Politecnico di Milano
Scuola di Ingegneria Industriale e dell'Informazione
Laurea Magistrale in Ingegneria Matematica

Wasserstein K-means per clustering di misure di probabilità e applicazioni

Relatore: Prof. Federico Bassetti

Laurea Magistrale di: Riccardo Taffoni
ID: 898787

Anno accademico 2019-2020

Ringraziamenti

Vorrei ringraziare tutte le persone che mi sono state accanto tutti questi anni al Politecnico che hanno fatto sì che potessi raggiungere questo traguardo.

In particolare ringrazio il professor Federico Bassetti, che mi ha fatto scoprire questo interessante argomento e mi ha accompagnato in questi mesi alla produzione e scrittura di questo lavoro.

Ringrazio la mia famiglia per il sostegno morale ed economico ed aver creduto sempre in me.

Grazie a Federica per essermi stata accanto in questo anno e soprattutto negli ultimi mesi.

Un grazie a tutti gli amici con cui ho condiviso lo studio e non solo, per essermi stati di ispirazione.

Indice

Introduzione

Negli ultimi anni, l'avvento del Machine Learning ha portato con sè tante scoperte scientifiche. Un importante contributo, in particolare, è stato dato dalla scoperta di nuovi metodi e modelli per l'analisi dati.
Uno tecnica importante Machine Learning è il Clustering, oggetto del presente elaborato. Dal momento che tale settore è molto vario e applicabile a diversi ambiti (come quello economico, sanitario, ambientale etc.), i dati analizzabili attraverso questi metodi sono molteplici. La varietà e la numerosità dei campi di applicazione del clustering comportano la costante introduzione di nuovi algoritmi, alcuni dei quali trattano determinate tipologie di dati come misure di probabilità o istogrammi, e proprio per la sua versatilità, questo tipo di clustering è stato oggetto di svariati studi, tra cui quelli trattati in [9] ed in [6].
Il presente elaborato tratta, tuttavia, un approccio diverso, nel quale viene utilizzata la distanza di Wasserstein per effettuare clustering di dati in forma di istogrammi. Infatti la distanza di Wasserstein permette di calcolare una distanza fra misure di probabilità. In particolare in questa tesi analizzeremo come la distanza di Wasserstein possa essere utilizzata come distanza base in un algoritmo di tipo K-means. In ultima istanza verrà data prova del funzionamento dell'algoritmo con dati sia artificiali che reali riguardanti l'inquinamento provacato da alcune sostanze chimiche in Lombardia.

Capitolo 1

Distanza di Kantorovich e Wasserstein

Il Trasporto ottimo è un campo di studi matematici ancora oggi attivo, i cui primi studi furono fatti Gaspard Monge (1781)[10] e poi portati avanti da Leonid Kantorovich (1942)[7]. Questo campo è nato dall'esigenza di risolvere un problema pratico affidato a Monge, ovvero quello di capire quanto terreno spostare da un luogo prestabilito ad un altro, cercando di minimizzare il costo che il trasporto implica. Da qui, prima Monge, poi Kantorovich, diedero un contributo importante a questo ramo della matematica. Oggi, la teoria del trasporto ottimo è adoperata in moltissimi ambiti, da quello economico, al Computer science, fino alla matematica applicata.

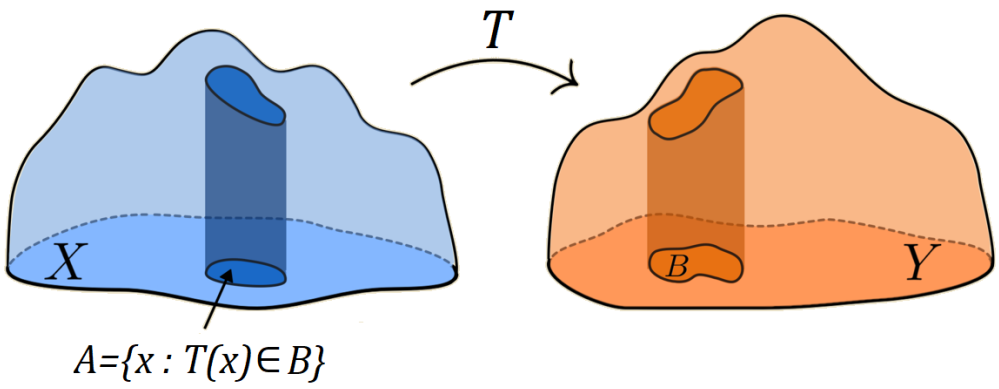


Figura 1.1: Esempio di trasporto ottimo preso da [12]

La Figura 1.1 riporta un esempio che mostra come funzioni il trasporto

da un insieme A ad un insieme B tramite la funzione di trasporto T.
Il problema di trasporto ottimo che si andrà ad analizzare è quello del problema di Kantorovich, in cui la teoria sopracitata si applica a due misure di probabilità.
In questo capitolo verrà affrontata al livello teorico la costruzione della formula di questa distanza, arrivando poi a formulare la distanza di Wasserstein, sia nel caso generale che nel caso discreto.

1.1 Distanza nel caso generale

Il problema di Kantorovich, come appena accennato, calcola la distanza tra due misure di probabilità. Se ne riporta la definizione matematica.
Siano $(X, \mathcal{X})$ e $(Y, \mathcal{Y})$ due spazi polacchi, ovvero spazi metrici separabili e completi. Si considerino due misure di probabilità P e Q appartenenti all'insieme delle misure di probabilità su X e su Y rispettivamente indicate con $\mathcal{P}(X)$ e $\mathcal{P}(Y)$. Si definisce l'insieme

$$\Pi(P, Q) := \{\pi \in \mathcal{P}(X \times Y) : \pi(X \times \cdot) = P(\cdot), \pi(\cdot \times Y) = Q(\cdot)\} \tag{1.1}$$

ovvero la classe di misure di probabilità su $(X \times Y, \mathcal{X} \times \mathcal{Y})$ con marginali P e Q. Gli elementi π di questa classe vengono chiamati piani di trasferimento. Tutti gli elementi di Π sono misure di probabilità che possono essere interpretate come indicatori di come trasferire la "massa" da trasferire di P in Q. Ma spostare questa "massa" ha un costo, che viene definito dalla funzione di costo $c(\cdot, \cdot) : X \times Y \to \mathbb{R}$. Dopo di che, fissato un piano di trasferimento $\pi \in \Pi$ si definisce

$$I_c(\pi) := \int_{X \times Y} c(x, y)\pi(dxdy) \tag{1.2}$$

che rappresenta il costo medio per trasferire P in Q, associato a quel piano di trasferimento. Possiamo quindi definire il problema di Kantorovich nel seguente modo:

$$K_c(P, Q) := \inf_{\pi \in \Pi} \int_{X \times Y} c(x, y)\pi(dxdy) \tag{1.3}$$

In sostanza, quindi , questo problema ha il fine di trovare il minimo costo medio per effettuare uno spostamento da una misura P ad una misura Q.
La seguente proposizione, dimostrata in [12], ci garantisce che esiste un piano ottimo π^* che minimizza I_c per cui $I_c(\pi^*) = K_c(P, Q)$:

Proposizione 1. *Siano $P \in \mathcal{P}(X), Q \in \mathcal{P}(Y)$ con X, Y spazi polacchi e assumiamo $c : X \times Y \to [0, +\infty)$ semicontinua inferiormente. Allora esiste $\pi^* \in \Pi(P, Q)$ che minimizza la (1.2) tra tutti i $\pi \in \Pi(P, Q)$*

Definito il problema di Kantorovich, assumiamo di essere nel caso in cui $X = Y$ e sia d una distanza su X. Allora è possibile definire la distanza di Wasserstein di ordine p W_p nel seguente modo:

$$W_p(P,Q) := K_{d^p}(P,Q)^{1\wedge 1/p} = (\inf_{\pi\in\Pi(P,Q)} \int_{X^2} d^p(x,y)\pi(dxdy))^{1\wedge 1/p} \qquad (1.4)$$

La differenza che c'è tra la definizione di Kantorovich e quella di Wasserstein è la definizione della funzione di costo, che viene definita da un distanza $d(x,y)$ elevata alla p , ordine della distanza di Wasserstein. Proprio questa definizione di funzione di costo ci garantisce, tramite la seguente proposizione dimostrata ad esempio in [11], che la W_p sia una distanza.

Proposizione 2. *Assumiamo $X = Y$ e che per p>1, $c(x,y) = d(x,y)^p$, dove d è una distanza su X, cioè*

1. $d(x,y) = d(y,x) \geq 0$;
2. $d(x,y) = 0$ *se solo se x=y;*
3. $\forall(x,y,z) \in X^3, d(x,z) \leq d(x,y) + d(y,z)$.

Allora la W_p definito in (1.4) *è una distanza, ossia è simmetrica, positiva, $W_p(P,Q) = 0$ se solo se $P = Q$ e soddisfa la disuguaglianza triangolare*

$$\forall(P,Q,Z) \in \mathcal{P}(X)^3 \quad W_p(P,Z) \leq W_p(P,Q) + W_p(Q,Z)$$

1.2 Distanza nel caso discreto

Oltre alla formulazione generale, il problema di Kantorovich può essere espresso anche in forma discreta. Il presente paragrafo intende dunque riprendere quanto detto nel precedente, riscrivendo il problema in un formato matriciale.
Supponiamo che $X = \{x_1, \cdots, x_n\}$ e $Y = \{y_i, \cdots, y_n\}$. In questo caso P e Q possono essere identificati con due vettori di probabilità $\mathbf{a} \in \Sigma_n$ e $\mathbf{b} \in \Sigma_m$, dove $\Sigma_n = \{\mathbf{a} \in \mathbb{R}_+^n : \sum_{i=1}^n \mathbf{a}_i = 1\}$. Questi vettori di probabilità sono interpretabili come degl'istogrammi, in cui ogni $\mathbf{a}_i$ esprime la probabilità associata ad x_i, mentre $\mathbf{b}_j$ la probabilità associata a y_j . Si può definire l'insieme analogo alla classe di misura di probabilità $\Pi(P,Q)$

$$\mathbf{U}(\mathbf{a},\mathbf{b}) := \{\mathbf{P} \in \mathbb{R}_+^{n\times m} : \mathbf{P}\mathbb{1}_m = \mathbf{a}, \mathbf{P}^T\mathbb{1}_n = \mathbf{b}\} \qquad (1.5)$$

dove $\mathbf{P}$ è la matrice di trasporto o di accoppiamento, l'analogo del piano di trasferimento π nel caso discreto. Perciò $\mathbf{P}_{i,j}$ ci dice quanta "massa" spostare

dal bin i di $\mathbf{a}$ al bin j di $\mathbf{b}$. I vincoli relativi all'insieme $\mathbf{U}(\mathbf{a},\mathbf{b})$ possono essere relazionati a quelli di Π, ovvero che la somma delle righe, o delle colonne, restituisce $\mathbf{a}$, o $\mathbf{b}$, tale che $\mathbf{P}\mathbb{1}_m = \mathbf{a} = \sum_{j=1}^{m} \mathbf{P}_{i,j} = \mathbf{a}_i \quad \forall i \in \{1, \cdots, n\}$ e $\mathbf{P}^T\mathbb{1}_n = \sum_{j=1}^{n} \mathbf{P}_{i,j} = \mathbf{b}_j \quad \forall j \in \{1, \cdots, m\}$.
La funzione di costo c sarà rappresentata da una matrice $\mathbf{C} \in \mathbb{R}^{n\times}$ dove ogni $\mathbf{C}_{i,j} = c(x_i, y_j)$ spiega il costo di passaggio da $\mathbf{a}_i$ ad $\mathbf{b}_j$. Il costo medio analogo a I_c definito in (1.2), nel caso discreto verrà definito come

$$\bar{I}_c = \sum_{i,j} \mathbf{C}_{i,j}\mathbf{P}_{i,j}. \tag{1.6}$$

Si può quindi definire il problema di Kantorovich discreto nel seguente modo:

$$L_C(a, b) := \min_{P\in\mathbf{U(a,b)}} \sum_{i,j} \mathbf{C}_{i,j}\mathbf{P}_{i,j} \tag{1.7}$$

Come nel caso continuo, assumendo che $X, Y \subset E$, con E spazio metrico con distanza d, anche nel caso discreto si può definire la distanza di Wasserstein di ordine p come segue:

$$\bar{W}_p(a, b) := \min_{\mathbf{P}\in\mathbf{U(a,b)}} (\sum_{i,j} \mathbf{D}^p_{i,j}\mathbf{P}_{i,j})^{1\wedge 1/p} \tag{1.8}$$

dove $\mathbf{D}^p_{i,j} = d^p(x_i, y_i)$ è la matrice delle distanze tra i punti x_i e y_j.
Nel caso in cui $X, Y \subseteq \mathbb{R}^n$ un esempio di distanza tra due punti nel caso discreto è $\mathbf{D}_{i,j} = ||x_i - y_j||$, con $||\cdot||$ la norma euclidea.
Nel caso della Wasserstein,considereremo la matrice di costo $\mathbf{C}$=$\mathbf{D}$, con $\mathbf{D}$ la matrice delle distanze euclidee. Perciò la seguente proposizione, si veda ad esempio in [11], ci assicura che la W_p sia una distanza tra 2 misure di probabilità o istogrammi.

Proposizione 3. *Supponiamo n=m e che per* $p \geq 1, \boldsymbol{C} = \boldsymbol{D}^p = (\boldsymbol{D}^p_{i,j})_{i,j} \in \mathbb{R}^{n\times n}$ *e dove* $\boldsymbol{D} \in \mathbb{R}^{n\times n}_+$ *è una distanza, cioè*

1. $\boldsymbol{D} \in R^{n\times n}_+$ *è simmetrico;*
2. $\boldsymbol{D}_{i,j} = 0$ *se solo se* $i = j$*;*
3. $\forall(i, j, k) \in 1 \cdots n^3, \boldsymbol{D}_{i,k} \leq \boldsymbol{D}_{i,j} + \boldsymbol{D}_{j,k}$.

Allora la distanza $\hat{W}_p$ *definito in* (1.8) *è simmetrica, positiva,*$\hat{W}_p(\boldsymbol{a}, \boldsymbol{b}) = 0$ *se solo se* $\boldsymbol{a}$=$\boldsymbol{b}$ *e soddisfa la disuguaglianza triangolare*

$$\forall(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \in \Sigma_n \quad \hat{W}_p(\boldsymbol{a}, \boldsymbol{c}) \leq \hat{W}_p(\boldsymbol{a}, \boldsymbol{b}) + \hat{W}_p(\boldsymbol{b}, \boldsymbol{c})$$

.

Capitolo 2

La distanza di Wasserstein nelle applicazioni

La distanza di Wasserstein, trattata nel capitolo precedente, viene utilizzata in varie applicazioni a seconda dei casi, con situazioni che sono monodimensionali e multidimensionali. Questi due casi sono molto differenti tra loro e nelle prossime sezioni verrano spiegate le loro differenze e le loro peculiarità.

2.1 Caso Monodimensionale

In questa sezione mostreremo come viene riformulata la Wassertsein nel caso monodimensionale. Per fare ciò è necessario definire il concetto di funzione di ripartizione.

2.1.1 Funzione di ripartizione

Per la definizione di alcune distanze verrà utilizzata la funzione di ripartizione, la cui definizione è la seguente. Dato uno spazio di probabilità (Ω, M, P) ed una variabile aleatoria X, fissato un $x \in \mathbb{R}$, la funzione di ripartizione F: $\mathbb{R} \to [0, 1]$ è definita come:

$$F(x) := P(X \leq x). \tag{2.1}$$

Possiamo definire la funzione di ripartizione inversa F^{-1} come

$$F^{-1}(q) = \inf\{x : F(x) = q, 0 \leq q \leq 1\} \tag{2.2}$$

Nelle applicazioni, è utile definire anche la approssimazione empirica di F, ovvero la funzione di ripartizione empirica $\bar{F}$ nel seguente modo. Date n

variabili aleatorie $X_i ... X_n$ indipendenti identicamente distribuite (i.i.d). con comune funzione di ripartizione F e fissato $x \in \mathbb{R}$, allora la funzione di ripartizione empirica è definita da:

$$\bar{F}_n(x) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{X_i \leq x\}}, \tag{2.3}$$

dove $\mathbb{1}_{\{X_i \leq x\}} = 1$ se $X_i \leq x$, 0 altrimenti. Questo è il naturale stimatore della funzione di ripartizione F. Inoltre questo stimatore ha importanti propietà di convergenza, prese da [3]:

- per un punto fissato di $x \in \mathbb{R}$, la quantità $\bar{F}_n(x) \sim Bi(n, F(x))$. Perciò la media e la varianza saranno le seguenti:

  $$\mathbb{E}[\bar{F}_n(x)] = F(x) \quad Var(\bar{F}_n(x)) = \frac{F(x)(1 - F(x))}{n}.$$

  Inoltre, grazie alla disuguaglianza di Chebyshev abbiamo

  $$P(|\bar{F}_n(x) - F(x)| \geq \epsilon) \leq \frac{F(x)(1 - F(x))}{n\epsilon}$$

  il quale implica che $\bar{F}_n(x)$ converge in probabilità ad $F(x)$ con $n \to +\infty$. Tuttavia questo risultato implica anche la legge dei grandi numeri, ovvero $\bar{F}_n(x)$ converge ad F(x) quasi ovunque per ogni $x \in \mathbb{R}$ fissata.

- Il teorema di Glivenko-Cantelli implica un risultato di convergenza più forte, ovvero

  $$\sup_{x \in \mathbb{R}} |\bar{F}_n(x) - F(x)| \xrightarrow{q.o.} 0$$

2.1.2 Distanza tra due funzioni di ripartizioni

Nel caso in cui l'obiettivo è quello di misurare la distanza dei punti appartententi a $\mathbb{R}^n$, si può considerare una distanza tra quelle di uso comune, come quella Euclidea o quella di Manhattan. Nelle applicazioni che vedremo, tuttavia non si considererà la distanza tra due punti, bensì tra due misure di probabilità e risulterà utile utilizzare le distanze introdotte nelle sezioni precedenti.
Richiamando la definizione di distanza Wasserstein di ordine p (1.4), nel caso in cui $E = \mathbb{R}$ e $P, Q \in \mathcal{P}(\mathbb{R})$ e definendo la distanza $d(x, y) = |x - y|$, la W_p nel caso monodimensionale, detta anche distanza di Mallow di ordine p, risulta definita da:

$$W_p(P, Q) = \inf_{\pi \in \Pi(P,Q)} \left(\int_E |x - y|^p \pi(dxdy) \right)^{1 \wedge 1/p} \tag{2.4}$$

Nel nostro caso, considereremo la distanza di Wasserstein con p=2, che può essere semplificata riconducendola alla distanza tra quantili come espresso nella seguente proposizione

Proposizione 4. *Date due variabili aleatorie X e Y , con relative funzioni di ripartizioni F_X e F_Y e leggi P_X e P_Y, allora vale la seguente uguaglianza:*

$$W_2(P_X, P_Y) = \sqrt{\int_0^1 (F_X^{-1}(q) - F_Y^{-1}(q))^2 dq}. \tag{2.5}$$

Inoltre definiremo

$$d_{F^{-1}}(F_X, F_Y) := \sqrt{\int_0^1 (F_X^{-1}(q) - F_Y^{-1}(q))^2 dq} \tag{2.6}$$

Assumendo di non conoscere le vere espressioni di F_X e F_Y ma di avere a disposizione n variabili $X_1, ..X_n$ i.i.d. con funzione di ripartizione comune F_X ed m variabili $Y_1, ..Y_m$ i.i.d. con funzione di ripartizione comune F_Y, possiamo ricavare le funzioni di ripartizione empiriche $\bar{F}_X$ $\bar{F}_Y$ da cui si può approssimare la (2.6) con la distanza fra le funzioni di ripartizione empiriche:

$$d_{F^{-1}}(\bar{F}_X, \bar{F}_Y) := \sqrt{\int_0^1 (\bar{F}_X^{-1}(q) - \bar{F}_Y^{-1}(q))^2 dq} \tag{2.7}$$

Un'altra distanza che si può prendere in considerazione come alternativa alla Wasserstein è la norma L_2 fra le funzioni di ripartizioni, ossia

$$d_F(F_X, F_y) := \int_{\mathbb{R}} (F_X(t) - F_Y(t))^2 dt. \tag{2.8}$$

Nel caso empirico diventa

$$d_F(\bar{F}_X, \bar{F}_y) := \int_{\mathbb{R}} (\bar{F}_X(t) - \bar{F}_Y(t))^2 dt. \tag{2.9}$$

2.2 Caso Multidimensionale

A differenza del caso monodimensionale, in quello multidimensionale non si hanno espressioni semplici della distanza, come quella riportata nella Proposizione 4.

2.2.1 Distanza di Wasserstein con fattore di regolarizzazione entropico

Per trovare la distanza tra due misure di probabilità bisogna ricondursi all'equazione della distanza di Wasserstein (1.4), già trattata nelle sezioni precedenti :

$$W_p(P,Q) := K_{d^p}(P,Q)^{1\wedge 1/p} = inf_{\pi\in\Pi(P,Q)}(\int_{E^2} d^p(x,y)\pi(dxdy))^{1\wedge 1/p}$$

In particolare ci soffermiamo alla sua versione discreta già definita nell'equazione (1.8), che qui riportiamo:

$$\bar{W}_p(a,b) := \min_{\mathbf{P}\in\mathbf{U(a,b)}} (\sum_{i,j} \mathbf{D}^p_{i,j}\mathbf{P}_{i,j})^{1\wedge 1/p} \tag{2.10}$$

Questo problema è tipico della programmazione lineare e può essere risolto con qualunque algoritmo di settore, ad esempio l'algoritmo del simplesso, descritto in [4], ma applicarlo in questa formulazione avrebbe un costo computazionale molto alto (nel caso peggiore la complessità è esponenziale).
Perciò per ridurre il costo computazionale si può approssimare la distanza di Wasserstein aggiungendo un fattore di regolarizzazione alla formula originale. Questa regolarizzazione ha vantaggi importanti, che permettono di risolvere il problema di minimizzazione regalirizzato usando un semplice schema basato su prodotti matrice-vettore. I risultati scritti in seguito sono presi dall'articolo [11].
L'entropia di una matrice di trasporto è definita nel seguente modo:

$$H(\mathbf{P}) := -\sum_{i,j} \mathbf{P}_{i,j}(\log(\mathbf{P}_{i,j}) - 1) \tag{2.11}$$

con la convenzione che $H(\mathbf{a}) = -\infty$ se esiste un j per cui $\mathbf{a}_j$ è 0 o negativo. La funzione H è strettamente concava, perchè la sua matrice Hessiana è $\partial^2 H(\mathbf{P}) = -diag(1/\mathbf{P}_{i,j})$ e $\mathbf{P}_{i,j} \leq 1$. Lo scopo della regolarizzazione entropica è di usare -H come funzione regolarizzante per ottene la seguente approssimazione della distanza di Wasserstein:

$$L^{\epsilon}_{\mathbf{C}}(\mathbf{a},\mathbf{b}) := \min_{\mathbf{P}\in\mathbf{U(a,b)}} \sum_{i,j} \mathbf{P}_{i,j}\mathbf{C}_{i,j} - \epsilon H(\mathbf{P}) \tag{2.12}$$

Un risultato importante è dato dalla seguente proposizione, che ci mostra come la soluzione del problema regolarizzato $\mathbf{P}_\epsilon$ converga al tendere di ϵ a 0 alla soluzione del problema di Kantorovich $\mathbf{P}$.

Proposizione 5. *L'unica soluzione* $\boldsymbol{P}_\epsilon$ *di* (2.12) *converge alla soluzione ottimale con massima entropia tra tutti gli insiemi di soluzione ottima del problema di Kantorovich, ovvero*

$$\boldsymbol{P}_\epsilon \xrightarrow{\epsilon\to 0} \arg\min_{\boldsymbol{P}}\{-H(\boldsymbol{P}) : \boldsymbol{P} \in \boldsymbol{U}(\boldsymbol{a},\boldsymbol{b}), \sum_{i,j} \boldsymbol{P}_{i,j}\boldsymbol{C}_{i,j} = L_{\boldsymbol{C}}(\boldsymbol{a},\boldsymbol{b})\} \tag{2.13}$$

in particolare

$$L^\epsilon_{\boldsymbol{P}}(\boldsymbol{a},\boldsymbol{b}) \xrightarrow{\epsilon\to 0} L_{\boldsymbol{P}}(\boldsymbol{a},\boldsymbol{b})$$

Dimostrazione. Si consideri una sequenza $(\epsilon_l)_l$ tale che $\epsilon_l \to 0$ e $\epsilon_l > 0$. Si denota $\mathbf{P}_l$ soluzione della (2.12) per $\epsilon = \epsilon_l$. Siccome $\mathbf{U}(\mathbf{a},\mathbf{b})$ è limitato, allora si può estrarre una sequenza (che non verrà rieticchetato per semplicità) tale che per $\mathbf{P}_l \to \mathbf{P}^*$. Siccome $\mathbf{U}(\mathbf{a},\mathbf{b})$ è chiuso, $\mathbf{P}^* \in \mathbf{U}(\mathbf{a},\mathbf{b})$. Si consideri $\mathbf{P}$ tale che $\langle \mathbf{P},\mathbf{C}\rangle = L_{\mathbf{C}}(\mathbf{a},\mathbf{b})$. Per ottimalità di $\mathbf{P}$ e $\mathbf{P}_l$, per i loro rispettivi problemi di ottimizzazione (per $\epsilon = 0$ e $\epsilon = \epsilon_l$), si ha:

$$0 \le \langle \mathbf{P}_l,\mathbf{C}\rangle - \langle \mathbf{P},\mathbf{C}\rangle \le \epsilon_l(H(\mathbf{P}_l) - \mathbf{P}). \tag{2.14}$$

Siccome H è continuo, prendendo il limite $l \to +\infty$ in questa espressione si mostra che $\sum_{i,j}\mathbf{P}^*_{i,j}\mathbf{C}_{i,j} = \sum_{i,j}\mathbf{P}_{i,j}\mathbf{C}_{i,j}$ così che $\mathbf{P}^*$ è ammissibile per (2.13). Inoltre, dividendo per ϵ_l in (2.14) e facendo il limite si mostra che $H(\mathbf{P}) \le H(\mathbf{P}^*)$ il quale mostra $\mathbf{P}^*$ è la soluzione. Grazie alla stretta convessità di -H, la soluzione è anche unica, e l'intera sequenza converge. □

La formula (2.13) afferma quindi che più ϵ va a 0 più la soluzione tende alla massima entropia.
Per scrivere la soluzione in modo tale che sia adattabile all'algoritmo iterativo del Sinkhorn che andremo a descrivere nella prossima sezione, sfruttiamo i seguenti risultati. Si definisce la divergenza di Kullback-Leibler come

$$\mathbf{KL}(\mathbf{P}|\mathbf{K}) := \sum_{i,j}\mathbf{P}_{i,j}\log(\frac{\mathbf{P}_{i,j}}{\mathbf{K}_{i,j}}) - \mathbf{P}_{i,j} + \mathbf{K}_{i,j}, \tag{2.15}$$

allora l'unica soluzione $\mathbf{P}_\epsilon$ può essere vista come la proiezione su $\mathbf{U}(\mathbf{a},\mathbf{b})$ del Gibbs kernel associata alla matrice di costo $\mathbf{C}$, $\mathbf{K}_{i,j} := e^{-\frac{\mathbf{C}_{i,j}}{\epsilon}}$. Infatti si ha che

$$\mathbf{P}_\epsilon = \mathbf{Proj}^{\mathbf{KL}}_{U(\mathbf{a},\mathbf{b})(\mathbf{K})} := \arg\min_{\mathbf{P}\in U(\mathbf{a},\mathbf{b})} \mathbf{KL}(\mathbf{P}|\mathbf{K}). \tag{2.16}$$

Un risultato utile per la creazione dell'algoritmo è il seguente:

Proposizione 6. *La soluzione della* (2.12) *è unica ed è della forma :*

$$\boldsymbol{P}_{i,j} = \boldsymbol{u}_i\boldsymbol{K}_{i,j}\boldsymbol{v}_j \quad \forall i \in \{1...n\}, \forall j \in \{1...m\} \tag{2.17}$$

con $(u,v) \in \mathbb{R}^n_+ \times \mathbb{R}^m_+$.

Dimostrazione. Considerando due variabili $\mathbf{f} \in \mathbb{R}^n, \mathbf{g} \in \mathbb{R}^m$, la Lagrangiana della (2.12)

$$L(\mathbf{P}, \mathbf{f}, \mathbf{g}) = \langle \mathbf{C}, \mathbf{P} \rangle - \epsilon H(\mathbf{P}) - \langle \mathbf{f}, \mathbf{P}\mathbb{1}_m - \mathbf{a} \rangle - \langle \mathbf{g}, \mathbf{P}\mathbb{1}_n - \mathbf{b} \rangle.$$

Applicando la derivata prima rispetto e ponendola uguale a 0 si ottiene $\mathbf{P}_{i,j}$

$$\frac{\partial L(\mathbf{P}, \mathbf{f}, \mathbf{g})}{\mathbf{P}_{i,j}} = \mathbf{C}_{i,j} + \epsilon \log(\mathbf{P}_{i,j}) - \mathbf{f}_i - \mathbf{g}_j = 0.$$

Risolvendo l'equazione il risultato è $\mathbf{P}_{i,j} = e^{\frac{\mathbf{f}_i}{\epsilon}} e^{\frac{-\mathbf{C}_{i,j}}{\epsilon}} e^{\frac{\mathbf{g}_j}{\epsilon}}$ □

2.2.2 Distanza di Wasserstein con l'algoritmo del Sinkhorn

L'approssimazione entropica spiegata nella sezione precedente, come già accennato, permette di definire un algoritmo che si basa su prodotti matrice-vettore. In particolare è possibile sfruttare la soluzione (2.17) e riscriverla in modo tale che si possa risolvere in modo iterativo.
Perciò riscriviamo la (2.17) come $\mathbf{P}$=diag($\mathbf{u}$)$\mathbf{K}$diag($\mathbf{v}$) ed i vincoli relativi ad U($\mathbf{a}$,$\mathbf{b}$), che è definito da (1.5), come:

$$diag(\mathbf{u})\mathbf{K}diag(\mathbf{v})\mathbb{1}_m = \mathbf{a}$$

$$diag(\mathbf{v})\mathbf{K}^T diag(\mathbf{u})\mathbb{1}_m = \mathbf{b}.$$

Queste due condizioni servono per rispettare i vincoli dati dalle distribuzioni marginali. Esse possono essere semplificate ulteriormente, riscrivendole come:

$$\mathbf{u} \odot (\mathbf{K}\mathbf{v}) = \mathbf{a}$$
$$\mathbf{v} \odot (\mathbf{K}^T\mathbf{u}) = \mathbf{b},$$

dove il simbolo $\odot$ corrisponde al prodotto di Hadamard. Questo problema è conosciuto in analisi numerica come problema di ridimensionamento della matrice ([Nemirovski and Rothblum, 1999]). Grazie a questi risultati, ad un iterazione generica l dell'algoritmo del Sinkhorn vengono fatti i seguenti aggiornamenti:

$$\mathbf{u}^{(l+1)} := \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}^{(l)}} \quad \mathbf{v}^{(l+1)} := \frac{\mathbf{b}}{\mathbf{K}^T\mathbf{u}^{(l+1)}}, \tag{2.18}$$

inizializzando $\mathbf{v}^{(0)} = \mathbb{1}_m$

Capitolo 3

Baricentro

Il baricentro è un concetto essenziale in molti campi scientifici, da quello fisico a quello statistico. In fisica questo viene spesso chiamato centro di massa, poichè viene inteso come il punto di un corpo avente la stessa massa del medesimo, a cui il corpo stesso può essere approssimato. Tale concetto può essere espresso in modo più generale sostituendo al corpo un insieme qualsiasi; il baricentro, quindi, andrebbe a sostiuire l'insieme stesso.
In questa sezione verrà illustrata la formula matematica del baricentro, detto anche centroide, definendo ed elencando le sue caratteristiche.
Possiamo definire il baricentro di un insieme $Z \subset \mathbb{R}^d$ secondo questa formula:

$$C := \frac{\int_Z zg(z)dz}{\int_Z g(z)dz},$$

dove $g : Z \to \mathbb{R}$ esprime la densità della "massa" in un certo punto z. Si può dare anche un formulazione discreta del problema, in cui si considerano un insieme di n punti $z_i, \cdots z_n \in \mathbb{R}^d$ con le relative "masse" $c_1, \cdots, c_n \in \mathbb{R}$, allora il baricentro discreto si può scrivere come

$$\hat{C} := \frac{\sum_{i=1}^n c_i z_i}{\sum_{i=1}^n c_i}. \tag{3.1}$$

Un caso particolare di (3.1) è quando l'insieme degli c hanno tutti lo stesso valore $c_i = 1/n$, in questo caso lo possiamo riscrivere come:

$$\bar{z} := \frac{1}{n}\sum_{i=1}^n z_i \tag{3.2}$$

Quindi in generale possiamo interpretare questo valore come un valore medio di un certo insieme. In particolare una proprietà importante del baricentro

è che esso è il valore che minimizza z $\mapsto \sum_{i=1}^{m} ||z^i - z||^2$ dove la norma $||\cdot||$ viene intesa come quella euclidea. Questo risultato viene assicurato dalla seguente proposizione.

Proposizione 7. *Si considerino i punti $z_1, z_2, ..., z_m$, dove $m \geq 1$ e per $i \in \{1, 2, ..., m\}$, $z_i \in \mathbb{R}^d$. Sia $\bar{z} = \frac{1}{m}\sum_{i=1}^{m} z_i$ la media di questi punti, allora $\forall z \in \mathbb{R}^d$ un punto casuale nello stesso spazio, allora*

$$\sum_{i=1}^{m} ||z^i - z||^2 \geq \sum_{i=1}^{m} ||z^i - \bar{z}||^2 \tag{3.3}$$

Se ne riporta qui di seguito la dimostrazione:

$$\begin{aligned}
\sum_{i=1}^{m} ||z^i - z||^2 &= \sum_{i=1}^{m} ||(z^i - \bar{z}) + (\bar{z} - z)||^2 \\
&= \sum_{i=1}^{m} (||z^i - \bar{z}||^2 + ||\bar{z} - z||^2 + 2(z^i - \bar{z}) \cdot (\bar{z} - z)) \\
&= \sum_{i=1}^{m} ||z^i - \bar{z}||^2 + \sum_{i=1}^{m} ||\bar{z} - z||^2 + 2\sum_{i=1}^{m} (z^i \cdot \bar{z} - z^i \cdot z - \bar{z} \cdot \bar{z} + \bar{z} \cdot z) \\
&= \sum_{i=1}^{m} ||z^i - \bar{z}||^2 + m||\bar{z} - z||^2 + 2(mz\bar{z}z - mz^i \cdot z - m\bar{z} \cdot \bar{z} + m\bar{z} \cdot z) \\
&= \sum_{i=1}^{m} ||z^i - \bar{z}||^2 + m||\bar{z} - z||^2 \\
&\geq \sum_{i=1}^{m} ||z^i - \bar{z}||^2
\end{aligned} \tag{3.4}$$

Nelle applicazioni che mostreremo nelle prossime sezioni, gli z_i non saranno in punti appartenenti a $\mathbb{R}^d$, ma saranno delle misure di probabilità o istogrammi. Perciò abbiamo bisogno di una definizione generale di baricentro che vada bene per dei punti di uno spazio metrico. Per fare ciò possiamo ricondurci alla proprietà (3.3), assumendo $z_1, z_2, ..., z_m$ appartenti ad un generico spazio metrico Z con metrica d. Allora chiameremo il baricentro il punto $\bar{z} \in Z$ tale per cui per ogni $z \in Z$ vale la relazione:

$$\sum_{i=1}^{m} d(z^i, z)^2 \geq \sum_{i=1}^{m} d(z^i, \hat{z})^2. \tag{3.5}$$

In particolare vedremo come trovare il baricentro nel caso in cui lo spazio metrico sarà l'insieme delle misure di probabilità e la distanza sarà inizialmente la Wasserstein di ordine 2, poi la distanza d_F.

3.1 Baricentro caso monodimensionale

Per calcolare il baricentro nella metrica di Wasserstein nel caso monodimensionale ci riconduciamo alle distanze usate nella Sezione 2.1. I baricentri che prenderemo in considerazione saranno relativi a un insieme di misure di probabilità con ognuno una sua funzione di ripartizione. Perciò, sfruttando i risultati appena elencati in Sezione 3, definiamo i baricentri nel caso della Wasserstein di ordine 2 e della distanza delle funzioni di ripartizioni al quadrato.
Date $F_1 ... F_N$ funzioni di ripartizioni, allora il baricentro $\bar{F}_{inv}$, che soddisfa la (3.5) nel caso in cui si considera la distanza $d_{F^{-1}}$, o W_2, è definita come quella misura di probabilità per cui il suo quantile vale:

$$\bar{F}^{-1}_{inv}(q) := \frac{1}{N}\sum_{i=1}^{N} F_i^{-1}(q); \tag{3.6}$$

$$q \in [0,1]. \tag{3.7}$$

Mentre il baricentro, che chiameremo $\bar{F}$, che soddisfa la (3.5) nel caso in cui si consideri la distanza d_F, è definita come la misura di probabilità per cui:

$$\bar{F}(t) := \frac{1}{N}\sum_{i=1}^{N} F_i(t) \tag{3.8}$$

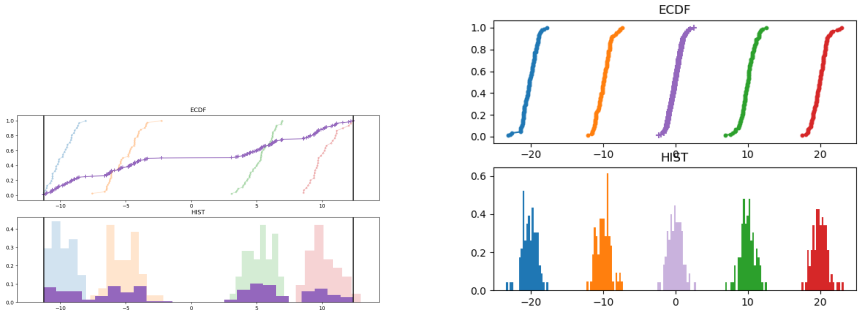
$$t \in \mathbb{R} \tag{3.9}$$

Entrambi i risultati si possono dimostrare sfruttando l'equazione (**??**). In particolare per il baricentro (3.6) possiamo dire che, data una qualunque

funzione di ripartizione G, abbiamo il seguente risultato:

$$\begin{aligned}
\sum_{i=1}^{N} d_{F^{-1}}(F_i, G) &= \sum_{i=1}^{N} \int_0^1 |F_i^{-1}(q) - G^{-1}(q)|^2 dq \\
&= \int_0^1 \sum_{i=1}^{N} |F_i^{-1}(q) - G^{-1}(q)|^2 dq \\
&\geq \int_0^1 \sum_{i=1}^{N} |F_i^{-1}(q) - \bar{F}_{inv}^{-1}(q)|^2 dq \qquad \text{Per la (3.5)} \qquad (3.10) \\
&= \sum_{i=1}^{N} \int_0^1 |F_i^{-1}(q) - \bar{F}_{inv}^{-1}(q)|^2 dq \\
&= \sum_{i=1}^{N} d_{F^{-1}}(F_i, \bar{F}_{inv}^{-1})
\end{aligned}$$

La stessa dimostrazione può essere fatta per il baricentro (3.8), considerando nella dimostazione (3.10) la distanza d_F al posto della $d_{F^{-1}}$. Inoltre si può aggiungere che i baricentri $\bar{F}$ e $\bar{F}_{inv}$ non sono uno l'inverso dell'altro.



(a) (b)

Figura 3.1: Esempio di baricentro (istogramma viola)

In Figura 3.1 mostra 4 istogrammi e le relative funzioni di ripartizione generati da campioni gaussiani con diverse medie e i corrispondenti baricentri. In particolare in (a) viene sfruttata la formula (3.8), mentre in (b) la formula (3.6). Quello che si nota è che il baricentro nel caso (b) tende a posizionarsi come "media" dei valori osservati, mentre nel caso (a), il baricentro è una mistura di distribuzioni di frequenza delle osservazioni.

3.1.1 Calcolo baricentro con fattore di regolarizzazione entropico

Nella Sezione 2.2 abbiamo visto come trovare la distanza tra due misure di probabilità. Ora vogliamo vedere come trovare una misura di probabilità che minimizza la distanza tra un insieme di misure di probabilità.
Date N misure di probabilità $\{\mathbf{b}_i\}_{i=1}^N$ dove $\mathbf{b}_i \in \mathbb{R}^d$, ed i pesi relativi ad ogni $\mathbf{b}_i$, il baricentro di Wasserstein viene calcolato minimizzando

$$\min_{\mathbf{a}\in\Sigma_n} \frac{1}{N}\sum_{i=1}^{N} L_{\mathbf{C}}(\mathbf{a},\mathbf{b}_i). \tag{3.11}$$

Sfruttando le propietà della regolarizzazione entropica e i risultati ricavati nella sezione precedente, la (3.11) può essere riformulata tramite regolarizzazione entropica nel seguente modo:

$$\min_{\mathbf{a}\in\Sigma_n} \frac{1}{N}\sum_{i=1}^{N} L_C^{\epsilon}(\mathbf{a},\mathbf{b}_i) \tag{3.12}$$

che può essere visto come il corrispettivo della (2.12). Un approccio più semplice, come osservato da [2],di riscrivere la (3.12) è quello di sfruttare la divergenza di Kullback-Leibler definita in (2.15)

$$\min_{(\mathbf{P}_i)_i} \{\frac{1}{N}\sum_i \epsilon \mathbf{KL}(\mathbf{P}_i|\mathbf{K}_i) : \forall i, \mathbf{P}_i^T\mathbb{1} = \mathbf{b}_i, \mathbf{P}_1^T\mathbb{1}_1 = \cdots = \mathbf{P}_N\mathbb{1}_N\} \tag{3.13}$$

dove $\mathbf{K}_i := e^{-\frac{C_i}{\epsilon}}$. I vincoli dell'equazione (3.13) evidenziano come la somma delle colonne di ogni $\mathbf{P}_i$ restituisca $\mathbf{b}_i$ mentre la somma delle righe si uguale per ogni i, in particolare che sia uguale al baricentro $\mathbf{a}$.

3.1.2 Calcolo del baricentro con l'algoritmo del Sinkhorn

Come nel caso della distanza, è possibile sfruttare i risultati trovati nella precedente sezione per ricavare gli step necessari per risolvere il baricentro di Wasserstein in modo iterativo sfruttando l'algoritmo del Sinkhorn. In particolare, in modo analogo alla distanza vengono definire le matrici di accoppiamento $\mathbf{P}_i$ per ogni misura di probabilità $\mathbf{b}_i$ come $\mathbf{P}_i$ =diag$(\mathbf{u}_i)\mathbf{K}_i$diag$(\mathbf{v}_i)$ e ad ogni iterazione le variabili vengono aggiornate nel seguente modo:

$$\forall i \in \{1\cdots N\} \quad \mathbf{v}_i^{(l+1)} := \frac{\mathbf{b}_i}{\mathbf{K}_i^T\mathbf{u}_i^{(l)}}, \tag{3.14}$$

$$\forall i \in \{1 \cdots N\} \quad \mathbf{u}_i^{(l+1)} := \frac{\mathbf{a}^{(l+1)}}{\mathbf{K}_i \mathbf{v}_i^{(l+1)}}, \tag{3.15}$$

$$\mathbf{a}^{(l+1)} := \prod_i (\mathbf{K}_i \mathbf{v}_i^{(l+1)})^{\frac{1}{N}}. \tag{3.16}$$

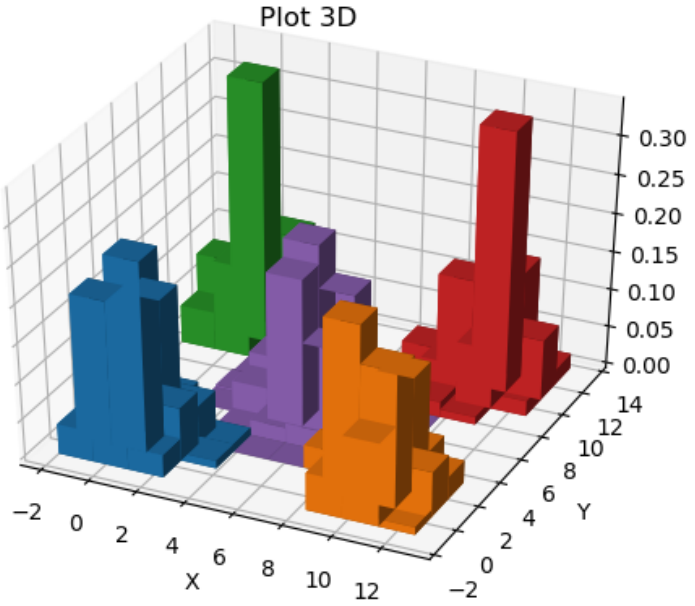


Figura 3.2: Esempio di baricentro in 2 dimensioni(istogramma viola).

In Figura 3.2 abbiamo generato 4 istogrammi derivanti da 4 distribuzioni gaussiane bidimensionali e poi calcolato, tramite l'algoritmo appena spiegato, il baricentro che in figura è rappresentato dall'istogramma viola.

Capitolo 4

Clustering e il K-means

La distanza di Wasserstein, ampiamente trattata in precedenza, trova applicazione nel K-means, uno degli algoritmi più comuni utilizzati per il clustering. Il presente Capitolo si propone di approfondire alcuni concetti: innanzitutto verrò presentato il clustering al livello generale; successivamente ci si focalizzerà sul K-means e sul suo funzionamento, andandone a studiare l'algoritmo, il costo computazionale e come analizzare i risultati dei suoi output. Come ultimo passaggio si tratterà il K-means in relazione alla distanza di Wasserstein, ovvero in che modo l'algoritmo sfrutta la formula della distanza.

4.1 Clustering

Il clustering consiste in un insieme di metodi di analisi multivariata dei dati, volti a raggruppare elementi appartenenti a tali dati in classi omogenee. Queste classi, dette anche cluster, sono insiemi di oggetti che presentano tra loro delle similarità, ma che, contemporaneamente, hanno caratteristiche diverse con oggetti in altri cluster. In molti approcci questa similarità, è concepita in termini di distanza in uno spazio multidimensionale. L'input di un algoritmo di clustering è costituito da un campione di elementi che possono essere sia di tipo categorico, quindi gli elementi sono delle categorie, che di tipo numerico. L'output, invece, è dato da un certo numero di cluster in cui gli elementi del campione sono suddivisi in base alla similarità, ovvero alla distanza. Gli algoritmi di clustering forniscono come output anche la descrizione delle caratteristiche di ciascun cluster, il che è fondamentale per poi prendere decisioni strategiche sulle azioni da compiere verso tali gruppi (marketing mirato, promozioni ad-hoc, creazione di nuovi prodotti/servizi). La risoluzione di questi problemi di clustering è affidata a numerosi modelli

e algoritmi, che vengono scelti ed utilizzati in base alla tipologia di dati posseduti. Tra questi, uno dei più utilizzati per la sua semplicità e adattabilità, è il K-means.

4.2 L'algoritmo del K-means

Il K-means è un algoritmo iterativo di clustering che divide l'insieme dei dati in un K predefinito di cluster, rappresentati da delle "medie" o baricentri, in modo tale da massimizzare la distanza tra i cluster e minimizzare la distanza degli elementi nei cluster. In particolare, dato un insieme di punti $X_1...X_n$ in uno spazio metrico con distanza d, l'obiettivo dell'algoritmo proposto da Lloyd [8] è quello di minimizzare la somma del quadrato delle distanze tra gli elementi ed i baricentri dei cluster, che può essere riscritto come :

$$\min_{\mathbf{B},\mathbf{V}} SS(\mathbf{V},\mathbf{B}) = \min_{\mathbf{B},\mathbf{V}} \sum_{i=1}^{N}\sum_{k=1}^{K} v_{i,k} d^2(X_i, B_k) \tag{4.1}$$

$$s.t. \sum_{j=1}^{K} v_{i,j} = 1 \quad \forall i \in \{1..N\} \tag{4.2}$$

dove $\mathbf{V} = \{v_{i,k}\}$ è una $N \times K$ matrice di partizione, cioè $v_{i,k} = 1$ se l'elemento i-esimo è assegnato al cluster k-esimo, 0 altrimenti, $\mathbf{B} = \{B_1, B_2..., B_k\}$ è l'insieme di baricentri relativi al j-esimo cluster, e $d(\cdot,\cdot)$ è una distanza generica. Il valore minimo di SS può essere approssimato iterativamente risolvendo questi due problemi:

1. Problema 1 (Pb_1): Fissato $\mathbf{B}$=$\hat{\mathbf{B}}$, risolvere il problema ridotto SS($\mathbf{V}$,$\hat{\mathbf{B}}$)
2. Problema 2 (Pb_2): Fissato $\mathbf{V}$=$\hat{\mathbf{V}}$, risolvere il problema ridotto SS($\hat{\mathbf{V}}$,$\mathbf{B}$)

Il Problema 1 può essere risolto facendo:

$$v_{i,l} = 1 \quad d(X_i, B_l) \leq d(X_i, B_m), \quad per \quad 1 \leq m \leq K, \tag{4.3}$$

$$v_{i,m} = 0 \quad per \quad m \neq l. \tag{4.4}$$

Per risolvere il Problema 2, si considera il baricentro tra gli elementi del cluster assegnato secondo la distanza d che si usa. Se prendiamo ad esempio la distanza euclidea, il baricentro è

$$B_j = \frac{\sum_{i=1}^{N} v_{i,j} X_i}{\sum_{i=1}^{N} v_{i,j}} \quad per \quad 1 \leq j \leq K \tag{4.5}$$

Quindi per trovare il minimo di SS viene sfruttato il seguente algoritmo:

1. Si sceglie una $\mathbf{B}_0$ iniziale e si risolve $SS(\mathbf{V}, \mathbf{B}_0)$ per ottenere $\mathbf{V}_0$. Si fissa l'iterazione t=0;

   1.a la $\mathbf{B}_0$ si sceglie casualmente e si risolve $SS(\mathbf{V}, \mathbf{B}_0)$ per ottenere W_0;

   1.b la $\mathbf{B}_0$ si sceglie in modo ottimizzato e si risolve $SS(\mathbf{V}, \mathbf{B}_0)$ per ottenere $\mathbf{V}_0$;

2. Sia $\hat{\mathbf{V}} = \mathbf{V}^t$ si risolve $SS(\hat{\mathbf{V}}, \mathbf{B})$ ottenendo $\mathbf{B}^{t+1}$, se $SS(\hat{\mathbf{V}}, \mathbf{B}^t) = SS(\hat{\mathbf{V}}, \mathbf{B}^{t+1})$ stop,altrimenti si torna allo step 3 ;

3. $\hat{\mathbf{B}} = \mathbf{B}^{t+1}$ e si risolve $SS(\mathbf{V}, \hat{\mathbf{B}})$ per ottenere $\mathbf{V}^{t+1}$, se $SS(\mathbf{V}^t, \hat{\mathbf{B}}) = SS(\mathbf{V}^{t+1}, \hat{\mathbf{B}})$ stop, altrimenti si torna allo step 2 ;

Uno dei grandi vantaggi di questo algoritmo è nella semplicità dei passaggi, che si rispecchia soprattutto nel suo costo computazionale, che è dell'ordine $O(NKt)$, dove N sono gli elementi considerati,t il numero di iterazioni e K il numero dei cluster. Uno degli svantaggi di questo algoritmo però è che non ha un unica soluzione ottima, perciò ripetendo più volte l'algoritmo non è detto che si abbia lo stesso risultato.
Nel caso del Wasserstein K-means, è sufficiente sostituire dalla formula (4.1) la distanza d, con le distanze considerate nel capitolo 1, mentre per risolvere il Problema 2 (Pb_2) è sufficiente considerare i baricentri nel capitolo 3 relativi alla distanza considerata.
Nelle prossime sezione mostreremo come l'algoritmo converge ad un ottimo locale e come viene inizializzato.

4.2.1 Convergenza k-means

Un'altro dei vantaggi dell'algoritmo è che assicura la convergenza della soluzione, anche se solo locale e non globale. Qui di seguito la dimostrazione.
Si supponga che l'algoritmo proceda dall'iterazione t a t+1. Definiamo $SS(\mathbf{V}^t, \mathbf{B}^t) = \sum_{i=1}^{N} \sum_{k=1}^{K} v_{i,j}^t d(X_i, B_i^t)^2$. Basta mostrare che $SS(\mathbf{V}^{t+1}, \mathbf{B}^{t+1}) < SS(\mathbf{V}^t, \mathbf{B}^t)$. Per farlo bastano due step, il primo è che

$$SS(\mathbf{V}^{t+1}, \mathbf{B}^t) < SS(\mathbf{V}^t, \mathbf{B}^t) \tag{4.6}$$

e poi mostrare che

$$SS(\mathbf{V}^{t+1}, \mathbf{B}^{t+1}) \leq SS(\mathbf{V}^{t+1}, \mathbf{B}^t). \tag{4.7}$$

Il primo step segue dalla logica dell'algoritmo ovvero che si passa da $\mathbf{V}^t$ a $\mathbf{V}^{t+1}$ cambia se trova un cluster più vicino a $\mathbf{B}^t$ che uno assegnato a lui da

$\mathbf{V}^t$:

$$\begin{aligned} SS(\mathbf{V}^{t+1}, \mathbf{B}^t) &= \sum_{i=1}^{N}\sum_{k=1}^{K} v_{i,j}^{t+1} d(X_i, B_i^t)^2 \\ &< \sum_{i=1}^{N}\sum_{k=1}^{K} v_{i,j}^{t} d(X_i, B_i^t)^2 \\ &= SS(\mathbf{V}^t, \mathbf{B}^t) \end{aligned} \tag{4.8}$$

Il secondo step si ottiene sfruttando la (3.5):

$$\begin{aligned} SS(\mathbf{V}^{t+1}, \mathbf{B}^{t+1}) &= \sum_{i=1}^{N}\sum_{k=1}^{K} v_{i,j}^{t+1} d(X_i, B_i^{t+1})^2 \\ &\leq \sum_{i=1}^{N}\sum_{k=1}^{K} w_{i,j}^{t+1} d(X_i, B_i^t)^2 \\ &= SS(\mathbf{V}^{t+1}, \mathbf{B}^t). \end{aligned} \tag{4.9}$$

4.2.2 Inizializzazione ottimizzata (k-means++)

In genere vengono condiserati due modi per inizializzare l'algoritmo del k-means:
Random: Alla prima iterazione dell'algoritmo, vengono scelti k baricentri in modo casuale tra le osservazioni;
Ottimizzato: Alla prima iterazione dell'algoritmo, vengono scelti k baricentri in modo tale che questi siano più distanziati possibili tra loro;
Tra i due, vista la complessità dei dati da analizzare, si preferisce usare il secondo metodo (tipico dell'algoritmico K-means++) proposto da Lloyd [8]. Lo schema di questo metodo è il seguente:

1. Il primo baricentro viene preso in modo casuale tra le osservazioni;
2. L'i-esimo baricentro viene preso tra le osservazioni rimaste, con probabilità proporzionale alla distanza della singola osservazione rispetto al baricentro più lontano.
3. Ripetere questo schema fino al raggiungimento del numero di baricentri prefissati k.

Questo tipo di inizializzazione permette di avere dei centri ben distanziati tra loro. Questo metodo, al contrario di quello random che non da peso alla distanza dei centri, per quanto più costoso, permette di trovare più velocemente la partizione finale.

4.3 Indici

Giudicare i cluster solamente da una loro rappresentazione o da una rappresentazione dei baricentri, spesso non è sufficiente. Basti pensare che se si va a considerare un insieme di dati con più di 3 dimensioni, diventa impossibile rappresentare graficamente i cluster. Perciò è utile utilizzare degli indici che ci possano spiegare la bontà delle partizioni trovate.
Quindi in questa Sezione si elencheranno gli indici che verranno utilizzati per valutare l'analisi dei cluster. In generale, si definiscono $X^1...X^N$ le N variabili, $\mathbf{B} = \{B_1...B_K\}$ i baricentri riferiti ai K cluster,C_k l'insieme delle osservazioni relativo al cluster $k \in \{0...K\}$, $\bar{\mathbf{B}}$ il baricentro di tutte le osservazioni e con $d(\cdot,\cdot)$ una generica distanza.

4.3.1 Elbow Analysis

L'Elbow Analysis è un metodo euristico di interpretazione e validazione della coerenza all'interno dell'analisi dei cluster progettata per aiutare a trovare il numero appropriato di cluster in un set di dati. In particolare si va a cercare il punto per cui un certo score, dopo un certo valore di K, non ha una variazione significativa.

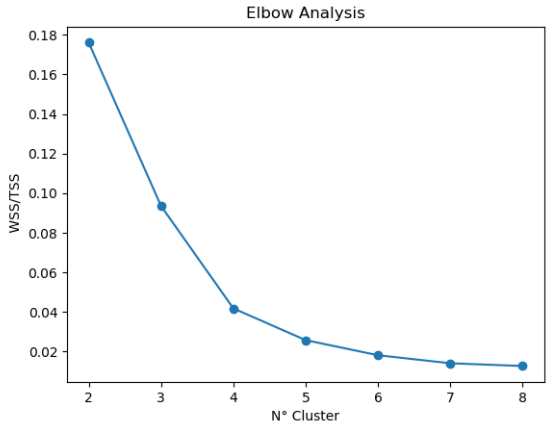


Figura 4.1: Esempio di Elbow Analysis

Nel caso preso in considerazione lo score preso in considerazione è il rapporto tra $\frac{WSS}{TSS}$, dove:

$$WSS = \sum_{k=1}^{K} \sum_{i \in C_k} d(X^i, B_k) \tag{4.10}$$

$$BSS = \sum_{k=1}^{K} |C_k| d(Q_k, \bar{\mathbf{B}}) \tag{4.11}$$

$$TSS = WSS + BSS \tag{4.12}$$

4.3.2 Silhouette Coefficient

Il silhouette coefficient è una misura di quanto un oggetto sia simile al suo cluster rispetto ad altri cluster . Il silhouette varia da -1 a +1, dove un valore elevato indica che l'oggetto è ben abbinato al proprio cluster e scarsamente abbinato ai cluster vicini. Se la maggior parte delle variabili ha un valore elevato, la configurazione del clustering è appropriata. Viceversa, se molti di queste hanno un valore basso o negativo, la configurazione del cluster può avere troppi o troppo pochi cluster.

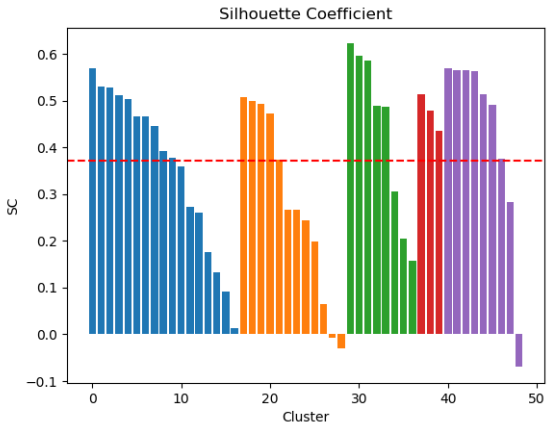


Figura 4.2: Esempio di Silhouette

La formula è la seguente:

$$a(i) = \frac{1}{|C_k| - 1} \sum_{j \in C_k, i \neq j} d(X^i, X^j) \tag{4.13}$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(X^i, X^j) \tag{4.14}$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad se \quad |C_i| > 1 \tag{4.15}$$

$$s(i) = 0, \quad se \quad |C_i| = 1 \tag{4.16}$$

Questo coefficiente sarà utilizzato per analizzare la qualità dei cluster con un K già scelto.

4.3.3 ARI: Adjusted Rand Index

L'ARI è una misura di similarità tra due partizioni di cluster. Ha un valore compreso tra 0 e 1, con 0 che indica che le due partizioni non concordano su nessuna coppia di punti e 1 che indica che le due partizioni sono esattamente identiche. La formulazione è la seguente: Dato un insieme S di N osservazioni e 2 partizioni di queste osservazioni, ovvero $A = \{A_1, A_2, \ldots, A_r\}$ e $T = \{T_1, T_2, \ldots, T_s\}$, la sovrapposizione di queste due partizioni può essere rappresentata in una tabella di contingenza $[n_{ij}]$ dove ogni casella n_{ij} denota il numero di osservazione in comune tra A_i e T_j : $n_{ij} = |A_i \cap T_j|$.

$A\|T$	T_1	T_2	...	T_s	*Somme*
A_1	n_{11}	n_{12}	...	n_{1s}	a_1
A_2	n_{21}	n_{22}	...	n_{2s}	a_2
:	:	:	:	:	:
A_r	n_{r1}	n_{r2}	...	n_{rs}	a_r
Somme	t_1	t_2	...	t_s	

Allora la definizione è la seguente:

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{t_i}{2}]/\binom{n}{2}}{1/2[\sum_i \binom{a_i}{2} + \sum_j \binom{t_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{t_j}{2}]/\binom{n}{2}} \tag{4.17}$$

Capitolo 5

Analisi dei dati artificiali

Dopo aver dato nei capitoli precedenti le nozioni utili per sviluppare l'algoritmo del K-means, sarà interessante vedere come questo si adatti alle varie tipologie di dati che verranno messe in input. In particolare vedremo come si comporterà nel caso Monodimensionale, sfruttando funzioni per il calcolo della distanza e del baricentro riscritte per l'occasione, e nel caso Multidimensionale, sfruttando funzioni già esistenti del pacchetto di *Python* **ot** che utilizza l'algoritmo del Sinkhorn spiegato nei capitoli precedenti, nel dettaglio **ot.emd2** per la distanza e **ot.bregman.barycenter** per il calcolo del baricentro.

5.1 Simulazioni caso monodimensionale

In questa Sezione vengono analizzate le performance dei metodi descritti con i dati simulati nel caso Monodimensionale.
Prima di andare nel dettaglio, verrà illustrato in modo schematico come vengono generati i dati.

1. Si hanno K cluster con k=$1\cdots$ K;

2. Per ogni cluster k si hanno N_k insiemi di punti $H_{k,i} = \{X_{k,i,j} : X_{k,i,j} \overset{i.i.d.}{\sim} P_{\theta_{k,i}}, j \in \{1, \cdots, n_{k,i}\}\}$, $i \in \{1 \cdots N_k\}$, con le relative funzioni di ripartizioni empiriche $\bar{F}_{k,i}$, definita in (2.3);

3. $\theta_{k,i}$ è generato a sua volta perturbando aleatoriamente un parametro θ_k che identifica il cluster, ossia $\theta_{k,i} = Unif(C_k - d_k, C_k + d_k)$ con C_k il centro relativo al cluster k, e d_k la deviazione dal centro C_k.

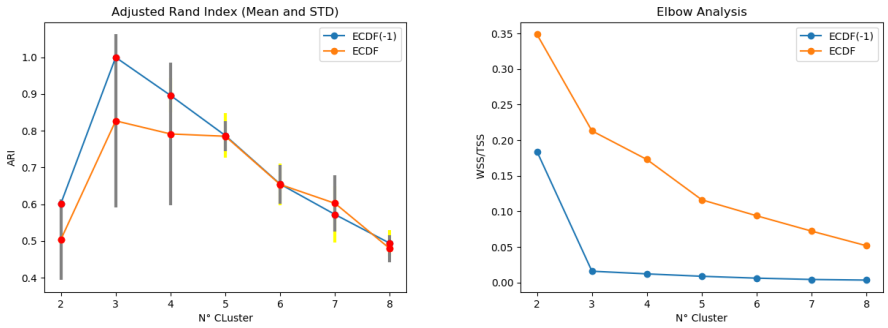
Per rendere il tutto più chiaro, mostriamo un esempio.
Fissati i valori di K,N_k,C_k e d_k, consideriamo la distribuzione $P_\theta = N(\mu, \sigma^2)$,

fissiamo σ^2=1 e $\theta = \mu$, quindi $\theta_{k,i} = \mu_{k,i} \overset{i.i.d.}{\sim} Unif(C_k - d_k, C_k + d_k)$ così da costruire gli $H_{k,i} = \{X_{k,i,j} : X_{k,i,j} \overset{i.i.d.}{\sim} N_{\mu_{k,i},1}, j \in \{1, \cdots, n_{k,i}\}\}$.
Questi passaggi sono stati pensati per generare K cluster ciascuno dei quali sia composta da N_k insieme di punti $H_{k,i}$ generati da misure di probabilità con parametri vicini tra loro, ma significativamente diversi rispetto a quelli degli altri cluster.
Dopodichè, nell'analisi che faremo, si considereranno tutte le $\bar{F}_{k,i}$ assieme, si farà inferenza sui cluster senza assumere note nè K nè l'appartenenza di ogni $\bar{F}_{k,i}$ al cluster k. Un ultima precisazione va fatta sui i due metodi che verrano usati: verrà chiamato ECDF il metodo in cui si userà la distanza d_F e verrà chiamato ECDF(-1) quello in cui si userà la distanza $d_{F^{-1}}$, ossia la distanza di W_2.

5.1.1 Simulazione 1: 3 cluster di gaussiane distanti

In questa simulazione abbiamo provato entrambi i metodi su degli $H_{k,i}$ con $P_{\theta_{k,i}} = N(\theta_{k,i}, 1)$ con queste caratteristiche:

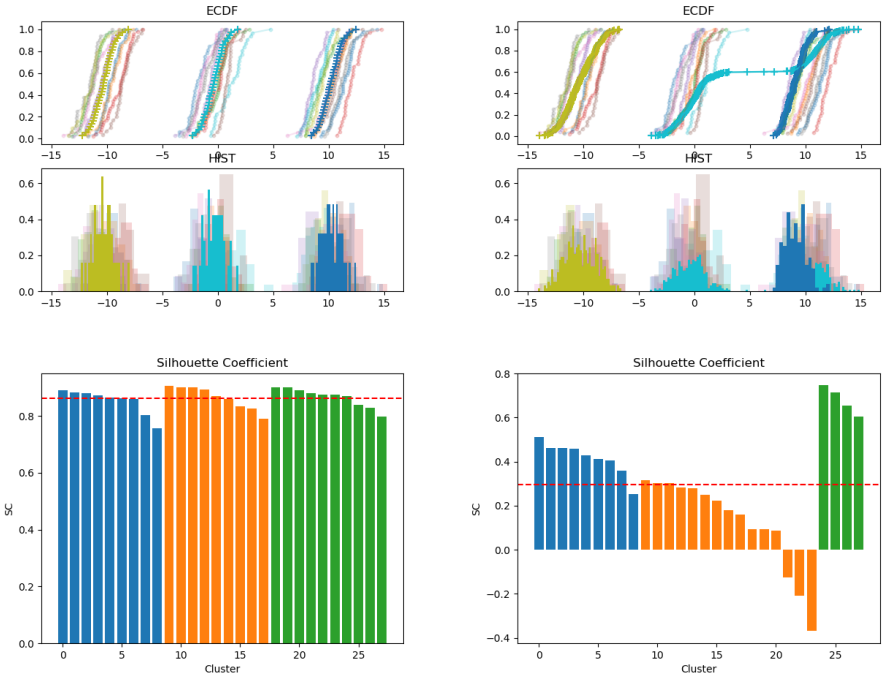
- Numero dei cluster, K= 3
- Centri dei cluster, C= [-10,0,10]
- Distanza dal centro, d= 2
- Varianza degli $H_{k,i}$, σ^2= 1
- Numero di $H_{k,i}$ per cluster, N_k : tra 5 e 10
- Numero di osservazione per istogramma, $n_{k,i}$: tra 30 e 50
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) Elbow Analysis (b) ARI

Figura 5.1

Per il metodo ECDF(-1) sia l'ARI che la Elbow Analysis dimostrano che il numero di cluster che rispetta i criteri di validità corrisponde a quello reale, cosa che per il metodo ECDF è più difficile da notare anche a causa della variabilità dei risultati generata da questo metodo.



(a) ECDF(-1) (b) ECDF

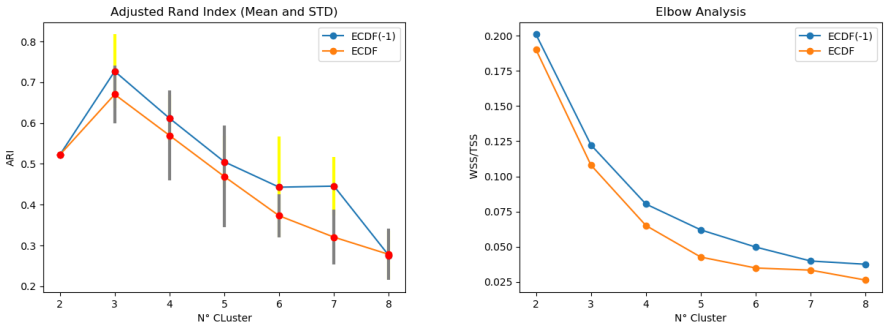
Figura 5.2: Istogrammi e Silhouette nel casodi 3 cluster di gaussiane distanti

Testando gli algoritmi con K=3, quello che si nota è che se il metodo ECDF(-1) trova i baricentri esattamente al centro dei cluster, il metodo ECDF in alcune inizializzazioni, come in Figura 5.2 tende a prendere più cluster insieme in uno solo, infatti il baricentro relativo al cluster trovato ha una struttura bimodale. Questo viene dimostrato anche dalla Silhouette, dove nel caso ECDF(-1) ha una media > 0.8, nel caso ECDF è circa 0.3 ed in uno dei cluster assume valori negativi.

5.1.2 Simulazione 2: 3 cluster di gaussiane non distanti

In questa simulazione abbiamo provato entrambi i metodi su degli $H_{k,i}$ con $P_{\theta_{k,i}} = N(\theta_{k,i}, 1)$ con queste caratteristiche:

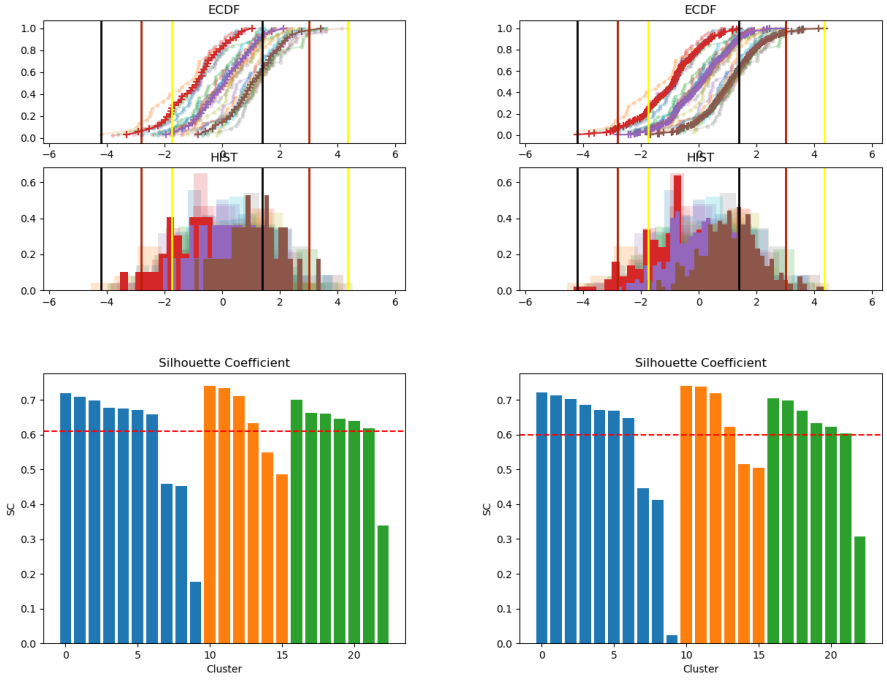
- Numero dei cluster, $K = 3$
- Centri dei cluster, C_k= [-1,0,1]
- Distanza dal centro, d_k= 0.3
- Varianza degli $H_{k,i}$, σ^2= 1
- Numero degli $H_{k,i}$ per cluster, N_k : tra 5 e 10
- Numero di punti per $H_{k,i}$, $n_{k,i}$: tra 30 e 50
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) Elbow Analysis (b) ARI

Figura 5.3: Simulazione 3 cluster non distanti

A differenza della simulazione precedente, entrambi i metodi danno risultati simili, anche se, come prevedibile, gli algoritmi restituiscono un maggior numero di partizioni diverse da loro rispetto alla simulazione precedente, e la qualità dei cluster, come si vede nel grafico dell'ARI in Figura 5.3, è diminuità. Sempre però grazie a questo risultato si può dire che per entrambi i metodi il numero di cluster scelto è 3.



(a) ECDF(-1) (b) ECDF

Figura 5.4: Istogrammi e Silhouette nel casodi 3 cluster di gaussiane non distanti

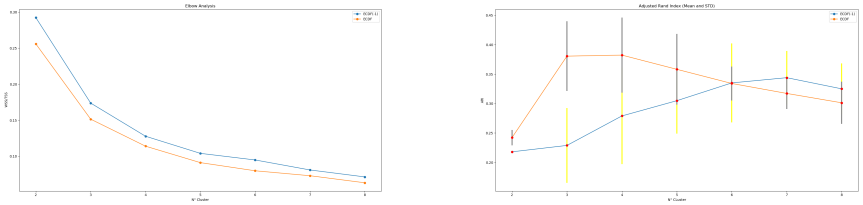
Andando ad analizzare questo caso specifico, si può notare che sia per il metodo ECDF che ECDF(-1), i risultati sono molto simili, come anche i baricentri stessi. Anche la silhouette conferma il risultato, poichè i cluster hanno valori simili (medie circa 0.6).

5.1.3 Simulazione 3: 5 cluster di distribuzioni esponenziali

In questa simulazione a differenza di quelle precedenti, si va vedere come i metodi si comportano con distribuzioni diverse dalla gaussiana, in particolare con una distribuzione esponenziale.
Sono stati provati entrambi i metodi su degli $H_{k,i}$ con $P_{\theta_{k,i}} = Exp(\lambda_{k,i})$ con queste caratteristiche:

- Numero dei cluster, $K = 5$
- Parametro λ_k dei cluster= [1/5,1/10,1/15,1/20,1/25]

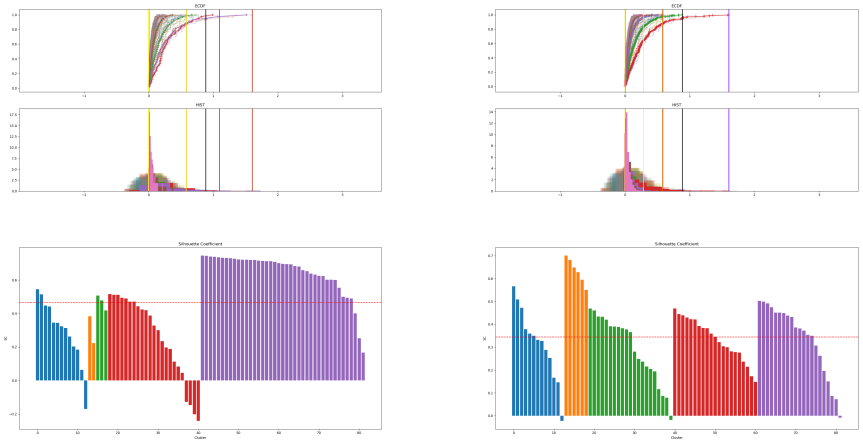
- Distanza dal centro, d_k= 1
- Numero di $H_{k,i}$ per cluster, N : tra 5 e 10
- Numero di punti per $H_{k,i}$, $n_{k,i}$: tra 30 e 50
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) Elbow Analysis (b) ARI

Figura 5.5: Simulazione 5 distribuzioni esponenziali

In questa simulazione se da una parte l'Elbow analysis in Figura 5.5 ci porta a dire che entrambi i metodi hanno un andamento simile (se non leggermente migliore quello ECDF), nell'ARI si nota una predizione migliore dei cluster per il metodo ECDF, anche se comunque entrambi i metodi hanno una variazione delle partizione alta per ogni K provato. In ogni caso, riferendoci all'elbow analysis, avremmo scelto come K 4 o 5.



(a) ECDF(-1) (b) ECDF

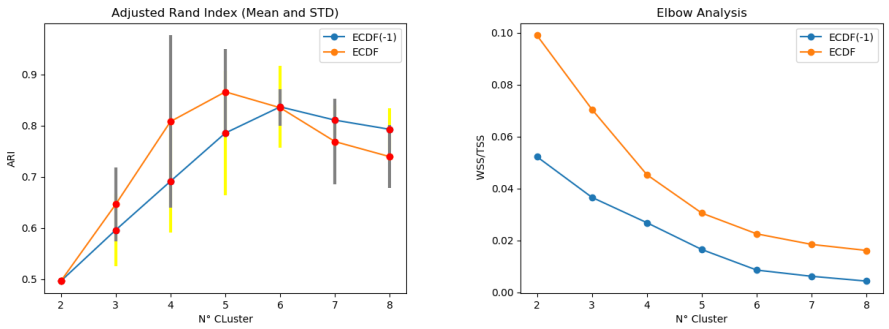
Figura 5.6: Istogrammi e Silhouette nel caso di 5 distribuzioni esponenziali

Nel caso specifico di K=5, si può notare che, grazie alla Figura 5.6. nel caso del metodo ECDF, le distribuzioni esponenziali sono ben distinte tra di loro, mentre nel caso del ECDF(-1) le distribuzioni tendono a sovrapporsi tra di loro. Questo risultato conferma l'ipotesi fatta nel caso dell'ARI nella Figura 5.5.

5.1.4 Simulazione 4: 2 cluster di distribuzioni gaussiane e 2 cluster di distribuzioni esponenziali

In questa simulazione vogliamo verificare se entrambi i metodi riescono a distinguere una distribuzione gaussiana sovrapposta ad una distribuzione esponenziale. Abbiamo provato entrambi i metodi su degli $H_{k,i}$ con $P_{\theta_{k,i}} = N(\theta_{k,i}, 1)$ nel caso di $k = 1, 2$, con $P_{\theta_{k,i}} = Exp(\lambda_{k,i})$ nel caso di $k = 3, 4$, con queste caratteristiche:

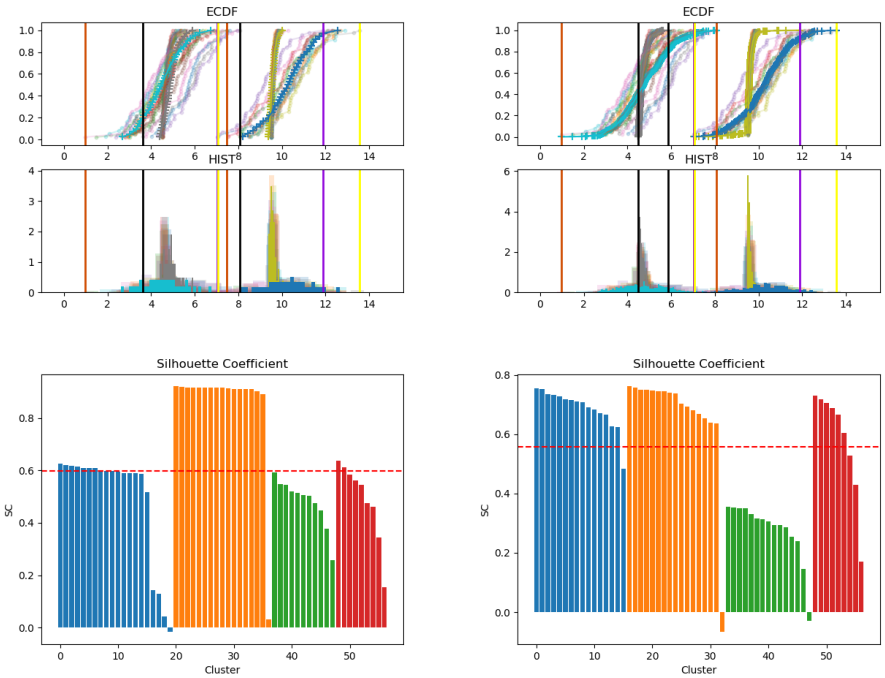
- Numero dei cluster, $K = 4$
- Parametro λ_k dei cluster= $1/5$
- Centri dei cluster, C_k = [5,10]
- Distanza dal centro d_k= 1
- Numero di $H_{k,i}$ per cluster, N_k : tra 5 e 10
- Numero di punti per $H_{k,i}$, $n_{k,i}$: tra 30 e 50
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) Elbow Analysis (b) ARI

Figura 5.7: Simulazione 2 distribuzioni gaussiani e 2 esponenziali

Nella Figura 5.7, vediamo che i valori dell'ARI tra i 4 e o 6 numero di cluster assumono dei valori alti anche se effetivamente quello più alto non è riferito al valore reale, ovvero 4, e le partizioni dei cluster hanno un'alta variabilità. Dalla elbow analysis si intuisce che il metodo ECDF(-1) è quello che genera cluster con variabilità rispetto ad ogni baricentro dei cluster minore.



(a) ECDF(-1) (b) ECDF

Figura 5.8: Istogrammi e Silhouette nel caso di 2 cluster di distribuzioni di gaussiane e 2 esponenziali

Nel caso specifico di K=4, si può notare in Figura 5.8 che i baricentri assumono effettivamente la forma di distribuzioni diverse, rispettivamente gaussiana e esponenziale. Anche la silhouette conferma la bontà del risultato (media circa 0,6).

5.2 Simulazioni caso Multidimensionale

In questa Sezione vengono analizzate le performance dei metodi descritti nel caso in cui i dati artificiali vengono generati da distribuzioni multidimensionali.

Prima di andare nel dettaglio, mostreremo schematicamente, in modo analogo al caso monodimensionale, come vengono generati i dati.

1. Si hanno K cluster con k=1$\cdots$ K;
2. Per ogni cluster k si hanno N insiemi di punti $H_{k,i} = \{X_{k,i,j} : X_{k,i,j} \sim P_{\theta_k}, j \in \{1, \cdots, n_i\}\}$ con le relative misure di probabilità empiriche $\bar{E}_{k,i}(\cdot) = \frac{1}{n_i} \sum_{j=1}^{n_{k,i}} \delta_{X_{k,i,j}}(\cdot)$;
3. A differenza del caso monodimensionale, in questo caso i θ_k saranno fissati.

Perciò i parametri presi in considerazione saranno fissati all'inizio di ogni simulazione. Dopodichè, come nel caso monodimensionale, si considereranno tutte le $\bar{E}_{k,i}$ assieme, si farà inferenza sui cluster senza assumere note nè K nè l'appartenenza.
Per applicare l'algoritmo del Sinkhorn, abbiamo bisogno della matrice di costo, che viene costrutita nel seguente modo.
Il dominio che contiene tutti i punti degli $H_{k,i}$ viene suddiviso in in modo tale da avere un ipercubo di dimensione d suddiviso in r^d sottoipercubi, tutti con la stessa dimensione, con r numero fissato. Inoltre si definisce $\mathbf{g}_{l,m} \in \mathbb{R}^{r^d \times d}$ con $l \in \{1, \cdots, r^d\}$ e $m \in \{1, \cdots, d\}$ come la matrice per cui ogni riga identifica le coordinate del sottoipercubo. Allora è utile definire la matrice di costo $\mathbf{C} \in \mathbb{R}^{r^d \times r^d}$ tale che $\mathbf{C}_{l_1,l_2} = ||\mathbf{g}_{l_1,\cdot} - \mathbf{g}_{l_2,\cdot}||$ con $||.||$ la norma euclidea. Quindi $\mathbf{C}$ è la matrice delle distanze euclidee tra i centri dei sottoipercubi. Inoltre, siccome l'algoritmo funziona per vettori discretizzati, discretizziamo ulteriormente, trasformando la misura di probabilità $\bar{E}_{k,i}$ in un vettore $\mathbf{e}_{k,i}$ r^d dimensionale per cui il valore $\mathbf{e}_{k,i,j}$ esprimerà la probabilità che un valore appartenga al sottoipercubo j-esimo.

5.2.1 Simulazione 1: 2 Cluster 2D a forma di circonferenza con raggio diverso e 2 cluster 2D di gaussiano molto correlate

In questa simulazione è stato pensato di generare sia degli $H_{k,i}$ che fossero generati da 2 cluster distributi in modo uniforme su circoferenze centrate nell'origine con raggio diverso, che degli $H_{k,i}$ che fossero generati da due distribuzione gaussiane con stessa media ma varianza diversa($P_{\theta_{k,i}} = N(\theta_{k,i}, \sigma_k)$).
In particolare le caratteristiche sono le seguenti:

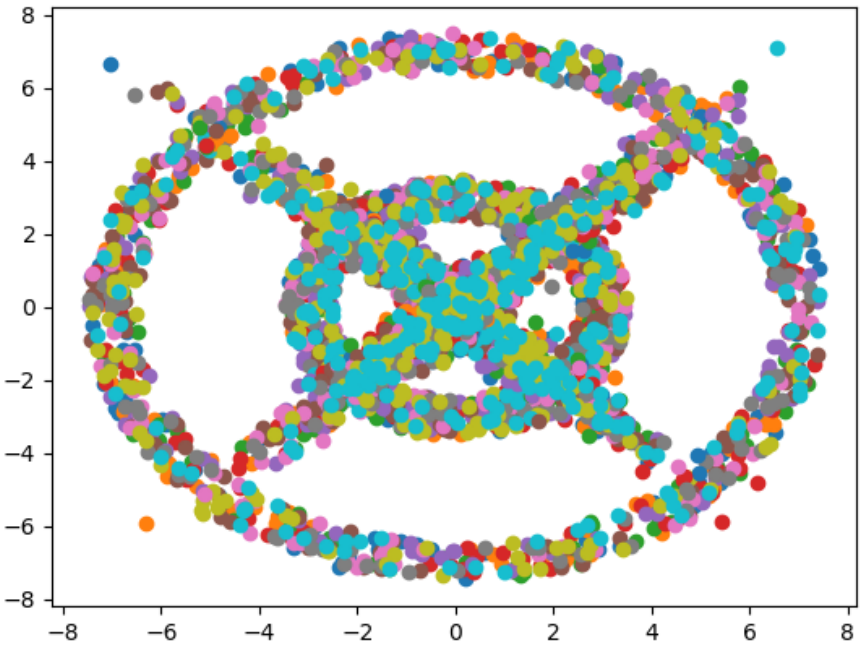
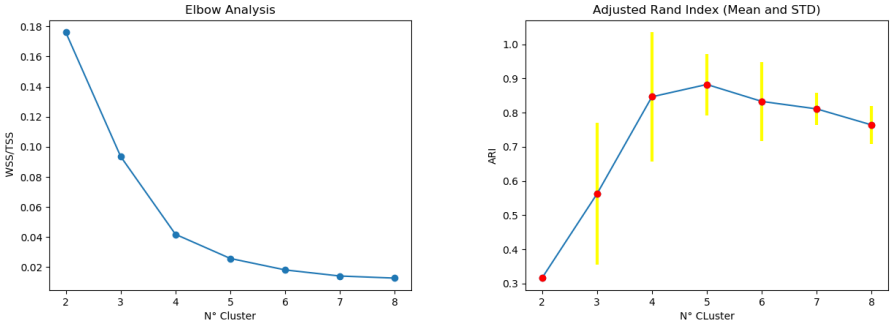


Figura 5.9: Plot degli $H_{k,i}$ nelle due circoferenze e nelle due gaussiane

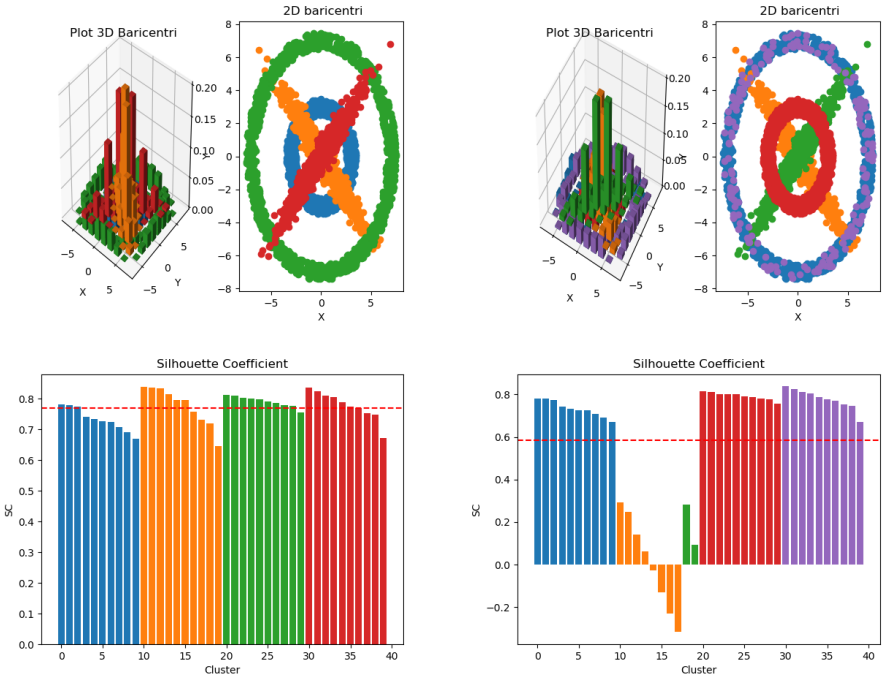
Figura 5.10

- Numero dei cluster, $K = 4$
- Centri delle circoferenze e μ_k: $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- Varianza degli $H_{k,i}$ gaussiani, σ_k^2: $\begin{bmatrix} 4 & 3,5 \\ 3,5 & 4 \end{bmatrix} \begin{bmatrix} 4 & -3,5 \\ -3,5 & 4 \end{bmatrix}$
- Numero di $H_{k,i}$ per cluster, N = 10
- Numero di osservazione per $H_{k,i}$, $n = 100$
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) Elbow Analysis (b) ARI

Studiando la elbow analysis e l'ARI in Figura 5.11a si osserva che i cluster che rispettano di più le condizioni di ottimalità sono K=4,5. Inoltre dall'analisi dalla 5.11b notiamo che c'è variabilità nei risultati, in particolare in K=4, che è anche il numero orinario di cluster. Vediamo in un analisi più dettagliata nei casi K=4,5 per vedere come sono distribuiti i baricentri dei cluster.



(a) K=4 (b) K=5

Figura 5.12: Baricentri 2D e Silhouette nel caso di K=4 e 5

Testando gli algoritmi con K=4,5, la differenza che si nota è che i primi quattro baricentri si dispongono allo stesso modo siano nel caso K=4 che K=5, il baricentro in più del secondo caso si dispone su un cluster esistente, in particolare in quello con circonferenza di raggio maggiore, quindi da un risultato ridondante, che non da informazioni in più all'analisi. Anche la silhouette conferma il risultato, in cui si nota che nel caso di K=4 i cluster hanno valori stabili e alti (media $\simeq 0.8$) ,mentre nel caso K=5 abbiamo sia una media più bassa (media $\simeq 0.6$) che due cluster ad occhio di valore più basso rispetto ad altri, che effettivamente sono quelli che si sovrappongono.
Questo esempio è stato pensato cercando di far venir fuori come l'algoritmo riesca a distinguere i vari gruppi di $H_{k,i}$ oltre che per le loro medie, anche per come sono distribuiti. Infatti nel caso del semplice k-means probabilmente avrebbe restituito un'alta variabilità dei cluster e lontani dalla loro effettiva partizione originale, poichè tutti gli $H_{k,i}$ hanno la stessa media.

5.2.2 Simulazione 2: 4 gaussiane 3D distanti

In questa simulazione è stato pensato di generare degli $H_{k,i}$ che fossero generati da 4 cluster distributi secondo distribuzione gaussiana con medie diverse ma stessa varianza ($P_{\theta_{k,i}} = N(\theta_{k,i}, \sigma)$). Verranno analizzate nel caso r=5,7,10, per studiare la variazione dell'algoritmo rispetto alla grandezza della griglia.
In particolare le caratteristiche sono le seguenti:

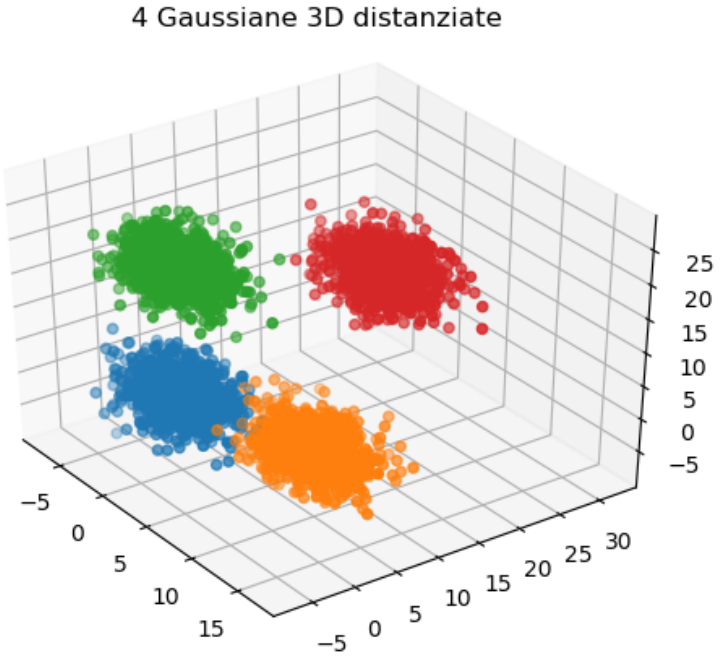
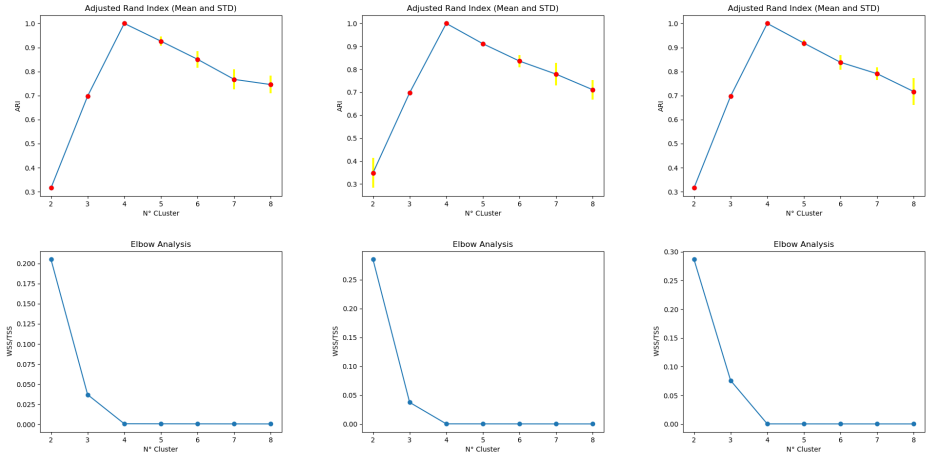


Figura 5.13: Plot degli 4 $H_{k,i}$ gaussiani

- Numero dei cluster, $K = 4$
- medie delle gaussiane μ_k: $\begin{bmatrix}0\\0\\1\end{bmatrix}$ $\begin{bmatrix}10\\5\\0\end{bmatrix}$ $\begin{bmatrix}0\\0\\20\end{bmatrix}$ $\begin{bmatrix}0\\25\\10\end{bmatrix}$
- Varianza degli $H_{k,i}$ gaussiani, σ_k^2: $\begin{bmatrix}5&0&0\\0&5&0\\0&0&5\end{bmatrix}$
- Numero di $H_{k,i}$ per cluster, N = 10
- Numero di osservazione per $H_{k,i}$, $n = 100$
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) ARI e Elbow Analysis caso r=5

(b) ARI e Elbow Analysis caso r=7

(c) ARI e Elbow Analysis caso r=10

Figura 5.14

In questo caso le analisi in Figura 5.14 ci dicono che per tutti i valori presi in considerazione di r, il valore di K più appropiato è quello originario, ovvero K=4. Il risultato era prevedibile poichè l'obbiettivo è quello di vedere il funzionamento dell'algoritmo con un caso semplice da analizzare. In seguito veranno considerati casi più complessi.

5.2.3 Simulazione 3: 4 gaussiane 3D vicine

In questa simulazione è stato pensato di generare degli $H_{k,i}$ che fossero generate da 4 cluster di distribuzione gaussiana con medie diverse e stessa varianza ma che, diversamente dalla simulazione precedente, non distano molto tra loro ($P_{\theta_{k,i}} = N(\theta_{k,i}, \sigma)$). Verranno analizzate nel caso di r=5,7,10. In particolare le caratteristiche sono le seguenti:

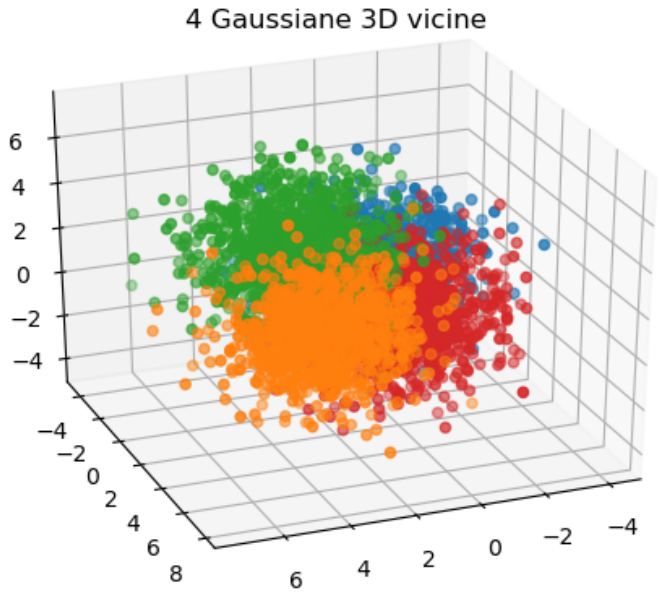
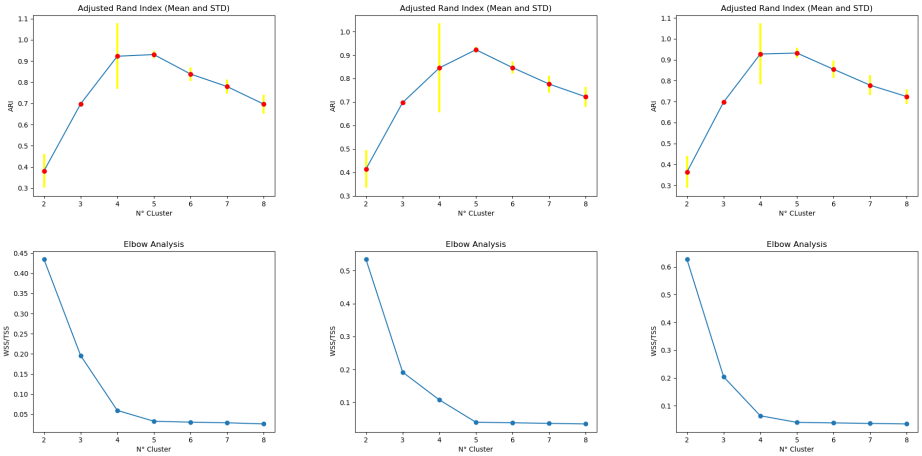


Figura 5.15: Plot degli $H_{k,i}$ nelle due circoferenze e nelle due gaussiane

Figura 5.16

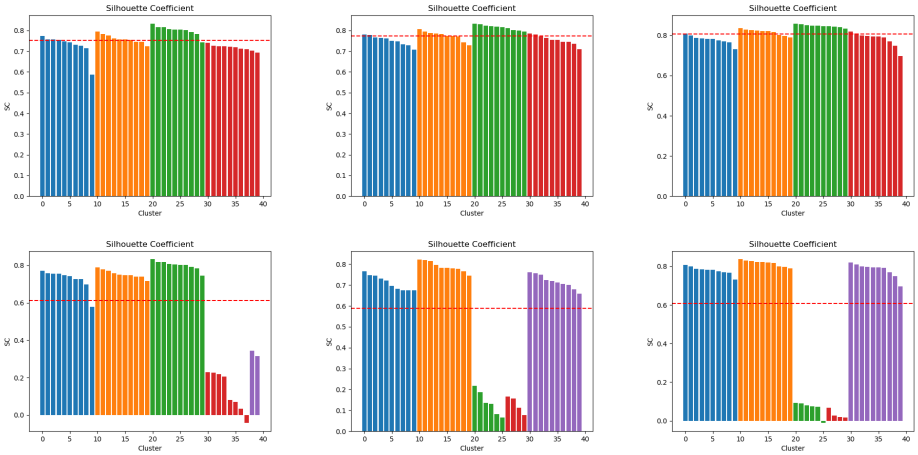
- Numero dei cluster, $K = 4$
- medie delle gaussiane μ_k: $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 2,5 \\ 2,5 \\ 0 \end{bmatrix}$ $\begin{bmatrix} 2,5 \\ 0 \\ 2,5 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$
- Varianza degli $H_{k,i}$ gaussiani, σ_k^2: $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$
- Numero di $H_{k,i}$ per cluster, N $= 10$
- Numero di osservazioni per $H_{k,i}$, $n = 100$
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) ARI e Elbow Analysis caso r=5 (b) ARI e Elbow Analysis caso r=7 (c) ARI e Elbow Analysis caso r=10

Figura 5.17

A differenza del caso precedente in cui le gaussiane erano distanziate e facilmente localizzabili, in questo caso si evidenzia una maggiore difficoltà nel trovare il numero di cluster giusto. In particolare per tutti i valori presi in considerazione di r, si nota dalla Figura 5.17 che i cluster ottimali sono K=4,5, dove per il primo valore di K le partizioni hanno un alta variabilità. Perciò è utile andare nel dettaglio con l'analisi del Silhouette coefficient per capire quale valori di K è il più corretto.



(a) Silhouette caso K=4 e K=5 con r=5 (b) Silhouette caso K=4 e K=5 con r=7 (c) Silhouette caso K=4 e K=5 con r=10

Con lo studio del Silhouette coefficient, si ha una situazione simile come nella Simulazione 1,dove si nota che la differenza tra i due casi è che i primi quattro baricentri si dispongono allo stesso modo mentre il baricentro in più del secondo caso si sovrappone su un cluster esistente, quindi da un risultato ridondante. Anche l'analisi della silhouette conferma il risultato, in cui si nota che nel caso di K=4 i cluster hanno valori simili e alti (media $\simeq 0.8$), mentre nel caso K=5 ho sia una media più bassa (media $\simeq 0.6$) che un valore della Silhouette associato a due dei 5 cluster più basso rispetto agli altri, che effettivamente sono legati ai cluster che si sovrappongono.

5.2.4 Simulazione 4: 2 gaussiane 3D con varibili poco correlate e 2 gaussiane 3D variabili molto correlate

In questa simulazione è stato pensato di generare degli $H_{k,i}$ che fossero generati da a 4 cluster distributi secondo distribuzione gaussiana per cui avessimo coppie di gaussiane con la stessa media in cui in ogni coppia ci sia un cluster con covarianza tra le variabili poco correlata ed un'altra coppia con covarianza tra le variabili fortemente correlata ($P_{\theta_{k,i}} = N(\theta_{k,i}, \sigma_k)$). Le caratteristiche sono le seguenti:

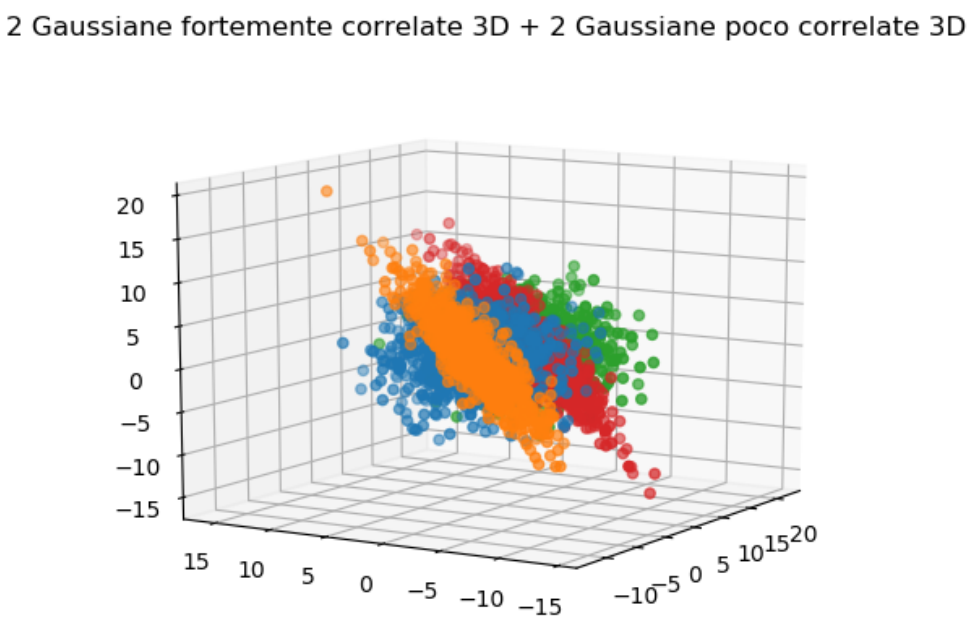
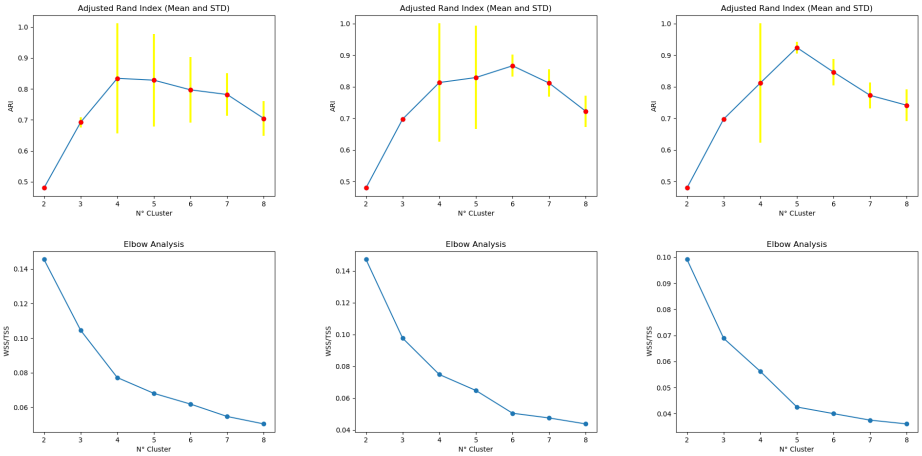


Figura 5.19: Plot degli $H_{k,i}$.

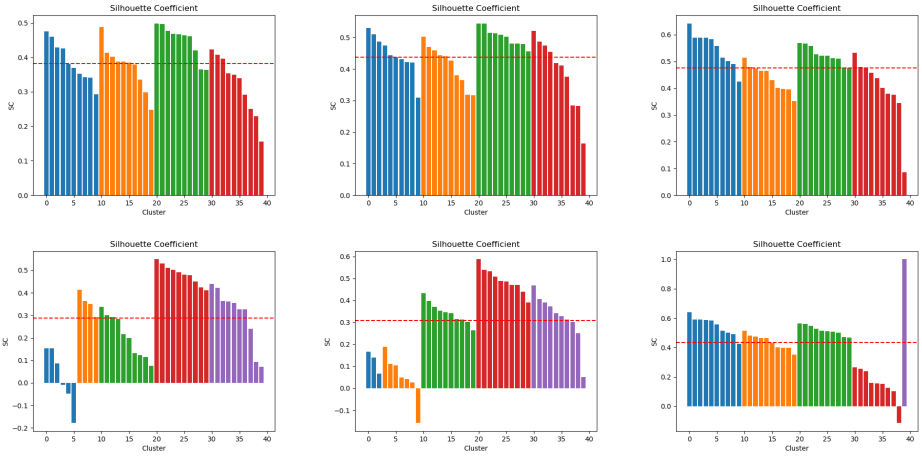
- Numero dei cluster, K : 4
- medie delle gaussiane μ_k: $\begin{bmatrix}0\\0\\1\end{bmatrix}$ $\begin{bmatrix}0\\0\\1\end{bmatrix}$ $\begin{bmatrix}0\\0\\10\end{bmatrix}$ $\begin{bmatrix}0\\0\\10\end{bmatrix}$
- Varianza degli $H_{k,i}$ gaussiani, σ_k^2: $\begin{bmatrix}15 & 10 & 1\\10 & 15 & 10\\1 & 10 & 15\end{bmatrix}$ $\begin{bmatrix}10 & 0 & 0\\0 & 10 & 0\\0 & 0 & 10\end{bmatrix}$
- Numero di $H_{k,i}$ per cluster, N : 10
- Numero di osservazione per $H_{k,i}$, n : 100
- Numero di cluster con cui viene fatta l'analisi: da 2 a 8 cluster



(a) ARI e Elbow Analysis caso r=5 (b) ARI e Elbow Analysis caso r=7 (c) ARI e Elbow Analysis caso r=10

Figura 5.20

In questo caso, per ogni valore di r otteniamo risultati diversi, dove i K migliori da scegliere sono 4,6,5 rispettivamente a r=5,7,10. Il fatto di avere dei risultati diversi tra loro è anche causato dalla variabilità dei risultati visibili nell'analisi dell'ARI in Figura 5.20. Perciò è necessario un analisi più approfondita tramite la silhouette.



(a) Silhouette caso K=4 e K=5 con r=5 (b) Silhouette caso K=4 e K=5 con r=7 (c) Silhouette caso K=4 e K=5 con r=10

Nella silhouette anche se la media in tutti i casi varia tra 0,3 e 0,5, i valori tra K=4 e K=5 sono differenti, dove nel primo caso si intuisce che ci

sono due cluster che hanno un valore basso, probabilmente dato dal fatto che appartengono allo stesso cluster, nel secondo caso i cluster hanno valori simili tra loro e comunque alti.
Questa simulazione è stata pensata per vedere se l'algoritmo fosse in grado di riconoscere i cluster che hanno stessa media ma varianza diversa, cosa che il k-means classico non riesce a riconoscere, cosa che l'algoritmo in analisi riesce a fare.

Capitolo 6

Analisi di un Dataset reale

A differenza delle simulazioni studiate nel capitolo precedente, in cui si è provato a vedere come entrambi metodi si comportassero al variare di diverse casistiche, in questo caso testiamo la qualità dei cluster generati dall'algoritmo su un dataset reale nel caso monodimensionale e multidimensionale, perciò molto più complesso rispetto ai dati usati precedentemente.
Nel caso specifico, vengono utilizzati dati che provengono dall'ARPA Lombardia, acronimo di agenzia operante per la protezione dell'ambiente, ente nazionale che si occupa della prevenzione e della protezione dell'ambiente. In particolare faremo riferimento al problema dell'inquinamento, basandoci su sostanze chimiche presenti nell'aria, come Ossido di Azoto, Biossido di Azoto e Ozono, misurate nel 2017.
Riassumiamo schematicamente le caratteristiche del dataset:

- 49 comuni in tutta la Lombardia
- Ogni comune ha delle stazioni che raccolgono dati su: Ossido di Azoto, Biossido di Azoto e Ozono.
- I dati raccolti sono una media giornaliera di queste sostanze
- É stata fatta la PCA con i dati standardizzati, passando da un dataset di 3 variabili ad una (varianza spiegata: 82,19%) nel caso monodimensionale, a due (varianza spiegata: 96,78%) nel caso multidimensionale
- I valori relativi ad ogni comune sono tra i 250 e i 360.

Per analizzare meglio questi dati, vediamo come le variabili legate alle sostanze chimiche sono correlati tra loro.

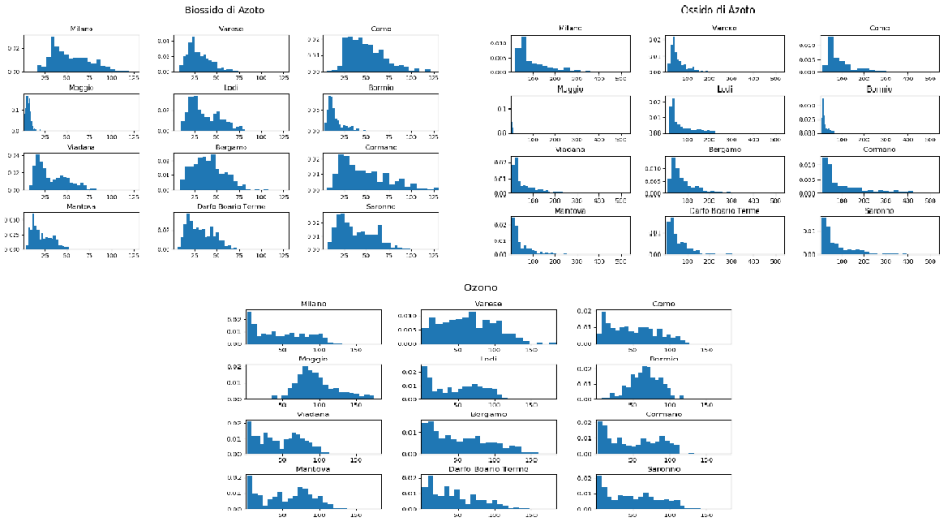


Figura 6.1: Distribuzioni di Ossido di Azoto,Biossido di Azoto e Ozono per alcune città

Correlazione	**Oss. di Azoto**	**Bioss. di Azoto**	**Ozono**
Oss. di Azoto	1	0.903060	-0.635056
Bioss. di Azoto	0.903060	1	-0.649494
Ozono	-0.635056	-0.649494	1

Dalla matrice di correlazione si ha una forte correlazione tra le variabili relative all'Azoto (-0.903060), mentre l'Ozono è abbastanza correlato negativamente all'ossido di azoto e biossido di azoto (-0.635056 e -0.649494). Questo risultato è anche dovuto a come viene prodotto l'ozono. Infatti si può trovare sia in zone rurali , dovuto essenzialmente al trasporto di ozono dall'alta troposfera e da produzione locale provocata da irraggiamento solare, che in zone industriali, dovuto al processo chimico-fisico che da origine allo smog fotochimico. Nel nostro caso questa variabile sarà un fattore secondario per quantificare l'inquinamento di una città, poichè, come mostrato in Figura 6.1, o è molto variabile, o è presente in alte concentrazioni in città come Moggio o Bormio, che non sono città industriali.
Per quanto riguarda i coefficienti generati dalla PCA, abbiamo i seguenti risultati:

Componente principale	Coeff. Ossido di Azoto	Coeff. Biossido di Azoto	Coeff. Ozono
PC1	0.59945981	0.60267001	-0.5267227
PC2	0.39038147	0.35434652	0.84972987
PC3	0.69874906	-0.71500169	-0.02285471

La prima variabile si può interpretare come indice di quantità di Azoto o Ozono presente nell'aria poichè i coefficienti dell'Ossido di Azoto e del Biossido di Azoto sono positivi, mentre quello dell'Ozono è negativo. Quindi se questa variabile assume valori positivi avrò alta concentrazione di Azoto, viceversa avrò alta concentrazione di Ozono. La variabile PC2 assumerà sempre valori positivi, poichè i suoi coefficienti sono tutti positivi. In particolare si può dire che l'incremento di questa variabile sarà dato in modo significativo dall'Ozono. L'ultima variabile si concentra esclusivamente sulle variabili Ossido di Azoto e Biossido di Azoto, dove se questo valore è positivo, la concentrazione di Ossido di Azoto sarà più alta di quella del Biossido di Azoto, viceverso se il valore è negativo. In generale quest'ultima variabilie avrò poco peso a causa dell'alta correlazione dei due elementi. Perciò in questo dataset ogni comune viene considerato come un $H_{k,i}$ delle simulazione viste precedentemente di cui non conosciamo però i cluster a priori. Questi $H_{k,i}$ verrano costruiti in due modi: o si aggregano i dati su una finestra temporale di un anno e con i dati giornalieri si costruisce la corrispettiva distribuzione empirica, o si considerano le stagioni come fasce temporali.
Un ulteriore precisazione va fatta sul significato dei cluster che saranno messi in ordine di fasce di inquinamento. In particolare vengono calcolate le medie di ogni cluster traslate nel primo quadrante e poi messe in ordine crescente secondo la loro distanza euclidea rispetto all'origine così che il cluster 0 sia quello più vicino all'origine ed il cluster 4 sia quello più lontano. Provvisoriamente interpretiamo il cluster 0 come quello meno inquinato ed il cluster 4 quello più inquinato.

6.1 Analisi dataset reale: caso monodimensionale

In questa sezione vedremo come analizzare i cluster del dataset sull'inquinamento, appena descritto, nel caso mondodimensionale. In particolare, rispetto alle simulazioni nel caso monodimensionale, è stato scelto di usare solo il metodo ECDF(-1), poichè, in seguito ai risultati precedenti, si è evinto che quello fosse il metodo che restituisse risultati migliori e meno variabili, ri-

spetto al metodo ECDF che, oltre a restituire risultati meno soddisfacenti e più variabili, ha tempi di esecuzione maggiori.

6.1.1 Analisi dei cluster

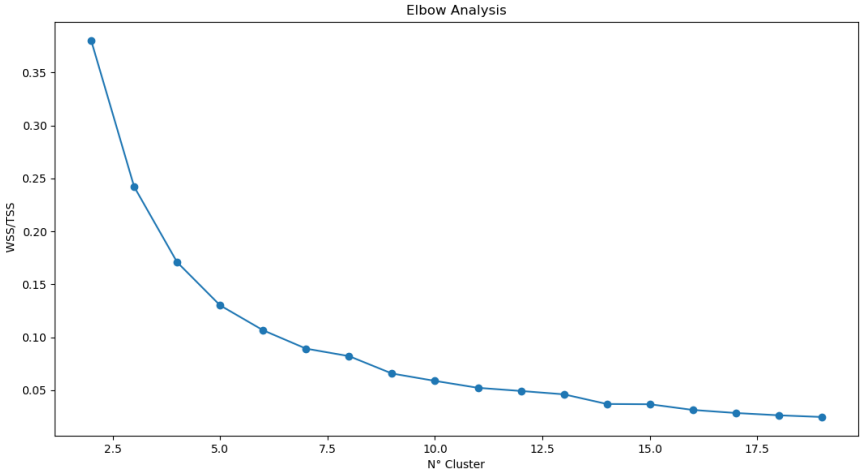
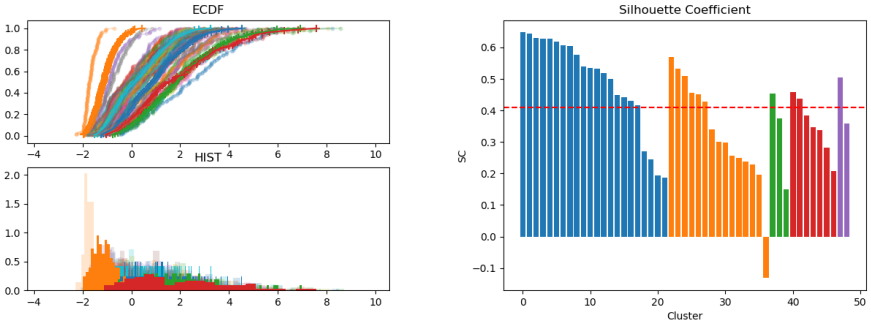


Figura 6.2: Elbow Analysis

Dall'Elbow analysis è difficile dare un numero di cluster ottimo per far si che si possano ottenere cluster sensati, ma il valore più significativo sembra essere il K=5.
A differenza delle simulazioni con dati artificiali, quello che si nota è che, ad ogni inizializzazione, si ha una partizione diversa. Per dare un risultato stabile, abbiamo deciso di fare 20 inizializzazioni e di vedere quante volte ogni città veniva assegnata un cluster. Di conseguenza ogni città viene assegnata al cluster con frequenza più alta, così da creare una nuova partizione.



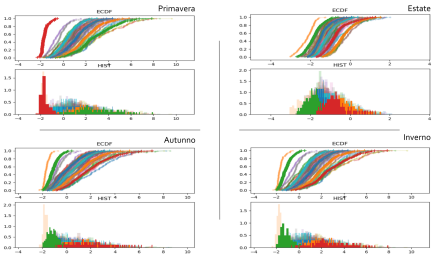
(a) Plot Baricentri (b) Silhouette

Figura 6.3: Dataset Reale

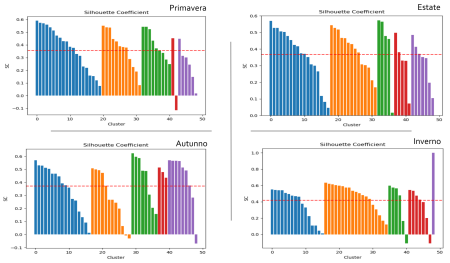
Dalla Figura 6.3 si può notare che i baricentri si sovrappongono molto di più a differenza delle simulazioni con dati artificiali, dove era anche facile identificare i cluster originali anche perchè venivano costruiti appositamente. In particolare si nota che la funzione di ripartizione del cluster con media più alta interseca quella di altri due cluster. Questo risultato può essere spiegato dalla dipendenza dei dati dal tempo.
Perciò abbiamo assunto ragionevole che ci potesse essere una dipendenza dalle stagioni.

6.1.2 Analisi dei cluster al variare delle stagioni

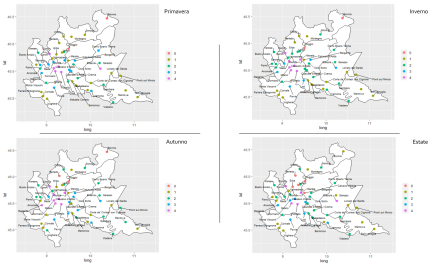
In questo paragrafo analizzaremo come cambiano i cluster al variare delle stagioni.



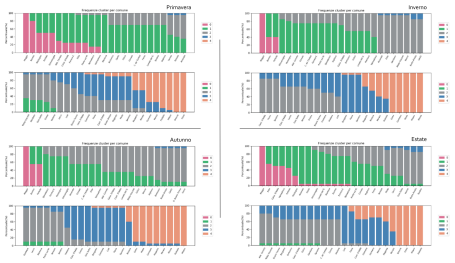
(a) Histogram & ECDF



(b) Silhouette



(c) Mappa Lombardia



(d) Frequenze dei cluster per comune

Figura 6.4: Sommario dei risultati al variare delle stagioni

Da i primi due plot in Figura 6.4 evince che c'è un effettiva separazione tra i vari cluster, cosa che nell'analisi precedente veniva meno. Dalla Figura (d) si nota che ci sono dei comuni che in base alla stagione cambiano cluster, ma la maggior parte rimangono nello stesso cluster al variare della stagione. Continua a rimanere una dipendenza geografica del grado di inquinamento, dove il cluster più "inquinato" si trova nella provincia di Milano, e più ci si allontana, più la fascia del cluster diminuisce.

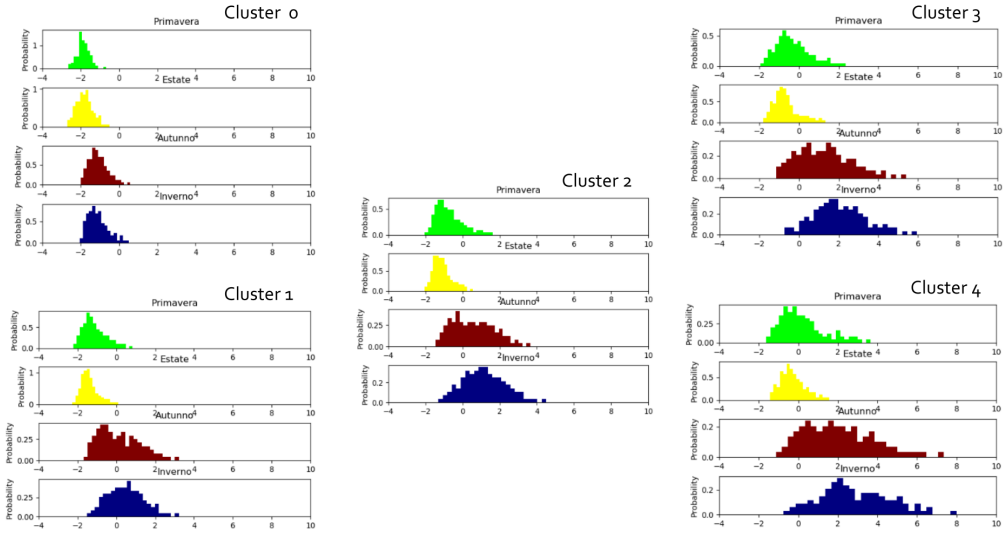


Figura 6.5: Baricentri di ogni stagioni relativi ai diversi cluster

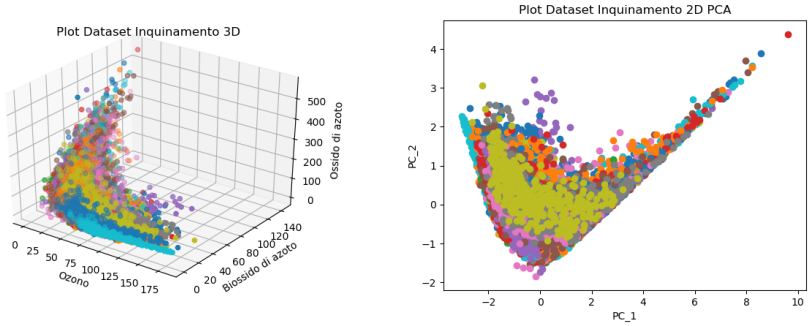
La Figura 6.5 rappresenta come in ogni cluster varia l'andamento dell'inquinamento in funzione delle stagioni. Si può notare come ci sia una netta differenza tra le coppie Primavera-Estate ed Autunno-Inverno da due punti di vista: il primo da quello della struttura del cluster, che per la prima coppia sembra assumere una forma a "campana unimodale", mentre per la seconda coppia si avvicina ad una mistura di 2 o più distribuzioni, considerando anche il più ampio intervallo di valori che assume la distribuzione; il secondo da punto di vista di medie, dove oltre che a crescere al variare del cluster, c'è una differenza tra le coppie di stagioni elencate prima, dove quella Autunno-Inverno ha un valore medio più alto.
In conclusione, è stato dimostrato che l'algoritmo nonostante l'alta variabilità delle partizioni, ci restituisce un risultato ragionevole ed interessante per poter osservare oltre alla media dei cluster, anche la distribuzione dei baricentri, cosa che con i metodi classici, come per esempio il k-means standard, non è possibile verificare.

6.2 Analisi dei dati reale: caso multidimensionale

In questa Sezione vedremo come analizzare i cluster del dataset dell'inquinamento nel caso multimensionale.

6.2.1 Analisi dei cluster

Come introdotto ad inizio capitolo, sono presi in considerazione i dati dell'inquinamento, selezionando le prime due componenti principali, come riportato in Figura 6.6.



(a) Plot 3D (b) Plot 2D PCA

Figura 6.6: Dataset Reale

Anche in questo dataset ogni comune viene considerato come un $H_{k,i}$ delle simulazione viste precedentemente di cui non conosciamo i cluster a priori. Perciò sarà interessante analizzare qual è il numero ideale di cluster da utilizzare.

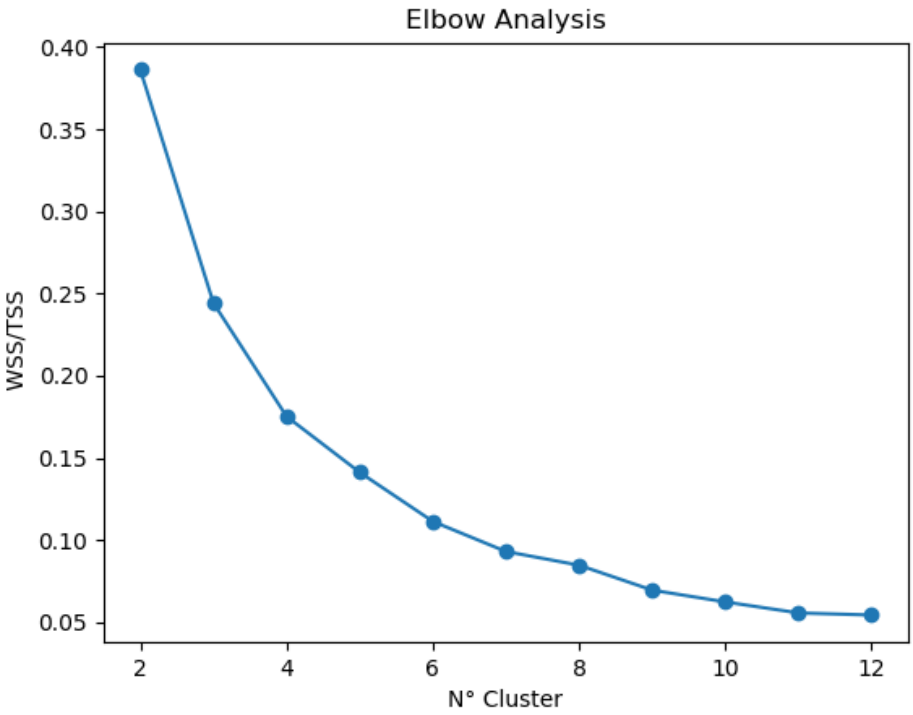
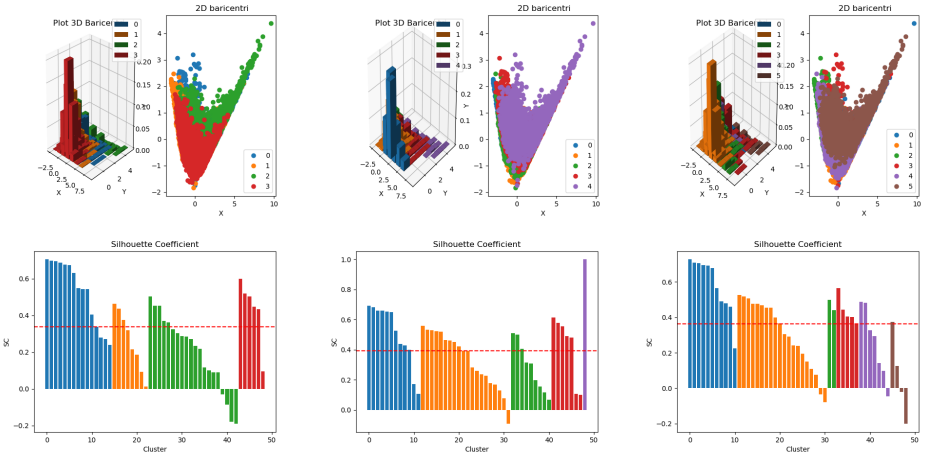


Figura 6.7: Elbow Analysis

Dalla elbow analysis è difficile stabilire un valore di numero di cluster K preciso, cosa che nelle simulazioni precedenti era più facile osservare, anche perchè erano esempi pensati appositamente. Perciò verranno considerate nello specifico i casi di K compresi tra 4 e 6, valori che rappresentano meglio il concetto di "gomito" della curva.



(a) K=4 (b) K=5 (c) K=6

Figura 6.8: Plot Baricentri e Silhouette

Nella Figura 6.8 abbiamo il plot dei baricentri, sia sotto forma di istogramma che come insieme dei punti relativi al cluster, ed il plot della Silhouette Coefficient.
Nel primo caso, si può notare che c'è una leggera differenza tra i baricentri dal punto di vista della posizione, mentre come forma tendono ad assumere tutti la stessa, perciò è difficile definire la bontà dei cluster sfruttando solamente questi plot rispetto alle simulazioni precedenti. Analizzando la Silhouette, la medie tendono ad essere simili in tutti e 3 i casi, $\simeq 0,4$. Nello specifico nel caso K=4 si hanno 4 cluster che sono sufficientemente separati, risultato simile a K=5, con un cluster in più composto solo da un comune, Moggio, che può essere interpretabile come un outlier. Nel caso K=6 si hanno 2 cluster che sono composti da un buon numero di comuni, mentre gli altri 4 sono più piccoli, segno che si possono creare dei cluster più grandi, assembleandoli insieme. Perciò la scelta più adeguata in questo caso sarebbe K=4.

6.2.2 Analisi dataset reale per stagioni

In questa simulazione vediamo come variano i cluster in base alle stagioni. In particolare verranno analizzati i dati considerati nelle ultime simulazioni, con le prime due componenti principali, in modo tale che si possano vedere graficamente come sono formati i cluster.

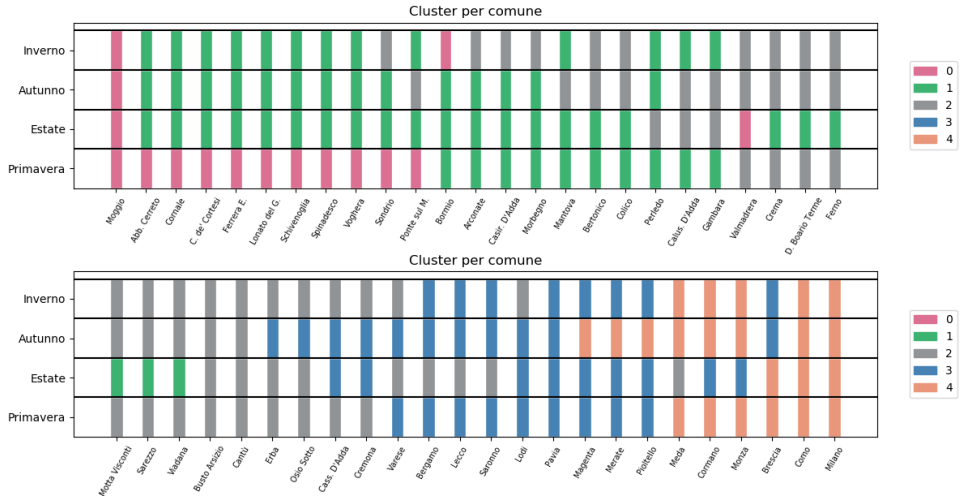


Figura 6.9: Cluster del Comune al variare della stagione.

Figura 6.10

Come nella sezione precedente, per generare il risultato sono state fatte 20 inizializzazioni ed ogni comune è stato assegnato al cluster in cui appariva più frequentemente.
La Figura 6.9 è stata creata dal risultato delle 20 inizializzazioni applicate ad ogni stagione dove ogni comune viene associato ad un colore che corrisponde all'ordine della fascia di inquinamento.
In generale si può dire che non ci siano grossissime variazioni, ovvero per ogni comune al massimo c'è una passaggio ad una fascia successiva o precedente ad eccezione di Meda, che in estate passa dal cluster 4 al 2 e Valmadrera che passa in estate da il cluster 2 al cluster 0. Un ulteriore osservazione si possono fare sui i comuni di Moggio, che appartiene sempre al cluster 0, confermato anche dalle altre simulazioni e dal fatto che nel territorio non sono presenti industrie, e Milano e Como che appartengono sempre alla fascia più alta che, come ben noto soprattutto per la prima città, sono molto industriali.

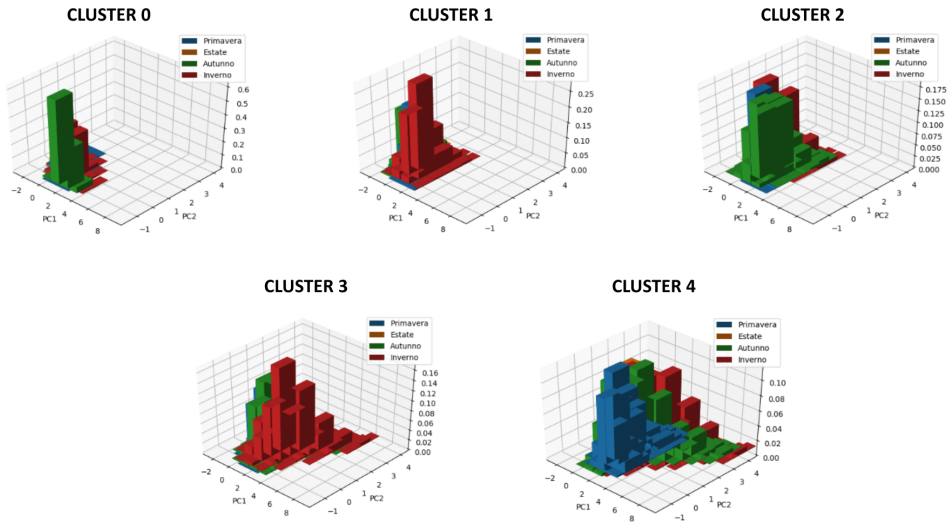


Figura 6.11: Plot Baricentri dei cluster per ogni stagione

La Figura 6.11 invece è stata creata calcolando il baricentro di ogni cluster per ogni stagione e sono stati messi nello stesso piano i baricentri della stessa fascia. Si nota che c'è una dipendenza dalla fascia di inquinamento, ovvero all'aumentare della fascia di inquinamento c'è uno spostamento del baricentro, sintomo di un valore significativo del cluster.
Considerando invece le stagioni in ogni cluster, non sembra ci sia una significativa differenza sia di posizione che di forma dei baricentri, ad eccezione del Cluster 4, in cui c'è una distinzione tra le stagioni, in particolare i valori nelle stagioni Primavera-Estate sono più bassi dei valori di Autunno-Inverno.

6.2.3 Confronto metodi nel dataset reale

In questa sottosezione si confrontano i risultati dei metodi studiati tra caso monodimensionale e multidimensionale,sfruttando il dataset reale sull'inquinamento.
In particolare verrà analizzato il metodo ECDF(-1) e il metodo del Sinkhorn nei casi 1, 2 e 3 dimensioni. Come nelle simulazioni nel caso monodimensionali, sono stato fatte 20 inizializzazione dell'algoritmo per ogni metodo a causa della variabilità delle partizioni generate dagli algoritmi, ed ogni comune viene assegnato al cluster più frequente. Per queste analisi è stato scelto K=5 per avere dei risultati allineati a quelli del metodo ECDF(-1) già generati nelle simulazioni precedenti.

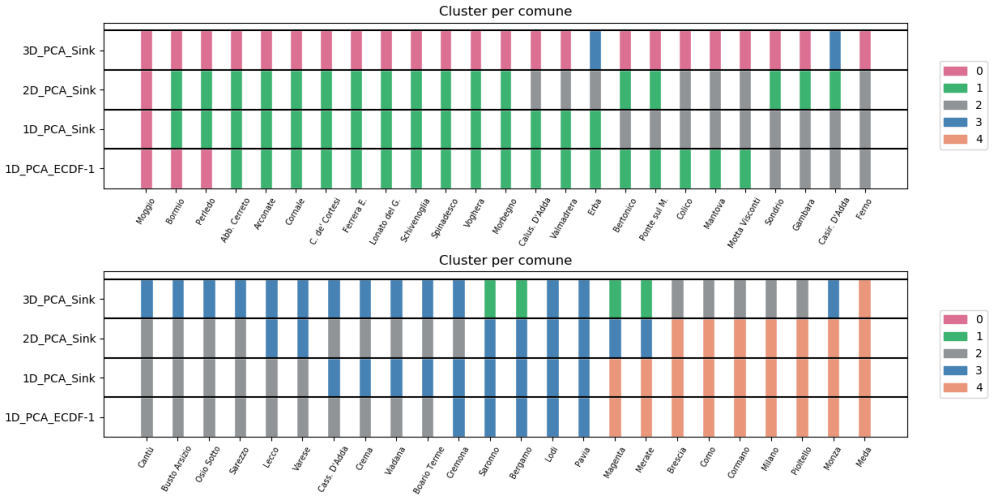


Figura 6.12: Cluster del Comune al variare del metodo usato.

Nella Figura 6.12, ad ogni comune viene assegnato un colore che corrisponde ad una fascia di inquinamento che viene ricavata al variare dei vari metodi. Da questa Figura evince che i primi tre metodi, ovvero ECDF(-1), Sinkhorn nel caso 1 e 2 dimensioni, generano partizioni che, ad eccezione di alcuni comuni che si spostano in un cluster di fascia maggiore o minore, rimangono costanti e coerenti con l'ordinamento dato. Nel caso del Sinkhorn in 3 dimensioni, le partizioni sono molto diverse dai metodi precedenti, e anche un ordinamento poco ragionevole, probabilmente dato dal fatto che alcune sostanze sono inversamente proporzionali tra di loro, perciò rende difficile poter dare un criterio di ordinamento secondo le zone più inquinate.

6.3 Analisi dati reali senza PCA

In questa Sezione vogliamo mostrare l'analisi monodimensionale e bidimensionale del dataset reale sull'inquinamento, senza però sfruttare l'utilizzo della PCA, in modo tale da poter vedere come variano i risultati utilizzando dati con la loro unità di misura. In particolare nel caso monodimensionale prenderemo come variabile l'Ossido di Azoto, poichè è quella con varianza maggiore, mentre nel caso multidimensionale aggiungeremo la variabile del Biossido di Azoto poichè più significativa come indice di inquinamento rispetto alla variabile Ozono.

6.3.1 Caso Monodimensionale

In questa sezione come già descritto precedentemente studieremo come verranno generati i cluster secondo la variabile di Ossido di Azoto, utilizzando il metodo ECDF(-1), come nelle precedenti analisi.

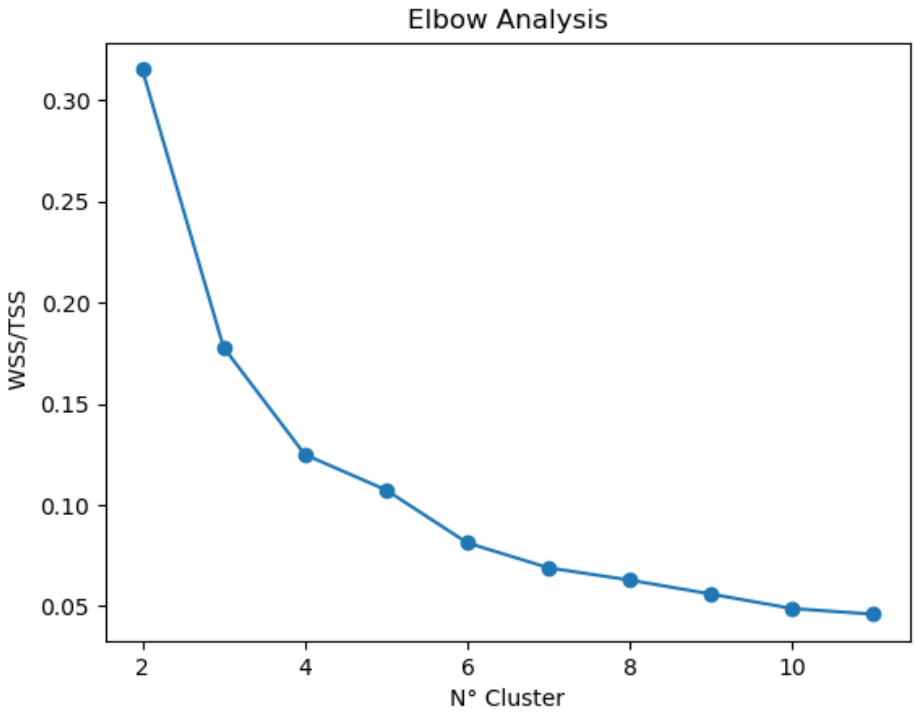
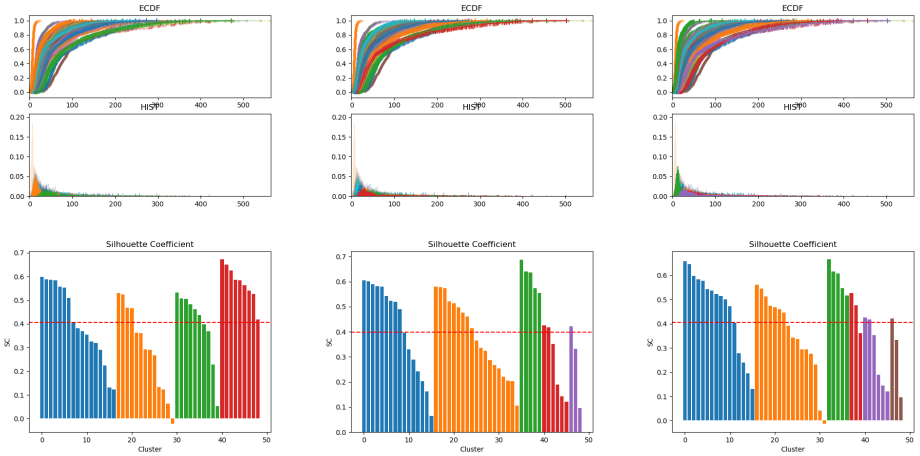


Figura 6.13: Elbow Analysis

Dalla Elbow Analysis in Figura 6.13 si può intuire che il numero di cluster che rispecchiano il criterio di scelta sono i valori di K=4,5,6. Per ciò per fare un analisi più approfondita andremo ad analizzare i coefficienti della silhouette per questi valori di K.



(a) K=4 (b) K=5 (c) K=6

Figura 6.14: Istogrammi e Silhouette

Dalla Figura 6.14 possiamo vedere quali sono i risultati degli algoritmi nel caso di K=4,5,6, analizzando in particolare gli istogrammi dei baricentri e la silhouette coefficient. Per prima cosa notiamo che i baricentri a differenza dei casi con la PCA sono molto più variabili, e nel caso di K=5 e K=6 le due funzioni di ripartizoni relative agli ultimi due cluster si intersecano.
Riguardo lo studio della Silhouette, tutti e 3 hanno media circa parti a 0.4, però guardando come sono distribuiti i valori ed anche alla grandezza dei cluster notiamo che, nel caso di K=4, i cluster sono omogenei, mentre negli altri due casi ci sono alcuni cluster con poche città e valori bassi di Silhouette. Perciò riteniamo corretto un valore di K=4.

6.3.2 Caso Multidimensionale

In questa Sezione studieremo come verranno generati i cluster secondo le variabili di Ossido di Azoto e Biossido di Azoto, utilizzando l'algoritmo del Sinkhorn nel caso bidimensionale, come nelle precedenti analisi.

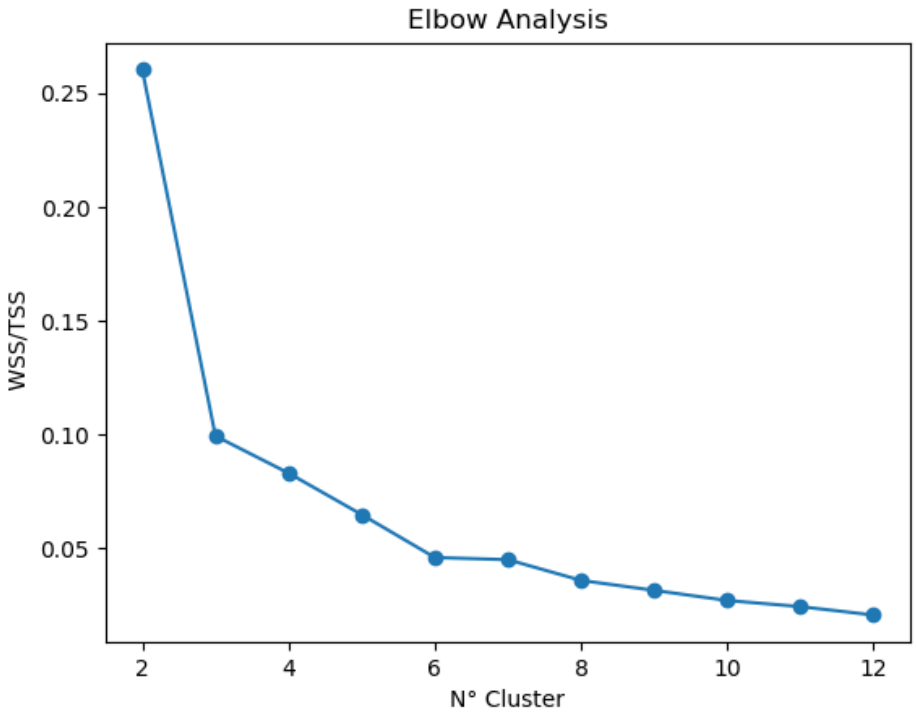
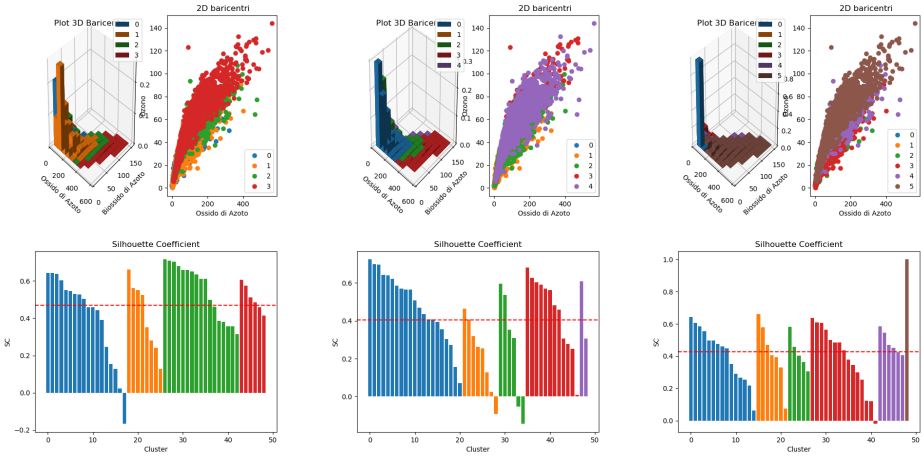


Figura 6.15: Elbow Analysis

Dalla Elbow Analysis in Figura 6.15, riteniamo che il numero di cluster che possa rispecchiare il criterio del "gomito", possano essere i casi di K=4,5,6. Per capire effettivamente quale valore di K scegliere, analizzeremo nello specifico andando ad osservare i plot dei baricentri ed il calcolo della Silhouette.



(a) K=4 (b) K=5 (c) K=6

Figura 6.16: Istogrammi e Silhouette

Dai i plot in Figura 6.16, notiamo che i baricentri sono leggermente distanziati tra di loro in base alla loro fascia appartente ed in più si può notare la forte correlazione che c'è tra le due variabili. Dalla Silhouette Analysis, notiamo che al variare di K, i cluster tendono a distribuirsi omogeneamente, escluso nel caso K=6 dove abbiamo un cluster con un singolo valore. Riguardo alle medie delle SC, se nel caso K=5,6 abbiamo circa un valore simile, ovvero di 0.4, nel caso di K=4 abbiamo un valore più grande, circa K=0.5. Perciò riteniamo corretto come valore di K il valore 4.

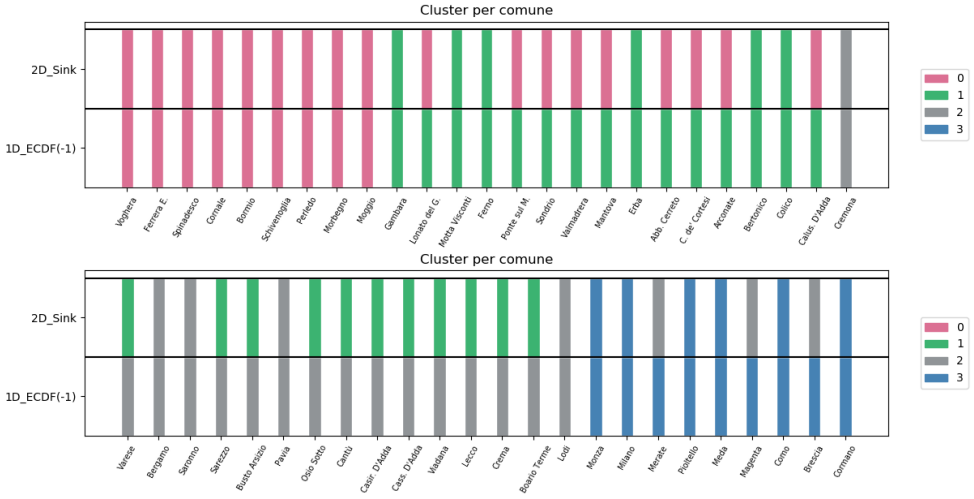


Figura 6.17: Confronto cluster con i due metodi usati

Nella Figura 6.17, abbiamo il riepilogo delle partizioni risultanti dagli algoritmi nel caso monodimensionale e multidimensionale con K=4. Quello che si nota è che c'è una differenza di raggruppamento per alcune città. In particolare nel caso del cluster 0, il metodo del Sinkhorn in 2 dimensioni si hanno moltw più città rispetto al metodo ECDF(-1), portando ad avere anche negli altri cluster delle partizioni diverse. Il cluster 3, ovvero quello più inquinato, tranne per alcune città, non è molto diverso nei due metodi usati. In generale le fasce associate ai cluster danno dei risultati ragionevoli per quanto riguarda il grado di inquinamento.

Conclusioni

In seguito allo studio e alle analisi svolte, è possibile trarre delle conclusioni, in riferimento, in particolare, agli ultimi risultati ottenuti.
Per quanto riguarda i dati simulati è stato possibile constatare che il metodo ECDF(-1) dia risultati migliori rispetto al metodo ECDF in termini di qualità dei cluster, variabilità delle partizioni, e tempo di esecuzione. Nel caso multidimensionale abbiamo visto come, dopo delle analisi approfondite grazie agli indici di ARI, Silhouette e la Elbow analysis, l'algoritmo ha predetto in modo corretto i cluster che sono stati pensati nelle simulazioni con dati artificiali.
Per quanto riguarda l'analisi sul dataset reale, è stato interessante osservare come, sia nel caso monodimensionale che nel caso multidimensionale, si potessero condurre delle analisi approfondite analizzando non solo la posizione del valore medio dei punti del cluster, ma anche dalla forma della distribuzione ottenuta calcolando il baricentro. AD esempio confrontando l'analisi fatta senza distinguere le stagioni e distinguendo le stagioni abbiamo notato che le funzioni di ripartizioni associate ai baricentri ed ai relativi istogrammi sono nettamente più distinguibili.
In generale, si può ritenere che questo algortimo sia utile per fare delle analisi approfondite, non basandosi soltanto sulla posizione di un cluster, come nel caso del K-means classico, bensì basandosi anche sulle distribuzioni degli stessi. Questo fatto è stato dimostrato anche con esempi specifici sulle simulazioni: nel caso monodimensionale, confrontando distribuzioni diverse, come quelle gaussiane ed esponenziali; nel caso multidimensionale, confrontando insiemi di punti distribuiti in modo differente, ma con centro del cluster identico per tutti.
Per contro però, aggiungere informazioni implica aumentare la variabilità dei risultati, portando quindi ad analisi più lunghe e approfondite per avere dei risultati ragionevoli. Inoltre, soprattutto nel caso multidimensionale, l'algoritmo ha un tempo di esecuzione elevato, con dipendenza esponenziale dal numero di variabili del dataset d ($O(r^{2d})$). Basti pensare che, se nel caso monodimensionale un'inizializzazione dell'algoritmo può durare qualche se-

condo, già a due dimensioni può durare qualche minuto.
Per concludere, come ulteriore sviluppo di questo lavoro, si potrebbero svolegere dei confronti con altri metodi di clustering, come il classico K-means o altri metodi che sfruttano la distanza di Wasserstein. In alternativa, sarebbe possibile condurre uno studio sull'abbattimento dei tempi di esecuzione, andando a migliorare ed ottimizzare la scrittura del codice dell'algoritmo.

Bibliografia

[1] Federico Bassetti, Stefano Gualandi, and Marco Veneroni. On the computation of kantorovich-wasserstein distances between 2d-histograms by uncapacitated minimum cost flows. *arXiv preprint arXiv:1804.00445*, 2018.

[2] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[3] Rui Castro. Introduction and the empirical cdf, 2013. Lecture notes. https://www.win.tue.nl/ rmcastro/AppStat2013/.

[4] Gerard Cornuejols and Reha Tütüncü. *Optimization methods in finance*, volume 5. Cambridge University Press, 2006.

[5] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[6] Antonio Irpino, Rosanna Verde, and Francisco de AT De Carvalho. Dynamic clustering of histogram data based on adaptive squared wasserstein distances. *Expert Systems with Applications*, 41(7):3351–3366, 2014.

[7] LV Kantorovich. On translation of mass (in russian), c r. In *Doklady. Acad. Sci. USSR*, volume 37, pages 199–201, 1942.

[8] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[9] Liang Mi, Wen Zhang, Xianfeng Gu, and Yalin Wang. Variational wasserstein clustering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 322–337, 2018.

[10] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

[11] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[12] Matthew Thorpe. Introduction to optimal transport. lecture notes. http://www.damtp.cam.ac.uk/research/cia/introduction-optimal-transport-lent-2019.