

# SVD randomizzata

Un algoritmo per la riduzione della dimensionalità

Gianluca Covini

Università di Pavia

January 9, 2026

# Indice

- 1 Introduzione
- 2 SVD standard
- 3 Proprietà matematiche
- 4 SVD randomizzata
  - Introduzione
  - Algoritmo
  - Proprietà
  - Implementazione
- 5 Conclusioni e risultati

# Introduzione

L'obiettivo del progetto è presentare la **decomposizione in valori singolari casuale**, un algoritmo per la riduzione della dimensionalità per quando si tratta con dati in dimensione alta.

# SVD

La **decomposizione in valori singolari** (SVD) è una tecnica usata per gestire dati con dimensioni molto alte.

## SVD standard

Data una matrice  $A \in \mathbb{R}^{n \times d}$ , avente rango  $r$ , la sua **decomposizione in valori singolari standard** (versione economica) è la seguente:

$$A = UDV^t$$

Dove  $U \in \mathbb{R}^{n \times r}$  e  $V \in \mathbb{R}^{m \times r}$  con colonne ortogonali e  $D \in \mathbb{R}^{r \times r}$  diagonale.

Notiamo la definizione analoga:  $A = \sum_{i=1}^r d_{ii} \mathbf{u}_i \mathbf{v}_i^t$ .

# Sottospazio best-fit

La proprietà fondamentale della SVD è che permette di calcolare la **miglior approssimazione di rango  $k$**  dei dati, contenuti in una matrice  $A$ .

Spazio *best-fit* di rango  $k$

Il **sottospazio *best-fit* di rango  $k$**  è lo spazio  $\mathcal{L}(\mathbf{v}_1, \dots, \mathbf{v}_k)$  generato dalle prime  $k$  colonne della matrice  $V$ .

Miglior approssimazione di rango  $k$

La **miglior approssimazione di rango  $k$**  della matrice  $A$  è la matrice

$$A_k = \sum_{i=1}^k d_{ii} \mathbf{u}_i \mathbf{v}_i^t$$

# Proprietà

È importante notare alcune proprietà matematiche della SVD. Notiamo innanzitutto che la SVD è la generalizzazione della decomposizione mediante matrici ortogonali delle matrici simmetriche secondo il teorema spettrale.

# Parallelismo con autovalori

A partire da questo parallelismo, studiamo alcune proprietà delle matrici  $AA^t$  e  $A^tA$ .

Si calcola facilmente che

$$AA^t = UD^2U^t \quad e \quad A^tA = VD^2V^t$$

Da cui:

$$AA^tU = UD^2 \quad e \quad A^tAV = VD^2$$

Cioè ogni valore singolare (non nullo) di  $A$  è la radice di un autovalore di  $AA^t$  e  $A^tA$ .

Da ciò segue che se  $A$  è simmetrica i valori singolari sono il valore assoluto degli autovalori di  $A$ .

# Interpretazione intuitiva

Questo ci fornisce un'**interpretazione intuitiva** della SVD: le colonne di  $U$  sono autovettori di  $AA^t$  e le colonne di  $V$  sono autovettori di  $A^tA$ .



# Proprietà

Un'altra proprietà da notare è la seguente:

Le colonne di  $U$  forniscono una base ortonormale per lo spazio delle colonne di  $A$ .

Le colonne di  $V$  forniscono una base ortonormale per lo spazio delle righe di  $A$ .

# Invarianza per trasformazioni ortogonali

Vediamo, infine, come ultima proprietà che la SVD è invariante per trasformazioni ortogonali:

Sia  $B = QA$  con  $Q$  ortogonale.

Vale che:

$$B^t B = A^t Q^t Q A = A^t A$$

Quindi, per quanto osservato precedentemente,  $V$  e  $D$  della decomposizione di  $B$  sono le stesse di  $A$ .

Calcoliamo, ora,  $U_B$ :

$$U_B = B V_A D_A^{-1} = Q U_A$$

Riassumendo:

$$B = QA = Q U_A D_A V_A = U_B D_A V_A$$

# SVD randomizzata

Si è dimostrato che, nel caso in cui la matrice dei dati  $A$  ha rango  $r$  basso, esistono algoritmi di decomposizione molto efficienti basati su metodi randomici. La **SVD randomizzata** è uno di questi.

# Algoritmo: step 1

L'algoritmo si articola in tre step:

**Step 1:** Costruiamo una matrice di proiezione casuale  $P \in \mathbb{R}^{m \times r}$  per lo spazio delle colonne di  $A \in \mathbb{R}^{n \times m}$ :

$$B = AP$$

Dato che  $B$  approssima  $A$  con buona probabilità, possiamo calcolare la decomposizione  $QR$  della matrice  $B$  che ci permette di ottenere una base ortonormale per  $A$ :

$$B = QR$$

# Algoritmo: step 2

**Step 2:** Dato che le colonne di  $Q$  costituiscono una base ortonormale di uno spazio di dimensione  $r \ll m$ , possiamo usare  $Q$  per proiettare  $A$  in uno spazio più piccolo:

$$C = Q^t A$$

Vale che  $A \approx QC$ , con approssimazione maggiore tanto più i valori singolari  $\sigma_k$  decadono rapidamente per  $k > r$ .

Calcoliamo poi la SVD per  $C$ :

$$C = U_C D_C V_C^t$$

# Algoritmo: step 3

**Step 3:** Infine, sfruttiamo le proprietà viste precedentemente per calcolare la SVD di  $A$ .

Vale, infatti, che:

$$A = UDV^t$$

Dove  $U = QU_C$ ,  $D = D_C$ ,  $V = V_C$ .

# Oversampling

Nel caso in cui la matrice  $A$  sia una matrice con valori singolari  $\sigma_k$  non nulli per  $k > r$ , la matrice approssimata  $B$  non genera esattamente lo spazio delle colonne di  $A$ . In questi casi aumentare il numero delle colonne di  $P$  da  $r$  a  $r + p$  migliora significativamente i risultati.

## Oversampling

Parliamo di **oversampling** quando aumentiamo di  $p$  il numero delle colonne di  $P$ .

In generale, maggiore è  $p$ , minore è la discordanza dei valori singolari della matrice approssimata.

# Iterazioni a potenza

Un altro problema si ha quando i valori singolari di  $A$  decadono lentamente: in tal caso i valori che vengono troncati rappresentano una perdita significativa di informazione.

In tal caso si processa  $A$  tramite le **iterazioni a potenza** creando una nuova matrice  $A^{(q)}$ :

$$A^{(q)} = (AA^t)^q A$$

I valori singolari di  $A^{(q)}$  decadono più rapidamente, infatti vale che:

$$A^{(q)} = UD^{2q-1}V^t$$

Tuttavia le iterazioni a potenza sono molto costose.



# Errore

Alcune delle proprietà più importanti della SVD randomizzata è l'esistenza di maggiorazioni e minorazioni dell'errore.

## Stime di errore

Dato  $r$  il rango della SVD desiderata,  $p$  il parametro di oversampling e  $q$  il parametro di iterazioni a potenza, allora valgono le seguenti:

$$\|A - QC\|_2 \geq \sigma_{r+1}(A)$$

$$\mathbb{E}(\|A - QC\|_2) \leq \left(1 + \sqrt{\frac{r}{p-1}} + \frac{e\sqrt{r+p}}{p} \sqrt{m-r}\right)^{\frac{1}{2q+1}} \sigma_{r+1}(A)$$

# Scelta delle proiezioni casuali

Le ultime osservazioni riguardano la scelta di  $P$ .

- Proiezioni casuali gaussiane;
- Matrici casuali uniformi;
- Matrici di Rademacher;
- Matrici sparse;
- Permutazioni della matrice identità.

# Implementazione

Per l'implementazione si rimanda allo script di matlab rSVD.mlx

# Considerazioni

Dai risultati visti notiamo immediatamente che, con una corretta scelta dei parametri di oversampling e di iterazioni a potenza  $p$  e  $q$ , l'algoritmo di rSVD si dimostra molto più efficiente della SVD standard, dal momento che garantisce tempi di esecuzione molto inferiori garantendo, però, un'accuratezza pressoché analoga.

# Considerazioni

Per quanto riguarda la scelta dei parametri notiamo che:

- All'aumentare del rango  $r$  aumenta l'accuratezza ma aumenta anche il tempo d'esecuzione;
- Il numero di iterazioni a potenza  $q$  ottimale nel caso studiato è tra le 2 e le 5. Per valori più piccoli e più grandi l'errore aumenta di molto, e al crescere del parametro  $q$  aumenta tempo di esecuzione;
- All'aumentare del parametro  $p$  l'errore decresce rapidamente e il tempo di esecuzione non aumenta in modo significativo.

# Conclusioni

Possiamo concludere che la SVD randomizzata fornisce un algoritmo molto efficiente per la riduzione della dimensionalità per certi tipi di matrici: i parametri da cui dipende possono essere scelti opportunamente per permettere un errore analogo alla SVD standard ma garantendo un tempo di esecuzione significativamente inferiore.

## Bibliografia:



Brunton, Kutz, *Data Driven Science & Engineering*, Sezioni 1.3, 1.8.