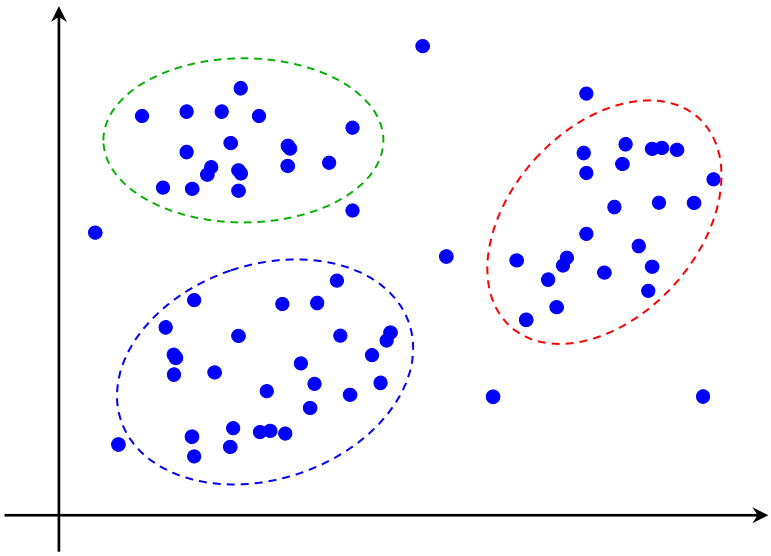
Definizioni

Con il termine Clustering (in italiano «raggruppamento») si denota un famiglia di metodi *non supervisionati* in grado di individuare raggruppamenti intrinseci (cluster) di pattern nello spazio multidimensionale, e (opzionalmente) di definire in corrispondenza di tali raggruppamenti le classi (incognite).

Il clustering ha applicazioni in numerose discipline (*pattern recognition, machine learning, computer vision, data mining, data base, ...*) e pertanto ha sempre ricevuto un notevole interesse.



Il problema è molto complesso (NP hard): determinare la soluzione ottima con ricerca esaustiva è possibile solo nei casi semplici (pochi pattern).

Quante soluzioni?

- dati n pattern, assumendo di conoscere s, il numero di soluzioni è dell'ordine di $\frac{s^n}{s!}$ (approssimazione numero di Stirling di seconda specie).
  - https://it.wikipedia.org/wiki/Numeri_di_Stirling
  - Esempio: per n =100, s =5, il numero di soluzioni $\approx 10^{67}$.

- se s non è noto, il numero di soluzioni è dato dal numero di Bell, ottenibile sommando i casi precedenti per tutti i valori di s da 1 a n.
  - https://it.wikipedia.org/wiki/Numeri_di_Bell

- Nel caso $s = 2$ è molto semplice dimostrare che il numero di soluzioni è $2^{n-1} - 1$.
  - Esempio: dati i 4 pattern {A, B, C, D} e $s = 2$, le soluzioni di clustering sono 7:

    (A) (B,C,D)
    (B) (A,C,D)
    (C) (A,B,D)
    (D) (A,B,C)
    (A,B) (C,D)
    (A,C) (B,D)
    (A,D) (B,C)

Criteri di Clustering

- Criteri e algoritmi di clustering sono due cose ben distinte: i primi descrivono cosa si vuol ottenere specificando il grado di ottimalità di ogni soluzione ammissibile; i secondi, dato un criterio di clustering, forniscono una procedura algoritmica per determinare soluzioni che lo ottimizzano.
- La maggior parte dei criteri di clustering sono definiti sulla base delle due osservazioni seguenti:
  1) i pattern all'interno dello stesso cluster devono essere tra loro più simili rispetto a pattern appartenenti a cluster diversi
  2) i cluster sono costituiti da nuvole di punti a densità relativamente elevata separate da zone dove la densità e più bassa.

- Tra i diversi criteri possibili:
  - minimizzazione distanze dai centroidi: minimizza la somma dei quadrati delle distanze dei pattern $\mathbf{x}$ dai centroidi (i.e. baricentri) delle classi.

$$J_e = \sum_{i=1..s} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \bar{\mathbf{x}}_i\|^2 , \qquad \bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

    dove C_i è l'i-esimo cluster, n_i il numero di pattern che contiene e $\bar{\mathbf{x}}_i$ il suo centroide (media).

    È un buon criterio per cluster a simmetria radiale (i.e., circolari), ma penalizza forme allungate o cluster innestati (i.e. un cluster a forma di anello con all'interno un altro cluster).

Criteri di Clustering (2)

- minimizzazione distanze intra-classe:

$$J_e = \sum_{i=1..s} n_i \cdot \bar{s}_i, \qquad \bar{s}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} f_s(\mathbf{x}, C_i)$$

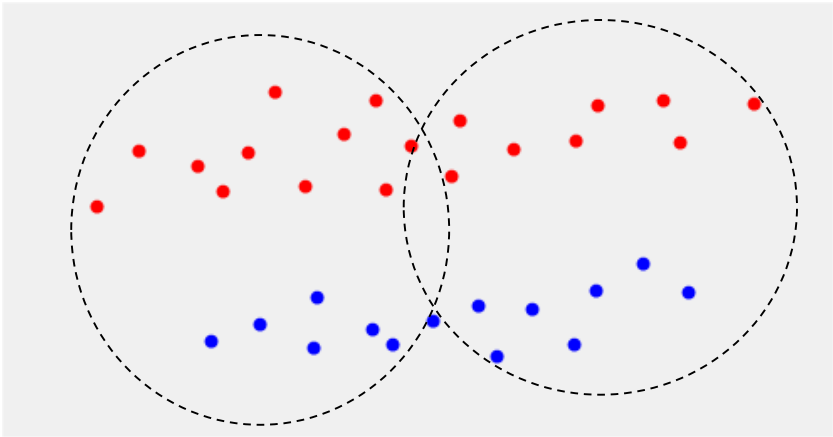
dove $f_s(\mathbf{x}, C_i)$ è una misura di distanza tra $\mathbf{x}$ e il cluster cui appartiene. Ad esempio:

1. $f_s(\mathbf{x}, C_i) = \frac{1}{n_i} \sum_{\mathbf{x}' \in C_i} \|\mathbf{x} - \mathbf{x}'\|^2$

   criterio simile alla minimizzazione distanze dai centroidi (slide precedente)

2. $f_s(\mathbf{x}, C_i) = \min_{\substack{\mathbf{x}' \in C_i \\ \mathbf{x}' \neq \mathbf{x}}} \|\mathbf{x} - \mathbf{x}'\|^2$

   non penalizza cluster allungati, infatti affinché $f_s(\mathbf{x}, C_i)$ assuma valore ridotto, non è necessario che tutti i (o gran parte dei) pattern di C_i siano vicini a $\mathbf{x}$, ma è sufficiente un vicino. Nell'esempio sotto, per $s = 2$, (1) favorirebbe l'aggregazione come da cerchi tratteggiati, mentre (2) come da colori (rosso e blu).



Algoritmi di Clustering

- Le principali famiglie di algoritmi sono:
  - Clustering gerarchico: attraverso operazioni tipicamente «bottom-up», che aggregano pattern in base a una misura di distanza, si organizzano i dati in struttura ad albero (dendogramma).
  - Clustering basato su centroidi: attraverso euristici (iterativi) si individuano i cluster cercando di minimizzare la distanza dei pattern dai centroidi dei cluster cui appartengono:
    - K-means
    - Fuzzy K-means
    - Expectation – Maximization (Gaussian Mixture)
  - Clustering basato sulla densità: i cluster individuati sono regioni connesse in aree ad elevata densità. L'approccio più noto è DBSCAN (*https://en.wikipedia.org/wiki/DBSCAN*)

- Distinguiamo inoltre:
  - Clustering hard (esclusivo): un pattern è assegnato (in modo esclusivo) a un solo cluster.
  - Clustering soft (fuzzy): i pattern appartengono ai diversi cluster con un certo grado di appartenenza (es. tra 0 e 1).

    È più efficace nel gestire pattern vicino al bordo di due o più clusters e outliers. L'assegnazione può diventare esclusiva scegliendo, per ogni pattern, il cluster verso cui il grado di appartenenza è massimo.

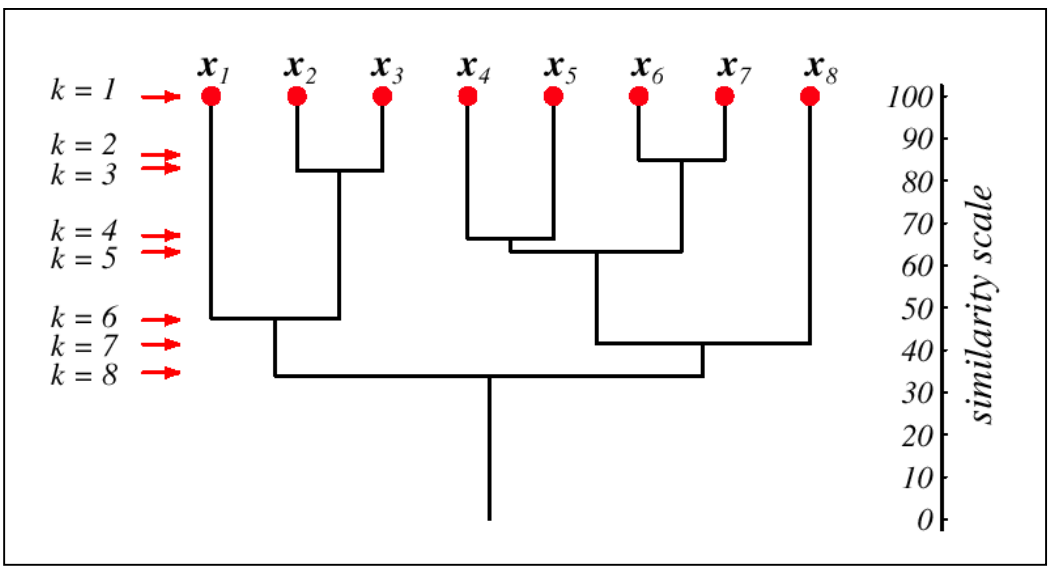
Clustering Gerarchico

Simile al modo di eseguire classificazione in tassonomia biologica:

- gli insetti sono gerarchicamente classificati specializzandone le specie a partire da famiglie molto ampie fino a famiglie più ridotte.

Gli algoritmi sono generalmente bottom-up (agglomerativi):

- si parte cercando di aggregare singoli elementi e ad ogni passo (livello) si aggregano (*pattern a pattern*, *pattern a cluster*, o *cluster a cluster*) gli elementi tra loro più simili (i.e. meno distanti) rispetto a una soglia (che dipende dal livello).
- Le principali differenze implementative dipendono dalla definizione utilizzata per calcolare le distanze tra cluster e cluster:
  - Single link: distanza minima tra due pattern dei cluster
  - Average link: distanza media tra i pattern dei due cluster
  - Complete link: distanza massima tra due pattern dei cluster



K-means

K-means (o C-means) è un metodo computazionalmente molto semplice e altrettanto semplice da implementare: per questo motivo è spesso la prima scelta per risolvere problemi di clustering.

- Minimizza «implicitamente» le distanze dai centroidi.
- Richiede in input il numero di cluster (s) e una soluzione iniziale. Produce buoni risultati a patto di fornire una ragionevole soluzione iniziale e un numero adeguato di classi.
- Il tipo di ottimizzazione è iterativa e locale; pertanto il metodo può convergere a massimi locali della soluzione. La convergenza si ottiene solitamente in pochi passi: < 10).
- Identifica cluster iper-sferici nel caso in cui venga utilizzata la distanza euclidea come misura di distanza tra i pattern o cluster iper-ellissoidali nel caso di distanza di Mahalanobis.
- Nella sua versione base, l'algoritmo può essere così descritto:

Inizializza n, s

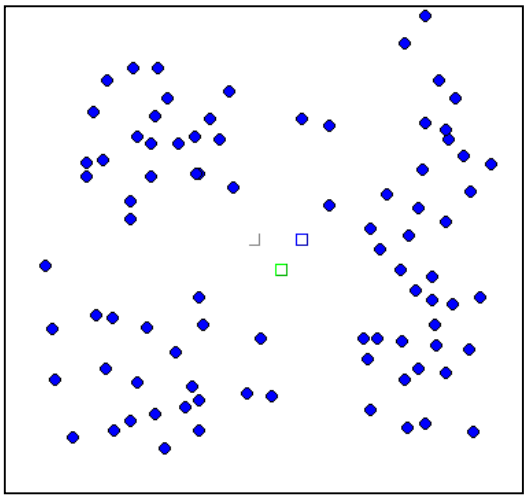
Scegli casualmente s pattern da utilizzare come centroidi iniziali

do { assegna ogni pattern al cluster per cui è minima la distanza
dal centroide.
calcola i centroidi dei cluster come media dei rispettivi pattern
} while (*i cluster sono stati modificati & iteration < max*)

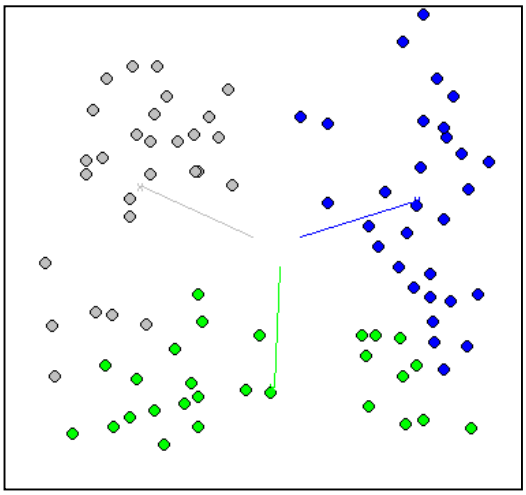
I cluster sono modificati iterativamente a seguito del ricalcolo del loro centroide. L'algoritmo termina (converge) quando i centroidi sono stabili e quindi le partizioni non cambiano.

K-means: esempio ($n = 86$, $s = 3$)

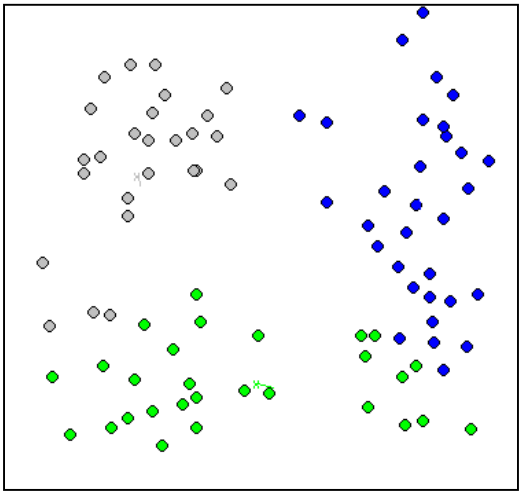
soluzione iniziale



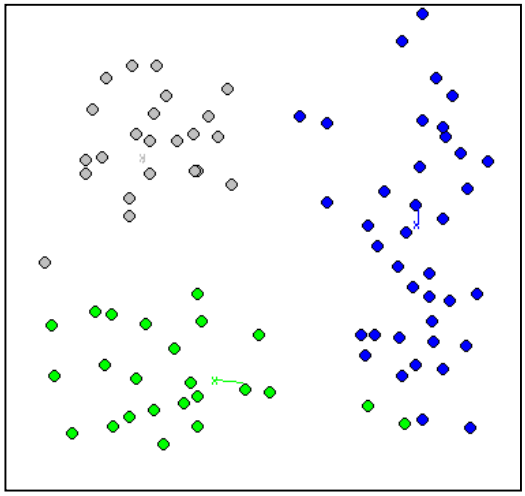
iterazione 1



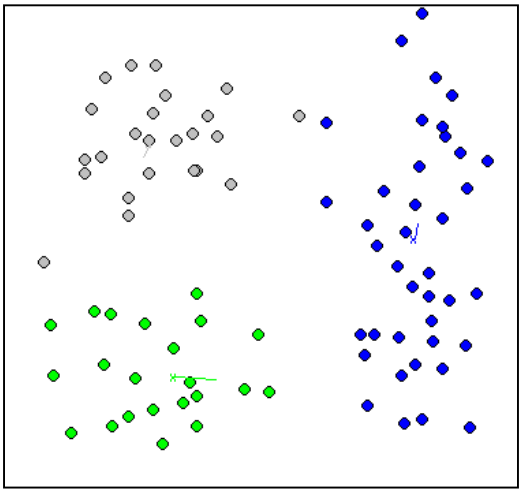
iterazione 2



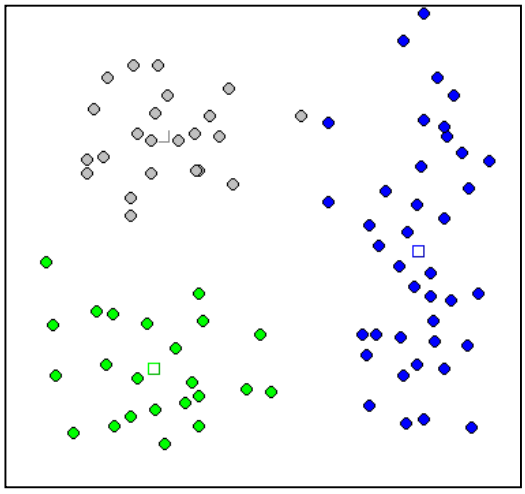
iterazione 4



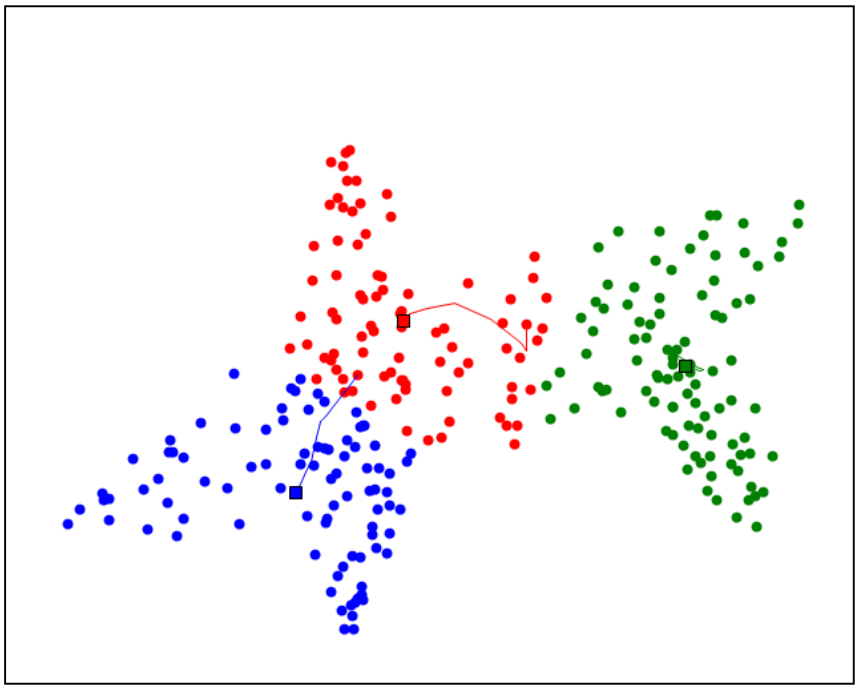
iterazione 6



iter. 7 (convergenza)

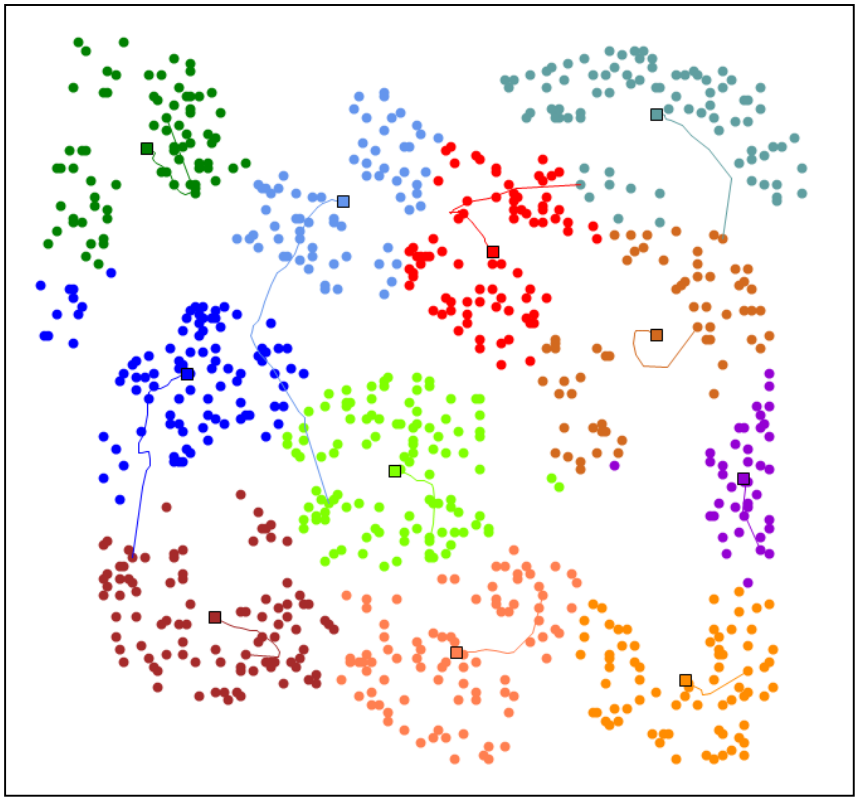


K-means: limitazioni



3 classi – 8 iterazioni

Minimizzando le distanze dai centroidi, K-means non è in grado di identificare cluster dalla forma non sferica.



11 classi, 31 iteraz.

anche in questo caso non tutti i cluster hanno forma sferica ...

K-means: soluzione iniziale e validazione

Diverse varianti sono state proposte per generare buone soluzioni iniziali e di determinare il numero di classi (clustering validation).

- Per minimizzare il rischio di convergenza verso minimi locali, l'algoritmo può essere eseguito più volte a partire da soluzioni iniziali diverse:
  - random
  - prodotte da un (diverso) euristico
  - o magari da un metodo evoluzionistico (algoritmo genetico).

Quanti cluster ?

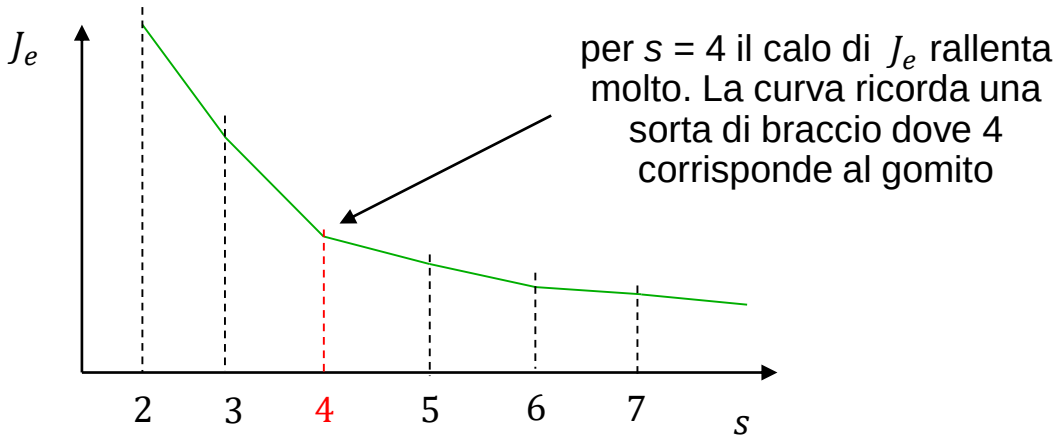
- Le tecniche di validazione tendono a valutare a posteriori la bontà delle soluzioni prodotte per diversi valori di s, e a sceglierne una sulla base di un criterio di validazione che tenga conto sia della bontà della soluzione sia della sua complessità:
  - Problema: considerando il criterio di «minimizzazione distanze dai centroidi» e lanciando K-means con diversi valori di s, è molto più facile ottenere valori ridotti di J_e per valori elevati di s. Quanto vale J_e per $s = n$ (ogni cluster un solo pattern) ?
  - Possibile soluzione: penalizzare le soluzioni con molti cluster; ad esempio:

$$J_e^* = J_e + penalty \cdot s$$

Di fatto però il problema si ribalta nella scelta di $penalty$

K-means: validazione (2)

Discontinuità nel grafico: un modo più efficace per determinare il numero ottimale di cluster è quello di cercare punti di inflessione (anche detti *elbow*) nel valore di J_e al variare di s. Infatti, se i pattern formano raggruppamenti evidenti e ben identificabili, scegliendo il corretto valore di s, dovremmo riscontrare una discontinuità nel grafico di J_e al variare di s.



Silhouette score: un criterio più oggettivo (che funziona per cluster sferici come quelli prodotti da k-means) e quello di calcolare il silhouette score come media dei *silhouette cefficient* di ciascun pattern e scegliere s che lo massimizza.

$$Silhouette\ cefficient = (b - a)/max(a, b)$$

dove a è la distanza media intra cluster del pattern e b la distanza tra il pattern e i pattern del cluster ad esso più vicino (escluso quello di appartenenza). È presente in scikit-learn la funzione *silhouette_score*() che calcola questa metrica. Risulta però computazionalmente pesante per grandi dataset.

Il *silhouette diagram* (vedi [A. Géron]) è una rappresentazione grafica degli score utile per la validation.