

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI SCIENZE  
Corso di Laurea in Matematica

**DUE ALGORITMI PER LA  
FATTORIZZAZIONE MATRICIALE  
NON NEGATIVA CON APPLICAZIONE  
AL CLUSTERING**

Tesi di Laurea in Analisi Numerica

Relatore:  
Chiar.ma Prof.ssa  
VALERIA SIMONCINI

Presentata da:  
GABRIELE L.G.  
SCRIVANTI

Sessione Autunnale  
Anno Accademico 2016-2017

*A mamma e papà,  
Robi e Checco.  
Till mamma och pappa,  
Nadja och Georg.*



# Abstract

In questa tesi vengono descritti il problema della fattorizzazione non negativa ortogonale (ONMF) con applicazione al clustering e due algoritmi elaborati da F. Pompili, N. Gillis, P.A. Absil e F. Gilneur per l'approssimazione numerica della coppia di matrici soluzione di tale problema: il primo algoritmo legato a una variante delle  $k$ -medie sferiche e il secondo basato sul metodo della Lagrangiana aumentata. Particolare attenzione viene prestata alla base teorica su cui si fonda il primo algoritmo, cioè l'equivalenza tra il problema delle  $k$ -medie sferiche pesate e il problema ONMF descritta dal Teorema 2.6, di cui viene riportata la dimostrazione. Per ciascun algoritmo vengono analizzati punti di forza e di debolezza e suggerita la tipologia di data set per cui risultano più indicati al fine di determinare un'opportuna divisione in cluster.

Il primo capitolo, di carattere introduttivo, descrive i concetti di *clustering* e di fattorizzazione non negativa, proponendo una formulazione matematica utile ai fini della trattazione. Il secondo capitolo è dedicato all'algoritmo EM-ONMF, di cui è proposta la descrizione teorica e l'applicazione al problema di *text clustering* con oggetto una matrice termine-documento di articoli medici. Il terzo capitolo è dedicato all'algoritmo ONP-MF di cui sono descritti i metodi di ottimizzazione su cui è costruito, cioè il metodo del gradiente proiettato e della Lagrangiana aumentata, e gli esperimenti numerici sono applicati all'Iris data set contenuto nel file matlab `fisheriris`. Infine, nel quarto e ultimo capitolo vengono proposti due confronti numerici degli algo-

ritmi, che vengono analizzati in termini di iterazioni, tempi di elaborazione, stabilità e precisione della fattorizzazione e del clustering. Il primo confronto è applicato all'*hyperspectral unmixing* con oggetto il data set Hubble e il secondo è applicato al *pattern recognition* con oggetto U.S. Postal Service database.

Tutti gli esperimenti sono stati realizzati con un Intel<sup>®</sup> Core<sup>™</sup> i5 con 8 GB di RAM. I codici Matlab per i due algoritmi e il data set Hubble sono disponibili all'indirizzo <https://github.com/filippo-p/onmf>.

# Indice

<b>Abstract</b>	<b>i</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Clustering . . . . .	1
1.2 Fattorizzazione non negativa . . . . .	3
1.3 Fattorizzazione non negativa ortogonale . . . . .	4
<b>2 EM-ONMF</b>	<b>7</b>
2.1 Il metodo delle $k$ -medie . . . . .	7
2.2 Le $k$ -medie sferiche e ONMF . . . . .	9
2.3 Algoritmo EM-ONMF . . . . .	13
2.4 Esperimenti Numerici . . . . .	15
2.4.1 Un primo esempio . . . . .	15
2.4.2 <i>Document Clustering</i> . . . . .	17
<b>3 ONP-MF</b>	<b>21</b>
3.1 Metodo del Gradiente proiettato . . . . .	22
3.2 Metodo della Lagrangiana aumentata . . . . .	23
3.3 Struttura dell'algoritmo . . . . .	24
3.4 Esperimenti Numerici . . . . .	25
3.4.1 Gli iris di Fisher . . . . .	26
3.4.2 Medline . . . . .	28

<b>4</b>	<b>Confronti Numerici</b>	<b>31</b>
4.1	<i>Hyperspectral Unmixing</i> su Hubble data set . . . . .	31
4.2	<i>Handwritten digits</i> . . . . .	35
 <b>Conclusioni</b>		 <b>39</b>
 <b>A Minimi Quadrati Alternati</b>		 <b>41</b>
 <b>Bibliografia</b>		 <b>43</b>

# Elenco delle figure

1.1	Spy della matrice . . . . .	5
1.2	Mesh della matrice . . . . .	6
2.1	Cluster distinti . . . . .	16
2.2	Cluster allineati . . . . .	16
2.3	Ortogonalità della matrice $V$ : $VV^T = I_{50}$ . . . . .	17
2.4	Errori Relativi con EM-ONMF (sinistra) e con ALS (destra) .	18
3.1	ONP-MF - Aggiornamento ogni 100 iterazioni (colonna 1) dei valori: Parametro di Penalizzazione inizializzato $\rho = 100$ (colonna 2), Errore Relativo di Fattorizzazione $e_{rel} = \frac{\ M-UV\ _F}{\ M\ _F}$ (colonna 3), Residuo di Negatività della matrice $V$ $res_{neg} = \frac{\ \min\{V,0\}\ _F}{\ V\ _F}$ (colonna 4) . . . . .	26
3.2	Sinistra: <b>Residuo di negatività</b> . Destra: <b>Errore relativo</b> .	27
3.3	Sinistra: plot bidimensionale della matrice $V$ con in ascissa l'indice di colonna e in ordinata il valore dei tre elementi della colonna, differenziati per colore. Destra: plot tridimensionale della matrice $V$ . . . . .	28
3.4	Mesh della matrice $V$ . . . . .	29
3.5	Spy della matrice $V$ . . . . .	29
3.6	Mesh della matrice $VV^T$ . . . . .	30
3.7	Spy della matrice $VV^T$ . . . . .	30
4.1	Immagini campione dell'hubble data set a due diverse frequenze	32

## ELENCO DELLE FIGURE

---

4.2	Decomposizione in base ai materiali . . . . .	32
4.3	Clustering con EM-ONMF . . . . .	32
4.4	Clustering con ONP-MF . . . . .	33
4.5	Residuo Relativo con EM-ONMF (a) e ONP-MF (b) . . . . .	34
4.6	Cluster elaborati da altri algoritmi: $k$ -medie, $k$ -medie sferiche, CHNMF, CH(SVD),O-PNMF ,O-P(SVD). L'immagine compare in [1], dove sono presenti anche tutti i riferimenti agli articoli in cui vengono trattati CHNMF e O-PNMF . . . . .	34
4.7	Valori medi . . . . .	35
4.8	Coppie di centroidi elaborate dai tre algoritmi: $k$ -medie (a) e (b), EM-ONMF (c) e (d), ONP-MF (e) e (f) . . . . .	37

# Elenco delle tabelle

2.1	Esempio di clustering di una matrice termine-documento . . .	19
4.1	EM-ONMF . . . . .	33
4.2	Elapsed Time ONP-MF . . . . .	33



# Capitolo 1

## Introduzione

Nel seguente lavoro verranno presentati i due algoritmi elaborati da F. Pompili, N. Gillis, P.A. Absil e F. Gilneur e da loro descritti nell'articolo *Two algorithms for ONMF with application to clustering* [1]: l'algoritmo **Expectation-Maximization Orthogonal Nonnegative Matrix Factorization** (EM-ONMF) basato su una variante pesata delle  $k$ -medie sferiche e l'algoritmo **Orthogonal Nonnegatively Penalized Matrix Factorization** (ONP-MF) basato su due metodi di ottimizzazione, Lagrangiana Aumentata e Gradiente Proiettato.

In questo primo capitolo verranno definiti i concetti fondamentali di *Clustering* e *fattorizzazione ortogonale non negativa* attraverso una formalizzazione matematica utile ai fini della trattazione.

### 1.1 Clustering

Un metodo importante per la compressione e la classificazione dei dati è la loro organizzazione in **cluster**: un cluster è un sottoinsieme di dati che sono simili tra loro in base ad un criterio quantitativo prestabilito, generalmente una misura di distanza. È possibile calcolare il valore medio di ciascun cluster separatamente ed utilizzare tale valore come rappresentante per quel determinato sottoinsieme oppure, equivalentemente, gli oggetti possono es-

sere rappresentati da vettori definiti da una combinazione lineare dei valori medi.

**Definizione 1.1.** Sia  $M = (m_1, \dots, m_n) \in \mathbb{R}_+^{m \times n}$  una matrice di  $n$  dati  $m_j \in \mathbb{R}_+^m$  con  $j = 1, \dots, n$  e  $\Pi$  l'insieme dei loro indici.

Determinare  $k$  **cluster** significa trovare una partizione di  $k$  elementi  $\{\pi_i\}_{i=1}^k$  di  $\Pi$  tale che:

- $\pi_i \subseteq \{1, 2, 3, \dots, n\} \quad \forall i$
- $\bigcup_{1 \leq i \leq k} \pi_i = \{1, 2, \dots, n\}$
- $\pi_i \cap \pi_j = \emptyset \quad \forall i \neq j$

Risulta a questo punto opportuno introdurre un legame importante che intercorre tra matrici e partizioni, definendo il concetto di **compatibilità**.

**Definizione 1.2.** Una partizione  $\{\pi_i\}_{i=1}^k$  e una matrice  $V$  con  $k$  righe si dicono **compatibili** se vale:

$$j \in \pi_i \Leftrightarrow V_{ij} \neq 0$$

e si indica  $V \sim \pi$ .

*Osservazione 1.3.*  $V \sim \pi$  se e solo se ogni colonna di  $V$  ha al massimo un elemento non nullo (quello di posizione  $(i, j)$  dove  $i$  è determinato da  $\pi_i \ni j$ ).

**Esempio 1.4.** Considero una possibile partizione di 8 elementi in 5 cluster:

**cluster 1** 2 elementi

**cluster 2** 2 elementi

**cluster 3** 2 elementi

**cluster 4** 1 elemento

**cluster 5** 1 elemento

## 1.2 Fattorizzazione non negativa

---

Una matrice  $V \in \mathbb{R}^{5 \times 8}$  che sia compatibile con tale partizione potrebbe essere la seguente matrice binaria:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

## 1.2 Fattorizzazione non negativa

La fattorizzazione non negativa è un problema di algebra lineare che consiste nella scomposizione di una data matrice non negativa, cioè priva di elementi negativi,  $M \in \mathbb{R}_+^{m \times n}$  nel prodotto di due matrici  $U \in \mathbb{R}_+^{m \times k}$  e  $V \in \mathbb{R}_+^{k \times n}$ , entrambe a loro volta non negative. La dimensione della fattorizzazione  $k$  viene generalmente scelta inferiore alle dimensioni  $m$  e  $n$  di  $M$  in modo da ottenere una coppia di matrici di fattorizzazione più piccole di quella originale: ciò che ne risulta è una versione compressa della matrice di partenza.

$$M = UV \text{ tale che } M, U, V \geq 0 \quad (1.1)$$

Da un punto di vista applicativo, la restrizione di non negatività è ereditata dalla non negatività dei dati considerati: dato un set di  $n$  vettori  $m$ -dimensionali non negativi, tali vettori vengono inseriti nelle colonne della matrice  $M$ , che viene poi approssimata dalla fattorizzazione tramite le matrici  $U$  e  $V$ . L'equazione (1.1) può essere allora riscritta colonna per colonna come:

$$m_i = Uv_i \text{ per } i = 1, \dots, n \quad (1.2)$$

In altre parole ogni vettore  $m_i$  è approssimato da una combinazione lineare delle colonne di  $U$  pesata rispetto alle componenti del vettore  $v_i$ , quindi si può affermare che le colonne di  $U$  rappresentano una base ottimizzata per l'approssimazione lineare dei dati in  $M$ . Dato che vengono utilizzati relativamente pochi vettori di una base per rappresentare molti dati, si ottiene una

buona approssimazione solo se i vettori di tale base svelano una struttura che è latente nei dati.

Non essendo il problema risolvibile esplicitamente, vengono solitamente cercate delle approssimazioni numeriche. In termini di ottimizzazione il problema assume la formulazione (1.3):

Data una matrice non negativa  $M \in \mathbb{R}_+^{m \times n}$  e un rango di fattorizzazione  $k$ :

$$\min_{U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{k \times n}} \|M - UV\|_F^2 \quad \text{tale che } U \geq 0, V \geq 0 \quad (1.3)$$

La natura non lineare e non convessa in  $U$  e  $V$  di (1.3) viene ovviata facendo ricorso al metodo dei **Minimi Quadrati Alternati**, di cui è presente una descrizione in appendice.

### 1.3 Fattorizzazione non negativa ortogonale

L'aggiunta della condizione di ortogonalità e normalità sulle righe della matrice  $V$

$$VV^T = I$$

a (1.1) determina una profonda modifica della natura del problema. In termini di ottimizzazione il problema assume la formulazione (1.4):

Data una matrice non negativa  $M \in \mathbb{R}_+^{m \times n}$  e un rango di fattorizzazione  $k$ :

$$\min_{U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{k \times n}} \|M - UV\|_F^2, \quad \text{tale che } U \geq 0, V \geq 0 \text{ e } VV^T = I_k \quad (1.4)$$

La più immediata conseguenza delle condizioni imposte sulla matrice  $V$  è che ciascuna colonna presenti al massimo un termine non nullo. Sia  $i_j$  l'indice dell'eventuale elemento non nullo della  $j$ -esima colonna di  $V$ : una soluzione  $(U, V)$  del problema (1.4) è tale che:

### 1.3 Fattorizzazione non negativa ortogonale

---

- per  $j = 1, \dots, n$  la  $i_j$ -esima colonna di  $U$  minimizza l'angolo con la  $j$ -esima colonna di  $M$ ;
- il coefficiente  $V(i_j, j)$  scala  $U(:, i_j)$  in modo da renderla il più simile possibile a  $M(:, j)$ .

**Esempio 1.5.** Si consideri la matrice non negativa  $M \in \mathbb{R}_+^{5735 \times 1033}$  (i cui riferimenti sono rimandati alla sezione 2.4.1) e un'approssimazione di una fattorizzazione ortogonale non negativa di rango  $k = 50$ :

$$M = UV.$$

Un *mesh* della matrice  $V$  permette di verificare che tutti gli elementi hanno magnitudine compresa tra 0 e 1 (come ci si aspetta da una matrice le cui righe hanno norma unitaria) e che pochi sono gli elementi di magnitudine elevata. Riducendo a 0 tutti gli elementi di magnitudine inferiore a 0.2, si può dunque apprezzare la sparsità delle colonne della matrice.

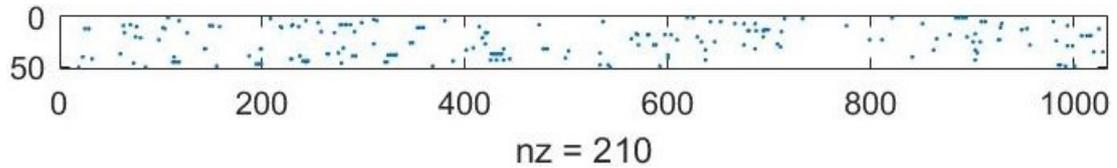


Figura 1.1: Spy della matrice

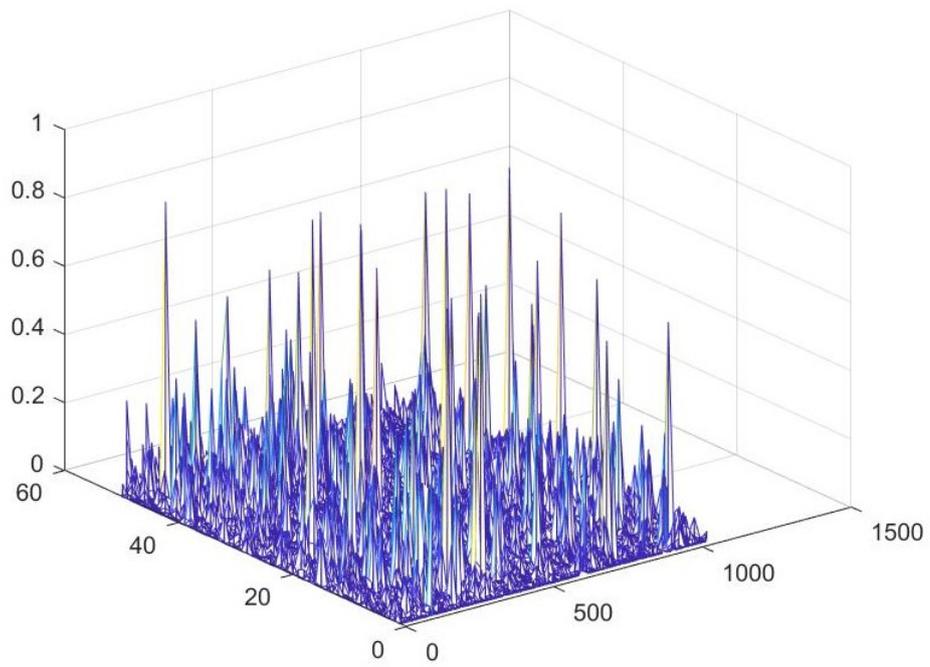


Figura 1.2: Mesh della matrice

# Capitolo 2

## EM-ONMF

### 2.1 Il metodo delle $k$ -medie

Se il criterio quantitativo scelto nella *Definizione 1.1* del problema del clustering è la distanza euclidea tra i punti, si ottiene il problema delle  $k$ -medie:

$$\min_{\{\pi_i\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} \|m_j - c_i\| \quad \text{dove } c_i = \frac{\sum_{j \in \pi_i} m_j}{|\pi_i|} \quad (2.1)$$

L'idea dell'algoritmo è immediata: data una partizione arbitraria iniziale dei dati  $m_j$ , vengono calcolati i centroidi, cioè i punti medi  $c_i$  di ciascuna partizione, cioè del cluster; successivamente per ogni punto in un dato cluster si verifica se esista un altro cluster con centroide più vicino e in caso positivo si procede con una riassegnazione del punto all'interno della partizione.

In (2.1) questo viene ottenuto chiedendo che i punti  $m_j$  all'interno di ciascun cluster  $\pi_i$  minimizzino la somma delle distanze dal relativo centroide  $c_i$ .

Equivalentemente si può definire una **matrice binaria indicatrice del cluster**  $B \in \{0, 1\}_{k \times n}$ :

$$B = \{b_{ij}\}_{k \times n} \quad \text{dove } b_{ij} = 1 \Leftrightarrow j \in \pi_i$$

*Osservazione 2.1.* Il fatto che i cluster  $\pi_i$  sono disgiunti implica che le righe di  $B$  siano ortogonali.

Normalizzando le righe di  $B$  si ottiene la matrice

$$V = \{v_{ij}\}_{k \times n} = (BB^T)^{-\frac{1}{2}}B$$

che soddisfa la seguente condizione:

$$\exists \{\pi_i\}_{i=1}^k \text{ tale che } v_{ij} = \begin{cases} \frac{1}{\sqrt{|\pi_i|}} & \text{se } j \in \pi_i \\ 0 & \text{altrimenti} \end{cases} \quad (2.2)$$

**Esempio 2.2.** Questo esempio riprende l'Esempio 1.4. Considerando nuovamente la matrice  $B$ , si calcola la norma di ciascuna riga

$$BB^T = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

e si determina la matrice  $V$  normalizzando le righe di  $B$ :

$$V = (BB^T)^{-\frac{1}{2}}B = \begin{bmatrix} 0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 & 0 \\ 0 & 0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 \\ 0.7071 & 0 & 0 & 0 & 0.7071 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Come dimostrato in [2], il problema (1.3) con una matrice  $V$  che soddisfa

## 2.2 Le $k$ -medie sferiche e ONMF

---

(2.2) equivale al problema delle  $k$ -medie:

$$\begin{aligned}\|M - UV\|_F^2 &= \sum_{j=1}^n \left\| m_j - \sum_{i=1}^k u_i v_{ij} \right\|^2 \\ &= \sum_{i=1}^k \sum_{j \in \pi_i} \left\| m_j - u_i v_{ij} \right\|^2 \\ &= \sum_{i=1}^k \sum_{j \in \pi_i} \left\| m_j - u_i \frac{1}{\sqrt{|\pi_i|}} \right\|^2\end{aligned}$$

dove ogni colonna di  $U$  corrisponde ad un centroide del cluster (a meno di fattori moltiplicativi).

## 2.2 Le $k$ -medie sferiche e ONMF

Il metodo delle  $k$ -medie può essere modificato considerando centroidi di norma unitaria e determinando l'assegnazione dei punti a ciascun cluster in base al quadrato della distanza dal centroide della loro proiezione sulla sfera unitaria. Quello che si ottiene si definisce come problema delle  $k$ -medie sferiche:

$$\min_{\{\pi_i, u_i\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} \left\| \frac{m_j}{\|m_j\|} - u_i \right\|^2 \quad \text{tale che } \|u_j\|_2 = 1. \quad (2.3)$$

Tramite operazioni elementari è possibile determinare una formulazione del problema delle  $k$ -medie sferiche in termini di massimo:

$$\begin{aligned}\left\| \frac{m_j}{\|m_j\|} - u_i \right\|^2 &= \left( \frac{m_j}{\|m_j\|} - u_i \right)^T \left( \frac{m_j}{\|m_j\|} - u_i \right) = \\ &= \frac{m_j^T m_j}{\|m_j\|^2} - \frac{m_j^T u_i}{\|m_j\|} - \frac{u_i^T m_j}{\|m_j\|} + u_i^T u_i = \\ &= 2 - 2 \frac{u_i^T m_j}{\|m_j\|}\end{aligned}$$

quindi

$$\min_{\{\pi_i, u_i\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} \left\| \frac{m_j}{\|m_j\|} - u_i \right\|^2 = \min_{\{\pi_i, u_i\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} \left( 2 - 2 \frac{u_i^T m_j}{\|m_j\|} \right)$$

e minimizzare una somma di termini negativi equivale a massimizzare la somma degli stessi termini ma con segno positivo. Così facendo si è ottenuto il problema:

$$\max_{\{\pi_i, u_i\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} \frac{m_j^T u_i}{\|m_j\|} \quad \text{tale che } \|u_j\| = 1 \quad (2.4)$$

*Osservazione 2.3.* In (2.4) risulta maggiormente evidente che la partizione dell'insieme dei dati viene fatta in base alle loro direzioni: l'argmax è una somma di coseni, il che si traduce nel massimizzare ogni singolo addendo, cioè ogni singolo coseno dell'angolo compreso tra il versore-dato e il versore-centroide del cluster a cui il dato appartiene. Massimizzare il coseno tra due vettori equivale a minimizzare l'angolo tra essi compreso e di conseguenza avvicinare le direzioni da essi individuate.

È possibile indebolire le richieste su  $V$  rispetto a (2.2), ottenendo la condizione

$$V \geq 0 \quad \text{e} \quad VV^T = I_k \quad (2.5)$$

e un nuovo problema

$$\min_{U \geq 0, V \geq 0} \|M - UV\|_F^2 \quad \text{tale che } VV^T = I_k. \quad (2.6)$$

*Osservazione 2.4.* La sostanziale differenza tra (2.2) e (2.5) è che la seconda non impone che tutti gli elementi non nulli delle righe di  $V$  assumano lo stesso valore.

Il risultato innovativo descritto in [1] è l'equivalenza tra il problema (2.6) e la seguente variante pesata del problema delle ***k*-medie sferiche**

$$\max_{\{\pi_i, u_i \in \mathbb{R}_+^m, \|u_i\|_2=1\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} \|m_j\|^2 \left( \frac{m_j^T u_i}{\|m_j\|} \right)^2. \quad (2.7)$$

*Osservazione 2.5.* In (2.7) compare una sommatoria di quadrati di coseni e ogni termine è pesato rispetto al quadrato della magnitudine del dato  $m_j$ , a differenza di (2.4) che è costruito su una sommatoria semplice di coseni. Le differenze tra i due problemi è apprezzabile a livello applicativo, come si vedrà più avanti.

## 2.2 Le $k$ -medie sferiche e ONMF

---

**Teorema 2.6.** Per una matrice  $M \in \mathbb{R}_+^{m \times n}$  il problema ONMF

$$\min_{U \geq 0, V \geq 0} \|M - UV\|_F^2 \quad \text{tale che} \quad VV^T = I_k$$

equivale alla seguente variante pesata delle  $k$ -medie sferiche

$$\max_{\{\pi_i, u_i \in \mathbb{R}_+^m, \|u_i\|_2=1\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} \|m_j\|^2 \left( \frac{m_j^T u_i}{\|m_j\|} \right)^2$$

dove  $\{\pi_i, u_i\}_{i=1}^k$  è un insieme di cluster distinti.

*Dimostrazione.* Si supponga  $k$  sufficientemente piccolo in modo da non avere soluzioni con colonne (nel caso di della matrice  $U$ ) o righe (nel caso della matrice  $V$ ) identicamente nulle. Nelle prossime righe sarà utilizzata la notazione  $U \geq 0$  per indicare che la matrice  $U$  è non negativa e priva **colonne** che si annullano,  $V \geq 0$  per indicare che la matrice  $V$  è non negativa e priva **righe** che si annullano.

Imponendo in (2.6) che siano le colonne di  $U$  invece delle righe di  $V$  ad avere norma unitaria, si ottiene un nuovo problema equivalente a (2.7):

$$\min_{U \geq 0, V \geq 0} \|M - UV\|_F^2 \quad \text{tale che} \quad (VV^T)_{ij} = 0 \quad \forall i \neq j \quad \text{e} \quad \|u_i\| = 1 \quad \forall i = 1, \dots, k \quad (2.8)$$

Si noti che la coppia di soluzioni  $(U, V)$  non è unica: è sempre possibile determinare una matrice  $D \in \mathbb{R}^{k \times k}$  tale che  $M = (UD)(D^{-1}V)$ . Inoltre, chiamando  $\Omega$  il *feasible set* di  $V$  in (2.6), risulta che  $V \geq 0$  e  $V \sim \pi$  implicano  $V \in \Omega$  e, viceversa,  $V \in \Omega$  implica che  $\exists \pi$  tale che  $V \sim \pi$ .

La dimostrazione del teorema si ottiene dall'equivalenza tra le seguenti quattro proprietà:

**Prop 1**  $U$  e  $V$  minimizzano  $\|M - UV\|_F^2$  soggetto a  $U \geq 0$ ,  $\|u_i\| = 1 \quad \forall i$ ,  $V \geq 0$ ,  $(VV^T)_{ij} = 0 \quad \forall i \neq j$  e  $\pi$  è ottenuto con operazioni elementari per soddisfare  $V \sim \pi$ .

**Prop 2**  $U$ ,  $V$  e  $\pi$  minimizzano  $\|M - UV\|_F^2$  soggetto a  $U \geq 0$ ,  $\|u_i\| = 1 \quad \forall i$ ,  $V \geq 0$  e  $V \sim \pi$ .

**Prop 3**  $U$ ,  $V$  e  $\pi$  minimizzano  $\sum_{i=1}^k \sum_{j \in \pi_i} \|m_j - u_i v_{ij}\|^2$  soggetto a  $U \geq 0$ ,  $\|u_i\| = 1 \forall i$ ,  $V \geq 0$  e  $V \sim \pi$ .

**Prop 4**  $U$  e  $\pi$  massimizzano  $\sum_{i=1}^k \sum_{j \in \pi_i} (m_j^T u_i)^2$  soggetto a  $U \geq 0$ ,  $\|u_i\| = 1 \forall i$ , e  $V$  è ottenuta con le operazioni elementari

$$\begin{cases} v_{ij} = 0 & \text{se } j \notin \pi_i \\ v_{ij} = m_j^T u_i & \text{se } j \in \pi_i \end{cases}$$

L'equivalenza tra **Prop 1** e **Prop 2** è una conseguenza dell'osservazione 2.2, mentre quella tra **Prop 2** e **Prop 3** è data da una riscrittura della funzione. Si consideri l'implicazione tra **Prop 3** e **Prop 4**: dati una matrice  $U$  e un insieme di cluster  $\pi$  ammissibili, per ogni termine  $\|m_j - u_i v_{ij}\|^2$ , il termine ottimale  $v_{ij}^*$  è dato da

$$\begin{aligned} v_{ij}^* &= \operatorname{argmin}_{x \geq 0} \|m_j - u_i x\|^2 \\ &= \operatorname{argmin}_{x \geq 0} (m_j^T m_j - 2x m_j^T u_i + x^2) \\ &= m_j^T u_i, \quad 1 \leq i \leq k, j \in \pi_i \end{aligned} \tag{2.9}$$

Sostituendo  $v_{ij}^*$  nell'espressione della Prop 3, si ottiene che  $U$  e  $\pi$  minimizzano

$$\begin{aligned} \sum_{i=1}^k \sum_{j \in \pi_i} \|m_j - (m_j^T u_i) u_i\|^2 &= \sum_{i=1}^k \sum_{j \in \pi_i} (m_j^T m_j - 2(m_j^T u_i)^2 + (m_j^T u_i)^2) \\ &= \sum_{i=1}^k \sum_{j \in \pi_i} -(m_j^T u_i)^2 + \text{costante} \end{aligned} \tag{2.10}$$

quindi massimizzano

$$\sum_{i=1}^k \sum_{j \in \pi_i} (m_j^T u_i)^2.$$

Il viceversa si ottiene per assurdo.

□

### 2.3 Algoritmo EM-ONMF

Questa variante pesata delle *k-medie sferiche*, in cui viene tenuto conto della norma dei dati, risulta molto efficace quando lo *scaling* dei dati non deve influenzare l'assegnazione dei punti ai cluster (es: due documenti riguardanti lo stesso argomento risultano uno multiplo dell'altro) e quando i dati con maggiore norma sono i più affidabili e ad essi deve essere attribuita maggiore importanza (es: quando il rumore che contamina i dati non dipende dalla norma dei dati stessi o quando i dati con norma maggiore sono statisticamente più rilevanti).

La struttura dell'algoritmo, come quella di molti altri algoritmi applicati al data clustering, è basata sullo schema EXPECTATION-MAXIMIZATION, ragione per cui viene chiamato "EM-ONMF": EM ipotizza dei valori per i parametri tenendo conto dei dati mancanti e modifica il modello per adattare ipotesi e osservazioni; successivamente ipotizza i valori dei parametri del nuovo modello e definisce una distribuzione di probabilità (questo passo è detto "E-step", da *Expected distribution*); i dati poi osservati vengono inseriti nel modello e la distribuzione di probabilità del passo precedente viene modificata per includere i nuovi dati (questo passo è talvolta detto "M-step", da *Maximum Likelihood*); gli ultimi tre passaggi vengono ripetuti finché non viene raggiunta stabilità (cioè una distribuzione che non cambia dal passo E alla fase M).

Nel caso specifico di EM-ONMF, presi in input  $M \in \mathbb{R}_+^{m \times n}$  matrice di dati non negativa e  $\{u_i\}_{i=1}^k$  set iniziale di centroidi è possibile definire due step principali:

**Step 1** - assegnazione di ogni punto al cluster più vicino (la formalizzazione è costruita sugli indici dei dati):

$$j \in \pi_i \Rightarrow i \in \underset{1 \leq l \leq k}{\operatorname{argmax}}(m_j^T u_l)^2 = \underset{1 \leq l \leq k}{\operatorname{argmax}}(m_j^T u_l)$$

**Step 2** - calcolo dei nuovi centroidi  $\{u_i\}_{i=1}^k$  sulla base dei nuovi cluster  $\{\pi_i\}_{i=1}^k$ :

$$\max_{\{u_i \geq 0, \|u_i\|=1\}_{i=1}^k} \sum_{i=1}^k \sum_{j \in \pi_i} (m_j^T u_i)^2 = \max_{\{u_i \geq 0, \|u_i\|=1\}_{i=1}^k} \sum_{i=1}^k \|M_i^T u_i\|_2^2$$

dove  $M_i = M(:, \pi_i)$ .

A questo punto si presentano  $k$  problemi indipendenti per cui bisogna determinare:

$$u_i^* = \operatorname{argmax}_{\|u\|_2=1} \|M_i^T u\|_2^2 = \operatorname{argmax}_{\|u\|_2=1} u^T M_i M_i^T u$$

e dato che

$$\|M_i^T u_i^*\|_2 = \sigma_1(M_i) = \|M_i\|$$

allora  $u_i^*$  corrisponde al vettore singolare sinistro associato alla matrice  $M_i$ . Inoltre, essendo  $M \geq 0$ , la teoria di **Perron-Frobenius** permette di scegliere  $u_i^*$  non negativo.

Ricapitolando:

**per**  $iter = 1, 2, \dots$  fino a convergenza

$$\{\pi_i\}_{i=1}^k \leftarrow \emptyset$$

**per**  $j$  **da** 1 **a**  $n$ :

trovare  $i \in \operatorname{argmax}_{1 \leq l \leq k} (m_j^T u_l)$

aggiornare il cluster  $\pi_i = \pi_i \cup \{j\}$

**se**  $\pi_i = \emptyset$  per qualche  $i$ , **allora** trasferire in modo random un punto al cluster  $\pi_i$

**per**  $i$  **da** 1 **a**  $k$ :

porre come  $u_i$  il vettore singolare dominante non negativo della sottomatrice  $M_i = M(:, \pi_i)$

*Osservazione 2.7.* I dati di output dell'algoritmo, per come esso è strutturato, non sono le esplicite soluzioni del problema ONMF (1.4), ma quelle del

## 2.4 Esperimenti Numerici

---

problema di clustering, cioè la partizione dei dati  $\{\pi_i\}_{i=1}^k$  e le direzioni dei centroidi di ogni cluster  $\{u_i\}_{i=1}^k$ .

Una soluzione al problema (2.6):

$$\min_{U \geq 0, V \geq 0} \|M - UV\|_F^2 \quad \text{t.c.} \quad (VV^T)_{ij} \neq 0 \quad \forall i \neq j \quad \text{e} \quad \|u_i\| = 1 \quad \forall i$$

può essere ottenuta considerando  $U = [u_1, \dots, u_k]$  e  $V = (v_{ij})$  dove da (2.9)

$$v_{ij} = m_j^T u_i, \quad 1 \leq i \leq k, j \in \pi_i.$$

Una soluzione al problema (1.4)

$$\min_{U \geq 0, V \geq 0} \|M - UV\|_F^2 \quad \text{t.c.} \quad VV^T = I_k$$

può essere ottenuta considerando  $(UD, D^{-1}V)$  dove  $U$  e  $V$  sono le matrici della trattazione precedente e  $D = \text{diag}(\|V(1, \cdot)\|_2, \dots, \|V(k, \cdot)\|_2)$ .

## 2.4 Esperimenti Numerici

### 2.4.1 Un primo esempio

In [1] è proposto un confronto tra i tre algoritmi ( $k$ -medie,  $k$ -medie sferiche e EM-ONMF) in cui essi vengono applicati a due diversi data set elementari per valutarne la performance in diverse situazioni.

La Figura 2.1 presenta un caso in cui i tre algoritmi agiscono in modo equivalente, perchè i due cluster, rappresentati tramite una diversa raffigurazione dei dati, sono ben separati: i diamanti neri indicano i centroidi calcolati con le  $k$ -medie, le linee nere continue indicano le direzioni dei centroidi delle  $k$ -medie sferiche e la linea rossa tratteggiata le direzioni dei centroidi di EM-ONMF.

La Figura 2.2 presenta un caso in cui  $k$ -medie e  $k$ -medie sferiche falliscono perchè i due cluster sono allineati: le  $k$ -medie riconoscono come cluster

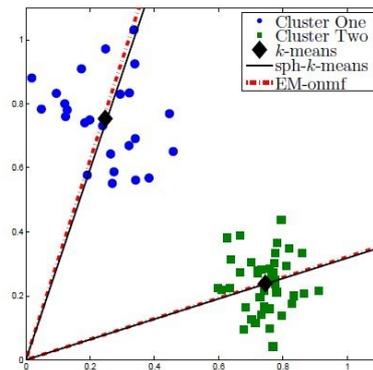


Figura 2.1: Cluster distinti

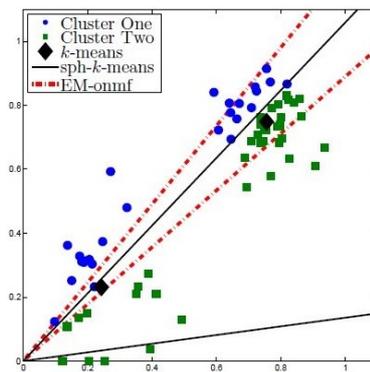


Figura 2.2: Cluster allineati

i gruppi di dati tra loro vicini dal momento che si basano unicamente sulla distanza euclidea, le  $k$ -medie sferiche si basano sulle direzioni dei dati e sono fortemente deviate dai dati con ascissa nulla ma norma molto piccola. EM-ONMF calcola invece i centroidi basandosi sulle direzioni ma attribuisce più importanza ai dati con norma maggiore.

## 2.4 Esperimenti Numerici

---

### 2.4.2 Document Clustering

Il *Document-Clustering* è l'applicazione della cluster-analysis a documenti testuali per, ad esempio, l'organizzazione automatica dei documenti, l'estrapolazione di un argomento, il filtraggio e il recupero di informazioni.

Si consideri ora la matrice termine-documento  $A \in \mathbb{R}^{5735 \times 1033}$  relativa ad un database di 1033 documenti di argomento medico da cui è stata estrapolata una lista di 5735 parole chiave (la stessa utilizzata nell'Esempio 1.5). A seguito di un'esecuzione dell'algoritmo per l'elaborazione di 50 clusters è stata ottenuta la matrice  $V \in \mathbb{R}^{50 \times 1033}$  la cui sparsità è già stata mostrata nell'Esempio 1.5. La Figura 2.3 mostra invece l'ortogonalità delle righe della matrice, a dimostrazione della correttezza dell'algoritmo nell'elaborazione di una fattorizzazione non negativa ortogonale.

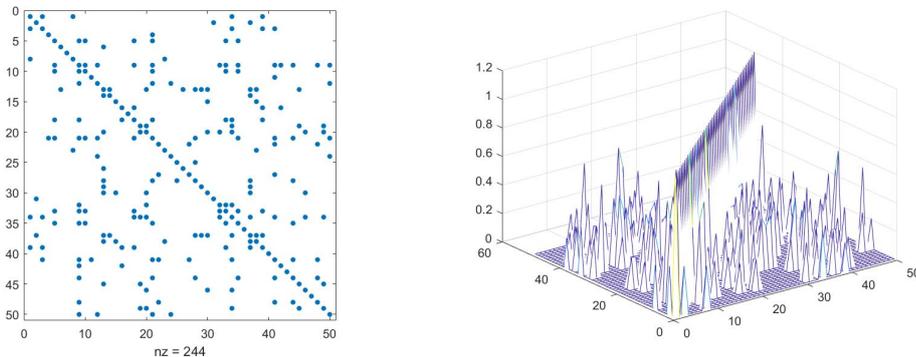


Figura 2.3: Ortogonalità della matrice  $V$ :  $VV^T = I_{50}$

Un confronto basato su numero di iterazioni e errore di approssimazione relativo

$$e_{rel} = \frac{\|A - UV\|_F}{\|A\|_F}$$

tra il metodo delle  $k$ -medie sferiche pesate e il metodo dei minimi quadrati alternati nell'elaborazione di una fattorizzazione non negativa, come mostrato dalla Figura 2.4, sottolinea la preferibilità del primo metodo: 11 iterazioni

e un errore relativo finale  $e_{rel} = 0.71$  con variazione tra la prima e l'ultima iterazione apprezzabile al secondo ordine di grandezza decimale, contro 30 iterazioni e un errore relativo finale  $e_{rel} = 0.923317$  con variazione tra la prima e l'ultima iterazione apprezzabile al settimo ordine di grandezza decimale.

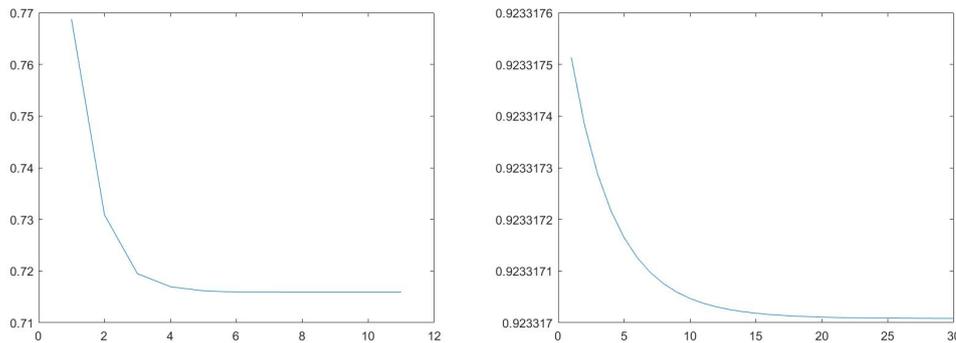


Figura 2.4: Errori Relativi con EM-ONMF (sinistra) e con ALS (destra)

Infine, procedendo nell'analisi dell'algoritmo e della sua applicazione alla matrice  $A$  è interessante vedere un esempio di come siano stati raggruppati i documenti nei relativi 50 cluster. La Tabella 2.1 mostra una parte del risultato: ciascuna delle 10 colonne corrisponde ad uno dei cluster, di cui è riportato nella seconda riga il numero totale di documenti contenuti; per ciascun cluster sono poi riportate in colonna le prime 28 parole chiave di un documento campione appartenente ad esso.

*Osservazione 2.8.* Il codice Matlab fornito in [1] elabora un cluster delle righe della matrice dei dati, di conseguenza per questo esempio è stata utilizzata la matrice  $A^T$ :

```
>>> [clusters,V,relError,actualIters] = emonmf(A',50,3000)
```

dove `clusters` è un vettore di  $\mathbb{R}^{1033}$  le cui componenti sono valori interi compresi tra 1 e 50: se alla  $j$ -esima componente è stato attribuito il valore  $k$ , significa che il  $j$ -esimo documento appartiene al  $k$ -esimo cluster. Per risalire

## 2.4 Esperimenti Numerici

---

ai documenti a partire dal vettore `clusters` è stato utilizzato il seguente codice:

```
for j=1:50
    documenti=find((clusters==j));
    fprintf('\n gruppo %1i : %1i documenti \n ',j, numerodocumenti);
    for i=1:length(documenti)
        fprintf('\n %1i) documento %1i\n',i,documenti(i,1));
        dict_med(find(A(:,documenti(i,1))),:))
    end
end
```

Tabella 2.1: Esempio di clustering di una matrice termine-documento

1	2	3	4	5	6	7	8	9	10	...
27 dox	26 dox	2 dox	25 dox	6 dox	28 dox	12 dox	21 dox	6 dox	11 dox	...
5	211	832	54	91	19	634	23	119	71	...
abnormally	albino	affected	account	action	cases	affected	abscess	absence	air	...
acid	anti	arterial	acetyl	activity	considered	breeding	addition	absorbed	alveolar	...
acids	antibodies	basically	acid	affected	death	carry	aetiological	absorption	apical	...
blood	antigenic	blood	active	appearance	erythematous	commercial	amyloid	affected	appreciable	...
body	assumed	cases	activity	cases	evidence	communication	amyloidosis	disease	areas	...
change	basement	commonest	affect	cells	immunologic	culture	biopsies	hepatic	basement	...
composition	bovine	complete	agglutination	cellular	infectious	culturing	biopsy	ingested	blood	...
concentration	capsule	conditions	alter	change	kidneys	describes	bronchiectasis	irrespective	bronchiolar	...
concentrations	common	consists	alteration	chronic	lupus	difficulty	cases	limits	capillary	...
content	component	contracture	aminocaprole	comparative	mechanical	establishment	chest	measured	caused	...
daily	demonstrate	damaged	antibody	elements	nephritis	examined	clinical	neuritis	cells	...
dawley	discussed	deprived	antigen	groups	patient	free	congo	normal	central	...
days	eye	die	antigenic	higher	patients	frequently	constant	normals	complete	...
determines	glomerulus	entire	antigens	incapable	presented	gallisepticum	data	orally	considerable	...
diabetes	heterologou	events	antiserum	initial	problems	high	definite	organs	correlated	...
elevations	immunofluo	fibrosis	ascites	investigation	renal	identify	deposits	peripheral	cytoplasmic	...
fat	inducing	fingers	cells	large	report	impended	determined	plasma	days	...
fatty	kidney	flexion	compared	leucocytos	resort	isolated	diagnosed	presence	degeneration	...
female	lens	forearm	derivate	leukaemia	responsible	known	diagnostic	present	earliest	...
fetal	membrane	fortunately	direct	leukaemic	secondary	lack	discovered	radioactive	edema	...
fetuses	participation	frank	drugs	lower	show	mycoplasma	disease	radioactivity	electron	...
free	pigmented	give	due	lymphatic	systemic	nucleus	emphasised	results	endoplasmic	...
gestation	possibility	group	eaca	lymphocytes	terminal	parent	evidence	size	endothelium	...
glucose	rat	held	epsilon	lymphoid	transplantation	poultis	factors	splenic	epithelium	...
group	reaction	hours	exerted	mitotic	transplanted	practical	failure	stage	evaluate	...
heparin	results	improved	found	normal	treated	presence	feature	subjects	flattened	...
high	sites	injury	globulins	occurs	treatment	production	foci	suffering	focal	...
hypotesis	source	ischaemic	immune	phenomena	uremia	proportion	frequency	vitamin	gradually	...



# Capitolo 3

## ONP-MF

Questo capitolo è dedicato alla descrizione di un secondo algoritmo elaborato per la risoluzione del problema di clustering attraverso la fattorizzazione non negativa; a differenza del primo algoritmo presentato, dove il risultato veniva raggiunto in seguito alla risoluzione di un problema equivalente, questo secondo algoritmo affronta direttamente il problema della fattorizzazione non negativa. Ciò che lo rende originale è la sua struttura: generalmente gli algoritmi di ONMF impongono la non negatività ad ogni iterazione e ottengono l'ortogonalità al limite tramite particolari accorgimenti, come l'utilizzo di un opportuno termine di penalizzazione, la formulazione di una matrice di proiezione o la scelta di una particolare direzione. Quanto viene proposto invece è esattamente lo schema inverso, cioè soddisfare la condizione di ortogonalità ad ogni iterazione tramite la tecnica del gradiente proiettato e gestire la condizione di non negatività tramite il metodo della lagrangiana aumentata, ottenendo così una fattorizzazione matriciale ortogonale non negativamente penalizzata, da cui il nome dell'algoritmo ONP-MF.

Risulta chiaro che la trattazione di questo secondo algoritmo è principalmente la descrizione di come adattare i metodi citati al problema considerato: le seguenti due sezioni sono dunque dedicate allo studio di tali metodi al fine di meglio comprendere la struttura di ONP-MF.

### 3.1 Metodo del Gradiente proiettato

Il Metodo del Gradiente proiettato viene utilizzato nel campo dell'ottimizzazione per la risoluzione di problemi di carattere vincolato. La sua particolarità è quella di permettere profonde modifiche all'interno del *working set* ad ogni iterazione, favorendo quindi la risoluzione del problema in un numero ridotto di iterazioni, a differenza di altri metodi per cui il *working set* perde o acquista al massimo una condizione ad ogni iterazione, rallentando così il raggiungimento di una soluzione, specialmente nei problemi in cui il *working set* iniziale è largamente differente dall'*active set* alla soluzione. Dati la proiezione sul *feasible set*  $P$  e la lunghezza del passo  $\alpha > 0$ , la ricerca della soluzione avviene lungo il cammino lineare a tratti

$$P(x_k - \alpha \nabla f(x_k)).$$

Si ottiene un nuovo punto ogni volta che si è determinato una opportuna lunghezza  $\alpha_k$ :

$$x_{k+1} = P(x_k - \alpha_k \nabla f(x_k)). \quad (3.1)$$

L'idea alla base di questo metodo è riassunta dalla seguente proposizione:

**Proposizione 3.1.** *Sia  $f$  differenziabile e convessa e sia  $Q : \mathbb{R}^n \rightarrow \mathbb{R}$  convessa; allora  $x^*$  è una soluzione ottimale di*

$$\operatorname{argmin}_x f(x) + Q(x)$$

*se e solo se*

$$x^* = \operatorname{prox}_{\nu, Q}(x^* - \nu \nabla f(x^*))$$

*dove*

$$\operatorname{prox}_{\nu, Q} : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$x \rightarrow \operatorname{argmin}_y \left( \frac{1}{2} \|x - y\|_2^2 + Q(y) \right)$$

*è detto proximal point mapping associato a  $Q$ .*

*Osservazione 3.2.* Per una funzione  $f$  non convessa, si noti che un punto fissato dell'iterazione del gradiente proiettato è un punto stazionario di  $f(x) + Q(x)$ .

## 3.2 Metodo della Lagrangiana aumentata

Il metodo della Lagrangiana aumentata è legato al metodo di penalizzazione quadratica, ma a differenza di quest'ultimo riduce le possibilità di malcondizionamento introducendo esplicitamente all'interno della funzione da minimizzare un set di moltiplicatori di Lagrange che vengono calcolati ad ogni iterazione. Si consideri il seguente problema vincolato con uguaglianza:

$$\min_x f(x) \text{ soggetto a } c_i(x) = 0, i \in I \quad (3.2)$$

la funzione di penalizzazione quadratica

$$Q(x, \mu) := f(x) + \frac{1}{2\mu} \sum_{i \in I} c_i^2(x) \quad (3.3)$$

penalizza le violazioni del vincolo elevando al quadrato le *infeasibility* e scalandole di  $\frac{1}{2\mu}$ . Tuttavia, come mostrato dal Teorema 17.2 in [3, p. 495] le approssimazioni  $x_k$  dei minimizzanti di  $Q(x, \mu_k)$  non soddisfano la *feasibility condition*  $c_i(x) = 0, i \in I$ : piuttosto infatti, sono lievemente perturbate per soddisfare in modo approssimato

$$c_i(x_k) = -\mu_k \lambda_i^*, i \in I \quad (3.4)$$

perturbazione che tende a 0 concordemente con  $\mu_k \downarrow 0$ .

Il metodo della Lagrangiana aumentata, attraverso la funzione  $\Lambda(x, \lambda, \mu)$ , permette, nel momento in cui si altera la funzione  $Q(x, \mu_k)$ , di evitare questa perturbazione sistematica, facendo in modo che le approssimazioni dei minimizzanti  $x_k$  soddisfino  $c_i(x) = 0$ . L'idea su cui si basa è quella di includere una stima esplicita dei moltiplicatori di Lagrange  $\lambda$  basata sulla Formula (17.8) in [3, p. 495] nella *objective function*:

$$\Lambda(x, \lambda, \mu) := f(x) + \frac{\mu_k}{2} \sum_{i \in I} c_i^2(x) - \sum_{i \in I} \lambda_i c_i(x). \quad (3.5)$$

*Osservazione 3.3.* La Lagrangiana aumentata differisce dalla Lagrangiana standard per la presenza del termine quadratico e dalla funzione di penalizzazione quadratica per la presenza di una sommatoria in cui sono coinvolti i moltiplicatori di Lagrange.

### 3.3 Struttura dell'algoritmo

Definita una matrice di moltiplicatori di Lagrange  $\Lambda \in \mathbb{R}_+^{k \times n}$  associata al vincolo di non negatività e un parametro di penalizzazione quadratica  $\rho$ , si costruisce la Lagrangiana aumentata che in notazione matriciale assume la forma

$$L_\rho(U, V, \Lambda) = \frac{1}{2} \|M - UV\|_F^2 + \langle \Lambda, -V \rangle + \frac{\rho}{2} \|\min\{V, 0\}\|_F^2. \quad (3.6)$$

Ciò che idealmente si cerca di risolvere è il problema

$$\max_{\Lambda \geq 0} f(\Lambda) \text{ dove } f(\Lambda) = \min_{U \geq 0, VV^T = I_k} L_\rho(U, V, \Lambda) \quad (3.7)$$

e indipendentemente dal parametro scelto, le soluzioni del problema (1.4) coincidono con le soluzioni di

$$\min_{U \geq 0, VV^T = I_k} \max_{\Lambda \geq 0} L_\rho(U, V, \Lambda). \quad (3.8)$$

Si pone poi la Varietà di Stiefel come codominio della proiezione che compare nel metodo del gradiente proiettato.

**Definizione 3.4.** Si definisce **Varietà di Stiefel** l'insieme delle matrici  $X \in \mathbb{R}^{n \times k}$  le cui colonne sono tra loro ortogonali:

$$St_{(k,n)} \stackrel{def}{=} \{X \in \mathbb{R}^{n \times k} : X^T X = I_k\}.$$

Data una matrice  $A$ , la sua proiezione nella varietà di Stiefel risulta essere:

$$Proj_{St}(A) \stackrel{def}{=} \underset{X \in St_{(k,n)}}{\operatorname{argmin}} \|A - X\|_F^2.$$

Per inizializzare l'algoritmo si pone  $\Lambda = 0$  e per le  $k$  righe di  $V$  si scelgono i primi  $k$  vettori singolari destri della matrice  $M$ , che possono essere ottenuti tramite SVD (i vantaggi di questa scelta saranno discussi in seguito). Lo schema di aggiornamento alternato delle variabili principali  $U, V, \Lambda$  a questo punto si presenta nel seguente modo:

### 3.4 Esperimenti Numerici

---

**Aggiornamento di  $U$**  - fissati  $V^{(t)}$  e  $\Lambda^{(t)}$ , si risolve il seguente problema ai minimi quadrati

$$U^{(t+1)} = \operatorname{argmin}_{X \in \mathbb{R}_+^{m \times k}} \|M - XV^{(t)}\|_F^2$$

**Aggiornamento di  $V$**  - fissati  $U^{(t+1)}$  e  $\Lambda^{(t)}$ , si applica il metodo del gradiente proiettato costruito su una proiezione nella Varietà di Stiefel

$$V^{(t+1)} = \operatorname{Proj}_{St}( V^{(t)} - \beta^{(t+1)} \nabla_V L_\rho(U^{(t+1)}, V^{(t)}, \Lambda^{(t)}) )$$

dove  $\beta$  è la lunghezza del passo, definita ad ogni iterazione tramite ricerca lineare: aumenta se la *objective function* è in diminuzione, altrimenti diminuisce.

**Aggiornamento di  $\Lambda$**  - i moltiplicatori di Lagrange vengono definiti al fine di penalizzare i termini negativi all'interno della matrice  $V^{(t+1)}$

$$\Lambda^{(t+1)} = \max(0, \Lambda^{(t)} - \frac{\alpha_0}{t} V^{(t+1)})$$

dove  $\alpha_0 > 0$  è un parametro costante.

*Osservazione 3.5.* Vale  $-V = \nabla_\Lambda L_\rho$ , quindi anche questo aggiornamento è costruito sullo schema del gradiente proiettato.

### 3.4 Esperimenti Numerici

L'inizializzazione dell'algoritmo tramite SVD (cioè l'inizializzazione delle righe di  $V$  con i vettori singolari di  $M$ ) influisce positivamente sulla stabilità del metodo: il problema (1.4) è di natura multimodale, cioè può avere molteplici minimi locali, e l'inizializzazione random di cui si caratterizzano molti algoritmi per l'ONMF proposti in letteratura non è in grado di garantire qualcosa oltre la convergenza ad un minimizzante locale e generalmente non globale. Inoltre tale scelta di  $V$  è la soluzione ottimale del problema di inizializzazione privato della condizione di non negatività e dato che quest'ultima è garantita al limite dalla struttura dell'algoritmo senza essere imposta ad ogni iterazione, non si rendono necessarie modifiche dei vettori singolari al

fine di renderli non negativi (come invece tipico di altri algoritmi): in questo modo l'inizializzazione risulta diretta.

### 3.4.1 Gli iris di Fisher

Utilizzando l'*iris data set* contenuto in Matlab all'interno del file `fisheriris` è stata verificata la stabilità dell'algoritmo. All'interno del data set è contenuta la matrice `meas`  $\in \mathbb{R}^{150 \times 4}$  che di 150 iris, appartenenti a tre specie diverse (*setosa*, *versicolor*, *virginica*), riporta indicazioni su 4 misure del fiore: lunghezza e larghezza del sepal, lunghezza e larghezza del petalo. L'identificazione di 3 cluster all'interno del data set ha ricondotto sempre ai seguenti risultati:

```

onpmf ls:init    plty:1.00e-02    misft:1.25e+02    neg:4.59e-01
onpmf ls: 100/3000 plty:2.70e-02    misft:5.20e-02    neg:3.80e-01
onpmf ls: 200/3000 plty:7.32e-02    misft:2.19e-02    neg:1.29e-01
onpmf ls: 300/3000 plty:1.98e-01    misft:1.51e-02    neg:2.94e-01
onpmf ls: 400/3000 plty:5.35e-01    misft:1.65e-02    neg:3.06e-01
onpmf ls: 500/3000 plty:1.45e+00    misft:1.43e-02    neg:2.62e-01
onpmf ls: 600/3000 plty:3.92e+00    misft:1.55e-02    neg:2.73e-01
onpmf ls: 700/3000 plty:1.06e+01    misft:1.78e-02    neg:2.83e-01
onpmf ls: 800/3000 plty:2.86e+01    misft:1.66e-02    neg:2.45e-01
onpmf ls: 900/3000 plty:7.75e+01    misft:1.46e-02    neg:2.68e-01
onpmf ls:1000/3000 plty:2.10e+02    misft:1.60e-02    neg:2.52e-01
onpmf ls:1100/3000 plty:5.67e+02    misft:2.80e-02    neg:1.32e-01
onpmf ls:1200/3000 plty:1.53e+03    misft:3.90e-02    neg:4.44e-02
onpmf ls:1300/3000 plty:4.15e+03    misft:4.60e-02    neg:1.83e-02
onpmf ls:1400/3000 plty:1.12e+04    misft:4.90e-02    neg:6.95e-03
onpmf ls:1500/3000 plty:3.03e+04    misft:5.01e-02    neg:2.61e-03

```

Figura 3.1: ONP-MF - Aggiornamento ogni 100 iterazioni (colonna 1) dei valori: Parametro di Penalizzazione inizializzato  $\rho = 100$  (colonna 2), Errore Relativo di Fattorizzazione  $e_{rel} = \frac{\|M - UV\|_F}{\|M\|_F}$  (colonna 3), Residuo di Negatività della matrice V  $res_{neg} = \frac{\|\min\{V, 0\}\|_F}{\|V\|_F}$  (colonna 4)

### 3.4 Esperimenti Numerici

---

Iterazioni	Param. di Penalizzazione	Errore Relativo	Residuo di Negatività
1597	8e+04	0.051	0.001028

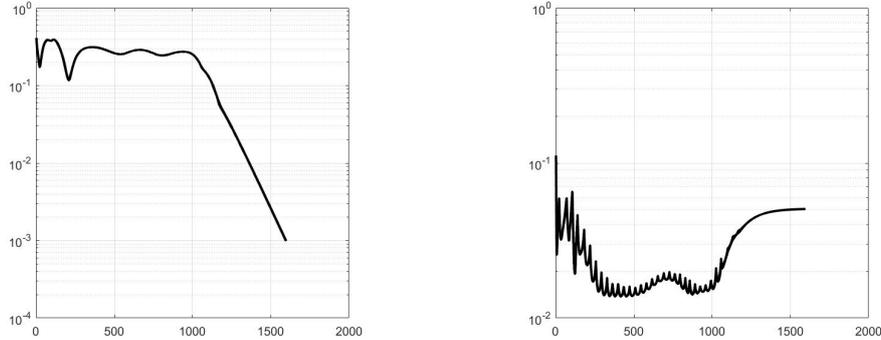


Figura 3.2: Sinistra: **Residuo di negatività**. Destra: **Errore relativo**

Anche i risultati legati alla fattorizzazione non negativa rimangono costanti: la norma della differenza tra le matrici  $U$  e  $V$  ottenute in diversi run dell'algoritmo è sempre nulla.

La matrice  $U \in \mathbb{R}^{4 \times 3}$  ha come colonne i centroidi dei tre cluster:

$$\begin{bmatrix} 0.66586 & 0 & 0 \\ 0.3414 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

*Osservazione 3.6.* Lo spazio iniziale quadridimensionale (in cui ogni direzione corrisponde ad una dimensione del fiore) è stato ridotto ad uno spazio tridimensionale in cui il primo vettore corrisponde ad una combinazione lineare delle dimensioni del sepalo, mentre gli ultimi due corrispondono rispettivamente alla lunghezza e alla larghezza del petalo.

La matrice  $V \in \mathbb{R}^{3 \times 150}$  rappresentata in Figura 3.3 attraverso la funzione *plot* riporta come colonne le coordinate di ciascun fiore nel nuovo spazio tridimensionale di cui i centroidi rappresentano una base. Si può osservare che

la suddivisione in cluster è ben identificata dalle righe 2 e 3 della matrice  $V$ , rappresentate in rosso e giallo, da cui si può dedurre che le dimensioni che permettono di discriminare i 3 gruppi sono la lunghezza e la larghezza del petalo, poichè corrispondono al secondo e terzo centroide. La prima riga si mantiene invece costante su tutti i campioni del data set risultando quindi poco interessante ai fini di una classificazione dei fiori.

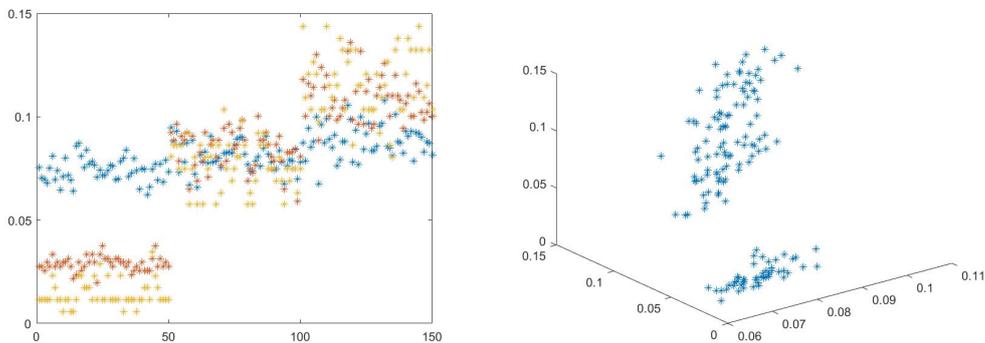


Figura 3.3: Sinistra: plot bidimensionale della matrice  $V$  con in ascissa l'indice di colonna e in ordinata il valore dei tre elementi della colonna, differenziati per colore. Destra: plot tridimensionale della matrice  $V$ .

Infine, il plot tridimensionale mostrato sempre in Figura 3.3 riporta una suddivisione dei dati in due gruppi principali: per distinguere ulteriormente gli elementi all'interno del gruppo più grande è necessaria un'analisi di discriminanza.

### 3.4.2 Medline

Questa sezione ha lo scopo di mostrare che l'algoritmo rispetta le previsioni mostrate nel primo capitolo circa la struttura della matrice  $V$ : a tale proposito è stata nuovamente utilizzata la matrice  $A$  presentata nel secondo capitolo.

### 3.4 Esperimenti Numerici

---

Tramite un *mesh* della matrice  $V$  è possibile verificare che sono pochi gli elementi di magnitudine elevata e che tutti hanno valori compresi tra 0 e 1.

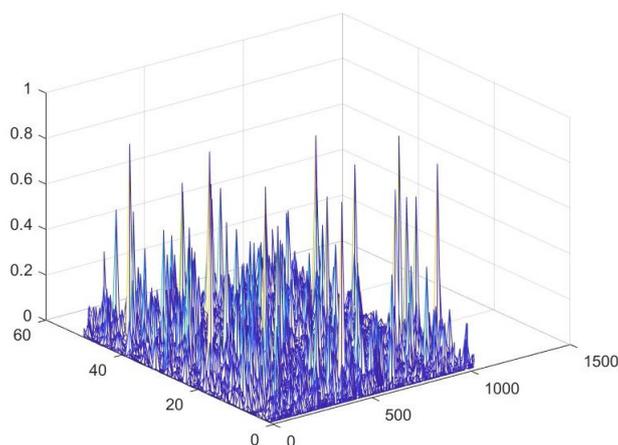


Figura 3.4: Mesh della matrice  $V$

Uno *spy* degli elementi superiori a 0.02 mette in evidenza la sparsità della matrice, in particolare la tendenza delle colonne a mantenere un unico elemento non nullo.

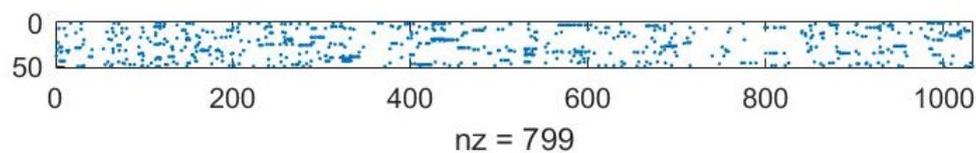
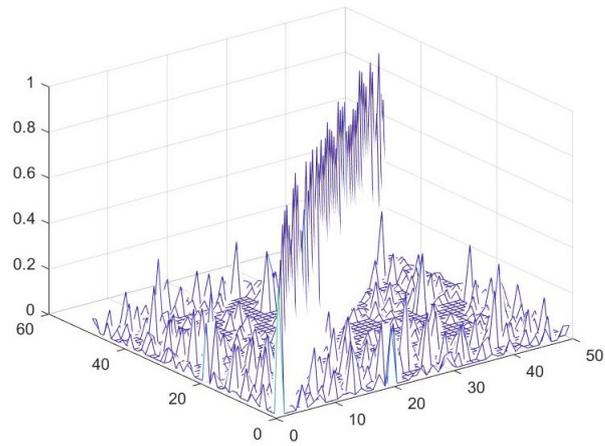
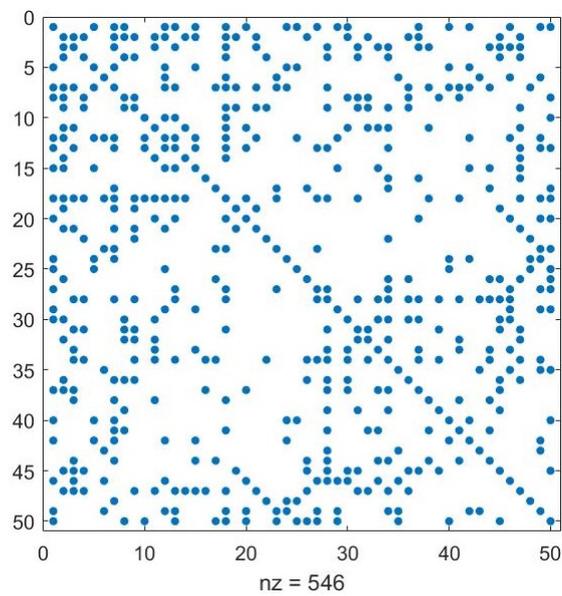


Figura 3.5: Spy della matrice  $V$

Infine, l'ortogonalità delle righe della matrice è invece testimoniata dalla tendenza della matrice  $VV^T$  alla matrice identità: un *mesh* mostra chiaramente la tendenza della matrice ad una struttura diagonale, fatto evidenziato anche dalla funzione *spy*, e la tendenza degli elementi al valore unitario.

Figura 3.6: Mesh della matrice  $VV^T$ Figura 3.7: Spy della matrice  $VV^T$

# Capitolo 4

## Confronti Numerici

### 4.1 *Hyperspectral Unmixing* su Hubble data set

Un'immagine iperspettrale è un insieme di fotogrammi dello stesso soggetto catturati a diverse lunghezze d'onda: ognuno di essi è acquisito misurando la riflettanza di ogni singolo pixel a una data lunghezza d'onda. La decomposizione spettrale consiste nel classificare i pixel in cluster diversi, corrispondenti ciascuno ad un materiale diverso. L'utilizzo di un metodo come la fattorizzazione non negativa è chiaramente suggerito dalla non negatività dei dati, poichè la riflettanza è un dato sempre positivo. In particolare vengono classificate le colonne di una matrice lunghezza d'onda - riflettanza del pixel in modo che ogni cluster, cioè un set di pixel, corrisponda a un particolare tipo di materiale.

Oggetto della seguente analisi è un confronto in parallelo delle prestazioni dei due algoritmi, dei quali verranno considerati velocità, stabilità ed efficienza. L'immagine campione a cui verrà applicata la decomposizione spettrale è quella del telescopio spaziale Hubble, di cui un esempio è offerto dalla Figura 4.1, mentre la Figura 4.2 riporta l'effettiva decomposizione del satellite in base ai materiali costituenti ed è il riferimento rispetto al quale verrà saggiata la capacità degli algoritmi di elaborare un buon clustering.

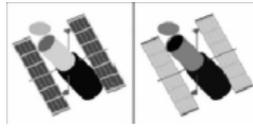


Figura 4.1: Immagini campione dell'hubble data set a due diverse frequenze

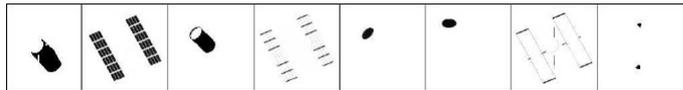


Figura 4.2: Decomposizione in base ai materiali

Diverse esecuzioni dell'algorithmo EM-ONMF restituiscono generalmente risultati diversi, in termini di clusters, numero di iterazioni e tempi di calcolo. Su un campione di 5 esecuzioni sono stati ottenuti i cluster mostrati dalla Figura 4.3 e i rispettivi valori mostrati dalla Tabella 4.1.

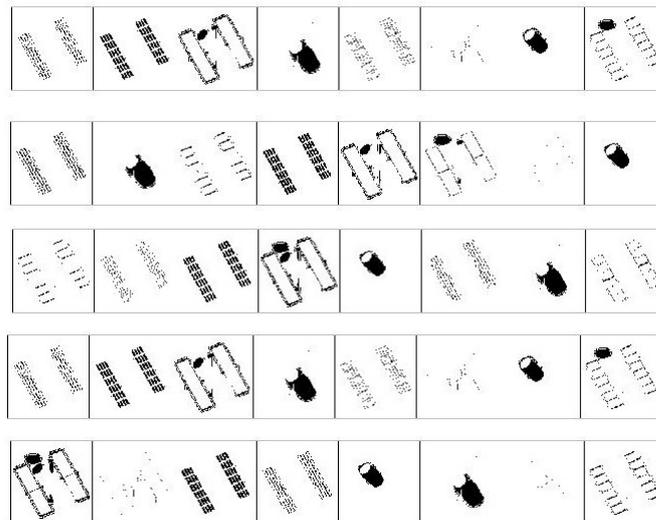


Figura 4.3: Clustering con EM-ONMF

Si può osservare che l'algorithmo è piuttosto performante quanto a riconoscimento dei diversi materiali rispetto ad altri algoritmi ONMF (i cui risultati sono mostrati in Figura 4.6) e sicuramente molto veloce a scapito tuttavia della stabilità.

#### 4.1 Hyperspectral Unmixing su Hubble data set

---

Tabella 4.1: EM-ONMF

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>valori medi</b>
<b>iterazioni</b>	23	18	35	23	17	21
<b>elapsed time</b>	0.8049	0.5629	1.3579	0.8049	0.4349	0.9694

Per quanto riguarda ONP-MF, 5 iterazioni hanno riportato sempre gli stessi risultati in termini di cluster (Figura 4.4), numero di iterazioni (2207), errore relativo (0.048), residuo di negatività (0.001002) e valore finale del parametro di penalizzazione ( $3 \times 10^7$ ). L'unico parametro soggetto a variazioni è il tempo di calcolo, come mostrato dalla Tabella 4.2.

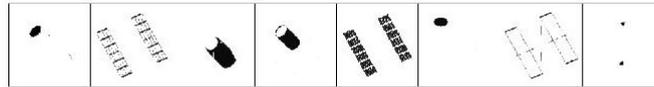


Figura 4.4: Clustering con ONP-MF

Tabella 4.2: Elapsed Time ONP-MF

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>valore medio</b>
83.7110	74.9943	78.7913	76.8211	75.074	77.8782

Si può apprezzare l'alto livello di prestazione nell'elaborazione di cluster perchè le componenti riconosciute sono esattamente le stesse mostrate dalla Figura 4.2. Tale livello di precisione in questo contesto inficia però i tempi di calcolo e l'elaborazione di una fattorizzazione matriciale precisa rispetto ad EM-ONMF. Un confronto basato sull'errore relativo

$$e_{rel} = \frac{\|M - UV\|_F}{\|M\|_F}$$

mostra infatti che lo scarto tra i due algoritmi corrisponde ad un ordine di grandezza:  $10^{-2}$  per EM-ONMF e  $10^{-1}$  per ONP-MF.

## 4. Confronti Numerici

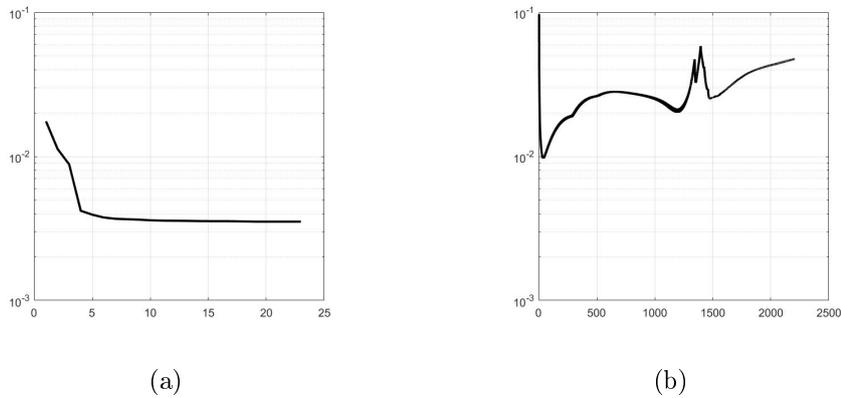


Figura 4.5: Residuo Relativo con EM-ONMF (a) e ONP-MF (b)

*Osservazione 4.1.* Il valore dell'errore relativo sembra convergere ad un valore fissato in ONP-MF, ma un'effettiva e rigorosa dimostrazione della convergenza non è ancora stata elaborata.

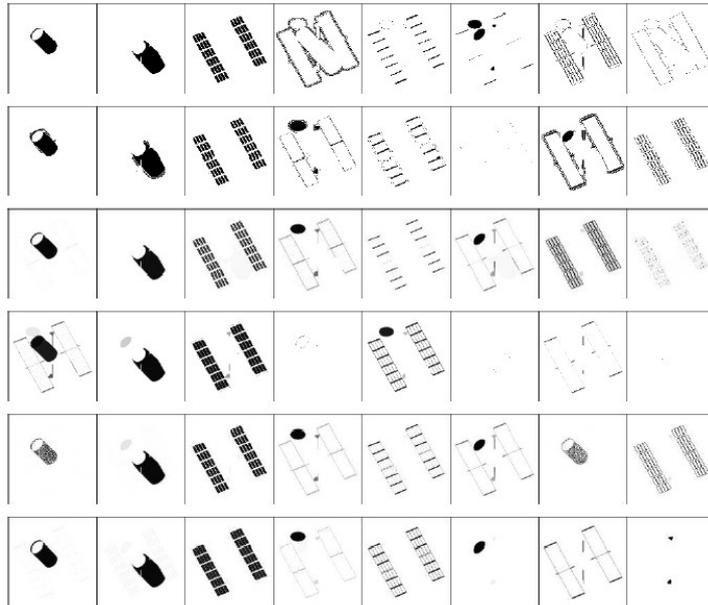


Figura 4.6: Cluster elaborati da altri algoritmi:  $k$ -medie,  $k$ -medie sferiche, CHNMF, CH(SVD), O-PNMF, O-P(SVD). L'immagine compare in [1], dove sono presenti anche tutti i riferimenti agli articoli in cui vengono trattati CHNMF e O-PNMF

## 4.2 *Handwritten digits*

La classificazione automatica di caratteri scritti a mano è un esempio standard di problema di *pattern recognition* e viene tipicamente utilizzato per il riconoscimento dei codici postali. In [4] sono presenti i riferimenti al database del servizio postale statunitense, utilizzato in questa sezione. Ciascun carattere viene trattato equivalentemente o come un'immagine  $28 \times 28$  pixel in scala di grigi o come un vettore in  $\mathbb{R}^{784}$  ottenuto prima traducendo l'immagine in una matrice in  $\mathbb{R}^{28 \times 28}$  i cui termini corrispondono al valore intero sulla scala di grigi del pixel a cui sono associati e poi incolonnando i vettori colonna di tale matrice.

Il seguente confronto mira a mostrare un esempio di problema di clustering in cui EM-ONMF e ONP-MF falliscono rispetto all'algoritmo delle  $k$ -medie standard. Il data set in questione è stato creato a partire dal database sopra citato utilizzando 10 digits del test set della cifra 3 e 10 del test set della cifra 4: ciascuno dei digit, trattato come vettore in  $\mathbb{R}^{784}$ , è stato inserito come colonna in una matrice  $T \in \mathbb{R}^{784 \times 20}$ . La Figura 4.7 mostra i due digit corrispondenti ai valori medi dei due cluster. La richiesta è quella di elaborare due cluster e oggetto del confronto sono i due centroidi che ciascun algoritmo è in grado di calcolare. I risultati sono mostrati nella Figura 4.8.

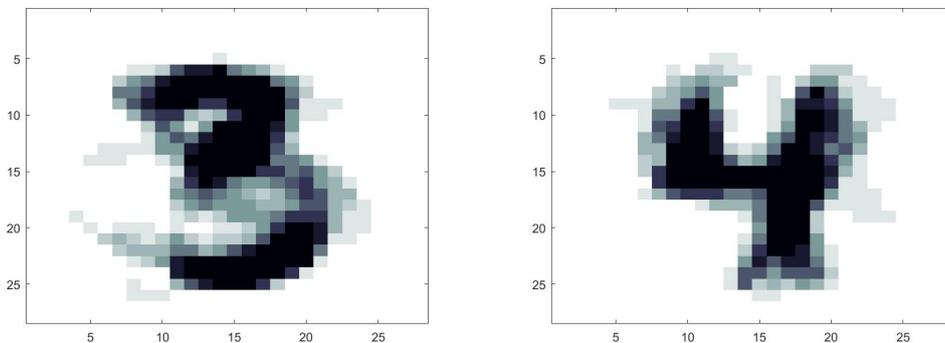


Figura 4.7: Valori medi

Le Figure (a) e (b) mostrano i centroidi che un'applicazione dell'algoritmo delle  $k$ -medie alla matrice  $T$  ha calcolato in 9 iterazioni: le cifre raffigurate sono chiaramente riconducibili ad un 3 ed un 4. Le Figure (c) e (d) mostrano i centroidi calcolati da EM-ONMF in 2 iterazioni: l'immagine (c) si avvicina ad un 9 e l'immagine (d) è molto simile ad un 3 ma l'approssimazione è meno precisa di quella fornita dalle  $k$ -medie. Le Figure (e) e (f) mostrano i centroidi calcolati da ONP-MF in 2962 iterazioni: le immagini sono frammentate e le cifre sono difficilmente riconoscibili.

ONP-MF risulta dunque il peggiore algoritmo da utilizzare in questo tipo di problemi, sia per qualità dei centroidi ottenuti, sia per numero di iterazioni richieste. Si può tuttavia osservare che anche in questo caso il metodo mostra grande stabilità: 5 esecuzioni dell'algoritmo hanno riportato sempre gli stessi risultati, esattamente come osservato nella sezione precedente.

## 4.2 Handwritten digits

---

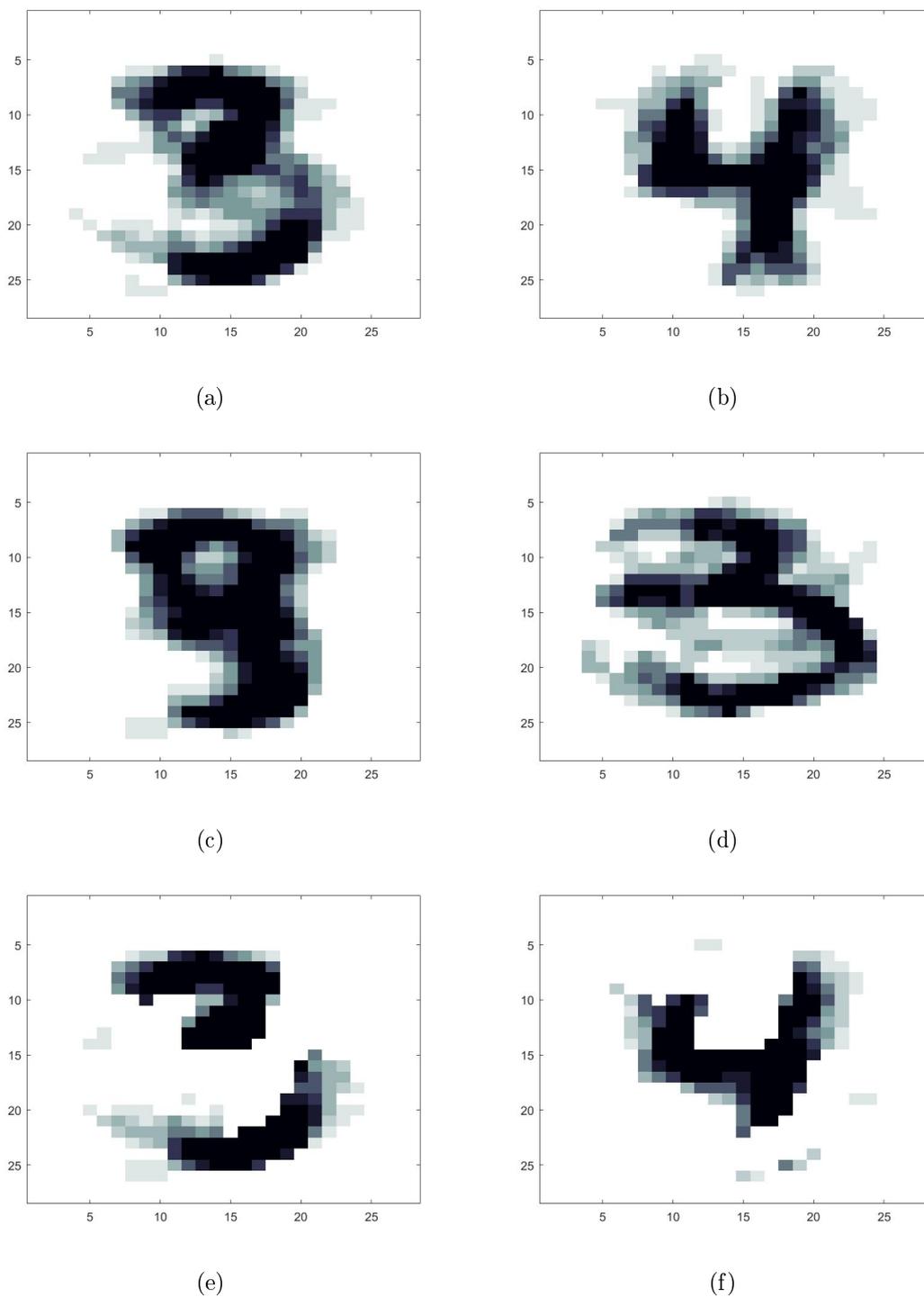


Figura 4.8: Coppie di centroidi elaborate dai tre algoritmi:  $k$ -medie (a) e (b), EM-ONMF (c) e (d), ONP-MF (e) e (f)



# Conclusioni

In questa tesi è stato presentato il problema della fattorizzazione non negativa ortogonale e i due metodi descritti in [1] per elaborarne un'approssimazione della soluzione, da sfruttare ai fini di una ripartizione di un set di dati in cluster: l'algoritmo EM-ONMF basato sullo schema *expectation-maximization* e sull'equivalenza tra una variante pesata delle  $k$ -medie sferiche e il problema in questione; l'algoritmo ONP-MF dove la fattorizzazione ortogonale ottenuta grazie al metodo del gradiente proiettato viene nonnegativamente penalizzata ad ogni iterazione tramite il metodo della Lagrangiana aumentata. Ciascun algoritmo è stato applicato a un diverso data set in modo da apprezzarne le peculiarità e verificarne eventuali difetti. Per EM-ONMF è stato preso in considerazione un problema di *document-clustering* utilizzando una matrice termine-documento di ambito medico sulla quale è stata verificata l'ortogonalità della fattorizzazione ottenuta e una migliore prestazione quanto a precisione della fattorizzazione e numero di iterazioni rispetto al metodo delle  $k$ -medie standard. Per ONP-MF è stato invece utilizzato l'Iris data set di Matlab sul quale si è verificato che, al prezzo di un alto numero di iterazioni, il metodo risulta stabile, grazie all'inizializzazione diretta tramite SVD. Infine il problema di *hyperspectral-unmixing* applicato a un set di immagini a diverse frequenze del telescopio spaziale Hubble e il problema di *pattern recognition* applicato al database del servizio postale statunitense hanno permesso rispettivamente l'uno il confronto tra i due metodi in un ambito in cui entrambi presentano prestazioni migliori rispetto ad altri algoritmi di ONMF, l'altro invece un esempio in cui entrambi falli-

## CONCLUSIONI

---

scono rispetto al metodo standard delle  $k$ -medie. Da notare che nel caso di *hyperspectral-unmixing* ONP-MF risulta meno preciso in termini di fattorizzazione e più lento in termini di numero di iterazioni ma è quello che più si avvicina alla reale decomposizione dei materiali.

# Appendice A

## Minimi Quadrati Alternati

Il problema (1.4) interpretato come problema di ottimizzazione nelle variabili  $U$  e  $V$  è di carattere non lineare, ma se una delle matrici fosse nota, ad esempio  $U$ , allora il problema di determinare  $V$  sarebbe un comune problema di minimi quadrati con condizione di non negatività. Su questa idea si basa il metodo dei minimi quadrati alternati: calcolare una matrice considerando nota l'altra. Lo schema è il seguente:

1. Ipotizzare una valore  $U^{(1)}$

2. **Per  $t=1,2,\dots$  fino a convergenza**

- Risolvere  $\min_{V \geq 0} \|M - U^{(t)}V\|_F$  per ottenere  $V^{(t)}$
- Risolvere  $\min_{U \geq 0} \|M - UV^{(t)}\|_F$  per ottenere  $U^{(t+1)}$

Se  $m_j$  e  $v_j$  sono colonne di  $M$  e  $V$ , si può notare che il problema ai minimi quadrati  $\min_{V \geq 0} \|M - U^{(t)}V\|_F$  equivale a  $n$  problemi ai minimi quadrati vettoriali indipendenti

$$\min_{v_j \geq 0} \|m_j - U^{(t)}v_j\|_2, \quad j = 1, 2, \dots, n.$$

Nello stesso modo, trasponendo le matrici, il problema ai minimi quadrati per determinare  $U$  può essere riformulato come  $m$  problemi ai minimi quadrati vettoriali indipendenti. Questo metodo è alla base della funzione Matlab `lsqnonneg`.



# Bibliografia

- [1] F. Pompili, N. Gillis, P. A. Absil, F. Glineur, *Two algorithms for Orthogonal Nonnegative Matrix Factorization with Application to Clustering*, Neurocomputing, vol. 141, pp. 15-25, Oct. 2014.
- [2] C. Ding, X. He, H. D. Simon, *On the equivalence of nonnegative matrix factorization and spectral clustering*, In SIAM Int. Conf. Data Mining (SDM'05), pp. 606–610, 2005.
- [3] J. Nocedal, S. J. Wright,, *Numerical optimization* (2nd ed.), Berlin, New York: Springer-Verlag, 2006.
- [4] L. Eldén, *Matrix methods in data mining and pattern recognition* (2nd ed.), Philadelphia, SIAM, 2007.

