Dimensionality Reduction: PCA & SVD & NMF & CUR

Perché la riduzione della dimensionalità?

- Molte fonti di dati che possono essere viste come matrici di grandi dimensioni.
  - Il Web può essere rappresentato come una matrice di transizione.
  - La matrice di utilità nel sistema di raccomandazione.
  - Matrici che rappresentano i social network.

- La matrice può essere riassunta da matrici "più strette" vicine all'originale

- Matrici strette
  - Poche righe o poche colonne;
  - molto più efficienti di quelle originali

- Come trovare queste matrici strette:
  - riduzione della dimensionalità

Perche’ la riduzione della dimensionalità?

- Alcune feature possono essere irrilevanti

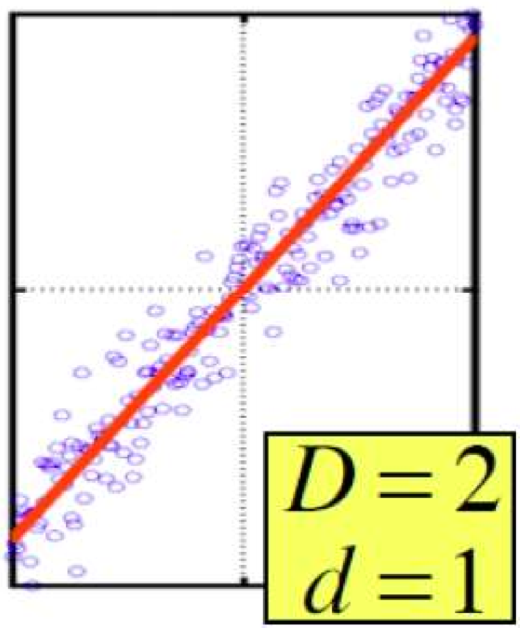
- Necessità di visualizzare dati ad alta dimensione

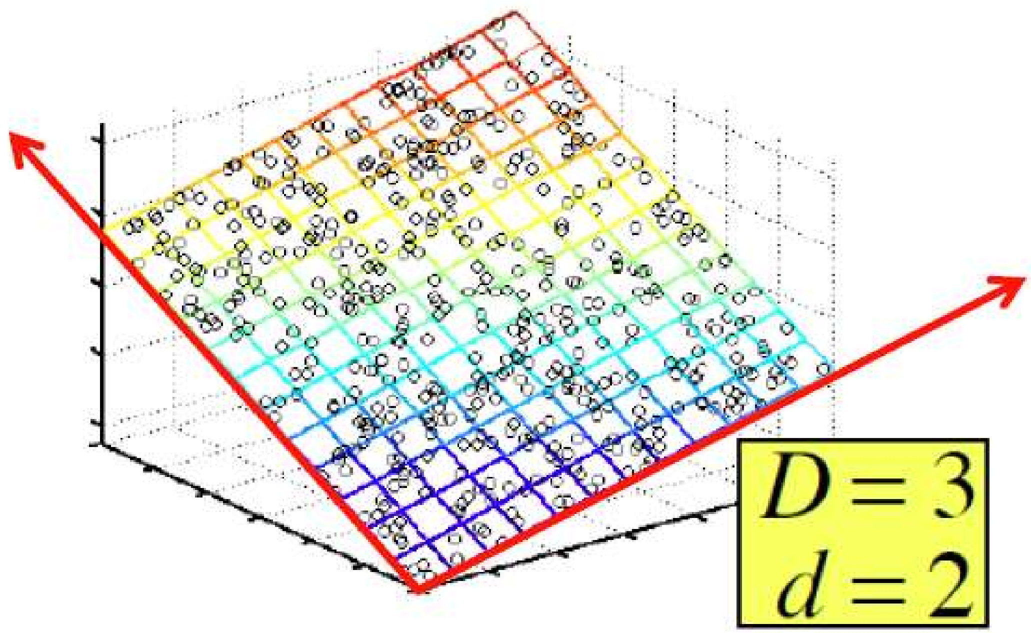
- La dimensionalità “intrinseca” può essere inferiore al numero di feature

Unsupervised feature selection

- Differisce dalla feature selection per due motivi:
  - Invece di selezionare sottoinsiemi di feature
  - Vengono create nuove feature (dimensioni) definite come funzione di tutte le feature
  - Non si considerano etichette di classe ma solo punti di uno spazio multidimensionale

Dimensionality Reduction





- **Assunzione:** I dati cadono su o vicini a un sottospazio *d*-dimensionale
- **Gli assi di questo sottospazio sono l'effettiva rappresentazione dei dati**

Dimensionality Reduction

- **Comprimere / redure la dimensonalità:**
  - 10^6 righe; 10^3 colonne; dati stabili (non aggiornati)
  - Accesso casuale ad una singola cella(e); **errore piccolo: OK**

customer \ day	We 7/10/96	Th 7/11/96	Fr 7/12/96	Sa 7/13/96	Su 7/14/96
ABC Inc.	1	1	1	0	0
DEF Ltd.	2	2	2	0	0
GHI Inc.	1	1	1	0	0
KLM Co.	5	5	5	0	0
Smith	0	0	0	2	2
Johnson	0	0	0	3	3
Thompson	0	0	0	1	1

La matrice di sopra è "2-dimensionale." Tutte le righe possono essere ricostruite a partire da questa base [1 1 1 0 0], [0 0 0 1 1]

Rango di una matrice

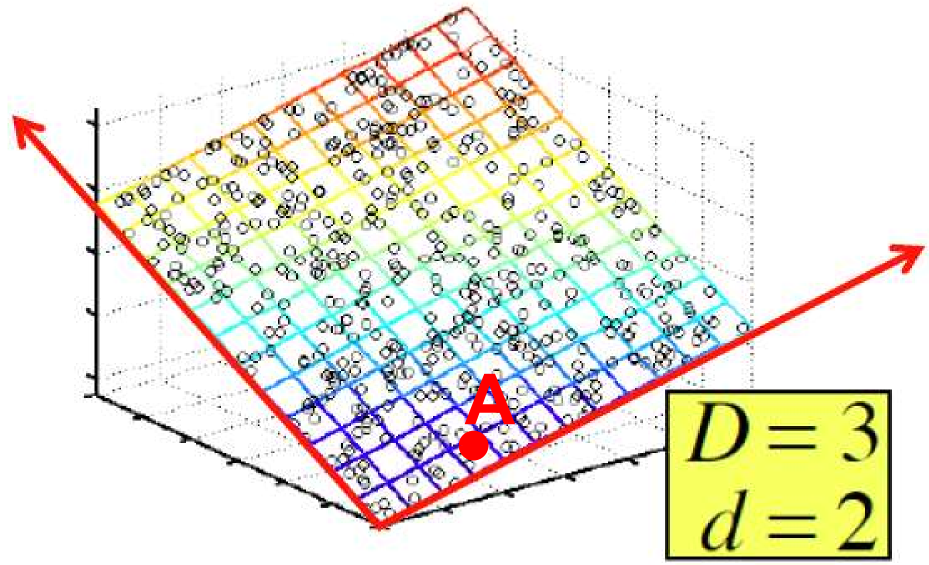
- **Q:** Cos'è il **rango** di una matrice **A**?
- **A:** Il numero colonne **linearmente indipendenti** di **A**
- **Esempio:**
  -
  - **A =** $\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$ ha rango **r=2**

  Perché? Le prime due righe sono linearmente indipendenti quindi li rango è almeno 2, ma tutte e tre le righe sono linearmente dipendenti (la prima è uguale alla somma delle altre due) quindi il rango è minore di 3.
- **Perché ci serve un rango basso?**
  - **A** può essere riscritta con una nuova «base»: [1 2 1] [-2 -3 1]
  - Ottenendo nuove coordinate per le tre righe: [1 0] [0 1] [1 -1]

Il rango è la “dimensione”

- **Nuvola di punti in uno spazio 3D:**
  - Rappresentiamo i punti come matrice:

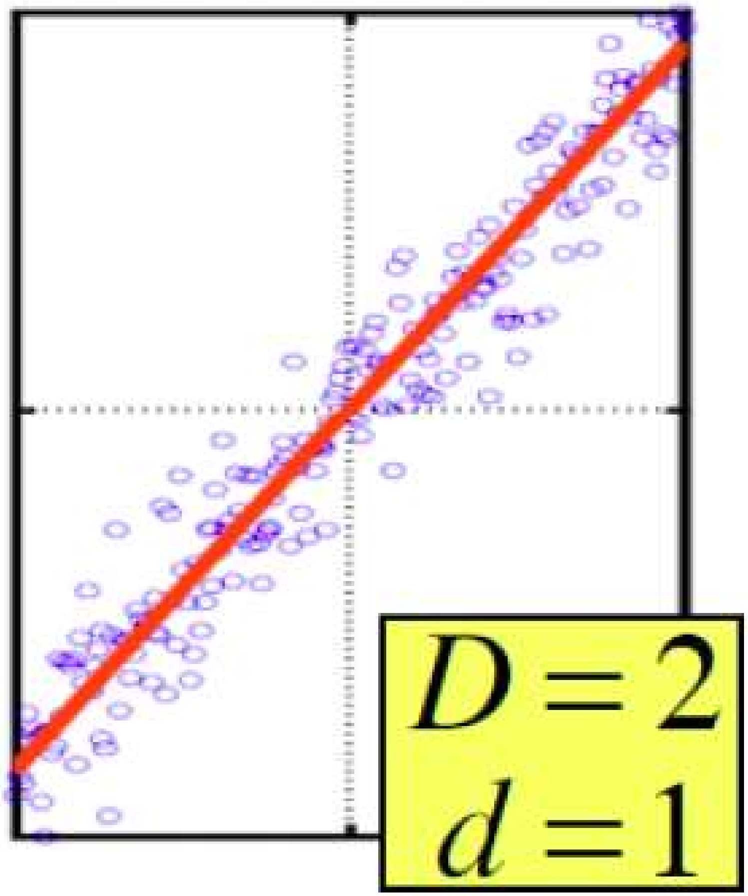
1 riga per punto: $$\begin{array}{cc} \begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix} & \begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{array} \end{array}$$



- **Riscriviamo le coordinate in modo efficiente!**
  - Vecchia base: [1 0 0] [0 1 0] [0 0 1]
  - **Nuova base vettoriale: [1 2 1] [-2 -3 1]**
  - **A** ha le nuove coordinate: [1 0]. **B**: [0 1], **C**: [1- 1]
    - **Nota: abbiamo ridotto il numero di coordinate!**

Dimensionality Reduction

- **Obiettivo della dimensionality reduction è identificare gil assi dei dati!**



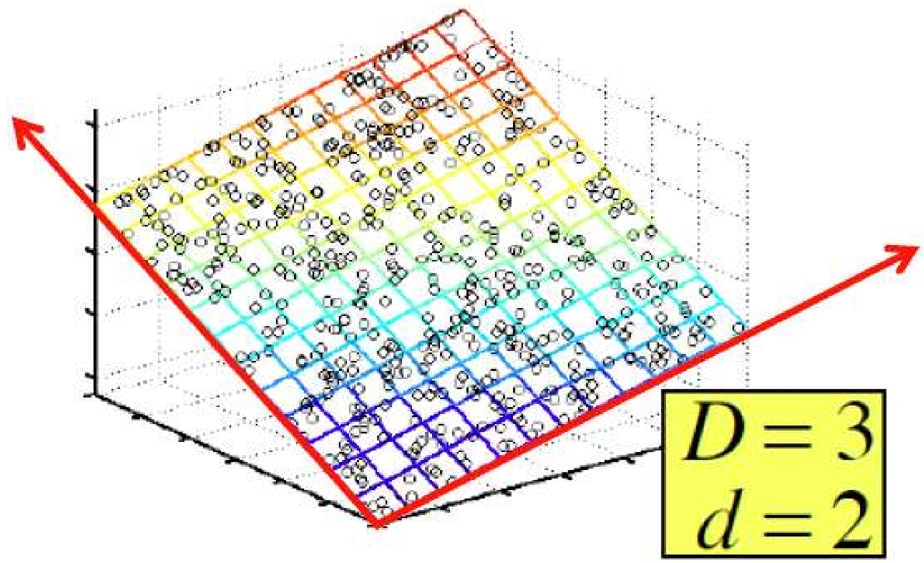
Invece di rappresentare
ogni punto con 2 coordinate
rappresentiamo ogni punto con una
Coordinata (il punto sulla linea rossa)

Introduciamo un piccolo **errore**
In quanto i punti non giacciono
esattamente sulla linea

Perché ridurre le dimensioni?

Perché?

- **Scoprire correlazioni nascoste/argomenti**
  - Parole che si presentano spesso assieme
- **Rimuovere le feature ridondanti e le feature con rumore**
  - Non tutte le parole sono utili
- **Interpretazione e visualizzazione**
- **Piu' semplice immagazzinare e processare i dati**



Unsupervised feature selection

- Idea:
  - Dato un insieme di punti in uno spazio in d-dimensionale,
  - Proiettare i dati in uno spazio con meno dimensioni preservando quanta più informazione possibile
  - Scegliamo la proiezione che **minimizza il quadrato dell'errore** quando ricostruiamo i dati originali

Autovalori e Autovettori (Eigenvalues and Eigenvectors)

- Sia **M** una matrice quadrata. Sia **λ** una costante, sia **e** un vettore colonna non-zero con lo stesso numero di righe di **M**.

- Diciamo che **λ** è un ***autovalore*** di **M** ed **e** è il suo corrispondente ***autovettore*** di **M** se:

$$\mathbf{Me} = \boldsymbol{\lambda}\mathbf{e}$$

- SE **e** è un autovettore di **M** e *c* è una qualsiasi costante, allora anche *c***e** è un autovettore di **M** con lo stesso autovalore.

Autovalori e Autovettori

- Moltiplicare il vettore per una costante cambia la **lunghezza** del vettore ma non la sua direzione.

- Per evitare ambiguita riguardo la lunghezza, assumeremo che ogni autovettore è un **unit vector (vettore unitario):**

- ***la somma dei quadrati delle sue componenti è pari a 1.***

Esempio

$$M = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}$$

One of the eigenvectors of M is

$$\begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}$$

and its corresponding eigenvalue is 7.

We have

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix} \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix} = 7 \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}$$

The eigenvector is a unit vector

$$\left(1/\sqrt{5}\right)^2 + \left(2/\sqrt{5}\right)^2 = 1$$

Calcolare le autocoppie

- ***Autocoppia:*** un autovalore e il suo corrispondente autovettore

- C'è un algoritmo con un running time pari a $O(n^3)$ per il calcolo esatto di tutte le autocoppie in una matrice simmetrica $n \times n$

- Possiamo riscrivere l'equazione $M\mathbf{e} = \lambda\mathbf{e}$ come $(M - \lambda I)\mathbf{e} = 0$, dove:
    1. I è la matrice identità $n \times n$ con 1 nella diagonale principale e 0 nelle altre posizioni.
    2. 0 vettore con tutte le entry pari a 0.

- Affinché $(M - \lambda I)\mathbf{e} = 0$ per un vettore $\mathbf{e} != 0$, il determinante della matrice $M - \lambda I$ deve essere 0.

- Possiamo osservare che $(M - \lambda I)$ è simile alla matrice M, ma se M ha il valore c in un elemento delle diagonale, allora $(M - \lambda I)$ avrà $c - \lambda$ nella stessa posizione.

- Sebbene il determinante di una matrice $n \times n$ abbia $n!$ termini, questo può essere calcolato in diversi modi in $O(n^3)$; Un esempio è il metodo della "**pivotal condensation**" (condensazione pivotale).

Pivotal condensation

Il metodo Chió pivotal condensation consente di calcolare il determinante di una matrice $n \times n$ in funzione di un determinante $(n-1) \times (n-1)$. Abbiamo bisogno che $a_{ii} \neq 0$

$$b_{ij} = a_{11} \times a_{i+1\ j+1} - a_{1\ j+1} \times a_{i+1\ 1}$$

$$\det(A) = \frac{\det(B)}{a_{11}^{n-2}}$$

$$\det(A) = \frac{1}{a_{11}^{n-2}} \begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{1n} \\ a_{21} & a_{2n} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{1n} \\ a_{31} & a_{3n} \end{vmatrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{n1} & a_{n2} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{n1} & a_{n3} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{12} \\ a_{n1} & a_{nn} \end{vmatrix} \end{vmatrix}$$

- Il determinante di **(M − λI)** è un polinomio di grado *n* in λ, dal quale possiamo ottenere gli **n** valori di **λ** che sono gli autovalori di **M**.

- Per uno qualsiasi di questi valori, es. **c**, possiamo quindi risolvere l'equazione M**e** = c**e**.

- So sono **n** equazioni in **n** incognite (gli n componenti di e), poiché non c'e' un termine noto nelle equazioni, possiamo risolvere **e** rispetto a un fattore costante.

- In ogni caso, usando una qualsiasi soluzione, possiamo normalizzare in modo tale che la somma dei quadrati delle entry sia pari a 1 (vettore unitario). In questo modo otterremo l'autovettore che corrisponde all'autovlore c.

Power Iteration

- Sia M una matrice per la quale desideriamo calcolare le autocoppie. Iniziamo con un qualsiasi vettore nonzero x_0 e iteriamo:

$$\mathbf{x}_{k+1} := \frac{M\mathbf{x}_k}{||M\mathbf{x}_k||}$$

- Moltiplichiamo il vettore corrente x_k per la matrice M fino a quando non converge (es. $||x_k - x_{k+1}||$ è al disotto di una certa costante piccolo a piacere). x_k è (approssimativamente) l'autovettore principale di M.

- Per ottenere il corrispondente autovalore calcoliamo $\lambda_1 = x^T M x$

Norma di Frobenius:

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

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.

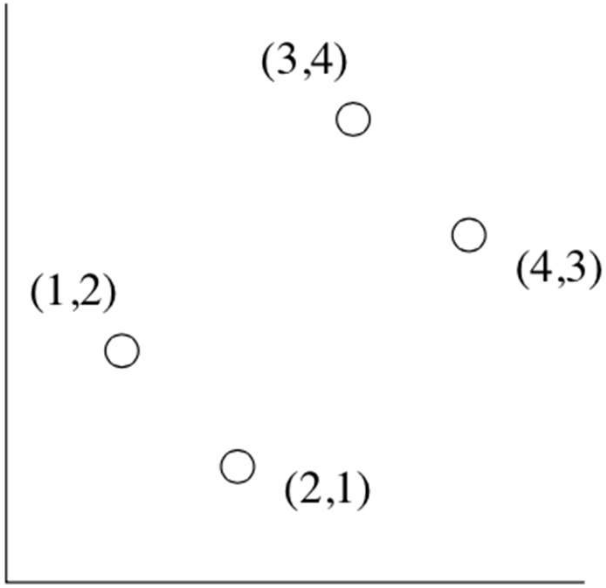
- Trattiamo l'insieme di tuple come una matrice $\mathbf{M}$ e troviamo gli autovettori di $\mathbf{M}\mathbf{M}^T$ o $\mathbf{M}^T\mathbf{M}$.

- La matrice di questi autovettori può essere pensata coma 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



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

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

Trova gli autovalori

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

$$\lambda = 58$$

$$\lambda = 2$$

I corrispondenti autovettori

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

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

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

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

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

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

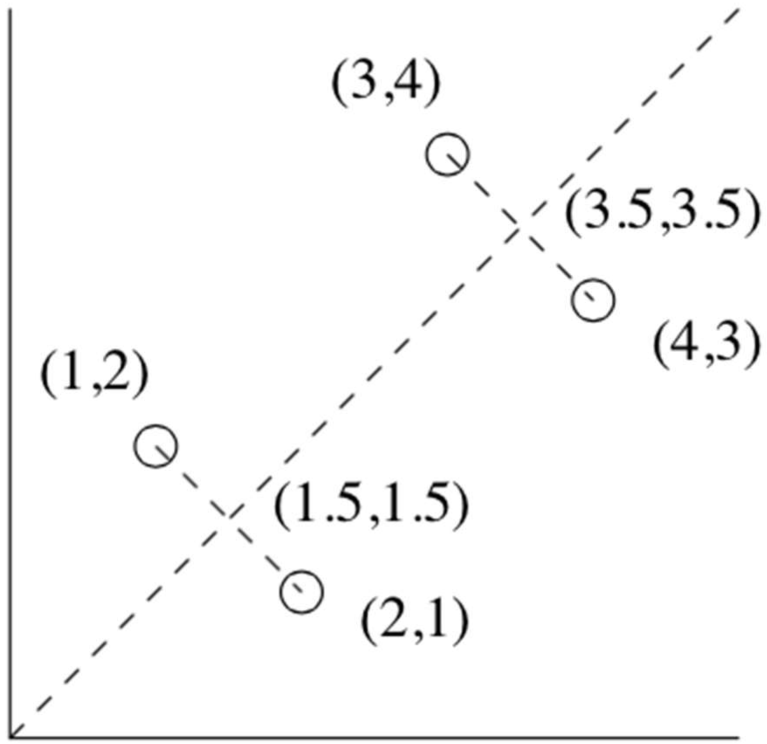
$$28x + 30y = 2y$$

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

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

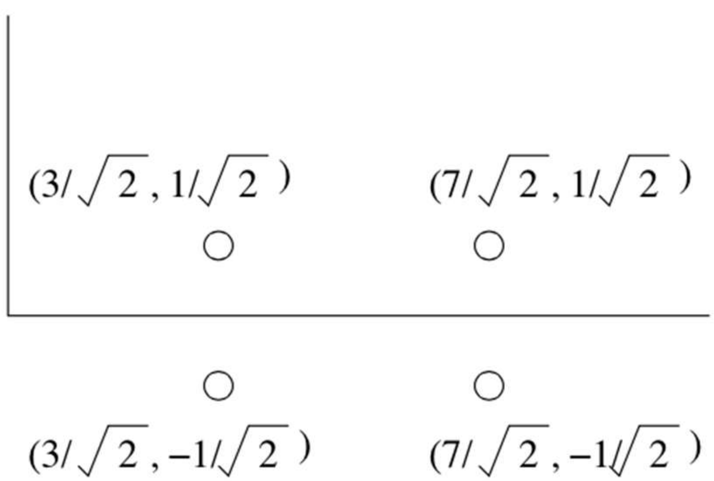
$$E = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

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



- **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