UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

Fondamenti di Intelligenza Artificiale

Lezione 18 - Clustering

Clustering

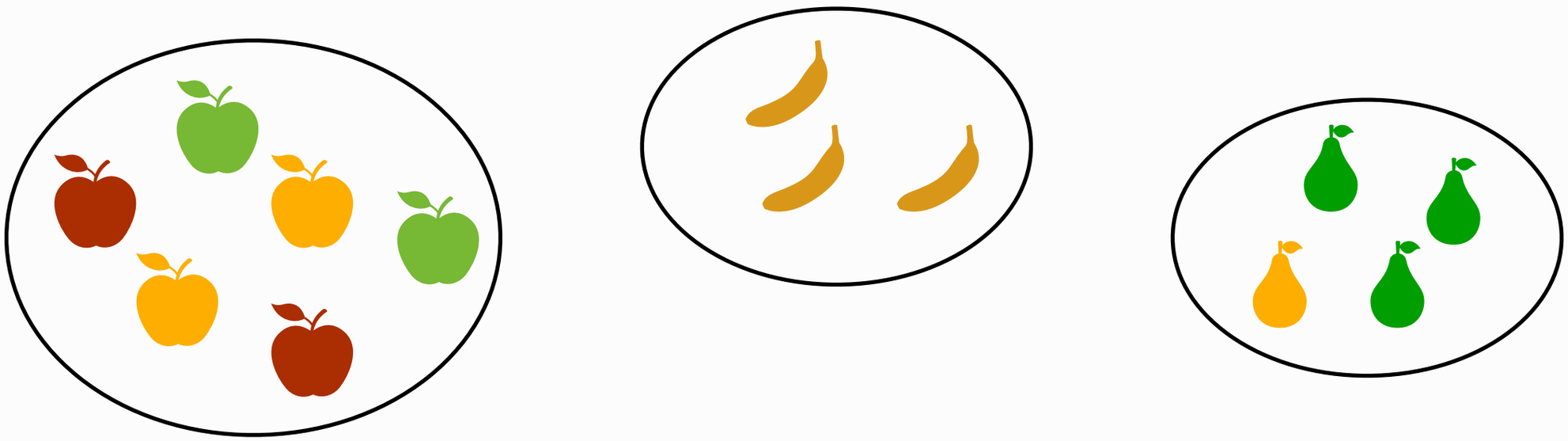
Problemi di raggruppamento

Clustering: Task in cui l'obiettivo è raggruppare oggetti in gruppi che abbiano un certo grado di omogeneità ma che, al tempo stesso, abbiamo un certo grado di eterogeneità rispetto agli altri gruppi.

I problemi di clustering sono istanze di problemi di apprendimento *non* supervisionato.

In maniera simile alla classificazione, un problema di clustering ha l'obiettivo di *classificare* i dati, ma *senza assegnare loro un'etichetta*. In questo caso, infatti, non abbiamo classi predefinite, ma ogni cluster può essere interpretato come *una classe di oggetti simili*, ovvero avanti simili caratteristiche.

Gli algoritmi di clustering si basano su un concetto fondamentale, che è quello della similarità, la quale è interpretata in maniera diversa quando parliamo di somiglianza intra-gruppo ed inter-gruppo.



Clustering

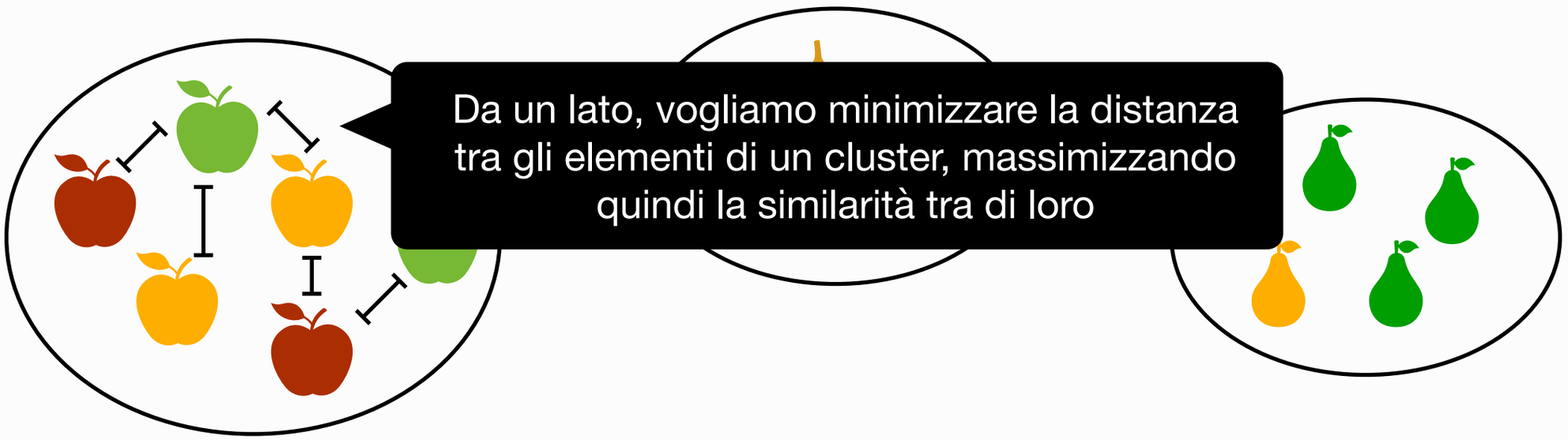
Problemi di raggruppamento

Clustering: Task in cui l'obiettivo è raggruppare oggetti in gruppi che abbiano un certo grado di omogeneità ma che, al tempo stesso, abbiamo un certo grado di eterogeneità rispetto agli altri gruppi.

I problemi di clustering sono istanze di problemi di apprendimento *non* supervisionato.

In maniera simile alla classificazione, un problema di clustering ha l'obiettivo di *classificare* i dati, ma *senza assegnare loro un'etichetta*. In questo caso, infatti, non abbiamo classi predefinite, ma ogni cluster può essere interpretato come *una classe di oggetti simili*, ovvero avanti simili caratteristiche.

Gli algoritmi di clustering si basano su un concetto fondamentale, che è quello della similarità, la quale è interpretata in maniera diversa quando parliamo di somiglianza intra-gruppo ed inter-gruppo.



Clustering

Problemi di raggruppamento

Clustering: Task in cui l'obiettivo è raggruppare oggetti in gruppi che abbiano un certo grado di omogeneità ma che, al tempo stesso, abbiamo un certo grado di eterogeneità rispetto agli altri gruppi.

I problemi di clustering sono istanze di problemi di apprendimento *non* supervisionato.

In maniera simile alla classificazione, un problema di clustering ha l'obiettivo di *classificare* i dati, ma *senza assegnare loro un'etichetta*. In questo caso, infatti, non abbiamo classi predefinite, ma ogni cluster può essere interpretato come *una classe di oggetti simili*, ovvero avanti simili caratteristiche.

Gli algoritmi di clustering si basano su un concetto fondamentale, che è quello della similarità, la quale è interpretata in maniera diversa quando parliamo di somiglianza intra-gruppo ed inter-gruppo.

Clustering

Problemi di raggruppamento: La bontà del clustering

Come facilmente intuibile, la qualità di un algoritmo di clustering dipenderà dalla misura di similarità utilizzata e dall'algoritmo stesso.

Da un punto di vista *esterno*, la qualità del clustering è misurato in base alla sua abilità di scoprire alcuni o tutti i pattern nascosti, ovvero le caratteristiche che legano elementi simili —> Questo è però qualcosa che non potremo calcolare in maniera automatica, poiché non abbiamo le etichette di partenza!

La nozione di clustering può però essere ambigua. Il problema frequente è quello dell'identificazione del *numero ideale di cluster* che un algoritmo dovrà generare… d'altro canto, essendo un apprendimento non supervisionato, gli algoritmi di clustering non hanno a disposizione delle informazioni su quante classi dovranno produrre.

Altre proprietà che rendono un algoritmo di clustering *buono* possono essere la scalabilità, robustezza agli outlier, o interpretabilità dei risultati.

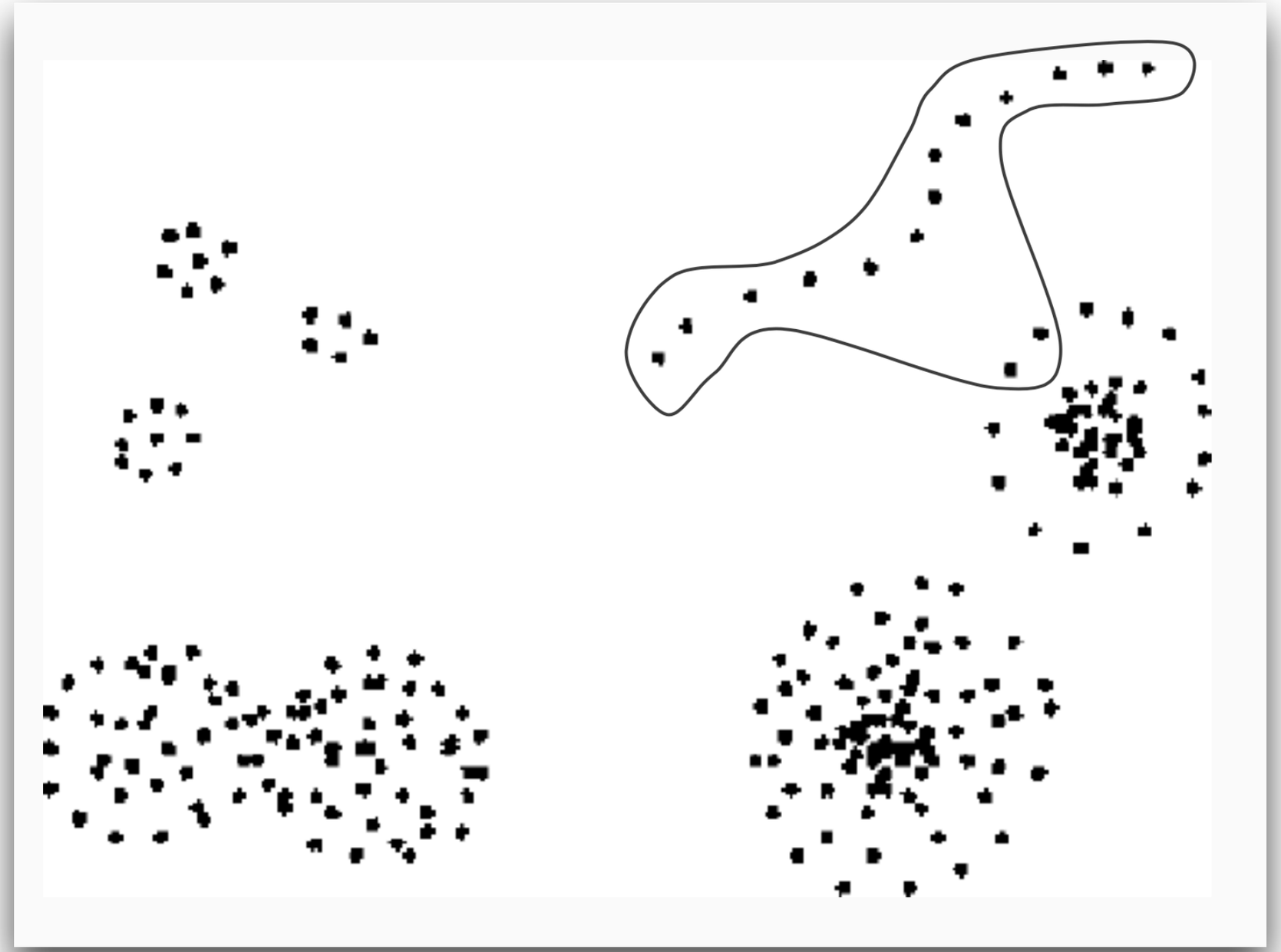
Al tempo stesso, la nozione di similarità può essere ambigua. Cosa significa che due elementi sono simili? Da che punto di vista?

In tutto ciò, non dimentichiamo che esistono diversi tipi di dati (strutturati, non strutturati, ecc.) e la misura di similarità dipende chiaramente anche da questo.

Per non farci mancare niente, dovremmo anche considerare la *dimensionalità dei dati*: quando gli attributi di un elemento aumentano, i dati diventano sempre più sparsi e difficili da raggruppare.

Problemi di raggruppamento: La bontà del clustering

Ad esempio, osserviamo i dati in figura. Il numero di cluster dipende dalla risoluzione con cui vediamo i dati… quanti cluster vediamo in figura? 5, 8, 10 o più?



Clustering

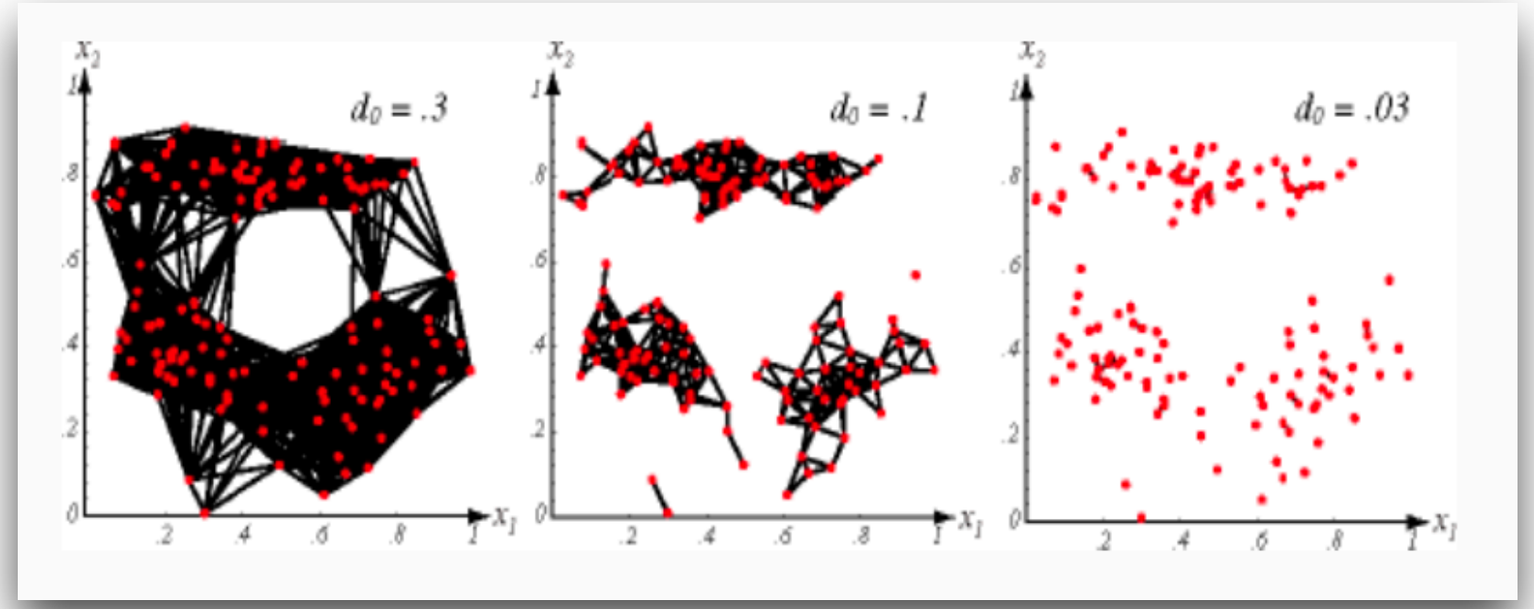
Problemi di raggruppamento: Misure di similarità

La più ovvia misura di similarità (o di diversità) tra due pattern è la distanza fra essi. **Ma attenzione:** Non sempre la distanza tra due pattern è significativa per indicare diversità!

Se la distanza rappresenta una buona misura di diversità, allora possiamo imporre che la distanza tra due pattern nello stesso cluster sia significativamente più piccola della distanza tra due pattern appartenenti a cluster diversi.

Così facendo, fare clustering sarebbe semplicissimo, poiché potremmo definire una soglia sulla distanza e raggruppare i pattern al di sotto di tale soglia.

La scelta della soglia influenzerebbe sia il numero che la forma dei cluster.



Il punto però diventa rispondere alla domanda: Qual è la giusta soglia da utilizzare?

Problemi di raggruppamento: Misure metriche

Per rispondere a questa domanda, iniziamo con il definire una misura metrica, ovvero una quantità calcolabile di distanza tra più elementi di una popolazione.

Misura metrica: Dato un insieme *S* di campioni, una distanza *d* è metrica se valgono le seguenti proprietà:

1. *Identità*: $\forall x \in S, d(x, x) = 0$;
2. *Positività*: $\forall x \neq y \in S, d(x, y) > 0$;
3. *Simmetria*: $\forall x, y \in S, d(x, y) = d(y, x)$;
4. *Disuguaglianza triangolare*: $\forall x, y, z \in S, d(x, z) \leq d(x, y) + d(y, z)$.

Una misura è detta *semi-metrica* quando le proprietà 1, 2 e 3 sono soddisfatte, mentre è detta *pseudo-metrica* quando 1, 3 e 4 sono soddisfatte.

Un esempio classico di metrica è la distanza euclidea, che calcola la distanza tra due punti in un piano cartesiano:

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

A seconda delle esigenze, potremmo dover ricorrere ad altre tipologie di similarità. Ad esempio, nel caso di testi, non potremmo neanche calcolare la distanza euclidea.

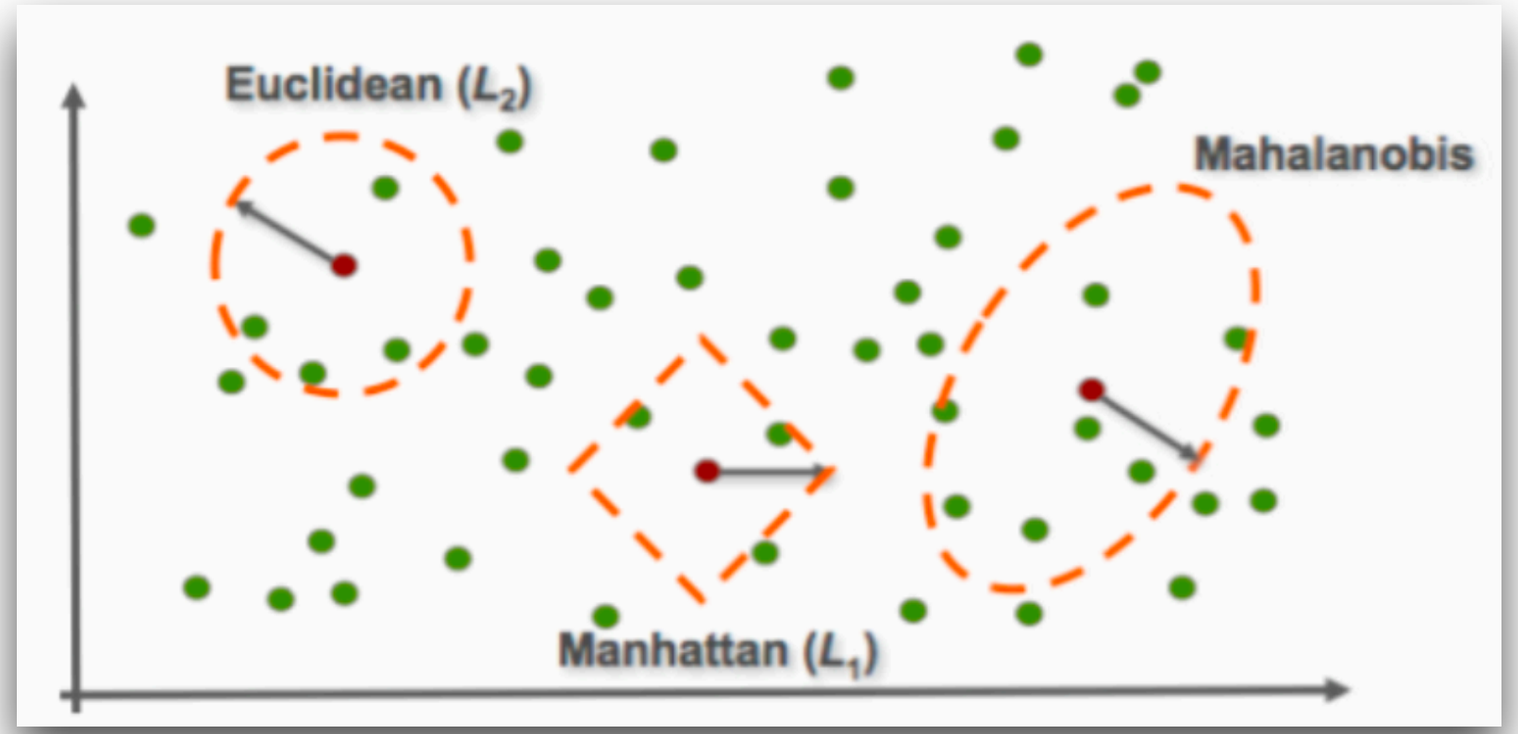
Problemi di raggruppamento: Misure metriche

Altre metriche di similarità generalmente riconosciute sono le seguenti:

Distanza di Manhattan — La distanza tra due punti è la somma del valore assoluto delle differenze delle loro coordinate.

Distanza di Mahalanobis — E' una forma di distanza standardizzata, in cui si tiene conto non solo della diversa dispersione delle variabili, ma anche della loro correlazione.

Potete anche vedere le tre distanze da un punto di vista grafico nel seguente modo:



Clustering

Problemi di raggruppamento: Misure metriche

Altre metriche di similarità generalmente riconosciute sono le seguenti:

Distanza di Manhattan — La distanza tra due punti è la somma del valore assoluto delle differenze delle loro coordinate.

Distanza di Mahalanobis — E' una forma di distanza standardizzata, in cui si tiene conto non solo della diversa dispersione delle variabili, ma anche della loro correlazione.

Distanza di Jaccard — L'indice misura la similarità tra insiemi campionari, ed è definito come la dimensione dell'intersezione divisa per la dimensione dell'unione degli insiemi campionari.

La distanza di Jaccard è particolarmente interessante poiché può essere facilmente utilizzata per valutare non solo la similarità tra due insiemi numerici, ma anche tra due stringhe. Vediamo perché:

$$J_\delta(X, Y) = \frac{X \cap Y}{X \cup Y}$$

S_1 = *"Mi piace andare al mare."*

S_2 = *"Ieri sono stato al mare."*

Quanto simili sono queste due stringhe?

$$\longrightarrow \quad \frac{\{\text{al, mare}\}}{\{\text{mi, piace, andare, al, mare, ieri, sono, stato}\}} = \frac{2}{8} = 0{,}25$$

Clustering

Problemi di raggruppamento: Misure metriche

Altre metriche di similarità generalmente riconosciute sono le seguenti:

Distanza di Manhattan	La distanza tra due punti è la somma del valore assoluto delle differenze delle loro coordinate.
Distanza di Mahalanobis	E' una forma di distanza standardizzata, in cui si tiene conto non solo della diversa dispersione delle variabili, ma anche della loro correlazione.
Distanza di Jaccard	L'indice misura la similarità tra insiemi campionari, ed è definito come la dimensione dell'intersezione divisa per la dimensione dell'unione degli insiemi campionari.
Distanza di Hamming	L'indice misura il numero di sostituzioni necessarie per convertire una stringa nell'altra.
Distanza di Levenshtein	L'indice misura il numero minimo di modifiche elementari (cancellazione, sostituzione, inserimento) che consentono di trasformare una stringa X in una stringa Y.

Sebbene ci siano molte altre metriche, *non solo strutturali*, è impossibile (ed inutile) dire quale sia meglio di un'altra. Questo è dipendente dal problema e, generalmente, il clustering ottenuto con varie distanze viene sperimentato e validato a posteriori.

Clustering

Problemi di raggruppamento: Criteri da adottare

Per identificare la "giusta" soglia, dobbiamo poi scegliere il criterio da utilizzare in fase di ottimizzazione dei cluster.

Più formalmente, supponiamo di avere un insieme $D = \{x_1, x_2, \ldots x_n\}$ composto da n campioni, che vogliamo partizionare in esattamente k insiemi disgiunti $D_1, D_2, \ldots, D_k$.

Ogni sottoinsieme rappresenta un cluster, con i campioni nello stesso cluster che sono per qualche motivo più simili l'un l'altro rispetto ai campioni negli altri cluster.

Come vedremo anche in seguito, il criterio più semplice e usato è quello della *somma degli errori quadrati* (squared sum estimate).

Supponiamo che l'insieme dato di n campioni sia stato partizionato in qualche modo in k cluster $D_1, D_2, \ldots, D_k$.

Supponiamo inoltre che n_i sia il numero di campioni in D_i e che m_i sia la media aritmetica dei campioni, ovvero:

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

Allora, la somma degli errori quadrati sarà uguale a:

$$j_e = \sum_{i=1}^{k} \sum_{x \in D_i} |x - m_i|^2$$

Per un dato cluster D_i, il vettore delle medie m_i (detti **centroidi**) è la migliore rappresentazione dei campioni nel dataset.

Clustering

Problemi di raggruppamento: Algoritmi

Gli algoritmi di clustering si suddividono innanzitutto in varie tipologie:

- **Esclusivi vs non esclusivi.** Un algoritmo di clustering è *esclusivo* se ogni pattern appartiene solo ad un cluster. Al contrario, se ogni pattern può essere assegnato a più di un cluster, allora parleremo di algoritmo *non esclusivo*.

- **Gerarchico vs partizionale.** Un algoritmo di clustering è detto *gerarchico* se mira a costruire delle gerarchie di cluster, anche dette sequenze innestate di partizioni. Al contrario, un algoritmo *partizionale* effettua solo una partizioni dei pattern.

Gli algoritmi di clustering si suddividono poi in categorie:

- **Agglomerativi vs divisivi.** Un algoritmo di clustering è *agglomerativo* se parte da cluster atomici che punta ad unire iterativamente in cluster più grandi. Un algoritmo *divisivo* parte invece da ampi cluster per dividerli poi in cluster più piccoli.

- **Seriali vs simultanei.** Un algoritmo di clustering è *seriale* se elabora i pattern uno alla volta. Un algoritmo è *simultaneo* se invece elabora i pattern insieme.

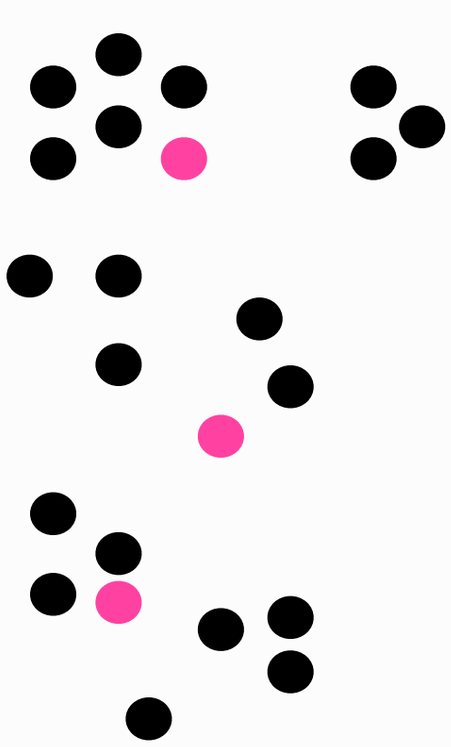
- **Graph-theoretic vs algebrici.** Un algoritmo di clustering è *graph-theoretic* se elabora i pattern sulla base della loro collegabilità. Un algoritmo è *algebrico* se invece elabora i pattern sulla base di criteri di errore.

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.



Step #1. Seleziona k centroidi in maniera casuale —> k pattern sono eletti come rappresentanti;

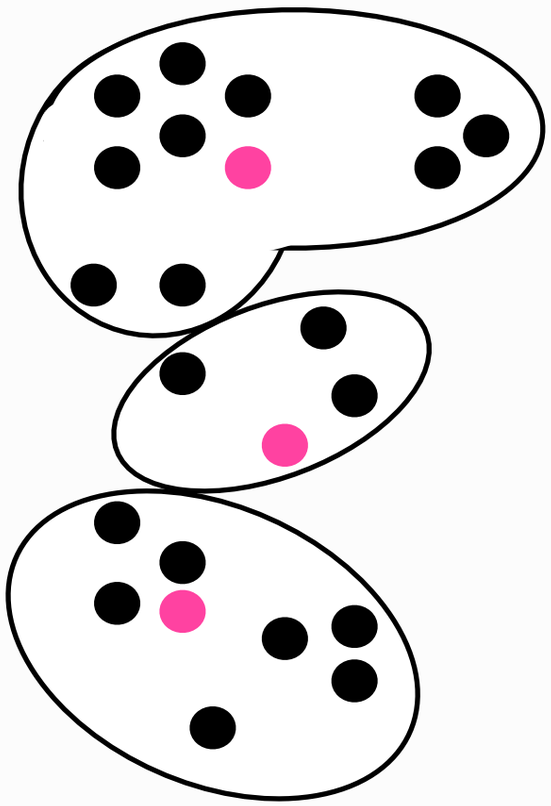
Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.



Step #1. Seleziona k centroidi in maniera casuale —> k pattern sono eletti come rappresentanti;

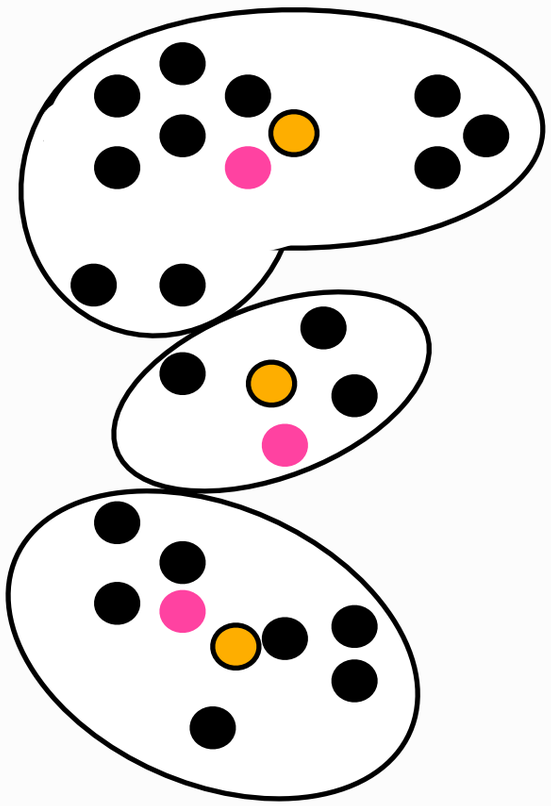
Step #2. Genera un partizionamento assegnando ogni campione al centroide più vicino.

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.



Step #1. Seleziona k centroidi in maniera casuale —> k pattern sono eletti come rappresentanti;

Step #2. Genera un partizionamento assegnando ogni campione al centroide più vicino.

Step #3. Calcola i nuovi centroidi del cluster, considerando la media dei valori dei cluster generati al punto #2.

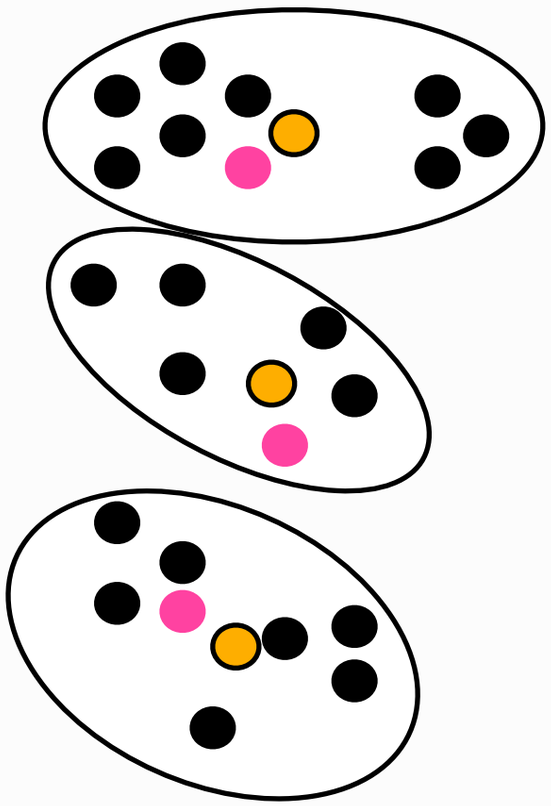
Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.



Step #1. Seleziona k centroidi in maniera casuale —> k pattern sono eletti come rappresentanti;

Step #2. Genera un partizionamento assegnando ogni campione al centroide più vicino.

Step #3. Calcola i nuovi centroidi del cluster, considerando la media dei valori dei cluster generati al punto #2.

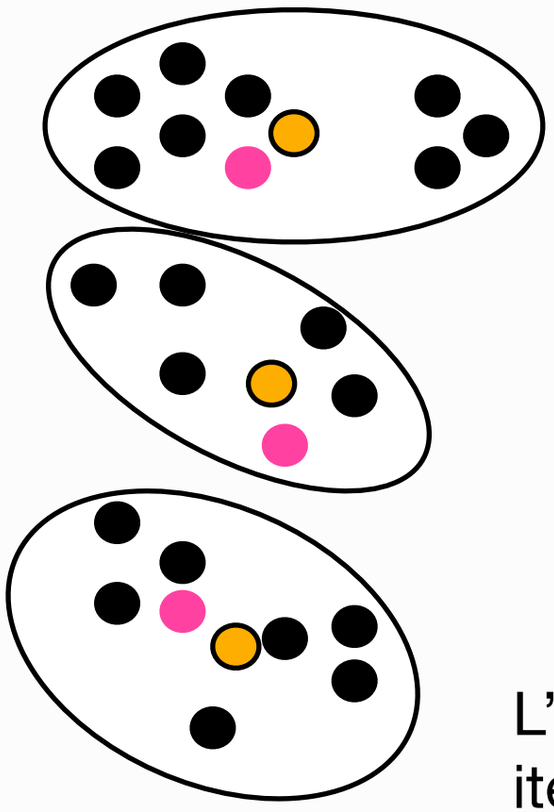
Step #4. Ripeti lo step #2 fin quando i centroidi non cambiano.

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.



Step #1. Seleziona k centroidi in maniera casuale —> k pattern sono eletti come rappresentanti;

Step #2. Genera un partizionamento assegnando ogni campione al centroide più vicino.

Step #3. Calcola i nuovi centroidi del cluster, considerando la media dei valori dei cluster generati al punto #2.

Step #4. Ripeti lo step #2 fin quando i centroidi non cambiano.

L'algoritmo k-means si può definire come *iterativo* poiché costruisce iterativamente i cluster. E' inoltre un algoritmo di *partizionamento* poiché fornisce un'unica partizione degli elementi. E' ad errore quadratico poiché mira a minimizzare l'errore rispetto ai centroidi.

Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Come facilmente intuibile, l'algoritmo ha una buona efficienza. Fornisce buoni risultati se i cluster sono compatti, ipersferici e ben separati nelle caratteristiche.

Dovremmo però impostare un valore *k* di cluster a priori: non avendo conoscenza del numero di classi del problema, sarà difficile stimare questo valore. Questo ci potrebbe portare ad un ottimo locale!

Per verificare il valore migliore di *k*, dovremo affidarci a dati empirici o alla conoscenza del dominio. O altrimenti?

Ma certo, *gli algoritmi di ricerca*! Potremmo, ad esempio, utilizzare un algoritmo di ricerca locale per restituire il valore *k* che ottimizza una funzione obiettivo come, ad esempio, la metrica di valutazione della bontà dei cluster generati.

Ulteriori miglioramenti potrebbero coinvolgere la generazione iniziale dei centroidi, ad esempio passando da una generazione casuale ad una pseudo-casuale.

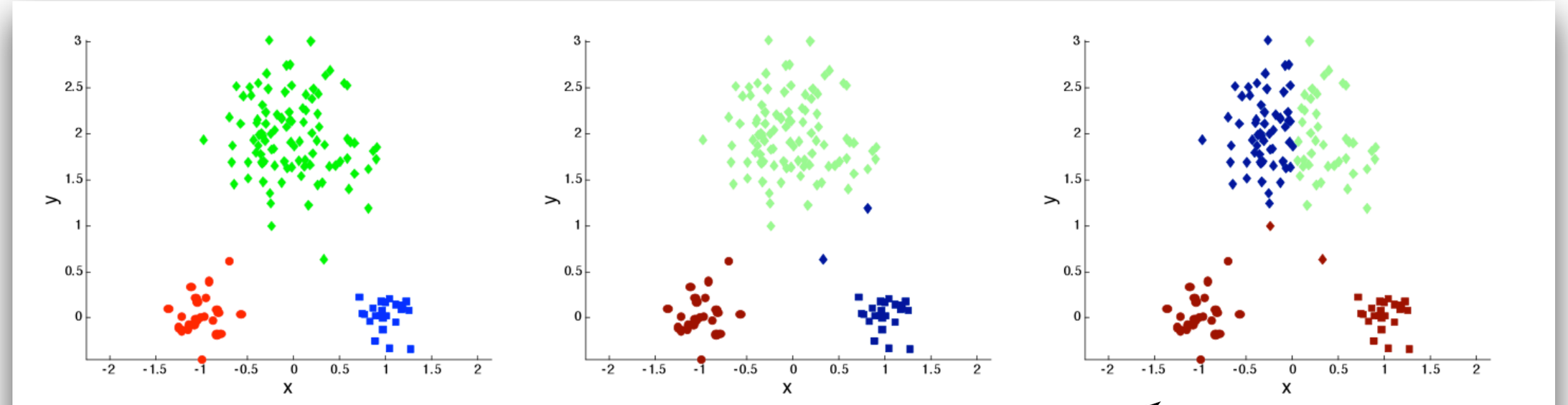
Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Vediamo graficamente quali sono i problemi più comuni di k-means.



Una scelta errata del valore *k* potrebbe portarci a creare più o meno cluster rispetto al numero ideale. Parliamo quindi di risultato sub-ottimale.

Clustering

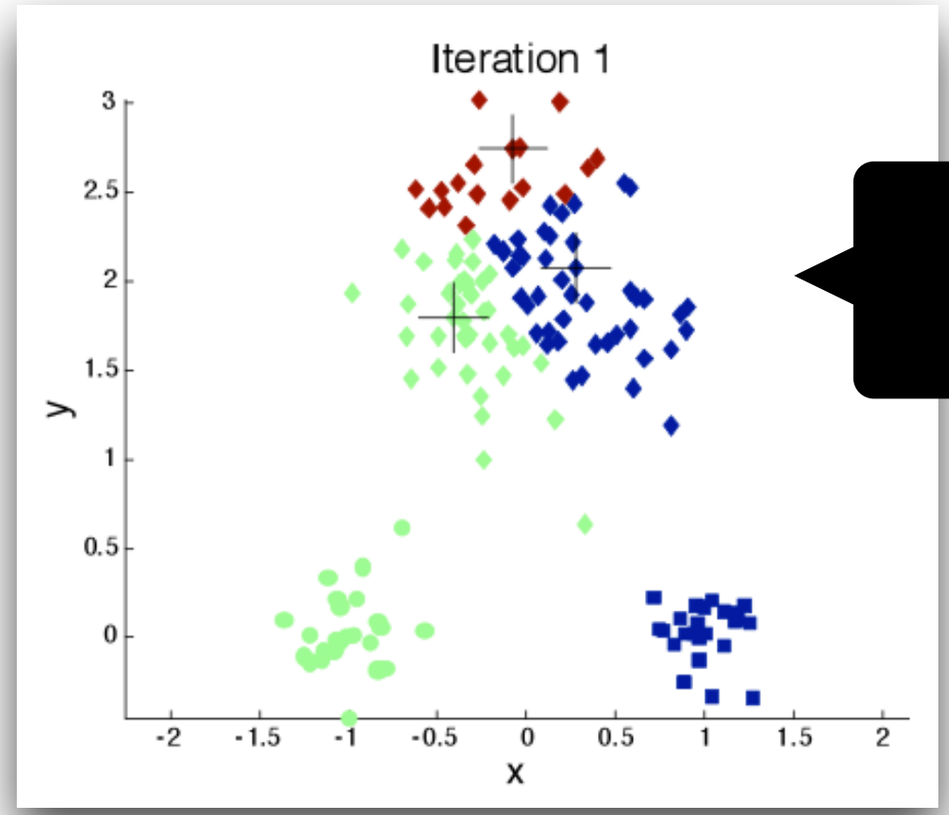
Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Vediamo graficamente quali sono i problemi più comuni di k-means.



Una scelta errata dei centroidi potrebbe influenzare negativamente il risultato.

Clustering

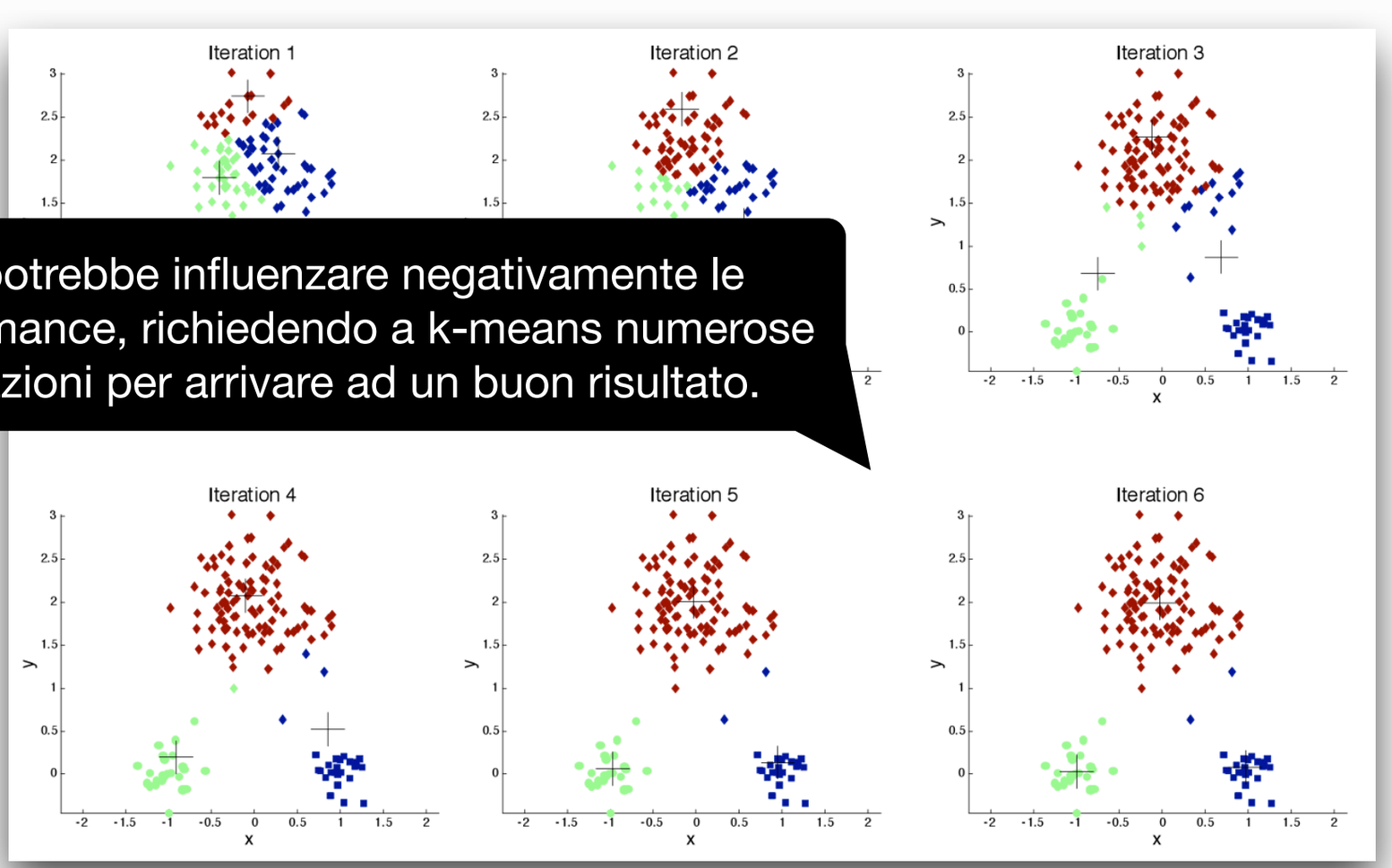
Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Vediamo graficamente quali sono i problemi più comuni di k-means.



O potrebbe influenzare negativamente le performance, richiedendo a k-means numerose iterazioni per arrivare ad un buon risultato.

Clustering

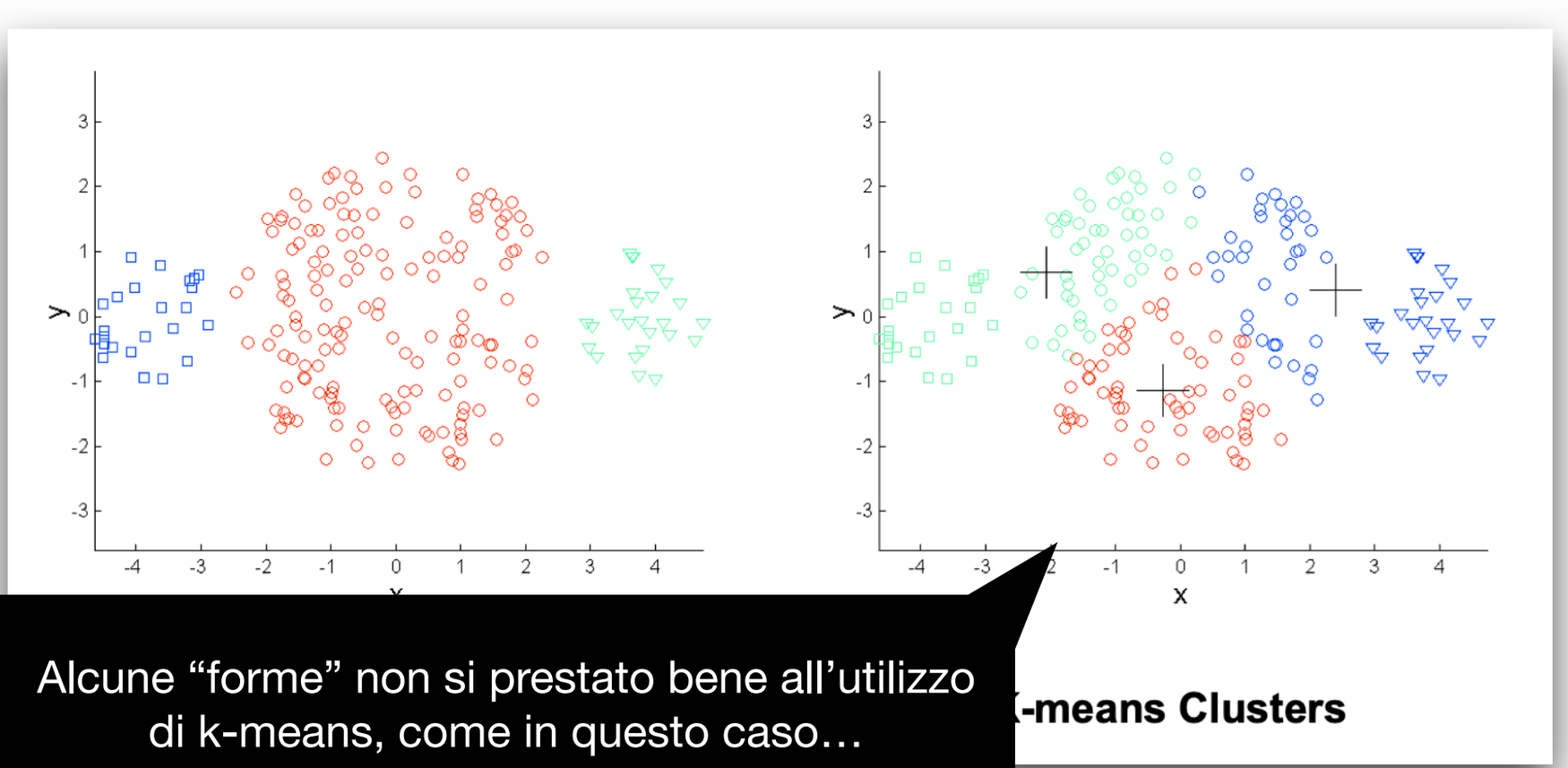
Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Vediamo graficamente quali sono i problemi più comuni di k-means.



Alcune "forme" non si prestato bene all'utilizzo di k-means, come in questo caso…

Clustering

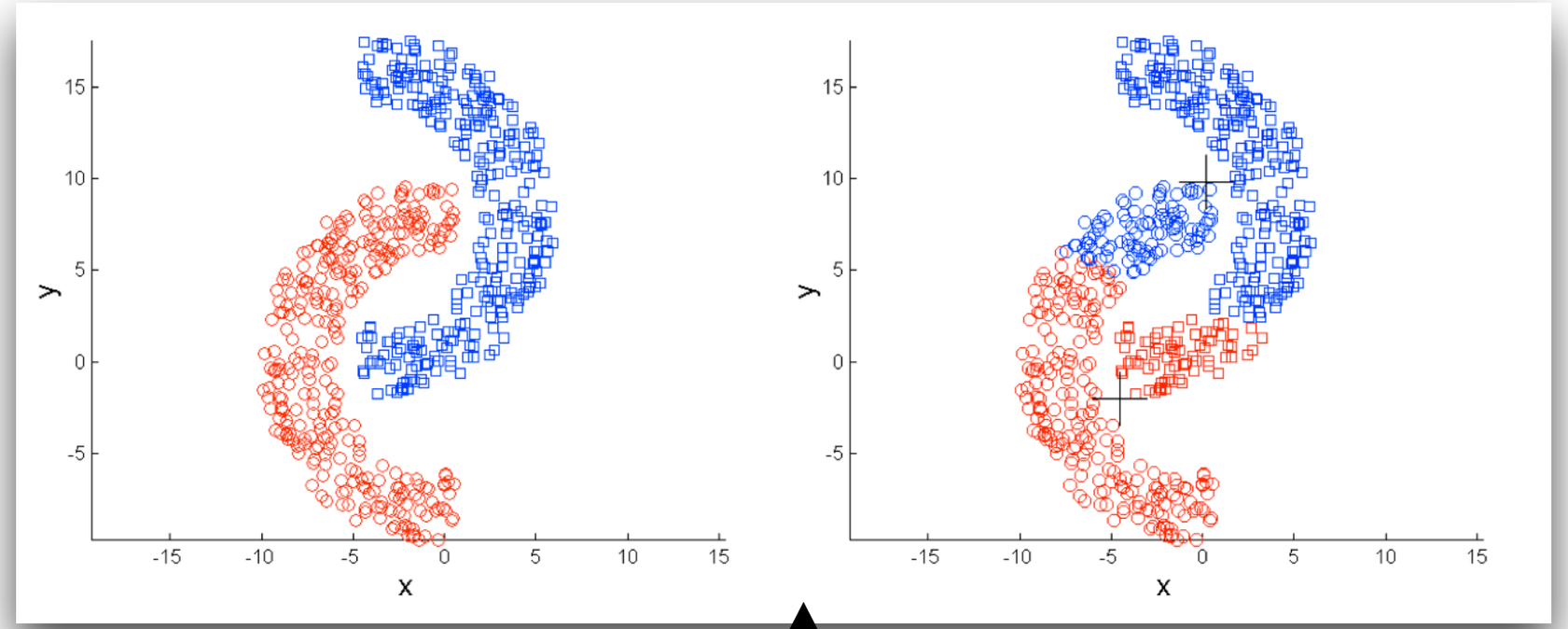
Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Vediamo graficamente quali sono i problemi più comuni di k-means.



O come in questo caso…

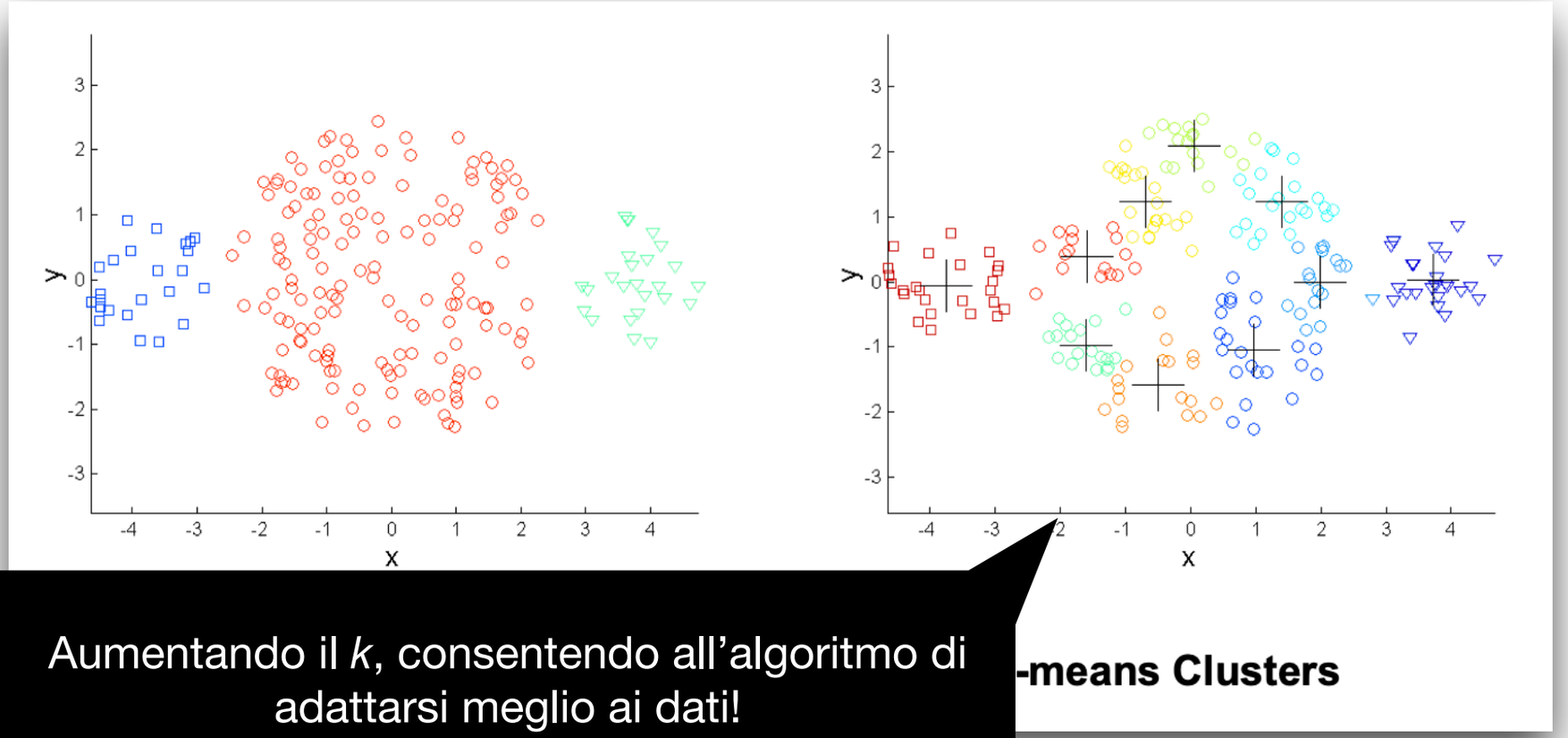
Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Come possiamo, quindi, migliorare le prestazioni di k-means in questi casi?



Aumentando il *k*, consentendo all'algoritmo di adattarsi meglio ai dati!

Clustering

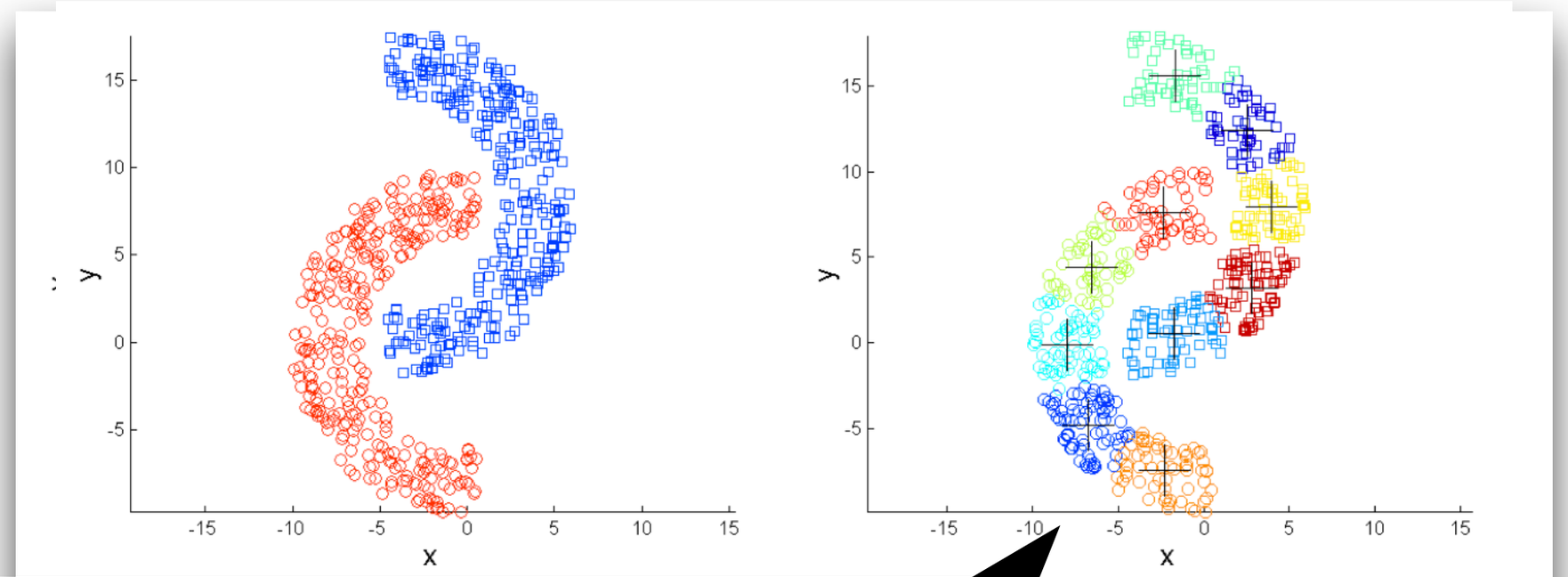
Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Come possiamo, quindi, migliorare le prestazioni di k-means in questi casi?



Aumentando il *k*, consentendo all'algoritmo di adattarsi meglio ai dati!

Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Algoritmo k-means

L'algoritmo delle k-medie è di gran lunga quello a partizionamento iterativo ad errore quadratico più famoso.

Inoltre, alcune attività di pre- e post-processing sono assolutamente vitali.

La normalizzazione dei dati e la rimozione degli outlier sono necessari per consentire a k-means di lavorare su dati più uniformemente distribuiti e, quindi, ridurre il rischio di ottenere risultati sub-ottimi.

Ma non basta. Il risultato del clustering può essere manipolato a posteriori dall'utente utilizzatore: l'eliminazione dei cluster piccoli (che possono rappresentare degli outlier) o l'unione dei cluster "vicini" sono operazioni che aiutano ad ottenere un migliore risultato. Questi passi possono anche essere implementati durante l'esecuzione.

Un'ultima nota. Per i dati categorici, l'algoritmo k-means non può funzionare. In questi casi si può utilizzare l'algoritmo *k-medoids*. Un medoide è il punto *meno dissimile* di una distribuzione, ad esempio la moda di una distribuzione.

Identificati i k medoidi, l'algoritmo è identico. L'unica accortezza è identificare una funzione di similarità adatta ai dati.

Clustering

Problemi di raggruppamento: Algoritmi

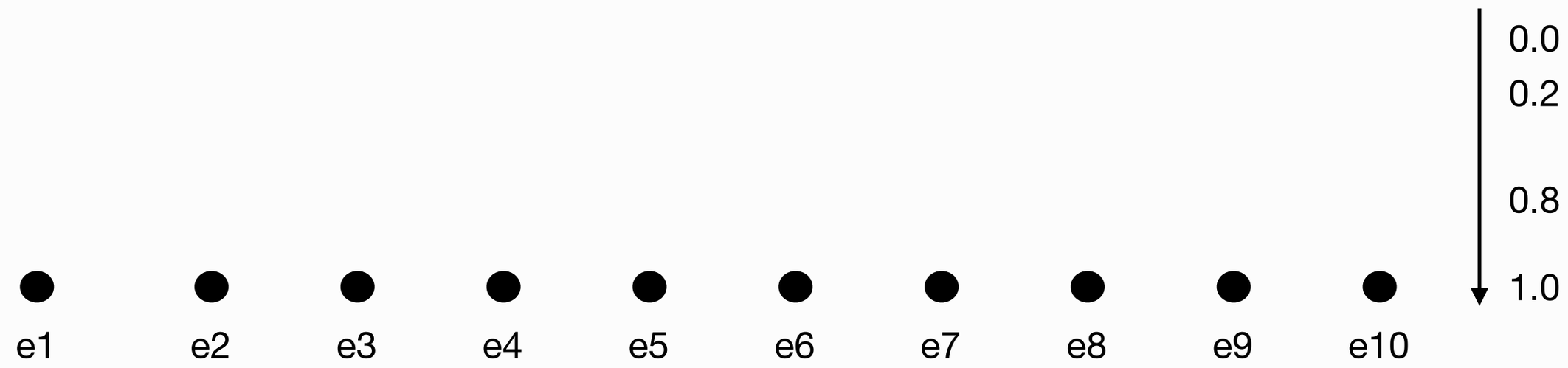
La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

Mentre l'algoritmo k-means restituisce delle partizioni disgiunte, alcuni gruppi di pattern potrebbero avere caratteristiche simili quando osservati ad un certo livello.

Il primo livello conterrà *n* cluster, ovvero ogni cluster avrà un solo elemento.



Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

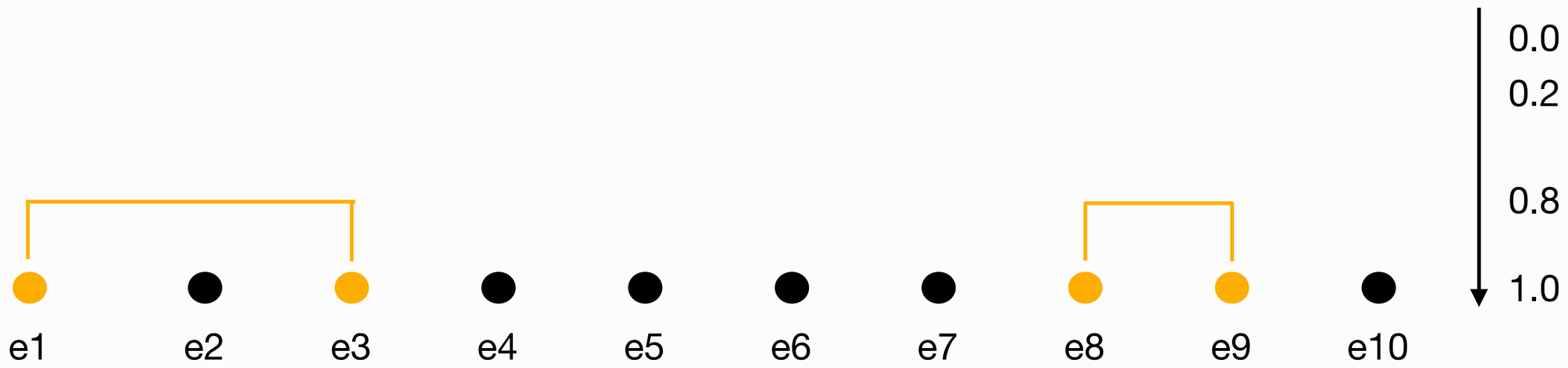
Clustering gerarchico

Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

Mentre l'algoritmo k-means restituisce delle partizioni disgiunte, alcuni gruppi di pattern potrebbero avere caratteristiche simili quando osservati ad un certo livello.

Il primo livello conterrà *n* cluster, ovvero ogni cluster avrà un solo elemento.

Il secondo conterrà *n-1* cluster, ovvero un cluster sarà formato sulla base della similarità delle caratteristiche.



Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

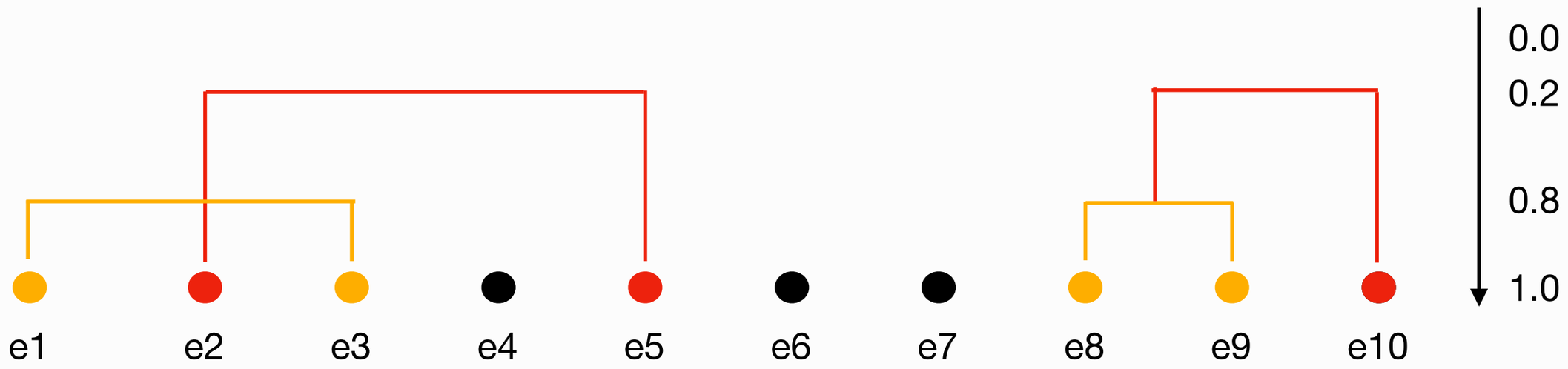
Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

Mentre l'algoritmo k-means restituisce delle partizioni disgiunte, alcuni gruppi di pattern potrebbero avere caratteristiche simili quando osservati ad un certo livello.

Il primo livello conterrà *n* cluster, ovvero ogni cluster avrà un solo elemento.

Il secondo conterrà *n-1* cluster, ovvero un cluster sarà formato sulla base della similarità delle caratteristiche.

Il processo andrà avanti fino a quando tutti gli elementi non formeranno un unico cluster.



Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

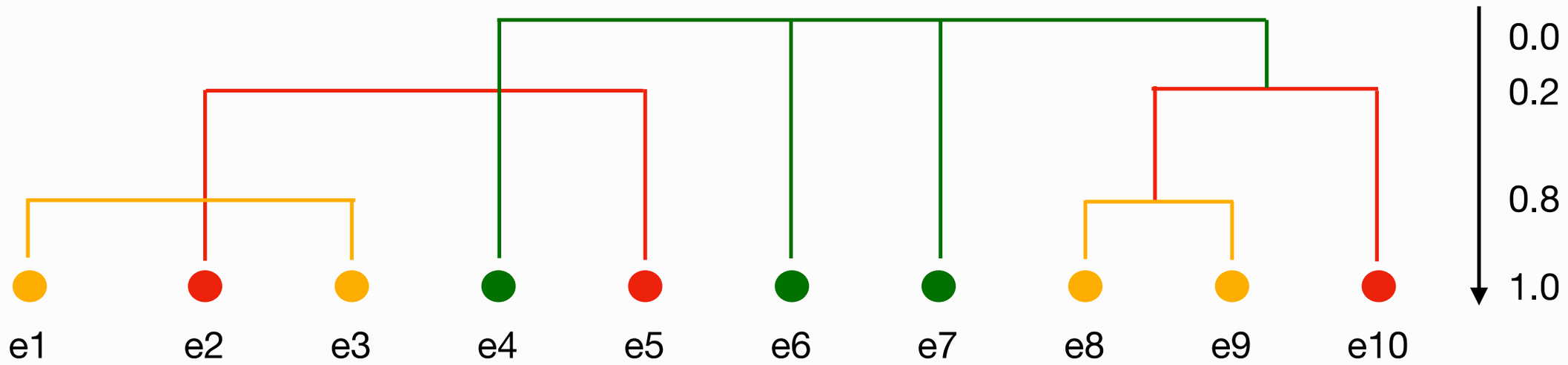
Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

Mentre l'algoritmo k-means restituisce delle partizioni disgiunte, alcuni gruppi di pattern potrebbero avere caratteristiche simili quando osservati ad un certo livello.

Il primo livello conterrà *n* cluster, ovvero ogni cluster avrà un solo elemento.

Il secondo conterrà *n-1* cluster, ovvero un cluster sarà formato sulla base della similarità delle caratteristiche.

Il processo andrà avanti fino a quando tutti gli elementi non formeranno un unico cluster.



Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

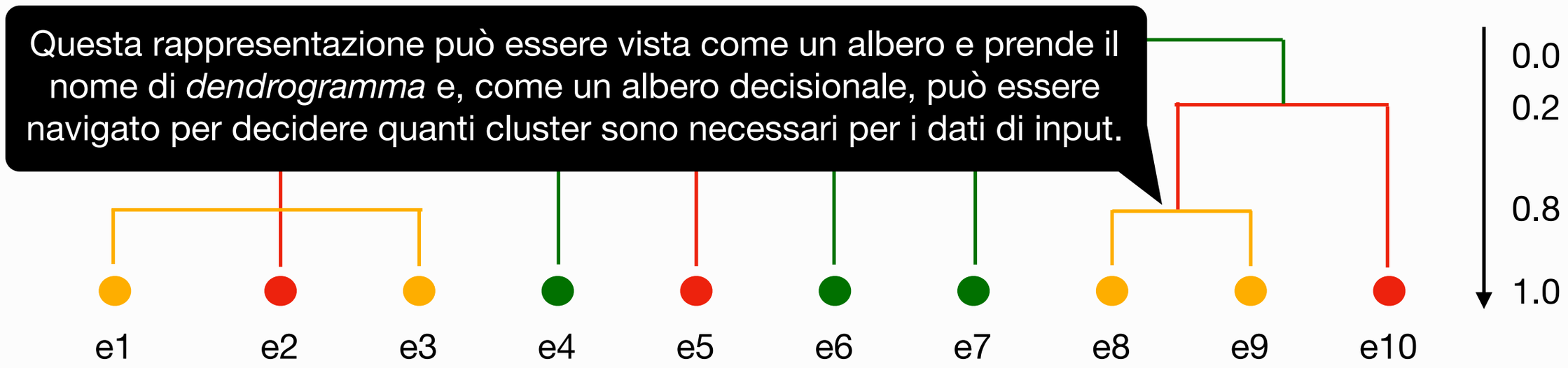
Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

Mentre l'algoritmo k-means restituisce delle partizioni disgiunte, alcuni gruppi di pattern potrebbero avere caratteristiche simili quando osservati ad un certo livello.

Il primo livello conterrà *n* cluster, ovvero ogni cluster avrà un solo elemento.

Il secondo conterrà *n-1* cluster, ovvero un cluster sarà formato sulla base della similarità delle caratteristiche.

Il processo andrà avanti fino a quando tutti gli elementi non formeranno un unico cluster.



Clustering

Problemi di raggruppamento: Algoritmi

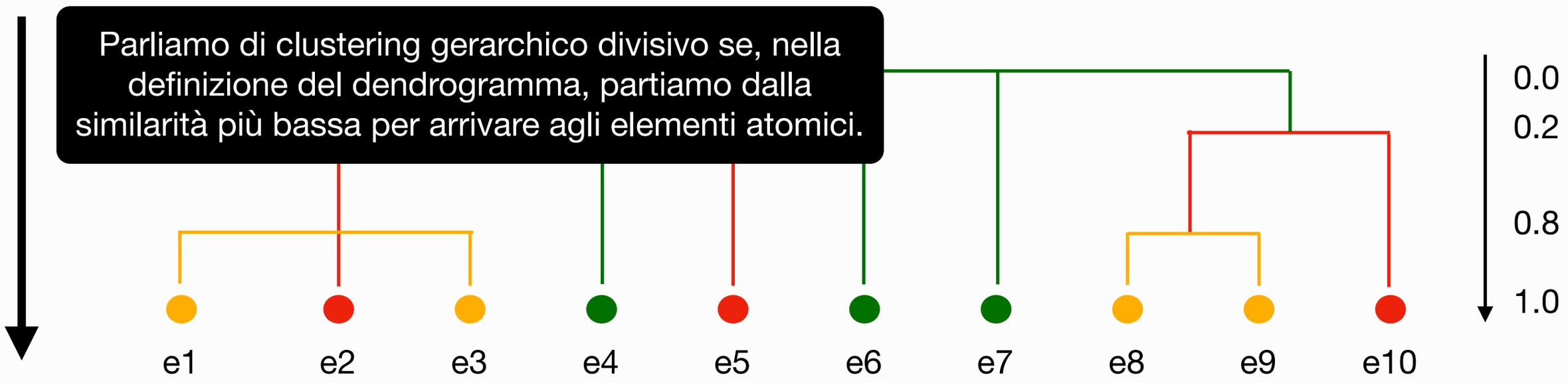
La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

In altri termini, nel clustering gerarchico non bisogna stabilire un numero *k* di cluster da generare, ma l'algoritmo andrà a raggruppare elementi sulla base della loro (de)crescente similarità.

Questo dà all'utente utilizzatore una maggiore capacità di interpretazione dei risultati, consentendo di scegliere a posteriori il livello di similarità ideale per i dati di input.



Clustering

Problemi di raggruppamento: Algoritmi

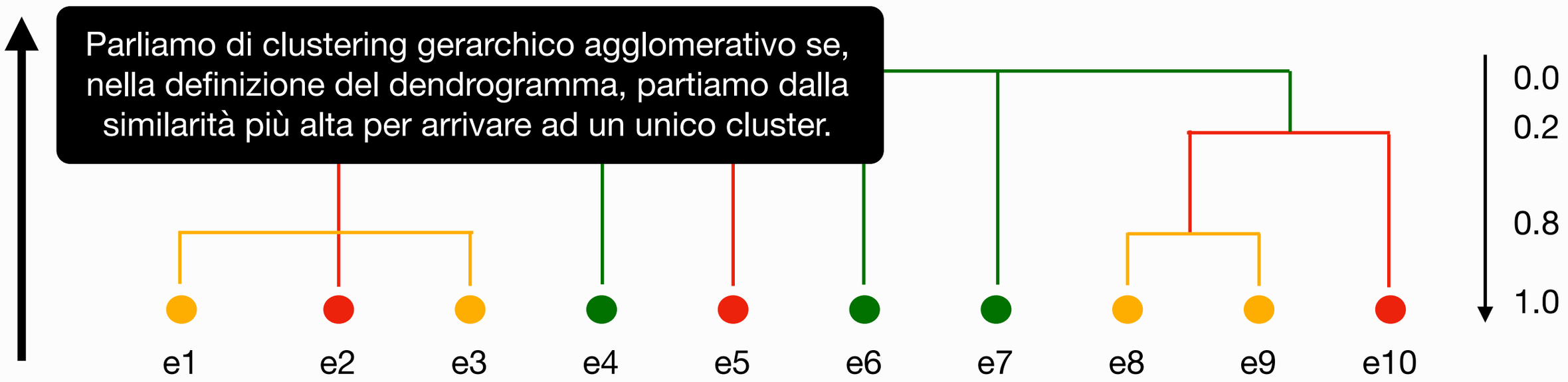
La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

In altri termini, nel clustering gerarchico non bisogna stabilire un numero *k* di cluster da generare, ma l'algoritmo andrà a raggruppare elementi sulla base della loro (de)crescente similarità.

Questo dà all'utente utilizzatore una maggiore capacità di interpretazione dei risultati, consentendo di scegliere a posteriori il livello di similarità ideale per i dati di input.



Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

La scelta principale nel clustering gerarchico consiste nella misura di distanza tra cluster, che sarà utilizzata per dividere o agglomerare cluster. **NB: Prima abbiamo visto le distanze metriche tra due elementi, qui parliamo di distanze tra cluster!**

A prescindere dalla misura metrica utilizzata, possiamo determinare la distanza tra cluster in vari modi. Le più note sono le seguenti.

$$d_{min}(D_i, D_j) = \min_{x \in D_i, x' \in D_j} |x - x'|$$

Minima distanza tra due punti nei cluster D_i e D_j.

$$d_{max}(D_i, D_j) = \max_{x \in D_i, x' \in D_j} |x - x'|$$

Massima distanza tra due punti nei cluster D_i e D_j.

Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

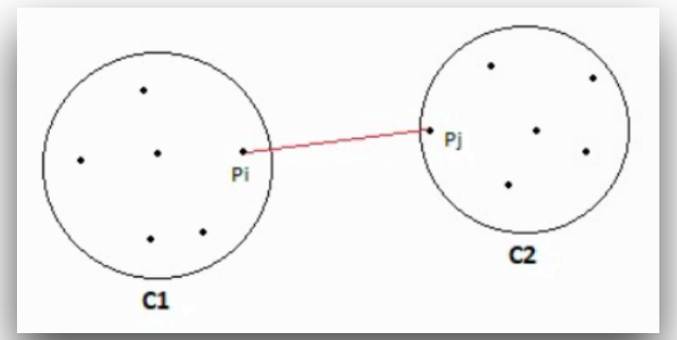
La scelta principale nel clustering gerarchico consiste nella misura di distanza tra cluster, che sarà utilizzata per dividere o agglomerare cluster. **NB: Prima abbiamo visto le distanze metriche tra due elementi, qui parliamo di distanze tra cluster!**

A prescindere dalla misura metrica utilizzata, possiamo determinare la distanza tra cluster in vari modi. Le più note sono le seguenti.

$$d_{min}(D_i, D_j) = \min_{x \in D_i, x' \in D_j} |x - x'|$$

Minima distanza tra due punti nei cluster D_i e D_j.

Quando utilizziamo d_{min} per determinare la distanza tra cluster, allora parliamo di *clustering del nearest neighbor*.



Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Clustering gerarchico

Il clustering gerarchico cerca di considerare raggruppamenti multi-livello (ovvero, a diversi livelli di similarità).

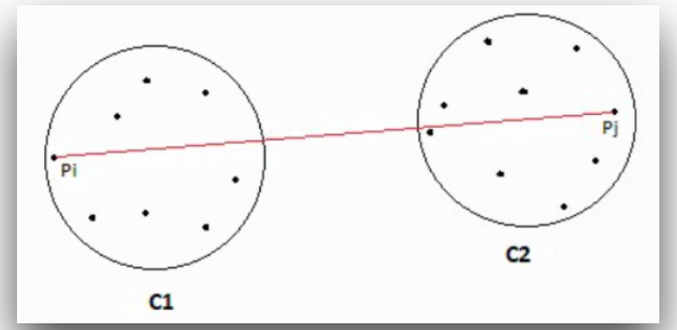
La scelta principale nel clustering gerarchico consiste nella misura di distanza tra cluster, che sarà utilizzata per dividere o agglomerare cluster. **NB: Prima abbiamo visto le distanze metriche tra due elementi, qui parliamo di distanze tra cluster!**

A prescindere dalla misura metrica utilizzata, possiamo determinare la distanza tra cluster in vari modi. Le più note sono le seguenti.

$$d_{max}(D_i, D_j) = \max_{x \in D_i, x' \in D_j} |x - x'|$$

Massima distanza tra due punti nei cluster D_i e D_j.

Quando utilizziamo d_{max} per determinare la distanza tra cluster, allora parliamo di *clustering del farthest neighbor*.



Clustering

Problemi di raggruppamento: Algoritmi

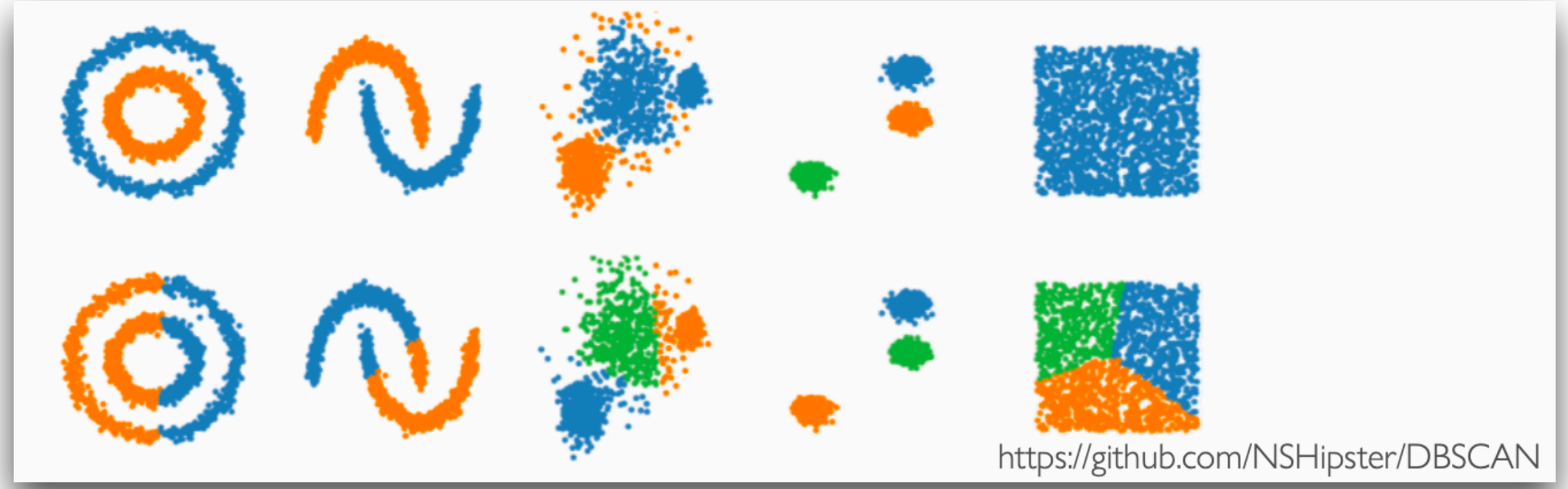
La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Density-based clustering

Il clustering basato sulla densità raggruppa pattern considerando la loro densità nella distribuzione.

DBSCAN

k-means



Questo tipo di clustering andrà a risolvere il problema delle “forme” che abbiamo visto con k-means

Clustering

Problemi di raggruppamento: Algoritmi

La maggior parte degli algoritmi si basa su (1) partizionamento iterativo ad errore quadratico e (2) clustering gerarchico agglomerativo. Vari algoritmi differiscono tra di loro per la misura di similarità usata o per il criterio di ottimizzazione.

Density-based clustering — Il clustering basato sulla densità raggruppa pattern considerando la loro densità nella distribuzione.

Non entreremo troppo nel dettaglio di questi algoritmi, ma basta sapere che si basano su due parametri per stabilire il criterio di raggruppamento dei cluster.

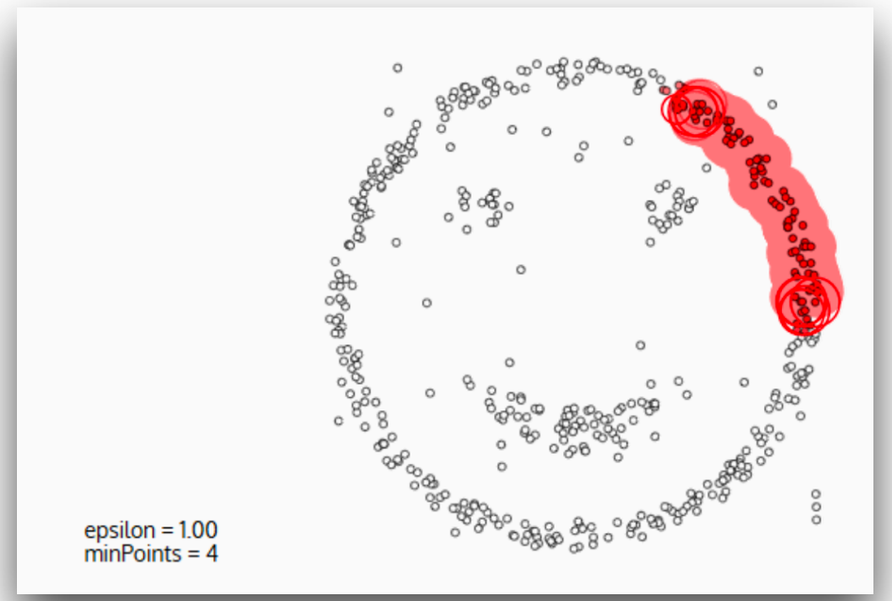
Il parametro *minPts*, che stabilisce il numero minimo di punti per considerare un intorno di un punto denso.

Il parametro ϵ, che definisce un intorno circolare di ciascun punto dai suoi vicini.

Aumentando o riducendo il valore di questi parametri, potremo migliorare la qualità del clustering.

La cosa importante da capire è che questo tipo di clustering è in grado di scoprire forme arbitrarie.

L'algoritmo più noto basato sulla densità è chiamato DBSCAN.



Problemi di raggruppamento: Valutazione dei risultati

L'ultimo punto da affrontare quando si parla di clustering è relativo alla valutazione della bontà dei cluster. Come detto in precedenza, questa dipenderà da vari fattori, come la misura di similarità utilizzata, dall'algoritmo stesso, o da fattori esterni.

Il vero problema è però un altro: essendo algoritmi non supervisionati, l'assenza di etichette rende impossibile stimare con esattezza l'accuratezza/precisione dei cluster che sono stati formati da un algoritmo.

Pertanto, dobbiamo utilizzare delle metriche di stima. Tra le tante a disposizione, vengono generalmente utilizzate le seguenti tre metriche:

1. Il punto di *gomito*, detto Elbow point;
2. Il coefficiente di forma, detto Silhouette coefficient;
3. MoJo distance.

Il punto di gomito è spesso utilizzato per valutare il numero migliore di cluster da generare negli algoritmi di clustering partizionale, ad esempio k-means.

Il coefficiente di forma misura la consistenza dei cluster, andando a misurare quanto simili sono gli elementi che compongono un singolo cluster.

La Move-Join (MoJo) distance può essere calcolata solo se abbiamo a disposizione delle etichette per poter valutare la bontà del clustering —> in alcuni casi, le etichette vengono ignorate in fase di costruzione del modello per poi essere usate in fase di validazione.

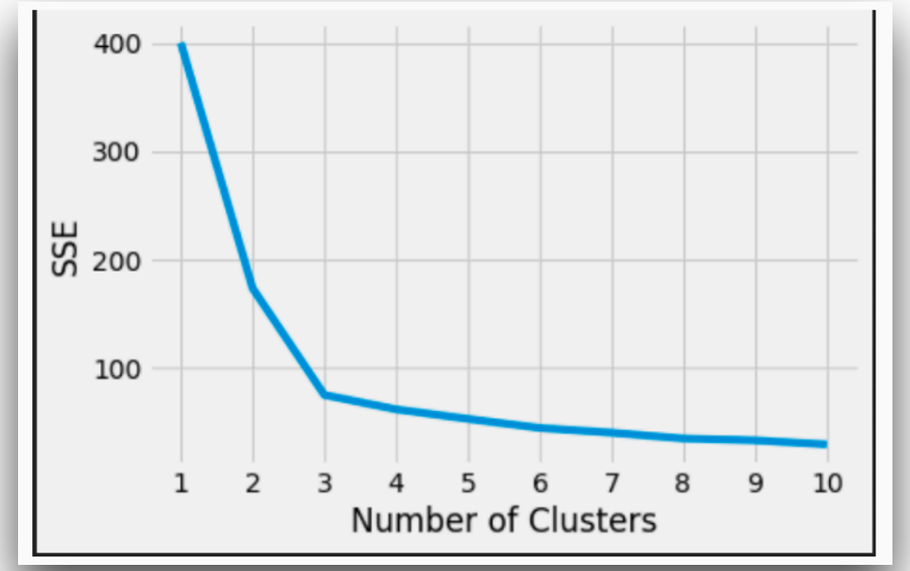
Clustering

Problemi di raggruppamento: Valutazione dei risultati - Elbow point

Il punto di gomito è un *metodo empirico*, che consiste nel graficare i valori candidati del parametro *k* rispetto alla somma degli errori quadratici ottenuti dall'algoritmo configurato per generare *k* cluster.

Consideriamo questa figura, in cui vengono rappresentati come, al variare del numero di cluster, varia la somma degli errori quadratici.

Da qui, possiamo vedere come l'errore diminuisce drasticamente quando si passa da 2 a 3 cluster. L'errore decresce ancora man mano che il numero di cluster aumenta.



Sebbene l'errore venga minimizzato quando i cluster aumentano, avere un numero eccessivo di cluster (in figura, 10) implica avere tanti gruppi formati da pochissimi elementi. Nel caso estremo, avremo l'errore minimo quando ci sarà un cluster per ogni elemento, il ché significa non fare clustering.

Questa è perciò una situazione da evitare. Vogliamo avere il giusto compromesso tra errore e capacità di raggruppamento. Il punto di Elbow consente di identificare questo compromesso.

Nel caso di esempio, un buon compromesso sarebbe quello di avere k=3 o k=4.

Clustering

Problemi di raggruppamento: Valutazione dei risultati - Silhouette coefficient

Il silhouette coefficient è una misura della coesione e separazione tra i dati. Più in particolare, quantifica quanto i dati siano ben disposti nei cluster generati.

Il coefficiente si basa su due parametri: (1) Quanto bene i dati sono *ammassati* nel cluster di riferimento; (2) Quanto è distante ciascun campione da qualsiasi altro cluster.

Il coefficiente varia tra -1 e +1. Valori alti indicano condizioni maggiori di coesione e di separazione dei dati. Il coefficiente, per un punti *i*-esimo appartenente al cluster *c*, è calcolato tramite la seguente formula:

$$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))}$$

con:

- *a(i)* che rappresenta la distanza media dell'*i*-esimo punto rispetto a tutti gli altri punti appartenente allo stesso cluster;
- *b(i)* che rappresenta la distanza media dell'*i*-esimo punto rispetto a tutti gli altri punti appartenente al cluster più vicino del cluster a cui è stato assegnato;
- *s(i)* che rappresenta il coefficiente di silhouette dell'*i*-esimo punto.

Il valore finale di silhouette è dato dalla media dei coefficienti di silhouette calcolati per ogni elemento del problema.

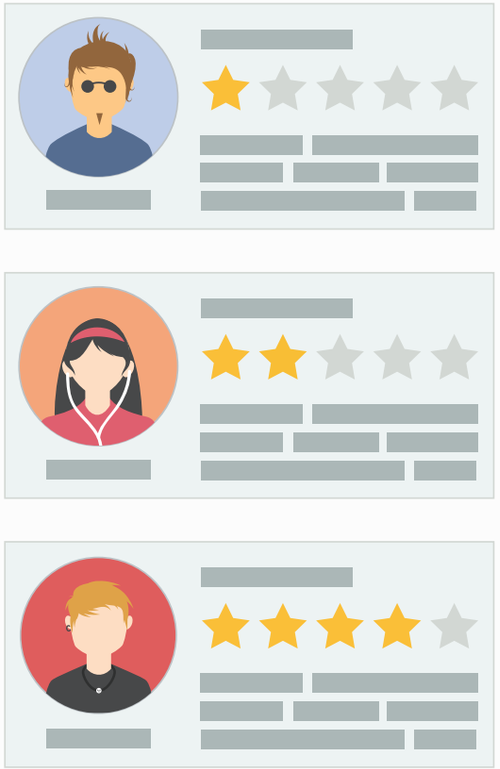
Clustering

Problemi di raggruppamento: Valutazione dei risultati - MoJo distance

La Move-Join (MoJo) distance è una metrica che calcola il numero minimo di operazioni di spostamento e raggruppamento di elementi sono necessari per passare dal clustering identificato da un algoritmo al clustering ideale degli elementi.

Qualcuno potrebbe chiedersi: *Ma se ho a disposizione un oracolo che riporta il raggruppamento ideale, perché dovrei fare clustering con un algoritmo?*

La risposta è semplice: per valutare un algoritmo di clustering prima di utilizzarlo su nuovi dati sconosciuti! Facciamo un esempio.



Conoscete tutti il meccanismo delle user review, che consente agli utenti di applicazioni mobile di esprimere opinioni su un'app, far notare agli sviluppatori eventuali problemi e/o funzionalità mancanti che dovrebbero essere implementate.

Applicazioni popolari, come Instagram, Whatsapp o TikTok, ricevono migliaia di review al giorno e tenere traccia di quello che succede è difficile, se non impossibile.

Tuttavia, sebbene il numero di review sia enorme, è probabile che gli utenti facciano notare cose simili! Ad esempio, un bug può essere notato da più utenti.

Quindi, possiamo pensare di usare una tecnica di clustering!

Clustering

Problemi di raggruppamento: Valutazione dei risultati - MoJo distance

Ma, esattamente, clustering di cosa? E come?

Dovremmo sicuramente estrarre tutte le user review dell'app di interesse (o una parte, magari le più recenti o quelle relative all'ultima versione rilasciata) e provare a raggrupparle in base ad una misura di somiglianza testuale.

Quindi, come metrica di somiglianza potremmo usare, ad esempio, la distanza di Jaccard o quella di Hamming.

A quel punto, dovremmo ottimizzare la generazione dei cluster. Per farlo, potremmo usare diversi algoritmi di clustering (k-medoids o altri algoritmi più specializzati nel raggruppamento di stringhe e testi scritti in linguaggio naturale).

Non sappiamo però quale di questi algoritmi è più efficace a risolvere un problema di questo tipo - consideriamo che, parlando di user review, parliamo di testi rumorosi che sono naturalmente difficili da raggruppare.

Per confrontare i vari algoritmi potremmo quindi provare la seguente idea. Facciamo un sacrificio ed estraiamo le user review della release R_i, raggruppiamole manualmente formando dei cluster semanticamente validi e valutiamo quanto i vari algoritmi di clustering sperimentati si avvicinano al nostro operato.

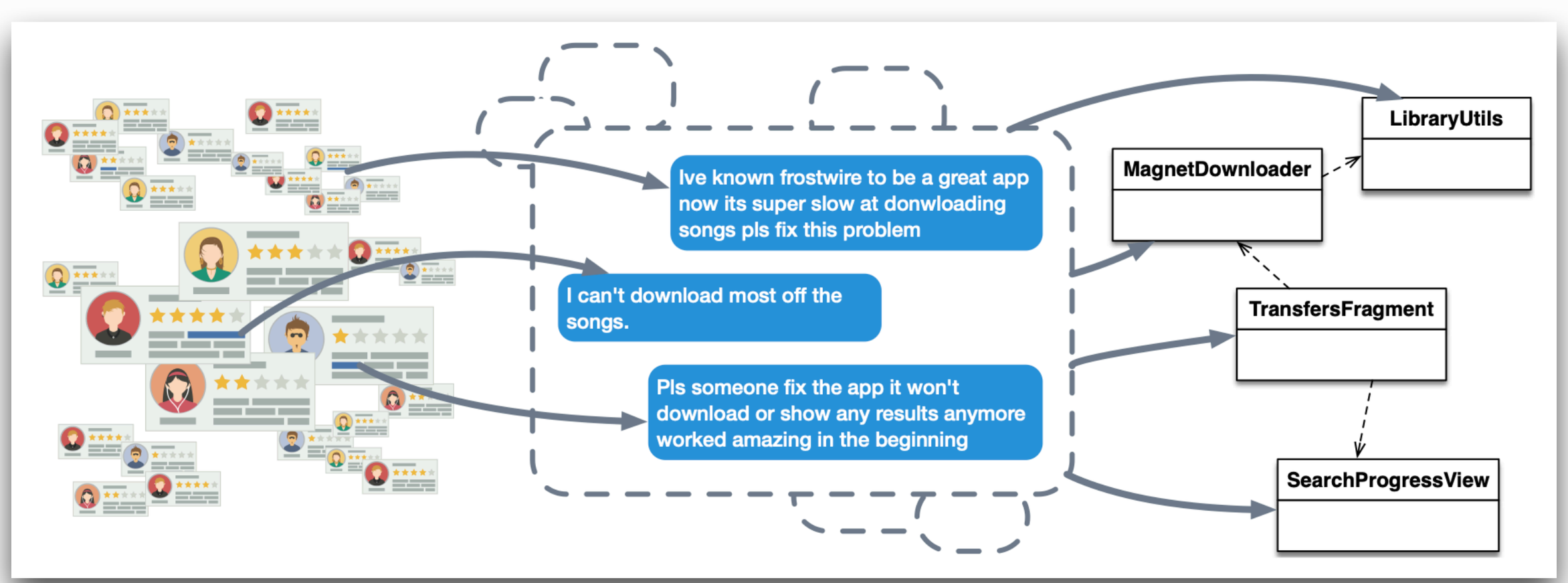
Una volta stabilito il miglior algoritmo potremo poi assumere che, per nuove release dell'app, possiamo affidarci al risultato per prendere le appropriate decisioni.

Problemi di raggruppamento: Valutazione dei risultati - MoJo distance

In questo senso, la MoJo distance ci fornisce una misura molto più precisa della bontà dei cluster, in quanto si basa su dati reali (l'oracolo).

Più in generale, una valutazione fatta tramite il calcolo della MoJo distance ci dà maggiore confidenza nell'utilizzare i risultati del clustering. Inoltre, può più facilmente abilitare l'utilizzo del clustering come mezzo per altre operazioni.

Ad esempio, ChangeAdvisor.



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Laurea triennale in Informatica

Fondamenti di Intelligenza Artificiale

Lezione 18 - Clustering