Principal-Component Analysis

- **Principal-Component Analysis**, o **PCA**, è una tecnica che prende un dataset relativo ad un insieme di tuple in uno spazio ad alta dimensione e trova le direzioni lungo il quale le tuple si allineano meglio.

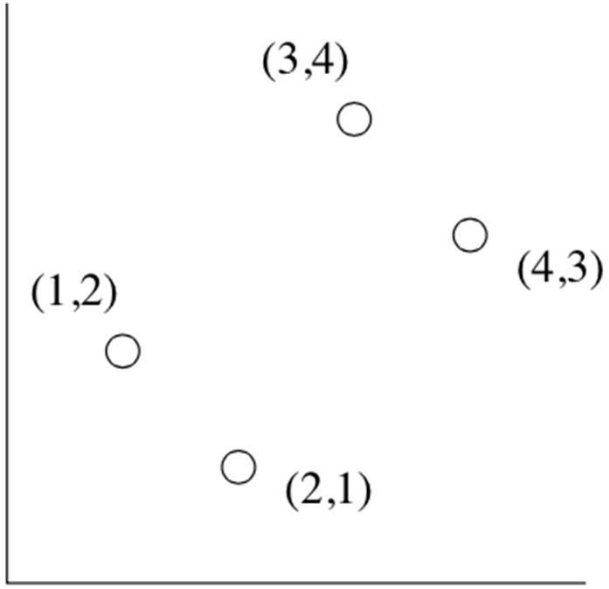
- Trattiamo l'insieme di tuple come una matrice **M** e troviamo gli autovettori di $\mathbf{MM^T}$ o $\mathbf{M^TM}$.

- La matrice di questi autovettori possono essere pensati come una rotazione rigida dello spazio ad alta dimensione.

PCA

- Algoritmo PCA:
  1. M ← Crea una matrice di dati N x d, con ogni riga un vettore riga m_n dei dati
  2. M ← sottraiamo la media *m* da ogni vettore riga m_n in M
  3. Σ ← matrice di covarianza di M
  4. Trova gli autovalori e autovettori di Σ

- PC's ← gli X autovettori con I piu' grandi autovalori



$$M=\begin{bmatrix}1&2\\2&1\\3&4\\4&3\end{bmatrix}$$

$$M^TM=\begin{bmatrix}1&2&3&4\\2&1&4&3\end{bmatrix}\begin{bmatrix}1&2\\2&1\\3&4\\4&3\end{bmatrix}=\begin{bmatrix}30&28\\28&30\end{bmatrix}$$

Trova gli autovalori

$$(30-\lambda)(30-\lambda)-28\times 28=0$$

$$\lambda=58$$

$$\lambda=2$$

I corrispondenti autovettori

$$\begin{bmatrix}30&28\\28&30\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}=58\begin{bmatrix}x\\y\end{bmatrix}$$

$$30x+28y=58x$$

$$28x+30y=58y$$

$$\begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{bmatrix}$$

$$\begin{bmatrix}30&28\\28&30\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}=2\begin{bmatrix}x\\y\end{bmatrix}$$

$$30x+28y=2x$$

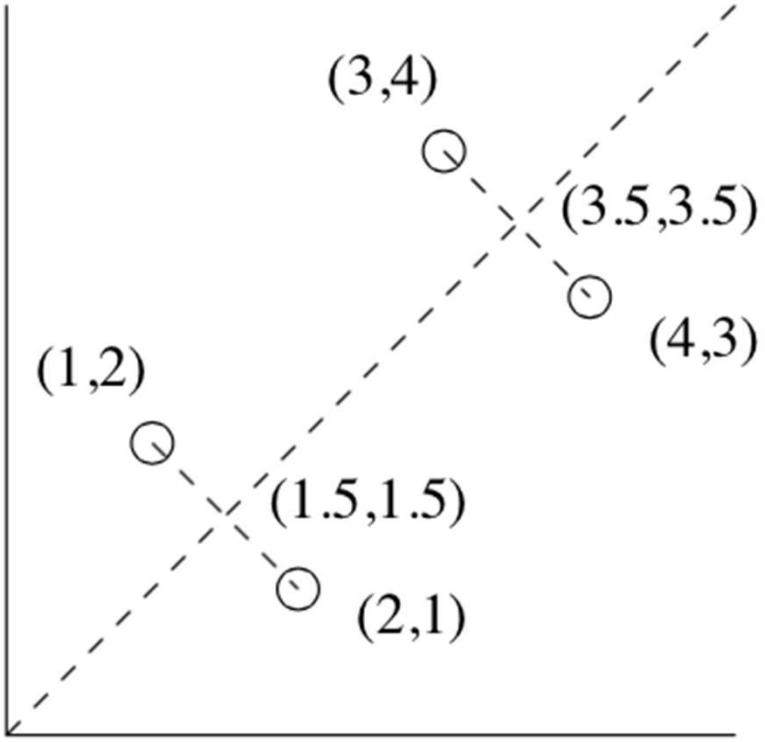
$$28x+30y=2y$$

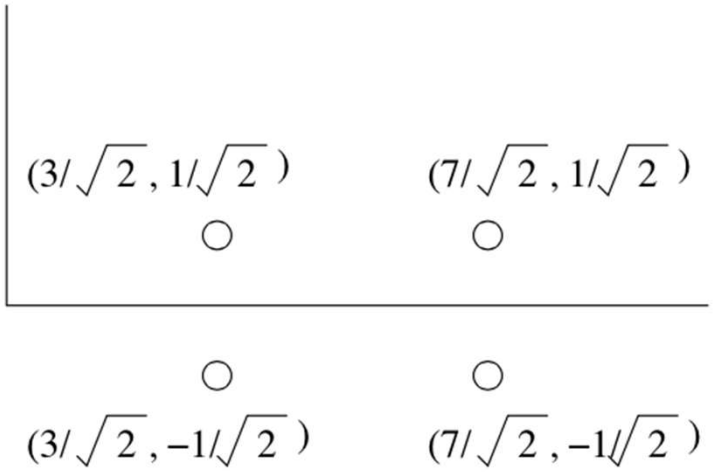
$$\begin{bmatrix}-1/\sqrt{2}\\1/\sqrt{2}\end{bmatrix}$$

Costruisci **E**, matrice degli autovettori di M^TM. Mettere per primo il primo autovettore

$$E=\begin{bmatrix}1/\sqrt{2}&-1/\sqrt{2}\\1/\sqrt{2}&1/\sqrt{2}\end{bmatrix}$$

$$ME=\begin{bmatrix}1&2\\2&1\\3&4\\4&3\end{bmatrix}\begin{bmatrix}1/\sqrt{2}&-1/\sqrt{2}\\1/\sqrt{2}&1/\sqrt{2}\end{bmatrix}=\begin{bmatrix}3/\sqrt{2}&1/\sqrt{2}\\3/\sqrt{2}&-1/\sqrt{2}\\7/\sqrt{2}&1/\sqrt{2}\\7/\sqrt{2}&-1/\sqrt{2}\end{bmatrix}$$





- **ME** è il punto di **M** trasformato in uno spazio di nuove coordinate. In questo spazio, **il primo asse (quello che corrisponde al piu' grande autovalore)** è il più significativo; formalmente, la **varianza** di un punto lungo quest'asse **è la più grande**.

- Il secondo asse, che corrisponde al secondo autovalore, è il successive secondo autovalore più significativo nello stesso senso. Questo pattern si presenta per ogni autocoppia.

- Per **trasformare** **M** in uno spazio con meno dimensioni: **Preserviamo le dimensioni che usano gli autovettori associati ai piu' alti autovalori e cancelliamo gli altri**

$$E = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \\ 7/\sqrt{2} \\ 7/\sqrt{2} \end{bmatrix}$$

Notebook

https://colab.research.google.com/drive/1lykoVbdYyHVvdJ7dUme5yN2wNGbQOrQV?usp=sharing

Singular Vaule Decomposition (SVD)

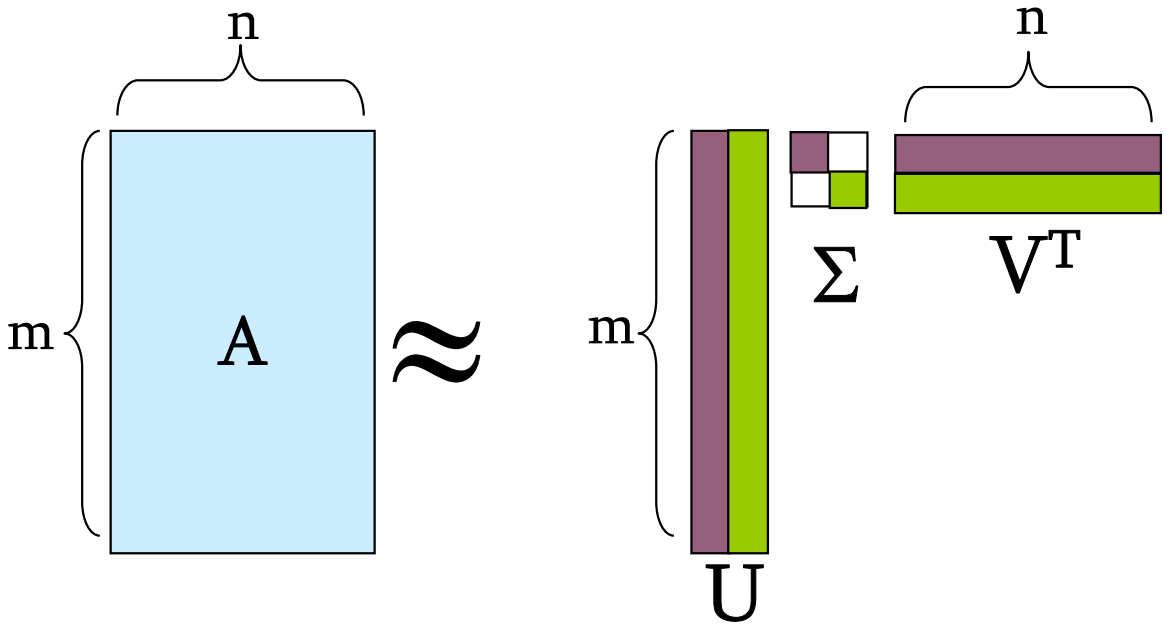
SVD - Definizione

$$\mathbf{A}_{[m \times n]} = \mathbf{U}_{[m \times r]} \Sigma_{[r \times r]} (\mathbf{V}_{[n \times r]})^T$$

- **A**: **Matrice dei dati di input**
  - Matrice *m* x *n* (e.g., *m* documenti, *n* termini)
- **U**: **Vettori singolari di sinistra**
  - Matrice *m* x *r* matrix (*m* documenti, *r* concetti)
- Σ: **Valori singolari**
  - Matrice *r* x *r* diagonale (forza di ogni 'concetto')
    (*r* : rango di **A**)
- **V**: **Vettori singolari di destra**
  - Matrice *n* x *r* (*n* termini, *r* concetti)

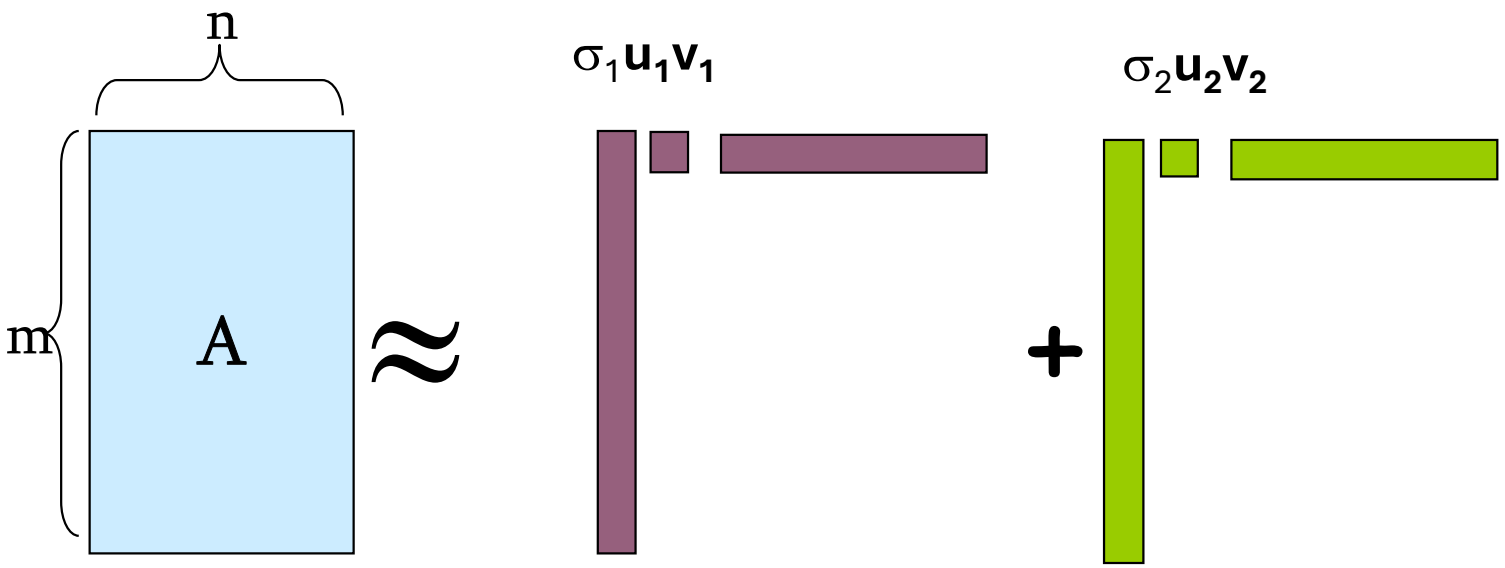
SVD

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$



SVD

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$



σ_i … scalar

u_i … vector

v_i … vector

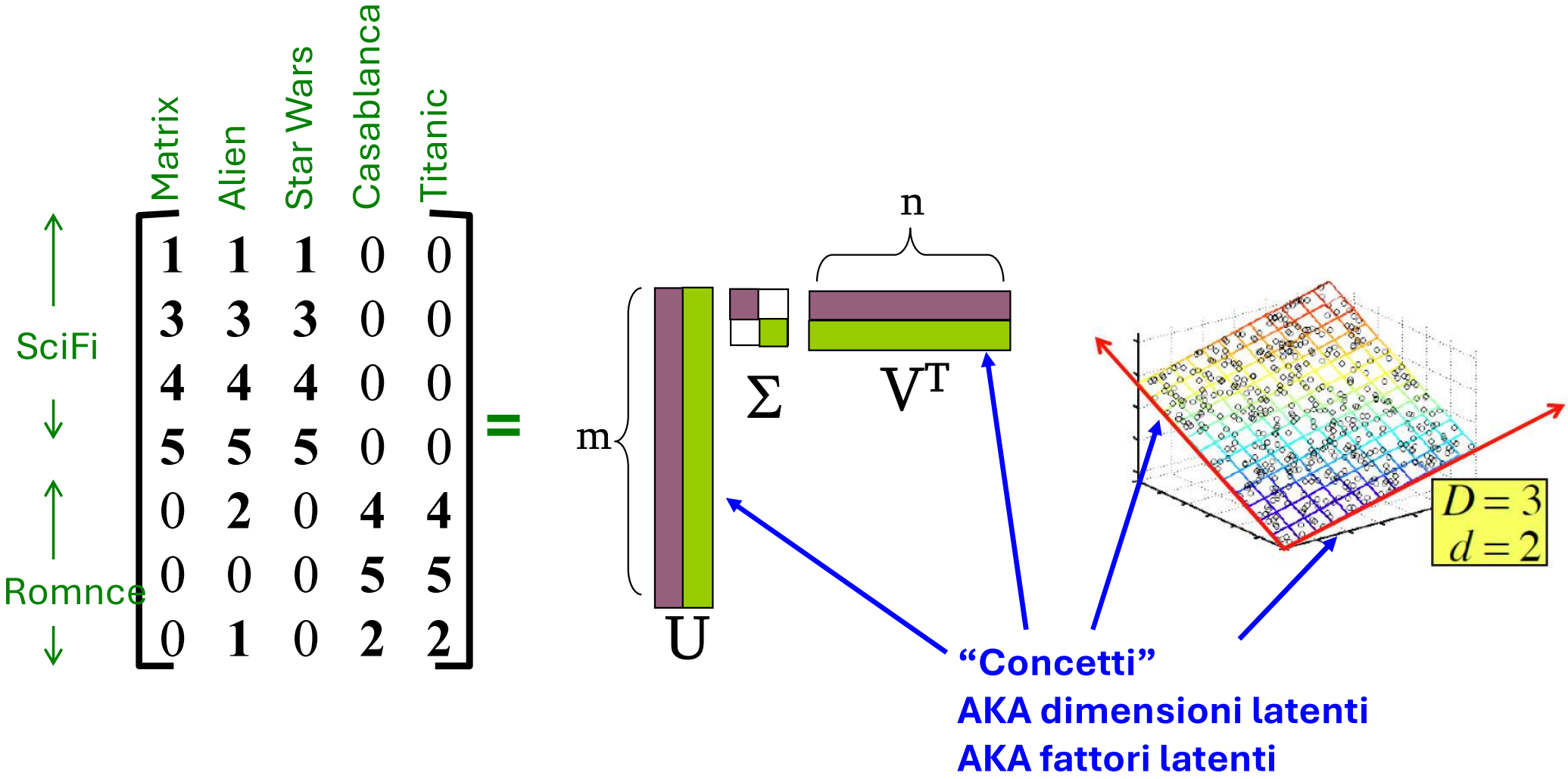
SVD - Proprietà

E' **sempre** possibile decomporre una matrice reale $\boldsymbol{A}$ in $\boldsymbol{A = U \Sigma V^T}$, dove

- $\boldsymbol{U}$, Σ, $\boldsymbol{V}$: unici
- $\boldsymbol{U}$, $\boldsymbol{V}$: Ortonormali
  - $\boldsymbol{U^T U = I}$; $\boldsymbol{V^T V = I}$ ($\boldsymbol{I}$: matrice identità)
  - (Le colonne sono vettori unitari ortonormali)
- Σ: diagonale
  - Entry (**valori singolari**) sono positivi, e ordinati in modo decrescente ($\boldsymbol{\sigma_1 \geq \sigma_2 \geq \ldots \geq 0}$)

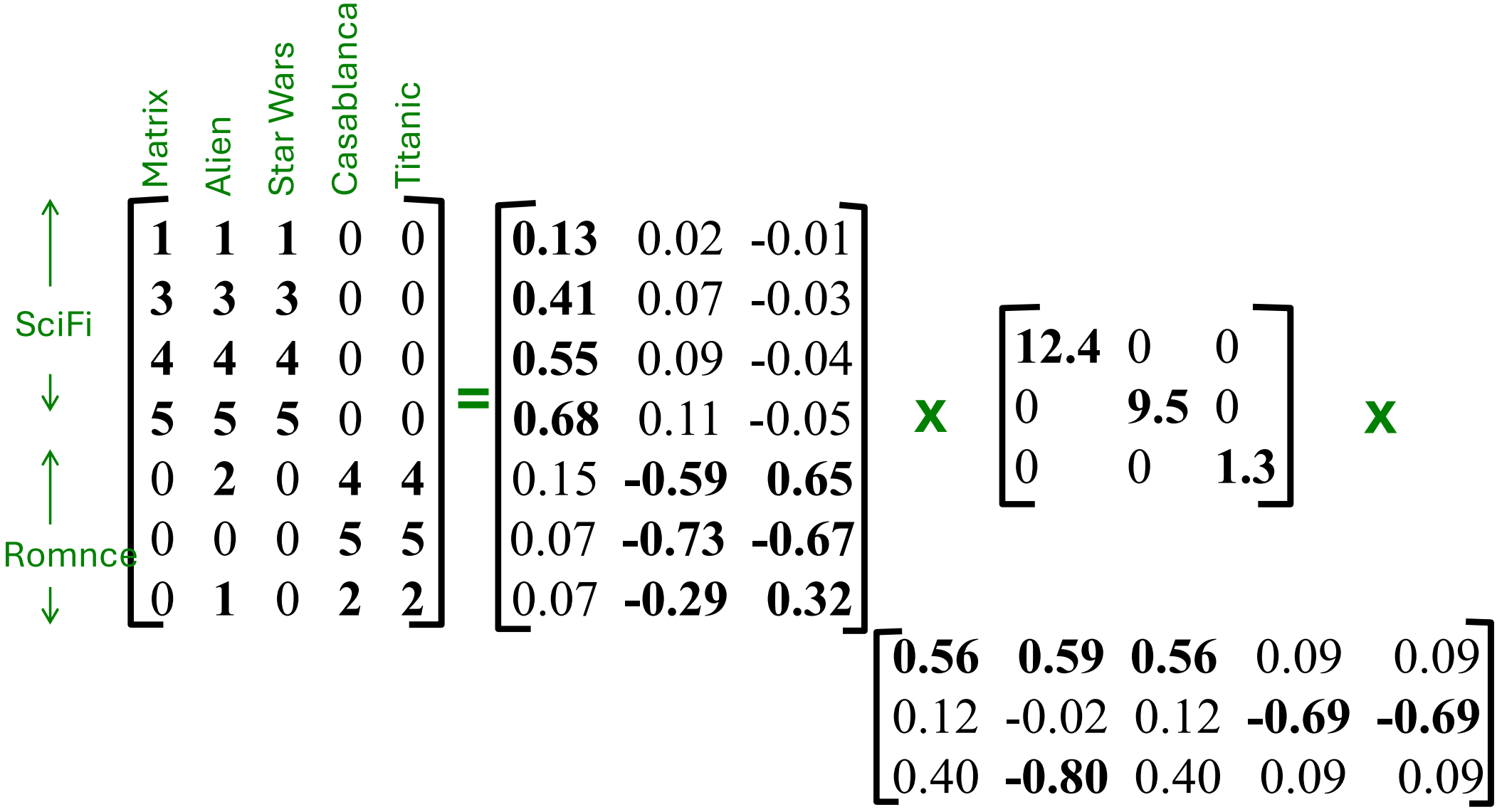
SVD – Esempio: utenti-film

- **$A = U \Sigma V^T$ - Esempio: utenti-film**



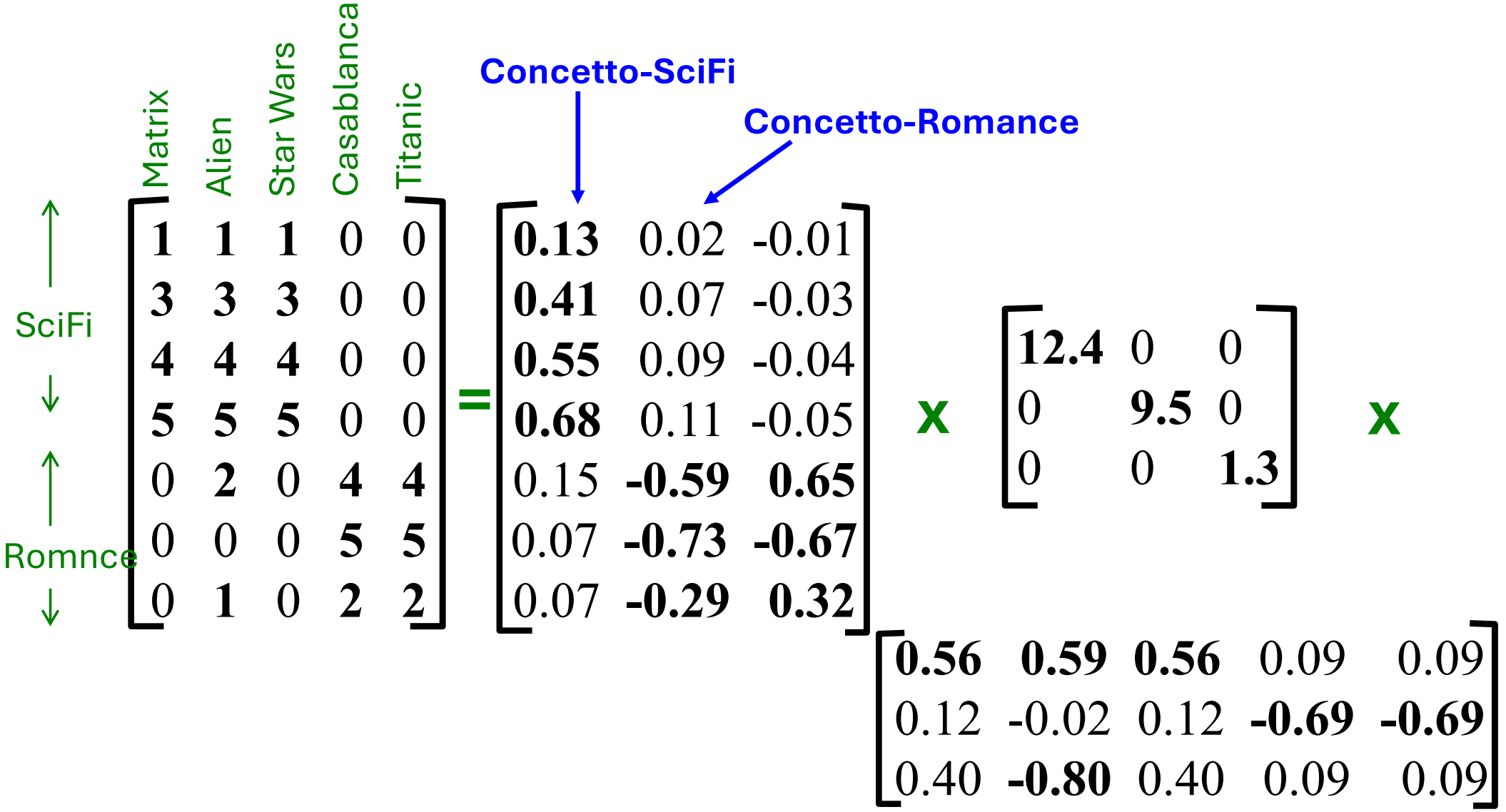
SVD – Esempio: utenti-film

- **$A = U \Sigma V^T$ - Esempio: utenti-film**



SVD – Esempio: utenti-film

- **$A = U \Sigma V^T$ - esempio: utenti-film**



SVD – Esempio: utenti-film

- **$A = U \Sigma V^T$ - esempio:**

_U_ Matrice di similarità "utenti-concetti"

	Matrix	Alien	Star Wars	Casablanca	Titanic
SciFi	1	1	1	0	0
	3	3	3	0	0
	4	4	4	0	0
	5	5	5	0	0
Romnce	0	2	0	4	4
	0	0	0	5	5
	0	1	0	2	2

=

Concetto-SciFi	Concetto-Romance	
0.13	0.02	-0.01
0.41	0.07	-0.03
0.55	0.09	-0.04
0.68	0.11	-0.05
0.15	**-0.59**	**0.65**
0.07	**-0.73**	**-0.67**
0.07	**-0.29**	**0.32**

x

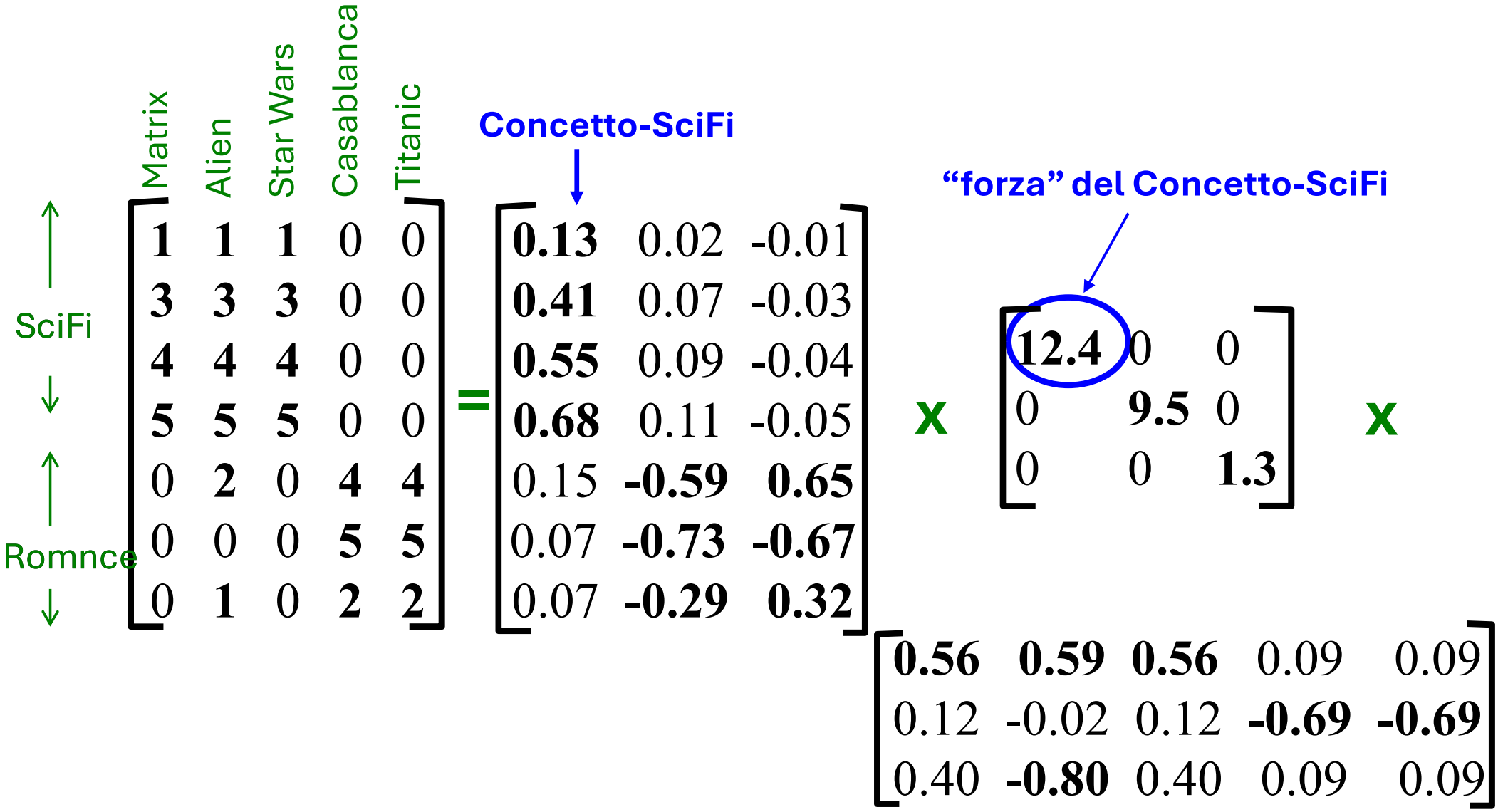
12.4	0	0
0	**9.5**	0
0	0	**1.3**

x

0.56	**0.59**	**0.56**	0.09	0.09
0.12	-0.02	0.12	**-0.69**	**-0.69**
0.40	**-0.80**	0.40	0.09	0.09

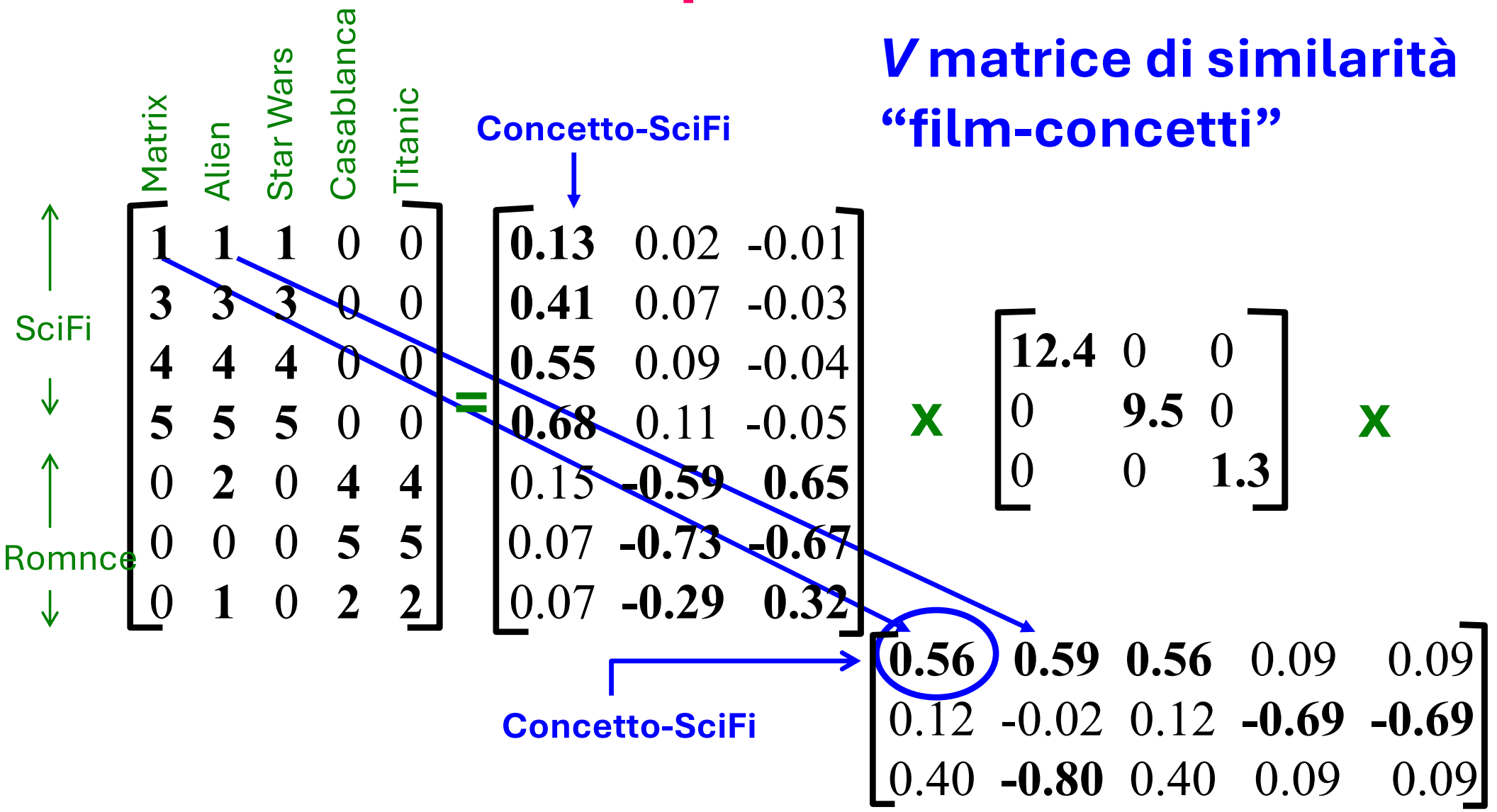
SVD – Esempio: utenti-film

- **$A = U \Sigma V^T$ - esempio:**



SVD – Esempio: utenti-film

- **$A = U \Sigma V^T$ - esempio:**



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 & 0.65 \\ 0.07 & -0.73 & -0.67 \\ 0.07 & -0.29 & 0.32 \end{bmatrix} \times \begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix} \times \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\ 0.40 & -0.80 & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

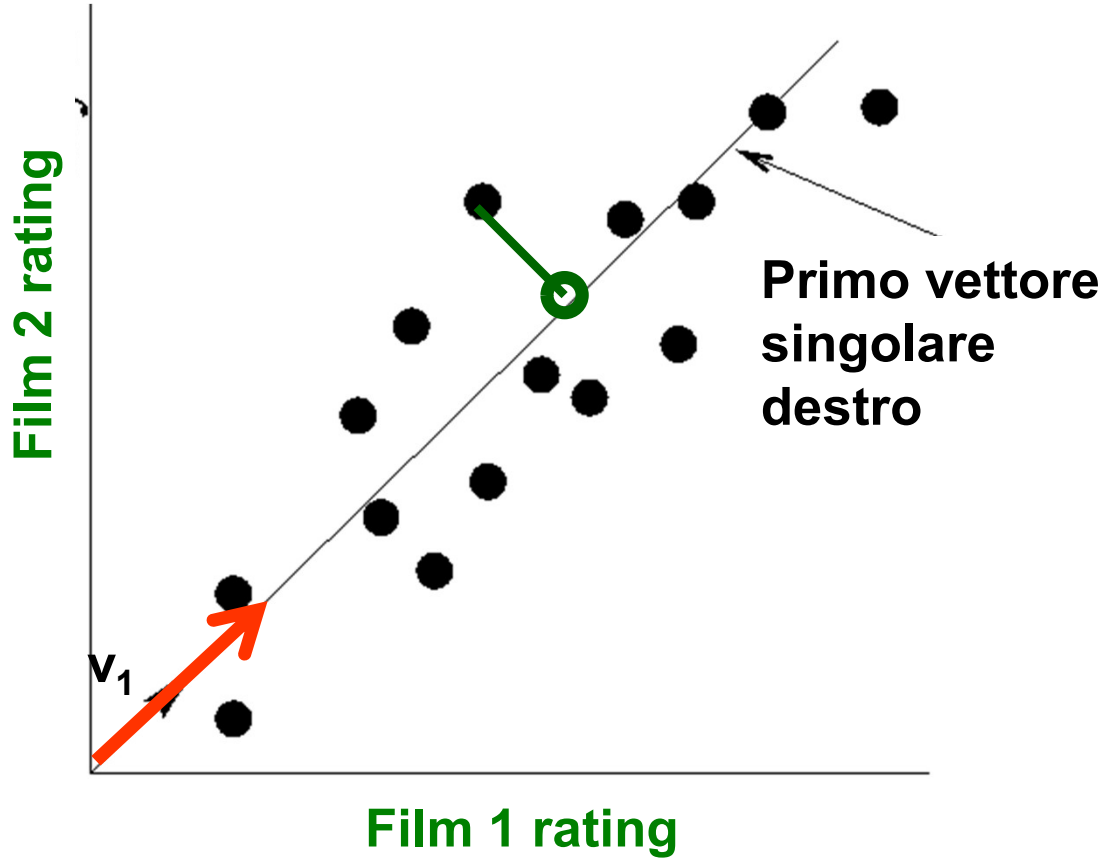
Dimensionality Reduction con SVD

SVD – Interpretazione #1

‘**film**’, ‘**utenti**’ and ‘**concetti**’:

- ***U***: matrice di similarità utenti-concetti
- ***V***: matrice di similarità film-concetti
- Σ: ogni elemento diagonale misura la “forza” di ogni concetto

SVD – Dimensionality Reduction



- Invece di usare due coordinate (x, y) per descrivere la posizione di un punto, usiamo solo una coordinata (z)
- La posizione di un punto è la sua posizione lungo $\mathbf{v_1}$

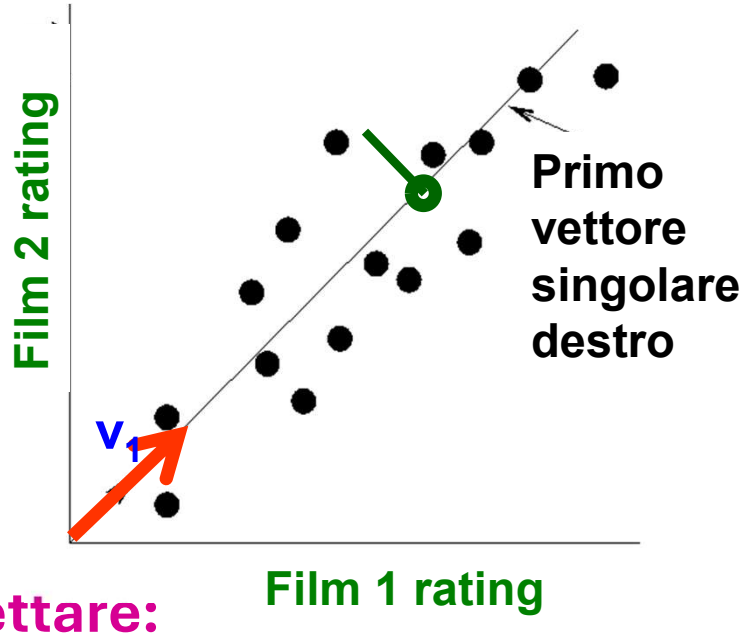
- **Come scegliamo v_1 ?** **Minimizzare l'errore di ricostruzione**

SVD – Dimensionality Reduction

- **Goal: Minimizzare la somma degli errori di ricostruzione:**

$$\sum_{i=1}^{N}\sum_{j=1}^{D}\left\| x_{ij} - z_{ij} \right\|^2$$

  - dove x_{ij} valori "originali" e z_{ij} sono le "nuove" coordinate

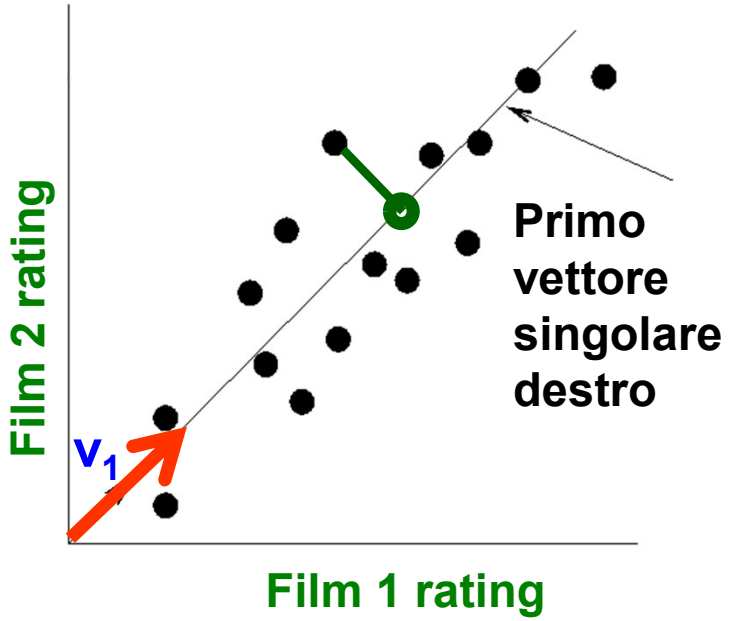


- **SVD restituisce il 'migliore' asse dove proiettare:**
  - '**migliore**' = minimizza l'errore di ricostruzione
- **Minimo errore di ricostruzione**

SVD - Interpretazione

$A = U \Sigma V^T$ - esempio:

- $U \Sigma$: coordinate dei punti nell'asse di proiezione



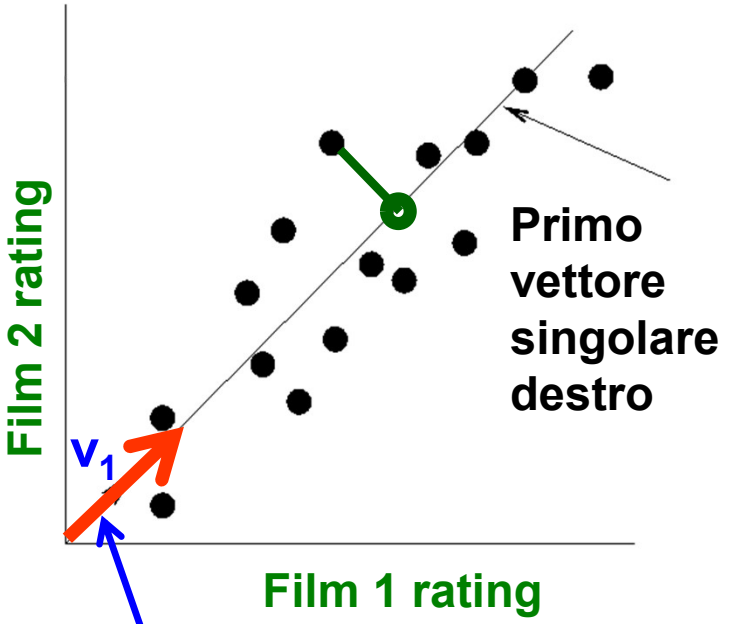
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}$$

Proiezione degli utenti sull'asse "Sci-Fi" $(U \Sigma)^T$:

$$\begin{bmatrix} 1.61 & 0.19 & -0.01 \\ 5.08 & 0.66 & -0.03 \\ 6.82 & 0.85 & -0.05 \\ 8.43 & 1.04 & -0.06 \\ 1.86 & -5.60 & 0.84 \\ 0.86 & -6.93 & -0.87 \\ 0.86 & -2.75 & 0.41 \end{bmatrix}$$

SVD - Interpretazione

- **$A = U \Sigma V^T$ - esempio:**
  - **V**: matrice "film-concetti"
  - **U**: matrice "utenti-concetti"



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 & 0.65 \\ 0.07 & -0.73 & -0.67 \\ 0.07 & -0.29 & 0.32 \end{bmatrix} \times \begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix} \times$$

$$\begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\ 0.40 & -0.80 & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

SVD - Interpretazione

Dettagli

- **Q: Come si reduce la dimensione?**

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} = \begin{bmatrix} \mathbf{0.13} & 0.02 & -0.01 \\ \mathbf{0.41} & 0.07 & -0.03 \\ \mathbf{0.55} & 0.09 & -0.04 \\ \mathbf{0.68} & 0.11 & -0.05 \\ 0.15 & \mathbf{-0.59} & \mathbf{0.65} \\ 0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\ 0.07 & \mathbf{-0.29} & \mathbf{0.32} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix} \times$$

$$\begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\ 0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

SVD - Interpretazione

Dettagli

- **Q: Come si reduce la dimensione?**
- **A: Settando a zero i valori singolari piu' piccoli**

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} = \begin{bmatrix} \mathbf{0.13} & 0.02 & -0.01 \\ \mathbf{0.41} & 0.07 & -0.03 \\ \mathbf{0.55} & 0.09 & -0.04 \\ \mathbf{0.68} & 0.11 & -0.05 \\ 0.15 & \mathbf{-0.59} & \mathbf{0.65} \\ 0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\ 0.07 & \mathbf{-0.29} & \mathbf{0.32} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\ 0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

SVD - Interpretazione #2

Dettagli

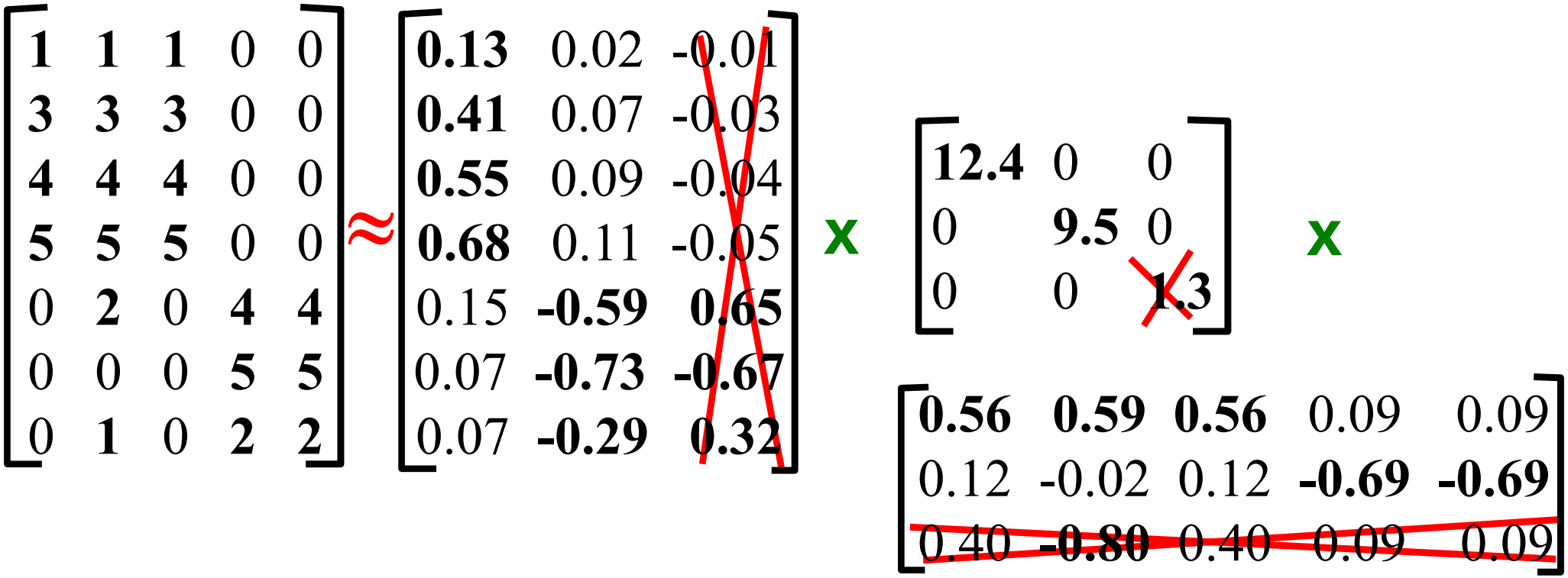
- **Q: Come si reduce la dimensione?**
- **A: Settando a zero i valori singolari piu' piccoli**

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.13} & 0.02 & -0.01 \\ \mathbf{0.41} & 0.07 & -0.03 \\ \mathbf{0.55} & 0.09 & -0.04 \\ \mathbf{0.68} & 0.11 & -0.05 \\ 0.15 & \mathbf{-0.59} & \mathbf{0.65} \\ 0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\ 0.07 & \mathbf{-0.29} & \mathbf{0.32} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\ 0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

SVD - Interpretazione #2

Dettagli

- **Q: Come si reduce la dimensione?**
- **A: Settando a zero i valori singolari piu' piccoli**



SVD - Interpretazione #2

Dettagli

- **Q: Come si reduce la dimensione?**
- **A: Settando a zero i valori singolari piu' piccoli**

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} 0.13 & 0.02 \\ 0.41 & 0.07 \\ 0.55 & 0.09 \\ 0.68 & 0.11 \\ 0.15 & -0.59 \\ 0.07 & -0.73 \\ 0.07 & -0.29 \end{bmatrix} \times \begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix} \times \begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \end{bmatrix}$$

SVD - Interpretazione #2

Dettagli

- **Q: Come si reduce la dimensione?**
- **A: Settando a zero i valori singolari piu' piccoli**

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.92} & \mathbf{0.95} & \mathbf{0.92} & 0.01 & 0.01 \\ \mathbf{2.91} & \mathbf{3.01} & \mathbf{2.91} & -0.01 & -0.01 \\ \mathbf{3.90} & \mathbf{4.04} & \mathbf{3.90} & 0.01 & 0.01 \\ \mathbf{4.82} & \mathbf{5.00} & \mathbf{4.82} & 0.03 & 0.03 \\ 0.70 & \mathbf{0.53} & 0.70 & \mathbf{4.11} & \mathbf{4.11} \\ -0.69 & 1.34 & -0.69 & \mathbf{4.78} & \mathbf{4.78} \\ 0.32 & \mathbf{0.23} & 0.32 & \mathbf{2.01} & \mathbf{2.01} \end{bmatrix}$$

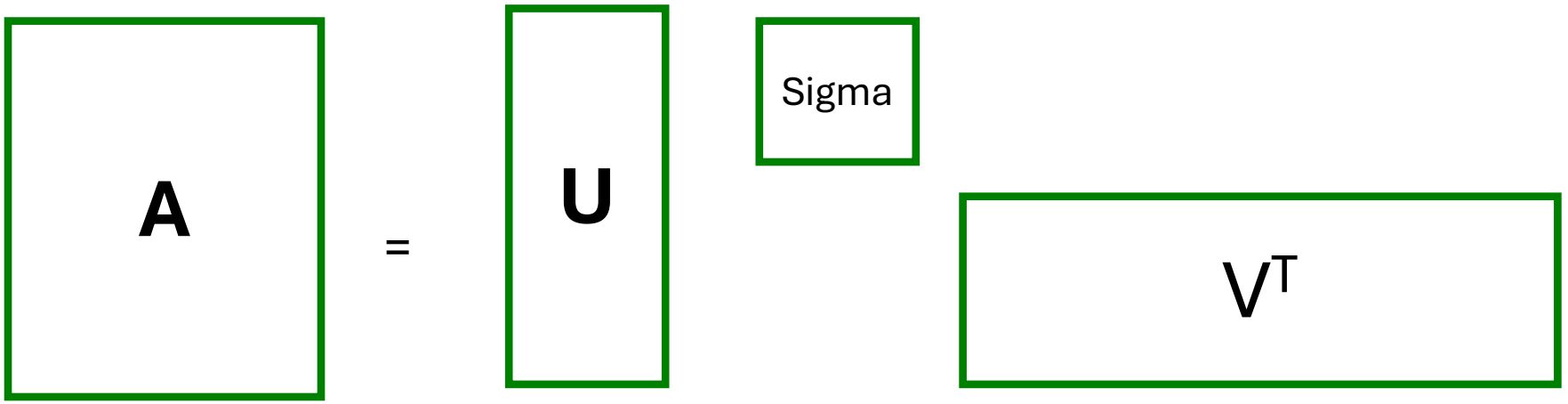
Norma di Frobenius:

$$\|M\|_F = \sqrt{\Sigma_{ij} M_{ij}^2}$$

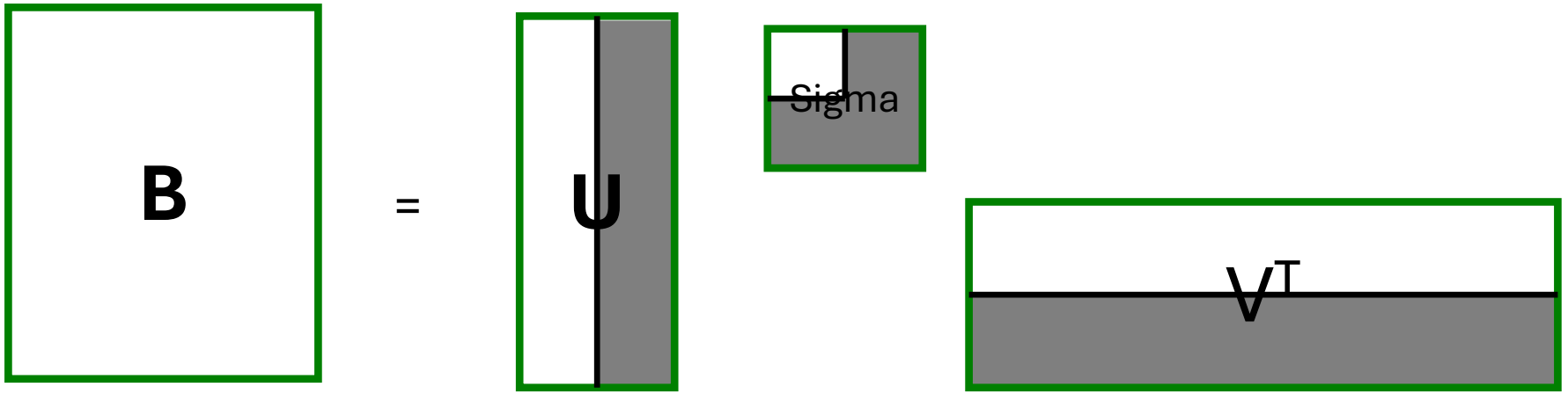
$$\|A-B\|_F = \sqrt{\Sigma_{ij} (A_{ij}-B_{ij})^2}$$

is "piccolo"

SVD – Best Low Rank Approx.



B is best approximation of A

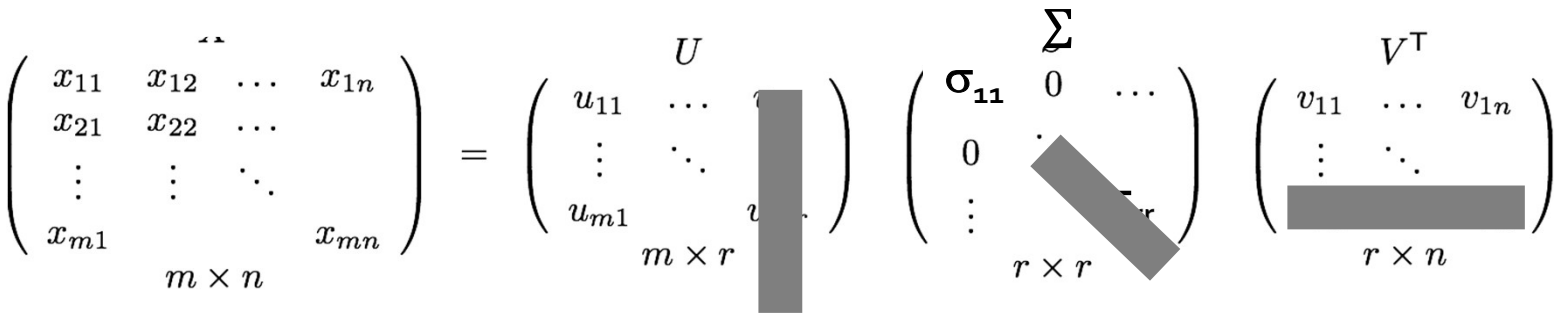


SVD – Best Low Rank Approx.

- **Teorema:**
  Sia **A = U Σ V^T** e **B = U S V^T** dove
  S = diagonale *rxr* matrice con $s_i=\sigma_i$ ($i=1\ldots k$) altrimenti $s_i=0$
  allora **B** è **la migliore** rank(**B**)=***k*** approssimazione ad ***A***

Cosa intendiamo per "migliore":

- B **soluzione che** $\min_B \|A-B\|_F$ **dove** $\text{rank}(B)=k$

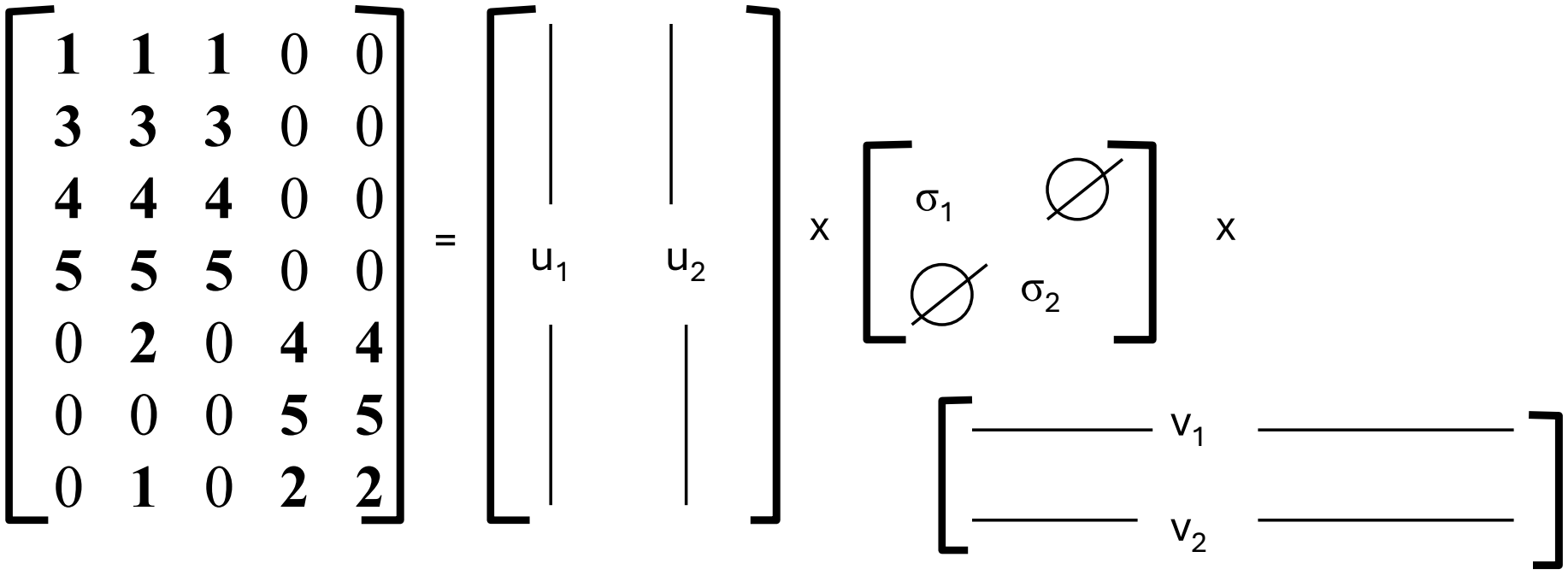


$$\|A-B\|_F = \sqrt{\Sigma_{ij} (A_{ij}-B_{ij})^2}$$

SVD - Interpretazione #2

Equivalente:

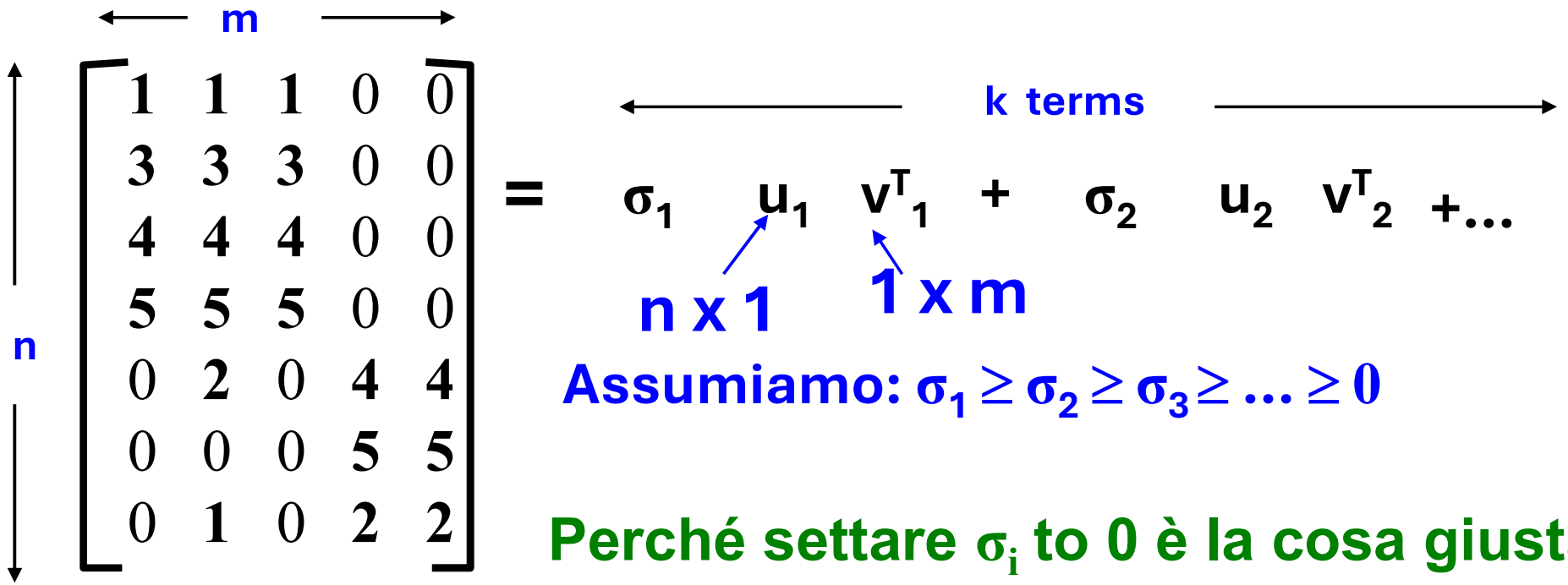
'spectral decomposition' (decomposiozione spettrale) della matrice:



SVD - Interpretazione #2

Equivalente:

'spectral decomposition'



Assumiamo: $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq 0$

Perché settare σ_i to 0 è la cosa giusta?

Vettori $\mathbf{u_i}$ e $\mathbf{v_i}$ sono unitari, quindi σ_i scala questi.

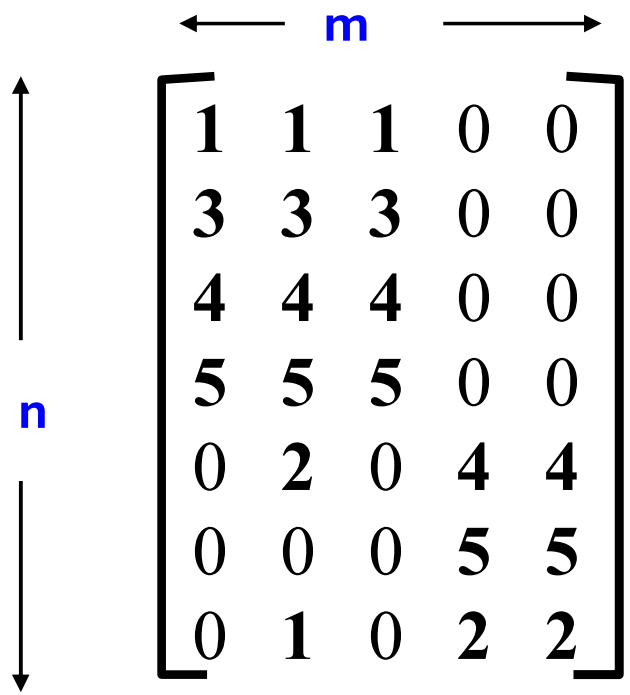
Quindi, mettere a zero σ_i piccolo introduce meno errore.

SVD - Interpretazione #2

Q: Quanti σs mantenere?

A: regola empirica:

mantenere 80-90% dell''energia' $\sum_i \sigma_i^2$



Assumiamo: $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots$

$$= \quad \sigma_1 \quad u_1 \quad v^T_1 \quad + \quad \sigma_2 \quad u_2 \quad v^T_2 \quad + \ldots$$

La somma dei quadrati dei valori singolari dovrebbe essere almeno il 90% della somma dei quadrati di tutti i valori singolari

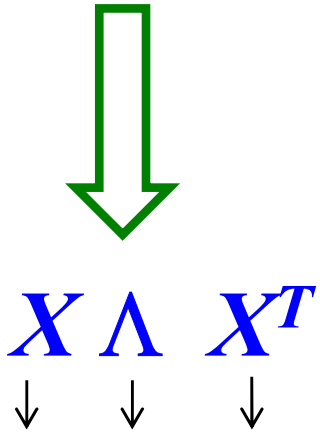
Relazione con l'auto-decomposizione

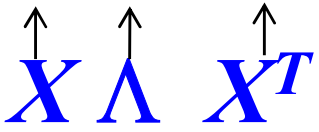
- **SVD :**
  - $A = U \Sigma V^T$
- **Eigen-decomposition:**
  - $A = X \Lambda X^T$
    - A simmetrica
    - U, V, X are ortonormali ($\mathbf{U^TU=I}$),
    - Λ, Σ diagonali
- **calcoliamo:**
  - $\mathbf{AA^T = U\Sigma V^T(U\Sigma V^T)^T = U\Sigma V^T(V\Sigma^T U^T) = U\Sigma\Sigma^T U^T}$
  - $\mathbf{A^TA = V\,\Sigma^T U^T (U\Sigma V^T) = V\,\Sigma\Sigma^T V^T}$

Relazione con Eigen-decomposition

- **SVD :**
  - $A = U \Sigma V^T$
- **Eigen-decomposition:**
  - $A = X \Lambda X^T$
    - A simmetrica
    - U, V, X are ortonormali ($\mathbf{U^TU=I}$),
    - Λ, Σ diagonali
- **calcoliamo:**
  - $\mathbf{AA^T = U\Sigma V^T(U\Sigma V^T)^T = U\Sigma V^T(V\Sigma^T U^T) = U\Sigma\Sigma^T U^T}$
  - $\mathbf{A^TA = V\,\Sigma^T U^T (U\Sigma V^T) = V\,\Sigma\Sigma^T V^T}$

Mostra come calcolare
SVD usando la decomposizione
Ad autovalori!





Algortimo per il calcolo dell'SVD

- $\mathbf{A}^T\mathbf{A} = \mathbf{V}\,\Sigma\Sigma^T\,\mathbf{V}^T$
- Ne segue che V è la matrice degli autovettori e $\Sigma\Sigma^T$ la matrice degli autovalori
- Usiamo l'algoritmo di power iteration per calcolare la prima autocoppia (x, λ) di $\mathbf{A}^T\mathbf{A}$
  - Sottraiamo l'effetto dell'autocpiia alla matrice $\mathrm{M}' = \mathrm{A}^T\mathrm{A} - \lambda x x^T$
  - Usiamo l'algoritmo di power iteration per calcolare la prima autocoppia di M'. Iteriamo questo processo per ottenere tutte le autocoppie di $\mathbf{A}^T\mathbf{A}$.

- $\mathbf{A}\mathbf{A}^T = \mathbf{U}\Sigma\Sigma^T\,\mathbf{U}^T$
- Usiamo lo stesso algoritmo per calcolare gli autovettori di AA^T

SVD - Complessità

- **Per calcolare SVD:**
  - **$O(nm^2)$** o **$O(n^2m)$** (il minore dei due)
- **MA:**
  - Meno lavoro se vogliamo solo i valori singolari
  - O se vogliamo i primi k valori singolari
  - O se la matrice è sparsa

- **Implementata** in diversi pacchetti
  - LINPACK, Matlab, SPlus, Mathematica, R

SVD - ricapitolando

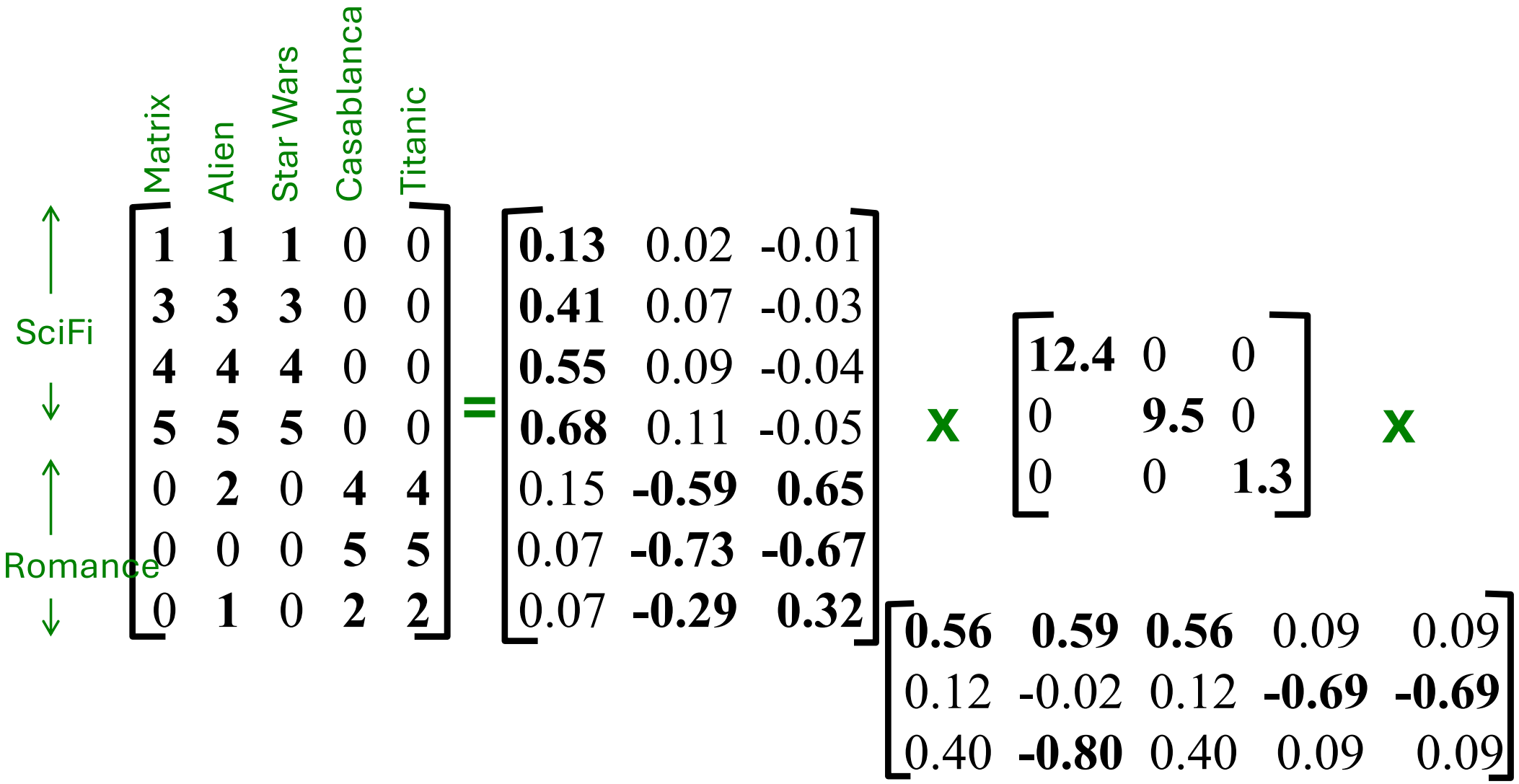
- **SVD: $A = U \Sigma V^T$: uniche**
  - **U**: similarità utenti-concetti
  - **V**: similarità film-concetti
  - Σ : forza (importanza) di ogni concetto

- **Dimensionality reduction:**
  - Mantenere I valori singolari che hanno (80-90% dell''energia')
  - SVD: correlazioni lineari

Esempio d’uso di SVD e conclusioni

Case study: come fare delle query?

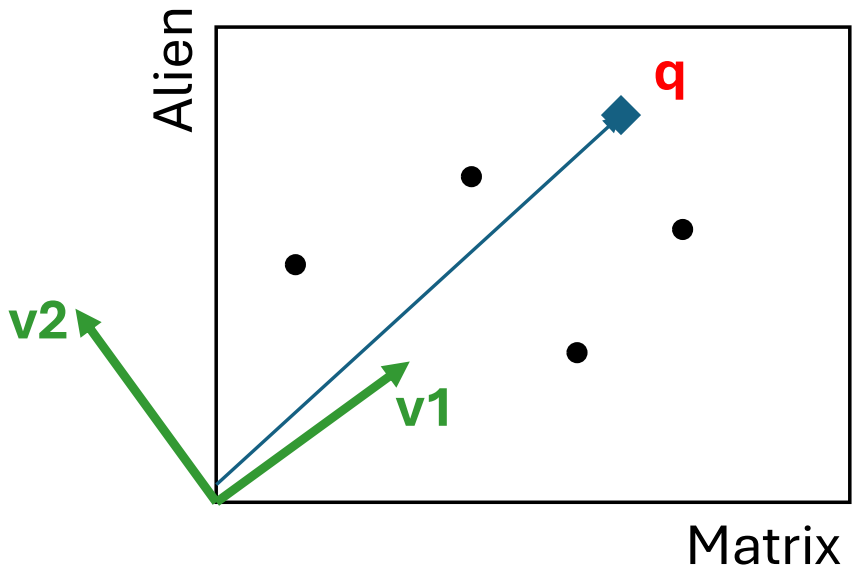
- **Q: trovare gli utenti a cui piace 'Matrix'**
- **A: Map la query nello spazio dei concetti come?**



Case study: come fare delle query?

- **Q: trovare gli utenti a cui piace ‘Matrix’**
- **A: Map la query nello spazio dei concetti come?**

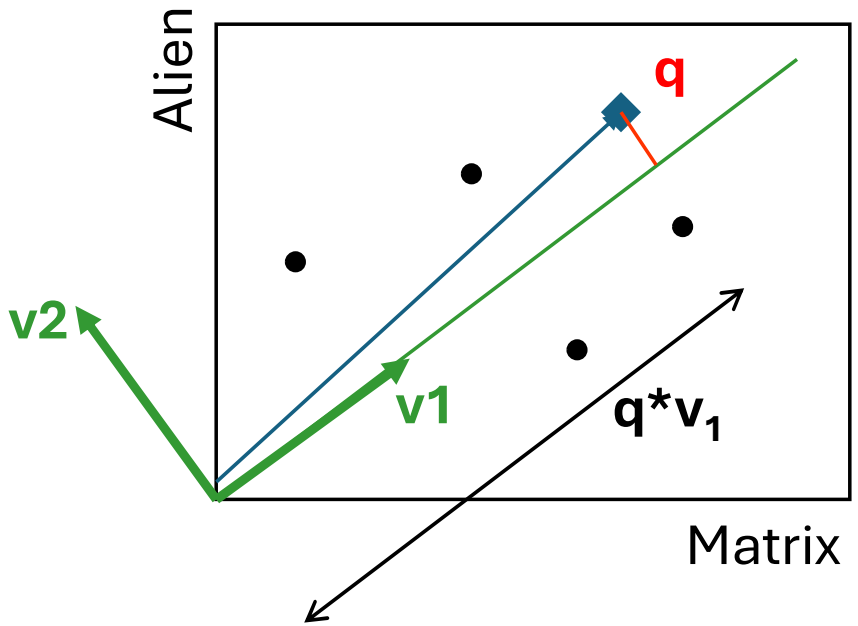
$$q = \begin{bmatrix} \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ 5 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Case study: come fare delle query?

- **Q: trovare gli utenti a cui piace 'Matrix'**
- **A: Map la query nello spazio dei concetti come?**

$$q = \begin{bmatrix} \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ 5 & 0 & 0 & 0 & 0 \end{bmatrix}$$

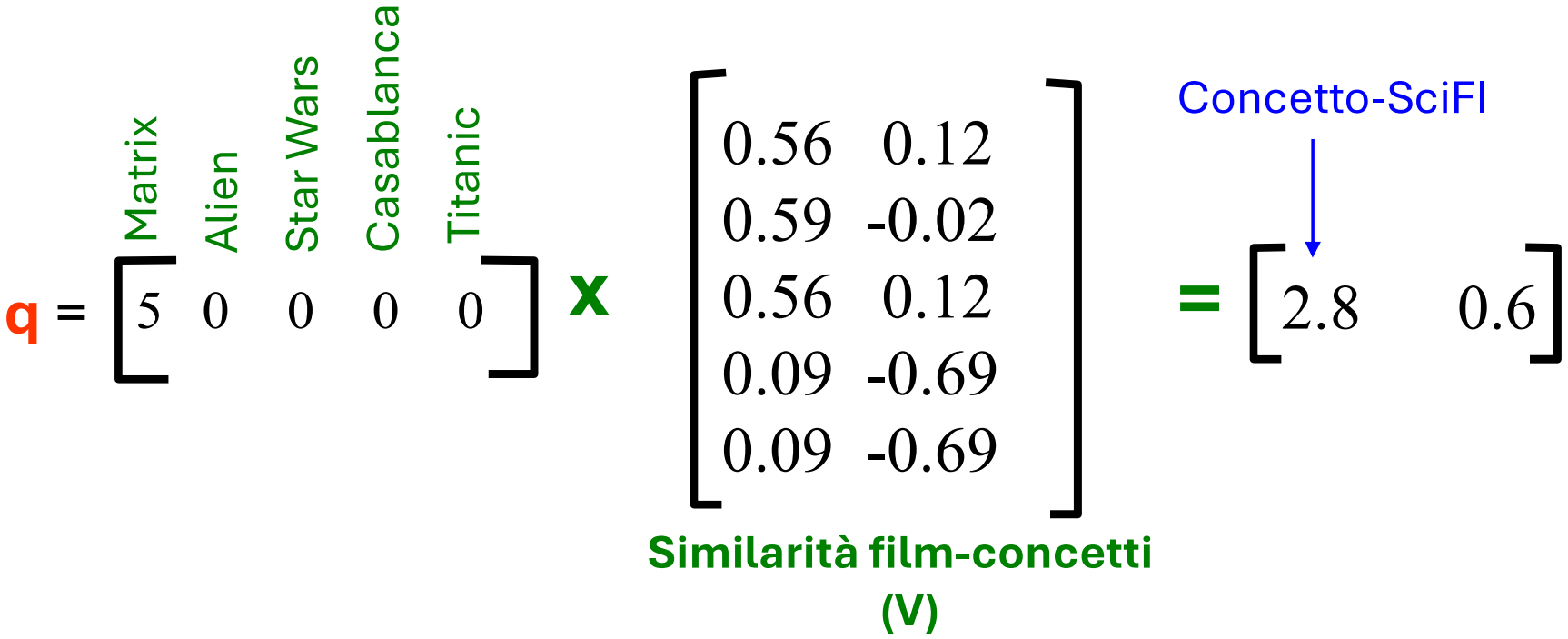


Case study: come fare delle query?

In modo compatto abbiamo:

$q_{concept} = q\,V$

Es.:



$$\begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \end{bmatrix} \times \begin{bmatrix} 2.8 \\ 0.6 \end{bmatrix}$$

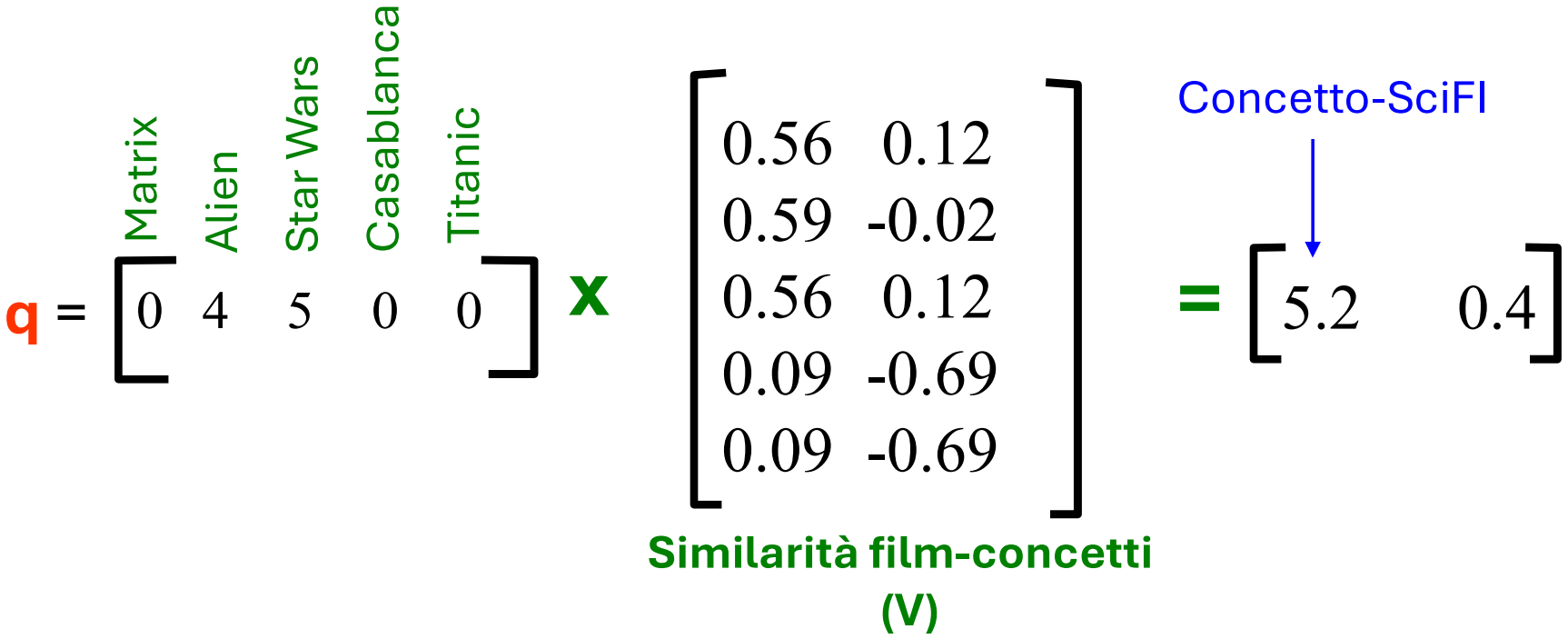
$$\mathbf{q} = \begin{bmatrix} \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ 5 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ 1{,}64 & 1{,}64 & ..1{,}64 & 0.16 & 0{,}16 \end{bmatrix}$$

Case study: come fare delle query?

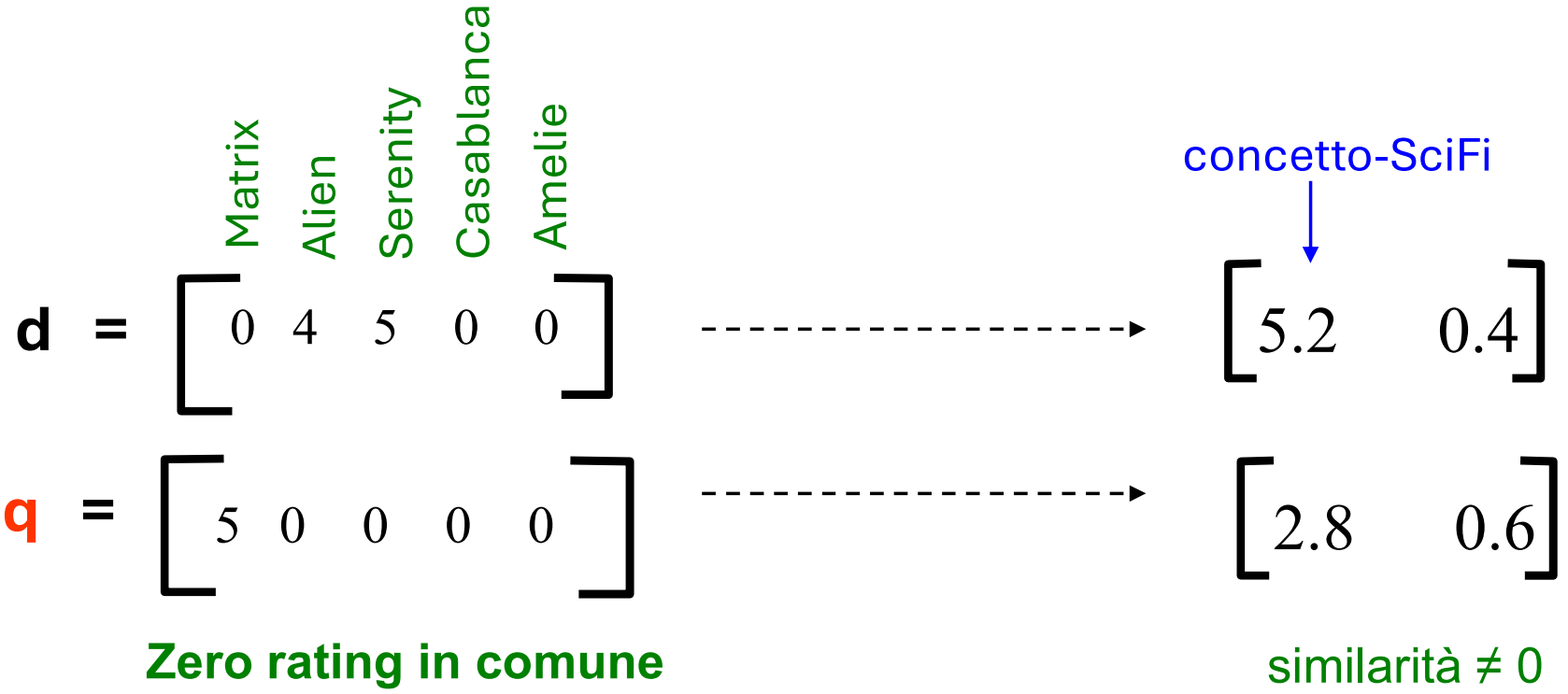
- ***Come viene mappato d* che ha valutato('Alien', 'Star Wars')?**
  $d_{concept} = d\,V$

Es.:



Case study: come fare delle query?

- **Osservazione:** l'utente ***d*** che ha valutato ('*Alien*', '*Star Wars*') sarà **simile** all'utente **q** che ha valutato ('*Matrix*'), sebbene ***d*** e **q non hanno rating in comune!**

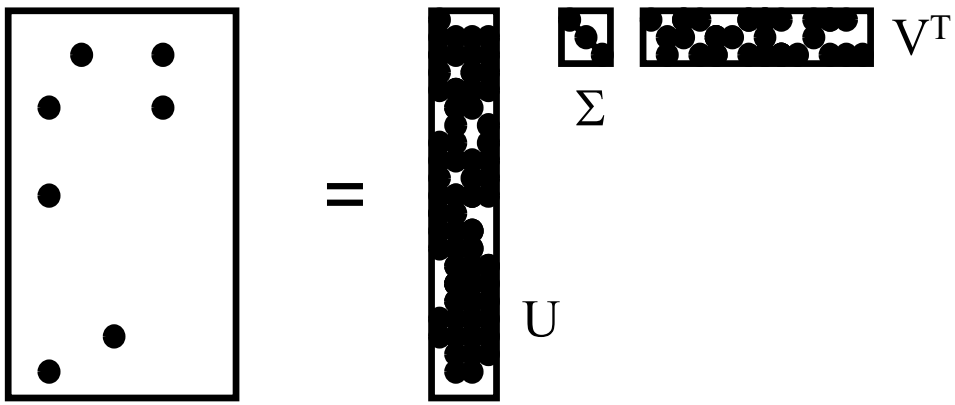


SVD: limiti e problemi

+ **Approssimazione low-rank ottimale**
  norma di Frobenius

- **Difficile da interpretare:**
  - Un vettore singolare specifica una combinazione lineare di colonne e righe di input

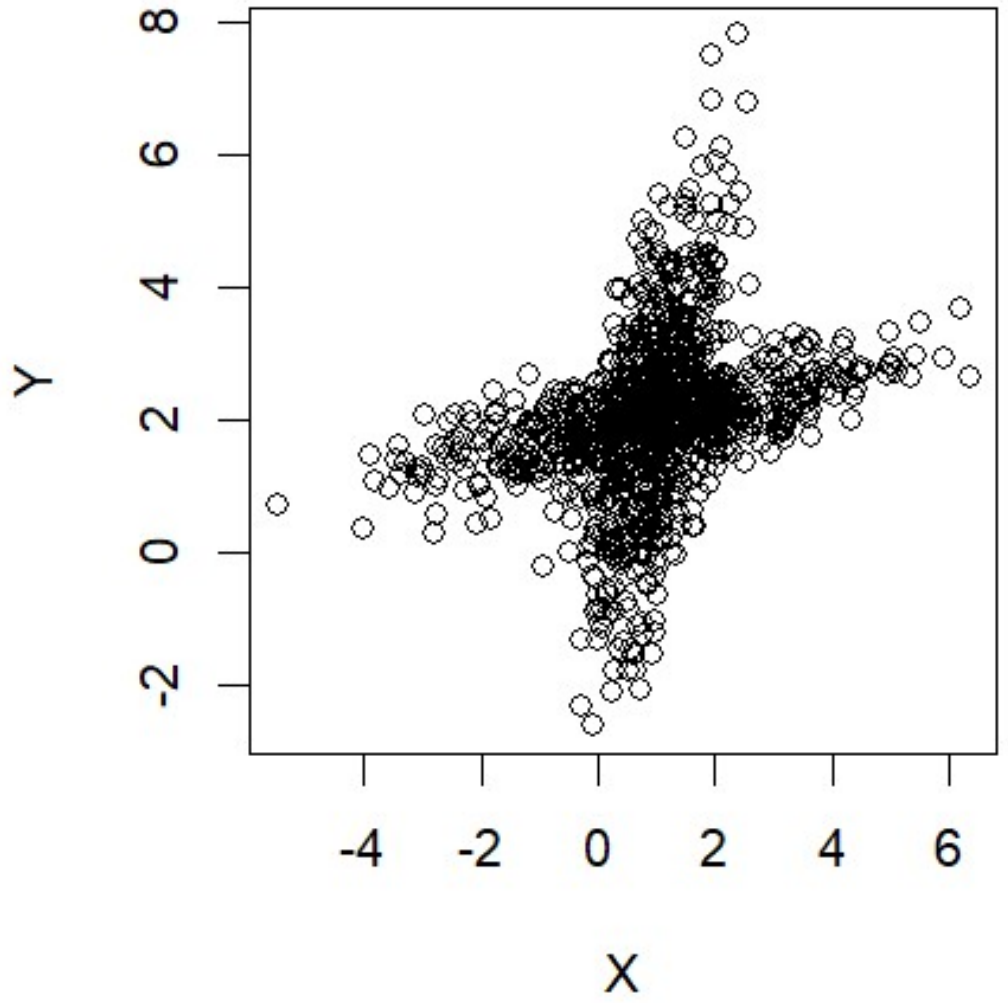
- **Mancanza di matrici sparse:**
  - I vettori singolari sono **densi!**



Decomposizione CUR

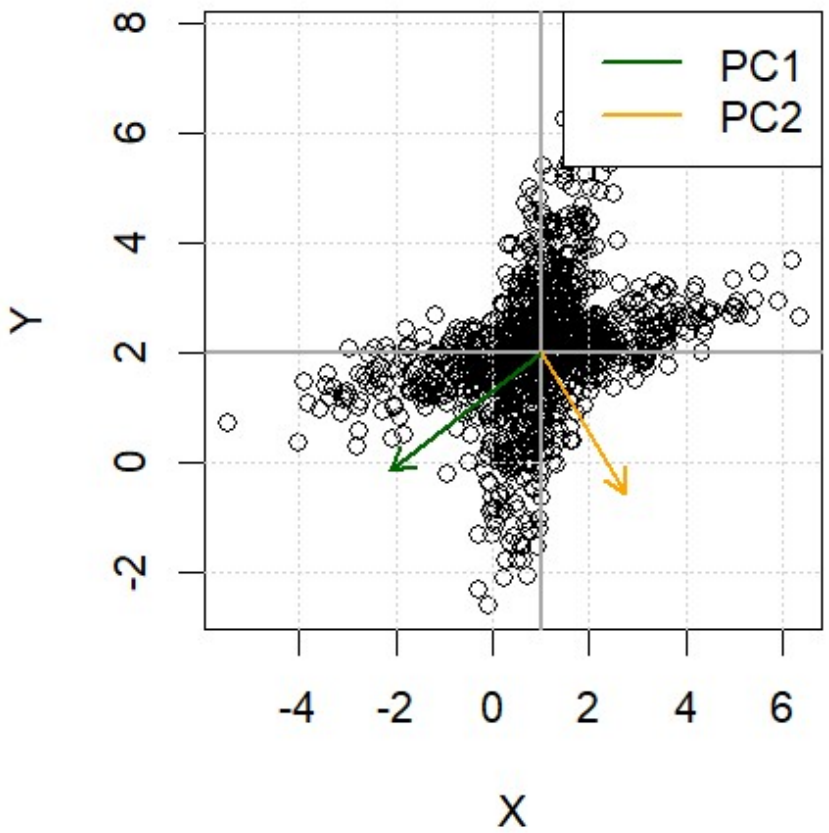
Semplice motivazione per la decomposizione CUR

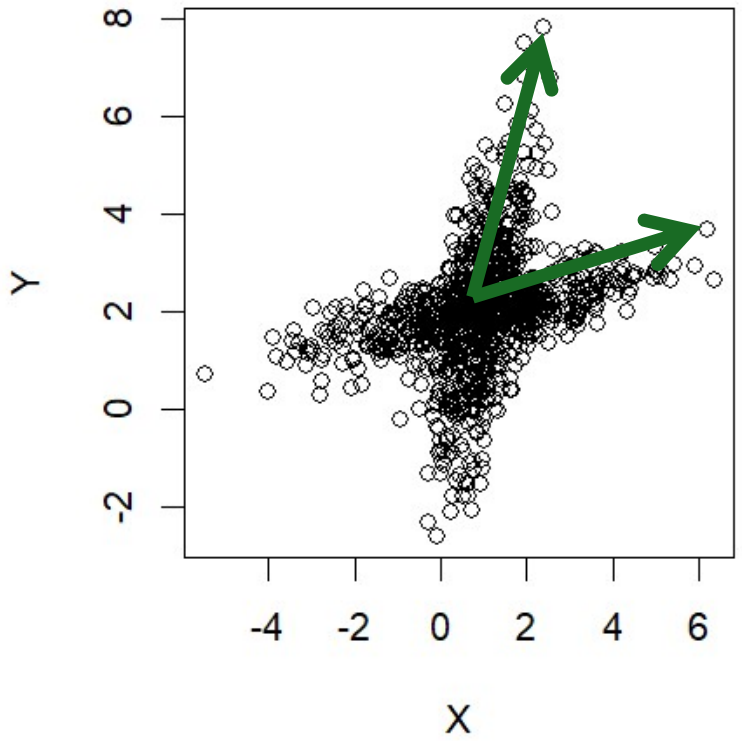
- Supponiamo di avere i seguenti dati
  - 1000 vettori a due dimensioni
  - Due gruppi di elementi che vanno su due assi completamente diversi



Calcoliamo la PCA

- Non catturiamo la direzione dei nostri dati





https://www.pnas.org/doi/10.1073/pnas.0803205106

Norma di Frobenius:

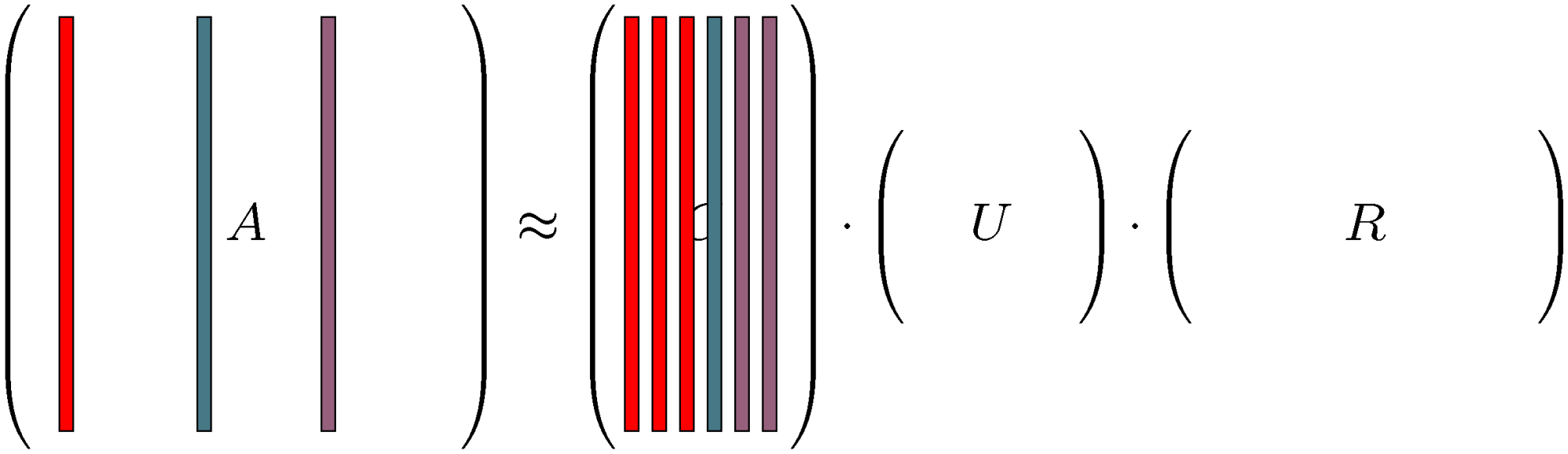
$$\|X\|_F = \sqrt{\Sigma_{ij} X_{ij}^2}$$

Decomposizione CUR

- **Goal: Eprimere A come il prodotto delle matrici C,U,R**

  Tale che $\|A-C\cdot U\cdot R\|_F$ piccolo

- **"Vincoli" su C e R:**



Norma di Frobenius:

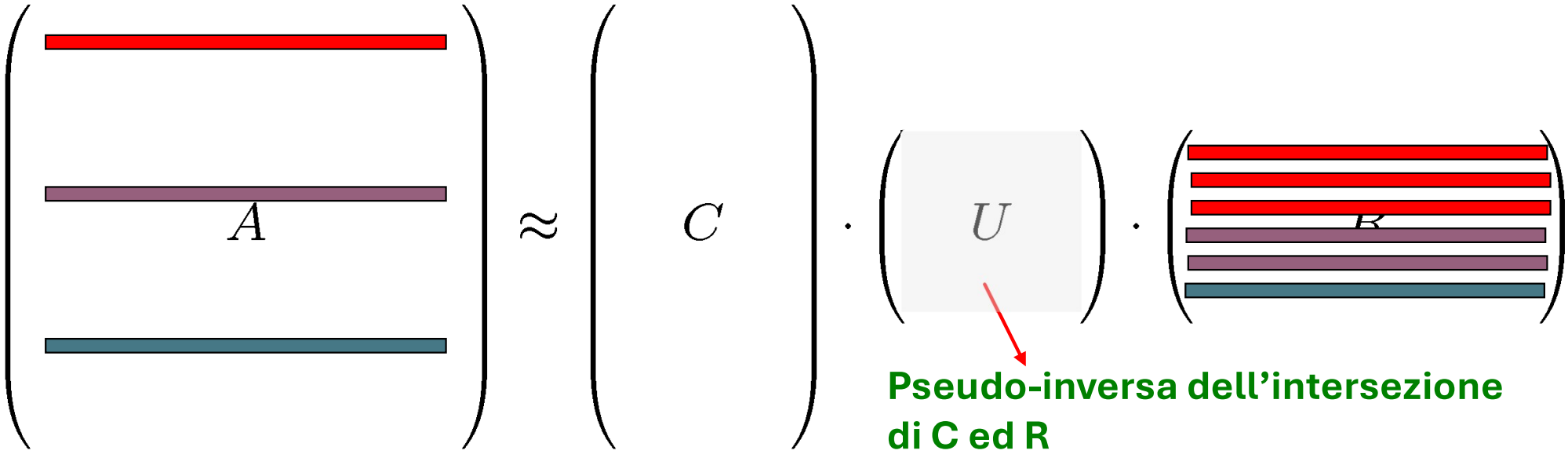
$$\|X\|_F = \sqrt{\Sigma_{ij} X_{ij}^2}$$

CUR Decomposition

- **Goal: Esprimere A come il prodotto delle matrici C,U,R**

  Tale che $\|A-C\cdot U\cdot R\|_F$ piccolo

- **"Vincoli" su C e R:**



Pseudo-inversa dell'intersezione di C ed R

A C U R

CUR: Buona approssimazione ad SVD

- **Sia:**
  $\mathbf{A_k}$ la "migliore" approssimazione ***k rank***
  ad **A** ($\mathbf{A_k}$ è l'SVD di A)

Teorema [Drineas et al.]

CUR in tempo O(**m·n**)

- $\|\mathbf{A-CUR}\|_F \leq \|\mathbf{A-A_k}\|_F + \varepsilon\|\mathbf{A}\|_F$

Con probabilità almeno **1-**δ, selezionando

- **O(k log(1/**δ**)/**ε^2**)** colonne
- **O(k^2 log^3(1/**δ**)/**ε^6**)** righe

In pratica:
Selezionare 4*k* col/righe

CUR: Dettagli

- **Sampling delle colonne (simile per le righe):**

Input: matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, sample size c

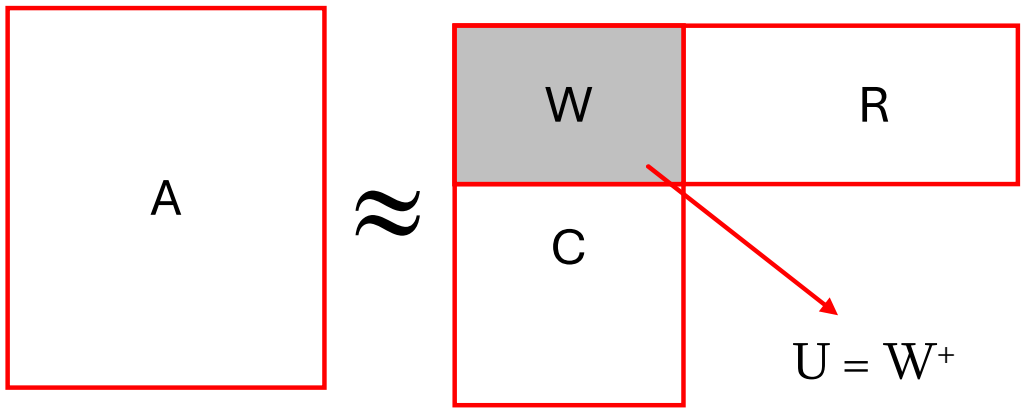
Output: $\mathbf{C}_d \in \mathbb{R}^{m \times c}$

1. for $x = 1 : n$ [column distribution]
2. $P(x) = \sum_i \mathbf{A}(i,x)^2 / \sum_{i,j} \mathbf{A}(i,j)^2$
3. for $i = 1 : c$ [sample columns]
4. Pick $j \in 1 : n$ based on distribution $P(x)$
5. Compute $\mathbf{C}_d(:,i) = \mathbf{A}(:,j) / \sqrt{cP(j)}$

Algoritmo randomizzato, la stessa Colonna potrebbe essere campionata piu' volte

Come calcolare U

- Sia **W** l'"intersezione" delle colonne e delle righe campionate **C** ed **R**
  - Calcoliamo l'SVD **W** = **X Z** $\mathbf{Y}^T$
- **Quindi: U** = $\mathbf{W}^+$ = **Y** $\mathbf{Z}^+$ $\mathbf{X}^T$
  - Z^+: **reciproci dei valori singolari non zero di Z:** $Z^+_{ii} = 1/Z_{ii}$
  - W^+ è la "**pseudoinversa**"



Perché la pseudoinversa funziona?

W = X Z Y abbiamo $W^{-1} = X^{-1}$ Z^{-1} Y^{-1}

Per l'ortonormalità $X^{-1}=X^T$ and $Y^{-1}=Y^T$

Poiché Z è diagonale $Z^{-1} = 1/Z_{ii}$

Quindi, se **W** è non-singolare, la pseudoinversa è la vera inversa

CUR: Buona approssimazione ad SVD

- **Per esempio:**
  - **Selezonare $c = O\left(\frac{k \log k}{\varepsilon^2}\right)$ colonne di A con l'algoritmo ColumnSelect**
  - **Selezionare $r = O\left(\frac{k \log k}{\varepsilon^2}\right)$ righe di A con l'algoritmo RowSelect**
  - **Impostare $U = W^+$**
- **Allora:**

  Con probabilità 98%

CUR error SVD error

$$\|A - CUR\|_F \leq (2 + \epsilon) \|A - A_k\|_F$$

In pratica:
Selezionare 4*k* col/righe per un approssimazione "rank-k"

https://arxiv.org/pdf/1903.09698.pdf

CUR: Pro & Contro

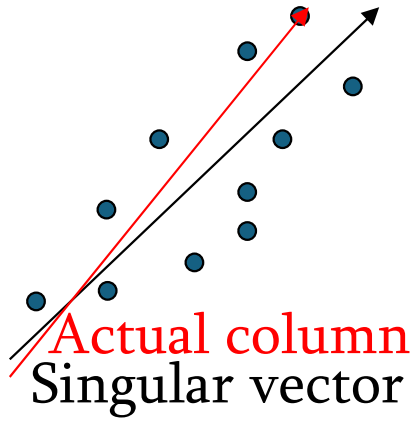
+ Facile da interpretare

- I vettori della base sono esattamente le righe e le colonne della matrice

+ Base sparsa

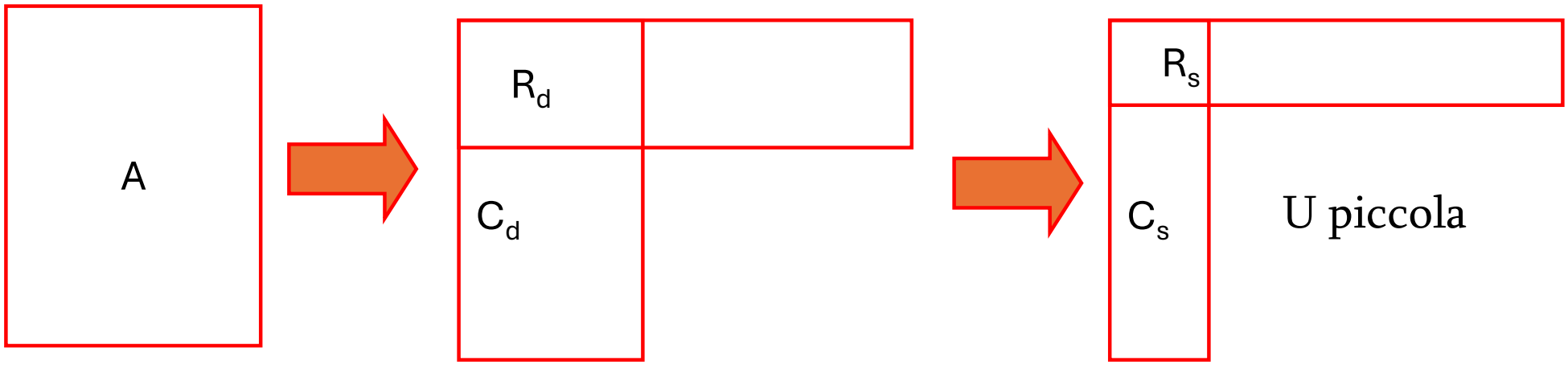
- Colonne e righe duplicate

- Colonne con norme molto alte verranno selezionate spesso

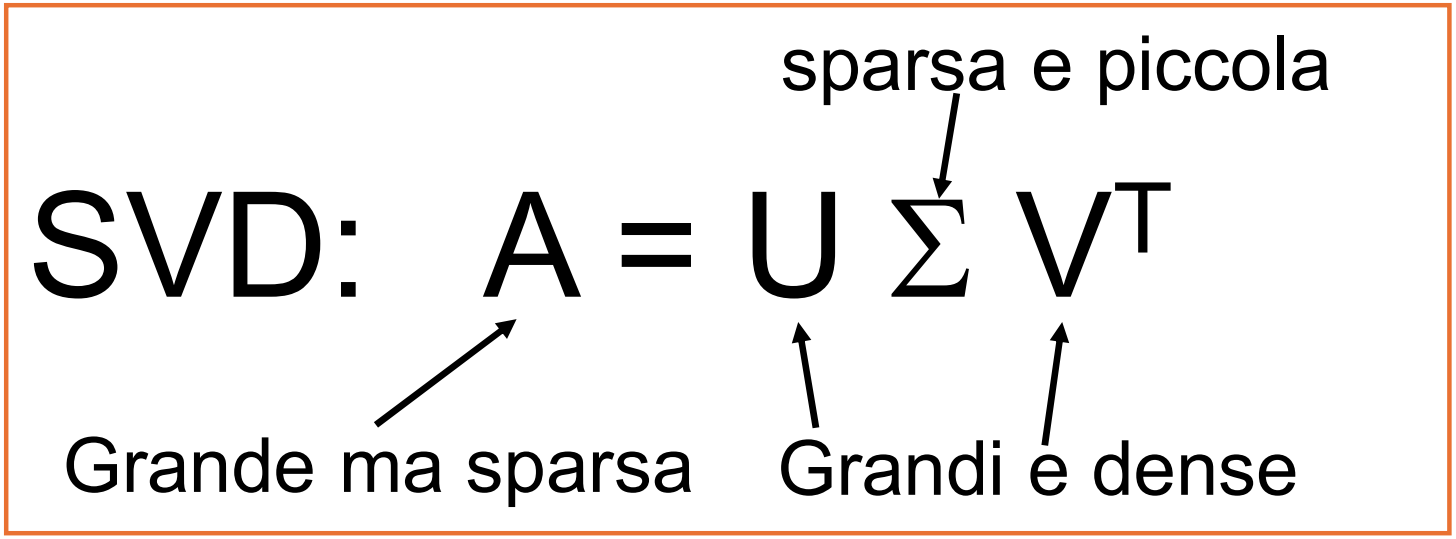


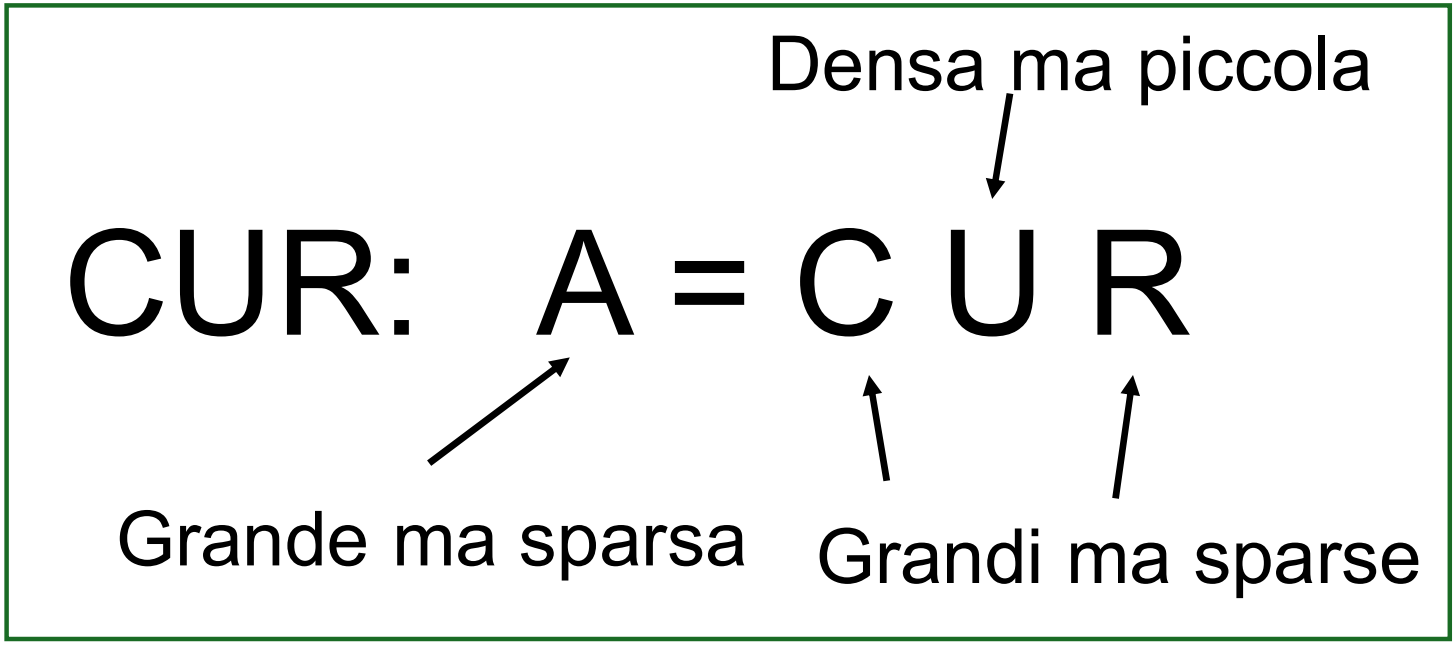
Soluzione

- **Se vogliamo sbarazzarci dei duplicati:**
  - Rimuoverli
  - Scalare (moltiplicare) le colonne/righe per la radice quadrata del numero dei duplicati



SVD vs. CUR

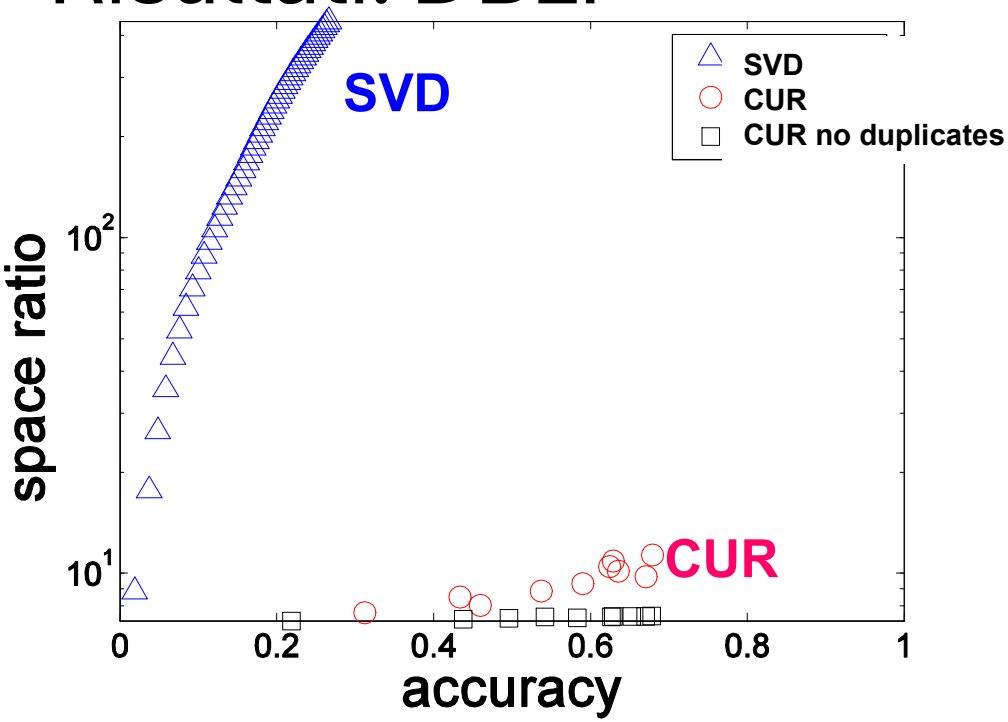

sparsa e piccola
SVD: A = U Σ V^T
Grande ma sparsa
Grandi e dense

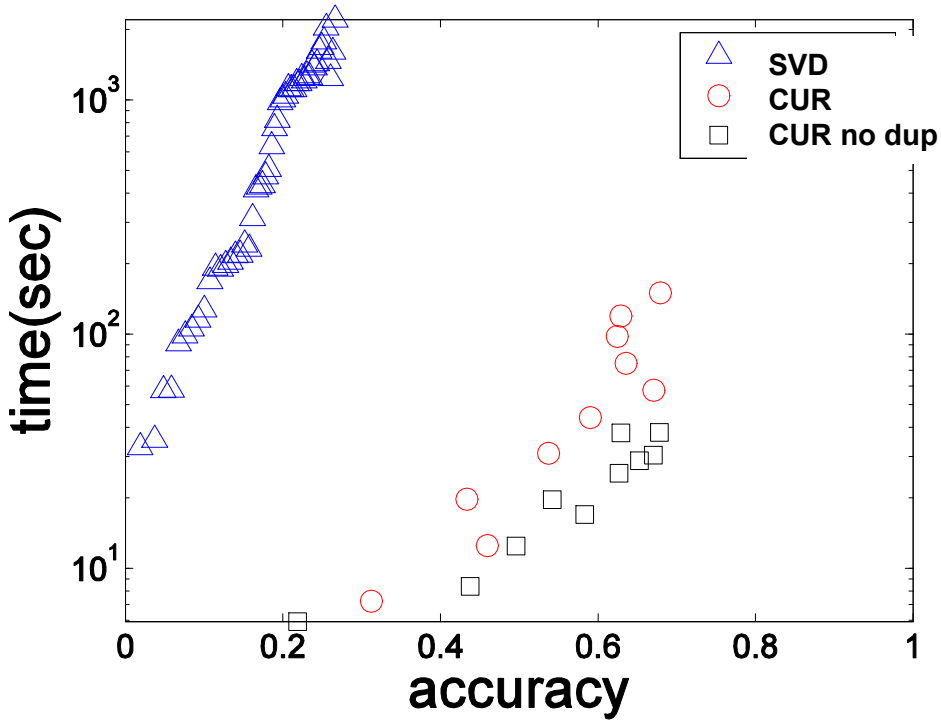

Densa ma piccola
CUR: A = C U R
Grande ma sparsa
Grandi ma sparse

SVD vs. CUR: un esperimento

- **DBLP Dati bibliografici**
  - Autori-conferenze matrice sparsa molto grande
  - A_{ij}: Numero di articoli pubblicati dall'autori *i* nella conferenza *j*
  - 428K autori (righe), 3659 conferenze(colonne)
    - **Molto sparsa**
- **Vogliamo ridurre la dimensionalità**
  - Quanto tempo ci vuole?
  - Qual è l'errore di ricostruizione?
  - Quanto spazio ci serve?

Risultati: DBLP





- **Accuratezza:**
  - 1 – relative sum squared errors
- **Rapporto spazio:**
  - #entry matrice output / #entry matrice input
- **CPU time**

Sun, Faloutsos: *Less is More: Compact Matrix Decomposition for Large Sparse Graphs*, SDM '07.

Approfondimenti: CUR

- Drineas et al., *Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition*, SIAM Journal on Computing, 2006.

- J. Sun, Y. Xie, H. Zhang, C. Faloutsos: *Less is More: Compact Matrix Decomposition for Large Sparse Graphs*, SDM 2007

- *Intra- and interpopulation genotype reconstruction from tagging SNPs*, P. Paschou, M. W. Mahoney, A. Javed, J. R. Kidd, A. J. Pakstis, S. Gu, K. K. Kidd, and P. Drineas, Genome Research, 17(1), 96-107 (2007)

- *Tensor-CUR Decompositions For Tensor-Based Data*, M. W. Mahoney, M. Maggioni, and P. Drineas, Proc. 12-th Annual SIGKDD, 327-336 (2006)

Non Negative Matrix Factorisation

Nonnegative Matrix Factorisation

- NMF consente una rappresentazione dei dati lineare:

$$A \approx WH$$

  - Dove
  - $A = [a_{mn}]$
  - $W = [w_{mk}]$ tale che $w_{mk} \geq 0$
  - $H = [h_{kn}]$ tale che $h_{kn} \geq 0$

  - Il rango ***k*** della fattorizzazione è scelto in modo tale che ***(m + n)k < mn***, e il prodotto *WH* è una rappresentazione compressa di *A*.

Perche la fattorizzazione a fattori non negativi?

- La nonnegatività induce sparsità.
- Consente una decomposizione in parti

NMF

- A: Matrice dei dati m X n
  - m features (righe)
  - n osservazioni/esempi/vettori di feature (colonne)
- $\mathbf{a_i} = (a_{1i}, a_{2i}, \ldots, a_{mi})^T$: i-simo vettore di osservazione di feature estratto dagli n
- $\mathbf{a_i}$ vettore colonna in $\mathbb{R}_+^m$

W e H

- W : M x K **matrice dizionario**:
  - w_{mk} coefficiente della matrice,
  - **$\mathbf{w_m}$** a dizionario/vettore base con K elementi;

- H : K X N **matrice di attivazione/espansione**:
  - **$\mathbf{h_n}$** : vettore colonna di coefficienti di attivazione per l'osservazione **$\mathbf{a_n}$**:

$$a_{fn} \approx \sum_{k=1}^{K} w_{fk} \times h_{kn}$$

- **$\mathbf{h_k}$**: **vettore riga** di coefficienti di attivazione per il vettore base **$\mathbf{w_f}$**.

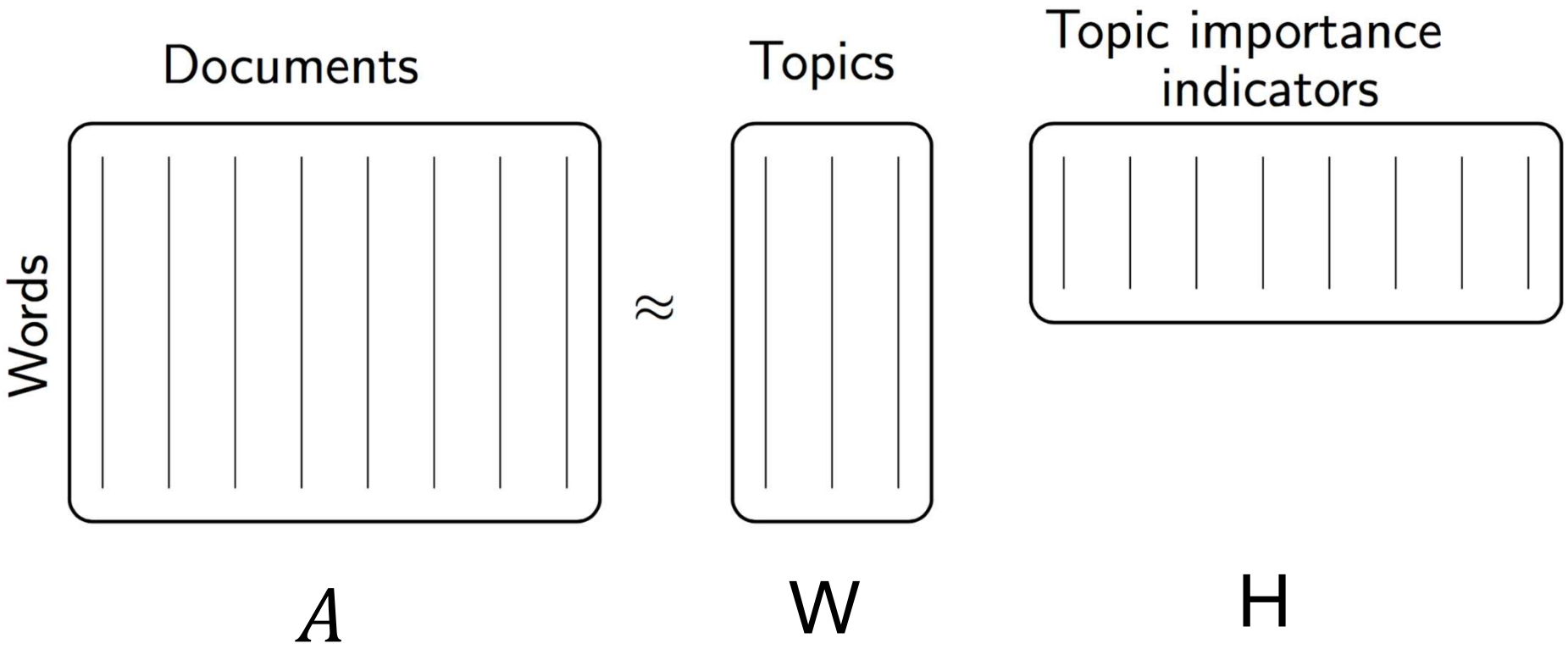
Interpretazione di W e H

- Le colonne di **W** sono i vettori della base, ogni colonna di **A** può essere costruita come combinazione lineare delle k colonne di **W**.

- Le colonne di **H** danno i pesi (coefficienti) associati ad ogni vettore della base.

- NMF è una decomposizione dei dati non non-supervised, consente di effettuare una analisi delle variabili latenti, che possono essere usati per diversi scopi.

- **Scoperta dei concetti:**
  - assumiamo **A** = [a_{mn}] è una di co-occorrenza matrice termini-documenti: a_{mn} è la frequenza della x_m nel documento d_n;


Documents
Topics
Topic importance indicators
Words
≈
A
W
H

NMF legame con Probabilistic Latent Semantic Analysis (PLSA)

- Modello PLSA (Hofmann, 1999)

$$P(m_f, d_n) = \sum_{k=1}^{K} P(t_k) P(d_n | t_k) P(m_f | t_k)$$

- I documenti possono essere descritti attraverso alcuni concetti sottostanti t_k.

- Sia $w_{fk} = \hat{P}(t_k)\hat{P}(m_f|t_k)$ e $h_{kn} = \hat{P}(d_n|t_k)$
- Il modello puo' essere riscritto come:

$$[\hat{P}(m_f d_n)] = [\hat{a}_{fn}] = \mathbf{WH}$$

- Con $\mathbf{w}_k$ interpretati come concetti che possono spiegare i dati analizzati attraverso i relativi $\mathbf{h}_k$

Funzione obiettivo

- NMF Approssimazione A ≈ WH usualmente ottenuta con:

$$min_{W,H\geq 0} D(A|WH)$$

- Funzione obiettivo Euclidea

$$\sum_{f,n}(a_{fn} - \hat{a}_{fn})^2$$

- Divergenza di Kullback-Leibler (KL)

$$\sum_{f,n}(a_{fn} log \frac{a_{fn}}{\hat{a}_{fn}} - a_{fn} + \hat{a}_{fn})$$

Algoritmo iterativo per NMF

$$w_{ia} = w_{ia} \sum_{\mu} \frac{a_{i\mu}}{(WH)_{i\mu}} H_{a\mu}$$

$$w_{ia} = \frac{w_{ia}}{\sum_j w_{ja}}$$

$$h_{a\mu} = h_{a\mu} \sum_{i} w_{ia} \frac{a_{i\mu}}{(WH)_{i\mu}}$$

L'iterazione di queste regole di aggiornamento convergono ad un massimo locale della funzione obiettivo

$$X = \sum_{i=1}^{f} \sum_{\mu=1}^{n} \left[a_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu} \right]$$

Notebook

https://colab.research.google.com/drive/15yp-FWlZZeAavVqcR3XnhnvwHYYSMAp_?usp=sharing

Pacchetti R per la riduzione della dimensionalità

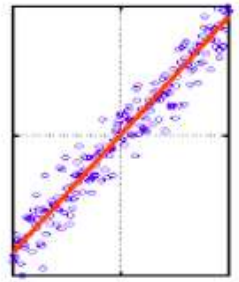
- **prcomp** e **svd** disponibili come funzioni di base di R

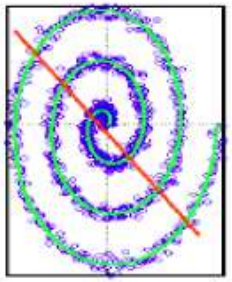
- Package utili:
  - rCUR
  - NMF
  - irlba
  - rARPACK

- Matrixcalc
  - Per il calcolo della Norma di Frobenius

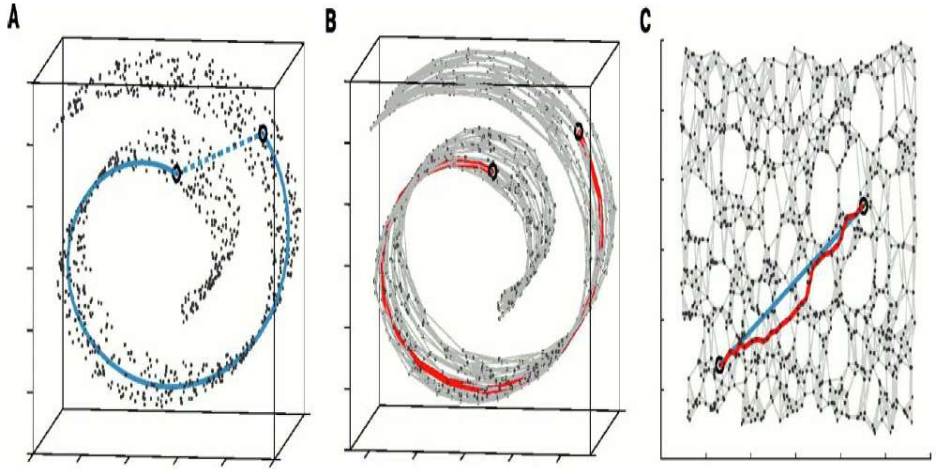
L’assunsione della linearità

- **SVD è limitata alla proiezione lineare:**
  - Lower-dimensional linear projection che preservano la distanza Euclidea





- Metodi Non-lineari: **Isomap**
  - I dati cadono in una curva (manifold) a piu’ bassa dimensione
    - Usare una distanza congrua per il manifold
  - **Come?**
    - Grafo di adiacenza
    - Distanza geodesica è la distanza nel grafo
    - SVD/PCA della matrice delle distanze del grafo



Tecniche piu’ recenti

- t-SNE (t-distributed stochastic neighbor embedding)
- https://lvdmaaten.github.io/tsne/

- UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction
- https://umap-learn.readthedocs.io/en/latest/clustering.html