



Università degli Studi di Bari Aldo Moro

Report Progetto 1

Emanuele Fontana

Esame di Metodi Numerici per l'Informatica
Anno accademico 2023/2024

Indice

1	Introduzione	2
1.1	Traccia Progetto 1	2
1.2	Dati iniziali	2
2	Criterio 1: GUTTMAN-KEISER	3
3	Criterio 2: SCREENPLOT DI COTTEL	5
4	Criterio 3: ENTROPIA	7
5	Criterio 4: ENERGIA	9
6	Criterio 5: K-MEANS ISOLATION FOREST	11
6.1	Con logaritmo	11
6.2	Senza logaritmo	12
7	Conclusioni	14

1 Introduzione

1.1 Traccia Progetto 1

- Leggere le seguenti due immagini in Matlab, usando i comandi:

```
imt1 = imread('Bern1.bmp');
imt2 = imread('Bern2.bmp');

if(size(imt1,3)>1)
    imt1 = rgb2gray(imt1);
end
if(size(imt2,3)>1)
    imt2 = rgb2gray(imt2);
end
```

- Costruire la matrice della differenza, avendo prima trasformato i dati in tipo double:

```
imt1 = double(imt1)
imt2 = double(imt2)
Diff_Mat = abs(imt2-imt1)
```

- Calcolare la SVD troncata di Diff Mat, in particolare trascrivere in Matlab almeno tre criteri per selezionare automaticamente il numero di componenti k da trattenere.
- Calcolare errore relativo in norma 2 e in norma di Frobenius per l'approssimazione ottenuta.
- Trasformare la matrice dell'approssimazione in un vettore colonna e scalare i valori tra 0 e 1.
- Applicare un algoritmo di binarizzazione variando la soglia (per un esempio, si veda il file Grad img.m).
- Visualizzare l'immagine finale, fancendo un opportuno reshape e utilizzando il comando imshow. L'immagine cos' ottenuta dovrebbe avere in bianco i pixel che identificano zone di cambiamento tra l'immagine I1 e l'immagine I2, mentre in nero, i pixel relativi alle zone non cambiate.
- Commentare i risultati

1.2 Dati iniziali

La matrice **Diff_Mat** è una matrice di dimensione 301×301 , in quanto le immagini **Bern1.bmp** e **Bern2.bmp** sono immagini di tale dimensione.

Il rank della matrice **Diff_Mat** è 301 dunque massimo. Il che significa che tutti i valori singolari sono non nulli. Le matrici U , S e V hanno dimensioni rispettivamente 301×301 , 301×301 e 301×301 .

2 Criterio 1: GUTTMAN-KEISER

Il criterio di Guttman-Keiser per la selezione dei valori singolari da mantenere della SVD troncata prevede di mantenere i valori singolari σ_i che sono maggiori di 1:

$$\sigma_i > 1 \quad (1)$$

Si tratta dunque di un criterio "oggettivo" in quanto è basato su una soglia numerica ben definita.

In questo caso specifico tutti i valori singolari sono maggiori di 1, quindi tutti i valori singolari vengono mantenuti.

Il criterio in questo caso, di fatto, si è dunque rivelato inutile.

Matrice	Righe	Colonne
U_troncata	301	301
S_troncata	301	301
V_troncata	301	301

Tabella 1: Dimensioni matrici troncate

Norma	Errore assoluto
2	0.0
Frobenius	0.0

Tabella 2: Norme ed errori

Dato che tutti i valori singolari sono stati mantenuti, l'errore relativo è nullo.

Per la binarizzazione sono state utilizzate le seguenti soglie:

- 0.25
- 0.5
- 0.75
- Soglia automatica calcolata con **graythresh**

La funzione **graythresh** calcola la soglia di binarizzazione utilizzando l'algoritmo di Otsu, il quale minimizza la varianza intra-classe (equivalente a massimizza la varianza inter-classe).

La varianza intra-classe è definita come:

$$\sigma^2(t) = w_0(t) \cdot \sigma_0^2(t) + w_1(t) \cdot \sigma_1^2(t) \quad (2)$$

dove:

- $\sigma^2(t)$ è la varianza totale per la soglia t
- $w_0(t)$ e $w_1(t)$ sono le probabilità che le due classi siano separate dalla soglia t e dalla varianza σ_i^2
- $\sigma_0^2(t)$ e $\sigma_1^2(t)$ sono le varianze delle due classi separate dalla soglia t

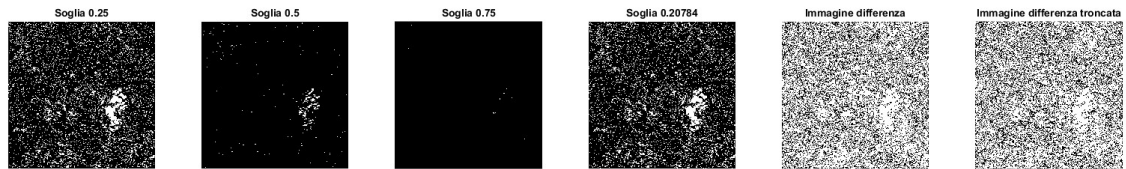


Figura 1: Immagini binarizzate con criterio 1

Come è possibile notare, ovviamente, in questo caso non vi è differenza tra l'immagine originale e quella approssimata.

Per quanto riguarda la binarizzazione si può notare come le soglie 0.5 e 0.75 abbiano prodotto pessimi risultati, mentre la soglia 0.25 e quella automatica abbiano prodotto risultati migliori.

Per il processo di troncamento sono stati necessari in media circa **0.0018** secondi.

3 Criterio 2: SCREENPLOT DI COTTEL

Il criterio di Screenplot di Cattel per la selezione dei valori singolari da mantenere della SVD troncata prevede di effettuare un grafico dei valori singolari σ_i e di individuare il "gomito" del grafico e mantenere i valori singolari che si trovano prima di esso.

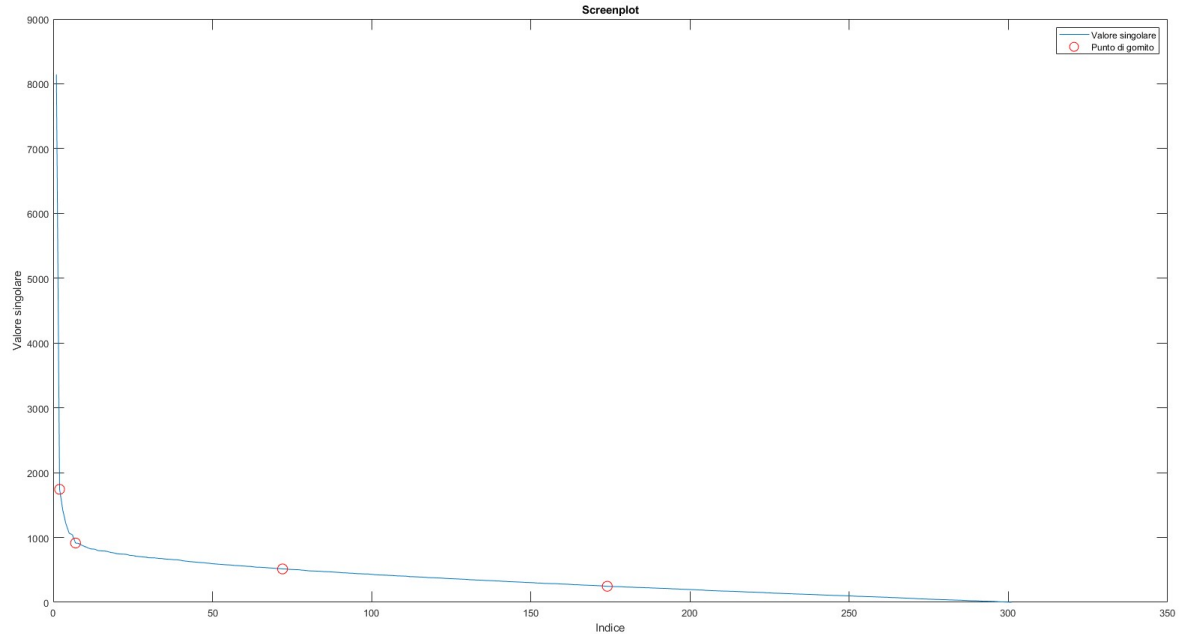


Figura 2: Plot dei valori singolari

Mediante la funzione **findchangepts** sono stati individuati 4 punti di gomito.

La funzione **findchangepts** restituisce i punti di cambio in un vettore. Questi punti di cambio sono i punti in cui la media degli x cambia in modo significativo.

In questo caso si è deciso di procedere con il secondo gomito, ovvero $x=7$ (di fatto questo criterio è soggettivo). Sono stati mantenuti i primi 7 valori singolari.

Matrice	Righe	Colonne
U_troncata	301	7
S_troncata	7	7
V_troncata	301	7

Tabella 3: Dimensioni matrici troncate

Norma	Errore assoluto
2	907.434068
Frobenius	6782.450233

Tabella 4: Norme ed errori

E' noto inoltre che l'errore assoluto in norma due risulta essere σ_{k+1} e in norma Frobenius

$$\sqrt{\sum_{i=k+1}^r \sigma_i^2} \quad (3)$$

dove k è il numero di valori singolari mantenuti.

Per la binarizzazione sono state utilizzate le seguenti soglie:

- 0.25
- 0.5
- 0.75
- Soglia automatica calcolata con **graythresh**

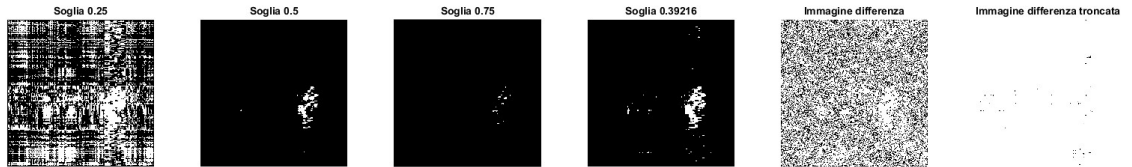


Figura 3: Immagini binarizzate con criterio 2

Come è possibile notare in questo caso vi è molta differenza tra l'immagine originale e quella approssimata. Osservando le soglie di binarizzazione possiamo vedere come in questo caso la soglia automatica e la soglia 0.5 abbiano prodotto il risultato più "coerente" con l'approssimazione, mentre la soglia 0.25 ha prodotto più "coerente" con l'originale.

Per il processo di troncamento sono stati necessari in media circa **0.07** secondi.

4 Criterio 3: ENTROPIA

Il criterio basato sull'entropia prevede di calcolare il contributo di ogni valore singolare mediante la seguente formula:

$$f_j = \frac{\sigma_j^2}{\sum_{i=1}^r \sigma_i^2} \quad (4)$$

e successivamente calcolare l'entropia totale della matrice mediante la seguente formula:

$$E = -\frac{1}{\log(r)} \sum_{j=1}^r f_j \log(f_j) \quad (5)$$

Con questo criterio, vengono selezionati i primi σ_k valori singolari, dove k viene scelto come il più piccolo indice tale che:

$$\sum_{j=1}^k f_j \geq E \quad (6)$$

In questo caso l'entropia totale della matrice è risultata essere 0.517648. Volendo mantenere il 75% dell'entropia totale il k suggerito è 116.

Matrice	Righe	Colonne
U_troncata	301	116
S_troncata	116	116
V_troncata	301	116

Tabella 5: Dimensioni matrici troncate

Norma	Errore assoluto
2	384.477333
Frobenius	2892.327572

Tabella 6: Norme ed errori

E' noto inoltre che l'errore assoluto in norma due risulta essere σ_{k+1} e in norma Frobenius

$$\sqrt{\sum_{i=k+1}^r \sigma_i^2} \quad (7)$$

dove k è il numero di valori singolari mantenuti.

Per la binarizzazione sono state utilizzate le seguenti soglie:

- 0.25
- 0.5
- 0.75
- Soglia automatica calcolata con **graythresh**

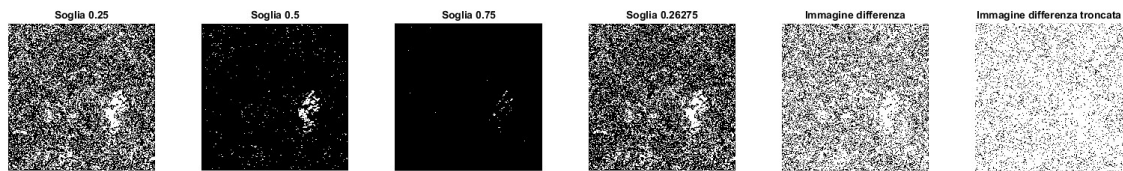


Figura 4: Immagini binarizzate con criterio 3

Le soglie 0.25 e automatica sono molto simili tra loro e sono "coerenti" sia con l'immagine originale che con quella troncata.

Per il processo di troncamento sono stati necessari in media circa **0.0009** secondi.

5 Criterio 4: ENERGIA

Il criterio basato sull'energia prevede di calcolare l'energia totale della matrice mediante la seguente formula:

$$Energia = \sum_{j=1}^r \sigma_j^2 \quad (8)$$

e di scegliere un numero k di valori singolari tale che

$$\sum_{j=1}^k \sigma_j^2 \quad (9)$$

sia almeno il 90% dell'energia totale.

In questo caso l'energia totale della matrice è risultata essere 121874051. Volendo mantenere il 90% dell'energia totale, ovvero 109686645.9, il k suggerito è 94, con un energia pari a 109708426.061981

Matrice	Righe	Colonne
U_troncata	301	94
S_troncata	94	94
V_troncata	301	94

Tabella 7: Dimensioni matrici troncate

Norma	Errore assoluto
2	442.680374
Frobenius	3487.925592

Tabella 8: Norme ed errori

E' noto inoltre che l'errore assoluto in norma due risulta essere σ_{k+1} e in norma Frobenius

$$\sqrt{\sum_{i=k+1}^r \sigma_i^2} \quad (10)$$

dove k è il numero di valori singolari mantenuti.

Per la binarizzazione sono state utilizzate le seguenti soglie:

- 0.25
- 0.5
- 0.75
- Soglia automatica calcolata con **graythresh**

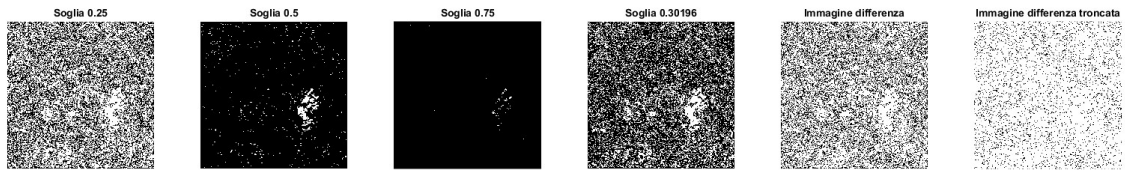


Figura 5: Immagini binarizzate con criterio 4

Le soglie 0.25 e automatica sono molto simili tra loro e sono "coerenti" sia con l'immagine originale che con quella troncata.

Per il processo di troncamento sono stati necessari in media circa **0.006** secondi.

6 Criterio 5: K-MEANS ISOLATION FOREST

6.1 Con logaritmo

Dati i contributi di ciascun valore singolare, descritti in (4), si è scelto di utilizzare il criterio K-Means Isolation Forest.

La prima operazione è stata quella di calcolare il logaritmo di ciascun contributo e successivamente dividere i valori in 2 cluster: **Rumore** e **Informazione**.

Il K-Means però è un apprendimento non supervisionato, dunque non è possibile sapere a priori quale cluster corrisponda a Rumore e quale a Informazione.

Poichè nei valori singolari

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_r \geq 0 \quad (11)$$

allora sicuramente il cluster contenente il contributo di σ_1 sarà quello relativo all'informazione.

Una volta identificato tale cluster, si procede a rilevare anomalie mediante l'algoritmo di Isolation Forest.

Il risultato dell'algoritmo ha portato a mantenere i primi 225 valori singolari

Matrice	Righe	Colonne
U_troncata	301	225
S_troncata	225	225
V_troncata	301	225

Tabella 9: Dimensioni matrici troncate

Norma	Errore assoluto
2	141.942286
Frobenius	723.122392

Tabella 10: Norme ed errori

E' noto inoltre che l'errore assoluto in norma due risulta essere σ_{k+1} e in norma Frobenius

$$\sqrt{\sum_{i=k+1}^r \sigma_i^2} \quad (12)$$

dove k è il numero di valori singolari mantenuti.

Per la binarizzazione sono state utilizzate le seguenti soglie:

- 0.25
- 0.5
- 0.75
- Soglia automatica calcolata con **graythresh**

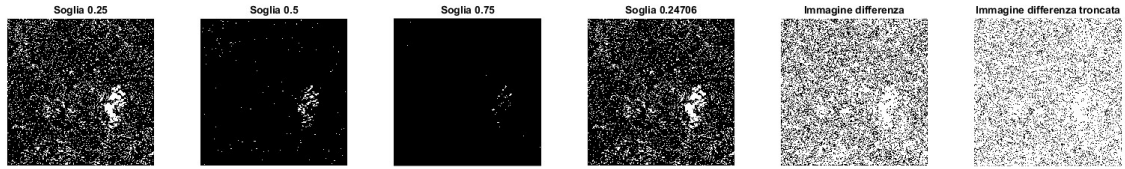


Figura 6: Immagini binarizzate con criterio con logaritmo

Le soglie 0.25 e automatica sono molto simili tra loro e sono "coerenti" sia con l'immagine originale che con quella troncata.

Per il processo di troncamento sono stati necessari in media circa **0.35** secondi.

6.2 Senza logaritmo

Il risultato dell'algorithm ha portato a mantenere i primi 225 valori singolari

Matrice	Righe	Colonne
U_troncata	301	1
S_troncata	1	1
V_troncata	301	1

Tabella 11: Dimensioni matrici troncate

Norma	Errore assoluto
2	1742.857352
Frobenius	7454.849653

Tabella 12: Norme ed errori

E' noto inoltre che l'errore assoluto in norma due risulta essere σ_{k+1} e in norma Frobenius

$$\sqrt{\sum_{i=k+1}^r \sigma_i^2} \quad (13)$$

dove k è il numero di valori singolari mantenuti.

Per la binarizzazione sono state utilizzate le seguenti soglie:

- 0.25
- 0.5
- 0.75
- Soglia automatica calcolata con **graythresh**

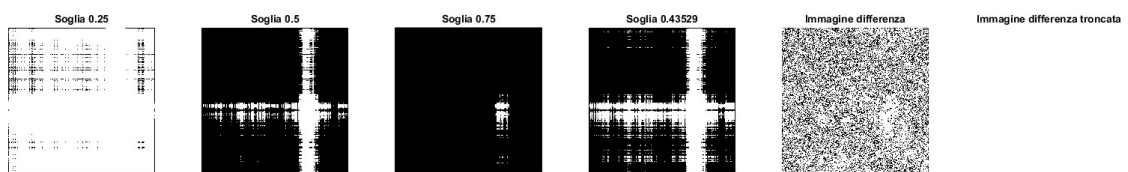


Figura 7: Immagini binarizzate con criterio 5 senza logaritmo

L'immagine troncata è difatto completamente bianca ad'occhio nudo, mentre l'immagine binarizzata mostra punti bianchi e neri (quindi di fatto ci sono altri valori oltre lo 0 del bianco ma sono impercettibili ad'occhio nudo).

Di fatto non è possibile capire quale sia la soglia migliore in quanto l'immagine troncata è completamente bianca.

Per il processo di troncamento sono stati necessari in media circa **0.0055** secondi.

7 Conclusioni

In questo progetto sono stati analizzati diversi criteri per la selezione del numero di valori singolari da mantenere nella SVD troncata di una matrice di differenza tra due immagini e diverse soglie per la binarizzazione dell'immagine risultante.

Criterio	Valore di k	Errore assoluto $\ \cdot \ _2$	Errore assoluto in $\ \cdot \ _F$	Tempo	Memoria utilizzata	Soglia
Guttman-Keiser	301	0	0	0.0018s	$301 \times 301 + 301 + 301 \times 301 = 181.503$	0.25 oppure 0.020784 (automatica)
Screenplot di Cottel	7	907.43406	6782.450233	0.07s	$7 \times 301 + 7 + 301 \times 7 = 4.221$	0.25 simile all'originale, 0.39216 (automatica) simile all'approssimazione
Entropia	116	384.477333	2892.327572	0.0009s	$301 \times 116 + 116 + 301 \times 116 = 69.948$	0.25 oppure 0.26275 (automatica)
Energia	94	442.680374	3487.925592	0.006s	$301 \times 94 + 94 + 301 \times 94 = 56.682$	0.25 oppure 0.30196 (automatica)
K-Means Isolation Forest (log)	225	141.942286	723.122392	0.35s	$301 \times 225 + 225 + 301 \times 225 = 135.675$	0.25 oppure 0.24706 (automatica)
K-Means Isolation Forest	1	1742.857352	7454.849653	0.0055s	$301 \times 1 + 1 + 301 \times 1 = 603$	/

Tabella 13: Risultati

Per il calcolo della memoria utilizzata si è tenuto conto delle dimensioni delle matrici U, S e V troncate, in particolare di S si tiene conto della diagonale.

Il criterio di Guttman-Keiser non si è rivelato utili in questo campo in quanto ha sovrastimato di molto il numero di valori singolari da mantenere, in questo caso specifico 301, che è il numero massimo possibile.

Il criterio di Cottel, essendo basato sullo screenplot e sulla scelta soggettiva del gomito, in questo caso ha sottostimato di molto il numero di valori singolari da mantenere, in questo caso 7.

Gli errori di approssimazione sono molto alti e il tempo di esecuzione è anche maggiore rispetto ad altri criteri che portano a errori minori. In questo caso una soglia più bassa scelta arbitrariamente (e non automaticamente) ha portato ad avere risultati più simili all'immagine originale che a quella troncata.

Il criterio del K-Means isolation forest con logaritmo è il secondo per memoria occupata e primo per tempo di esecuzione impiegato. Sono stati selezionati 225 valori singolari. Senza logaritmo risulta un metodo di fatto inutile. Due criteri molto simili dal punto di vista di errori, tempo di esecuzione e memoria sono Entropia ed Energia. In particolare quello dell'energia ha richiesto meno tempo, occupato meno memoria al costo di un errore leggermente superiore, rivelandosi di fatto il miglior criterio.

Possiamo infine notare come in genere soglie molto basse abbiano portato a risultati migliori. In particolare la soglia 0.25 è risultata essere sempre molto simile a quella calcolata automaticamente, il che potrebbe portare vantaggi (si pensi a matrici molto più grandi in cui l'algoritmo di graythresh potrebbe impiegare molto tempo e ottenere risultati molto vicini a 0.25).