Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

Fattorizzazione Matriciale Non Negativa: algoritmi e applicazioni

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.ssa
VALERIA SIMONCINI

Presentata da:
LUCA CAMANZI

IV Sessione
Anno Accademico 2018/2019

Indice

Capitolo 1

Preambolo

1.1 Introduzione

Questa tesi ha lo scopo di presentare una classe di metodi di approssimazione a rango ridotto resa popolare nel 1999 da Lee e Seung, ovvero la fattorizzazione matriciale non negativa (NMF). Data una matrice X a termini non negativi il problema si riconduce alla ricerca di due matrici anch'esse a termini non negativi, che chiameremo W e H, il cui prodotto approssimi al meglio la matrice di partenza X. La difficoltà che si incontra nel calcolo di questa approssimazione è ciò che ha stimolato lo studio di algoritmi di ottimizzazione per approssimare numericamente la NMF, questione che sarà affrontata nel capitolo 3. I risultati sperimentali hanno fatto emergere una particolare proprietà di questa fattorizzazione, ovvero la tendenza a generare matrici sparse come soluzione. Nel capitolo 2 descriviamo metodi di riduzione di dimensione lineare che possono fornire approssimazioni di rango basso più accurate, spesso usate nell'ambito del data mining e del machine learning; la fattorizzazione non negativa in questo contesto si caratterizza per la maggiore interpretabilità delle sue soluzioni e fruibilità in ambito applicativo. Per comprendere appieno questa proprietà è utile presentare un esempio:

- *Text mining*: Fissato un insieme di m termini presenti in una collezione di n documenti possiamo identificare un singolo documento con un vettore di dimensione m le cui componenti rappresentano le occorrenze di ogni termine al suo interno. Per cui la matrice di dimensione $m \times n$ in cui le colonne rappresentano i vari documenti, chiamata matrice *termini* $\times$ *documenti*, conterrà solo elementi non negativi. Uno

degli obiettivi del text mining è quello di trovare le tematiche principali, ovvero i gruppi di termini collegati, presenti in tutta la collezione.
Per capire le motivazioni dietro la NMF è utile invece risolvere il problema inverso, ovvero partire dalle tematiche, supponendo di averne k, per ricomporre i documenti. Vorremo rappresentare intuitivamente le tematiche attraverso dei vettori di dimensione m, ovvero il numero di termini, in modo che le componenti siano:

- > 0 se il termine è legato alla tematica;
- 0 altrimenti.

Quindi anche le tematiche saranno vettori non negativi. Per poter passare dalle tematiche ai documenti abbiamo bisogno di sapere il "peso" di ogni tematica all'interno del documento fissato. Per fare ciò definiamo un vettore di dimensione pari a k, cioè il numero di tematiche, che abbia come componenti:

- > 0 se la tematica è affrontata nel documento, con valore crescente in base all'importanza assunta;
- 0 altrimenti

Anche in questo caso il vettore ha componenti non negative. La fattorizzazione non negativa in questo ambito risolve il problema inverso, per cui a partire dalla matrice $termini \times documenti$ verranno generate la matrice delle tematiche $W \in \mathbb{R}^{m \times k}$ e la matrice dei "pesi" $H \in \mathbb{R}^{k \times n}$ non negative in modo che l'interpretazione intuitiva sia quella appena descritta.

A questo esempio e al riconoscimento facciale sarà dedicata un'analisi approfondita nel capitolo 4 dedicato alle applicazioni della NMF.
Infine, nel capitolo 5 introdurremo una variazione del problema originario, chiamata **ONMF** (*Orthogonal Nonnegative Matrix Factorization*), che permetterà di inquadrare la NMF all'interno dei metodi di clustering mostrando l'equivalenza matematica tra l'ONMF e una variante del metodo delle k-medie sferiche.

1.2 Notazioni

Introduciamo alcune definizioni che verranno utilizzate durante tutto l'elaborato.

Definizione 1.1 (Matrice Non Negativa)**.** Sia A una matrice $m \times n$ a coefficienti reali. Allora A è una matrice non negativa se tutti i suoi elementi sono non negativi, cioè se:

$$a_{ij} \geq 0 \qquad \forall i = 1, ..., m, \ \forall j = 1, ..., n.$$

Nel corso della trattazione denoteremo con $A \geq 0$ una matrice A non negativa e con $\mathbb{R}_+^{m \times n}$ l'insieme della matrici a coefficienti reali non negative di dimensione $m \times n$.

Definizione 1.2 (Matrice Unitaria e Matrice Ortogonale)**.** Sia $A \in \mathbb{C}^{n \times n}$. Allora diciamo che A è **unitaria** se è invertibile e:

$$AA^* = A^*A = I_n \ ,$$

dove A^* indica la matrice trasposta coniugata di A e I_n indica la matrice identità $n \times n$. In particolare se $A \in \mathbb{R}^{n \times n}$, diciamo che A è **ortogonale** se è invertibile e:

$$AA^T = A^TA = I_n \ ,$$

dove A^T indica la matrice trasposta di A.
Se la matrice a valori reali A è ortogonale è anche unitaria e inoltre la sua inversa coincide con la trasposta, cosicché le sue colonne formano una base ortonormale di $\mathbb{R}^n$.

Definizione 1.3 (Matrice Sparsa e Piena)**.** Sia $A \in \mathbb{R}^{m \times n}$. Allora diciamo che A è **sparsa** se per ogni riga di A, gli elementi non nulli sono solo il $3 - 5\%$; in altre parole se $\forall j = 1, ..., n \quad a_{ij} = 0$ per quasi tutti gli indici $i = 1, ..., m$. Se A non è sparsa allora è detta **piena**.

Definizione 1.4 (Norma-1 e Norma-2 di vettori)**.** Sia $x = [x_1, x_2, ..., x_n]^T \in \mathbb{R}^n$, si definisce la **norma-1** di x come:

$$\|x\|_1 := \sum_{i=1}^{n} |x_i|$$

e la **norma-2** o **Euclidea** di x come:

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

Definizione 1.5 (Norma matriciale indotta e di Frobenius)**.** Sia $A \in \mathbb{R}^{m\times n}$, si definisce **norma-2 matriciale indotta** di A come:

$$\|A\|_2 := \max_{0\neq x\in\mathbb{R}^n} \frac{\|Ax\|_2}{\|x\|_2} = \max_{\substack{x\in\mathbb{R}^n \\ \|x\|_2=1}} \|Ax\|_2$$

e la norma matriciale di **Frobenius** come:

$$\|A\|_F := \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{i,j}|^2\right)^{\frac{1}{2}}.$$

Definizione 1.6 (Traccia di una matrice)**.** Sia $A \in \mathbb{R}^{n\times n}$, si definisce la **traccia** di A come la somma degli elementi sulla diagonale principale, cioè:

$$tr(A) := \sum_{i=1}^{n} a_{i,i}.$$

Definizione 1.7 (Problema di ottimizzazione Convessa)**.** Un problema di ottimizzazione convessa è un problema di ottimizzazione in cui l'obiettivo è trovare una soluzione x^* che verifichi:

$$\inf_{x\in C} f(x),$$

dove $f : \mathbb{R}^n \to \mathbb{R}$, detta funzione oggetto, è una funzione convessa e C è un insieme convesso.
Questi problemi in particolare godono della proprietà che se f ammette un minimo locale $x_{min} \in C$ allora esso è anche un minimo globale.

Definizione 1.8 (Problema NP-hard)**.** In teoria della complessità definiamo **NP-hard** un problema decisionale H tale che:

$$\forall L \in \text{NP-completo}, \quad L \leq_T H \ ,$$

dove NP-completo indica l'insieme dei problemi risolvibili in tempo polinomiale da una macchina di Turing dotata di un oracolo per H mentre la notazione $L \leq_T H$ indica che il problema L è polinomialmente riducibile ad H.
Poiché è possibile formulare un problema di ottimizzazione come problema decisionale concludiamo che la complessità di un problema di ottimizzazione sarà almeno pari a quello del problema di decisione ad esso associato.

Capitolo 2

Approssimazione a rango ridotto

2.1 Rappresentazione dei dati

Il primo problema da affrontare nello studio dei dati è la loro effettiva rappresentazione attraverso le matrici.

Definizione 2.1 (Matrice dei Dati)**.** Sia $x \in \mathbb{R}^m$ un vettore di m variabili che rappresenta numericamente un **dato**. Un campione di n dati, detto **dataset**, è quindi rappresentabile attraverso un insieme $\{x\}_{j=1}^{n}$ di vettori della stessa dimensione che permettono di costruire una matrice $M \in \mathbb{R}^{m \times n}$ in questo modo:

$$M = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix},$$

dove $x_{i,j}$ indica il valore dell'i-esimo vettore per la j-esima variabile e può assumere valori discreti (interi), continui oppure di dicotomia (0/1). Indicando con x_i l'i-esima colonna di M definiamo la **matrice dei dati** X come una delle seguenti:

$$X = [x_1, x_2, \cdots, x_n] = M \qquad \text{oppure} \qquad X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} = M^T.$$

Nel corso della trattazione utilizzeremo la prima di queste notazioni e in particolare nel capitolo 4 descriveremo come è formata la matrice dei dati per ogni tipo di applicazione che analizzeremo.

2.2 Riduzione di Dimensione Lineare

Dato un campione di vettori $\{x_j\}_{j=1}^n \in \mathbb{R}^m$ ed una dimensione $k < \min\{m, n\}$, la riduzione di dimensione lineare (LDR) consiste nel trovare k vettori $\{w_i\}_{i=1}^k \in \mathbb{R}^m$, detti vettori di base, tali che lo spazio generato dai w_i approssimi il più possibile lo spazio dei dati, cioè che $\exists\{h_j\}_{j=1}^n \in \mathbb{R}^k$ per cui:

$$x_j \approx \sum_{i=1}^{k} w_i h_j(i), \tag{2.1}$$

dove $h_j(i)$ indica la i-esima componente di h_j.
In pratica, supponendo che i $\{w_i\}_{i=1}^k$ siano linearmente indipendenti, stiamo rappresentando l'insieme dei vettori $\{x_j\}_{j=1}^n$ m-dimensionali in un sottospazio di dimensione $k < m$ la cui base è formata dai vettori w_i e in cui le coordinate sono fornite dagli h_j. Passando alla forma matriciale si ha:

- $X = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{m \times n}$ è la matrice dei dati le cui colonne rappresentano i vettori $\{x_j\}_{j=1}^n$;

- $W = [w_1, w_2, \cdots, w_k] \in \mathbb{R}^{m \times k}$ le cui colonne rappresentano i vettori di base $\{w_i\}_{i=1}^k$;

- $H = [h_1, h_2, \cdots, h_n] \in \mathbb{R}^{k \times n}$ le cui colonne rappresentano le coordinate $\{h_j\}_{j=1}^n$ relative ai vettore $\{x_j\}_{j=1}^n$ nella base W.

Per cui la scrittura (2.1) equivale in forma matriciale a:

$$X \approx WH.$$

Da questa espressione risulta evidente come la riduzione di dimensione lineare sia in relazione con un'approssimazione a rango ridotto k.

2.3 Decomposizione ai Valori Singolari

Introduciamo ora la **SVD** (*Singular Value Decomposition*), un importante strumento di fattorizzazione di matrici di qualsiasi dimensione basato sull'uso di autovalori e autovettori. Da questo risultato dedurremo un primo esempio di approssimazione a rango ridotto che prende il nome di **SVD troncata**.

Teorema 2.2 (Teorema di Esistenza dell'SVD)**.** *Sia $A \in \mathbb{C}^{m\times n}$ e sia $q = \min\{m, n\}$. Allora esistono una matrice $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_q, 0, \cdots, 0) \in \mathbb{R}^{m\times n}$ con $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_q$ e due matrici unitarie $U \in \mathbb{C}^{m\times m}, V \in \mathbb{C}^{n\times n}$ tali che:*

$$A = U\Sigma V^*. \tag{2.2}$$

La fattorizzazione (2.2) è detta **decomposizione ai valori singolari di A**.
Le colonne $\{u_i\}_{i=1}^m$ di U sono dette *vettori singolari sinistri* di A, le colonne $\{v_j\}_{j=1}^n$ di V *vettori singolari destri* e gli scalari $\{\sigma_k\}_{k=1}^q$ *valori singolari.*

Corollario 2.3. Nel caso $A \in \mathbb{R}^{m\times n}$, esistono $U \in \mathbb{R}^{m\times m}$,$V \in \mathbb{R}^{n\times n}$ ortogonali tali che:

$$A = U\Sigma V^T. \tag{2.3}$$

Prima di dimostrare il teorema 2.2 premettiamo qualche osservazione:

Osservazione 2.3.1. Poiché U e V sono ortogonali si ha:

$$\|A\|_F^2 = \left\|U\Sigma V^T\right\|_F^2 = \left\|\Sigma V^T\right\|_F^2 = \|\Sigma\|_F^2 = \textstyle\sum_{i=1}^q \sigma_i^2.$$

Osservazione 2.3.2. La decomposizione ci permette di scrivere: $A = \sum_{i=1}^{q} u_i \sigma_i v_i^T$, ovvero considerare A come somma di matrici di rango 1.

Osservazione 2.3.3.

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \sigma_1.$$

Infatti si ha che:

$$\frac{\|Ax\|_2^2}{\|x\|_2^2} = \frac{\left\|U\Sigma V^T x\right\|_2^2}{\|x\|_2^2} = \frac{\left\|\Sigma V^T x\right\|_2^2}{\|V^T x\|_2^2} \overset{y=V^T x}{=} \frac{\|\Sigma y\|_2^2}{\|y\|_2^2} =$$
$$= \frac{\sum_{i=1}^{n} \sigma_i^2 y_i^2}{\|y\|_2^2} \leq \frac{\sum_{i=1}^{n} \sigma_1^2 y_i^2}{\|y\|_2^2} = \sigma_1^2 \frac{\|y\|_2^2}{\|y\|_2^2} = \sigma_1^2.$$

E il valore singolare σ_1 è raggiunto dal vettore singolare destro $x = v_1$, infatti essendo $AV = U\Sigma$:

$$\|Av_1\|_2 = \|u_1\sigma_1\|_2 = \sigma_1.$$

Dimostrazione del Teorema 2.2. Possiamo supporre $m \geq n$ poiché in caso contrario basterà applicare il teorema ad A^*. Consideriamo il problema $\max_{\|x\|_2=1} \|Ax\|_2$ e sia x^* il vettore soluzione che esiste per il teorema di Weierstrass.
Ricordando l'Osservazione 2.3.3 definiamo y come il vettore di norma unitaria di $\mathbb{R}^m$ tale che $Ax^* = \sigma_1 y$.
Definiamo quindi $X_1 = \left[x^*, \widetilde{X_2}\right] \in \mathbb{C}^{n\times n}$ e $Y_1 = \left[y, \widetilde{Y_2}\right] \in \mathbb{C}^{m\times m}$ in modo che siano entrambe unitarie. Quindi:

$$A_1 := Y_1^T A X_1 = \begin{bmatrix} \sigma_1 & d^T \\ 0 & B \end{bmatrix}$$

(dato che $y^T Ax = \sigma_1$ e $\widetilde{Y}_2^T Ax = \sigma_1 \widetilde{Y}_2 y = 0$).
Osserviamo che $\frac{\|Ax\|_2^2}{\|x\|_2^2} = \frac{\|A_1 x\|_2^2}{\|x\|_2^2}$ poiché A_1 è ottenuta da A attraverso trasformazioni unitarie.

Sia ora $x = (\sigma_1,\ d)^T \in \mathbb{C}^n$ con $d \in \mathbb{C}^{n-1}$. Allora si ha:

$$\frac{\|Ax\|_2^2}{\|x\|_2^2} = \frac{\left\|A_1 \cdot (\sigma_1,\ d)^T\right\|_2^2}{\|(\sigma_1,\ d)^T\|_2^2} = \frac{1}{\sigma_1^2 + d^T d}\left\|A_1 \cdot (\sigma_1, d)^T\right\|_2^2$$
$$= \frac{1}{\sigma_1^2 + d^T d}\left\|\left[\sigma_1^2 + d^T d; B \cdot d\right]\right\|_2^2 \ \geq \sigma_1^2 + d^T d.$$

Poiché $\sigma_1 = \max \frac{\|Ax\|_2}{\|x\|_2}$ allora, per non avere contraddizioni, si ottiene $d = 0$. Quindi sia la prima riga che la prima colonna di A_1 sono zero, eccetto l'elemento diagonale.
La procedura prosegue in modo iterativo con B al posto di A, ottenendo infine
$U = Y_1 \cdot ... \cdot Y_{n-1}$ e $V = X_1 \cdot ... \cdot X_{m-1}$. □

Definizione 2.4. Sia $A \in \mathbb{R}^{m\times n}$, $q = \min\{m, n\}$ e sia $A = U\Sigma V^T$ la sua decomposizione in valori singolari. Fissato $k \in \{1, 2, ..., q-1\}$ si definisce **SVD troncata** di A come la matrice A_k di rango k tale che:

$$A \approx A_k = U\begin{bmatrix}\Sigma_k & 0\\ 0 & 0\end{bmatrix}V^T = \begin{bmatrix}U_k & \widehat{U}\end{bmatrix}\begin{bmatrix}\Sigma_k & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_k^T\\ \widehat{V}\end{bmatrix} = U_k\Sigma_k V_k^T, \tag{2.4}$$

dove $U_k \in \mathbb{R}^{m\times k}, \Sigma_k = diag(\sigma_1, ..., \sigma_k) \in \mathbb{R}^{k\times k}, V_k \in \mathbb{R}^{n\times k}$.

Osservazione 2.3.4. Se per un certo $k \in \{1, 2, ..., q-1\}$ vale $\sigma_k \gg \sigma_{k+1}$, la SVD troncata di A di rango k è un'approssimazione particolarmente accurata di A, infatti:

$$A = \sum_{i=1}^{n} u_i\sigma_i v_i^T = \sum_{i=1}^{k} u_i\sigma_i v_i^T + \sum_{i=k+1}^{q} u_i\sigma_i v_i^T \approx \sum_{i=1}^{k} u_i\sigma_i v_i^T = A_k.$$

Osservazione 2.3.5. L'espressione (2.4) ci fornisce un primo esempio di approssimazione a rango ridotto. Si può dimostrare, inoltre, che la SVD troncata gode di un'importante proprietà.
Sia $A \in \mathbb{R}^{m\times n}$ e sia $A_k = U_k\Sigma_k V_k^T$ la SVD troncata di rango k di A, allora:

$$A_k = \underset{\substack{Z\in\mathbb{R}^{m\times n}\\ rank(Z)=k}}{\operatorname{argmin}} \|A - Z\|_F. \tag{2.5}$$

Sostituendo in (2.5) la norma di Frobenius con la norma-2 indotta la proprietà continua a valere.

2.4 Approssimazione con matrici non negative

Sia data una matrice dei dati $X \in \mathbb{R}^{m\times n}$. Riprendendo le notazioni della sezione 2.2, vogliamo determinare due matrici $W \in \mathbb{R}^{m\times k}, H \in \mathbb{R}^{k\times n}$ di rango k tali che:

$$X \approx WH.$$

Un punto focale per questo tipo di approssimazione nel *Data Mining* sta nella scelta delle ipotesi da imporre su W e H e tale scelta dipende dalla natura del problema considerato.
Abbiamo visto nell'Osservazione 2.3.5 che la SVD troncata, fissato il rango, fornisce la migliore fattorizzazione nel senso della norma-2 indotta e di Frobenius. Tuttavia, i risultati ottenuti da questo metodo possono non essere soddisfacenti in quanto alcune proprietà della matrice di partenza vengono perse. Nello specifico, se la matrice X è sparsa, la sua approssimazione X_k sarà in generale piena e di conseguenza utilizzerà una maggiore memoria del calcolatore per salvare i coefficienti. La SVD troncata rimane ciononostante un ottimo metodo per valutare l'accuratezza di tutte le altre approssimazioni grazie anche al costo computazionale non elevato.

Per lo studio della fattorizzazione non negativa (**NMF**) lavoreremo con matrici dei dati $X \geq 0$ e vorremo assicurare che anche W e H siano non negative; la SVD troncata chiaramente non potrà restituire la soluzione in quanto non possiede nessun vincolo sui coefficienti negativi.

Presentiamo quindi il problema oggetto della relazione.
Sia $X \in \mathbb{R}_+^{m\times n}$ matrice dei dati. Fissato il rango di approssimazione $k < \min(m,n)$ vogliamo risolvere:

$$\min_{\substack{W\in\mathbb{R}^{m\times k}, H\in\mathbb{R}^{k\times n}\\ W\geq 0, H\geq 0}} \|X - WH\|_F^2 . \tag{2.6}$$

Possiamo effettuare una serie di considerazioni:

- Per quanto visto nella sezione 2.2, tale fattorizzazione consente di rappresentare i vettori $\{x_j\}_{j=1}^n$ come combinazione non sottrattiva dei vettori di base rappresentati dalle colonne $\{w_i\}_{i=1}^k$ di W, dove le colonne $\{h_j\}_{j=1}^n$ di H indicheranno quindi il "peso" dei vettori di base nel comporre il dato iniziale;

- La scelta di minimizzare in norma di *Frobenius* ci permette innanzitutto di valutare la bontà dell'approssimazione attraverso il confronto con la SVD troncata e secondariamente è indicata per le applicazioni in cui il rumore N di $X = WH + N$ ha distribuzione *Gaussiana*;

- La soluzione (W, H) non è unica, infatti ogni matrice Q tale che $WQ \geq 0, Q^{-1}H \geq 0$ genera una nuova soluzione $(W' = WQ, H' = Q^{-1}H)$. Tuttavia se Q è una permutazione di una matrice diagonale questo non genera problemi poiché corrisponde ad una scalatura e permutazione dei fattori di rango 1 $\{W(:, j)H(j, :)\}_{j=1}^k$ che non altera i risultati nelle applicazioni. In altri casi, invece, bisognerà imporre dei vincoli di sparsità per evitare di generare soluzioni di diverso tipo;

- Il rango di approssimazione k non è in generale noto a priori; per stabilirlo in modo ottimale occorre fare diverse prove per stabilire in quale caso il residuo relativo $\frac{\|X-WH\|_F}{\|X\|_F}$ sia minore. Un altro metodo per scegliere il rango è osservare l'andamento dei valori singolari in modo da considerare solo i primi k valori per cui la decrescita è rapida.

Capitolo 3

Algoritmi per la NMF

3.1 Considerazioni computazionali

Riprendiamo il problema (NMF):

$$\min_{\substack{W\in\mathbb{R}^{m\times k},H\in\mathbb{R}^{k\times n}\\ W\geq 0,H\geq 0}} \|X - WH\|_F^2 \tag{3.1}$$

Si tratta di un problema non lineare e non convesso (in W e in H) per cui non abbiamo a priori la convergenza ad un minimo globale; inoltre, a differenza della SVD troncata, la complessità è in generale NP-hard.
La presenza di molteplici minimi locali fa sì che l'inizializzazione delle matrici W o H all'interno degli algoritmi risulti determinante al fine del della loro convergenza. Affronteremo tale questione nella sezione 3.4.

Definendo $F(W,H) := \frac{1}{2}\|X - WH\|_F^2$ la funzione oggetto con X fissata, la condizioni di ottimalità di (3.1) del primo ordine, riportate nell'articolo [3], sono:

$$W \geq 0, \quad \nabla_W F = WHH^T - XH^T \geq 0, \quad W \circ \nabla_W F = 0, \tag{3.2}$$

$$H \geq 0, \quad \nabla_W F = W^TWH - W^TX \geq 0, \quad H \circ \nabla_H F = 0, \tag{3.3}$$

dove $\circ$ indica il prodotto componente per componente.
Le equazioni (3.2) e (3.3) indicano le condizioni necessarie affinché una soluzione (W^*, H^*) di (3.1) sia stazionaria. Queste conclusioni ci forniscono una giustificazione formale al

fatto che la NMF generi fattori **sparsi**, infatti $\forall i, j$ gli elementi w_{ij}^* e h_{ij}^* delle soluzioni stazionarie o sono nulli o dovranno esserlo i corrispondenti elementi (i, j) della derivata parziale di F rispetto alla matrice stessa W^* o H^*.

Osserviamo che la minimizzazione in norma di *Frobenius* permette di trasferire il problema (3.1) alla risoluzione di n problemi ai **minimi quadrati**, infatti:

$$\|X - WH\|_F^2 = \sum_{j=1}^{n} \sum_{i=1}^{m} |x_{ij} - (WH)_{ij}|^2 = \sum_{j=1}^{n} \|x_j - Wh_j\|_2^2 , \tag{3.4}$$

dove x_j e h_j indicano la j-esima colonna rispettivamente di X e di H mentre $(WH)_{i,j}$ denota la componente (i, j) della matrice WH.
Inoltre notiamo che il problema NMF è simmetrico in W e in H, infatti:

$$\|X - WH\|_F^2 = \left\|X^T - H^T W^T\right\|_F^2 .$$

Questa uguaglianza ci consente di utilizzare lo stesso algoritmo per risolvere il problema ai minimi quadrati rispetto a W e rispetto ad H con l'accortezza di aggiungere le trasposizioni e scambiare l'ordine dei fattori.

3.2 Algoritmi a schema alternato

Chiameremo NNLS (Non Negative Least Squares) entrambi i seguenti problemi di minimizzazione:

$$\min_{H \geq 0} \|X - WH\|_F^2 = \sum_{j=1}^{n} \min_{h_j \in \mathbb{R}_+^k} \|x_j - Wh_j\|_2^2 , \tag{3.5a}$$

$$\min_{W \geq 0} \left\|X^T - H^T W^T\right\|_F^2 = \sum_{i=1}^{m} \min_{w_{i,:}^T \in \mathbb{R}_+^k} \left\|x_{i,:}^T - H^T w_{i,:}^T\right\|_2^2 , \tag{3.5b}$$

dove la notazione $x_{i,:}$ sta ad indicare la i-esima riga della matrice X.

Possiamo quindi trasformare la NMF in un problema convesso adottando uno schema a doppio blocco discendente, ovvero risolvendo NNLS (3.5) in maniera alternata prima

rispetto ad un fattore e poi rispetto all'altro.
L'algoritmo a schema alternato standard per la NMF avrà questa forma:

INPUT: Matrice dei dati $X \in \mathbb{R}_+^{m \times n}$ e rango di fattorizzazione k

---

OUTPUT: $W \in \mathbb{R}_+^{m \times k}, H \in \mathbb{R}_+^{k \times n}$ tali che valga $X \approx WH$

---

1. Inizializzare $W^{(0)} \in \mathbb{R}_+^{m \times k}$ (vedi sezione 3.4)
2. Per t=1,2,. . . fino a convergenza:
   (a) Aggiornare $H^{(t)}$ in funzione di $(X, W^{(t-1)})$ risolvendo (3.5a)
   (b) Aggiornare $W^{(t)}$ in funzione di $(X, H^{(t)})$ risolvendo (3.5b)

Presentiamo un paio di esempi di algoritmi di questo tipo che differiscono per il metodo di risoluzione di NNLS.

3.2.1 Minimi quadrati alternati (*Alternating Least Squares ALS*)

Il metodo dei minimi quadrati alternati consiste nel risolvere il problema ai minimi quadrati senza vincolo di non negatività e solo successivamente porre uguale a zero gli elementi negativi. In pratica aggiorniamo la matrice H (e allo stesso modo W) come segue:

$$H \longleftarrow \max(\operatorname*{argmin}_{Z \in \mathbb{R}^{k \times n}} \|X - WZ\|_F, 0),$$

dove il massimo è inteso su ogni elemento della matrice.

L'algoritmo così strutturato è semplice da implementare ed ha basso costo computazionale, tuttavia solitamente per matrici piene non converge. Inoltre la proiezione sullo spazio delle matrici non negative genera problemi di scalatura della soluzione, per cui risulta necessario introdurre un fattore correttivo α^* tale che:

$$\alpha^* = \operatorname*{argmin}_{\alpha \geq 0} \|X - \alpha WH\|.$$

I problemi di convergenza rendono questo metodo non ottimale per la risoluzione della NMF, tuttavia alcune iterazioni di questo algoritmo possono essere utilizzate prima di

passare ad un algoritmo più sofisticato per fornire una stima iniziale di W e H a basso costo computazionale, specialmente per matrici dei dati sparse.

3.2.2 Minimi quadrati non negativi alternati (*Alternating Nonnegative Least Squares ANLS*)

I minimi quadrati non negativi alternati sono una gruppo di metodi che risolvono esattamente i problemi convessi in W e H, per cui l'aggiornamento delle matrici è il seguente:

$$H \longleftarrow \underset{H \geq 0}{\operatorname{argmin}} \|X - WH\|_F .$$

Per questo problema su MatLab è presente l'implementazione della funzione `lsqnonneg` che utilizza un metodo di tipo active-set. A differenza dell'algoritmo ALS, l'esattezza dell'ANLS causa la maggiore decrescita dell'errore per ogni iterazione e la convergenza ad una soluzione stazionaria è garantita, seppur il costo computazionale sia notevolmente più elevato. Queste caratteristiche rendono l'ANLS adatto per rifinire l'approssimazione di W e H, mentre il suo apporto nelle fasi iniziali è inutilmente dispendioso. Esperimenti numerici riportati nell'articolo [3] hanno evidenziato che il metodo è più indicato per il trattamento di matrici dense, mentre nel caso di matrici sparse i risultati sono mediocri.

3.3 Algoritmi con penalizzazione

Una variante dello schema alternato classico consiste nell'introduzione nel problema originale di un termine di penalizzazione che controlli la regolarità delle colonne di W e di H. Infatti in base al tipo di applicazione possiamo chiedere che i vettori di base $\{w_i\}_{i=1}^{k}$ od i coefficienti $\{h_j\}_{j=1}^{n}$ abbiano poche componenti significative per fornire una migliore interpretazione dei dati e ridurre la quantità di memoria utilizzata.
Presentiamo un paio di esempi di algoritmi recentemente sviluppati :

3.3.1 Minimi quadrati forzati alternati
Alternating Constrained Least Squares (ACLS)

Il metodo dei minimi quadrati alternati penalizzati consiste nella risoluzione dei seguente problemi di minimo:

$$\min_{h_j \geq 0} \left[\|x_j - W h_j\|_2^2 + \lambda_H \|h_j\|_2^2 \right].$$

dove $\lambda_H \geq 0$ è il parametro di penalizzazione dato a priori su H che è tanto più grande quanto meno chiediamo che H sia densa.
Notiamo che il termine di penalizzazione legato alla sparsità è dato dal quadrato della norma euclidea delle colonne di H. Al passaggio seguente dovremo aggiornare la matrice W introducendo l'analogo parametro $\lambda_W \geq 0$:

$$\min_{w_j^T \geq 0} \left[\left\|x_j^T - w_j^T H\right\|_2^2 + \lambda_W \left\|w_j^T\right\|_2^2 \right].$$

Come nel caso non penalizzato possiamo decidere se risolvere NNLS (3.5) attraverso la proiezione sullo spazio delle matrici non negative (vedi ALS) o se risolvere il problema in modo esatto (vedi ANLS). Per semplificare l'implementazione e accelerare l'esecuzione dell'algoritmo seguiremo la prima opzione.
Matematicamente si vuole risolvere un problema di minimo del funzionale F così definito:

$$F(h_j) := \|x_j - W h_j\|_2^2 + \lambda_H \|h_j\|_2^2 = h_j^T (W^T W + \lambda_H I_k) h_j - 2 h_j^T W^T x_j + x_j^T x_j.$$

Per cui ricaviamo la soluzione h_j^* di $\nabla F(h_j^*) = 0$ risolvendo il sistema lineare di ordine k:

$$(W^T W + \lambda_H I_k)\ h_j^* = W^T x_j. \tag{3.6}$$

Analogamente, per trovare i vettori riga $w_{i,:}^*$:

$$(H H^T + \lambda_W I_k)\ (w_{i,:}^*)^T = H x_{i,:}^T. \tag{3.7}$$

Quindi l'algoritmo si presenta in questo modo:

Algoritmo ACLS:

INPUT: Matrice dei dati $X \in \mathbb{R}_+^{m \times n}$, rango di fattorizzazione k, parametri di

penalizzazione $\lambda_H, \lambda_W \geq 0$

OUTPUT: $W \in \mathbb{R}_+^{m \times k}, H \in \mathbb{R}_+^{k \times n}$ sparse tali che valga $X \approx WH$

1. Inizializzare $W^{(0)} \in \mathbb{R}_+^{m \times k}$ (vedi sezione 3.4)
2. Per t=1,2,. . . fino a convergenza:
   (a) Risolvere (3.6) per $j = 1, ..., n$, ovvero trovare la soluzione $H^{(t)}$ del sistema di equazioni:
   $(W^T W + \lambda_H I_k)H = W^T X$
   (b) Imporre $H^{(t)} \longleftarrow \max(H^{(t)}, 0)$
   (c) Risolvere (3.7) per $i = 1, ..., m$, ovvero trovare la soluzione $W^{(t)}$ del sistema di equazioni:
   $(HH^T + \lambda_W I_k)\, W^T = HX^T$
   (d) Imporre $W^{(t)} \longleftarrow \max(W^{(t)}, 0)$

3.3.2 Minimi quadrati forzati alternati di Hoyer *(Alternating Hoyer Constrained Least Squares AHCLS)*

Un metodo più sofisticato per penalizzare la sparsità è stato introdotto da Hoyer [2], che ha fornito la seguente definizione:
Dato il vettore $v \in \mathbb{R}^n$, definiamo la sparsità di v come

$$spar(v) := \frac{\sqrt{n} - \frac{\|v\|_1}{\|v\|_2}}{\sqrt{n} - 1}.$$

Osserviamo che:

$$\begin{cases} 0 \leq spar(v) \leq 1 \quad \forall v \in \mathbb{R}^n; \\ spar(v) = 1 \Leftrightarrow \exists\, !\ i \in \{1, ..., n\} \text{ t.c. } v(i) \neq 0; \\ spar(v) = 0 \Leftrightarrow \forall i \in \{1, ..., n\} \quad v(i) = \pm c, \text{ con } c \in \mathbb{R} \text{ costante.} \end{cases}$$

Con questa definizione è stata creata una scala normalizzata di sparsità in cui al valore 1 è associata la massima sparsità ed al valore 0 la minima (vedi Figura 3.1).

Figura 3.1: Illustrazione di vari livelli di sparsità. Le barre rappresentano il valore della componente del vettore. Bassi livelli di sparsità implicano meno differenze tra le componenti, viceversa alti livelli portano alcune componenti ad annullarsi.

Se stabiliamo a priori un valore di sparsità per le colonne di H e per le righe di W, che indichiamo rispettivamente con α_H e α_W, valgono le seguenti equivalenze :

$$\begin{aligned}\alpha_H \in (0,1), \quad \alpha_H = spar(h_j) &\iff (\alpha_H - \sqrt{k}(\alpha_H - 1))^2 \frac{\|h_j\|_1^2}{\|h_j\|_2^2} = 0 \\ &\iff \underbrace{(\alpha_H - \sqrt{k}(\alpha_H - 1))}_{\gamma_H}{}^2 \ \|h_j\|_2^2 - \|h_j\|_1^2 = 0 \\ &\iff \gamma_H^2 h_j^T h_j - h_j^T \ \mathbf{1}\mathbf{1}^T h_j = 0 \ ,\end{aligned}$$

dove $\mathbf{1}$ indica il vettore $(1, 1, ..., 1)^T \in \mathbb{R}^k$.

Otteniamo risultati analoghi partendo dalle righe $\{w_{i,:}\}_{i=1}^m$ di W.

Similmente al metodo ACLS anche quello di Hoyer si basa sulla risoluzione dei seguenti problemi di minimo:

$$\min_{h_j \geq 0} \left[\|x_j - W h_j\|_2^2 + \lambda_H (\gamma_H^2 h_j^T h_j - h_j^T \mathbf{1}\mathbf{1}^T h_j) \right] \qquad \text{con } \lambda_H \geq 0.$$

Definendo quindi il funzionale F in questo modo:

$$\begin{aligned}F(h_j) :&= \|x_j - W h_j\|_2^2 + \lambda_H (\gamma_H^2 h_j^T h_j - h_j^T \mathbf{1}\mathbf{1}^T h_j) \\ &= h_j^T (W^T W + \lambda_H (\gamma_H^2 I_k - 11^T)) h_j - 2 h_j^T W^T x_j + x_j^T x_j \ ,\end{aligned}$$

cerchiamo il vettore h_j^* che risolve il minimo imponendo $\nabla F(h_j^*) = 0$ e ottenendo il sistema lineare:

$$(W^T W + \lambda_H (\gamma_H^2 I_k - 11^T)) h_j^* = W^T x_j. \tag{3.8}$$

Analogamente, per trovare i vettori riga $w^*_{i,:}$ otteniamo:

$$(HH^T + \lambda_W(\gamma_W^2 I_k - 11^T))(w^*_{i,:})^T = Hx^T_{i,:}. \tag{3.9}$$

Quindi l'algoritmo si presenta in questo modo:

Algoritmo AHCLS:

INPUT: Matrice dei dati $X \in \mathbb{R}_+^{m\times n}$, rango di fattorizzazione k, parametri di penalizzazione $\lambda_H, \lambda_W \geq 0$, coefficienti di sparsità $\alpha_H, \alpha_W \in (0, 1)$

OUTPUT: $W \in \mathbb{R}_+^{m\times k}, H \in \mathbb{R}_+^{k\times n}$ sparse tali che valga $X \approx WH$

1. Inizializzare $W^{(0)} \in \mathbb{R}_+^{m\times k}$ (vedi sezione 3.4);
2. Calcolare $\gamma_H = \alpha_H - \sqrt{k}(\alpha_H - 1)$ e $\gamma_W = \alpha_W - \sqrt{k}(\alpha_W - 1)$;
3. Per t=1,2,. . . fino a convergenza:
   (a) Risolvere (3.8) per $j = 1, ..., n$, ovvero trovare la soluzione $H^{(t)}$ del sistema di equazioni:
   $\big(W^TW + \lambda_H(\gamma_H^2 I_k - 11^T)\big)H = W^TX$
   (b) Imporre $H^{(t)} \longleftarrow \max(H^{(t)}, 0)$
   (c) Risolvere (3.9) per $i = 1, ..., m$, ovvero trovare la soluzione $W^{(t)}$ del sistema di equazioni:
   $\big(HH^T + \lambda_W(\gamma_W^2 I_k - 11^T)\big)W^T = HX^T$
   (d) Imporre $W^{(t)} \longleftarrow \max(W^{(t)}, 0)$

3.4 Inizializzazione e criteri d'arresto

Strategie di inizializzazione

A causa della non convessità della NMF la scelta iniziale della matrice W risulta determinante ai fini della convergenza dei metodi. Seppur non ci siano tuttora giustificazioni teoriche sulla bontà delle soluzioni per le diverse strategie, è però sperimentato che un'opzione iniziale più sofisticata riduca drasticamente il numero di iterazioni per

la convergenza. Presentiamo, quindi, alcune delle scelte iniziali più utilizzate in ordine crescente di costo computazionale:

1. *Scelta di un sottoinsieme delle colonne di X*: Si può inizializzare W usando i vettori iniziali, ovvero ponendo $W = X(:, \Omega)$ dove Ω ha cardinalità k pari al rango di approssimazione;

2. *Random*: Il modo più semplice è partire con gli elementi di W generati uniformemente nell'intervallo [0,1];

3. *Tecniche di Clustering*: Sfruttando alcuni metodi di clustering a basso costo come k-medie e k-medie sferiche possiamo generare la matrice W dei centroidi con il numero di cluster uguale al rango di fattorizzazione. In tal caso la matrice H sarà la matrice di partizionamento, ovvero $H_{i,j} \neq 0 \Leftrightarrow x_j$ appartiene all'i-esimo cluster.

4. *SVD troncata*: Sia $X_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T$ la miglior approssimazione di rango k della matrice dei dati X. Per il teorema di **Perron-Frobenius**, essendo $X \geq 0$ si ha che $u_1 \geq 0$ (se i primi 2 valori singolari sono diversi) e quindi possiamo porre $w_1 = u_1$. Successivamente consideriamo $u_2 \sigma_2 v_2^T$: essa ha componenti negative, ma costruendo $C^{(2)} := u_2 \sigma_2 v_2^T$: la proiezione $C_+^{(2)}$ è non negativa. Quindi riutilizzando il teorema otteniamo che $u_1(C^{(2)}) \geq 0$ e ci basterà porre $w_2 = u_1(C^{(2)})$. Possiamo iterare il procedimento fino a $w_k = u_1(C^{(k)})$.

Strategie di arresto

Sono stati ideati molteplici criteri d'arresto basati principalmente sull'andamento dell'errore e sulla differenza tra le due soluzioni iterate successive. Tra i più utilizzati troviamo:

- Residuo relativo: $\frac{\|X - WH\|_F}{\|X\|_F} < \epsilon$.
  Per accelerare il calcolo possiamo sviluppare il numeratore in questo modo:

$$\begin{aligned} \|X - WH\|_F^2 &= tr(X^T X) - 2tr(H^T(W^T X)) + tr(H^T(W^T W H) \\ &= tr(\underbrace{X^T X}_{\substack{(2m-1)n \text{ flops} \\ \text{per } n \text{ elementi} \\ \text{diagonali}}}) - 2tr((\underbrace{W^T X) H^T}_{\substack{(2mn+n-1)k \text{ flops} \\ \text{per } k \text{ elementi} \\ \text{diagonali}}}) + tr(\underbrace{(W^T W H) H^T}_{\substack{(2k(m+n)-k+n-1)k \text{ flops} \\ \text{per } k \text{ elementi} \\ \text{diagonali}}}), \end{aligned}$$

effettuando il calcolo di $tr(X^T X)$ una volta sola. Grazie a questa stratagemma la macchina dovrà calcolare e memorizzare soltanto gli elementi diagonali delle matrici evidenziate;

- Residuo scalato: $\frac{\|X-WH\|_F^{\frac{1}{2}}}{nm} < \epsilon$.
  Questa variante del residuo tiene conto della dimensione $m \times n$ della matrice dei dati;

- Confronto con la migliore approssimazione di rango k, ovvero la SVD troncata:

$$\frac{\|X - WH\|_F^2 - \rho}{\rho} < \epsilon, \quad \text{con} \quad \rho = \left\|A - U_k \Sigma_k V_k^T\right\|_F \ ;$$

- Massimo numero di iterazioni o limite massimo di tempo;

- Variazione della soluzione:

$$\max\left\{\left\|H^{(k)} - H^{(k-1)}\right\|_F, \left\|W^{(k)} - W^{(k-1)}\right\|_F\right\} < \epsilon.$$

Capitolo 4

Applicazioni

Le ragioni per cui la NMF sta riscuotendo grande successo nelle applicazioni risiedono nella sua capacità di raccogliere le caratteristiche principali dei dati generando matrici sparse e facilmente interpretabili. In questo capitolo ci concentreremo su due settori in cui la NMF è utilizzata, ovvero il **text mining** e il **riconoscimento facciale**. Altri impieghi possibili si trovano, ad esempio, nel riconoscimento di pattern, nella separazione di segnali misti, nell'analisi musicale e nella bioinformatica. Nel prossimo capitolo vedremo come, variando leggermente il problema, possiamo ottenere un metodo di clustering che applicheremo allo studio delle immagini iperspettrali.

4.1 Text Mining

Il text mining è un processo attraverso il quale riusciamo a estrarre informazioni utili da collezioni di testi di grandi dimensioni.

Per raccogliere i dati utilizzeremo la matrice *termini*$\times$*documenti*, ovvero la matrice X in cui le colonne rappresentano i documenti mentre le righe si riferiscono alle parole chiave. Nel nostro caso l'elemento (i, j) della matrice X rappresenta il numero di volte in cui il termine i-esimo appare all'interno del documento j-esimo. Osserviamo, quindi, che la matrice dei dati sarà in generale abbastanza sparsa poiché la maggior parte dei documenti utilizza un piccolo sottoinsieme di vocaboli.

Data la matrice termini $\times$ documenti $X \in \mathbb{R}^{m\times n}$ e un rango di approssimazione k, la

NMF genera i due fattori $W, H \geq 0$ tali che, $\forall\ 1 \leq j \leq n$ otteniamo:

$$\underbrace{X(:,j)}_{j\text{-esimo documento}} \approx \sum_{i=1}^{k} \underbrace{W(:,i)}_{i\text{-esima tematica}} \underbrace{H(i,j)}_{\substack{\text{importanza della } i\text{-esima} \\ \text{tematica nel } j\text{-esimo documento}}} .$$

Facciamo un paio di osservazioni:

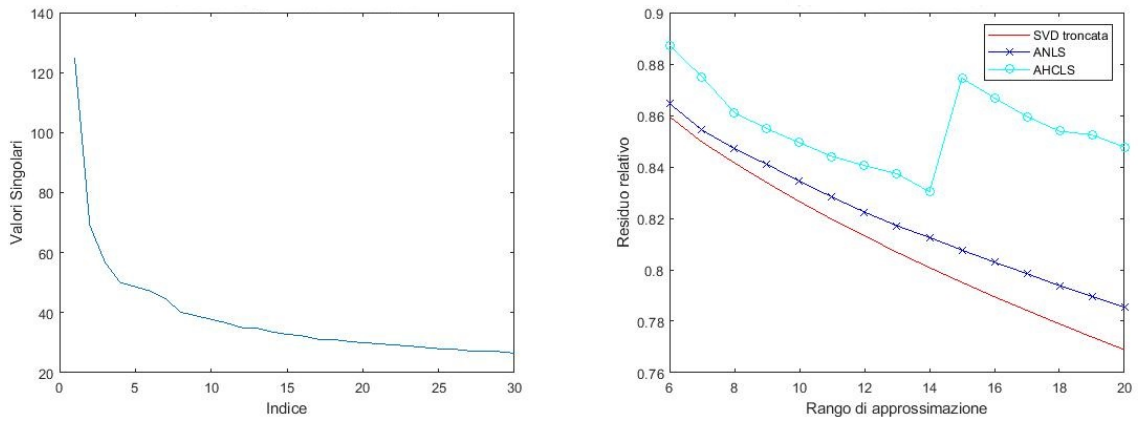
- Dato che i coefficienti della combinazione lineare sono non negativi ($H \geq 0$), si possono soltanto aggregare i gruppi di parole identificati dalle colonne di W per ricreare i documenti originali;
- Essendo il numero dei documenti presenti nel dataset maggiore dei vettori di base dati dalle colonne di W, queste ultime rappresenteranno un insieme di parole presenti parallelamente in più documenti. Possiamo quindi considerare le colonne di W come tematiche che comprendono insiemi di parole collegate e le colonne di H, non essendo ortogonali, assegnano proporzionalmente i documenti alle diverse tematiche.

Per effettuare gli esperimenti utilizzeremo il database CRANFIELD[1] che contiene documenti riguardanti l'ingegneria aerospaziale e fornisce la matrice termini $\times$ documenti $\boldsymbol{A\_cran} \in \mathbb{R}^{4563\times1398}$ sparsa e la matrice dei vocaboli $\boldsymbol{dict\_cran}$ che permette di convertire gli indici delle righe di $\boldsymbol{A\_cran}$ in parole. Specifichiamo che i dati che analizzeremo sono stati pre-elaborati, ovvero è stato effettuato un processo di "stoplisting", in cui sono stati eliminati dall'elenco dei termini quelli più frequenti (articoli, preposizioni, congiunzioni, etc). Non è stato invece effettuato il processo di "stemming", in cui si accorpano termini con la stessa radice.

Per accelerare l'esecuzione dei metodi senza perdere eccessive informazioni ridurremo la matrice $\boldsymbol{A\_cran}$ alle prime 500 colonne ottenendo $\boldsymbol{A\_cranrid} \in \mathbb{R}^{4563\times500}$.
Effettueremo gli esperimenti utilizzando gli algoritmi NMF più sofisticati tra quelli presentati, ovvero ANLS e ACHLS, con inizializzazione della matrice W ottenuta tramite il metodo della SVD troncata spiegato nella sezione 3.4. Specifichiamo che il termine

[1] Gli articoli contenuti all'interno di CRANFIELD sono stati organizzati nella matrice A_cran dai gestori del sito `http://scgroup20.ceid.upatras.gr:8000/tmg/`.

sparsità in questa sezione sarà inteso nel senso di Hoyer, quindi non riguarderà obbligatoriamente il numero dei non zeri dei vettori o delle matrici.
Cerchiamo il rango di fattorizzazione k ottimale per il nostro problema confrontando l'andamento dei valori singolari e il residuo al variare di k dei 2 algoritmi.



(a) Andamento dei primi 30 valori singolari (b) Confronto tra residui relativi

Figura 4.1: Scelta del rango di approssimazione

La decrescita dei valori singolari (Figura 4.1a) ha il classico andamento a gomito e rallenta dopo circa il decimo indice. Confrontando i residui relativi (Figura 4.1b) dei metodi ANLS e AHCLS intorno a quell'indice emerge che il primo decresce abbastanza linearmente all'aumentare dell'ordine di approssimazione mentre il secondo ha un minimo locale in $rank = 14$. Non avendo a priori indicazioni sull'ordine di approssimazione poniamo $k = 14$ in modo da ottenere delle soluzioni accurate con entrambi gli algoritmi.
Testando inizialmente l'algoritmo AHCLS con parametri $\lambda_H = \lambda_W = 0.5$, $\alpha_W = \alpha_H = 0.8$ troviamo un residuo relativo finale $r_{fin} = 0.827$ e una sparsità media delle colonne di H=0.75, che quindi è più modesta rispetto alla richiesta iniziale α_H.
Nella Tabella 4.1 riportiamo i 15 termini dominanti relativi alle prime 5 colonne di W, che indichiamo con $\{w_j\}_{j=1}^{5}$ ordinando le componenti in modo decrescente di importanza.

Questi dati forniscono un'idea delle tematiche principali affrontate nei documenti, infatti ad esempio possiamo associare w_1 all'argomento *"Aerodinamica"*, w_2 a *"Fenomeni ondulatori dei fluidi"*, w_3 a *"Forma e movimenti"*, etc.

$w_1(1:15)$	$w_2(1:15)$	$w_3(1:15)$	$w_4(1:15)$	$w_5(1:15)$
teoria	*flusso*	*strutture*	*mach*	*transizione*
fluttuare	*piastra*	*struttura*	*numero*	*ruvidezza*
aerodinamica	*magnetico*	*ipersonico*	*profilo alare*	*reynolds*
risultati	*campo*	*flusso*	*flusso*	*confine*
sollevamenti	*fluido*	*metodo*	*pressione*	*numero*
ala	*free*	*teoria*	*numeri*	*strato*
derivate	*convezione*	*shock*	*metodo*	*raffreddamento*
aeroplano	*trasversale*	*pressione*	*sezione*	*effetto*
effetti	*piatto*	*rivoluzione*	*alto*	*supersonico*
resistenza	*condurre*	*punta*	*naca*	*dimensionale*
velocità	*effetto*	*smussato*	*distribuzioni*	*superficie*
analisi	*elettricamente*	*sottile*	*caratteristiche*	*cilindro*
movimento	*viscoso*	*superficie*	*dati*	*numeri*
valori	*soluzioni*	*ordine*	*assalto*	*sfrecciare*
rapidità	*verticale*	*risultati*	*angolo*	*turbolento*

Tabella 4.1: Parole chiave dei primi 5 vettori di base ottenuti con l'algoritmo AHCLS.

Osservando la matrice dei coefficienti H notiamo che isolando le colonne $h_{70}, h_{149}, h_{202}, h_{407}$ relative al medesimo documento e considerando su esse solo le prime 5 componenti (relative ai primi 5 vettori di base w_j) otteniamo la matrice:

$$H(1:5, I) = \begin{bmatrix} 0.0969 & 0.6713 & 3.9504 & 0 \\ 0 & 0.5500 & 0 & 2.2186 \\ 0 & 0.6282 & 0 & 0 \\ 6.3700 & 0.6970 & 0 & 0.0777 \\ 0 & 0.4930 & 0 & 0 \end{bmatrix} \quad \text{dove } I = \{70, 149, 202, 407\}.$$

Le componenti restanti delle colonne di H sono nulle o estremamente piccole, per cui sono state omesse.
Si deduce che il documento 407 tratta maggiormente la tematica *"Fenomeni ondulatori dei fluidi"* relativa al vettore di base w_2, il documento 202 si concentra sull' *"Aerodinamica"* relativa a w_1 mentre il documento 149 è rappresentato da una colonna poco sparsa e ciò si traduce nella presenza simultanea delle tematiche relative a $w_{1,\dots,5}$.

Confrontiamo questi risultati con quelli ottenuti dall'algoritmo ANLS di tipo active set:

in questo caso troviamo un residuo relativo finale $r_{fin} = 0.8126$ quindi leggermente minore e una sparsità media delle colonne di H=0.71 che come previsto è inferiore. Da sottolineare il fatto che, sebbene il numero di iterazioni sia inferiore rispetto all'algoritmo AHCLS, il tempo di esecuzione è decisamente superiore (vedi Tabella 4.2) e questa marcata differenza è dovuta all'esattezza della risoluzione del problema NNLS (3.5) da parte degli algoritmi ANLS. Inoltre, come riportato nel paragrafo 3.2.2, la scelta del metodo ANLS in questo esperimento non è appropriata a causa della sparsità della matrice ***A_cranrid***.

	N° iterazioni	Tempo d'esecuzione
ANLS	48	302.10 s
AHCLS	371	7.69 s

Tabella 4.2: Confronto della convergenza dei 2 algoritmi.

Utilizzando lo stesso metodo di AHCLS riportiamo nella Tabella 4.3 i 15 termini dominanti relativi ai primi 5 vettori di base, che indicheremo con $\{\hat{w}_j\}_{j=1}^{5}$, dopo aver ordinato le componenti in ordine decrescente di importanza.

$\hat{w}_1(1:15)$	$\hat{w}_2(1:15)$	$\hat{w}_3(1:15)$	$\hat{w}_4(1:15)$	$\hat{w}_5(1:15)$
teoria	*magnetico*	*supersonico*	*velocità*	*flusso*
fluttuare	*campo*	*alto*	*soluzioni*	*gas*
aerodinamica	*piastra*	*velocità (plur.)*	*equazione*	*scorso*
derivate	*free*	*mach*	*metodo*	*non viscoso*
risultati	*fluido*	*velocità (sing.)*	*equazioni*	*viscoso*
valori	*convezione*	*superficie*	*soluzioni*	*bordo*
aeroplano	*condurre*	*subsonico*	*distribuzione*	*principale*
teorico	*trasversale*	*metodo*	*problema*	*corrente*
effetti	*elettricamente*	*aria*	*temperatura*	*condizioni*
ala	*piatto*	*dimensionale*	*laminare*	*dimensionale*
parte	*verticale*	*design*	*fluido*	*piatto*
alettone	*presenza*	*risultato*	*ottenuto*	*incomprimibile*
sperimentale	*effetto*	*assiale*	*approssimativo*	*forze di taglio*
profilo alare	*soluzioni*	*flussi*	*movimento*	*semplice*
pistone	*lastre*	*compressore*	*comprimibile*	*simmetrico*

Tabella 4.3: Parole chiave dei primi 5 vettori di base ottenuti con l'algoritmo ANLS.

Osserviamo che il vettore di base $\hat{w}_1$ rappresenta grossomodo la stessa tematica presentata dal vettore w_1. Anche il vettore $\hat{w}_2$, che possiamo etichettare con la tematica "*Elettromagnetismo*" ha molti elementi comuni con w_2, seppur non contenga la parola chiave *"viscoso"* che compare invece nel vettore $\hat{w}_5$ insieme a termini che possiamo raggruppare nell'argomento *"Proprietà dei fluidi"*. Notiamo inoltre che la tematica relativa a $\hat{w}_4$ riguarda la risoluzione di equazioni, quindi un ambito matematico.

Passando invece allo studio dei coefficienti di H, consideriamo stavolta le colonne h_{149},h_{202},h_{407},h_{498}, per cui la porzione interessata è data da:

$$H(1:5,I) = \begin{bmatrix} 0.4794 & 6.7377 & 0 & 0.2340 \\ 0 & 0 & 2.1995 & 0 \\ 0 & 0 & 0 & 0 \\ 1.7525 & 0 & 0 & 4.0002 \\ 3.2577 & 0 & 2.3847 & 0 \end{bmatrix} \quad \text{dove } I = \{149, 202, 407, 498\}.$$

Osserviamo che, come nell'esperimento precedente, la colonna 202 ha valore significativo solo nella prima componente, ovvero riferito a $\hat{w}_1 \approx w_1$. Per la colonna 407, invece, notiamo che la differenza dei vettori base w_2 e $\hat{w}_2$ determini un differente peso delle tematiche all'interno del relativo documento, in particolare possiamo dedurre che il testo presenta parti legate all' *"Elettromagnetismo"*($\hat{w}_2$) e alle *"Proprietà dei fluidi"*($\hat{w}_5$). Notare, infine, come la colonna 149 continui ad essere debolmente sparsa, viceversa la 498 suggerisce un collegamento forte tra il medesimo documento e la risoluzione di equazioni.

In conclusione non possiamo stabilire quale dei 2 algoritmi sia più adatto per questo tipo di problema poiché non è confrontabile l'importanza di un residuo basso rispetto ad una elevata sparsità, tuttavia è emerso come combinando gli esperimenti si possa giungere a risultati più evidenti.

4.2 Riconoscimento Facciale

La seconda applicazione della NMF che consideriamo è l'elaborazione digitale delle immagini, in particolare l'estrazione di caratteristiche dai volti umani.

Data una collezione di n immagini facciali a livelli di grigio, che vengono convertite in

matrici di dimensione $b \times h$ (dove $b \cdot h = m$ è il numero totale di pixel), la matrice dei dati $X \in \mathbb{R}^{m \times n}$ è strutturata in modo che le colonne $X(:, j)$ rappresentino le immagini (intese come matrici) vettorizzate. Di conseguenza l'elemento (i, j) di X indica l'intensità di grigio dell'i-esimo pixel rispetto alla j-esima faccia. Per cui, fissato il rango di approssimazione k, si ha $\forall\ 1 \leq j \leq n$:

$$\underbrace{X(:,j)}_{\substack{j\text{-esima immagine} \\ \text{facciale}}} \approx \sum_{i=1}^{k} \underbrace{W(:,i)}_{\substack{i\text{-esima caratteristica} \\ \text{facciale}}} \underbrace{H(i,j)}_{\substack{\text{importanza della } i\text{-esima} \\ \text{caratteristica nella } j\text{-esima immagine}}} .$$

La non negatività di W ci permette di interpretare le sue colonne, ovvero i vettori di base, come delle immagini. I coefficienti in H, essendo non negativi, possono solo sommare i vettori base per ricomporre l'immagine originale.
Solitamente nelle applicazioni i dataset contengono numerose collezioni di immagini e quindi utilizzando un rango di approssimazione basso, la NMF genererà i vettori di base corrispondenti a caratteristiche comuni a molte immagini. La speranza è che la fattorizzazione riesca ad individuare elementi localizzati quali occhi, bocca, etc; in tal caso le colonne di W saranno molto sparse. Per i nostri esperimenti utilizzeremo il database di facce pre-processate *CBCL*[2] composto da un train set di 2429 immagini a livelli di grigio di dimensione 19×19 pixel (quindi piccole miniature, vedi Figura 4.2).

Figura 4.2: Esempi di facce del database CBCL

Di conseguenza la matrice dei dati X ha dimensione 361×2429, dove $361 = 19 \cdot 19$, e ogni elemento (i, j) rappresenta l'intensità del grigio dell'i-esimo pixel nella j-esima immagine espressa attraverso un numero naturale compreso tra 0 (nero) e 256 (bianco). Poiché il

[2]Il dataset è accessibile da WikiData attraverso l'url `http://www.ai.mit.edu/courses/6.899/lectures/faces.tar.gz`.

nostro interesse è evidenziare con zone scure le caratteristiche facciali isolate associando la sparsità ad immagini prevalentemente bianche, invertiamo la scala dei grigi ponendo $(X)_{ij} = 256 - (X)_{ij} \quad \forall i, j$.
Come verificato da risultati precedenti su questo dataset, una buona scelta per il rango di approssimazione è $k = 49$, che indica il numero di caratteristiche facciali che ricerchiamo. Effettuiamo le prove solo con l'algoritmo ANLS che grazie alla discreta densità della matrice X genera risultati di gran lunga più accurati in termini di residuo relativo rispetto agli altri metodi ($r_{rel}^{ANLS} = 0.08$). Sapendo che le colonne di W rappresentano delle immagini e che i fattori W, H della NMF <u>non</u> sono unici, è utile bilanciare gli elementi di W in modo che appartengano all'intervallo $[0, 1]$, da cui poi è possibile ricavare il livello di grigio moltiplicando per 256. Questo passaggio si effettua ponendo come nuove soluzioni $(W' = WD^{-1}, H' = DH)$ dove $D = diag(\max(W)) \in \mathbb{R}^{k \times k}$, ovvero la matrice diagonale in cui l'elemento diagonale i-esimo è il valore massimo presente nella colonna i-esima di W.
Trasformando in immagini le colonne di W ottenute con questa procedura, facendo attenzione a re-invertire il livello di grigio, otteniamo i risultati raffigurati in Figura 4.3.

Figura 4.3: I 49 vettori di base riportati in uno schema 7×7

Osserviamo che, come previsto, le immagini sono prevalentemente bianche e presentano

zone scure localizzate su alcune caratteristiche facciali e sulle zone d'ombra presenti negli angoli al bordo delle immagini. Analizziamo ora più nello specifico come le immagini di partenza vengono decomposte nei vettori di base attraverso un esempio.
Consideriamo l'immagine corrispondente al vettore colonna x_{2230} della matrice dei dati. La sua approssimazione Wh_{2230} è data da:

Figura 4.4: Immagine 2230 (a sx) e sua approssimazione (a dx) con rango k=49

Gli indici delle componenti di modulo maggiore del vettore colonna h_{2230} sono relativi ai vettori di base dominanti nell'immagine. Nell'esempio in questione i primi 6 vettori di base sono raffigurati in Figura 4.5.

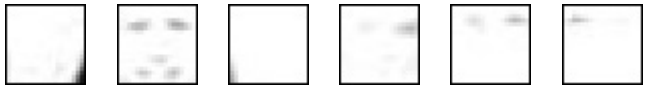


Figura 4.5: Vettori di basi dominanti nell'immagine 2230

Diversamente dalla matrice W, la sparsità di H non è necessaria per i nostri fini, osserviamo però che a causa della scelta iniziale sulla dominanza del bianco nei vettori di base, le immagini più scure avranno le relative componenti in H mediamente più grandi e saranno meno sparse rispetto alle immagini più luminose. Denotando con $\hat{h}_j$ il vettore in cui le componenti di h_j sono ordinate in modo decrescente, osserviamo in Figura 4.6 il confronto tra $\hat{h}_{2230}$ (x_{2230} rappresenta un'immagine chiara) e $\hat{h}_{460}$, (x_{460} rappresenta un'immagine scura) per le prime 30 componenti.

Nell'ambito del riconoscimento facciale possiamo perciò concludere che la NMF ha buone potenzialità poiché suddividendo i volti in base a caratteristiche isolate permette di associare la stessa persona ad immagini in cui è presente una modifica localizzata (ad esempio a causa di occhiali, barba, etc). Tuttavia rispetto ad approcci più "olistici" come PCA (*Principal component analyses*) e VQ (*Vector quantization*) la NMF è sensibile ai differenti punti di vista dell'immagine dovute ad una traslazione o ad una rotazione del volto.

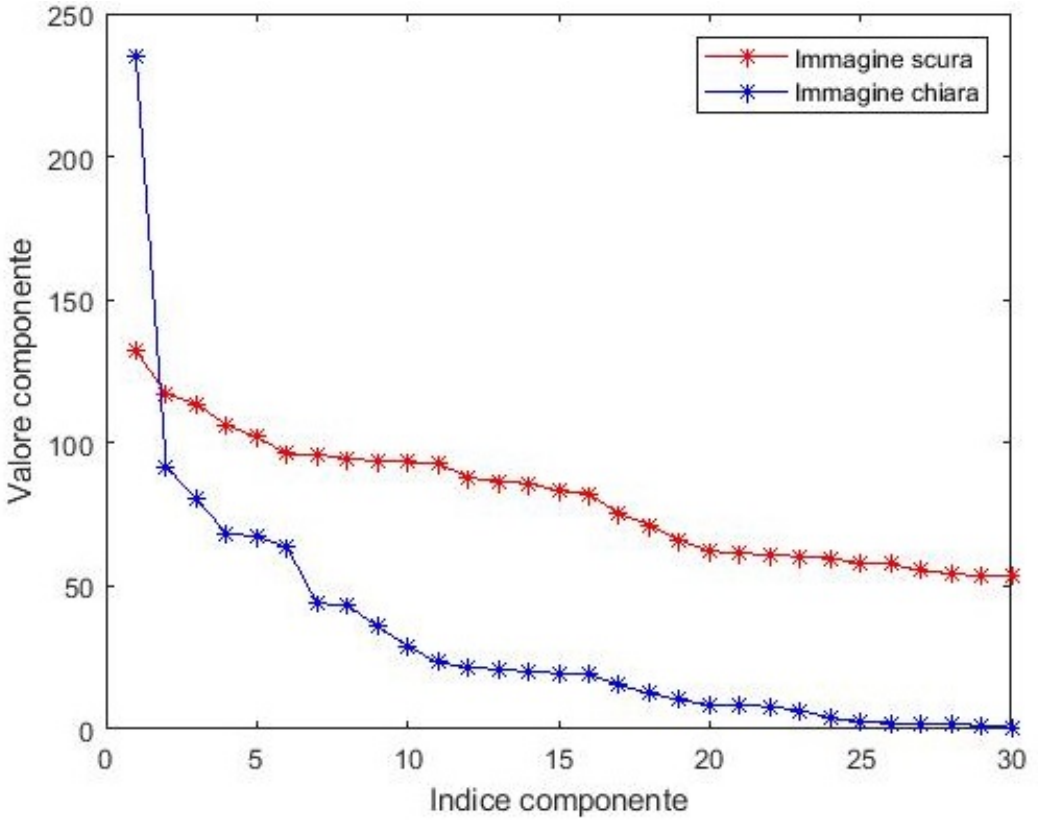


Figura 4.6: Confronto decrescita componenti di $\hat{h}$

Capitolo 5

La variante ortogonale ONMF

Introduciamo una variante del problema originale NMF che aggiunge il vincolo di **ortogonalità** alle righe della matrice dei coefficienti H. Il nuovo problema, detto ONMF (*Orthogonal Nonnegative Matrix Factorization*) si presenta in questo modo:
Data una matrice $X \in \mathbb{R}_{+}^{m \times n}$ e un rango di fattorizzazione k (con $k < n$) risolvere:

$$\min_{W \in \mathbb{R}^{m \times k}, H \in \mathbb{R}^{k \times n}} \|X - WH\|_F^2 , \tag{5.1a}$$

dove

$$W \geq 0, H \geq 0 \ ; \tag{5.1b}$$

$$HH^T = I_k. \tag{5.1c}$$

Effettuiamo un paio di osservazioni:

Osservazione 5.0.1. La condizione (5.1c) unita alla precedente (5.1b) implica che le colonne di H contengono al più un elemento non zero e perciò ogni vettore $\{x_j\}_{j=1}^{n}$ della matrice dei dati si potrà approssimare attraverso un unico vettore di base che denoteremo come w_{i_j}.

Osservazione 5.0.2. Denotando con i_j l'elemento non zero della colonna h_j otteniamo:

$$\begin{aligned}
\min_{W,H\geq 0} \|X - WH\|_F^2 &= \min_{W,h_j\geq 0} \sum_{j=1}^{n} \|x_j - Wh_j\|_2^2 = \min_{w_{i_j},h_{i_j,j}\geq 0} \sum_{j=1}^{n} \left\|x_j - w_{i_j}h_{i_j,j}\right\|_2^2 = \\
&= \min_{w_{i_j},h_{i_j,j}\geq 0} \sum_{j=1}^{n} \left[\, \|x_j\|_2^2 + h_{i_j,j}^2 \left\|w_{i_j}\right\|_2^2 - 2h_{i_j,j} \left\langle x_j, w_{i_j} \right\rangle \right] = \\
&= cost + \min_{w_{i_j},h_{i_j,j}\geq 0} \sum_{j=1}^{n} \Big[\, h_{i_j,j} \left\|w_{i_j}\right\|_2 \Big(\, h_{i_j,j} \left\|w_{i_j}\right\|_2 - \\
&\quad - 2 \|x_j\|_2 \underbrace{\left\langle \frac{x_j}{\|x_j\|_2}, \frac{w_{i_j}}{\left\|w_{i_j}\right\|_2} \right\rangle}_{=:\, \cos\alpha_j(w_{i_j})} \, \Big) \, \Big] = \\
&= cost + \min_{w_{i_j},h_{i_j,j}\geq 0} \sum_{j=1}^{n} \beta_j(w_{i_j}, h_{i_j,j}) \ ,
\end{aligned} \tag{5.2}$$

dove $\alpha_j(v)$ è l'angolo tra x_j e un generico vettore $v \in \mathbb{R}^m$ mentre β_j indica l'espressione contenuta nella sommatoria. Osserviamo che $\cos\alpha_j(w_{i_j}) \geq 0 \ \forall j$ poiché $X, W \geq 0$.

Vogliamo dimostrare che facendo il minimo di (5.2) rispetto ad H e W otteniamo che $\forall j$ la soluzione w_{i_j} è il vettore base che forma l'**angolo minore** con x_j.
Quindi fissato j poniamo per assurdo che $\exists w \neq w_{i_j}$ tale che risolva il minimo e proviamo che $\forall h \geq 0 \ \exists h_{i_j}$ tale che $\beta_j(w,h) \geq \beta_j(w_{i_j}, h_{i_j,j})$.
Si ha per ipotesi che $\cos\alpha_j(w) \leq \cos\alpha_j(w_{i_j})$, quindi preso un qualsiasi $h \geq 0$ possiamo porre $h_{i_j,j} = h\frac{\|w\|_2}{\left\|w_{i_j}\right\|_2}$ che quindi è ≥ 0 e otteniamo:

$$\begin{aligned}
\beta_j(w,h) &= h\|w\|_2 \left(\, h\|w\|_2 - 2\|x_j\|_2 \, \cos\alpha_j(w)\right) \\
&= h_{i_j,j}\frac{\left\|w_{i_j}\right\|_2}{\cancel{\|w\|_2}}\cancel{\|w\|_2} \left(\, h_{i_j,j}\frac{\left\|w_{i_j}\right\|_2}{\cancel{\|w\|_2}}\cancel{\|w\|_2} - 2\|x_j\|_2 \, \cos\alpha_j(w)\right) \\
&\geq h_{i_j,j}\left\|w_{i_j}\right\|_2 \left(\, h_{i_j,j}\left\|w_{i_j}\right\|_2 - 2\|x_j\|_2 \, \cos\alpha_j(w_{i_j})\right) = \beta_j(w_{i_j}, h_{i_j,j}).
\end{aligned}$$

Possiamo estendere lo stesso procedimento ad ogni $j = 1, ..., n$ per cui l'osservazione è provata.

5.1 Metodi di clustering

Introduciamo in questa sezione uno degli strumenti più utilizzati nel Data Mining, ovvero il *clustering*, che consentirà di inserire il problema ONMF all'interno di un quadro più ampio, come vedremo nell'Esempio 5.1.3.

Definizione 5.1 (Clustering partizionale)**.** Sia $\{x_1, x_2, \ldots, x_n\}$ un insieme di vettori di $\mathbb{R}^m$. Risolvere un problema di clustering partizionale significa trovare una k-partizione $\{\pi_i\}_{i=1}^k$, ovvero:

1. $\emptyset \neq \pi_i \subseteq \{1, 2, \ldots, n\}$;
2. $\bigcup_{1 \leq i \leq k} \pi_i = \{1, 2, \ldots, n\}$;
3. $\pi_i \cap \pi_j = \emptyset \ \forall i \neq j$,

tale che l'insieme dei vettori $\big\{ \{x_j\}_{j \in \pi_i} \mid 1 \leq i \leq k \text{ fissato} \big\}$, detto **cluster**, contenga elementi "simili" secondo qualche criterio quantitativo.

NOTA: d'ora in avanti con il termine *clustering* intenderemo il clustering partizionale appena definito.

Esempio 5.1.1 (k-medie)**.** Uno dei metodi di *clustering* più noti è il metodo delle ***k*-medie**. Scegliendo la distanza Euclidea come criterio di similarità, risolvere il problema delle k-medie equivale matematicamente a trovare:

$$\min_{\{\pi_i, c_i\}_{i=1}^k} \sum_{i=1}^{k} \sum_{j \in \pi_i} \|x_j - c_i\|^2 , \tag{5.3}$$

dove $c_i := \frac{\sum_{j \in \pi_i} x_j}{|\pi_i|}$ sono detti i **centroidi** del cluster.

Esempio 5.1.2 (k-medie sferiche)**.** Il metodo delle ***k*-medie sferiche** è una variante del metodo delle k-medie dove sia i centroidi che i vettori iniziali $\{x_j\}_{j=1}^n$ hanno norma unitaria. Per cui risolvere il problema delle k-medie sferiche equivale matematicamente a trovare:

$$\min_{\{\pi_i, c_i\}_{i=1}^k} \sum_{i=1}^{k} \sum_{j \in \pi_i} \left\| \frac{x_j}{\|x_j\|} - c_i \right\|^2 , \tag{5.4}$$

dove $c_i := \frac{\sum_{j\in\pi_i} x_j}{\left\|\sum_{j\in\pi_i} x_j\right\|}$ sono i centroidi normalizzati del cluster.
Se inoltre $x_j \in \mathbb{R}^m_+ \ \forall j$ allora (5.4) diventa:

$$\max_{\{\pi_i,c_i\}_{i=1}^k} \sum_{i=1}^k \sum_{j\in\pi_i} \frac{x_j^T}{\|x_j\|} c_i = \max_{\{\pi_i,c_i\}_{i=1}^k} \sum_{i=1}^k \sum_{j\in\pi_i} \cos(\alpha_{i,j}) \ , \tag{5.5}$$

dove $\alpha_{i,j}$ è l'angolo tra x_j e c_i.

Osservazione 5.1.1. È importante sottolineare che in entrambi gli esempi precedenti, gli algoritmi per risolvere (5.3) e (5.4) non garantiscono la convergenza al minimo globale. Infatti, questi metodi si basano su una scelta iniziale casuale di un set di cluster e proseguono attraverso modifiche progressive degli stessi secondo criteri di ottimalità fino alla convergenza. Di conseguenza diverse inizializzazioni possono portare a diversi risultati finali, che corrispondono ai minimi locali della funzione obiettivo.

Esempio 5.1.3. Considerando l'insieme dei vettori dato dalle colonne $\{x_j\}_{j=1}^n \in \mathbb{R}^m_+$ della matrice dei dati X, possiamo interpretare il problema ONMF (5.1) come un metodo di *clustering* ponendo la condizione:

$$x_j \in \pi_i \Leftrightarrow h_j(i) \neq 0.$$

Abbiamo visto nell'Osservazione 5.0.2 che questo equivale a raggruppare i vettori $\{x_j\}_{j=1}^n$ in k cluster a seconda del vettore di base $\{w_i\}_{i=1}^k$ con cui l'angolo formato è minore.

5.2 Teorema di Equivalenza

Teorema 5.2. *Per una matrice dei dati $X \in \mathbb{R}^{m\times n}_+$ risolvere il problema ONMF* (5.1) *equivale a risolvere la seguente variante pesata delle k-medie sferiche:*

$$\max_{\{\pi_i,w_i\in\mathbb{R}^m_+\}_{i=1}^k} \sum_{i=1}^k \sum_{j\in\pi_i} \|x_j\|^2 \Big(\underbrace{\frac{x_j^T}{\|x_j\|_2}\frac{w_i}{\|w_i\|_2}}_{=:\ \cos(\alpha_{i,j})}\Big)^2, \tag{5.6}$$

dove $\{\pi_i\}_{i=1}^k$ è la k-partizione che identifica i cluster e $\alpha_{i,j}$ è l'angolo tra x_j e c_i.

Osservazione 5.2.1. Il problema (5.6) differisce da (5.5) per la presenza del "peso" $\|x_j\|^2$ e del quadrato del coseno all'interno della sommatoria. Notiamo inoltre che i centroidi in (5.6) sono rappresentati dalle colonne normalizzate di W.

Dimostrazione del Teorema 5.2. Supponendo che le colonne di W abbiano almeno una componente non nulla, definiamo:

$$\Phi : (W, H) \to (\hat{W} = WD^{-1}, \hat{H} = DH) \quad \text{dove } D = diag(\|w_1\|, \|w_2\|, ..., \|w_k\|).$$

Se (W, H) è soluzione di (5.1) allora lo è anche $\Phi(W, H)$ e viceversa. Per cui il problema (5.1) diventa:

$$\min_{\hat{W}\geq 0, \hat{H}\geq 0} \left\| X - \hat{W}\hat{H} \right\|_F^2 \quad \text{con } \left(\hat{H}\hat{H}^T\right)_{i,j} = 0 \ \ \forall i \neq j, \ \|\hat{w}_i\| = 1 \ \forall i. \tag{5.7}$$

Data una partizione $\pi = \{\pi_i\}_{i=1}^k$ denotiamo con $\sim$ la relazione t.c. $\hat{H} \sim \pi \ \Leftrightarrow \ \left(H_{i,j} \neq 0 \Rightarrow j \in \pi_i\right)$. Osserviamo che per $\hat{H} \geq 0$:

$$\exists \pi \text{ t.c. } \hat{H} \sim \pi \Longleftrightarrow (\hat{H}\hat{H}^T)_{i,j} = 0 \quad \forall i \neq j.$$

Per cui il problema (5.7) equivale a:

$$\min_{\{\hat{W},\hat{H},\pi\}} \sum_{i=1}^{k} \sum_{j\in\pi_i} \left\| \hat{x}_j - \hat{w}_i \hat{h}_{ij} \right\|^2 \quad \text{dove} \quad \begin{cases} \hat{W} \geq 0 \ \ \|\hat{w}_i\| = 1 \ \forall i; \\ \hat{H} \geq 0; \\ \pi \text{ deve soddisfare } \hat{H} \sim \pi. \end{cases} \tag{5.8}$$

Minimizzando rispetto ad $\hat{H}$ (ricordando che $\hat{H} \sim \pi$, $X, \hat{H} \geq 0$) otteniamo $\forall\ 1 \leq i \leq k$:

$$\begin{cases} \hat{h}^*_{ij} = 0 \quad \ se\ j \notin \pi_i \\ \hat{h}^*_{ij} = \text{argmin}_{\beta\geq 0} \|x_j - \hat{w}_i\beta\|^2 = \text{argmin}_{\beta\geq 0}(x_j^T x_j - 2\beta x_j^T \hat{w}_i + \beta^2) = x_j^T \hat{w}_i \quad \ se\ j \in \pi_i. \end{cases}$$

Sostituendo $\hat{h}^*_{ij}$ nel problema (5.8) otteniamo:

$$\min_{\{\hat{W},\pi\}} \sum_{i=1}^{k} \sum_{j\in\pi_i} \left\| x_j - \left(x_j^T \hat{w}_i\right) \hat{w}_i \right\|^2 = \min_{\{\hat{W},\pi\}} \sum_{i=1}^{k} \sum_{j\in\pi_i} - \left(x_j^T \hat{w}_i\right)^2 + costante.$$

Minimizzare rispetto a $\hat{W}$ e π significa quindi trovare:

$$\max_{\{\hat{W},\pi\}} \sum_{i=1}^{k} \sum_{j\in\pi_i} \left(x_j^T w_i\right)^2 \text{ con } \hat{W} \geq 0 \ \|\hat{w}_i\| = 1 \ \forall i \ \Leftrightarrow \max_{\{\pi_i, w_i \in \mathbb{R}_+^m\}_{i=1}^k} \sum_{i=1}^{k} \sum_{j\in\pi_i} \|x_j\|^2 \left(\frac{x_j^T}{\|x_j\|_2} \frac{w_i}{\|w_i\|_2} \right)^2 .$$

□

5.3 Algoritmo di tipo Expectation-Maximization (EM)

Sfruttando il Teorema 5.2 costruiamo un algoritmo per la ONMF che alterna i seguenti passaggi:

1. Dati $\{w_i\}_{i=1}^k$ centroidi, si trova la partizione $\{\pi_i\}_{i=1}^k$ che assegna ai punti il cluster più "vicino" secondo:

$$j \in \pi_i \Rightarrow i \in \underset{1\leq l\leq k}{\operatorname{argmax}} \left(x_j^T w_l\right)^2 = \underset{1\leq l\leq k}{\operatorname{argmax}} \left(x_j^T w_l\right).$$

2. Data la partizione $\{\pi_i\}_{i=1}^k$ si trovano i nuovi centroidi in questo modo:
   sia $X_i \in \mathbb{R}^{m\times|\pi_i|}$ la sottomatrice di X che contiene le colonne $x_j \in \pi_i$.
   Risolviamo il seguente problema:

$$\max_{\{w_i\geq 0, \|w_i\|_2=1\}_{i=1}^k} \sum_{i=1}^{k} \sum_{j\in\pi_i} (x_j^T w_i)^2 = \max_{\{w_i\geq 0, \|w_i\|_2=1\}_{i=1}^k} \sum_{i=1}^{k} \left\|X_i^T w_i\right\|_2^2 = \sum_{i=1}^{k} \sigma_1\left(X_i\right)^2,$$

   dove $\sigma_1(X_i)$ è il **valore singolare dominante di** X_i.
   Consideriamo i relativi **vettori singolari sinistri** w_i^*, che risolvono

$$\underset{\{\|w_i\|_2=1\}_{i=1}^k}{\operatorname{argmax}} \left\|X_i^T w_i\right\|_2^2$$

   come nuovi centroidi, infatti per il teorema di Perron-Frobenius sono non negativi.

Osservazione 5.3.1. Per inizializzare la matrice dei centroidi $W = [w_1, w_2, \ldots, w_k]$ si possono utilizzare le strategie descritte nella sezione 3.4.

Osservazione 5.3.2. Sia $\hat{W} = [\hat{w_1}, \hat{w_2}, \ldots, \hat{w_k}]$ la matrice con le colonne normalizzate tale che $(\hat{W}, \hat{H})$ risolva (5.1) con $\hat{H} = (\hat{h})_{ij}$. Definendo $\hat{v}_i := (\hat{h}_{ij})_{j \in \pi_i}$ vale l'equivalenza:

$$\left\| X - \hat{W}\hat{H} \right\|_F^2 = \sum_{i=1}^{k} \sum_{j \in \pi_i} \left\| x_j - \hat{w}_i \hat{h}_{ij} \right\|^2 = \sum_{i=1}^{k} \left\| X_i - \hat{w}_i \hat{v}_i^T \right\|_F^2 ,$$

da cui il problema originale ONMF diventa:

$$\min_{\|\hat{w}_i\|=1, \hat{w}_i \geq 0, \hat{v}_i \geq 0} \sum_{i=1}^{k} \left\| X_i - \hat{w}_i \hat{v}_i^T \right\|_F^2 .$$

Si tratta perciò di problemi in cui trovare la migliore approssimazione di rango 1. Prendendo il primo fattore di rango 1 dalla **SVD**, che è non negativo per il teorema di Perron-Frobenius, otteniamo il minimo che corrisponde a:

$$\sum_{i=1}^{k} \left(\|X_i\|_F^2 - \sigma_1^2(X_i) \right) = \|X\|_F^2 - \sum_{i=1}^{k} \sigma_1^2 (X_i) .$$

Per i risultati precedenti, la ONMF equivale quindi a trovare una partizione $\{\pi_i\}_{i=1}^k$ tale che valga:

$$\max_{\{\pi_i\}_{i=1}^k} \sum_{i=1}^{k} \sigma_1^2 (X_i) .$$

5.4 Algoritmo di ottimizzazione vincolata (ONP-MF)

Costruiamo un secondo algoritmo per risolvere la ONMF imponendo ad ogni iterazione $W \geq 0, HH^T = I$, e attraverso l'uso della Lagrangiana vogliamo ottenere anche $H \geq 0$.

Definiamo la Lagrangiana del problema:

$$L_\rho(W, H, \Lambda) = \frac{1}{2} \|X - WH\|_F^2 + \langle \Lambda, -H \rangle + \frac{\rho}{2} \|\min(H, 0)\|_F^2 ,$$

dove ρ è il parametro di penalizzazione, $\Lambda \in \mathbb{R}_+^{k \times n}$ è la matrice dei moltiplicatori di Lagrange e $\langle \ldots \rangle$ è il prodotto scalare in $\mathbb{R}^{k \times n}$.

Le soluzioni W, H del problema ONMF originario sono soluzioni di:

$$\min_{W\geq 0, HH^T=I_k} \max_{\Lambda\geq 0} L_\rho(W, H, \Lambda).$$

La struttura dell'algoritmo è uno schema alternato per aggiornare W, H, Λ ad ogni iterazione seguendo i passaggi:

1. Per H, Λ fissati, troviamo W ottimale risolvendo un problema NNLS (3.5) con un metodo opportuno:
$$W \longleftarrow \operatorname*{argmin}_{Z\in\mathbb{R}_+^{m\times k}} \|X - ZH\|_F^2 .$$

2. Per W, Λ fissati, vogliamo che $HH^T = I_k$, per cui definiamo una proiezione: $Proj_{St}(\widehat{H}) = \operatorname{argmin}_Z \left\|\widehat{H} - Z\right\|_F^2$ con $ZZ^T = I_k$ sullo spazio delle matrici $\mathbb{R}^{k\times n}$ a righe ortogonali, ovvero con l'insieme delle righe appartenente alla varietà di Stiefel $V_k(\mathbb{R}^n)$.
Ora sfruttando il metodo di discesa del gradiente:
$$H \longleftarrow Proj_{St}\big(H - \beta\nabla_H L_\rho(W, H, \Lambda)\big), \quad \beta := \textit{passo di discesa.}$$

3. Per W, H fissati, troviamo Λ in modo da penalizzare i valori negativi di H attraverso una salita del gradiente:
$$\Lambda \longleftarrow \max(0, \Lambda - \alpha, H), \quad \alpha := \textit{passo di salita.}$$

Osservazione 5.4.1. Poniamo $\Lambda^{(0)} = \mathbf{0} \in \mathbb{R}^{k\times n}$ mentre, non essendoci il vincolo di non negatività su H, possiamo inizializzare la stessa con i primi k vettori singolari destri di X. Questo ci permette fin da subito di avere una buona approssimazione di X.

5.5 Applicazione alla separazione iperspettrale

La firma spettrale di un pixel è la frazione di luce incidente che viene riflessa dal pixel per diverse lunghezza d'onda ed è perciò non negativa. Nelle immagini iperspettrali ogni pixel non possiede quindi un singolo valore cromatico, ma bensì una serie di valori

dati dalla firma spettrale ottenuta attraverso bande di 100-200 lunghezze d'onda anche al di fuori dello spettro di luce visibile. Un'immagine iperspettrale può dunque essere considerata come un cubo di dati, costituito da tanti piani quante sono le bande che compongono lo spettro, e di larghezza e altezza pari alle dimensioni dell'area catturata (vedi Figura 5.1).

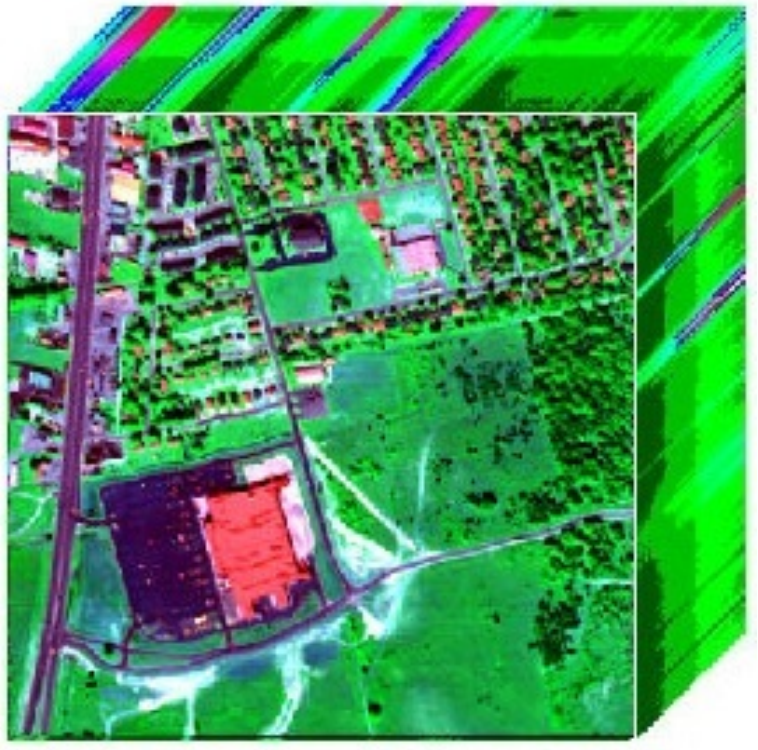


Figura 5.1: Immagine iperspettrale di un paesaggio urbano.

Vettorizzando le immagini si riesce ad ottenere una matrice dei dati $X \in \mathbb{R}^{m \times n}$ dove m è il numero di pixel, n è il numero di bande spettrali e ogni colonna $\{x_j\}_{j=1}^{n}$ è la firma spettrale del j-esimo pixel. L'obiettivo della separazione iperspettrale è classificare i materiali costitutivi presenti nell'immagine in base alla differente firma spettrale. Nelle riprese aeree ad esempio i materiali principali sono erba, terra, asfalto, etc.

Fissato il rango di approssimazione k si ha $\forall\ 1 \leq j \leq n$:

$$\underbrace{X(:,j)}_{\substack{\text{firma spettrale} \\ \text{del } j\text{-esimo pixel}}} \approx \sum_{i=1}^{k} \underbrace{W(:,i)}_{\substack{\text{firma spettrale} \\ \text{del } i\text{-esimo materiale}}} \underbrace{H(i,j)}_{\substack{\text{presenza del } i\text{-esimo} \\ \text{materiale nel } j\text{-esimo pixel}}} .$$

Applichiamo diversi metodi di *clustering* per stabilire a quale tipo di materiale appartengono i diversi pixel dell'immagine. Utilizziamo la matrice dei dati $Urban$[1]$\in \mathbb{R}^{162\times 94249}$ che contiene informazioni spettrali di immagini 307×307 pixel di un paesaggio urbano per 162 bande. Scegliamo $k = 6$, ovvero identifichiamo i materiali principali: "asfalto", "erba", "albero", "tetto", "metallo", "terra". Gli algoritmi per la ONMF (ONP-MF e EM) restituiscono:

1. W: le cui colonne rappresentano una stima delle firme spettrali dei 6 materiali;
2. H: in cui gli elementi non nulli della riga i-esima rappresentano i pixel relativi all'i-esimo materiale.

Gli algoritmi delle k-medie e k-medie sferiche restituiscono invece:

1. C: matrice 162×6 le cui colonne rappresentano i 6 centroidi delle firme spettrali dei pixel;
2. idx: vettore di 94249 variabili le cui componenti indicano da 1 a 6 qual è il cluster (materiale) a cui ogni pixel appartiene.

I risultati che abbiamo ottenuto dai 4 metodi, convertendo in immagine a livello di grigio le 6 righe di H e i vettori di componente fissata di idx, sono visibili nelle ultime quattro righe in Figura 5.2. La prima riga mostra invece il risultato ottenuto da algoritmi di clustering molto sofisticati a cui sono stati apportati degli aggiustamenti manuali.[2]

[1] Le immagini di Army Geospatial Centre sono state organizzate nella matrice $Urban$ dai gestori del sito `http://lesun.weebly.com/hyperspectral-data-set.html`.

[2] Le immagini presenti nella prima riga della Figura 5.2 rappresentano le righe della matrice $end6\_groundTruth$, resa disponibile dai gestori del sito `http://lesun.weebly.com/hyperspectral-data-set.html`.

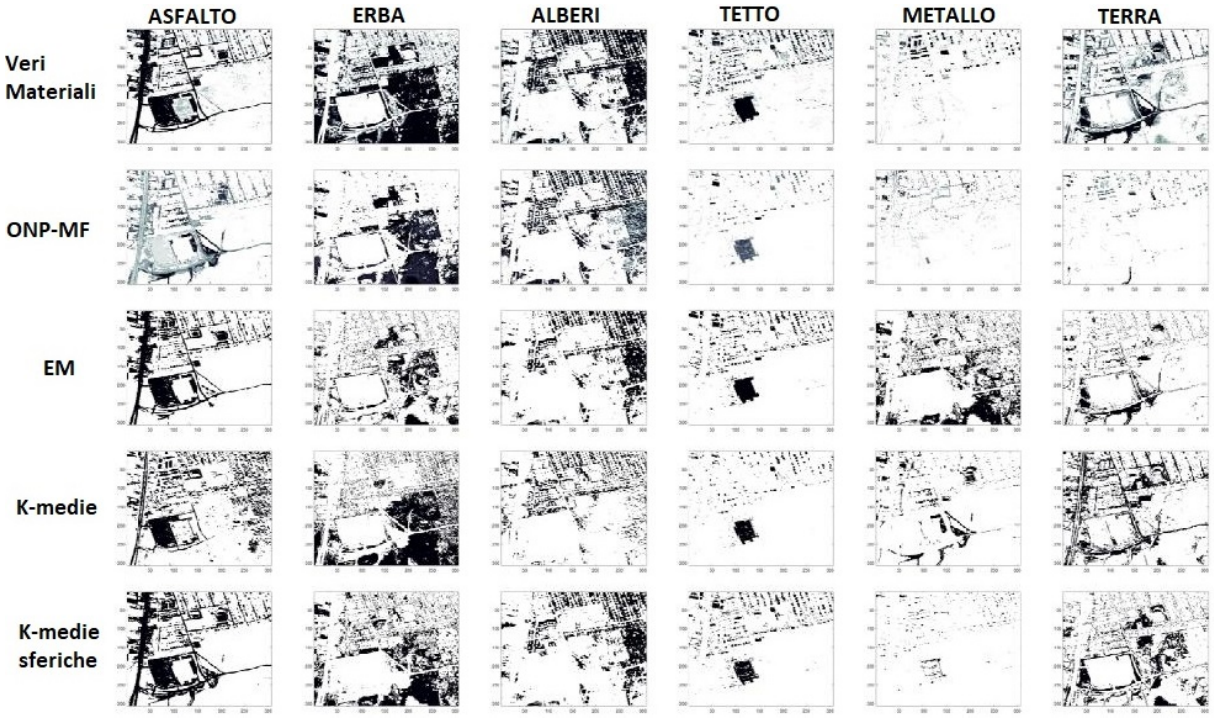


Figura 5.2: Confronto risultati clustering per lo studio di immagini iperspettrali.

Osserviamo che nel complesso l'algoritmo ONP-MF, a fronte di un maggior costo computazionale, classifica meglio i materiali; in particolare è l'unico algoritmo che riconosce il "metallo". Per gli altri 3 algoritmi la convergenza è nettamente più rapida ma i risultati sono abbastanza modesti. Questo fatto è in parte spiegabile dal diverso tipo di inizializzazione, infatti la SVD garantisce per l'ONP-MF minimi locali ottimali.

Capitolo 6

Conclusioni

In questo elaborato abbiamo studiato la classe di metodi NMF per la riduzione lineare di dimensione, consentendo una facile interpretazione dei risultati in molti ambiti in cui la non negatività ha un significato applicativo. Sono stati presentati una serie da algoritmi che differiscono per accuratezza delle soluzioni, costo computazionale e controllo della sparsità. La non convessità del problema ha condotto allo studio della convergenza e della dipendenza dalla strategia di inizializzazione. Nonostante i numerosi risultati sperimentali che ne caratterizzano l'efficacia, questi aspetti necessitano ancora di un approfondimento teorico che ne giustifichi l'applicabilità da un punto di vista matematico. Abbiamo visto che le condizioni di ottimalità delle soluzioni conferiscono alla NMF la proprietà di generare matrici sparse. Questa peculiarità permette, ad esempio, di isolare caratteristiche localizzate comuni all'interno di un dataset di immagini facciali. Infine è stata introdotta una variante del metodo, la ONMF, che abbiamo inserito e classificato all'interno dei metodi di clustering. La differenza con la NMF risiede nel dover approssimare ogni vettore iniziale attraverso un singolo vettore di base. Tale caratteristica ha indirizzato il suo utilizzo verso lo studio di immagini iperspettrali in cui l'obiettivo è associare ad ogni pixel il materiale principale che rappresenta.
Possiamo concludere che la moltitudine di ambiti di applicazione della NMF la rendono una tecnica molto versatile e particolarmente adatta all'estrazione di informazioni e all'interpretazione di grandi quantità di dati ("Big Data").

Bibliografia

[1] Russell Albright, James Cox, David Duling, Amy N. Langville, and Carl D. Meyer, *Algorithms, Initializations, and Convergence for the Nonnegative Matrix Factorization,* NCSU Technical Report Math 81706.

[2] Patrik Hoyer, *Non-negative Matrix Factorization with Sparseness Constraints,* Journal of Machine Learning Research 5, 1457–1469 (2004).

[3] Nicolas Gillis, *The Why and How of Nonnegative Matrix Factorization,* In: J. Suykens, M. Signoretto, A. Ar-gyriou (eds.) Regularization, Optimization, Kernels, and Support Vector Machines. Chapman and Hall/CRC,Machine Learning and Pattern Recognition Series (2014). To appear (arXiv:1401.5226).

[4] Daniel D. Lee, H. Sebastian Seung, *Learning the parts of objects by non-negative matrix factorization,* Nature 401, 788–791 (1999).

[5] Filippo Pompili, Nicolas Gillis, P.-A. Absil, Francois Glineur, *Two Algorithms for Orthogonal Nonnegative Matrix Factorization with Application to Clustering,* Neurocomputing, vol. 141, pp. 15–25, Oct. 2014.