Lezione 4

Status	Done
Attach file	4_Clustering.pdf

1. Clustering
    - Spazi metrici e funzione distanza
    - Spazi non euclidei
    - Tassonomia degli algoritmi di clustering
    - Bontà di un algoritmo
    - Problema della dimensionalità
        - Equidistanza dei punti
        - Ortogonalità dei vettori
2. Clustering gerarchico
    - Distanza tra cluster
        - **Misure Alternative di Distanza tra Cluster**
    - Dendogramma
    - Criteri di terminazione
    - Altri criteri di combinazione di cluster
        - Clustering agglomerativo (bottom-up)
        - **Clustering Divisivo (Top-Down)**
    - **Complessità del Clustering Gerarchico**
        - **Ottimizzazioni**
3. **Clustering Partizionale**
    - **K-Means**
        - **Scelta Iniziale dei k Centroidi (Approccio Greedy)**
        - **Funzione Obiettivo e Terminazione**
        - Scelta del valore di k
    - **Complessità e Limiti del K-Means**
    - **K-Means su Big Data**
4. **Clustering per Densità**
    - **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*)
        - **Parametri e Definizioni**

- Algoritmo DBSCAN
- Scelta dei Parametri e Complessità
- Vantaggi e Svantaggi di DBSCAN

OPTICS (*Ordering Points To Identify the Clustering Structure*)
- Reachability Score e Algoritm
- Reachability Plot
- Estrazione dei Cluster

HDBSCAN
- Mutual Reachability Distance e Graph
- Costruzione del Dendrogramma
- Estrazione dei Cluster Significativi (Stabilità)

1. Clustering

Il clustering è quel processo che consiste nel raggruppare un insieme di oggetti in gruppi detti cluster. In ogni cluster troverò oggetti “simili” tra loro (o con bassa distanza reciproca).

Il clustering è un processo “unsupervised”, poichè vuole suddividere un insieme di dati in n classi senza conoscenza a priori di quante e quali sono le classi e le loro etichette.

La classificazione è un processo “supervised” poichè si conoscono già a priori le n classi e le loro etichette. Quindi in questi casi si addestra un modello atto a predire una delle n classi da associare ai nuovi record.

Spazi metrici e funzione distanza

Gli oggetti da raggruppare sono punti appartenenti ad uno spazio metrico (S), ovvero dove è possibile definire una misura di distanza (D) per raggruppare i punti. Gli attributi rappresentano le coordinate.
Tale misura deve soddisfare le seguenti proprietà:

1. $D(X,Y) \geq 0 \quad \forall X,Y \in S$;
2. $D(X,Y) = D(Y,X) \, \forall X,Y \in S$ (proprietà simmetrica)

3. $D(X,Y) + D(Y,Z) \geq D(X,Z) \, \forall X, Y \in S$ (proprietà triangolare)

Posso definire delle distanze: prendo come esempio lo spazio euclideo.

1. Distanza euclidea $D(x,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
2. Distanza di Manhattan $D(x,y) = \sum_{i=1}^{n} |x_i - y_i|$
3. Norma L_r: $D(x,y) = (\sum_{i=1}^{1} |x_i - y_i|^r)^{\frac{1}{r}}$
4. Norma L_∞: $D(x,y) = max_{1 \leq i \leq n} |x_i - y_i|$

Spazi non euclidei

In uno spazio euclideo la media di un insieme di punti è sempre definita ed è un punto dello spazio, chiamato **centroide** (o centro geometrico).
In uno spazio **non euclideo** il concetto di centroide non è definito.
Se lo spazio non euclideo è metrico si può definire il concetto di **medoide** come il punto che minimizza la distanza media (o equivalentemente la somma) delle distanze dagli altri punti dell'insieme.
Esempi di spazi non euclidei sono gli spazi i cui oggetti sono **insiemi o stringhe**.

Posso definire anche qui delle distanze:

1. **Distanza di Jaccard**: dati due insiemi S e T $D(S,T) = 1 - \frac{|S \cap T|}{|S \cup T|}$
2. **Distanza di edit**: date due stringhe A e B, è il minimo numero di operazioni di cancellazione o inserzione di caratteri da effettuare partendo da A per ottenere **B**
3. **Distanza di Hamming**: dati due vettori A e B, è il numero di componenti in corrispondenza delle quali A e B differiscono

Tassonomia degli algoritmi di clustering

Una classificazione degli algoritmi di clustering si basa sull'approccio utilizzato.

1. **Metodi gerarchici o agglomerativi:** ciascun punto viene inizialmente posto in un cluster diverso e successivamente i cluster vengono combinati tra loro secondo una nozione di "vicinanza" definita opportunamente.
2. **Metodi di partizionamento:** l'insieme dei punti è partizionato in k cluster. Ogni punto è assegnato al cluster più "adatto" (K-means, BFR, CURE).

3. **Metodi basati sulla densità:** i cluster prodotti inizialmente vengono estesi fino a quando la densità (ovvero il numero di punti) in un intorno più o meno grande supera una certa soglia (DBSCAN, OPTICS, HDBSCAN).

Bontà di un algoritmo

La bontà di un algoritmo è legata a diversi fattori:

- Scalabilità;
- Capacità di trattare diversi tipi di attributi (numerici, binari, categoriali, ecc.);
- Capacità di ricercare cluster di forma diversa;
- Capacità di gestire dati con outlier e rumore (dati mancanti o errati);
- Insensibilità all'aggiunta di nuovi dati;
- Interpretabilità e usabilità dei risultati ottenuti;

Problema della dimensionalità

Spazi euclidei ad elevata dimensionalità sono affetti dal problema della dimensionalità)

- Quasi tutte le coppie di punti sono equidistanti e lontani tra loro;
- Quasi tutte le coppie di vettori sono quasi ortogonali.

Queste proprietà rendono molto più complicato il clustering.
Se consideriamo due punti X e Y in uno spazio n-dimensionale la distanza euclidea sarà:

$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Equidistanza dei punti

Essendo n è molto grande, è molto probabile che, in almeno una componente i, $|x_i - y_i|$ sia molto vicino al valore 1. Da ciò segue che:

1. D(X,Y) è almeno pari a 1. Pertanto, escludendo una frazione trascurabile di coppie, è possibile individuare un limite inferiore che risulta essere superiore a

per 1.

2. D(X,Y) è al massimo pari a √n (nel caso in cui i si ha $|x \ \forall \ i - yi| = 1$). Pertanto, escludendo un frazione trascurabile di coppie, tutte le altre coppie hanno un limite superiore che risulta essere inferiore a √n.

Solo una frazione trascurabile di coppie hanno una distanza vicina ai due limiti. La maggior parte, invece, ha una distanza vicina alla media e pari a √n /3

Ortogonalità dei vettori

Siano X e Y due punti in uno spazio n-dimensionale.
Vogliamo misurare l'angolo XY, mediante la distanza del coseno:

$$D(x,y) = \arccos \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2}}$$

Come è possibile notare, il denominatore è formato da quantità positive. Il numeratore, invece, sia positive che negative.
Si dimostra che, al crescere di n, il denominatore cresce linearmente in n, mentre il numeratore assume valore atteso 0 con una deviazione standard pari a √n.
Pertanto angolo di 90 gradi.

2. Clustering gerarchico

a. Assegna ciascun punto ad un cluster separato;
b. Unisci i due cluster **più vicini** in un unico cluster, sulla base di una nozione di distanza tra cluster;
c. Ripeti il passo b) finché non è soddisfatto un criterio di terminazione.

Distanza tra cluster

- **Spazio Euclideo:** Si usano i **centroidi** (media dei punti) dei cluster per calcolare la distanza e fonderli. Ad ogni fusione, bisogna ricalcolare il centroide del nuovo cluster.
- **Spazio Non Euclideo:** Poiché il centroide non è definito, si usa il **clustoide**, solitamente il **medoide**. Il medoide è il punto del cluster tale che la somma

delle distanze dagli altri punti è minima.

Misure Alternative di Distanza tra Cluster

- **Single link:** Distanza più piccola tra un elemento di X e uno di Y
- **Complete link:** Distanza massima tra un elemento di X e uno di Y.
- **Average link:** Media delle distanze tra tutte le coppie di punti di Y e Y.
- **Medoid distance:** Distanza tra i medoidi di X e Y.
- Altri criteri considerano la coppia di cluster la cui unione ha **raggio minimo** (distanza massima tra centroide e un punto del cluster) o **diametro minimo** (distanza massima tra due punti qualsiasi del cluster)

Dendogramma

Al clustering gerarchico è associato un albero chiamato **dendrogramma**, che descrive il modo in cui i cluster sono combinati. Tagliando il dendrogramma a un certo livello, i sottoalberi risultanti rappresentano i cluster prodotti a quel punto della computazione.

Criteri di terminazione

L'algoritmo può terminare quando:

- Si raggiunge un numero predefinito di cluster.
- Una successiva fusione di cluster produce un cluster inadeguato (ad es. la distanza media dei punti dai rispettivi centroidi cresce troppo).

Altri criteri di combinazione di cluster

Clustering agglomerativo (bottom-up)

(già descritto) Ogni punto forma inizialmente un cluster e ad ogni passo i due cluster più vicini vengono fusi in unico cluster.

Clustering Divisivo (Top-Down)

Gli algoritmi possono seguire anche un approccio **divisivo** (*top-down*), dove tutti i punti sono inizialmente nello stesso cluster. Ad ogni passo, un cluster viene suddiviso in due cluster più piccoli in base a criteri di ottimalità della separazione. Le metriche e i criteri dell'approccio agglomerativo rimangono validi anche per l'approccio divisivo.

Complessità del Clustering Gerarchico

Al primo passo, si calcola la distanza tra ogni coppia di cluster per scegliere la migliore da unire, con una complessità di $O(n^2)$. L'algoritmo procede per n iterazioni, quindi la complessità totale è $O(n^3)$.

Ottimizzazioni

È possibile ridurre la complessità a $O(n^2 \log n)$ utilizzando le **code di priorità**.

> 🐒 La coda di priorità permette di ottenere il minimo in tempo costante e consente inserimenti/cancellazioni in tempo $O(\log n)$.

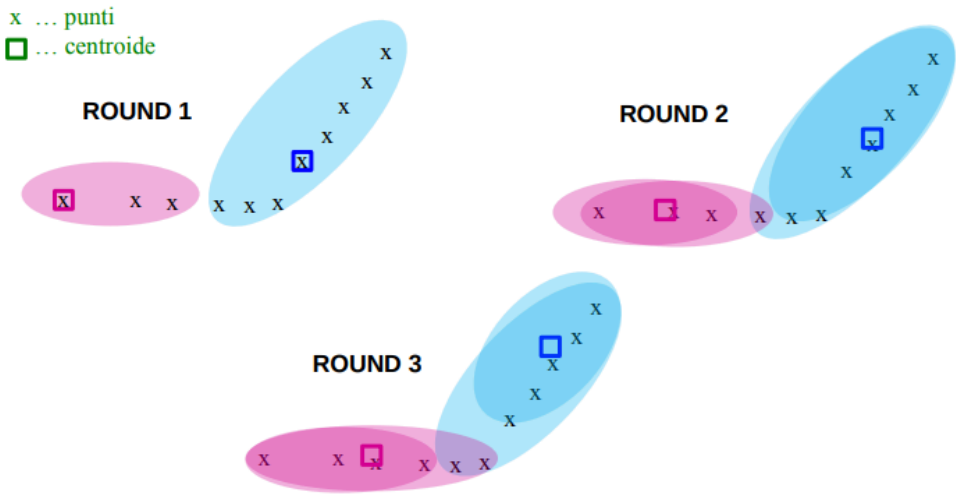
Ad ogni iterazione, si tolgono le distanze dei due cluster da fondere e si inseriscono le distanze del nuovo cluster verso gli altri, con un costo per iterazione di $O(\log n)$.

3. Clustering Partizionale

K-Means

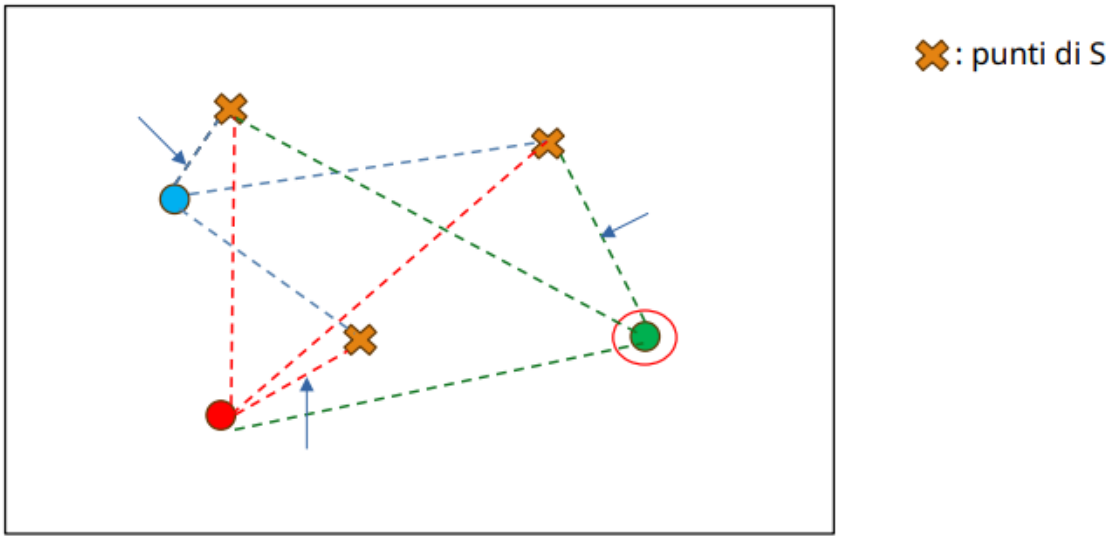
OK **K-means** è una classe di algoritmi che lavora su **spazi euclidei** e assume la conoscenza *a priori* del numero k di cluster.

1. Scegli inizialmente k punti come centroidi, con alta probabilità di finire in cluster diversi.
2. Assegna ciascun punto del dataset al cluster corrispondente al centroide ad esso più vicino.
3. Aggiorna il centroide per ogni cluster.
4. Sposta i punti che risultano essere più vicini a un nuovo centroide.
5. Ripeti i passi 2)-4) finché non si osserva uno spostamento di punti e i centroidi si stabilizzano, o quando una funzione obiettivo si mantiene stabile.



Scelta Iniziale dei k Centroidi (Approccio Greedy)

1. Seleziona il primo punto in modo casuale e aggiungilo all'insieme S dei punti selezionati.
2. Aggiungi a il punto P che **massimizza la distanza minima** di P dai punti in S.
3. Ripeti il passo 2) finché $|s| < k$.



Funzione Obiettivo e Terminazione

Il criterio di arresto si basa sulla funzione obiettivo E che è la somma dei quadrati delle distanze dei punti dai rispettivi centroidi C_i:

$$E = \sum_{i=1}^{k} \sum_{X \in P_i} ||X - C_1||^2$$

L'algoritmo termina quando la differenza tra i valori di E in due iterazioni consecutive scende sotto una soglia.

Scelta del valore di k

Quando k non è noto, l'algoritmo viene eseguito per diversi valori di k, e si sceglie il valore per cui il clustering è migliore, misurando la qualità dei cluster tramite la distanza media dei punti dai rispettivi centroidi.

- **Metodo Elbow (Gomito):** All'aumentare di k, la distanza media dai centroidi diminuisce, prima drasticamente e poi in modo più contenut. Il valore ideale di k è quello che corrisponde al **"gomito"** della curva, a partire dal quale la distanza media varia poco. Il punto può essere individuato matematicamente con una ricerca binaria.
- **Metodo Silhouette:** Misura in media quanto ciascun punto è vicino al cluster cui è stato assegnato rispetto agli altri cluster, calcolando lo **score di**

silhouette $s(X)$.

- $a(X)$: distanza media di X da tutti gli altri punti del suo cluster .
- $b(X)$: minima distanza media di dagli altri cluster.
- $s(X) = \frac{b(X)-a(X)}{max\{a(X),b(X)\}} \in [-1, 1]$
- $s(X) > 0$ l'assegnazione è buona (valori > 0.5 sono considerati buoni), se è negativo l'assegnazione è non buona, e valori prossimi a 0 indicano che X è a metà strada tra due o più cluster.
- Lo score di silhouette del clustering è la media di $s(X)$ su tutti i punti. Si seleziona il che lo massimizza.

Complessità e Limiti del K-Means

- **Complessità:** $O(tkn)$ dove t è il numero di *round*, k il numero di cluster, e n il numero di elementi. È più efficiente del clustering gerarchico.
- **Limiti:**
  - Spesso converge a una soluzione **localmente ottimale**.
  - Non è in grado di trovare cluster con **forma non convessa** o di dimensioni molto diverse.
  - È molto **sensibile al rumore e agli outlier** perché possono influenzare sostanzialmente la posizione dei centroidi.
  - Richiede la specifica di k.

K-Means su Big Data

Per grosse quantità di dati che non possono risiedere in memoria, si usano varianti del K-means, tra cui gli algoritmi **BFR** e **CURE**. Questi algoritmi usano statistiche per rappresentare i cluster in modo compatto, ottimizzando l'uso della RAM e gestendo gli outlier.

4. Clustering per Densità

DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*)

L' algoritmo si basa su densità in cui ogni cluster è una regione di punti **connessi per densità** con densità sufficientemente alta. Una regione densa è una regione con un numero di punti sufficientemente elevato in un intorno limitato.

Parametri e Definizioni

DBSCAN ha due parametri:

- ϵ **(raggio):** legato alla grandezza dei cluster.
- **MinPts:** numero minimo di punti che un cluster deve avere.
- **ϵ-intorno di Q ($N_\epsilon(Q)$):** Insieme dei punti a distanza al più ϵ da Q
- **Punto P direttamente raggiungibile per densità da Q:** Se $P \in N_\epsilon(Q)$ e Q è un **core-point** (ovvero $|N_\epsilon(Q)| \geq \mathrm{MinPts}$) .
- **Punto raggiungibile per densità da Q:** Se esiste una catena di punti $Q, A_2, \ldots, P$ tale che A_{i+1} è direttamente raggiungibile per densità da A_i.
- **Punto P connesso per densità a Q:** Se esiste un punto O tale che sia P che sono raggiungibili per densità da O.
- **Cluster (in DBSCAN):** Un insieme **massimale** di punti **connessi per densità**.

Algoritmo DBSCAN

1. Scegli un punto P random non ancora visitato.
2. Calcola l' ϵ-intorno S di P. Se $|S|[cite_start] \geq \mathrm{MinPts}$, crea un nuovo cluster C (Punto 3); altrimenti marca P come **rumore (outlier)** e torna al Punto 1.
3. Aggiungi P e il suo ϵ-intorno a C.
4. Aggiungi ricorsivamente ϵ-intorn di punti in C (ovvero punti raggiungibili per densità da) finché possibile.
5. Ripeti dal Punto 1 fino a quando non tutti i punti sono stati visitati.

Scelta dei Parametri e Complessità

- **MinPts:** Si consiglia $\text{MinPts} \geq n + 1$, dove n è la dimensione dello spazio. Si consiglia di fissare MinPts molto maggiori di n+1 con dataset molto grandi o in presenza di molto rumore.
- ϵ **(raggio):** Si usa il **metodo k-distance**, dove $k = \text{MinPts}$
  1. Trova la distanza dal k-esimo vicino più vicino per ogni punto115.
  2. Ordina le distanze in modo decrescente116.
  3. Cerca il **"gomito"** nel grafico Distanza vs Punti Ordinati.

L'ordinata del punto in cui la curva "piega" maggiormente è il valore ottimale di ϵ. Valori di ϵ troppo bassi non clusterizzano molti punti, mentre valori troppo alti creano cluster troppo grandi.

- **Complessità:** Utilizzando strutture indicizzate (ad esempio **R-tree**), l' ϵ-intorno di un punto è calcolato in $O(\log n)$. Poiché l' ϵ-intorno è calcolato una sola volta per ogni punto, la complessità totale è $O(n \log n)$.

Vantaggi e Svantaggi di DBSCAN

- **Vantaggi:** Non richiede la conoscenza di k, può trovare cluster di **forma arbitraria**, contempla la nozione di **outlier**, l'assegnamento è poco influenzato dall'ordine di esame dei punti.
- **Svantaggi:** La scelta dei due parametri dipende molto dal tipo di dati, non è in grado di individuare cluster che hanno **notevoli differenze nella densità**.

OPTICS (*Ordering Points To Identify the Clustering Structure*)

Algoritmo che si basa su densità che risolve il problema dell'identificazione di cluster a densità diversa. Non usa un ϵ predefinito per il raggio dei cluster, ma lo adatta in base alla **"densità locale"** dei punti in una regione.

Reachability Score e Algoritm

Per catturare cluster di diversa dimensione, OPTICS valuta se due punti sono vicini rispetto al contesto in cui si trovano. L'algoritmo calcola per ogni punto un **reachability score** che valuta la raggiungibilità del punto a partire dai punti vicini.

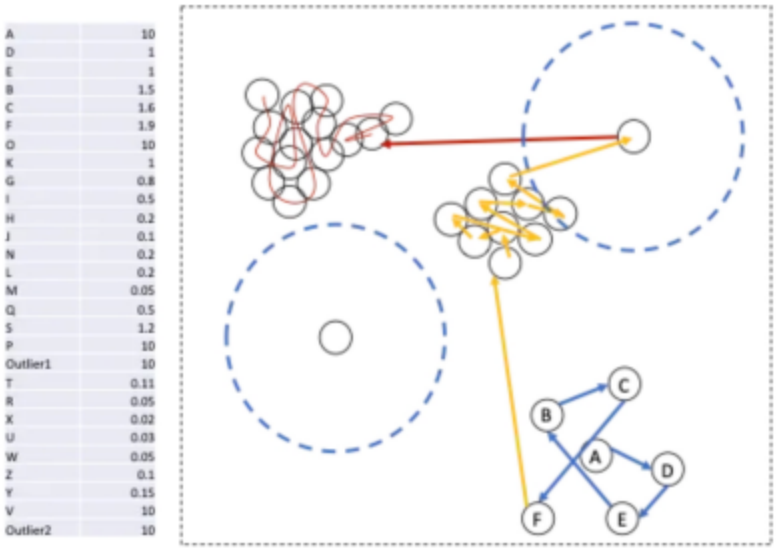
Un *reachability score* più basso indica che il punto è più vicino ad altri e quindi più facilmente raggiungibile.

Calcolo dei Reachability Score:

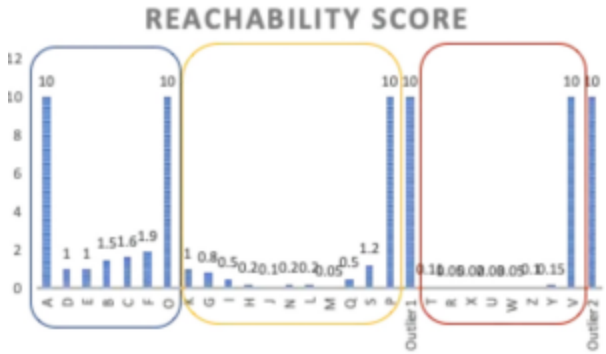
1. IInizializza il ReachScore di ogni punto a un valore di default sufficientemente alto.
2. Sia P un punto random.
3. Per ogni punto X nell' ϵ-intorno di P, calcola D(X,P). Se D(X,P)<ReachScore(X), assegna al *reachability score* il valore D(X,P).
4. Imposta come il punto (tra quelli non ancora esaminati come centri) con il *reachability score* più basso.
5. Ripeti i passi 3)-4) finché non ci sono più punti da esaminare come centri.

Reachability Plot

La lista finale degli score viene ordinata in ordine di esaminazione dei nodi. Il **reachability plot** è un istogramma ottenuto plottando i *reachability score* in questo ordine.



Gli **"avvallamenti"** nel plot corrispondono ai cluster; più sono profondi, più il cluster è denso.



Estrazione dei Cluster

Due possibili soluzioni per estrarre i cluster dal *reachability plot*:

1. Unire nello stesso cluster tutti i punti con *reachability score* $\leq \epsilon$ con il loro predecessore nel plot (Equivalente al DBSCAN).
2. Identificare i punti A e B in cui si ha un decremento (o un incremento) dello score di raggiungibilità di un fattore $1 - \delta$ e unire nello stesso cluster A,B e tutti i punti compresi tra essi (a patto che siano in totale almeno MinPts).

HDBSCAN

HDBSCAN è un algoritmo basato su densità che richiede come unico parametro **MinPts**. Costruisce un **dendrogramma** dei cluster ottenuti al variare del raggio ϵ, da cui estrae l'insieme dei cluster più significativi basandosi su una misura di **"stabilità"**.

Mutual Reachability Distance e Graph

- **Core distance di X ($d_{core}(X)$):** Distanza che X ha rispetto al MinPts-esimo punto più vicino.
- **Mutual Reachability distance ($d_{reach}(X, Y)$):** Definito come $d_{reach}(X, Y) = \max\{d_{core}(X), d_{core}(Y), d(X, Y)\}$. Serve ad "amplificare" la distanza tra due punti che si trovano in regioni sparse, facilitando l'identificazione di cluster meno densi.
- **Mutual Reachability graph (G_{MinPts}):** Grafo pesato in cui i nodi sono gli oggetti e l'arco ha peso pari alla *Mutual Reachability distance*.

Costruzione del Dendrogramma

Al variare di ϵ, si ottengono diversi partizionamenti dello spazio in cluster. Per la costruzione del dendrogramma:

1. Costruisci il **Minimum Spanning Tree (MST)** di G_{MinPts}.
2. Rimuovi archi dal MST in ordine decrescente di peso (archi con peso uguale vanno rimossi simultaneamente) ed estrai le componenti connesse ad ogni passo.

Estrazione dei Cluster Significativi (Stabilità)

L'idea è di catturare i cluster che **"resistono più a lungo possibile"** al diminuire di ϵ (risultano più stabili). Si definisce $\lambda = \frac{1}{\epsilon}$.

- **Stabilità del cluster C:** $S(C) = \Sigma_{X \in C}(\lambda_{max}(X, C) - \lambda_{min}(C))$.
  - $\lambda_{\min}(C)$: valore di λ per cui C appare per la prima volta.
  - $\lambda_{\max}(X, C)$: valore di λ per cui X esce da C.

Il problema di estrarre i k cluster più stabili evitando sovrapposizioni si risolve in maniera **bottom-up** partendo dalle foglie del dendrogramma. Per ogni cluster C, si decide se è meglio selezionare C al posto dei suoi discendenti confrontando la sua stabilità S(C) con la somma delle stabilità dei suoi figli ($S'(C_{i,left}) + S'(C_{i,right})$) .