Tecniche di algebra lineare numerica per l'analisi dei dati

Michele Benzi, Scuola Normale Superiore

SCUOLA
NORMALE
SUPERIORE

Scientific Data Analysis School

Pisa, 25–28 novembre 2019

Contenuti

- La decomposizione ai valori singolari (SVD)
- Applicazioni della SVD: compressione e filtraggio di dati, Indicizzazione Semantica Latente (LSI)
- Analisi delle Componenti Principali (PCA, SPCA)
- Fattorizzazioni matriciali non-negative (NMF, SNMF)
- Decomposizioni tensoriali (CP, Tucker)

La decomposizione ai valori singolari

Data una matrice A (ad elementi reali o complessi, quadrata o meno), una decomposizione (o fattorizzazione) di A è una rappresentazione di A come prodotto di matrici di forma più "semplice" o comunque tale da rivelare proprietà "latenti" della matrice (rango, autovalori, etc.).

Esempi includono le fattorizzazioni LU, QR, la decomposizione spettrale, di Schur, ed altre.

La Singular Value Decomposition (SVD) è senza ombra di dubbio la decomposizione matriciale più fondamentale, tanto dal punto di vista strettamente matematico quanto da quello delle applicazioni, specialmente all'analisi di dati.

Data una decomposizione SVD di A, è infatti possibile analizzare completamente A e risolvere, almeno in linea di principio, qualsiasi problema associato ad A.

La decomposizione ai valori singolari (cont.)

Cominciamo con alcuni richiami di algebra lineare.

Ricordiamo che una matrice ortogonale di ordine n è una matrice reale $n \times n$ tale che $QQ^T = Q^T Q = I_n$.

In altre parole, Q è invertibile, e $Q^{-1} = Q^T$. Di conseguenza, si ha che $\det(Q) = \pm 1$.

Le matrici ortogonali sono isometrie: $\|Q\mathbf{x}\|_2 = \|\mathbf{x}\|_2$, dove

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

è la norma euclidea del vettore $\mathbf{x} = (x_i) \in \mathbb{R}^n$.

Rotazioni e riflessioni sono esempi di trasformazioni ortogonali (rappresentati da matrici ortogonali rispetto alla base canonica di $\mathbb{R}^n$). Le matrici ortogonali formano gruppo.

La decomposizione ai valori singolari (cont.)

Una matrice $A \in \mathbb{R}^{m\times n}$ rappresenta, rispetto ad una opportuna coppia di basi fissate in $\mathbb{R}^n$ e $\mathbb{R}^m$, una trasformazione lineare

$$T_A : \mathbb{R}^n \longrightarrow \mathbb{R}^m.$$

Associati ad ogni matrice $A \in \mathbb{R}^{m\times n}$ vi sono diversi sottospazi vettoriali fondamentali. Lo spazio

$$\mathcal{R}(A) = \{\mathbf{y} \in \mathbb{R}^m \,|\, \mathbf{y} = A\mathbf{x} \text{ per un } \mathbf{x} \in \mathbb{R}^n\}$$

si dice l'immagine (inglese: range), di A. La dimensione di questo sottospazio di $\mathbb{R}^m$ è il rango di A, indicato con $\text{rank}(A)$. Il rango di A è uguale al massimo numero di righe (o colonne) di A che sono linearmente indipendenti.

Si ha naturalmente che $\text{rank}(A) \leq \min\{m, n\}$.

La decomposizione ai valori singolari (cont.)

Lo spazio nullo, o nucleo, della matrice A è il sottospazio di $\mathbb{R}^n$ costituito da tutti i vettori $\mathbf{x}$ tali che $A\mathbf{x} = \mathbf{0}$:

$$\mathcal{N}(A) = \{\mathbf{x} \in \mathbb{R}^n \,|\, A\mathbf{x} = \mathbf{0}\}$$

(si usa anche la notazione $\ker(A)$). La dimensione di $\mathcal{N}(A)$ si chiama la nullità di A, indicata con $\text{null}(A)$.

Nel caso di matrici quadrate ($m = n$), A è invertibile se e solo se $\mathcal{N}(A) = \{\mathbf{0}\}$.

Si dice poi complemento ortogonale di un sottospazio $\mathcal{S}$ di $\mathbb{R}^n$ il sottospazio di $\mathbb{R}^n$ formato dai vettori che sono ortogonali a tutti i vettori di $\mathcal{S}$:

$$\mathcal{S}^\perp = \{\mathbf{y} \in \mathbb{R}^n \mid \langle \mathbf{x}, \mathbf{y} \rangle = 0 \ \forall \mathbf{x} \in \mathcal{S}\}.$$

La decomposizione ai valori singolari (cont.)

Si ha $\mathbb{R}^n = \mathcal{S} \oplus \mathcal{S}^\perp$ e pertanto $\dim \mathcal{S} + \dim \mathcal{S}^\perp = n$.

Sussiste la seguente importante relazione:

$$\mathcal{R}(A) = \mathcal{N}(A^T)^\perp,$$

da cui discende che per ogni matrice $A \in \mathbb{R}^{m \times n}$, si ha

$$\text{rank}(A) + \text{null}(A) = n.$$

Gil Strang ha dato a questo risultato il nome di Teorema fondamentale del'Algebra Lineare.

Il seguente risultato è uno dei principali teoremi dell'Algebra Lineare.

La decomposizione ai valori singolari (cont.)

Teorema: Sia $A \in \mathbb{R}^{m\times n}$. Esistono due matrici ortogonali $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{n\times n}$ ed una matrice $\Sigma \in \mathbb{R}^{m\times n}$ della forma

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & \dots & \sigma_r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{pmatrix}$$

dove $r = \text{rank}(A)$ e $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, tali che

$$A = U\Sigma V^T.$$

I numeri σ_i si dicono valori singolari di A. Osserviamo che mentre la matrice Σ è univocamente determinata da A, le matrici U e V non sono uniche.

La decomposizione ai valori singolari (cont.)

Il teorema ora enunciato dice che per ogni trasformazione lineare $T : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, è sempre possibile trovare una base ortonormale in $\mathbb{R}^n$ ed una base ortonormale in $\mathbb{R}^m$ tali che la matrice di T rispetto a questa coppia di basi è "diagonale", con elementi positivi o nulli sulla diagonale principale. Il rango di A coincide con il numero di elementi diagonali strettamente positivi.

La SVD può essere vista come una sorta di "diagonalizzazione" ($U^T AV = \Sigma$) per matrici arbitrarie, anche non quadrate; si noti tuttavia che per questa diagonalizzazione si richiedono due basi ortonormali anzichè una sola. Solo quando A è simmetrica e semidefinita positiva i due tipi di diagonalizzazione coincidono.

La decomposizione ai valori singolari fu introdotta da Eugenio Beltrami nel 1875 nel caso di matrici quadrate. Contributi successivi, tra i quali l'estensione al caso rettangolare, sono dovuti a J. J. Sylvester, C. Jordan, E. Schmidt, L. Autonne ed altri.

La decomposizione ai valori singolari (cont.)

Denotiamo con $\mathbf{u}_i$ e $\mathbf{v}_i$ le colonne di U e V:

$$U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m], \quad V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n].$$

I vettori $\mathbf{u}_i$ si dicono vettori singolari sinistri e i $\mathbf{v}_i$ vettori singolari destri di A, rispettivamente.

Osserviamo che si ha

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i, \quad A^T \mathbf{u}_i = \sigma_i \mathbf{v}_i, \ \ 1 \leq i \leq r$$

e

$$A\mathbf{v}_i = \mathbf{0}, \quad A^T \mathbf{u}_i = \mathbf{0}, \ \ i \geq r+1.$$

Inoltre $AA^T = U\Sigma\Sigma^T U^T$ e $A^T A = V\Sigma^T \Sigma V^T$.

La decomposizione ai valori singolari (cont.)

Quindi, i vettori singolari sinistri $\mathbf{u}_i$ sono autovettori della matrice AA^T $(m \times m)$, i vettori singolari destri $\mathbf{v}_i$ sono autovettori della matrice A^TA $(n \times n)$, e i quadrati σ_i^2 sono gli autovalori non nulli corrispondenti.

Si ha inoltre

$$A = U\Sigma V^T = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

dunque ogni matrice di rango r si scrive come somma di r matrici di rango 1.

Se definiamo $U_r \in \mathbb{R}^{m \times r}$, $V_r \in \mathbb{R}^{n \times r}$, and $\Sigma_r \in \mathbb{R}^{r \times r}$ come segue:

$$U_r = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r], \quad V_r = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r], \quad \text{e} \quad \Sigma_r = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$$

si ha allora che $A = U_r \Sigma_r V_r^T$ (la cosiddetta SVD compatta di A).

La decomposizione ai valori singolari (cont.)

Data una SVD di A, è possibile identificare i quattro sottospazi fondamentali associati con A. Più precisamente si ha il seguente importante risultato.

Teorema: Sia $A \in \mathbb{R}^{m \times n}$ e sia $A = U\Sigma V^T$ una sua SVD. Se $r = \text{rank}(A)$, si ha:

(i) $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ è base ortonormale di $\mathcal{R}(A)$.

(ii) $\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_m\}$ è base ortonormale di $\mathcal{N}(A^T)$ $(= \mathcal{R}(A)^\perp)$.

(iii) $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ è base ortonormale di $\mathcal{R}(A^T)$.

(iv) $\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n\}$ è base ortonormale di $\mathcal{N}(A)$ $(= \mathcal{R}(A^T)^\perp)$.

La decomposizione ai valori singolari (cont.)

Una delle principali applicazioni della SVD è alla soluzione di problemi di minimi quadrati e, più in generale, di sistemi lineari a matrice dei coefficienti rettangolare, compatibili o meno.

Se $A \in \mathbb{R}^{m \times n}$ ha la SVD di rango r

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

e $\mathbf{b} \in \mathbb{R}^m$ è un vettore dato, la soluzione di norma minima del problema

$$\|A\mathbf{x} - \mathbf{b}\|_2 = \min$$

si può esprimere in maniera compatta con $\mathbf{x} = A^+\mathbf{b}$, dove $A^+ \in \mathbb{R}^{n \times m}$ è la pseudo-inversa di Moore–Penrose, definita da

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

In altri termini,

$$\mathbf{x} = A^+\mathbf{b} = \sum_{i=1}^{r} \left(\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \right) \mathbf{v}_i.$$

La decomposizione ai valori singolari (cont.)

In analogia con la norma di un vettore, si possono definire varie norme matriciali. Una norma matriciale $\|A\|$ soddisfa le seguenti condizioni:

(i) $\|A\| \geq 0$, e $\|A\| = 0$ se e solo se $A = 0$ (matrice nulla).

(ii) $\|\alpha A\| = |\alpha| \|A\|$ per ogni $\alpha \in \mathbb{R}$ e ogni matrice A.

(iii) $\|A + B\| \leq \|A\| + \|B\|$ (disuguaglianza triangolare).

(iv) $\|AB\| \leq \|A\| \|B\|$ (quando il prodotto AB è definito).

Due esempi di norme matriciali importanti sono la norma spettrale e la norma di Frobenius di A:

$$\|A\|_2 = \sigma_1, \qquad \|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\sum_{i=1}^{r} \sigma_i^2}\,.$$

La decomposizione ai valori singolari (cont.)

In moltissimi problemi (di analisi e filtraggio di dati sperimentali, ricostruzione o compressione di segnali e immagini, etc.) è spesso importante sapere come approssimare una matrice A, in genere di dimensioni molto grandi e di rango arbitrario, con matrici di rango basso rispetto al rango di A.

LA SVD fornisce una soluzione completa ed elegante a questo problema. Il seguente Teorema di Eckart e Young (ma in realtà dovuto a E. Schmidt) dice infatti che l'approssimazione ottimale in norma $\| \ \|_2$ o in norma $\| \ \|_F$ di una matrice data A di rango r con una matrice A_k di rango prefissato k (con $1 \leq k \leq r-1$) si ottiene "troncando" una SVD di A.

La decomposizione ai valori singolari (cont.)

Teorema: Sia data la matrice $A \in \mathbb{R}^{m\times n}$ di rango k con SVD

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i \, \mathbf{u}_i \mathbf{v}_i^T .$$

Per $1 \leq k \leq r$, definiamo

$$A_k = \sum_{i=1}^{k} \sigma_i \, \mathbf{u}_i \mathbf{v}_i^T .$$

Se $\| \ \|$ denota una delle due norme $\| \ \|_2$ o $\| \ \|_F$, si ha

$$\|A - B\|_2 \geq \|A - A_k\|_2, \quad \forall B \in \mathbb{R}^{m\times n} \text{ t.c. } \operatorname{rank}(B) = k.$$

Si ha inoltre che

$$\|A - A_k\|_2 = \sigma_{k+1}, \qquad \|A - A_k\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}.$$

Applicazioni alla compressione e filtraggio di dati

La possibilità di approssimare, in modo ottimale, una matrice arbitraria con una di rango "basso" permette la compressione di grandi insiemi di dati, tipicamente affetti da errore (o "rumore").

Sia data, per esempio, una collezione di dati numerici organizzati in forma di tabella $m \times n$. Supponiamo inoltre che i corrispondenti valori singolari siano $\sigma_1 \approx 1000$, $\sigma_2 \approx 500$, $\sigma_3 \approx 100$, $\sigma_4 \approx 10$, $\sigma_5 \approx 1$, ..., $\sigma_k \leq 0.1$ per $k > 30$.

Ponendo $A_{30} = \sigma_1 \, \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \sigma_{30} \, \mathbf{u}_{30} \mathbf{v}_{30}^T$ si ha pertanto

$$\frac{\|A - A_{30}\|_2}{\|A\|_2} = \frac{\sigma_{31}}{\sigma_1} \approx \frac{0.1}{1000} = 10^{-4},$$

che è un errore relativo modesto e comunque accettabile in quelle applicazioni in cui i dati siano conosciuti solo a meno di un errore relativo comparabile.

Applicazioni alla compressione e filtraggio di dati (cont.)

Per esempio, se A rappresenta un'immagine in bianco e nero su uno schermo da $256 \times 256 = 65.536$ pixel (qui a_{ij} rappresenta il "livello di grigio" nel corrispondente pixel), un'approssimazione di rango 30 significa avere compresso i 65.536 numeri dell'immagine iniziale in $30 + 30 * 256 + 30 * 256 = 15.390$ numeri, con un risparmio di memoria vicino all'80% a fronte di un errore di approssimazione più che accettabile.

È da sottolineare che risparmi di questa entità sono assolutamente tipici, in quanto le grandi masse di dati sono spesso caratterizzate da grande ridondanza; il "grosso" delll'informazione estraibile dai grandi insiemi di dati spesso giace in un sottospazio (o varietà) di dimensione bassa rispetto a quella dello spazio dei dati, tutto il resto è "rumore", o dettagli inessenziali.

Le conseguenze di questo fatto per la memorizzazione, elaborazione e soprattutto la trasmissione dell'informazione (testi, immagini, audio, video,...) sono facilmente immaginabili.

Applicazioni alla compressione e filtraggio di dati (cont.)

Qualunque sistema di raccolta di dati è necessariamente affetto da errore. La rappresentazione di numeri reali mediante numeri di macchina introduce inoltre inevitabili errori di arrotondamento. Dunque ogni matrice $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ i cui elementi sono dati sperimentali sarà affetta da errore o, come si dice, da rumore.

Un problema fondamentale nel trattamento delle informazioni è l'estrazione del segnale contenuto in stream di dati affetti da rumore (problema del denoising). Per esempio, qualunque file contenente messaggi audio, immagini o video è affetto da rumore.

Anche in questo caso la SVD troncata fornisce una soluzione efficace al problema.

Applicazioni alla compressione e filtraggio di dati (cont.)

Supponiamo che sia data la SVD

$$A = U_r \Sigma_r V_r^T = \sum_{i=1}^{r} \sigma_i \, \mathbf{u}_i \mathbf{v}_i^T = \sum_{i=1}^{r} \sigma_i \, Z_i,$$

where r è il rango di A e $Z_i = \mathbf{u}_i \mathbf{v}_i^T$.

Osserviamo che le matrici Z_i sono mutuamente ortogonali rispetto al prodotto scalare tra matrici definito da

$$\langle Z_i, Z_j \rangle = \mathrm{Tr}\,(Z_i^T Z_j).$$

Infatti usando l'ortogonalità dei vettori singolari, si verifica facilmente che $\langle Z_i, Z_j \rangle = \delta_{ij}$. Pertanto, la SVD fornisce uno "sviluppo ortogonale" di una matrice, analogo allo sviluppo di Fourier di una funzione.

Applicazioni alla compressione e filtraggio di dati (cont.)

Si ha dunque che $\sigma_i = \langle Z_i, A \rangle$ si lascia interpretare come la parte di A che giace nella "direzione" di Z_i.

In molti casi concreti il rumore contamina i dati in modo casuale e senza avere una "direzione preferenziale", di modo che l'errore si distribuisce in modo più o meno uniforme tra le varie direzioni possibili, Z_i. Di conseguenza, ci si può aspettare che ciascuna componente $\sigma_i Z_i$ di A sia contaminata grosso modo dallo stesso livello di rumore.

Dal momento che $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$, ci si aspetta quindi che il rapporto segnale-rumore, indicato con SNR, soddisfi

$$\mathrm{SNR}(\sigma_1 Z_1) \geq \mathrm{SNR}(\sigma_2 Z_2) \geq \cdots \geq \mathrm{SNR}(\sigma_r Z_r).$$

Applicazioni alla compressione e filtraggio di dati (cont.)

Ne segue che in presenza di valori singolari "piccoli" (rispetto al rumore) $\sigma_{k+1}, \ldots, \sigma_r$, le componenti corrispondenti $\sigma_i Z_i$ di A, con $k+1 \leq i \leq r$, hanno SNR basso e sono dunque "dominate" dal rumore.

Quindi l'uso di una SVD troncata $A_k = \sum_{i=1}^{k} \sigma_i Z_i$ ci consente di rimuovere dai dati le componenti che consistono più che altro di rumore, migliorando così in modo considerevole il rapporto segnale-rumore globale.

Applicazioni alla compressione e filtraggio di dati (cont.)

Naturalmente la scelta di k è la parte più delicata, specialmente nel caso di matrici A di grandi dimensioni. Fortunatamente disponiamo di tecniche iterative (come il classico algoritmo di Lanczos) che permettono di calcolare i valori singolari σ_i e i vettori singolari corrispondenti $\mathbf{u}_i$, $\mathbf{v}_i$ in ordine decrescente.

Se disponiamo di una stima della percentuale di rumore presente nei dati (spesso tra l'1 e il 10 per cento) possiamo decidere a che punto fermare il calcolo e troncare lo sviluppo SVD di A.

Riassumendo: la SVD, calcolata per esempio mediante il metodo di Lanczos, ci permette di ricostruire il segnale e filtrare il rumore.

Se è data una stima del rumore, disponiamo anche di un criterio di troncamento ragionevole.

L'Indicizzazione Semantica Latente

L'Indicizzazione Semantica Latente (inglese: Latent Semantic Indexing, o LSI) è una tecnica ampiamente usata nel settore dell'information retrieval, per es. nella progettazione dei motori di ricerca e in modo crescente in biologia e nelle scienze biomediche.

Dato un corpus (spesso di grandissime dimensioni) di documenti, per esempio tutti gli articoli pubblicati online da un grande quotidiano negli ultimi 10 anni, e data una parola o frase chiave (query) immessa da un utente, si vogliono identificare in "tempo reale" tutti i documenti contenenti quella parola o frase chiave; inoltre i documenti ritrovati nei quali la parola o frase chiave compare con maggiore frequenza dovranno avere priorità rispetto ai documenti dove la parola o frase chiave appare con minor frequenza.

L'Indicizzazione Semantica Latente (cont.)

Il termine "indexing" sottolinea l'analogia con l'indice analitico di un libro, che identifica le pagine in cui appare un dato nome o argomento. In questo caso il corpus è il libro stesso, e le singole pagine corrispondono ai documenti.

È importante sottolineare che l'indicizzazione viene effettuata "online", cioè nel momento della ricerca (query time), quindi si richiedono tempi rapidissimi di risposta.

L'Indicizzazione Semantica Latente (cont.)

Supponiamo di avere un corpus, o data base, di documenti $D_1, \ldots, D_n$ e un dizionario di termini, $T_1, \ldots, T_m$; qui un termine può essere una parola, o un'espressione come "algebra lineare" (per es.). La dimensione tipica di un dizionario può essere di qualche centinaia di migliaia di termini. Il numero n di documenti può essere molto grande; per esempio una biblioteca digitalizzata può contenere decine di milioni di documenti.

Inizialmente, ogni documento viene *indicizzato* mediante un vettore-documento $\mathbf{d}_j$ della forma

$$\mathbf{d}_j = [\, f_{1j},\, f_{2j}, \ldots, f_{mj} \,]^T, \quad 1 \leq j \leq n$$

dove f_{ij} = numero di volte che il termine T_i compare nel documento D_j.

L'Indicizzazione Semantica Latente (cont.)

La matrice $m \times n$

$$A = [f_{ij}] = [\,\mathbf{d}_1,\, \mathbf{d}_2, \ldots ,\mathbf{d}_n\,]$$

si dice matrice termine-documento. Dal momento che la grande maggioranza degli elementi di A è zero, la matrice A è sparsa, il che ne rende possibile la memorizzazione su computer nonostante le grandi dimensioni.

Quando un utente sottopone al motore di ricerca una richiesta contenente uno o più termini T_j, viene formato il "vettore di richiesta" (query vector) $\mathbf{q} \in \mathbb{R}^m$ dato da

$$\mathbf{q} = [\,q_1,\, q_2, \ldots, q_m\,]^T,$$

dove $q_i = 1$ o $q_i = 0$ a seconda che il termine T_i compaia o meno nella "query". Anche in questo caso naturalmente il vettore $\mathbf{q}$ è estremamente sparso.

L'Indicizzazione Semantica Latente (cont.)

Per valutare quanto un documento D_j è "ben allineato" col vettore di richiesta $\mathbf{q}$, viene utilizzato il coseno dell'angolo tra i due vettori $\mathbf{q}$ e $\mathbf{d}_j$ di $\mathbb{R}^m$:

$$\cos\theta_j = \frac{\mathbf{q}^T\mathbf{d}_j}{\|\mathbf{q}\|_2\|\mathbf{d}_j\|_2} = \frac{\mathbf{q}^T A\,\mathbf{e}_j}{\|\mathbf{q}\|_2\|A\,\mathbf{e_j}\|_2},$$

dove come sempre denotiamo con $\mathbf{e}_j$ la j-ma colonna della matrice identità I_n, da cui $\mathbf{d}_j = A\,\mathbf{e}_j$.

Se $|\cos\theta_j| \geq \tau$, dove $\tau \in (0,1)$ è un valore soglia fissato, il documento D_j viene giudicato di interesse e restituito all'utente.

La scelta di τ è lasciata a chi progetta il motore di ricerca, ed è parzialmente soggettiva.

L'Indicizzazione Semantica Latente (cont.)

In pratica sia il vettore $\mathbf{q}$ che le colonne $\mathbf{d}_j$ di A sono normalizzate (di modo che $\|\mathbf{q}\|_2 = \|\mathbf{d}_j\|_2 = 1$). Ne segue che $|\cos\theta_j| = |\mathbf{q}^T\mathbf{d}_j|$ e i calcoli risultano semplificati. Dal momento che sia $\mathbf{q}$ che i $\mathbf{d}_j$ sono estremamente sparsi, il calcolo dei $\cos\theta_j$ anche per n molto grande, è molto rapido, dal momento che solo gli elementi diversi da zero "contano".

Notiamo inoltre che per la stragrande maggioranza dei j si ha $\mathbf{q}^T\mathbf{d}_j = 0$ e quindi $\cos\theta_j = 0$.

In molti casi, tuttavia, i risultati sono poco soddisfacenti e affetti da errori anche grossolani, con molti documenti irrilevanti restituiti all'utente come rilevanti. Le ragioni sono molteplici: l'ambiguità del linguaggio naturale, refusi, errori nella lettura di testi a stampa (per es. documenti o libri antichi), e così via. I dati sono affetti da rumore.

L'Indicizzazione Semantica Latente (cont.)

È quindi opportuno utilizzare tecniche di filtraggio del rumore quali la SVD troncata. Se denotiamo con

$$A_k = \sigma_1 \mathbf{u}_1 \mathbf{v}_i^T + \cdots + \sigma_k \mathbf{u}_k \mathbf{v}_k^T \approx A$$

una SVD troncata di A, con $k < r$ (= rango di A), possiamo utilizzare in luogo dei coseni degli angoli θ_j le quantità

$$\cos \phi_j = \frac{\mathbf{q}^T A_k \, \mathbf{e}_j}{\|\mathbf{q}\|_2 \|A_k \, \mathbf{e_j}\|_2}.$$

Se $A_k = U_k \Sigma_k V_k^T$ è la SVD compatta di A_k, osservato che

$$\|A_k \mathbf{e}_j\|_2 = \|U_k \Sigma_k V_k^T \mathbf{e}_j\|_2 = \|U_k \mathbf{s}_j\|_2 = \|\mathbf{s}_j\|_2,$$

dove $\Sigma_k V_k^T = [\mathbf{s}_1, \ldots, \mathbf{s}_k]$, si ha che

$$\cos \phi_j = \frac{\mathbf{q}^T U_k \, \mathbf{s}_j}{\|\mathbf{q}\|_2 \|\mathbf{s}_j\|_2}.$$

L'Indicizzazione Semantica Latente (cont.)

L'espressione

$$\cos\phi_j = \frac{\mathbf{q}^T U_k \mathbf{s}_j}{\|\mathbf{q}\|_2 \|\mathbf{s}_j\|_2}$$

presenta dei chiari vantaggi computazionali. La matrice U_k e i vettori $\mathbf{s}_j$ possono essere calcolati offline, dal momento che non dipendono dalla particolare query, e inoltre possono essere ottenuti senza calcolare l'intera SVD di A (algoritmo di Lanczos). Possiamo inoltre assumere che sia $\|\mathbf{s}_j\|_2 = 1$, previa normalizzazione.

Dal momento che nelle applicazioni alla LSI si possono usare approssimazioni di rango piuttosto basso ($k \ll r$), il calcolo delle quantità $\cos\phi_j$, anche in vista dell'estrema sparsità del vettore $\mathbf{q}$, richiede un numero molto modesto di operazioni e può essere pertanto effettuata in tempo reale, dell'ordine della frazione di secondo.

Altre applicazioni della SVD

In molte applicazioni è richiesta la soluzione di sistemi di equazioni lineari in cui i dati sono affetti da rumore, per esempio

$$A\mathbf{x} = \mathbf{b}' = \mathbf{b} + \mathbf{n},$$

dove $\mathbf{n}$ rappresenta l'errore, o rumore, presente nel membro destro e $\|\mathbf{n}\|_2 \ll \|\mathbf{b}\|_2$. Inoltre la matrice A dei coefficienti è spesso mal condizionata, cioè vicina ad una matrice singolare. Questo significa che A^{-1} contiene elementi molto grandi rispetto a quelli di A.

Di conseguenza la soluzione del sistema perturbato soddisfa

$$\mathbf{x}_p = A^{-1}\mathbf{b}' = A^{-1}\mathbf{b} + A^{-1}\mathbf{n} = \mathbf{x}_e + A^{-1}\mathbf{n},$$

dove $\mathbf{x}_e$ è la soluzione del sistema "imperturbato", mentre $A^{-1}\mathbf{n}$ rappresenta il rumore presente nella soluzione dovuto alla presenza di rumore nei dati.

Altre applicazioni della SVD (cont.)

Dalla formula

$$\mathbf{x}_p = \mathbf{x}_e + A^{-1}\mathbf{n}$$

si evince che anche quando $\|\mathbf{n}\|_2$ è molto piccolo, la presenza di elementi molto grandi in A^{-1} ha l'effetto di amplificare il rumore presente nei dati, "distruggendo" la soluzione del sistema.

La SVD in questi casi può essere usata per calcolare una soluzione approssimata soddisfacente (si parla di regolarizzazione del sistema lineare). Partiamo dalla SVD di A:

$$A = U\Sigma V^T$$

dove U e V ora sono matrici ortogonali $n \times n$ e Σ è una matrice diagonale invertibile.

Altre applicazioni della SVD (cont.)

Si ha

$$A^{-1} = (U\Sigma V^T)^{-1} = V\Sigma^{-1}U^T$$

e dunque $\|A^{-1}\|_2 = \|\Sigma^{-1}\|_2$. Pertanto la presenza di elementi molto grandi in A^{-1} è causata dalla presenza di valori singolari molto piccoli.

Matrici con valori singolari che decadono rapidamente a zero sono comuni in molte applicazioni, per esempio nella ricostruzione di immagini ottenute da telescopio o da microscopio.

Tali problemi sono noti come problemi mal posti, nel senso che la soluzione, se esiste, può essere non unica o estremamente sensibile a piccole perturbazioni nei dati del problema.

Altre applicazioni della SVD (cont.)

Scriviamo la soluzione del sistema perturbato nella forma

$$\mathbf{x}_p = A^{-1}(\mathbf{b}+\mathbf{n}) = (V\Sigma^{-1}U^T)(\mathbf{b}+\mathbf{n}) = \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i + \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{n}}{\sigma_i}\mathbf{v}_i.$$

Il primo termine fornisce la soluzione del sistema imperturbato, $\mathbf{x}_e$, il secondo rappresenta il rumore amplificato che contamina la soluzione. Come prima, è legittimo supporre che le componenti del rumore, $\mathbf{u}_i^T\mathbf{n}$, siano più o meno uguali per tutti gli $i = 1, \ldots, n$.

La presenza di $\sigma_i \approx 0$ mostra come la soluzione sia dominata dal rumore. Il problema è quindi dovuto alla presenza di valori singolari molto piccoli.

Altre applicazioni della SVD (cont.)

Supponiamo ora di suddividere i valori singolari in due gruppi, il primo costituito dai primi k valori singolari σ_i, non troppo piccoli, il secondo costituito dai valori singolari σ_i (con $k+1 \leq i \leq n$) considerati piccoli.

Definiamo la soluzione regolarizzata come segue:

$$\mathbf{x}^* = V_k \Sigma_k U_k^T \mathbf{b}' = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T(\mathbf{b}+\mathbf{n})}{\sigma_i} \mathbf{v}_i = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i + \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{n}}{\sigma_i} \mathbf{v}_i.$$

Consideriamo ora separatamente il contributo di ciascuna delle due sommatorie, cominciando dalla seconda. Osservando che

$$|\mathbf{u}_i^T \mathbf{n}| \leq \|\mathbf{u}_i\|_2 \|\mathbf{n}\|_2 = \|\mathbf{n}\|_2$$

(piccolo per ipotesi) e che i valori $\sigma_1, \ldots, \sigma_k$ non sono piccoli, si vede che il secondo termine, che rappresenta la contaminazione presente nella soluzione regolarizzata a causa del rumore, rimane a livelli accettabili.

Altre applicazioni della SVD (cont.)

Il primo termine,

$$\mathbf{x}_{e,k} = \sum_{i=1}^{k} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i,$$

è più delicato. Osserviamo che tale somma è il troncamento dello sviluppo (in termini di SVD) della soluzione imperturbata $\mathbf{x}_e$ al termine k-esimo. È quindi importante essere in grado di stimare l'errore che si commette troncando la soluzione.

A differenza del rumore, che è spesso distribuito uniformemente in tutte le direzioni, il membro destro $\mathbf{b}$ ha molto spesso la proprietà seguente (condizione di Picard): le componenti $\mathbf{u}_i^T \mathbf{b}$ di $\mathbf{b}$ rispetto alla base ortonormale $\{\mathbf{u}_1, \ldots \mathbf{u}_n\}$ decadono a zero molto più rapidamente dei valori singolari σ_i.

Altre applicazioni della SVD (cont.)

Troncare la soluzione imperturbata al termine k significa ignorare la parte della soluzione data da

$$\mathbf{x}_e - \mathbf{x}_{e,k} = \sum_{i=k+1}^{n} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i.$$

Se la condizione di Picard è soddisfatta, i termini "critici"

$$\frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i}, \quad i = k+1, \ldots, n$$

si mantengono piccoli, dal momento che per i sufficientemente grande il numeratore è piccolo rispetto al denominatore.

Tutto quanto detto si estende facilmente a sistemi rettangolari, rimpiazzando l'inversa A^{-1} con la pseudo-inversa A^{+}.

Analisi delle Componenti Principali (PCA)

In statistica, la SVD (troncata) va spesso sotto il nome di Principal Component Analysis.

Se $A_0 \in \mathbb{R}^{m \times n}$ è la matrice dei dati, si denota con A la matrice $m \times n$ ottenuta sottraendo da ciascun elemento a_{ij} la media degli elementi sulla riga i-ma. Le n colonne di A sono punti di $\mathbb{R}^m$ la cui somma (o media) è $\mathbf{0}$, il vettore nullo.

Spesso questi n punti tendono a cadere (approssimativamente) in un sottospazio lineare di $\mathbb{R}^m$ di dimensione $k \ll m$.

Questo sottospazio è precisamente il sottospazio generato dai primi k vettori singolari sinistri $\mathbf{u}_1, \ldots, \mathbf{u}_k$ di A, le componenti principali di A.

Ricordiamo che i vettori $\mathbf{u}_i$ costituiscono una base ortonormale per $\mathcal{R}(A) = \mathrm{Col}\,(A)$.

Analisi delle Componenti Principali (cont.)

Il Teorema di Eckart–Young garantisce che questa è la migliore approssimazione k-dimensionale possibile dei dati.

L'identificazione di questo sottospazio ottimale va anche sotto il nome di regressione ortogonale, da non confondersi con il problema classico dei minimi quadrati.

Le componenti principali di A sono dunque gli autovettori di AA^T associati ai k autovalori più grandi.

La matrice $S = \frac{1}{n-1} AA^T$ è nota come matrice di covarianza relativa al campione in oggetto. Le varianze sono date dagli elementi diagonali, le covarianze dagli elementi extra-diagonali. La traccia di S è la varianza totale. La PCA fornisce la seguente approssimazione della varianza totale:

$$T = (\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_k^2)/(n-1).$$

Analisi delle Componenti Principali (cont.)

Benchè sia possibile effettuare la PCA lavorando con la matrice AA^T, di ordine n, dal punto di vista computazionale è preferibile lavorare direttamente sulla matrice A, calcolando pertanto una SVD (troncata) di A invece di una diagonalizzazione parziale di AA^T.

Il calcolo della SVD di matrici di grandi dimensioni è un problema non banale ma per il quale esistono algoritmi efficienti ed accurati dovuti a Golub, Kahan, Reinsch, ed altri.

Per questi aspetti numerici si rimanda per es. al classico trattato di Golub e Van Loan:

G. H. Golub and C. F. Van Loan, *Matrix Computations. Fourth Edition*, Johns Hopkins University Press, Baltimore, MD, 2013.

La PCA sparsa

I vettori singolari di A sono vettori pieni: tutte le componenti sono non nulle, anche se spesso quasi tutte molto piccole (a causa della normalizzazione $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$). In molte applicazioni tuttavia è preferibile una fattorizzazione sparsa.

Nella Sparse Principal Component Analysis (SPCA) (Zou et al., 2006), la sparsità dei fattori viene imposta aggiungendo un termine di penalizzazione a una certa funzione obiettivo. In questo modo le variabili più importanti sono selezionate in modo automatico.

Il numero di componenti non-nulle di un vettore $\mathbf{x}$, indicato con $\|\mathbf{x}\|_0$, non è una vera norma e non è utilizzabile negli algoritmi di ottimizzazione numerica; tuttavia, l'uso di $\|\mathbf{x}\|_1 = \sum_i |x_i|$ ha un effetto simile ("ℓ^1-relaxation").

H. Zou, T. Hastie, R. Tibshirani, *Sparse principal component analysis*, J. Comput. Graph. Stat., 15 (2006), pp. 265-286

La PCA sparsa (cont.)

Per esempio, la tecnica LASSO (Least Absolute Shrinkage and Selection Operator) consiste nel risolvere il problema dei minimi quadrati modificato

$$\text{Minimize } \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 \,.$$

Questo è un problema di ottimizzazione nonlineare, che può essere risolto in modo efficiente (algoritmo ADMM, di Bregman, etc.).

Una variante ("elastic net") è il problema ℓ^1+ ridge regression:

$$\text{Minimize } \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 + \beta\|\mathbf{x}\|_2 \,,$$

anche questo risolubile con l'algoritmo ADMM.

La PCA sparsa (cont.)

Se la quantità cercata è invece una matrice, la sparsità può essere imposta usando come termine di penalizzazione la norma nucleare di A:

$$\|A\|_N = \sigma_1 + \sigma_2 + \cdots + \sigma_r.$$

Questa norma gioca un ruolo importante nella soluzione del problema dei dati mancanti, o completamento della matrice:

supponiamo che nella matrice dei dati A manchino alcune entrate, per es. a causa della impossibilità di effettuare certe misurazioni. Qual'è il modo ottimale di "riempire i buchi"?

Una strategia ragionevole è quella di mantenere il rango di A il più basso possibile. Sfortunatamente, il rango di A (cioè il numero di componenti non nulle nel vettore $\mathbf{s}$ dei valori singolari di A) non è una norma. La norma nucleare, che è la norma ℓ^1 del vettore $\mathbf{s}$, è un buon sostituto (es.: la Netflix competition).

Fattorizzazioni matriciali non-negative (NMF, SNMF)

Spesso i dati a_{ij} sono per natura non-negativi: $a_{ij} \geq 0$. Scriviamo $A \geq 0$. Per il Teorema di Perron–Frobenius, i vettori singolari $\mathbf{u}_1$ e $\mathbf{v}_1$ hanno componenti non-negative; a causa delle condizioni di ortogonalità, gli altri vettori singolari (e quindi le matrici U e V) avranno necessariamente elementi sia positivi che negativi. Di conseguenza il significato degli elementi di U e V non è di facile interpretazione.

Scopo della Fattorizzazione Matriciale Non-negativa (NMF) è di trovare due matrici $U \geq 0$ e $V \geq 0$, tipicamente di rango basso, tali che

$$A \approx UV.$$

In generale non è possibile trovare una soluzione esatta ($A = UV$), ed è quindi necessario accontentarsi di un'approssimazione; e anche se il problema fosse risolubile, la soluzione può non essere unica. Quando $A = A^T$ poniamo $U = V^T$, ma nella maggior parte delle applicazioni A non è nemmeno quadrata.

Fattorizzazioni matriciali non-negative (cont.)

Sia data $A \in \mathbb{R}^{m\times n}$, $A \geq 0$. Il problema della NMF ha la seguente forma:

$$\min_{U,V} \|A - UV\|_F^2$$

dove $U \in \mathbb{R}^{m\times k}$, $U \geq 0$ e $V \in \mathbb{R}^{k\times n}$, $V \geq 0$ con k fissato, $1 \leq k \ll \min\{m, n\}$.

Questo è un problema di ottimizzazione vincolata nonlineare, non convesso: la funzione che si vuole minimizzare tipicamente avrà diversi minimi locali. Inoltre il problema è NP-hard.

Un'euristica molto usata e semplice da implementare è il metodo delle iterazioni alternanti: data un'approssimazione iniziale $V_0 \approx V$ si trova la matrice $U_1 \in \mathbb{R}^{m\times k}$ nonnegativa tale che

$$\|A - U_1 V_0\|_F^2 = \min .$$

Questo problema è facilmente ed univocamente risolubile.

Fattorizzazioni matriciali non-negative (cont.)

Calcolata U_1, si determina la matrice $V_1 \in \mathbb{R}^{k \times n}$, $V_1 \geq 0$ tale che

$$\|A - U_1 V_1\|_F^2 = \min,$$

e così via, fino a che $\|A - U_i V_i\|_F$ non è sufficientemente piccola.

In generale non c'è alcuna garanzia che l'algoritmo converga al minimo (U^*, V^*) globale, e inoltre le soluzioni approssimate così calcolate potranno dipendere dall'approssimazione iniziale, V_0.

La letteratura sulla NMF è vasta, e la ricerca di algoritmi robusti ed efficienti per la sua implementazione è tutt'ora molto attiva; il metodo ADMM è attualmente uno dei più utilizzati.

Fattorizzazioni matriciali non-negative (cont.)

Nella fattorizzazione $A \approx UV$, ognuna delle n colonne di A è (appross.) una combinazione lineare delle k colonne di U, dove $k \ll n$; i coefficienti sono contenuti nella matrice V.

La NMF è dunque una tecnica di riduzione della dimensione lineare. L'uso della norma di Frobenius può essere rigorosamente giustificato sotto l'ipotesi di rumore Gaussiano.

La NMF è particolarmante adatta per i problemi di estrazione delle caratteristiche (feature extraction) e di text (or document) mining:

$$\text{Documento } \mathbf{a}_j \approx \sum_{i=1}^{k} (\text{Peso } v_{ij}) \times (\text{Argomento } \mathbf{u}_j).$$

Come nel caso della PCA, è inoltre possibile imporre vincoli di sparsità, eventualmente in forma rilassata, nella NMF ($\rightarrow$ SNMF).

Decomposizioni tensoriali

I dati possono essere rappresentati con vettori e matrici, ma in certi casi sono necessari più di due indici; per es. un' immagine a colori è rappresentata da un array a tre indici (modello RGB); un video a colori richiede quattro indici.

In generale, chiameremo tensore una struttura di dati a più indici. Vettori e matrici sono esempi di tensori (a uno e due indici, risp.), ma le cose si fanno notevolmente più complicate quando il numero di indici è maggiore di due, in particolare per quanto riguarda le nozioni fondamentali di rango e decomposizione.

Limitandoci al caso di tensori a tre indici, si può facilmente definire un tensore di rango 1. Dati tre vettori $\mathbf{a} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^p$, ha rango 1 il tensore $T \in \mathbb{R}^{m \times n \times p}$ seguente:

$$T = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}, \quad T_{ijk} = a_i b_j c_k \, .$$

Decomposizioni tensoriali (cont.)

Sommando k tensori di rango 1 si ottiene un tensore di rango k.

Tuttavia, il problema di determinare il rango di un tensore dato, o anche solo stimarlo, è estremamente difficile. Per capirne la ragione, consideriamo il seguente tensore di rango 3:

$$T = \mathbf{u} \circ \mathbf{u} \circ \mathbf{v} + \mathbf{u} \circ \mathbf{v} \circ \mathbf{u} + \mathbf{v} \circ \mathbf{u} \circ \mathbf{u},$$

dove $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$. Si vede facilmente che

$$T = \lim_{n \to \infty} T_n,$$

dove T_n denota il tensore di rango 2

$$T_n = n \left(\mathbf{u} + \frac{1}{n} \mathbf{v} \right) \circ \left(\mathbf{u} + \frac{1}{n} \mathbf{v} \right) \circ \left(\mathbf{u} + \frac{1}{n} \mathbf{v} \right) - n \, \mathbf{u} \circ \mathbf{u} \circ \mathbf{u}.$$

Quindi, in un intorno arbitrariamente piccolo di un tensore di rango 3 ci sono infiniti tensori di rango 2!

Decomposizioni tensoriali (cont.)

Osserviamo che tale fenomeno è impossibile nel caso di matrici: per una matrice di rango k la distanza dalla più vicina matrice di rango $k-1$ è una quantità positiva fissata (per il Teorema di Eckart-Young).

Sfortunatamente, non esiste un vero e proprio analogo della SVD per tensori a tre o più indici.

Esistono però in letteratura molte proposte di decomposizioni che cercano di "imitare" almeno alcune delle buone proprietà della SVD, e che possono essere utilizzate per costruire approssimazioni di rango basso di un tensore, anche se non "ottimali" in alcun senso della parola.

C. Hillar and L. H. Lim, *Most tensor problems are NP-hard*, J. ACM, 60 (2013), Article 45.

Decomposizione CP

La CP Decomposition (per tensori T a tre indici) consiste in una decomposizione approssimata come somma di tensori di rango 1 del tipo

$$T \approx \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \cdots + \mathbf{a}_R \circ \mathbf{b}_R \circ \mathbf{c}_R \,.$$

È dovuta a vari autori (Hitchcock, Carroll, Chang, Harshman, ...) ed è anche nota con nomi come CANDECOMP, PARAFAC, etc.

È una sorta di SVD dove però i vettori $\mathbf{a}_i$, $\mathbf{b}_i$ e $\mathbf{c}_i$ non sono più ortogonali; vengono meno l'invarianza ortogonale e il Teorema di Eckart-Young sull'ottimalità dell'approssimazione di rango R.
Uno dei pochi risultati noti è dovuto a Martin Kruskal, che ha dimostrato che la migliore approssimazione CP di rango 1, se esiste, è unica.

Nonostante la scarsità di risultati teorici, la Decomposizione CP è la più utile tra quelle proposte.

Decomposizione di Tucker

Un altro approccio, dovuto a L. Tucker, consiste nell'approssimare un tensore T a tre indici con combinazioni lineari più generali di tensori di rango 1:

$$T \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}\, \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r \,,$$

dove i vettori $\{\mathbf{a}_p\}$, $\{\mathbf{b}_q\}$ and $\{\mathbf{c}_r\}$ sono ora ortogonali. Il tensore $G = (g_{pqr}) \in \mathbb{R}^{P \times Q \times R}$ è detto il core tensor della decomposizione.

Nella decomposizione CP, G è diagonale e i vettori non sono in generale ortogonali.

Altre decomposizioni tensoriali

- La High-Order SVD Decomposition (HOSVD, De Lathauwer et al.), un tipo particolare di decomposizione di Tucker.
- Per d grande, la decomposizione CP è proibitiva per tensori T a d indici. La Tensor Train Decomposition (Oseledets e Tyrtyshnikov) permette di scrivere T come un "treno" di tensori a 3 indici, cui si può efficientemente applicare una decomposizione CP.
- La CURT Decomposition, un'estensione ai tensori della CUR Decomposition per matrici, in parte basata su tecniche di randomizzazione.

Per dettagli, metodi di calcolo e riferimenti bibliografici si rimanda al libro di Gil Strang, *Linear Algebra and Learning from Data*, Wellesley, Cambridge, MA, 2019.

Bibliografia

L. Eldén, *Matrix Methods in Data Mining and Pattern Recognition*, Society for Industrial and Applied Mathematics, Philadelphia, 2007.

G. H. Golub and C. F. Van Loan, *Matrix Computations. Fourth Edition*, Johns Hopkins University Press, Baltimore, MD, 2013.

C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

G. Strang, *Linear Algebra and Learning from Data*, Wellesley, Cambridge, MA, 2019.