

# Soft Clustering e Rough K-Means

Elia Mercatanti

Università degli Studi di Firenze  
Scuola di Scienze Matematiche, Fisiche e Naturali  
Corso di Laurea Magistrale in Informatica - Data Science

Progetto MASL, Gennaio 2021

- È uno degli strumenti più utilizzati nell'analisi dei dati.
- Negli ultimi decenni c'è stato un crescente interesse verso tali tecniche, in particolare il **soft clustering**.

## Cluster Analysis

- Mira a determinare un piccolo numero  $k$  di gruppi omogenei da un insieme di  $n$  oggetti secondo una misura della dissomiglianza basata su  $p$  variabili osservate  $X_1, \dots, X_p$ .
- Ha il compito di suddividere un set di dati in gruppi (**clusters**) in modo significativo ed utile.

- Esistono però oggetti che hanno caratteristiche intermedie tra i cluster, spesso non possono essere chiaramente assegnati.
- L'approccio classico (*hard*) al raggruppamento porta ad un'assegnazione non realistica, gli oggetti sono costretti ad appartenere a un solo cluster.

## Soft Clustering

- L'approccio **soft** nasce per superare questo inconveniente.
- Idea: ogni dato può appartenere a più di un cluster.
- Esistono almeno quattro tipi di approcci di clustering *soft*: *fuzzy*, possibilistico, basato su modelli e *rough*.

# Fuzzy Clustering - Fuzzy K-Means

- I dati vengono assegnati ai cluster in base ad un grado di appartenenza.
- Grado  $\approx 1$ , oggetto simile al prototipo di quel cluster, grado 0 ne indica la non appartenenza, lontano dal relativo prototipo.
- Estensione dell'algoritmo *K-Means* al caso *fuzzy* (**FkM**).
- Ha lo scopo di determinare una partizione *fuzzy* di  $n$  oggetti in  $k$  cluster risolvendo il seguente problema di minimizzazione:

$$\min_{U, H} J_{FKM} = \sum_{i=1}^n \sum_{g=1}^k u_{ig}^m d^2(x_i, h_g),$$

$$\text{s.t. } u_{ig} \in [0, 1], \quad \sum_{g=1}^k u_{ig} = 1, \quad i = 1, \dots, n, \quad g = 1, \dots, k,$$

# Fuzzy Clustering - Fuzzy K-Means

- $m > 1$  è il parametro di **fuzziness**, di solito fissato tra 1.5 e 2.
- La soluzione è ottenuta mediante un algoritmo iterativo. Con il metodo del moltiplicatore Lagrangiano.
- La soluzione iterativa è la seguente:

$$u_{ig} = \frac{1}{\sum_{g'=1}^k \left( \frac{d^2(x_i, h_g)}{d^2(x_i, h_{g'})} \right)^{\frac{1}{m-1}}} \quad h_g = \frac{\sum_{i=1}^n u_{ig}^m x_i}{\sum_{i=1}^n u_{ig}^m}.$$

## Pro:

- I gradi di appartenenza sono inversamente correlati alle relative dissomiglianze tra gli oggetti e i centroidi.
- Per questo motivo, i gradi di appartenenza possono essere interpretati come gradi di condivisione.
- Semplice ed efficiente.

## Contro:

- Le prestazioni dei metodi basati sul *K-Means* sono influenzati dai valori anomali (*outliers*).
- Dovuto ai vincoli di somma unitaria dei *membership degrees*. Gli oggetti anomali assegnati ai cluster influenzano i centroidi.

# Possibilistic Clustering - Possibilistic K-Means

- Per cercare di superare l'inconveniente dei valori anomali.
- Rilassare i vincoli della somma unitaria sui *membership degrees*, aggiungendo un termine di penalizzazione.
- Gradi di appartenenza interpretati come gradi di compatibilità degli oggetti con i cluster.
- L'algoritmo di clustering possibilistico più noto è il metodo di clustering **PkM**, che può essere formalizzato come di seguito:

$$\min_{T, H} J_{PkM} = \sum_{i=1}^n \sum_{g=1}^k t_{ig}^{\eta} d^2(x_i, h_g) + \sum_{g=1}^k \gamma_g \sum_{i=1}^n (1 - t_{ig})^{\eta},$$
$$\text{s.t. } t_{ig} \in [0, 1], \quad i = 1, \dots, n, \quad g = 1, \dots, k,$$

# Possibilistic Clustering - Possibilistic K-Means

- $\gamma_g$  è *cluster-specific*, regola l'importanza dei cluster:

$$\gamma_g = \gamma \frac{\sum_{i=1}^n u_{ig}^m d^2(x_i, h_g)}{\sum_{i=1}^n u_{ig}^m}, \quad g = 1, \dots, k,$$

- $\gamma = 1$ ,  $h_g$  e i vari  $u_{ig}$  sono ottenuti dal FkM.
- I  $\gamma_g$  rappresentano il peso relativo del secondo termine della funzione obiettivo rispetto al primo.
- $\eta (> 1)$  è il **fuzzifier**.
- Il secondo termine evita inoltre la soluzione banale con  $T = 0$ .
- Soluzione iterativa del PkM:

$$t_{ig} = \frac{1}{1 + \left( \frac{d^2(x_i, h_g)}{\gamma_g} \right)^{\frac{1}{\eta-1}}} \quad h_g = \frac{\sum_{i=1}^n t_{ig}^{\eta} x_i}{\sum_{i=1}^n t_{ig}^{\eta}}.$$



## Pro:

- I gradi di tipicità,  $t_{ig}$ , sono inversamente correlati alle differenze tra le osservazioni e i centroidi.
- Calcolati considerando solo la dissomiglianza tra l'osservazione e il centroide più vicino.
- I valori anomali risultano lontani da tutti i centroidi. Di solito dunque hanno gradi di tipicità vicini allo 0.

## Contro:

- Può soffrire del rischio di cluster coincidenti, dovuto all'assenza del vincolo alla somma dei gradi di tipicità.

# Model-Based Clustering

- Presuppone che i dati siano generati da un modello statistico.
- In questo contesto, i dati seguono una mistura di distribuzioni.
- Dato  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^p$ , si presume che  $x_i$  derivi da una mistura finita di funzioni di densità di probabilità:

$$f(x_i; \Phi) = \sum_{g=1}^k \pi_g f(x_i | \theta_g),$$

$$\text{dove } \pi_g, g = 1, \dots, k \text{ s.t. } \pi_g > 0 \text{ e } \sum_{g=1}^k \pi_g = 1$$

- Stima dei parametri  $\Phi$  con l'approccio della massima verosimiglianza, eseguita applicando l'algoritmo EM.
- La partizione *soft* dei dati ottenuta mediante le probabilità a posteriori, utilizzando la regola del massimo a posteriori.

La mistura di distribuzione Gaussiana è la più utilizzata. La miscela finita di densità gaussiane (**FMG**) è quindi data da:

$$f(\mathbf{x}_i; \boldsymbol{\Phi}) = \sum_{g=1}^k \pi_g \phi(\mathbf{x}_i | \mu_g, \boldsymbol{\Sigma}_g),$$

- I cluster ellissoidali centrati sul vettore medio  $\mu_g$  sono generati dal modello precedente.
- Il parametro  $\boldsymbol{\Sigma}_g$  controlla le altre proprietà geometriche di ogni cluster.

Le FMG sono sensibili ai dati anomali.

Un **rough set**  $C_g$  è definito mediante la sua **lower approximation** (**LA**) e la sua **upper approximation** (**UA**), ovvero,  $\underline{A}(C_g)$  e  $\overline{A}(C_g)$ . LA e UA di  $C_g$  devono soddisfare le seguenti proprietà:

- (P1) Un oggetto  $x_i$  può far parte al massimo di una LA.
- (P2)  $x_i \in \underline{A}(C_g) \Rightarrow x_i \in \overline{A}(C_g)$ .
- (P3)  $x_i$  non fa parte di alcuna LA  $\Leftrightarrow x_i$  appartiene a due o più UA.

- Lo scopo è determinare se un oggetto appartiene alla UA o LA di un cluster.

Per ricavare le varie UA e LA sono utilizzate le seguenti regole:

- ① Se  $d(x_i, h_g)$  è il minimo, per  $1 \leq g \leq k$  e  $d(x_i, h_{g'})/d(x_i, h_g) \leq \psi$  per ogni coppia  $(g, g')$  allora  $x_i \in \bar{A}(C_g)$  e  $x_i \in \bar{A}(C_{g'})$ .
  - La proprietà (P3) viene dunque soddisfatta.
- ② Altrimenti,  $x_i \in \underline{A}(C_g)$  tale che  $d(x_i, h_g)$  è il minimo per  $1 \leq g \leq k$ .
  - Per la proprietà (P2),  $x_i \in \bar{A}(C_g)$ .
  - Segue che anche la proprietà (P1) è soddisfatta.

# Rough Clustering - Rough K-Means

- Un cluster è descritto da due approssimazioni *hard*, una LA e una UA (o regione di confine).
- Un dato ha due gradi di appartenenza bivalenti al cluster  $g$ , uno per la sua LA e uno per la UA:

$$\mu_{ig}^{LA} \in \{0, 1\} \quad \mu_{ig}^{UA} \in \{0, 1\}.$$

- LA: oggetti che appartengono al cluster. UA: oggetti che potrebbero appartenere al cluster o meno.
- Può essere considerato come un metodo *soft* a causa della regione di confine per gestire l'incertezza.
- I dati vengono assegnati alle LA o UA utilizzando le regole (1) o (2) precedenti.

# Rough Clustering - Rough K-Means

I centroidi sono calcolati come segue. Se le cardinalità  $|\underline{A}(c_g)|$  e  $|\overline{A}(c_g) - \underline{A}(c_g)|$  non sono uguali a 0, avremo che:

$$h_g = w_l \times \frac{\sum_{x_i \in \underline{A}(c_g)} x_i}{|\underline{A}(c_g)|} + w_u \times \frac{\sum_{x_i \in \overline{A}(c_g) - \underline{A}(c_g)} x_i}{|\overline{A}(c_g) - \underline{A}(c_g)|},$$

con  $w_l > 0$  e  $w_u > 0$  tali che  $w_l + w_u = 1$  (solitamente  $w_l > w_u$ ).

Se  $|\underline{A}(c_g)| = 0$  o  $|\overline{A}(c_g) - \underline{A}(c_g)| = 0$  di conseguenza uno dei due rapporti viene annullato.

# Implementazione in R - Ciclo Principale Rough K-Means

```
1 while ( !identical(old_upper_approx_matrix, upper_approx_matrix)
2     && iterations < max_iterations ) {
3
4     # Lower Approximation (LA) Matrix and UA-LA matrix (boundary)
5     lower_approx_matrix = assign_lower_approx(upper_approx_matrix)
6     boundary_matrix = upper_approx_matrix - lower_approx_matrix
7
8     # Calculate sums of observations in LA and boundary in every cluster
9     means_matrix_lower = crossprod(lower_approx_matrix, dataset)
10    means_matrix_boundary = crossprod(boundary_matrix, dataset)
11
12    # Update means matrix
13    for (i in 1:num_clusters) {
14
15        # Dividers means calculation, cardinalities of LA and UA-LA (boundary)
16        divider_lower_approx = sum(lower_approx_matrix[, i])
17        divider_boundary = sum(boundary_matrix[, i])
18
19        if (divider_lower_approx != 0 && divider_boundary != 0) {
20            means_matrix_lower[i,] = means_matrix_lower[i,] /
21                                    divider_lower_approx
22            means_matrix_boundary[i,] = means_matrix_boundary[i,] /
23                                       divider_boundary
24            means_matrix[i,] = weight_lower*means_matrix_lower[i,] +
25                               (1-weight_lower)*means_matrix_boundary[i,]
26        } else if (divider_boundary == 0) {
27            means_matrix[i,] = means_matrix_lower[i,] / divider_lower_approx
28        } else { # if(divider_lower_approx[,i]) == 0)
29            means_matrix[i,] = means_matrix_boundary[i,] / divider_boundary
30        }
31    }
```

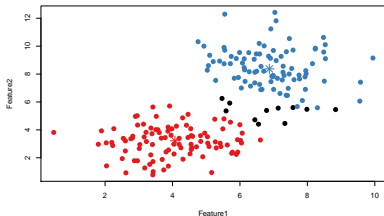


- Test su **FkM** (pacchetto R *fclust*), **PkM** (*ppclust*), **FMG** (*mclust*) e **RkM** (implementato).
- Valutazione delle prestazioni su 4 semplici dataset.
- 3 dataset sintetici 2d con 2 cluster: *DemoData*, *G2*, *Synth*.
- Dataset *Iris*, tre cluster, per sfruttarne le etichette di classe.

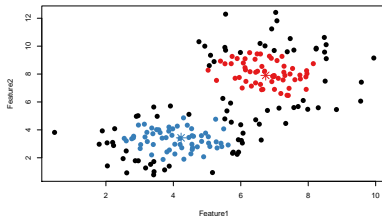
## Indici Utilizzati:

- **Indici Interno:** *Fuzzy Silhouette Index*.
- **Indici Esterni:**
  - *Purity*.
  - *Adjusted Rand Index* (ARI).
- Tempi di esecuzione.

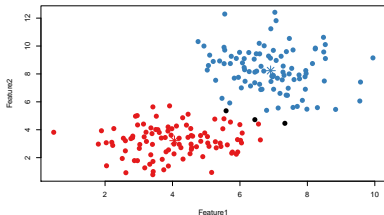
# Risultati del Clustering sul Dataset DemoDataC2D2a



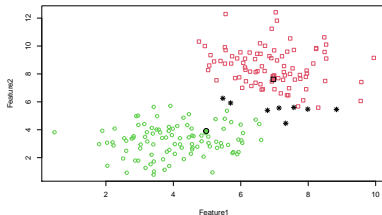
(a) Fuzzy K-Means



(b) Possibilistic K-Means

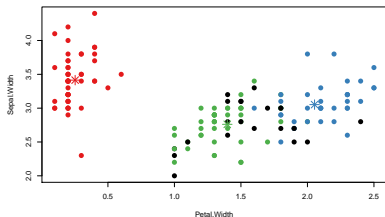


(c) Finite Mixture Gaussian Model

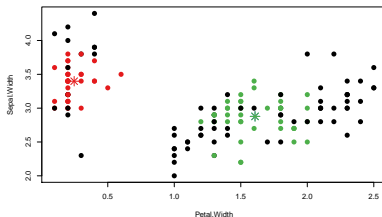


(d) Rough K-Means

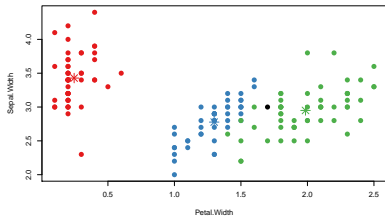
# Risultati del Clustering sul Dataset *Iris*



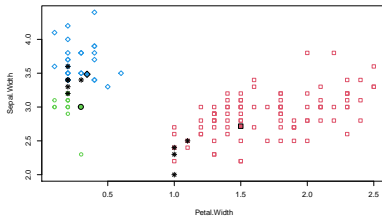
(a) Fuzzy K-Means



(b) Possibilistic K-Means



(c) Finite Mixture Gaussian Model



(d) Rough K-Means

# Indici di Valutazione e Tempi di Esecuzione

- Fuzzy Silhouette Index -	DemoDataC2D2a	G2	Synth	Iris
Fuzzy K-Means	0.8476	0.8525	0.8223	0.8091
Possibilistic K-Means	<b>0.8564</b>	<b>0.8633</b>	<b>0.8230</b>	<b>0.9538</b>
Finite Mixture Gaussian Model	0.8280	0.8346	0.8038	0.6579
Rough K-Means	0.8370	0.8388	0.8118	0.7532

- External Indices -	Purity	ARI
Fuzzy K-Means	0.9667	0.9039
Possibilistic K-Means	0.6667	0.5681
Finite Mixture Gaussian Model	0.9667	0.9039
Rough K-Means	0.5200	0.4413

- Execution Times -	DemoDataC2D2a	G2	Synth	Iris
Fuzzy K-Means	<b>0.0070</b>	<b>0.3590</b>	<b>0.1940</b>	<b>0.0050</b>
Possibilistic K-Means	0.7640	5.2400	4.7044	1.3802
Finite Mixture Gaussian Model	0.0240	1.765	0.3060	0.0680
Rough K-Means	0.1170	0.5850	0.3500	0.0140

# Grazie dell'Attenzione

**Link Progetto GitHub:**

`https://github.com/elia-mercantanti/  
soft-clustering-rough-kmeans`