Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

DECOMPOSIZIONE CUR E APPLICAZIONI

Tesi di Laurea in Analisi Numerica

Relatrice:
Chiar.ma Prof.ssa
VALERIA SIMONCINI

Presentata da:
TERESA ARGNANI

VII Sessione
Anno Accademico 2020-2021

Indice

Introduzione

Nell'ambito dell'analisi dati è spesso necessario fare uso di grandi matrici, poiché una matrice di dimensioni $m \times n$ rappresenta la struttura ideale per descrivere m oggetti tramite n caratteristiche.
In molti casi può essere utile trovare il modo per approssimare la matrice dei dati come prodotto di altre matrici affinché essa risulti più semplice da analizzare e interpretare.

In questo senso, la *Decomposizione in Valori Singolari (SVD)* è una decomposizione ampiamente utilizzata. Nel primo capitolo ne verrà data la definizione e verranno enunciati alcuni importanti risultati riguardanti l'accuratezza dell'approssimazione ottenuta troncando la SVD ad un certo numero di termini. Nonostante le proprietà di ottimalità, si osserverà che le matrici che si ottengono da tale decomposizione non risultano particolarmente significative in funzione dei dati stessi o del fenomeno da cui i dati provengono e risulta perciò difficile utilizzarle direttamente per interpretare i dati.

Nel secondo capitolo sarà introdotta la *Decomposizione CUR*, un particolare tipo di decomposizione di rango basso, in cui la matrice di partenza viene espressa in funzione di alcune, poche, righe e colonne della matrice stessa. L'obiettivo è quindi quello di approssimare una matrice $\mathbf{A}$ come prodotto di tre matrici $\mathbf{C}, \mathbf{U}$ ed $\mathbf{R}$, in modo che:

- $\mathbf{C}$ contenga alcune colonne di $\mathbf{A}$;
- $\mathbf{R}$ contenga alcune righe di $\mathbf{A}$;
- $\mathbf{U}$ renda il prodotto $\mathbf{CUR}$ abbastanza vicino a $\mathbf{A}$.

Poiché la Decomposizione CUR è costruita a partire da alcuni dati effettivi della matrice, risulterà possibile fare considerazioni sull'intero dataset a partire da pochi dati, più significativi secondo qualche criterio. Verrà pertanto proposto un algoritmo per la costruzione delle matrici $\mathbf{C}$ ed $\mathbf{R}$, attraverso una scelta opportuna delle righe e delle colonne di $\mathbf{A}$, tramite la tecnica del cosiddetto *Subspace Sampling*.

Nel terzo capitolo verrà presentata l'applicazione della Decomposizione CUR ad un dataset reale e verranno analizzati e discussi i risultati ottenuti.

Notazioni

In questa tesi, matrici e vettori saranno indicati con lettere in grassetto maiuscole e minuscole, rispettivamente.
Data una matrice $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] = [a_{i,j}] \in \mathbb{R}^{m \times n}$, indichiamo con $\mathbf{A}^T$ la sua trasposta e con $\mathrm{rg}(\mathbf{A})$ il suo rango.
Diciamo che la matrice è *simmetrica* se $\mathbf{A}^T = \mathbf{A}$.
Se $\mathbf{A} \in \mathbb{C}^{m \times n}$, indichiamo con $\mathbf{A}^*$ la sua trasposta coniugata.
Indichiamo con $\mathbf{I}_n$ la matrice identità di dimensioni $n \times n$.

Capitolo 1

La Decomposizione in Valori Singolari

1.1 Richiami

Ricordiamo alcune definizioni che risulteranno utili all'interno di questa tesi.

Definizione 1.1. *Una matrice* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *si dice* ortogonale *se*

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}_n.$$

Più in generale, $\mathbf{A} \in \mathbb{C}^{n\times n}$ *si dice* unitaria *se*

$$\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^* = \mathbf{I}_n.$$

Definizione 1.2. *Sia* $\mathbf{x} = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$. *Si definisce la* norma-2 *di* $\mathbf{x}$ *come:*

$$\|\mathbf{x}\|_2 := \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Definizione 1.3. *Una funzione* $\|\cdot\| : \mathbb{C}^{m\times n} \to \mathbb{R}$ *è una* norma di matrice *se, per ogni* $\mathbf{A}, \mathbf{B}$ *con dimensioni compatibili, per ogni* $\alpha \in \mathbb{C}$, *sono soddisfatte le seguenti proprietà:*

1. $\|\mathbf{A}\| \geq 0$ *e* $\|\mathbf{A}\| = 0$ *se e solo se* $\mathbf{A} = \mathbf{0}$;
2. $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|$;
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ *(disuguaglianza triangolare);*
4. $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$ *(sub-moltiplicatività).*

Data $\mathbf{A} = [a_{i,j}] \in \mathbb{C}^{m\times n}$ alcuni esempi di norma di matrice sono:

- La *norma-1*: $\|\mathbf{A}\|_1 = \sum_{j=1}^{n} \sum_{i=1}^{m} |a_{i,j}|$;
- La *norma di Frobenius*: $\|\mathbf{A}\|_F = \left(\sum_{i,j} |a_{i,j}|^2 \right)^{\frac{1}{2}}$;
- La *norma indotta* (da una norma vettoriale $|\cdot|$): $\|\mathbf{A}\| = \max_{|\mathbf{x}|\leq 1} |\mathbf{A}\mathbf{x}| = \max_{|\mathbf{x}|\neq 0} \frac{|\mathbf{A}\mathbf{x}|}{|\mathbf{x}|}$.

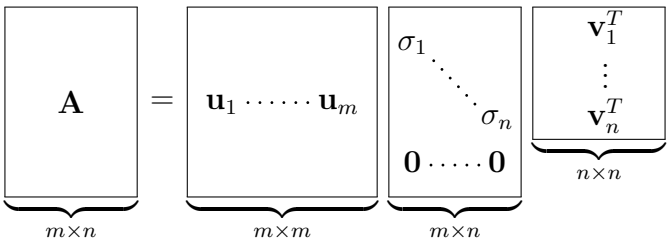
1.2 SVD completa

La *Decomposizione in Valori Singolari* è un tipo di decomposizione matriciale molto generale, che permette di estendere il concetto di *diagonalizzazione* anche per matrici rettangolari, permettendo di prendere due matrici ortogonali *diverse* come matrici di trasformazione [6]. In questa sezione ne sarà data la definizione e saranno mostrati i principali risultati [10].

Teorema 1.4 (Decomposizione in Valori Singolari)**.** *Sia* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *e supponiamo, senza perdere di generalità,* $m \geq n$*. Allora esistono* $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{m\times m}$, $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n] \in \mathbb{R}^{n\times n}$*, e* $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$*, tali che:*

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

dove $\mathbf{U}$*,* $\mathbf{V}$ *sono matrici ortogonali e* $\mathbf{\Sigma}$ *è una matrice diagonale con* $\sigma_1, \ldots \sigma_n$ *sulla diagonale tali che* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$*.*



I vettori $\mathbf{u}_1, \ldots, \mathbf{u}_m$ sono detti *vettori singolari sinistri*, i vettori $\mathbf{v}_1, \ldots, \mathbf{v}_n$ sono detti *vettori singolari destri*, e $\sigma_1, \ldots, \sigma_n$ sono detti *valori singolari*.

Osservazione. Nel caso in cui $m < n$, per trovare la SVD di $\mathbf{A}$ è sufficiente scrivere la decomposizione per $\mathbf{A}^T$ e poi trasporla.
In generale, il costo computazionale di tale decomposizione è $\mathcal{O}\{\min\{mn^2, m^2n\}\}$.

Dimostrazione. Sia $\mathbf{A} \in \mathbb{R}^{m\times n}$, con $m \geq n$. Proviamo che esistono tali $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$.
Se $\mathbf{A}$ è la matrice nulla, allora il risultato è banale.
Sia $\mathbf{A}$ una matrice non nulla e poniamo $\sigma_1 \coloneqq \max\limits_{\|x\|_2=1} \|\mathbf{A}x\|_2$.
Tale massimo esiste, poiché la funzione $x \mapsto \|\mathbf{A}x\|_2$ è continua e l'insieme $\{x \in \mathbb{R}^n : \|x\|_2 = 1\}$ è compatto.
Inoltre $\sigma_1 \geq 0$, ma avendo supposto $\mathbf{A}$ non nulla, sarà $\sigma_1 > 0$.
Sia $\mathbf{v}_1 \in \mathbb{R}^n$ il vettore tale che $\sigma_1 = \|\mathbf{A}\mathbf{v}_1\|_2$ e poniamo $\mathbf{u}_1 \coloneqq \frac{\mathbf{A}\mathbf{v}_1}{\sigma_1}$. In questo modo $\|\mathbf{u}_1\|_2 = 1$.
Definiamo due matrici $\mathbf{U} = [\mathbf{u}_1, \mathbf{U}_2] \in \mathbb{R}^{m\times m}$ e $\mathbf{V} = [\mathbf{v}_1, \mathbf{V}_2] \in \mathbb{R}^{n\times n}$ in modo che siano ortogonali. Chiaramente, la scelta di queste matrici non è univocamente determinata. Allora:

$$\mathbf{A}_1 \coloneqq \mathbf{U}^T\mathbf{A}\mathbf{V} = \begin{bmatrix}\mathbf{u}_1^T \\ \mathbf{U}_2^T\end{bmatrix}\mathbf{A}\begin{bmatrix}\mathbf{v}_1 & \mathbf{V}_2\end{bmatrix} = \begin{bmatrix}\mathbf{u}_1^T \\ \mathbf{U}_2^T\end{bmatrix}\begin{bmatrix}\sigma_1\mathbf{u}_1 & \mathbf{A}\mathbf{V}_2\end{bmatrix} =$$
$$= \begin{bmatrix}\mathbf{u}_1^T\sigma_1\mathbf{u}_1 & \mathbf{u}_1^T\mathbf{A}\mathbf{V}_2 \\ \mathbf{U}_2^T\sigma_1\mathbf{u}_1 & \mathbf{U}_2^T\mathbf{A}\mathbf{V}_2\end{bmatrix} = \begin{bmatrix}\sigma_1 & \mathbf{u}_1^T\mathbf{A}\mathbf{V}_2 \\ \mathbf{0} & \mathbf{U}_2^T\mathbf{A}\mathbf{V}_2\end{bmatrix}$$

dove, nell'ultimo passaggio, abbiamo sfruttato il fatto che $\mathbf{u}_1^T\mathbf{u}_1 = \|\mathbf{u}_1\|_2 = 1$ e che $\mathbf{U}_2^T\mathbf{u}_1 = 0$ in quanto i vettori di $\mathbf{U}_2$ sono ortogonali a $\mathbf{u}_1$.
Osserviamo che:

$$\max_{\|x\|_2=1} \|\mathbf{A}_1x\|_2^2 = \max_{\|x\|_2=1} \|\mathbf{U}^T\mathbf{A}\mathbf{V}x\|_2^2 = \max_{\|x\|_2=1} \|\mathbf{A}x\|_2^2 = \sigma_1^2.$$

Tuttavia, ponendo $\mathbf{z} \coloneqq \mathbf{V}_2^T\mathbf{A}^T\mathbf{u}_1$ e $\mathbf{B} \coloneqq \mathbf{U}_2^T\mathbf{A}\mathbf{V}_2$, si ha che:

$$\frac{1}{\sigma_1^2 + \mathbf{z}^T\mathbf{z}}\left\|\mathbf{A}_1\begin{bmatrix}\sigma_1 \\ \mathbf{z}\end{bmatrix}\right\|_2^2 = \frac{1}{\sigma_1^2 + \mathbf{z}^T\mathbf{z}}\left\|\begin{bmatrix}\sigma_1 & \mathbf{z}^T \\ \mathbf{0} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\sigma_1 \\ \mathbf{z}\end{bmatrix}\right\|_2^2 =$$
$$= \frac{1}{\sigma_1^2 + \mathbf{z}^T\mathbf{z}}\left\|\begin{bmatrix}\sigma_1^2 + \mathbf{z}^T\mathbf{z} \\ \mathbf{B}\mathbf{z}\end{bmatrix}\right\|_2^2 \geq \sigma_1^2 + \mathbf{z}^T\mathbf{z}.$$

Quindi necessariamente $\mathbf{z} = \mathbf{0}$, altrimenti si avrebbe una contraddizione.
E dunque $\mathbf{A}_1 = \begin{bmatrix}\sigma_1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{B}\end{bmatrix}$.
La dimostrazione si conclude per induzione:
assumendo che la matrice $\mathbf{B} \in \mathbb{R}^{(m-1)\times(n-1)}$ abbia Decomposizione in Valori Sin-

golari $\mathbf{B} = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^T$, allora $\mathbf{A}$ ha Decomposizione in Valori Singolari:

$$\begin{aligned}\mathbf{A} = \mathbf{U}\mathbf{A}_1\mathbf{V}^T &= \begin{bmatrix}\mathbf{u}_1 & \mathbf{U}_2\end{bmatrix}\begin{bmatrix}\sigma_1 & \mathbf{0}^T \\ \mathbf{0} & \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^T\end{bmatrix}\begin{bmatrix}\mathbf{v}_1^T \\ \mathbf{V}_2^T\end{bmatrix} = \\ &= \begin{bmatrix}\mathbf{u}_1 & \mathbf{U}_2\end{bmatrix}\begin{bmatrix}1 & \mathbf{0}^T \\ \mathbf{0} & \widetilde{\mathbf{U}}\end{bmatrix}\begin{bmatrix}\sigma_1 & \mathbf{0}^T \\ \mathbf{0} & \widetilde{\mathbf{\Sigma}}\end{bmatrix}\begin{bmatrix}1 & \mathbf{0}^T \\ \mathbf{0} & \widetilde{\mathbf{V}}^T\end{bmatrix}\begin{bmatrix}\mathbf{v}_1^T \\ \mathbf{V}_2^T\end{bmatrix} = \\ &= \underbrace{\begin{bmatrix}\mathbf{u}_1 & \mathbf{U}_2\widetilde{\mathbf{U}}\end{bmatrix}}_{ortogonale}\underbrace{\begin{bmatrix}\sigma_1 & \mathbf{0}^T \\ \mathbf{0} & \widetilde{\mathbf{\Sigma}}\end{bmatrix}}_{diagonale}\underbrace{\begin{bmatrix}\mathbf{v}_1^T \\ \widetilde{\mathbf{V}}^T\mathbf{V}_2^T\end{bmatrix}}_{ortogonale}.\end{aligned}$$

□

Osservazione. Seguono alcune proprietà di tale decomposizione:

- Per come abbiamo definito σ_1 nella dimostrazione precedente, $\sigma_1 = \max_{\|x\|_2=1} \|\mathbf{A}x\|_2$. Si può verificare anche che, se $\mathbf{A}$ è quadrata e non singolare, $\sigma_n = \min_{\|x\|_2=1} \|\mathbf{A}x\|_2$, infatti:
$$\min_{\|x\|_2=1} \|\mathbf{A}x\|_2 = \frac{1}{\|\mathbf{A}^{-1}\|_2} = \frac{1}{\|\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\|_2} = \frac{1}{\|\mathbf{\Sigma}^{-1}\|_2} = \frac{1}{\frac{1}{\sigma_n}} = \sigma_n.$$
- $\|\mathbf{A}\|_2^2 = \max_{\|x\|_2=1} \|\mathbf{A}x\|_2^2 = \max_{\|x\|_2=1} x^T\mathbf{A}^T\mathbf{A}x = \lambda_{max}\left(\mathbf{A}^T\mathbf{A}\right)$, cioè il massimo autovalore di $\mathbf{A}^T\mathbf{A}$. L'ultima uguaglianza segue dalla proprietà del quoziente di Rayleigh.
- In generale, la Decomposizione in Valori Singolari e quella in autovalori sono collegate, nonostante la prima sia molto più generale. Infatti nota la SVD di $\mathbf{A}$ è possibile trovare la decomposizione spettrale di $\mathbf{A}^T\mathbf{A}$:
$$\begin{aligned}\mathbf{A}^T\mathbf{A} &= \left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right)^T\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right) = \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \\ &= \mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_1^T & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_1 \\ \mathbf{0}\end{bmatrix}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}_1^2\mathbf{V}^T,\end{aligned}$$
dove abbiamo denotato con $\mathbf{\Sigma}_1$ la matrice diagonale $n \times n$ tale che $\Sigma = \begin{bmatrix}\mathbf{\Sigma}_1 \\ \mathbf{0}\end{bmatrix}$. Quindi, $\sigma_i(\mathbf{A}) = \sqrt{\lambda_i(\mathbf{A}^T\mathbf{A})}$ per ogni $i \in \{1, \ldots, n\}$, cioè gli autovalori di $\mathbf{A}^T\mathbf{A}$ sono proprio i quadrati dei valori singolari di $\mathbf{A}$.

1.3 SVD troncata

Sia $\mathbf{A} \in \mathbb{R}^{m\times n}$ con $m \geq n$. Per quanto mostrato precedentemente, possiamo trovare la sua Decomposizione in Valori Singolari: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ con

$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m] = [\mathbf{U}_1, \mathbf{U}_2]$, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$ e $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_1 \\ \mathbf{0} \end{bmatrix}$,
dove $\mathbf{U}_1$ è composta dai primi n vettori di $\mathbf{U}$ e $\mathbf{U}_2$ dai restanti $m-n$ vettori di $\mathbf{U}$.
In particolare, possiamo anche scrivere $\mathbf{A} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$, poiché:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}^T = [\mathbf{u}_1, \dots, \mathbf{u}_n] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_n^T \end{bmatrix} =$$

$$= [\mathbf{u}_1, \dots, \mathbf{u}_n] \begin{bmatrix} \sigma_1 \mathbf{v}_1^T \\ \vdots \\ \sigma_n \mathbf{v}_n^T \end{bmatrix} = \sum_{i=1}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T.$$

Se $\exists k \in \{1, \dots, n-1\}$ tale che $\sigma_{k+1} \ll \sigma_k$ allora:

$$\mathbf{A} = \sum_{i=1}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T = \sum_{i=1}^{k} \mathbf{u}_i \sigma_i \mathbf{v}_i^T + \sum_{i=k+1}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \approx \sum_{i=1}^{k} \mathbf{u}_i \sigma_i \mathbf{v}_i^T =: \mathbf{A}_k. \tag{1.1}$$

L'idea è quindi quella di approssimare la matrice $\mathbf{A}$ tenendo conto solo dei primi k valori singolari.
Questa approssimazione di $\mathbf{A}$ è la migliore tra tutte le approssimazioni di rango k di $\mathbf{A}$ [2]:

Teorema 1.5. *Sia* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *tale che* $\mathbf{A} = \sum_{i=1}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T$. *Sia* $k \in \{1, \dots, n-1\}$ *e sia* $\mathbf{A}_k = \sum_{i=1}^{k} \mathbf{u}_i \sigma_i \mathbf{v}_i^T$. *Allora:*

$$\mathbf{A}_k = \underset{\substack{\mathbf{B} \in \mathbb{R}^{m \times n}: \\ \operatorname{rg}(\mathbf{B}) = k}}{\arg\min} \|\mathbf{A} - \mathbf{B}\|_2.$$

Inoltre, $\|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}$.

Dimostrazione. $\operatorname{rg}(\mathbf{A}_k) = k$, poiché $\mathbf{U}^T \mathbf{A}_k \mathbf{V} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}$.

Inoltre $\mathbf{U}^T (\mathbf{A} - \mathbf{A}_k) \mathbf{V} = \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & \sigma_{k+1} \\ & & & & \ddots \\ & & & & & \sigma_n \end{bmatrix}$, quindi $\|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}$.

Sia $\mathbf{B} \in \mathbb{R}^{m \times n} : \operatorname{rg}(\mathbf{B}) = k$. Sia $\{\mathbf{x}_1, \dots, \mathbf{x}_{n-k}\}$ una base dello spazio nullo di $\mathbf{B}$, con $\mathbf{x}_i \in \mathbb{R}^n$.

Allora deve essere $Span\{\mathbf{x}_1, \ldots, \mathbf{x}_{n-k}\} \cap Span\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\} \neq \{\mathbf{0}\}$, poiché al massimo si possono avere n vettori linearmente indipendenti.
Sia dunque $\mathbf{z} \in Span\{\mathbf{x}_1, \ldots, \mathbf{x}_{n-k}\} \cap Span\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\}$ tale che $\mathbf{z} \neq \mathbf{0}$ e supponiamo che abbia norma unitaria. Valgono quindi:

1. $\mathbf{Bz} = \mathbf{0}$;
2. $\mathbf{Az} = \sum_{i=1}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \mathbf{z} = \sum_{i=1}^{k+1} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \mathbf{z} + \underbrace{\sum_{i=k+2}^{n} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \mathbf{z}}_{=\mathbf{0}} = \sum_{i=1}^{k+1} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \mathbf{z}.$

Quindi:

$$\|\mathbf{A} - \mathbf{B}\|_2^2 \geq \|(\mathbf{A} - \mathbf{B})\,\mathbf{z}\|_2^2 = \|\mathbf{Az} - \mathbf{Bz}\|_2^2 = \left\| \sum_{i=1}^{k+1} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \mathbf{z} \right\|_2^2 =$$
$$= \sum_{i=1}^{k+1} \sigma_i^2 \left(\mathbf{v}_i^T \mathbf{z}\right)^2 \geq \sigma_{k+1}^2.$$

Abbiamo visto che tale valore è raggiunto per $\mathbf{B} = \mathbf{A}_k$, dunque il teorema è dimostrato. □

1.4 Considerazioni sull'interpretabilità

La Decomposizione in Valori Singolari di una matrice $\mathbf{A}$ troncata ad un certo numero k di termini rappresenta la migliore approssimazione di rango k di $\mathbf{A}$ e per questo la SVD è una decomposizione ampiamente utilizzata.
Inoltre, essa ha una semplice interpretazione geometrica, in quanto rappresenta come la matrice $\mathbf{A}$ distorce il disco unitario [7].
Infatti, ogni matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ rappresenta una trasformazione lineare:

$$\mathbb{R}^n \longrightarrow \mathbb{R}^m$$
$$x \longmapsto \mathbf{A}x$$

e, poiché $\mathbf{A} = \mathbf{U\Sigma V}^T$, possiamo vedere $\mathbf{A}$ come combinazione di tre trasformazioni: una rotazione/riflessione di $\mathbb{R}^n$ tramite la matrice ortogonale $\mathbf{V}^T \in \mathbb{R}^{n \times n}$, un riscalamento delle coordinate tramite la matrice diagonale $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ e infine una rotazione o una riflessione di $\mathbb{R}^m$ tramite la matrice ortogonale $\mathbf{U} \in \mathbb{R}^{m \times m}$.

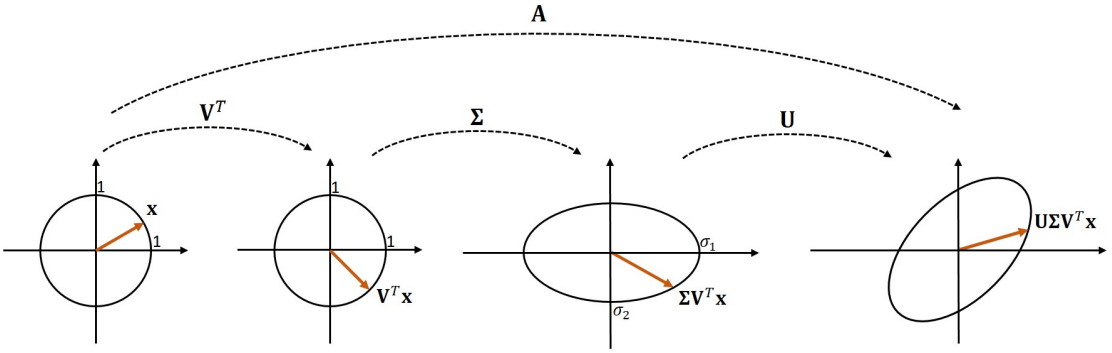


Figura 1.1: In due dimensioni, ogni vettore $\mathbf{x}$ di norma unitaria è mappato su un'ellisse con assi di lunghezza σ_1 e σ_2.

In particolare, per ogni $i = 1, \ldots, n$:

$$\mathbf{A}\mathbf{v}_i = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{v}_i = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_1 \\ \mathbf{0}\end{bmatrix}\mathbf{V}^T\mathbf{v}_i = [\mathbf{u}_1, \ldots, \mathbf{u}_n]\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_n\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^T \\ \vdots \\ \mathbf{V}_n^T\end{bmatrix}\mathbf{v}_i =$$

$$= [\mathbf{u}_1, \ldots, \mathbf{u}_n]\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_n\end{bmatrix}\mathbf{e}_i = [\mathbf{u}_1, \ldots, \mathbf{u}_n]\begin{bmatrix}0 & & & \\ & \ddots & & \\ & 0 & \sigma_i & \\ & & 0 & \ddots \\ & & & 0\end{bmatrix} = \mathbf{u}_i\sigma_i.$$

Ovvero, per ogni trasformazione lineare da $\mathbb{R}^n$ a $\mathbb{R}^m$ si può trovare, grazie alla SVD, una base di vettori ortogonali che, tramite l'applicazione indotta dalla matrice $\mathbf{A}$, vengono mappati in vettori ancora ortogonali tra loro. In particolare l'i-esimo vettore singolare destro viene trasformato in un multiplo dell'i-esimo vettore singolare sinistro.

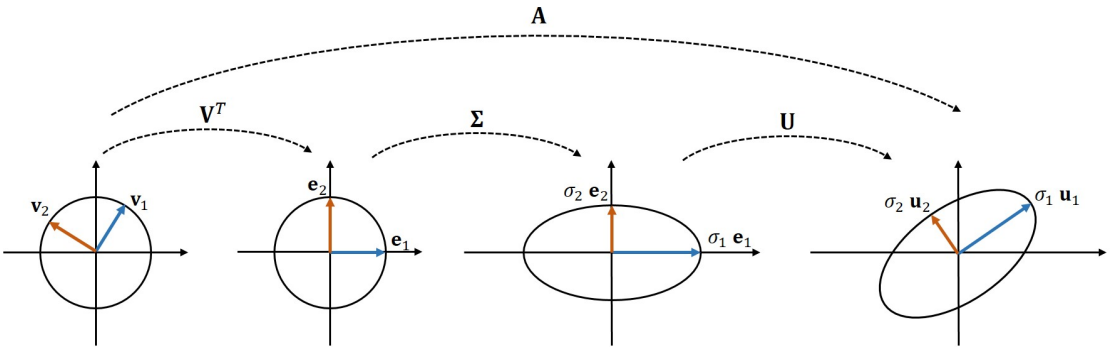


Figura 1.2: In due dimensioni, i vettori singolari destri vengono trasformati, tramite $\mathbf{A}$, in multipli dei vettori singolari sinistri.

Abbiamo quindi visto l'interpretazione geometrica della Decomposizione in Valori Singolari, tuttavia i vettori $\mathbf{u}_i$ e $\mathbf{v}_i$ non hanno necessariamente significato in

termini dei dati da cui essi provengono. Questo perché i vettori singolari sono astrazioni matematiche che possono essere calcolate per qualsiasi matrice, non sono "oggetti" con un significato "fisico".
In alcuni casi particolari è anche possibile dare un'interpretazione alle prime componenti singolari. Per esempio, in un dataset di punti del piano estratti da una distribuzione normale multivariata, le due componenti principali sono proprio le direzioni degli assi dell'ellissoide da cui sono stati presi i dati, ma nella maggior parte dei casi questo non accade, per esempio se i punti del piano sono estratti dall'unione di due distribuzioni normali, le componenti principali non hanno alcun significato "reale" [4].

Capitolo 2

La Decomposizione CUR

2.1 Formulazione del problema

In questo capitolo vogliamo descrivere un'approssimazione di rango basso di una matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ che sia strettamente legata ai dati che essa rappresenta. Cerchiamo una decomposizione che abbia buone proprietà algoritmiche e garanzie sull'errore relativo e che, a differenza della SVD permetta di preservare anche eventuali buone proprietà della matrice di partenza, come la non negatività o la sparsità [1].
La decomposizione proposta è la *Decomposizione CUR*, che definiamo come un caso particolare delle cosiddette *Approssimazioni per Colonne*:

Definizione 2.1. *Sia* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *e sia* $\mathbf{C} \in \mathbb{R}^{m \times c}$ *una matrice costituita da* c *colonne di* $\mathbf{A}$*. Diciamo che la matrice* $\mathbf{A'} = \mathbf{CX}$ *è una* Approssimazione per Colonne *di* $\mathbf{A}$ *(o una* Decomposizione CX *di* $\mathbf{A}$*) per ogni* $\mathbf{X} \in \mathbb{R}^{c \times n}$*.*

Osservazione. Ogni colonna di $\mathbf{A'}$ è espressa come combinazione lineare di alcune colonne di $\mathbf{A}$. Nelle applicazioni in generale è di interesse avere $c \ll n$, per questo spesso si sceglie $c = \mathcal{O}(\log(n))$.

Definizione 2.2. *Sia* $\mathbf{A} \in \mathbb{R}^{m \times n}$*. Si definisce la* pseudoinversa di Moore-Penrose *di* $\mathbf{A}$ *come la matrice* $\mathbf{A}^+$ *che soddisfa le seguenti proprietà:*

1. $\mathbf{AA}^+$ *e* $\mathbf{A}^+\mathbf{A}$ *sono simmetriche;*
2. $\mathbf{AA}^+\mathbf{A} = \mathbf{A}$*;*
3. $\mathbf{A}^+\mathbf{AA}^+ = \mathbf{A}^+$*.*

In particolare, se $\mathbf{A}$ *è invertibile, allora* $\mathbf{A}^+ = \mathbf{A}^{-1}$*.*
La matrice $\mathbf{P_A} \coloneqq \mathbf{AA}^+$ *è la* matrice di proiezione *sul sottospazio generato dalle colonne di* $\mathbf{A}$*.*

Osservazione. La matrice pseudoinversa di $\mathbf{A}$ esiste sempre ed è unica [2]. Infatti, nota la SVD di $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, la pseudoinversa di $\mathbf{A}$ si determina come

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T,$$

dove $\mathbf{\Sigma}^+$ è ottenuta da $\mathbf{\Sigma}$ prendendo i reciproci degli elementi diagonali non nulli, lasciando gli zeri invariati e facendo la trasposta.

Teorema 2.3 (Migliore Approssimazione per Colonne di una matrice)**.** *Sia* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, sia* $\mathbf{C} \in \mathbf{R}^{m\times c}$ *una matrice formata da c colonne di* $\mathbf{A}$*. L'approssimazione* $\mathbf{P_C A} = \mathbf{C}\mathbf{C}^+\mathbf{A}$ *è la migliore Approssimazione per Colonne di* $\mathbf{A}$*, nel senso che:*

$$\left\|\mathbf{A} - \mathbf{C}\left(\mathbf{C}^+\mathbf{A}\right)\right\|_F = \min_{\mathbf{X}\in\mathbb{R}^{c\times n}} \|\mathbf{A} - \mathbf{C}\mathbf{X}\|_F.$$

Questo teorema segue dai seguenti risultati:

Proposizione 1. *Sia* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, sia* $\mathbf{b} \in \mathbb{R}^m$*. Allora il vettore* $\mathbf{x}_{opt} = \mathbf{A}^+\mathbf{b}$ *è il vettore di norma minima che realizza:*

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Dimostrazione. Possiamo scrivere $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Cercare il vettore $\mathbf{x}_{opt}$ che realizzi $\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ e che abbia norma minima è equivalente a cercare il vettore che realizzi:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} - \mathbf{b}\right\|_2^2 = \min_{\mathbf{x}\in\mathbb{R}^n} \left\|\mathbf{U}\left(\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} - \mathbf{U}^T\mathbf{b}\right)\right\|_2^2 = \min_{\mathbf{x}\in\mathbb{R}^n} \left\|\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} - \mathbf{U}^T\mathbf{b}\right\|_2^2,$$

cioè è equivalente a cercare il vettore $\widetilde{\mathbf{w}} = \mathbf{V}^T\mathbf{x}$ di norma euclidea minima tale che:

$$\left\|\mathbf{\Sigma}\widetilde{\mathbf{w}} - \mathbf{U}^T\mathbf{b}\right\|_2^2 \leq \left\|\mathbf{\Sigma}\mathbf{y} - \mathbf{U}^T\mathbf{b}\right\|_2^2 \text{ per ogni } \mathbf{y} \in \mathbb{R}^n.$$

Se $\mathbf{w} = [w_1, \ldots, w_n]^T$ e $\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_r & \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix}$ allora vale:

$$\left\|\mathbf{\Sigma}\mathbf{w} - \mathbf{U}^T\mathbf{b}\right\|_2^2 = \sum_{i=1}^{r} \left(\sigma_i w_i - (\mathbf{U}^T\mathbf{b})_i\right)^2 + \sum_{i=r+1}^{m} \left(\mathbf{U}^T\mathbf{b}\right)_i^2,$$

dunque tale norma è minima se $\sigma_i w_i - (\mathbf{U}^T\mathbf{b})_i = 0$, cioè se $w_i = \frac{(\mathbf{U}^T\mathbf{b})_i}{\sigma_i}$ per ogni $i = 1, \ldots, r$. Tra tutti i vettori $\mathbf{w}$ di questa forma, chiaramente quello di norma minima è quello in cui le restanti $n - r$ componenti sono nulle, ovvero proprio $\widetilde{\mathbf{w}} = \mathbf{\Sigma}^+\mathbf{U}^T\mathbf{b}$ e quindi $\mathbf{x}_{opt} = \mathbf{V}\widetilde{\mathbf{w}} = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T\mathbf{b} = \mathbf{A}^+\mathbf{b}$. □

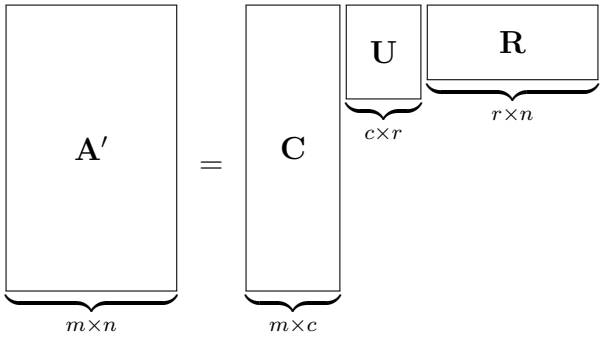
Proposizione 2. *Sia* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *con* $m \geq n$ *e sia* $\mathbf{B} \in \mathbb{R}^{m\times p}$*. Allora* $\mathbf{X}_{opt} = \mathbf{A}^+\mathbf{B}$ *è l'unica soluzione del problema di minimo:*

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times p}} \|\mathbf{B} - \mathbf{A}\mathbf{X}\|_F.$$

Dimostrazione. Si utilizza il risultato della proposizione precedente per risolvere il problema di minimo $\min_{\mathbf{x}_j \in \mathbb{R}^n} \|\mathbf{b}_j - \mathbf{A}\mathbf{x}_j\|_2$ per ogni colonna di $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ e si ottengono $\mathbf{x}_j = \mathbf{A}^+\mathbf{b}_j$, ovvero le colonne di $\mathbf{X}_{opt}$ da cui si ottiene la tesi. □

Definizione 2.4. *Sia* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, sia* $\mathbf{C} \in \mathbb{R}^{m\times c}$ *formata da* c *colonne di* $\mathbf{A}$*, sia* $\mathbf{R} \in \mathbb{R}^{r\times n}$ *formata da* r *righe di* $\mathbf{A}$*. Diciamo che la matrice* $\mathbf{A}' = \mathbf{CUR}$ *è una* Approssimazione per Righe-Colonne *di* $\mathbf{A}$*, per ogni* $\mathbf{U} \in \mathbb{R}^{c\times r}$*.*

Osservazione. L'Approssimazione per Righe-Colonne è un particolare tipo di Decomposizione CX in cui ogni colonna di $\mathbf{A}'$ si può scrivere come combinazione lineare di alcune colonne di $\mathbf{A}$, usando solo le informazioni contenute in un piccolo numero di righe di $\mathbf{A}$ e una matrice $\mathbf{U}$ di dimensioni più ridotte. Si notino infatti le dimensioni di tali matrici:



Teorema 2.5 (Migliore Approssimazione per Righe-Colonne di una matrice)**.** *Siano* $\mathbf{A} \in \mathbb{R}^{m\times n}$*,* $\mathbf{C} \in \mathbb{R}^{m\times c}$ *formata da* c *colonne di* $\mathbf{A}$ *e* $\mathbf{R} \in \mathbb{R}^{r\times n}$ *formata da* r *righe di* $\mathbf{A}$*. L'approssimazione* $\mathbf{A}' = \mathbf{C}\left(\mathbf{C}^+\mathbf{A}\mathbf{R}^+\right)\mathbf{R}$ *è la migliore Approssimazione per Righe-Colonne di* $\mathbf{A}$*, nel senso che:*

$$\left\|\mathbf{A} - \mathbf{C}\left(\mathbf{C}^+\mathbf{A}\mathbf{R}^+\right)\mathbf{R}\right\|_F = \min_{\mathbf{U}\in\mathbb{R}^{c\times r}} \|\mathbf{A} - \mathbf{CUR}\|_F.$$

Dimostrazione. Dal Teorema 2.3 segue che $\mathbf{X}_{opt} = \mathbf{C}^+\mathbf{A}$ realizza:

$$\min_{\mathbf{X}\in\mathbb{R}^{c\times n}} \|\mathbf{A} - \mathbf{CX}\|_F.$$

Cerchiamo dunque un'approssimazione per righe di $\mathbf{X}_{opt}$:

$$\min_{\mathbf{U}\in\mathbb{R}^{c\times r}} \|\mathbf{X}_{opt} - \mathbf{UR}\|_F = \min_{\mathbf{U}\in\mathbb{R}^{c\times r}} \left\|\mathbf{X}_{opt}^T - \mathbf{R}^T\mathbf{U}^T\right\|_F$$

Dalla Proposizione 2 segue che la soluzione di tale problema è:

$$\mathbf{U}_{opt}^T = (\mathbf{R}^T)^+\mathbf{X}_{opt}^T = (\mathbf{R}^+)^T\mathbf{X}_{opt}^T = \left(\mathbf{X}_{opt}\mathbf{R}^+\right)^T = \left(\mathbf{C}^+\mathbf{A}\mathbf{R}^+\right)^T.$$

Dunque $\mathbf{U}_{opt} = \mathbf{C}^+\mathbf{A}\mathbf{R}^+$. □

Definizione 2.6. *Sia* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, sia* $\mathbf{C} \in \mathbb{R}^{m\times c}$ *formata da* c *colonne di* $\mathbf{A}$*, sia* $\mathbf{R} \in \mathbb{R}^{r\times n}$ *formata da* r *righe di* $\mathbf{A}$ *e sia* $\mathbf{U} = \mathbf{C}^{+}\mathbf{A}\mathbf{R}^{+}$*. Allora l'Approssimazione per Righe-Colonne* $\mathbf{A}' = \mathbf{CUR}$ *è detta* Decomposizione CUR *di* $\mathbf{A}$*.*

2.2 Subspace Sampling

Affinché sia possibile dare garanzie sull'errore relativo della Decomposizione CUR di una matrice $\mathbf{A}$, occorre che le colonne di $\mathbf{A}$ scelte per costituire la matrice $\mathbf{C}$ (e analogamente le righe di $\mathbf{A}$ scelte per costituire la matrice $\mathbf{R}$) siano selezionate in modo opportuno.

Osservazione. Nel seguito della trattazione si discuterà solo della selezione delle colonne di $\mathbf{A}$, poiché per la selezione delle righe sarà sufficiente applicare ragionamenti analoghi alla trasposta di $\mathbf{A}$.

Nei primi articoli sull'argomento, sono stati proposti algoritmi in cui le colonne di $\mathbf{A}$ erano estratte in modo casuale, secondo una distribuzione di probabilità che dipendeva dalla norma euclidea delle colonne stesse [4].
Per fornire una buona approssimazione di rango k della matrice $\mathbf{A}$, illustriamo nel seguito la tecnica del *Subspace Sampling* attraverso la quale le colonne di $\mathbf{A}$ sono estratte in modo casuale secondo una distribuzione di probabilità che dipende dalla norma euclidea dei primi k vettori singolari destri di $\mathbf{A}$.
L'idea è quella di calcolare per ogni colonna di $\mathbf{A}$ un "coefficiente di importanza" di tale colonna, che rappresenti quanto essa influisce nella migliore approssimazione di rango k della matrice, e di utilizzare tali coefficienti per costruire una distribuzione di probabilità sulle colonne di $\mathbf{A}$. Vediamo dunque come definirli:

Abbiamo già osservato nella sezione 1.3 che, noti i valori singolari σ_i e i vettori singolari destri e sinistri, $\mathbf{v}_i$ e $\mathbf{u}_i$ della matrice, possiamo scrivere $\mathbf{A} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ e che possiamo approssimare $\mathbf{A}$ tenendo conto solo dei primi k termini:

$$\mathbf{A} \approx \mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

da cui in particolare segue che la colonna j-esima di $\mathbf{A}$, $\mathbf{a}_j$, si può approssimare come:

$$\mathbf{a}_j \approx \sum_{i=1}^{k} \sigma_i \mathbf{u}_i v_{j,i},$$

dove $v_{j,i}$ è l'elemento j-esimo dell'i-esimo vettore singolare destro.

Definizione 2.7. *Sia* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *e sia* $\mathbf{A}_k = \sum_{i=1}^{k} \mathbf{u}_i \sigma_i \mathbf{v}_i^T$ *la sua Decomposizione in Valori Singolari approssimata al rango* k*. Definiamo la* probabilità di

campionamento *della colonna j-esima come:*

$$p_j \coloneqq \frac{1}{k}\sum_{i=1}^{k}(v_{j,i})^2 \text{ per ogni } j = 1, \dots, n.$$

Tali p_j sono anche detti *leverage scores* e, per come li abbiamo normalizzati, osserviamo che valgono:

1. $p_j \geq 0$ per ogni $j = 1, \dots, n$;
2. $\sum_{j=1}^{n} p_j = \sum_{j=1}^{n} \frac{1}{k} \sum_{i=1}^{k} (v_{j,i})^2 = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} (v_{j,i})^2 \overset{*}{=} \frac{1}{k} \sum_{i=1}^{k} 1 = 1$,
   dove $*$ segue dall'ortonormalità dei vettori singolari destri.

Dunque i coefficienti p_j definiscono effettivamente una distribuzione di probabilità sulle colonne di $\mathbf{A}$.

Osservazione. Il costo computazionale per il calcolo dei leverage scores è dell'ordine del costo computazionale per il calcolo dalla SVD troncata ai primi k termini.

Definita tale probabilità di campionamento, possiamo descrivere l'algoritmo per la selezione delle colonne.

Gli algoritmi

L'algoritmo per la scelta delle colonne, `COLUMNSELECT`, prende in input una matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$, k un parametro di rango, ε un parametro di errore e restituisce in output una matrice $\mathbf{C} \in \mathbb{R}^{m \times c}$ costituita da un numero $c = \mathcal{O}\left(k \log k/\varepsilon^2\right)$ di colonne di $\mathbf{A}$, dove la colonna j-esima di $\mathbf{A}$ viene selezionata per costituire $\mathbf{C}$ con probabilità $\widetilde{p_j} = \min\{1, p_j c\}$ Questa scelta di c fornisce garanzie sull'errore relativo dell'approssimazione, come mostreremo nel seguito.
L'algoritmo `ALGORITHMCUR`, per determinare la Decomposizione CUR della matrice, dati in input la matrice $\mathbf{A}$, k il parametro di rango ed ε il parametro di errore, restituisce $\mathbf{C}, \mathbf{R}$ ed $\mathbf{U}$, richiamando l'algoritmo `COLUMNSELECT` due volte, una volta con in input $\mathbf{A}$, per selezionare le colonne, e una volta con input $\mathbf{A}^T$, per selezionare le righe.

Introduciamo inoltre un particolare formalismo matriciale, legato all'algebra lineare, per la costruzione delle matrici $\mathbf{C}$ ed $\mathbf{R}$ che sarà utile per mostrare i risultati successivi. Data $\mathbf{A} \in \mathbb{R}^{m \times n}$ e $\mathbf{C} \in \mathbb{R}^{m \times c}$ costituita da c colonne di $\mathbf{A}$, scriveremo $\mathbf{C}$ come:

$$\mathbf{C} = \mathbf{A}\mathbf{S}_C\mathbf{D}_C,$$

dove $\mathbf{S}_C \in \mathbb{R}^{n \times n}$ è detta *matrice di sampling* e ha ogni colonna formata da zeri tranne che per un elemento, $\mathbf{D}_C \in \mathbb{R}^{n \times c}$ è detta *matrice di scaling* ed è una matrice con elementi non nulli solo sulla diagonale.

Poiché per campionare r righe di $\mathbf{A}$ basta applicare gli stessi algoritmi su $\mathbf{A}^T$ e trasporre quanto ottenuto, si ha che:

$$\mathbf{R} = \left(\mathbf{A}^T\mathbf{S}_R\mathbf{D}_R\right)^T = \mathbf{D}_R{}^T\mathbf{S}_R{}^T\mathbf{A}$$

con $\mathbf{R} \in \mathbb{R}^{r\times n}$, $\mathbf{S}_R \in \mathbb{R}^{n\times n}$ e $\mathbf{D}_R \in \mathbb{R}^{n\times r}$.

Algoritmo 1: SAMPLING

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, p_j probabilità di campionamento della colonna j-esima per ogni $j = 1, \dots, n$, $\widetilde{c} \leq n$, il numero di colonne che si vogliono estrarre in media.
Output: $\mathbf{S}_C \in \mathbb{R}^{n\times n}$ matrice di sampling, $\mathbf{D}_C \in \mathbb{R}^{n\times c}$ matrice di scaling, dove c è il numero di colonne effettivamente selezionate.
Procedimento:

1: inizializza $\mathbf{S}_C$ e $\mathbf{D}_C$ matrici di zeri;
2: $t = 1$;
3: **per** $j = 1, \dots, n$ **esegue:**
4: sceglie j con probabilità $\widetilde{p_j} = \min\{1, \widetilde{c}p_j\}$;
5: **se** j è stato scelto **allora:**
6: $(\mathbf{S}_C)_{j,t} = 1$;
7: $(\mathbf{D}_C)_{t,t} = \frac{1}{\min\{1, \sqrt{\widetilde{c}p_j}\}}$;
8: $t = t + 1$;
9: restituisce le matrici $\mathbf{S}_C$ e $\mathbf{D}_C$.

Algoritmo 2: COLUMNSELECT

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, k il parametro di rango, ε il parametro di errore.
Output: La matrice $\mathbf{C} \in \mathbb{R}^{m\times c}$, che contiene le c colonne riscalate di $\mathbf{A}$, $\mathbf{S}_C \in \mathbb{R}^{n\times n}$ la matrice di sampling, $\mathbf{D}_C \in \mathbb{R}^{n\times c}$ la matrice di scaling.
Procedimento:

1: calcola i primi k vettori singolari destri di $\mathbf{A}$, $\mathbf{v}_1, \dots, \mathbf{v}_k$;
2: calcola la probabilità p_j di campionamento della colonna $j-$esima di $\mathbf{A}$, per $j = 1, \dots n$;
3: esegue l'algoritmo `SAMPLING` con input $\mathbf{A}$, p_j con $j = 1, \dots, n$ e $\widetilde{c} = \mathcal{O}(k \log k/\varepsilon^2)$ per costruire le matrici di sampling e scaling, $\mathbf{S}_C$ e $\mathbf{D}_C$;
4: restituisce $\mathbf{C} = \mathbf{A}\mathbf{S}_C\mathbf{D}_C$, $\mathbf{S}_C$ e $\mathbf{D}_C$.

Osservazione. Nella definizione della Decomposizione CUR abbiamo posto $\mathbf{U} = \mathbf{C}^+\mathbf{A}\mathbf{R}^+$, dunque, con le notazioni introdotte si ha che:

$$\begin{aligned}\mathbf{U} &= (\mathbf{A}\mathbf{S}_C\mathbf{D}_C)^+\mathbf{A}\left(\mathbf{D}_R{}^T\mathbf{S}_R{}^T\mathbf{A}\right)^+ = (\mathbf{S}_C\mathbf{D}_C)^+\mathbf{A}^+\mathbf{A}\mathbf{A}^+\left(\mathbf{D}_R{}^T\mathbf{S}_R{}^T\right)^+ = \\ &= (\mathbf{S}_C\mathbf{D}_C)^+\mathbf{A}^+\left(\mathbf{D}_R{}^T\mathbf{S}_R{}^T\right)^+ = \left(\left(\mathbf{D}_R{}^T\mathbf{S}_R{}^T\right)\mathbf{A}(\mathbf{S}_C\mathbf{D}_C)\right)^+ \end{aligned} \tag{2.1}$$

ovvero, $\mathbf{U}$ è la pseudo-inversa della matrice costituita dalle r righe scelte delle c colonne di $\mathbf{A}$ selezionate.

Nella sezione successiva, mostreremo i principali risultati riguardanti l'errore dell'approssimazione di $\mathbf{A}$ ottenuta con questi algoritmi [1].

Algoritmo 3: ALGORITHMCUR

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, k il parametro di rango, ε il parametro di errore.
Output: $\mathbf{C} \in \mathbb{R}^{m\times c}$, $\mathbf{R} \in \mathbb{R}^{r\times n}$ e $\mathbf{U} \in \mathbb{R}^{r\times c}$.
Procedimento:

1: esegue l'algoritmo `COLUMNSELECT` con input $\mathbf{A}, k, \varepsilon$ e restituisce $\mathbf{C} \in \mathbf{R}^{m\times c}$ con $c = \mathcal{O}(k \log k/\varepsilon^2)$ colonne di $\mathbf{A}$, $\mathbf{S}_C$ matrice di sampling e $\mathbf{D}_C$ matrice di scaling;
2: esegue l'algoritmo `COLUMNSELECT` con input $\mathbf{A}^T, c, \varepsilon$, traspone quanto ottenuto e restituisce $\mathbf{R} \in \mathbb{R}^{r\times n}$ con $r = \mathcal{O}(c \log c/\varepsilon^2)$ righe di $\mathbf{A}$, ${\mathbf{S}_R}^T$ e ${\mathbf{D}_R}^T$;
3: costruisce $\mathbf{W} \in \mathbb{R}^{r\times c}$ con le r righe di $\mathbf{C}$ che sono state selezionate per formare $\mathbf{R}$, cioè $\mathbf{W} = \left({\mathbf{D}_R}^T {\mathbf{S}_R}^T\right) \mathbf{A} (\mathbf{S}_C \mathbf{D}_C)$;
4: restituisce $\mathbf{U} = \mathbf{W}^+$.

2.3 Analisi dell'errore

Definizione 2.8. *Siano $\mathbf{A} \in \mathbb{R}^{m\times n}$, con $m \geq n$, $\mathbf{b} \in \mathbb{R}^m$. Poiché il sistema $\mathbf{Ax} = \mathbf{b}$ risulta sovradeterminato, in generale non è possibile trovare $\widetilde{\mathbf{x}} \in \mathbb{R}^n$ tale che $\mathbf{A}\widetilde{\mathbf{x}} = \mathbf{b}$, per questo si è interessati a risolvere il* problema ai minimi quadrati*, cioè a determinare il vettore $\mathbf{x}_{opt} \in \mathbb{R}^n$ che minimizzi la norma euclidea del residuo $r = \mathbf{b} - \mathbf{Ax}$, cioè che realizzi:*

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{b} - \mathbf{Ax}\|_2.$$

Abbiamo già visto (Proposizione 1) che il vettore che minimizza $\|\mathbf{b} - \mathbf{Ax}\|_2$ è $\mathbf{x}_{opt} = \mathbf{A}^+\mathbf{b}$.
Questo problema è importante per la Decomposizione CX, e di conseguenza per la Decomposizione CUR, poiché data una matrice $\mathbf{A} \in \mathbb{R}^{m\times n}$ e un insieme di sue colonne $\mathbf{C} \in \mathbb{R}^{m\times c}$, ci interessa risolvere un problema di minimo del tipo:

$$\min_{\mathbf{X}\in\mathbb{R}^{c\times n}} \|\mathbf{A} - \mathbf{CX}\|_F,$$

che è un problema ai minimi quadrati multidimensionale. É noto (dalla Proposizione 2) che la matrice $\mathbf{X}_{opt} = \mathbf{C}^+\mathbf{A}$ realizza tale minimo.
Vogliamo mostrare che:

1. La matrice $\mathbf{C}$ ottenuta tramite l'algoritmo `COLUMNSELECT` garantisce che l'errore relativo in norma $\|\mathbf{A} - \mathbf{CX}_{opt}\|_F = \|\mathbf{A} - \mathbf{CC}^+\mathbf{A}\|_F$ non sia troppo distante dalla norma dell'errore ottenuto con la migliore approssimazione di rango k di $\mathbf{A}$, cioè $\|\mathbf{A} - \mathbf{A}_k\|_F$.

2. Le matrici $\mathbf{C}$, $\mathbf{R}$ ed $\mathbf{U}$ ottenute tramite l'algoritmo `ALGORITHMCUR` garantiscono che l'errore relativo in norma $\|\mathbf{A} - \mathbf{CUR}\|_F$ non sia troppo distante da $\|\mathbf{A} - \mathbf{CC}^+\mathbf{A}\|_F$.

Descriviamo ora un algoritmo che ci sarà utile per mostrare questi risultati. Sia il problema ai minimi quadrati multidimensionale:

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times p}} \|\mathbf{B}-\mathbf{A}\mathbf{X}\|_F \tag{2.2}$$

con $\mathbf{A}\in\mathbb{R}^{m\times n}$ tale che $\mathrm{rg}(\mathbf{A})\leq k$, $\mathbf{B}\in\mathbb{R}^{m\times p}$.
Siano $p_1,\ldots,p_m$ le probabilità di campionamento delle righe di $\mathbf{A}$ e sia r intero positivo, allora l'algoritmo `APPROXMINQUAD` restituisce la soluzione del problema ai minimi quadrati

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times p}} \left\|\mathbf{D}^T\mathbf{S}^T\mathbf{B}-\left(\mathbf{D}^T\mathbf{S}^T\mathbf{A}\right)\mathbf{X}\right\|_F, \tag{2.3}$$

dove al posto di $\mathbf{A}$ si considera la matrice $\mathbf{D}^T\mathbf{S}^T\mathbf{A}$, cioè la matrice che contiene in media r righe di $\mathbf{A}$ e al posto di $\mathbf{B}$ si considera la matrice $\mathbf{D}^T\mathbf{S}^T\mathbf{B}$, cioè la matrice che contiene le corrispondenti righe di $\mathbf{B}$, dove $\mathbf{S}\in\mathbb{R}^{m\times m}$ e $\mathbf{D}\in\mathbb{R}^{m\times r}$ si ottengono con l'algoritmo `SAMPLING` descritto in precedenza.
Dunque l'algoritmo `APPROXMINQUAD` restituisce:

$$\widetilde{\mathbf{X}}_{opt}=\left(\mathbf{D}^T\mathbf{S}^T\mathbf{A}\right)^+\left(\mathbf{D}^T\mathbf{S}^T\mathbf{B}\right).$$

Algoritmo 4: APPROXMINQUAD

Input: $\mathbf{A}\in\mathbb{R}^{m\times n}$, $\mathbf{B}\in\mathbb{R}^{m\times p}$, p_i vettore che contiene le probabilità di campionamento di ogni riga, $r\leq m$.
Output: $\widetilde{\mathbf{X}}_{opt}\in\mathbb{R}^{n\times p}$.
Procedimento:

1: esegue l'algoritmo `SAMPLING` con input $\mathbf{A}^T$, le probabilità di campionamento delle righe di $\mathbf{A}$ $p_1,\ldots,p_m$ e restituisce la matrice di sampling $\mathbf{S}$ e la matrice di scaling $\mathbf{D}$;
2: costruisce la matrice $\mathbf{D}^T\mathbf{S}^T\mathbf{A}$ che contiene le righe selezionate di $\mathbf{A}$ e la matrice $\mathbf{D}^T\mathbf{S}^T\mathbf{B}$ che contiene le corrispondenti righe di $\mathbf{B}$;
3: restituisce $\widetilde{\mathbf{X}}_{opt}$ soluzione del problema di minimo $\min_{\mathbf{x}\in\mathbb{R}^{n\times p}} \left\|\mathbf{D}^T\mathbf{S}^T\mathbf{B}-\mathbf{D}^T\mathbf{S}^T\mathbf{A}\mathbf{X}\right\|_F$ ovvero $\widetilde{\mathbf{X}}_{opt}=(\mathbf{D}^T\mathbf{S}^T\mathbf{A})^+(\mathbf{D}^T\mathbf{S}^T\mathbf{B})$.

Vogliamo ora mostrare che tale $\widetilde{\mathbf{X}}_{opt}$ fornisce una buona approssimazione della soluzione del problema originario 2.2:

Teorema 2.9. *Siano $\mathbf{A}\in\mathbb{R}^{m\times n}$, tale che $\mathrm{rg}(\mathbf{A})\leq k$, $\mathbf{B}\in\mathbb{R}^{m\times p}$, $p_1,\ldots,p_m$ le probabilità di campionamento delle righe di $\mathbf{A}$ un parametro di errore $\varepsilon\in(0,1]$ e $r=\mathcal{O}\left(k\log k/\varepsilon^2\right)$ input di `APPROXMINQUAD`. Allora l'algoritmo restituisce $\widetilde{\mathbf{X}}_{opt}\in\mathbb{R}^{n\times p}$ tale che, con probabilità almeno 0.7 vale:*

$$\left\|\mathbf{B}-\mathbf{A}\widetilde{\mathbf{X}}_{opt}\right\|_F\leq(1+\varepsilon)\min_{\mathbf{x}\in\mathbb{R}^{n\times p}}\|\mathbf{B}-\mathbf{A}\mathbf{X}\|_F=(1+\varepsilon)\|\mathbf{B}-\mathbf{A}\mathbf{X}_{opt}\|_F, \tag{2.4}$$

con $\mathbf{X}_{opt}=\mathbf{A}^+\mathbf{B}$ e $\widetilde{\mathbf{X}}_{opt}=(\mathbf{D}^T\mathbf{S}^T\mathbf{A})^+(\mathbf{D}^T\mathbf{S}^T\mathbf{B})$.

Enunciamo alcuni lemmi che utilizzeremo per la dimostrazione:
Sia $\mathbf{A} \in \mathbb{R}^{m\times n}$ di rango $\rho \leq k$, e sia la sua SVD: $\mathbf{A} = \mathbf{U}_A\mathbf{\Sigma}_A{\mathbf{V}_A}^T$. Indichiamo la sua SVD ridotta ai primi k termini in questo modo: $\mathbf{A}_k = \mathbf{U}_{A,k}\mathbf{\Sigma}_{A,k}{\mathbf{V}_{A,k}}^T$.
Consideriamo inoltre $r = \mathcal{O}\left(k \log k/\varepsilon^2\right)$, dove $\varepsilon \in (0,1]$ è il parametro di errore. Sia $\mathbf{D}^T\mathbf{S}^T\mathbf{A}$ la matrice che contiene r righe riscalate di $\mathbf{A}$, con $\mathbf{D}$ e $\mathbf{S}$ ottenute tramite l'algoritmo di sampling precedentemente descritto, con in input le probabilità di campionamento delle righe di $\mathbf{A}$. Per semplificare la notazione d'ora in poi poniamo $\mathcal{S} \coloneqq \mathbf{D}^T\mathbf{S}^T$. Con queste ipotesi valgono i seguenti risultati:

Lemma 1. *Posto* $\mathbf{\Omega} \coloneqq (\mathcal{S}\mathbf{U}_{A,k})^+ - (\mathcal{S}\mathbf{U}_{A,k})^T$, *allora con probabilità almeno 0.9 valgono:*

1. $\widetilde{\rho} \coloneqq \operatorname{rg}(\mathcal{S}\mathbf{U}_{A,k}) = \operatorname{rg}(\mathbf{U}_{A,k}) = \operatorname{rg}(\mathbf{A}_k) = \rho$;
2. $\|\mathbf{\Omega}\|_2 = \left\|{\mathbf{\Sigma}_{\mathcal{S}U_{A,k}}}^{-1} - \mathbf{\Sigma}_{\mathcal{S}U_{A,k}}\right\|_2$;
3. $(\mathcal{S}\mathbf{A}_k)^+ = \mathbf{V}_{A,k}{\mathbf{\Sigma}_{A,k}}^{-1}(\mathcal{S}\mathbf{U}_{A,k})^+$;
4. $\left\|{\mathbf{\Sigma}_{\mathcal{S}U_{A,k}}}^{-1} - \mathbf{\Sigma}_{\mathcal{S}U_{A,k}}\right\|_2 \leq \frac{\varepsilon}{\sqrt{2}}$.

Lemma 2. *Con probabilità almeno 0.9 vale:*

$$\left\|{\mathbf{U}_{A,k}}^T\mathcal{S}^T\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}\left({\mathbf{U}_{A,k}}^{\perp}\right)^T\mathbf{B}\right\|_F \leq \frac{\varepsilon}{2}\left\|{\mathbf{U}_{A,k}}^{\perp}\left({\mathbf{U}_{A,k}}^{\perp}\right)^T\mathbf{B}\right\|_F$$

Lemma 3. *Con probabilità almeno 0.9 vale:*

$$\left\|\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}\left({\mathbf{U}_{A,k}}^{\perp}\right)^T\mathbf{B}\right\|_F \leq 10\left\|{\mathbf{U}_{A,k}}^{\perp}\left({\mathbf{U}_{A,k}}^{\perp}\right)^T\mathbf{B}\right\|_F$$

Dimostrazione del Teorema 2.9. Vogliamo trovare una maggiorazione di $\left\|\mathbf{B} - \mathbf{A}\widetilde{\mathbf{X}}_{opt}\right\|_F$ in termini di $\|\mathbf{B} - \mathbf{A}\mathbf{X}_{opt}\|_F$.

$$\begin{aligned}
\mathbf{B} - \mathbf{A}\widetilde{\mathbf{X}}_{opt} &= \mathbf{B} - \mathbf{A}_k\widetilde{\mathbf{X}}_{opt} = \mathbf{B} - \mathbf{A}_k(\mathcal{S}\mathbf{A}_k)^+(\mathcal{S}\mathbf{B}) = \qquad (2.5)\\
&= \mathbf{B} - \mathbf{U}_{A,k}\mathbf{\Sigma}_{A,k}{\mathbf{V}_{A,k}}^T\mathbf{V}_{A,k}{\mathbf{\Sigma}_{A,k}}^{-1}(\mathcal{S}\mathbf{U}_{A,k})^+\mathcal{S}\mathbf{B} =\\
&= \mathbf{B} - \mathbf{U}_{A,k}(\mathcal{S}\mathbf{U}_{A,k})^+\mathcal{S}\mathbf{B} =\\
&= \mathbf{B} - \mathbf{U}_{A,k}(\mathcal{S}\mathbf{U}_{A,k})^+\mathcal{S}\underbrace{\left(\mathbf{U}_{A,k}{\mathbf{U}_{A,k}}^T + \mathbf{U}_{A,k}{}^{\perp}({\mathbf{U}_{A,k}}^{\perp})^T\right)}_{\mathbf{I}_n}\mathbf{B} =\\
&= \mathbf{B} - \mathbf{U}_{A,k}(\mathcal{S}\mathbf{U}_{A,k})^+\mathcal{S}\mathbf{U}_{A,k}{\mathbf{U}_{A,k}}^T\mathbf{B} - \mathbf{U}_{A,k}(\mathcal{S}\mathbf{U}_{A,k})^+\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}({\mathbf{U}_{A,k}}^{\perp})^T\mathbf{B} = \qquad (2.6)\\
&= \mathbf{B} - \mathbf{U}_{A,k}{\mathbf{U}_{A,k}}^T\mathbf{B} - \mathbf{U}_{A,k}\underbrace{(\mathcal{S}\mathbf{U}_{A,k})^+}_{\mathbf{\Omega}+\left(\mathcal{S}\mathbf{U}_{A,k}\right)^T}\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}({\mathbf{U}_{A,k}}^{\perp})^T\mathbf{B} =\\
&= \mathbf{U}^{\perp}_{A,k}{\mathbf{U}^{\perp}_{A,k}}^T\mathbf{B} - \mathbf{U}_{A,k}\mathbf{\Omega}\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}({\mathbf{U}_{A,k}}^{\perp})^T\mathbf{B} - \mathbf{U}_{A,k}(\mathcal{S}\mathbf{U}_{A,k})^T\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}({\mathbf{U}_{A,k}}^{\perp})^T\mathbf{B}
\end{aligned}$$

dove in 2.5 abbiamo usato il risultato 3 del Lemma 1 e in 2.6 abbiamo usato che $(\mathcal{S}\mathbf{U}_{A,k})^{+}\mathcal{S}\mathbf{U}_{A,k} = \mathbf{V}_{\mathcal{S}\mathbf{U}_{A,k}}\mathbf{V}_{\mathcal{S}\mathbf{U}_{A,k}}{}^{T} = \mathbf{I}_{\rho}$; quest'ultimo risultato non è vero in generale, ma segue dal fatto che $\rho = \widetilde{\rho}$ per il punto 1 del Lemma 1.
Applicando la norma di Frobenius da entrambe le parti e utilizzando la disuguaglianza triangolare, possiamo maggiorare $\left\|\mathbf{B} - \mathbf{A}_k\widetilde{\mathbf{X}}_{opt}\right\|_F$ con:

$$\left\|\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F + \left\|\mathbf{U}_{A,k}(\mathcal{S}\mathbf{U}_{A,k})^{T}\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F + \left\|\mathbf{U}_{A,k}\mathbf{\Omega}\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F$$

che a sua volta possiamo maggiorare con:

$$\left\|\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F + \left\|\mathbf{U}_{A,k}{}^{T}\mathcal{S}^{T}\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F + \|\mathbf{\Omega}\|_2\left\|\mathcal{S}\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F$$

dove abbiamo sfruttato la sub-moltiplicatività della norma e il fatto le colonne di $\mathbf{U}_{A,k}$ sono ortogonali.
A questo punto i risultati dei Lemmi 1, 2 e 3 ci consentono di maggiorare quanto ottenuto con:

$$\left\|\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F + \frac{\varepsilon}{2}\left\|\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F + 10\frac{\varepsilon}{\sqrt{2}}\left\|\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F.$$

E dunque si ottiene:

$$\begin{aligned}\left\|\mathbf{B} - \mathbf{A}_k\widetilde{\mathbf{X}}_{opt}\right\|_F &\leq \left(1 + \frac{\varepsilon}{\sqrt{2}} + \frac{10\varepsilon}{\sqrt{2}}\right)\left\|\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F \leq \\ &\leq (1 + 8\varepsilon)\left\|\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B}\right\|_F\end{aligned}$$

da cui segue il risultato cercato, ponendo $\varepsilon' = 8\varepsilon$,
poiché $\mathbf{U}_{A,k}{}^{\perp}\left(\mathbf{U}_{A,k}{}^{\perp}\right)^{T}\mathbf{B} = \mathbf{B} - \mathbf{A}\mathbf{X}_{opt}$, infatti:

$$\begin{aligned}\mathbf{B} - \mathbf{A}\mathbf{X}_{opt} = \mathbf{B} - \mathbf{A}_k\mathbf{X}_{opt} &= \mathbf{B} - \mathbf{A}_k\mathbf{A}_k{}^{+}\mathbf{B} = \left(1 - \mathbf{A}_k\mathbf{A}_k{}^{+}\right)\mathbf{B} = \\ &= \left(1 - (\mathbf{U}_{A,k}\mathbf{\Sigma}_{A,k}\mathbf{V}_{A,k}{}^{T})(\mathbf{V}_{A,k}\mathbf{\Sigma}_{A,k}{}^{-1}\mathbf{U}_{A,k}^{T})\right)\mathbf{B} = \\ &= \left(1 - (\mathbf{U}_{A,k}\mathbf{U}_{A,k}^{T})\right)\mathbf{B} = \mathbf{U}_{A,k}{}^{\perp}(\mathbf{U}_{A,k}^{\perp})^{T}\mathbf{B}.\end{aligned}$$

Poiché ciascuno dei risultati dei tre Lemmi non è verificato con probabilità inferiore a 0.1, allora la probabilità che almeno uno dei tre non valga è minore di 0.3 e dunque la probabilità che siano verificati tutti e tre simultaneamente è almeno 0.7. □

Grazie a questo teorema possiamo mostrare i due risultati seguenti:

Teorema 2.10. *Siano $\mathbf{A} \in \mathbb{R}^{m\times n}$, k un parametro di rango ed $\varepsilon \in (0, 1]$ un parametro di errore gli input di* `COLUMNSELECT`*. Allora l'algoritmo restituisce $\mathbf{C} \in \mathbb{R}^{m\times c}$ con $c = \mathcal{O}\left(k \log k/\varepsilon^2\right)$ tale che, con probabilità almeno 0.7 vale:*

$$\left\|\mathbf{A} - \mathbf{C}\mathbf{C}^{+}\mathbf{A}\right\|_F \leq (1 + \varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F, \tag{2.7}$$

dove $\mathbf{A}_k$ è la migliore approssimazione di rango k di $\mathbf{A}$.

Dimostrazione. L'algoritmo `COLUMNSELECT` restituisce $\mathbf{C} = \mathbf{AS}\mathbf{D} = \mathbf{A}\mathcal{S}^T$ e $\mathbf{X}_{opt} = \mathbf{C}^+\mathbf{A}$ è la matrice che minimizza $\|\mathbf{A} - \mathbf{C}\mathbf{X}\|_F$. Allora:

$$\begin{aligned}\left\|\mathbf{A} - \mathbf{C}\left(\mathbf{C}^+\mathbf{A}\right)\right\|_F &= \left\|\mathbf{A} - (\mathbf{A}\mathcal{S}^T)(\mathbf{A}\mathcal{S}^T)^+\mathbf{A}\right\|_F \leq \\ &\leq \left\|\mathbf{A} - (\mathbf{A}\mathcal{S}^T)(\mathbf{P}_{\mathbf{A}_k}\mathbf{A}\mathcal{S}^T)^+\mathbf{P}_{\mathbf{A}_k}\mathbf{A}\right\|_F \end{aligned} \tag{2.8}$$

dove $\mathbf{P}_{\mathbf{A}_k} = \mathbf{U}_{A,k}{\mathbf{U}_{A,k}}^T$ è la proiezione sui primi k vettori singolari sinistri di $\mathbf{A}$. Per il teorema 2.9 applicato al problema di minimo $\min \left\|\mathbf{A}^T - {\mathbf{A}_k}^T\mathbf{X}\right\|_F$, vale:

$$\left\|\mathbf{A}^T - {\mathbf{A}_k}^T(\mathcal{S}{\mathbf{A}_k}^T)^+(\mathcal{S}\mathbf{A}^T)\right\|_F \leq (1+\varepsilon)\left\|\mathbf{A}^T - {\mathbf{A}_k}^T({\mathbf{A}_k}^T)^+\mathbf{A}^T\right\|_F$$

e, di conseguenza:

$$\begin{aligned}\left\|\left(\mathbf{A}^T - {\mathbf{A}_k}^T(\mathcal{S}{\mathbf{A}_k}^T)^+(\mathcal{S}\mathbf{A}^T)\right)^T\right\|_F &\leq (1+\varepsilon)\left\|\left(\mathbf{A}^T - {\mathbf{A}_k}^T({\mathbf{A}_k}^T)^+\mathbf{A}^T\right)^T\right\|_F \\ \left\|\mathbf{A} - \mathbf{A}\mathcal{S}^T(\mathbf{A}_k\mathcal{S}^T)^+\mathbf{A}_k\right\|_F &\leq (1+\varepsilon)\left\|\mathbf{A} - \mathbf{A}{\mathbf{A}_k}^+\mathbf{A}_k\right\|_F. \end{aligned} \tag{2.9}$$

poiché $\mathbf{P}_{\mathbf{A}_k}\mathbf{A} = \mathbf{A}_k$.
Allora, da 2.8 e 2.9 segue che:

$$\left\|\mathbf{A} - \mathbf{C}\left(\mathbf{C}^+\mathbf{A}\right)\right\|_F \leq (1+\varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F.$$

□

Tale risultato si può generalizzare affinché valga con probabilità almeno $1-\delta$.

Corollario (Generalizzazione del Teorema 2.10)**.** *Siano* $\mathbf{A} \in \mathbb{R}^{m\times n}$, $k \ll \min\{m,n\}$, $\varepsilon \in (0,1]$, $\delta \in (0,1]$ *e* $c = \mathcal{O}\left(k\log k\log\left(1/\delta\right)/\varepsilon^2\right)$ *il numero di colonne scelte in media da* `COLUMNSELECT` *per costruire* $\mathbf{C}$*. Allora con probabilità almeno* $1-\delta$ *vale:*

$$\left\|\mathbf{A} - \mathbf{C}\left(\mathbf{C}^+\mathbf{A}\right)\right\|_F \leq (1+\varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F. \tag{2.10}$$

Dimostrazione. L'idea è quella di eseguire l'algoritmo `COLUMNSELECT` $\ln\left(1/\delta\right)$ volte in modo indipendente, e di restituire la matrice $\mathbf{C}$ che minimizza $\|\mathbf{A} - \mathbf{C}\mathbf{C}^+\mathbf{A}\|_F$. Allora, poiché la probabilità che non valga la disuguaglianza 2.7 è minore di $0.3 < 1/e$ per ogni esecuzione dell'algoritmo, la probabilità che l'algoritmo fallisca ogni volta è minore $(1/e)^{\ln(1/\delta)} = \delta$. □

Teorema 2.11. *Siano* $\mathbf{A} \in \mathbb{R}^{m\times n}$*,* k *un parametro di rango ed* $\varepsilon \in (0,1]$ *un parametro di errore gli input dell'algoritmo* `ALGORITHMCUR`*. Allora l'algoritmo restituisce* $\mathbf{C} \in \mathbb{R}^{m\times c}$ *con* $c = \mathcal{O}\left(k\log k/\varepsilon^2\right)$*,* $\mathbf{R} \in \mathbb{R}^{r\times n}$ *con* $r = \mathcal{O}\left(c\log c/\varepsilon^2\right)$ *e* $\mathbf{U} \in \mathbb{R}^{r\times c}$ *tali che, con probabilità almeno 0.7 vale:*

$$\|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{R}\|_F \leq (1+\varepsilon)\left\|\mathbf{A} - \mathbf{C}\mathbf{C}^+\mathbf{A}\right\|_F. \tag{2.11}$$

Dimostrazione. Dal teorema 2.3 segue che:

$$\left\|\mathbf{A}-\mathbf{C}\left(\mathbf{C}^{+}\mathbf{A}\right)\right\|_F=\min_{\mathbf{X}\in\mathbb{R}^{c\times n}}\|\mathbf{A}-\mathbf{C}\mathbf{X}\|_F.$$

Quindi per il teorema 2.9:

$$\left\|\mathbf{A}-\mathbf{C}\left(\left(\mathcal{S}^T\mathbf{C}\right)^{+}\left(\mathcal{S}^T\mathbf{A}\right)\right)\right\|_F\leq(1+\varepsilon)\left\|\mathbf{A}-\mathbf{C}\left(\mathbf{C}^{+}\mathbf{A}\right)\right\|_F,$$

dove $\mathcal{S}\mathbf{A}=\mathbf{R}$ e $\left(\mathcal{S}^T\mathbf{C}\right)^{+}=\mathbf{U}$, per come tali matrici sono costruite in `ALGORITHMCUR`, per cui il teorema è dimostrato. □

Corollario (Generalizzazione del Teorema 2.11)**.** *Sia* $\mathbf{A}\in\mathbb{R}^{m\times n}$*, sia* $k\ll\min\{m,n\}$*,* $\varepsilon\in(0,1]$*,* $\delta\in(0,1]$ $c=\mathcal{O}\left(k\log k\log\left(1/\delta\right)/\varepsilon^2\right)$ *il numero di colonne scelte in media per costruire* $\mathbf{C}$ *e sia* $r=\mathcal{O}\left(c\log c\log\left(1/\delta\right)/\varepsilon^2\right)$ *il numero di righe scelte in media da* ***ALGORITHMCUR*** *per costruire* $\mathbf{R}$ *. Allora con probabilità almeno* $1-\delta$ *vale:*

$$\|\mathbf{A}-\mathbf{C}\mathbf{U}\mathbf{R}\|_F\leq(1+\varepsilon)\left\|\mathbf{A}-\mathbf{C}\mathbf{C}^{+}\mathbf{A}\right\|_F. \tag{2.12}$$

Dimostrazione. L'idea è quella di eseguire l'algoritmo `COLUMNSELECT` $\ln\left(2/\delta\right)$ volte in modo indipendente, restituendo la matrice $\mathbf{C}$ che minimizza $\|\mathbf{A}-\mathbf{C}\mathbf{C}^{+}\mathbf{A}\|_F$ e di eseguire l'algoritmo `COLUMNSELECT` su $\mathbf{A}^T$ $\ln\left(2/\delta\right)$ volte in modo indipendente, restituendo la matrice $\mathbf{R}$ che minimizza $\|\mathbf{A}-\mathbf{C}\mathbf{U}\mathbf{R}\|_F$, con la precedente scelta di $\mathbf{C}$.
Allora la probabilità che non valga:

$$\|\mathbf{A}-\mathbf{C}\mathbf{U}\mathbf{R}\|_F\leq(1+\varepsilon)\left\|\mathbf{A}-\mathbf{C}\mathbf{C}^{+}\mathbf{A}\right\|_F$$

è minore di:

$$(1/e)^{\ln\left(2/\delta\right)}+(1/e)^{\ln\left(2/\delta\right)}=\delta/2+\delta/2=\delta.$$

□

Osservazione. Unendo i risultati dei due Corollari, 2.10 e 2.12, si ottiene che:

$$\|\mathbf{A}-\mathbf{C}\mathbf{U}\mathbf{R}\|_F\leq(1+\varepsilon)\left\|\mathbf{A}-\mathbf{C}\mathbf{C}^{+}\mathbf{A}\right\|_F\leq(1+\varepsilon)^2\|\mathbf{A}-\mathbf{A}_k\|_F\leq(1+\varepsilon')\|\mathbf{A}-\mathbf{A}_k\|_F$$

con $\varepsilon'=3\varepsilon$, cioè:

$$\|\mathbf{A}-\mathbf{C}\mathbf{U}\mathbf{R}\|_F\leq(1+\varepsilon')\|\mathbf{A}-\mathbf{A}_k\|_F \tag{2.13}$$

con probabilità almeno $1-\delta$.
Abbiamo quindi mostrato l'esistenza di un algoritmo randomizzato, `ALGORITHMCUR`, che consente di trovare un'approssimazione di rango k della matrice $\mathbf{A}$, con garanzie sull'errore in funzione della migliore approssimazione di rango k di $\mathbf{A}$. Tale risultato sull'errore è garantito dal fatto che righe e colonne della matrice sono state campionate tramite la tecnica del Subspace Sampling. Inoltre tale algoritmo ha un costo computazionale nell'ordine del costo computazionale necessario per calcolare i primi k vettori singolari di $\mathbf{A}$.

Capitolo 3

Applicazioni

In questo capitolo è mostrata un'applicazione della Decomposizione CUR e degli algoritmi precedentemente descritti ad un dataset reale, che mette in luce come sia possibile utilizzare questa decomposizione per analizzare e interpretare dati. Essa infatti è stata applicata a dataset relativi a diversi ambiti, come alla biologia, alla genetica o all'analisi di documenti e pagine web [5]. In particolare vedremo nel dettaglio un'applicazione all'ambito sociale.

3.1 Dati

Il dataset che abbiamo preso in analisi contiene i risultati delle votazioni dei giudici della Corte Suprema degli Stati Uniti, durante il periodo tra il 1994 e il 2003 [5]. La Corte Suprema degli Stati Uniti è la più alta corte della magistratura federale ed è composta da nove membri, ciascuno eletto con mandato a vita e direttamente nominato dal presidente degli Stati Uniti, con il consenso del Senato. In quanto "suprema", rappresenta il tribunale di ultima istanza di Stato ed è l'unico direttamente disciplinato dalla costituzione.

Sono stati considerati i dati relativi al periodo tra il 1994 e il 2003 poiché, in tale periodo, la composizione dei nove giudici della Corte è rimasta invariata, sotto la guida del Presidente della Corte William Rehnquist. In tal senso, si tratta di uno dei più lunghi periodi di "stabilità" di composizione della Corte e questo ci consente di avere a disposizione un dataset che risulta abbastanza significativo. Vogliamo considerare questi dati cercando di dedurre eventuali informazioni latenti, che considereremo solo dal punto di vista matematico, senza fare riferimento alle questioni politiche che essi rappresentano per mancanza di sufficiente conoscenza dell'argomento.

La matrice $\mathbf{A}$ dei dati è stata costruita in modo che ogni colonna rappresenti il voto di un giudice e ogni riga un singolo caso giudiziario. I dati utilizzati sono stati scaricati dal sito "The Supreme Court Dataset" `http://scdb.wustl.edu/data.php` [3] e sono stati selezionati i voti relativi al periodo di nostro interesse.

La Corte discute circa 80 casi ogni anno, tuttavia sono state eliminate dall'analisi le righe con dati non completi e/o riguardanti casi particolari, mantenendo quindi circa il 70% dei dati iniziali. Il risultato è una matrice 513×9 di valori in $\{-1, 1\}$, dove 1 e -1 indicano, rispettivamente, se il voto del giudice era in accordo o in disaccordo con la maggioranza. Posto il vettore dei giudici:

$$\mathbf{G} = [\text{Rehnquist, Stevens, O'Connor, Scalia, Kennedy, Souter, Thomas, Ginsburg, Breyer}],$$

ogni riga della matrice $\mathbf{A}$ è un vettore riga del tipo:

$$\mathbf{n} = [n_R, n_{St}, n_O, n_{Sc}, n_K, n_{So}, n_T, n_G, n_B],$$

dove ogni n_i può assumere il valore $+1$ o -1. Per esempio il vettore:

$$\mathbf{u} = [1, 1, 1, 1, 1, 1, 1, 1, 1]$$

rappresenta una decisione unanime, mentre il vettore:

$$\mathbf{p} = [1, -1, 1, 1, 1, -1, 1, -1, -1]$$

rappresenta una votazione in cui la maggioranza ha vinto per 5 a 4 e i giudici contrari sono stati Stevens, Souter, Ginsburg e Breyer.
I possibili esiti di una votazione della Corte sono dunque $2^9 = 512$, ma avendo considerato ogni voto solo come "accordo" o "disaccordo" con la maggioranza, le possibilità dimezzano, quindi sono 256. Tuttavia, come si può facilmente immaginare, questi 256 possibili esiti non ricorrono tutti con la stessa frequenza. Nella Tabella 3.1 sono stati riportati i 12 esiti più frequenti, ogni altro esito ricorre meno dell'1% delle volte. La decisione unanime $\mathbf{u}$ ricorre il 50% delle volte, mentre la particolare maggioranza per 5 a 4, $\mathbf{p}$, è il secondo caso più frequente. Interessante è anche il fatto che il terzo caso più ricorrente sia quello in cui l'unico giudice a votare contro è stato il Giudice Stevens [8].
Nella Tabella 3.2 è stata riportata la matrice di correlazione di $\mathbf{A}$, in cui si nota che la coppia di giudici con il coefficiente di correlazione più alto è la coppia Scalia-Thomas, ma anche i giudici Souter-Ginsburg-Breyer sono molto correlati tra loro. Tali correlazioni sono visibili anche osservando la Figura 3.1a, dove è riportata la proiezione dei nove giudici sullo spazio generato dai primi vettori singolari destri di $\mathbf{A}$. Il fatto che esistano tali correlazioni ci permette, grazie all'analisi delle componenti principali, di rappresentare tutte le variabili, cioè i nove giudici, in funzione di un numero minore di variabili latenti, che mantengono la maggior parte delle informazioni [9]. Quello che abbiamo mostrato con la decomposizione CUR è che in particolare, è possibile approssimare tutta la matrice utilizzando direttamente i voti di alcuni giudici, in un numero basso di casi specifici.

Re	St	OC	Sc	Ke	So	Th	Gi	Br	frequenza
1	1	1	1	1	1	1	1	1	50.0 %
1	-1	1	1	1	-1	1	-1	-1	8.6 %
1	-1	1	1	1	1	1	1	1	7.8 %
-1	1	1	-1	-1	1	-1	1	1	4.1 %
1	1	1	-1	1	1	-1	1	1	3.7 %
-1	1	1	-1	1	1	-1	1	1	2.7 %
1	-1	1	1	1	1	1	1	-1	2.0 %
1	1	-1	-1	1	1	-1	1	1	2.0 %
-1	1	-1	-1	1	1	-1	1	1	1.6 %
1	-1	1	1	1	-1	1	-1	1	1.2 %
1	1	1	1	1	1	-1	1	1	1.0 %
1	1	1	-1	-1	1	-1	1	1	1.0 %

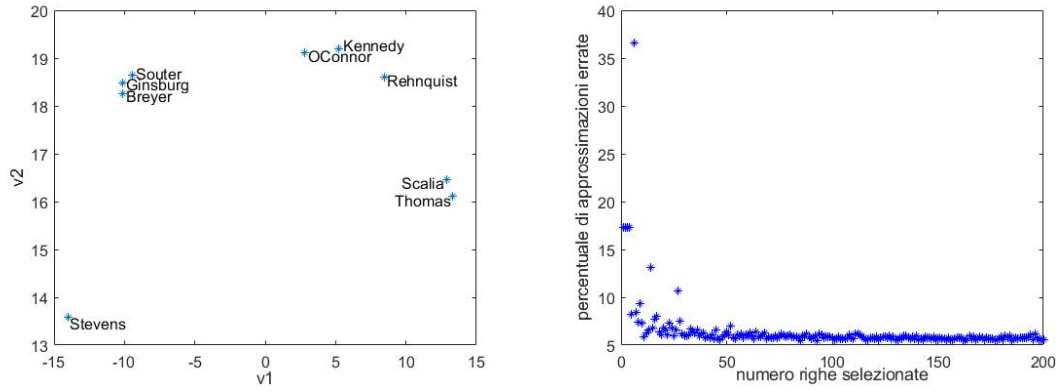
Tabella 3.1: I 12 risultati più ricorrenti delle votazioni della Corte, con la relativa frequenza.
Re: Rehnquist, St: Stevens, OC: O'Connor, Sc: Scalia, Ke: Kennedy, So: Souter, Th: Thomas, Gi: Ginsburg, Br: Breyer.

	Re	St	OC	Sc	Ke	So	Th	Gi	Br
Re	1.0000	−0.1976	0.2150	0.5286	0.5009	−0.1157	0.5358	−0.0449	−0.0727
St	−0.1976	1.0000	−0.0443	−0.2005	−0.0624	0.5393	−0.2048	0.5638	0.5578
OC	0.2150	−0.0443	1.0000	0.2805	0.1088	0.0294	0.2402	−0.0047	0.0815
Sc	0.5286	−0.2005	0.2805	1.0000	0.3678	−0.1016	**0.8324**	−0.1055	−0.1538
Ke	0.5009	−0.0624	0.1088	0.3678	1.0000	−0.0651	0.3791	−0.0189	−0.0719
So	−0.1157	0.5393	0.0294	−0.1016	−0.0651	1.0000	−0.1327	**0.7173**	**0.6070**
Th	0.5358	−0.2048	0.2402	0.8324	0.3791	−0.1327	1.0000	−0.1357	−0.1839
Gi	−0.0449	0.5638	−0.0047	−0.1055	−0.0189	0.7173	−0.1357	1.0000	**0.6699**
Br	−0.0727	0.5578	0.0815	−0.1538	−0.0719	0.6070	−0.1839	0.6699	1.0000

Tabella 3.2: La matrice di correlazione di $\mathbf{A}$.
Re: Rehnquist, St: Stevens, OC: O'Connor, Sc: Scalia, Ke: Kennedy, So: Souter, Th: Thomas, Gi: Ginsburg, Br: Breyer.

3.2 Decomposizione CUR

In questa sezione sono riportati i risultati di alcuni esperimenti in cui sono stati applicati gli algoritmi descritti nel capitolo precedente alla matrice dei dati $\mathbf{A}$.
Nella Tabella 3.3 sono riportati i risultati ottenuti applicando `ALGORITHMCUR` alla matrice dei dati, con in input il parametro ε di errore e il parametro k di rango. L'algoritmo è stato applicato più volte, per verificare con quale probabilità risulta verificato il risultato sull'errore: $\|\mathbf{A} - \mathbf{CUR}\|_F \leq (1+\varepsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$, mostrato nel capitolo precedente. Si osserva che esso è verificata con probabilità molto alta. Tuttavia, si nota che impostando in input parametri di ε bassi, l'algoritmo tende a scegliere un numero elevato di righe della matrice e tutte, o quasi, le sue colonne. Questo è dovuto al fatto che l'algoritmo sceglie in media $c = \mathcal{O}(k\log(k)/\varepsilon^2)$ colonne e $r = \mathcal{O}(c\log(c)/\varepsilon^2)$ righe per garantire il risultato sull'errore. Per questo



(a) (b)

Figura 3.1: Nella Figura (a) è mostrata la proiezione dei giudici sullo spazio generato dai primi due vettori singolari destri. L'asse x e l'asse y rappresentano rispettivamente il primo e il secondo vettore singolare destro.
Nella Figura (b) è riportata la percentuale di voti che non vengono correttamente approssimati da $\mathbf{A'} = \mathbf{CUR}$, dove la matrice $\mathbf{C}$ è stata ottenuta estraendo tre righe di $\mathbf{A}$, mentre la matrice $\mathbf{R}$ (e, di conseguenza, la matrice $\mathbf{U}$, definita come nel Capitolo 2, ovvero $\mathbf{U} = \mathbf{C}^{+}\mathbf{A}\mathbf{R}^{+}$) è costruita ad ogni iterazione estraendo un differente numero di righe.

motivo è stata implementata una variante dell'algoritmo che prende in input anche il numero di righe e di colonne che si vogliono estrarre (in media). In questo modo si verifica che, anche scegliendo un numero molto più basso di righe della matrice, la maggior parte dei risultati viene correttamente approssimata, come si nota dai risultati riportati nella Tabella 3.4. Infatti, già un numero di righe abbastanza basso è sufficiente per ottenere la migliore approssimazione della matrice e questo è confermato anche dalla Figura 3.1b, dove sono riportati i casi non correttamente approssimati da $\mathbf{A'} = \mathbf{CUR}$, al variare del numero di righe estratte per costituire $\mathbf{R}$. Per verificare quanti voti fossero stati correttamente approssimati è stato arrotondato ogni elemento di $\mathbf{A'} = \mathbf{CUR}$ al più vicino tra $+1$ e -1.
Infine, nella Tabella 3.5 sono sono riportati i voti correttamente approssimati dal prodotto $\mathbf{CUR}$, dove $\mathbf{C}$ è costruita scegliendo le tre colonne della matrice rappresentanti i giudici Scalia, Kennedy e Ginsburg, poiché dalla Figura 3.1a si deduce che i tre giudici sono molto dissimili tra loro, mentre le righe che costituiscono l matrice $\mathbf{R}$ sono estratte in modo casuale dall'algoritmo `COLUMNSELECT` con input $\mathbf{A}^T$ e le probabilità di campionamento delle righe di $\mathbf{A}$. Si nota che, con questa particolare scelta delle colonne, anche con un numero di righe molto basso, si riesce ad approssimare una percentuale molto alta dei voti del dataset iniziale.

Da questi esperimenti si deduce che, come ci possiamo aspettare, all'interno della Corte ci sono delle forti coalizioni che permettono di ridurre il numero di informazioni necessarie per approssimare la maggior parte del dataset. Dunque, è sufficiente conoscere i voti di pochi giudici in pochi casi, per poter prevedere con

alta probabilità quale sia stato il risultato del voto di ogni giudice, in ogni caso discusso.

k	ε	c'	r'	disug.verificata
2	0.8	2.37	2.78	99.80%
2	0.6	3.86	14.72	100.00%
2	0.4	8.09	116.77	100.00%
2	0.2	9.00	503.36	100.00%
3	0.8	5.11	13.33	94.90%
3	0.4	9.00	328.90	100.00%
4	0.5	9.00	254.22	100.00%

Tabella 3.3: Verifica del risultato del Teorema 2.11 sull'errore, applicando più volte l'algoritmo descritto, con input: k parametro di rango, ε parametro di errore.
Nella Tabella abbiamo riportato anche il numero di colonne selezionate in media, c', il numero di righe selezionate in media, r', e la percentuale di casi in cui è verificata la disuguaglianza $\|\mathbf{A} - \mathbf{CUR}\|_F \leq (1+\varepsilon')\|\mathbf{A} - \mathbf{A}_k\|_F$.

k	c	r	voti corrett. appross.
1	3	8	81.77%
1	4	12	85.54%
2	3	6	85.97%
2	4	10	89.45%
3	3	6	84.08%
3	4	12	90.28%

Tabella 3.4: Voti correttamente approssimati dalla variante dell'algoritmo che prende in input il numero c delle colonne che si vogliono estrarre in media e il numero r delle righe che si vogliono estrarre in media per la decomposizione, oltre al parametro di rango, k.

k	r	voti corrett. appross.
2	6	88.73%
2	9	91.27%
3	6	88.74%
3	9	90.81%

Tabella 3.5: Voti correttamente approssimati scegliendo le tre colonne rappresentanti i giudici Scalia, Kennedy e Ginsburg, il rango k e mediamente r righe estratte in modo casuale.

Conclusioni

Nel primo capitolo di questa tesi abbiamo definito la Decomposizione in Valori Singolari di una matrice e abbiamo mostrato come, se troncata ad un certo numero di termini k, essa fornisca la migliore approssimazione di rango k della matrice. Per questo motivo, è molto utilizzata nelle applicazioni ed in particolare nel campo dell'analisi delle componenti principali.
Tuttavia tale decomposizione presenta alcuni limiti in quanto non preserva eventuali buone proprietà della matrice dei dati di partenza, come la non negatività o la sparsità, ed inoltre i vettori singolari che si ottengono da questa decomposizione, in generale, non hanno alcun significato in termini dei dati iniziali e non consentono di fare considerazioni sul fenomeno da cui essi provengono.

Per fornire una soluzione a questi problemi, nel secondo capitolo abbiamo introdotto la Decomposizione CUR, una decomposizione di rango basso che approssima la matrice dei dati di partenza $\mathbf{A}$ come prodotto di tre matrici, $\mathbf{C}$, $\mathbf{U}$ e $\mathbf{R}$, dove $\mathbf{C}$ è composta da alcune colonne di $\mathbf{A}$ e $\mathbf{R}$ da alcune righe di $\mathbf{A}$. Si tratta quindi di una decomposizione che dipende fortemente da alcuni degli stessi dati iniziali. Abbiamo descritto inoltre l'algoritmo per la scelta delle righe e delle colonne da selezionare che fornisce garanzie sull'errore in termini della norma di Frobenius.

Nel terzo capitolo abbiamo visto come questa decomposizione ci consenta di approssimare un intero dataset in funzione di pochi dati e quindi di poter fare considerazioni su tutto il fenomeno, attraverso alcune, poche, informazioni significative. Nel caso che abbiamo preso in analisi, abbiamo visto come sia sufficiente conoscere pochi voti di alcuni giudici della Corte Suprema, per ricostruire quasi interamente il risultato delle votazioni di tutti i giudici in un numero molto più elevato di casi giudiziari.

Nonostante queste buone proprietà della Decomposizione CUR, essa è comunque strettamente legata alla SVD, poiché necessita del calcolo dei primi vettori singolari e, in particolare, questo calcolo è anche quello che maggiormente influenza il costo computazionale dell'algoritmo descritto. Vi sono inoltre alcune questioni ancora aperte, come l'esistenza o meno di un algoritmo deterministico efficiente che consenta di ottenere sempre la migliore scelta delle righe e delle colonne della matrice; come il problema dell'esistenza di una condizione che permetta di determinare a priori se le matrici ottenute, $\mathbf{C}$, $\mathbf{U}$ e $\mathbf{R}$ verificano il risultato sull'errore,

senza necessità di calcolare esplicitamente la norma di Frobenius di $\mathbf{A} - \mathbf{CUR}$ o se sia possibile estendere tale risultato sull'errore anche ad altre norme matriciali.

Appendice A

Elenco degli algoritmi

Di seguito sono riportati i codici degli algoritmi utilizzati nelle applicazioni. Siano $\mathbf{A}$ una matrice in $\mathbb{R}^{m\times n}$, p_j il vettore che contiene le probabilità di campionamento delle colonne di $\mathbf{A}$, c un intero tale che $c \leq n$, r un intero tale che $r \leq m$.

Algoritmo di sampling

La funzione `SAMPLING`, dati in input $\mathbf{A}$, p_j e c, restituisce le matrici $\mathbf{S} \in \mathbb{R}^{n\times n}$ e $\mathbf{D} \in \mathbb{R}^{n\times c}$ di sampling e scaling.

```
function [S,D]=SAMPLING(A,pj,c)
    [m,n]=size(A);
    S=zeros(n,1);
    D=[];
    t=1;
    indici=[];
    while isempty(indici) % per evitare che restituisca matrici vuote
        for i=1:n
            pi=pj(i);
            if rand<min(1,c*pi)
                indici=[indici,i];
                S(i,t)=1;
                D(t,t)=1/min(1,sqrt(c*pi));
                t=t+1;
            end
        end
    end
end
```

Algoritmo per la selezione delle colonne

La funzione `COLUMNSELECT`, dati in input $\mathbf{A}$, p_j e c, restituisce la matrice $\mathbf{C}$ con le colonne campionate e le matrici di sampling e scaling, $\mathbf{S}$ e $\mathbf{D}$.

```
function [C,S,D]=COLUMNSELECT(A,pj,c)
    [S,D]=SAMPLING(A,pj,c);
    C=A*S*D;
end
```

Algoritmi per la Decomposizione CUR

La funzione `ALGORITHMCUR`, dati in input $\mathbf{A}$, k il parametro di rango, ε il parametro di errore, restituisce le tre matrici della decomposizione, $\mathbf{C}$, $\mathbf{U}$, $\mathbf{R}$.
La funzione `ALGORITHMCUR_2` è una variante della precedente che prende in input $\mathbf{A}$, il numero c di colonne che si vogliono estrarre in media, il numero r di righe che si vogliono estrarre in media e restituisce le tre matrici della decomposizione.

```
function [C,U,R]=ALGORITHMCUR(A,k,epsilon)
    [U_k,~,V_k]=svds(A,k);
    pj=sum(V_k.*V_k,2)/k;
    pi=sum(U_k.*U_k,2)/k;
    c=k*log(k)/(epsilon^2);
    r=c*log(c)/(epsilon^2);
    [C,~,~]=COLUMNSELECT(A,pj,c);
    [R,S,D]=COLUMNSELECT(A',pi,r); R=R';
    W=D'*S'*C;
    U=pinv(W);
end
```

```
function [C,U,R]=ALGORITHMCUR_2(A,k,c,r)
    [U_k,~,V_k]=svds(A,k);
    pj=sum(V_k.*V_k,2)/k;
    pi=sum(U_k.*U_k,2)/k;
    [C,~,~]=COLUMNSELECT(A,pj,c);
    [R,S,D]=COLUMNSELECT(A',pi,r); R=R';
    W=D'*S'*C;
    U=pinv(W);
end
```

Bibliografia

[1] Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[2] Gene H. Golub and Charles F. Van Loan. Matrix computations. 1996.

[3] Washington University Law. The supreme court dataset. `http://scdb.wustl.edu/data.php`.

[4] Michael W Mahoney and Petros Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[5] Michael W Mahoney and Petros Drineas. Supplementary material: Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 2009.

[6] Davide Palitta and Valeria Simoncini. Dispense del corso di calcolo numerico. *Modulo di Algebra Lineare*, 2016.

[7] Valeria Simoncini. Lucidi del corso di matematica computazionale. 2021.

[8] Lawrence Sirovich. A pattern analysis of the second rehnquist us supreme court. *Proceedings of the National Academy of Sciences*, 100(13):7432–7437, 2003.

[9] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[10] Zhihua Zhang. The singular value decomposition, applications and beyond, 2015.