a ella.

Un rankings alto se puede dar por muchos links de poco peso, o pocos de mucho peso.

Algoritmo básico (simplificado)

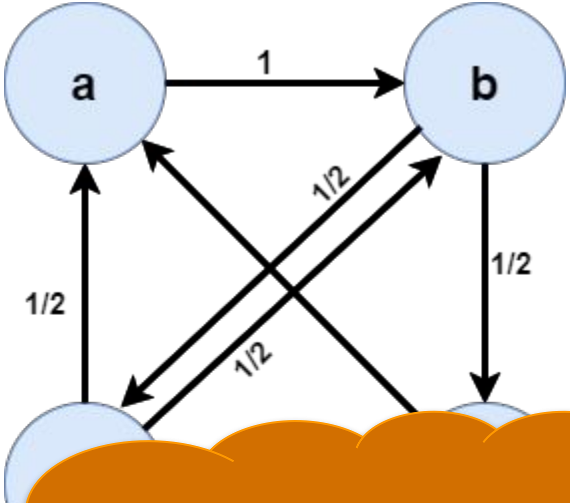


Sea u una página, $F(u)$ son las páginas a las que apunta , u y $B(u)$ las páginas que apuntan a u. Sea u $N(u) = |F(u)|$ (cantidad de páginas a las que apunta u). Luego

$$R(u) = c \sum_{v \in B(u)} \frac{R(v)}{N_v}$$

Algoritmo básico (simplificado)

A puede verse como la matriz de adyacencia del grafo



$$R(a) = \left(\frac{R(c)}{2} + \frac{R(d)}{2}\right) \qquad R(c) = \left(\frac{R(b)}{2}\right)$$

$$R(b) = \left(\frac{R(a)}{1} + \frac{R(d)}{2}\right) \qquad R(d) = \left(\frac{R(b)}{2} + \frac{R(c)}{2}\right)$$

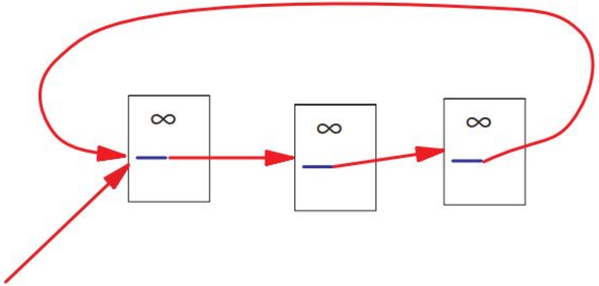
$$\begin{bmatrix} R(a) \\ R(b) \\ R(c) \\ R(d) \end{bmatrix} = \overbrace{\begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix}}^{A} \begin{bmatrix} R(a) \\ R(b) \\ R(c) \\ R(d) \end{bmatrix}$$

$$R = [0.217, 0.348, 0.174, 0.261]^T$$

R es el autovector de A asociado a $\lambda = 1$

Algoritmo completo

R es el autovector de (A+E) asociado a $\lambda = 1$



Para evitar los "sumideros" se modifica un poco el algoritmo:

$$R'(u) = c\sum_{v\in B(u)} \frac{R'(v)}{N_u} + cE(u)$$

Matricialmente se puede escribir como $R = AR + E = (A + \tilde{E})R,$

$\tilde{E}$ resulta de repetir el vector E hasta conseguir una matriz cuadrada.

Algoritmo

$$
\begin{aligned}
R_0 &\leftarrow S \\
\text{loop}: & \\
R_{i+1} &\leftarrow AR_i \\
d &\leftarrow ||R_i||_1 - ||R_{i+1}||_1 \\
R_{i+1} &\leftarrow R_{i+1} + dE \\
\delta &\leftarrow ||R_{i+1} - R_i||_1 \\
\text{while } \delta > \epsilon &
\end{aligned}
$$

S es cualquier vector sobre el conjunto de páginas. *d* se usa para acelerar la tasa de convergencia del algoritmo.

Bibliografía

Bibliografía

- Mathematics for Machine Learning
- Eigenvalues and Eigenvectors
- Video de aplicaciones
- Análisis de estabilidad
- Paper de PageRank