

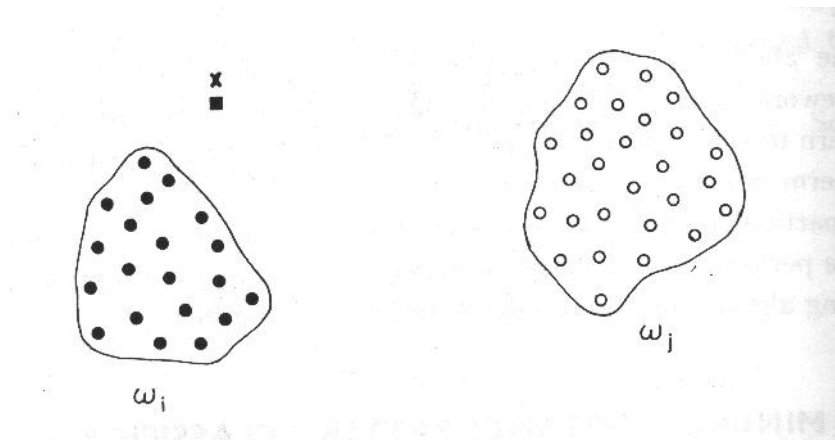
Pattern recognition

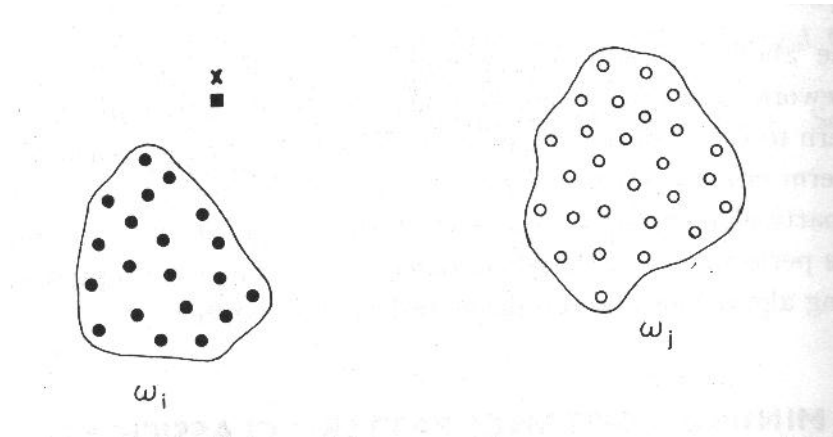
II Parte

Classificazione di pattern mediante funzioni di distanza

Per stabilire una misura di similarità fra pattern, che possono essere considerati punti in uno spazio euclideo, si può ricorrere al concetto di prossimità.

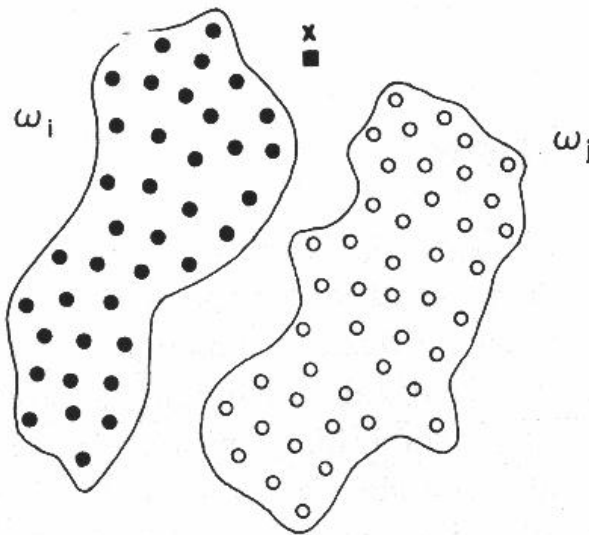
Esempio





È intuitivo concludere che \bar{x} appartiene alla classe ω_i perché più vicino ai punti che appartengono a questa classe. Questo vale quando i punti di una classe si aggregano in “cluster”.

Se ciò non accade, si possono avere difficoltà nella classificazione, come nel seguente esempio (\bar{x} è vicino sia agli elementi di ω_i e a quelli di ω_j)



Classificazione a minima distanza

È usata quando le classi dei pattern presentano un limitato grado di variabilità. Si possono verificare varie situazioni, a seconda di come si possono rappresentare le classi.

a) Singoli prototipi

Ogni classe è rappresentata da un unico elemento (prototipo). È il caso in cui la variabilità dei pattern e altre fonti di disturbo possono essere trascurate.

Esempio: riconoscimento di caratteri OCR ben stampati.

Il numero delle classi sia M ed ogni classe sia rappresentata dal pattern prototipo z_1, z_2, \dots, z_M

La distanza euclidea tra un generico vettore \bar{x} e l' i -esimo prototipo è:

$$D_i = \|\bar{x} - \bar{z}_i\| = \sqrt{(\bar{x} - \bar{z}_i)' (\bar{x} - \bar{z}_i)}$$

Il vettore \bar{x} è assegnato alla classe ω_i che soddisfa la relazione $D_i < D_j$ per tutti i $j \neq i$.

Poiché le distanze sono definite positive,
trovare il minimo D_i è equivalente a trovare il
minimo D_i^2 dove

$$D_i^2 = \|\bar{x} - \bar{z}_i\|^2 = (\bar{x} - \bar{z}_i)'(\bar{x} - \bar{z}_i) =$$

$$\bar{x}'\bar{x} - 2\bar{x}'\bar{z}_i + \bar{z}_i'\bar{z}_i = \bar{x}'\bar{x} - 2\left(\bar{x}'\bar{z}_i - \frac{1}{2}\bar{z}_i'\bar{z}_i\right)$$

Poiché $\bar{x}'\bar{x}$ è indipendente da i , il problema si
riduce a scegliere il *massimo* di

$$\left(\bar{x}'\bar{z}_i - \frac{1}{2}\bar{z}_i'\bar{z}_i\right).$$

Pertanto la funzione di decisione è:

$$d_i(\bar{x}) = \bar{x}' \bar{z}_i - \frac{1}{2} \bar{z}_i' \bar{z}_i \quad \text{con} \quad i = 1, 2, \dots, M$$

dove il pattern \bar{x} è assegnato alla classe ω_i se $d_i(\bar{x}) > d_j(\bar{x})$ per tutti i $j \neq i$

Si osservi che $d_i(\bar{x})$ è una funzione di decisione lineare: se z_{ij} con $j=1,2,\dots,n$ sono le componenti di \bar{z}_i , si può porre:

$$w_{ij}=z_{ij} \quad \text{con } j=1,2,\dots,n$$

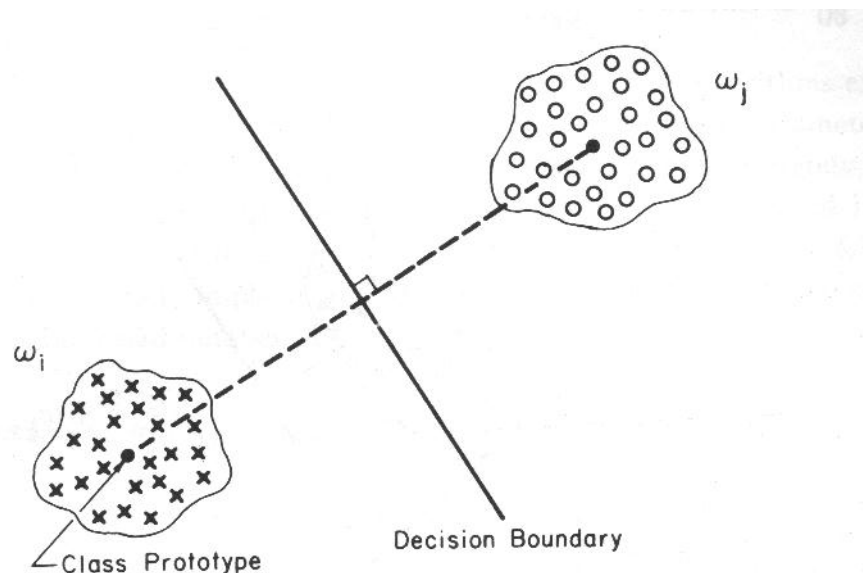
$$w_{i,n+1} = -\frac{1}{2} \bar{z}_i' \bar{z}_i$$

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{pmatrix}$$

e si può riscrivere la relazione precedente come :

$$d_i(\bar{x}) = w_i' \bar{x} \quad \text{con} \quad i = 1, 2, \dots, M$$

Esempio con due classi (due prototipi):



Si può dimostrare facilmente che la superficie di decisione che separa ogni coppia di prototipi \bar{z}_i e \bar{z}_j è l'iperpiano che biseca perpendicolarmente il segmento di linea che congiunge i due punti.

Conclusioni: il classificatore a minima distanza è un particolare caso di classificatore lineare.

Poiché questa classificazione è basata sull'accoppiamento (*match*) tra un pattern e il prototipo di classe più vicino, questo approccio è detto anche *correlazione* oppure *cluster matching*.

b) Prototipi multipli

Le considerazioni fatte nel caso di singoli prototipi si possono estendere al caso in cui i pattern di ciascuna classe ω_i tendano ad aggregarsi (“clusterizzare”) intorno ad un set di prototipi $z_i^1, z_i^2, \dots, z_i^{N_i}$, dove N_i è il numero di prototipi nella i -esima classe.

La funzione distanza tra un vettore \bar{x} ordinario e la classe ω_i è:

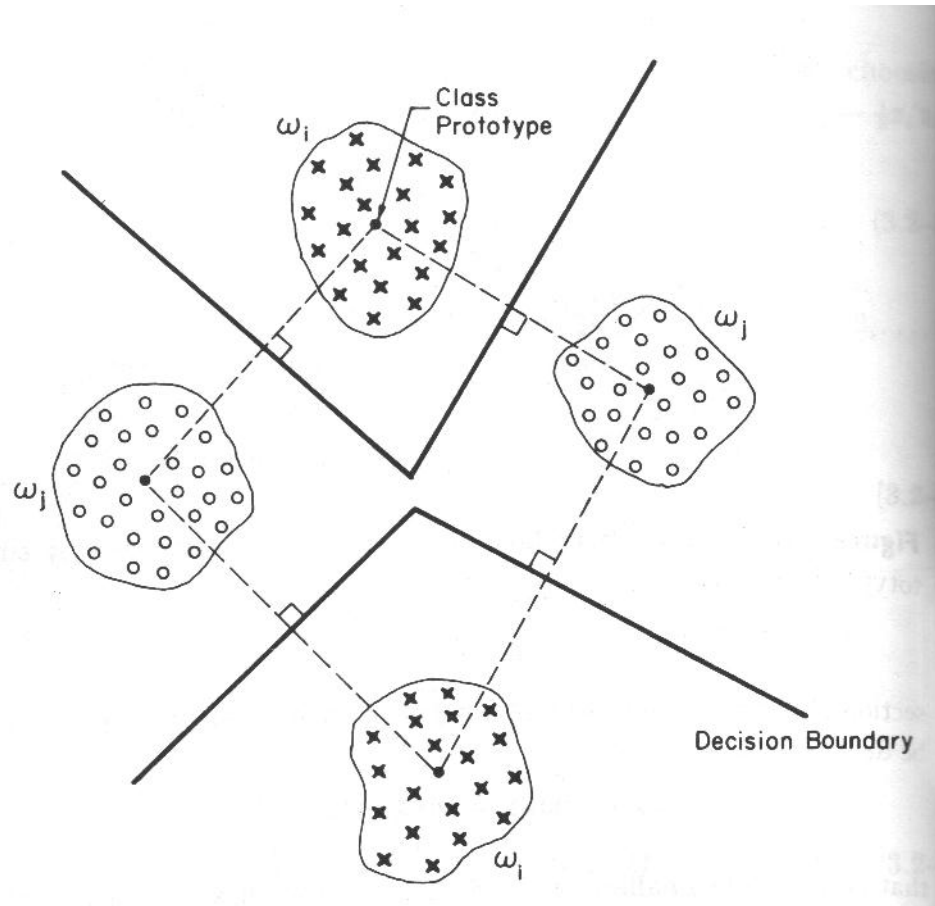
$$D_i = \min_l \left\| \bar{x} - z_i^l \right\|, \quad l = 1, 2, \dots, N_i$$

Anche qui si può ricavare la funzione di decisione che assume la forma di

$$d_i(\bar{x}) = \max_l \left\{ \left(\bar{x}' \bar{z}_i^l - \frac{1}{2} (\bar{z}_i^l)' (\bar{z}_i^l) \right) \right\} \quad \text{con } l = 1, 2, \dots, N_i$$

e \bar{x} è assegnata alla classe ω_i se $d_i(\bar{x}) > d_j(\bar{x})$ per tutti i $j \neq i$.

Esempio con due classi, ciascuno rappresentato da due prototipi:



Il confine tra le classi ω_i e ω_j è una spezzata (*piecewise linear boundaries*).

L'approccio basato sulla minima distanza è un caso particolare di una forma più generale di classificatori a spezzate (*piecewise linear classifier*).

Le funzioni di decisione in questo caso assumono la forma:

$$d_i(\bar{x}) = \max_l \{d_i^l(\bar{x})\} \quad \text{con} \quad i = 1, 2, \dots, M \quad \text{e} \\ l = 1, 2, \dots, N_i$$

Dove ciascuna distanza è

$$d_i^l(\bar{x}) = w_{i_1}^l x_1 + w_{i_2}^l x_2 + \dots + w_{i_n}^l x_n + w_{i,n+1}^l x = \bar{w}_i^l \bar{x}$$

(si noti che queste funzioni non corrispondono alle spezzate della figura precedente).

Estensione del concetto di minima distanza

Nel caso più generale, si ha un set di pattern di classificazione noto $\{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N\}$ dove si assume che ciascun pattern appartiene ad una delle classi $\omega_1, \omega_2, \dots, \omega_M$.

Si definisce regola di classificazione dell'intorno più vicino (*nearest neighbor, NN*) quella che assegna un vettore incognito alla classe del vicino più prossimo, dove si dice che $\bar{s}_i (\in \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_N\})$ è il vicino più prossimo (*nearest neighbor*) se

$$D(\bar{s}_i, \bar{x}) = \min_l \{D(\bar{s}_l, \bar{x})\} \quad l = 1, 2, \dots, N$$

Se si usa un solo vicino per la classificazione, questo schema è detto anche “*1-NN rule*”.

Se, invece, si considerano q vicini a \bar{x} e si adotta un criterio di maggioranza, la regola è detta “*q-NN rule*” (si assegna \bar{x} alla maggioranza dei q vicini più prossimi).

Come è intuitivo, in questo modo si riduce la probabilità di errata attribuzione.

Esempio di realizzazione:

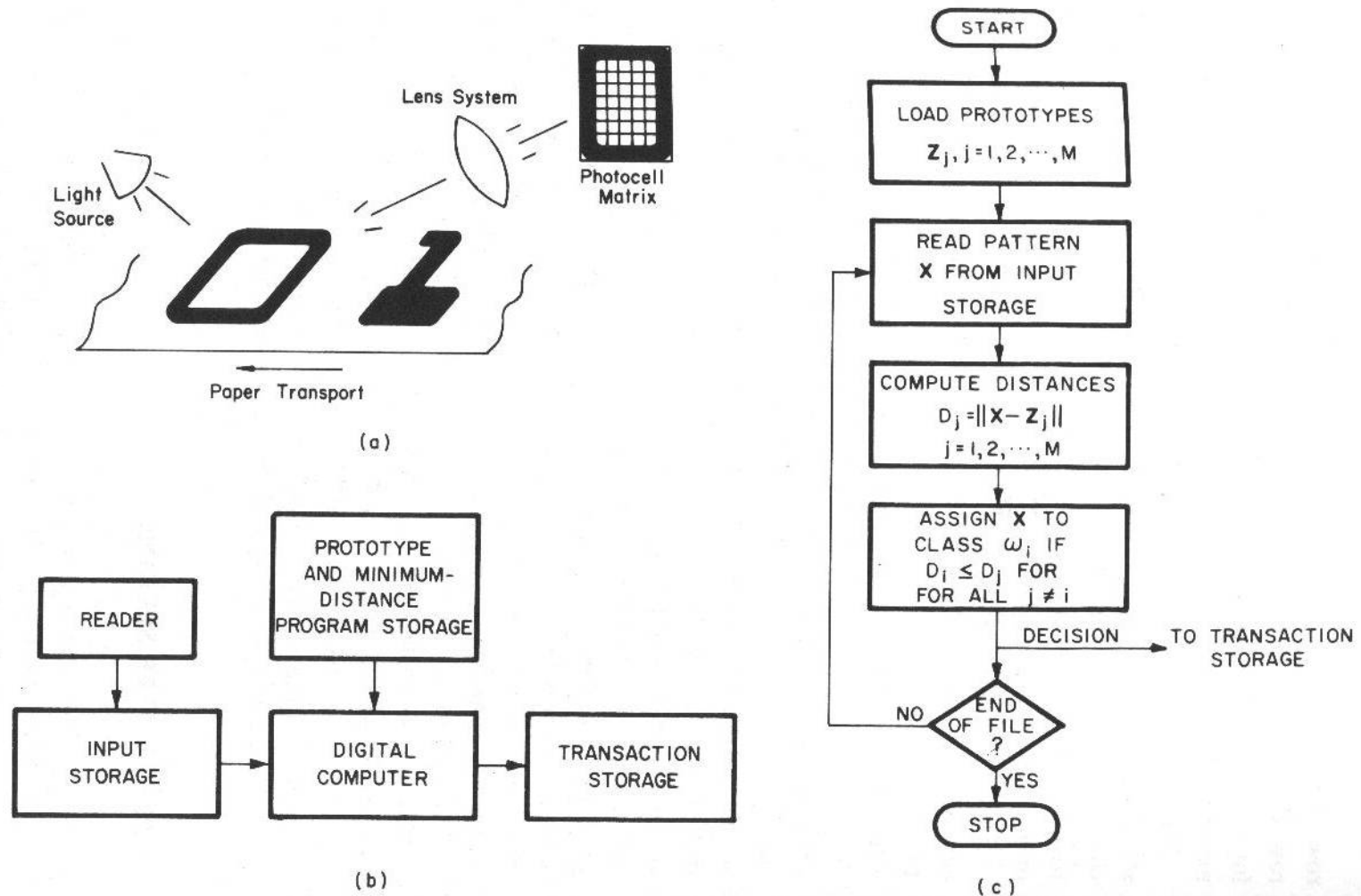


Figure 3.6. Stylized character recognition system. (a) Optical reader. (b) Computer system. (c) Basic software flowchart

La formazione dei cluster

Gioca un ruolo fondamentale per la classificazione basata sul criterio di minima distanza.

La definizione dei cluster di dati deve essere fondata di una misura di similarità.

Un criterio, già introdotto, è quello basato sulla distanza definita come

$$D = \|\bar{x} - \bar{z}\|$$

Il criterio stabilisce che “minore distanza \rightarrow più grande similarità”.

Ovviamente non è l'unica definizione di distanza che si possa adottare.

Se ad esempio si tiene conto di proprietà statistiche, una definizione di distanza molto usata è quella detta “distanza di Mahalanobis”:

$$D = (\bar{x} - \bar{m})' \bar{C}^{-1} (\bar{x} - \bar{m})$$

dove

\bar{x} è una variabile pattern

\bar{m} rappresenta il vettore medio

\bar{C} è la matrice di covarianza di una popolazione di pattern.

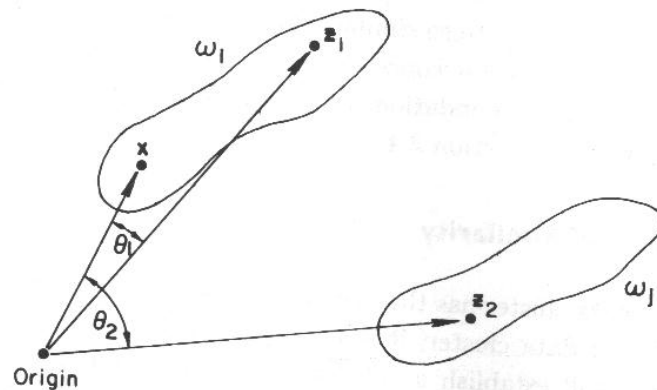
Le misure di similarità non sono necessariamente ristrette alle misure di distanza.

È infatti importante rilevare dai dati (e dal problema) l'attributo - o gli attributi - comuni che rendono simili i simili e differenziano i dissimili.

Se, ad esempio, le regioni di cluster tendono a distendersi lungo degli assi, si può scegliere come misura di similarità l'angolo tra i vettori (o un parametro a questo correlato).

Ad esempio,
$$S(\bar{x}, \bar{r}) = \frac{\bar{x}' \bar{r}}{\|\bar{x}\| \|\bar{r}\|}$$

fornisce il coseno tra i vettori \bar{x} e \bar{r} , ovvero la proiezione di uno sull'altro:



$$s(\bar{x}, \bar{z}_1) = \cos \theta_1 = \frac{\bar{x}' \bar{z}_1}{\|\bar{x}\| \|\bar{z}_1\|}$$

$$s(\bar{x}, \bar{z}_2) = \cos \theta_2 = \frac{\bar{x}' \bar{z}_2}{\|\bar{x}\| \|\bar{z}_2\|}$$

\bar{x} è più simile a \bar{r}_1 che a \bar{r}_2 perché $\cos(\theta_1) > \cos(\theta_2)$

Lo stesso indice di similarità, $\left(\frac{\bar{x}' \bar{r}}{\|\bar{x}\| \|\bar{r}\|} \right)$ può avere altre interpretazioni.

Se i pattern hanno le componenti che assumono valori binari (0,1), il termine $\bar{x}' \bar{r}$ fornisce il numero di componenti (*attributi*) che \bar{x}' e \bar{r} condividono (infatti \bar{x}' possiede l'*i*-esimo attributo se $x_i=1$!).

Poiché inoltre $\|\bar{x}\| \cdot \|\bar{r}\| = \sqrt{(\bar{x}' \bar{x})(\bar{r}' \bar{r})}$ può essere interpretato come il numero di attributi posseduti da \bar{x}' e \bar{r} , il numero di componenti comuni è “normalizzato” rispetto al numero totale di componenti presenti.

Nelle applicazioni di recupero dell'informazione (information retrieval), di nosologia e di tassonomia si usa una variante detta misura di Tanimoto così definita:

$$S(\bar{x}, \bar{r}) = \frac{\bar{x}' \bar{r}}{\bar{x}' \bar{x} + \bar{r}' \bar{r} - \bar{x}' \bar{r}}$$

che ha l'ovvia interpretazione di rapporto tra elementi comuni ed elementi dissimili nei due vettori.

Criteri di clustering

Dopo aver adottato una misura di similarità, occorre fissare il criterio di aggregazione: può essere basato su uno schema euristico oppure sulla minimizzazione (o massimizzazione) di qualche indice di prestazione.

Ad esempio, ricade nel primo gruppo il criterio basato sulla distanza euclidea: infatti la vicinanza di due pattern è una misura “relativa” di similarità, per cui occorre stabilire ancora una soglia al fine di definire il grado di accettabilità di un pattern come simile ad un altro (o appartenente ad un cluster).

Nel secondo gruppo ricade, ad esempio, il criterio di minimizzare l'errore quadratico medio definito come

$$J = \sum_{j=1}^{N_c} \sum_{\bar{x} \in S_j} \|\bar{x} - \bar{m}_j\|^2$$

dove:

N_c è il numero dei cluster

S_j è il set di campioni appartenenti al j-esimo cluster

$\bar{m}_j = \frac{1}{N_j} \sum_{x \in S_j} \bar{x}$ è il vettore medio del set S_j

Esistono anche algoritmi ibridi, che mescolano l'ottimizzazione rispetto a qualche indice con l'euristica.

Un semplice algoritmo di clusterizzazione

- Si abbiano N pattern $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_N\}$.
- Si sceglie un pattern a caso come centro del primo cluster, cioè come \bar{z}_1 .
- Si sceglie ancora una soglia non negativa T .
- Si supponga di aver scelto \bar{x}_1 come \bar{z}_1 (il primo del set).

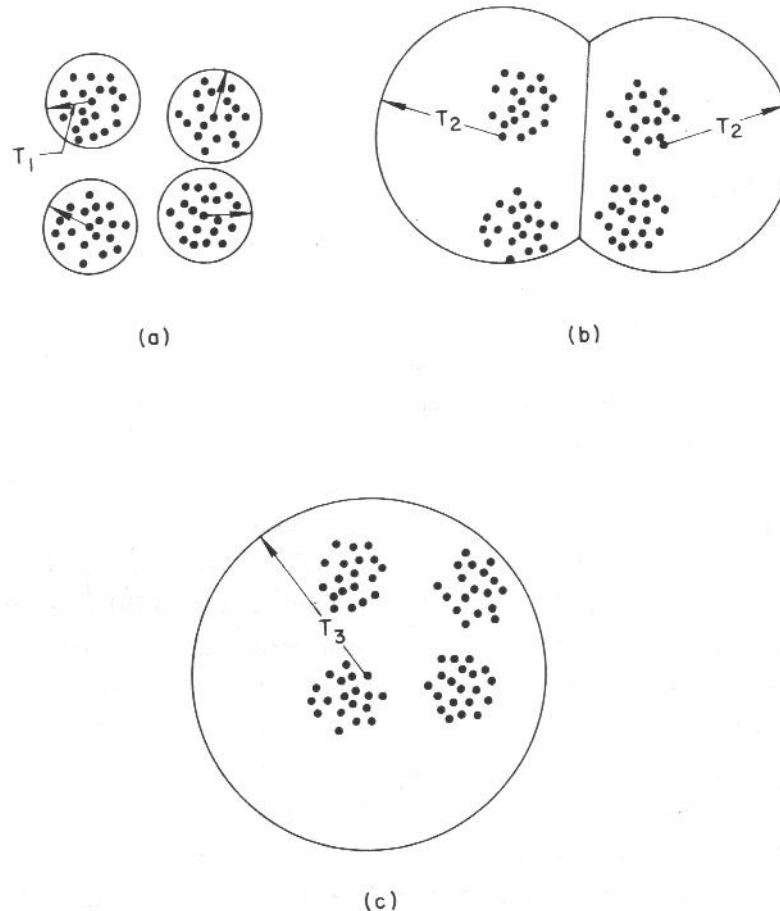
- Si calcola poi la distanza di \bar{x}_2 da \bar{z}_1 : sia D_{21} questa distanza.
- Se $D_{21} > T$, allora \bar{x}_2 è assunto come \bar{z}_2 , il centro del secondo cluster, altrimenti si scarta come appartenente al primo cluster.
- Si supponga $D_{21} > T$: si calcola allora D_{31} e D_{32} , rispettivamente la distanza di \bar{x}_3 da \bar{z}_1 e \bar{z}_2 . Se sono entrambe $> T$, allora \bar{x}_3 è il centro del terzo cluster, altrimenti si assegna al cluster più prossimo.

Si procede così per tutti gli altri pattern del set.

Si osservi che il risultato dipende da:

- La scelta iniziale
- l'ordine con cui si analizzano i pattern
- la soglia T

Ad esempio, al variare di T si possono avere situazioni come nella seguente figura:



Nonostante i limiti evidenti, è un algoritmo semplice, rapido (il set di pattern è visitato una volta sola), può dare indicazioni di massima sulla disposizione dei dati.

Nella pratica, si usano alcuni accorgimenti aggiuntivi.

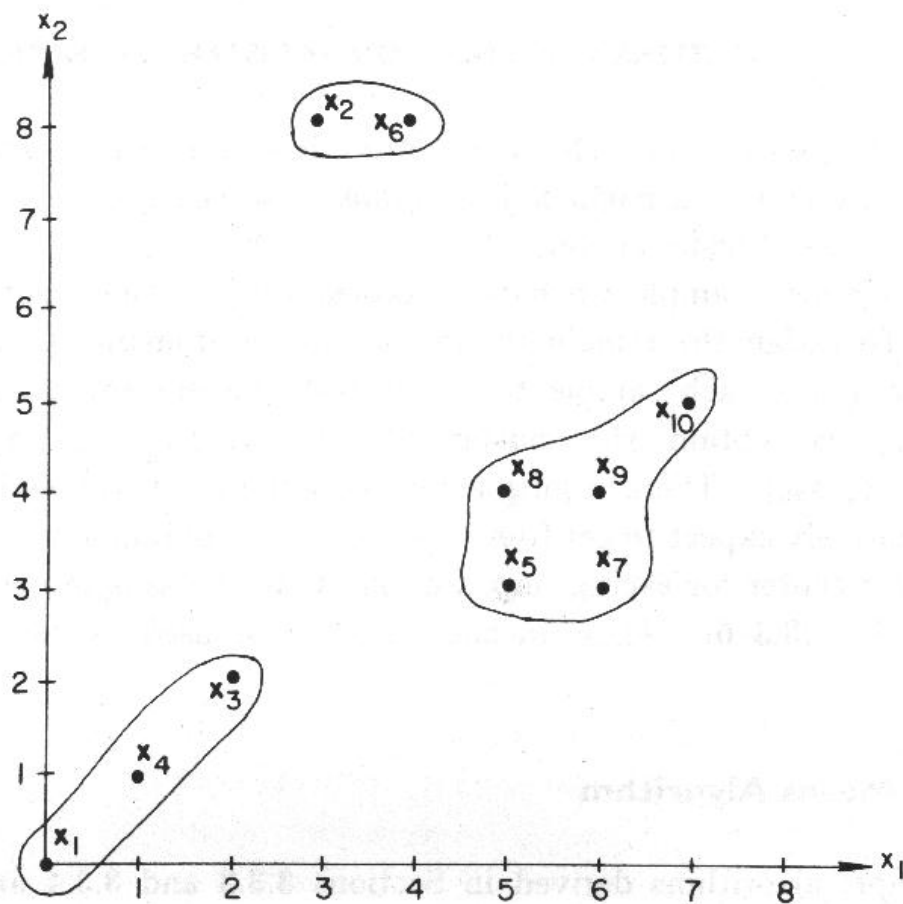
- Si prova con diversi valori di soglia;
- si considerano le distanze tra i centri dei cluster e il numero di campioni che cadono in ciascun cluster;
- si considera la distanza dal centro del più vicino e del più lontano pattern per ciascun cluster, nonché la varianza.

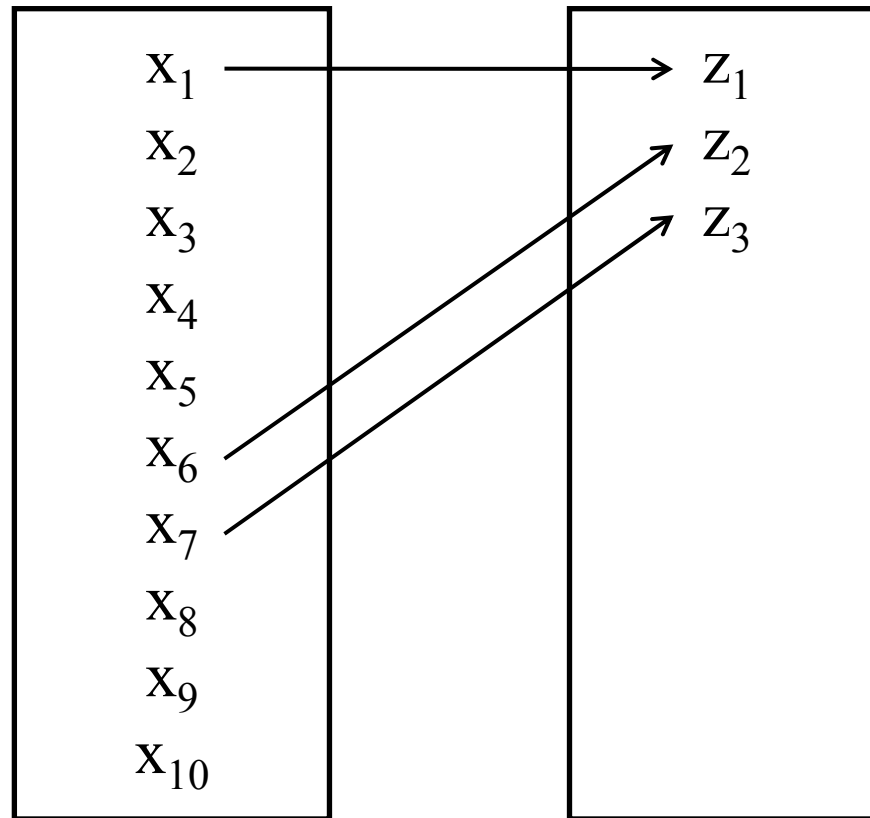
Questa procedura è efficace particolarmente quando i dati esibiscano un'aggregazione caratteristica “a sacchetti” (o “nuvolette”).

Algoritmo Maximin-Distance

È simile al precedente, ma cerca prima i cluster più lontani.

Si supponga che i pattern siano disposti come nella figura seguente.





Si pongono i campioni in una tabella (a sinistra). All'inizio la tabella di destra è vuota.

Si sceglie un campione a caso, ad esempio \bar{x}_1 , come primo centro di cluster, e lo si designa come \bar{z}_1 .

A questo punto si seleziona il punto più lontano, \bar{x}_6 , come secondo centro di cluster \bar{z}_2 .

Si calcola la distanza degli altri punti rispetto a \bar{z}_1 e \bar{z}_2 , memorizzando la minima tra le due.

Si seleziona la massima tra queste distanze minime: se è una frazione apprezzabile della distanza tra \bar{z}_1 e \bar{z}_2 , per esempio la metà, si sceglie il campione corrispondente come centro di cluster \bar{z}_3 (nell'esempio \bar{x}_7).

Si procede così, selezionando sempre la massima delle distanze minime. Se questa distanza è una frazione apprezzabile delle “tipiche” distanze massime precedenti (per esempio si sceglie il valore medio come valore “tipico”), si designa il pattern corrispondente come nuovo centro di cluster, altrimenti l'algoritmo si ferma.

Alla fine si assegnano i pattern al cluster di appartenenza e, per avere un valore più rappresentativo, si sceglie il valore medio calcolato all'interno del cluster come centro \bar{z}_i .

Algoritmo k-Means

È basato sul criterio di minimizzare un indice di prestazione, definito come somma dei quadrati delle distanze tra tutti i punti di ciascun cluster rispetto al proprio centro di cluster.

È un algoritmo basato sui seguenti passi:

1. Si scelgono k centri di cluster iniziali in modo arbitrario, e si designano come $\bar{z}_1(1), \bar{z}_2(1), \dots, \bar{z}_k(1)$.
Ad esempio, si scelgono i primi k campioni.
2. Alla k -esima iterazione, si distribuiscono i campioni $\{\bar{x}\}$ tra i k cluster usando la relazione

$$\bar{x} \in S_j(k) \quad \text{se} \quad \|\bar{x} - \bar{z}_j(k)\| < \|\bar{x} - \bar{z}_i(k)\|$$
$$\text{per } i = 1, 2, \dots, k \quad \text{e} \quad i \neq j$$

$S_j(k)$ indica il cluster che ha come centro $\bar{z}_j(k)$

3. Si calcola per ogni cluster il nuovo centro $\bar{z}_j(k+1)$ con $j = 1, 2, \dots, k$ in modo che sia minimizzata la distanza quadratica di tutti i punti del cluster rispetto al centro. In formule, si vuole che sia minimo l'indice di prestazione

$$J_j = \sum_{\bar{x} \in S_j} \|\bar{x} - \bar{z}_j(k+1)\|^2 \quad \text{con} \quad j = 1, 2, \dots, k$$

È noto che basta calcolare il valore medio di \bar{x} :

$$\bar{z}_j(k+1) = \frac{1}{N_j} \sum_{\bar{x} \in S_j(k)} \bar{x}$$

donde il nome “k-Means”

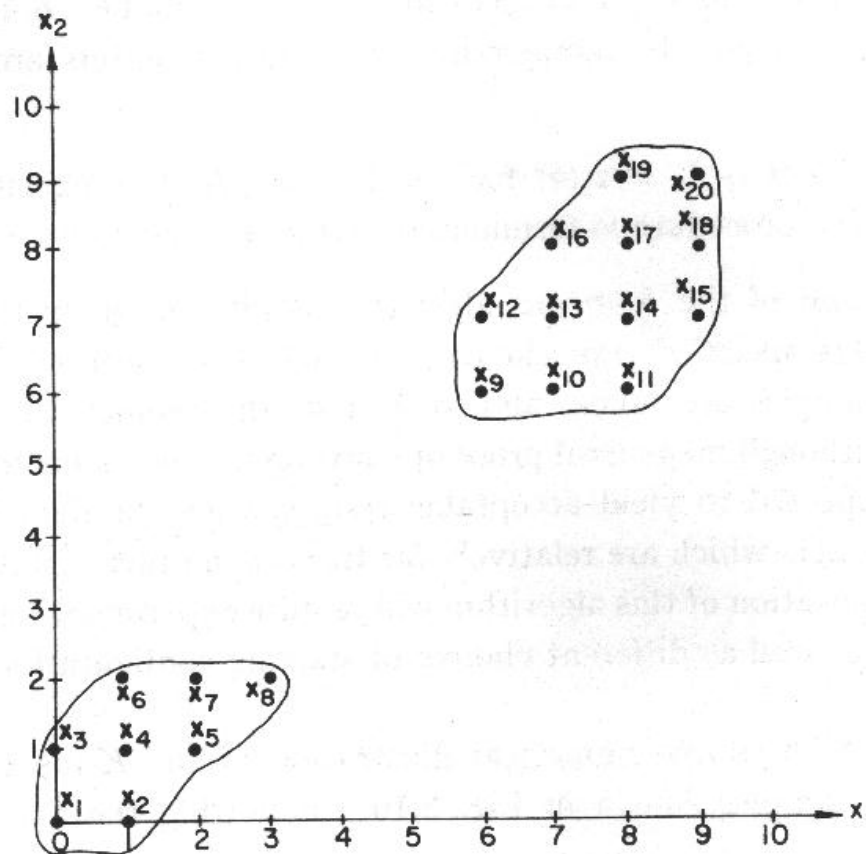
4. Se $\bar{z}_j(k+1) = \bar{z}_j(k)$ per $j = 1, 2, \dots, k$ la procedura è terminata, altrimenti si va al passo 2.

Il comportamento di questo algoritmo, di cui non esiste una prova di convergenza di tipo generale, è influenzato da:

- il numero k di centri prescelto
- la scelta iniziale dei centri di cluster
- l'ordine con cui vengono presi i campioni
- le proprietà geometriche (di disposizione) dei dati.

Esempio numerico

Siano dati i campioni in figura:



1. Si ipotizza $k = 2$. Si sceglie
 $\bar{z}_1(1) = \bar{x}_1 = (0,0)'$, $\bar{z}_2(1) = \bar{x}_2 = (1,0)'$
2. Poiché $\|\bar{x}_1 - \bar{z}_1(1)\| < \|\bar{x}_1 - \bar{z}_i(1)\|$, con $i = 2$, \bar{x}_1 appartiene a $S_1(1)$. Idem per \bar{x}_3 . Gli altri campioni sono più prossimi a $\bar{z}_2(1)$.

Quindi:

$$S_1(1) = \{\bar{x}_1, \bar{x}_3\}, S_2(1) = \{\bar{x}_2, \bar{x}_4, \dots, \bar{x}_{20}\}$$

3. Si aggiorna il valore dei centri dei cluster

$$\bar{z}_1(2) = \frac{1}{N_1} \sum_{\bar{x} \in S_1(1)} \bar{x} = \frac{1}{2} (\bar{x}_1 + \bar{x}_2) = \begin{pmatrix} 0.0 \\ 0.5 \end{pmatrix}$$

$$\bar{z}_2(2) = \frac{1}{N_2} \sum_{\bar{x} \in S_2(1)} \bar{x} = \frac{1}{18} (\bar{x}_2 + \bar{x}_4 + \dots + \bar{x}_{20}) = \begin{pmatrix} 5.67 \\ 5.33 \end{pmatrix}$$

4. Poiché $\bar{z}_j(2) \neq \bar{z}_j(1)$ con $j=1, 2$ si va al passo 2

5. Ricalcolando le distanze rispetto ai nuovi centri si conclude che

$$S_1(2) = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_8\},$$
$$S_2(2) = \{\bar{x}_9, \bar{x}_{10}, \dots, \bar{x}_{20}\}$$

6. Si calcolano di nuovo i centri dei cluster

$$\bar{z}_1(3) = \frac{1}{8} \{\bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_8\} = \begin{pmatrix} 1.25 \\ 1.13 \end{pmatrix},$$

$$\bar{z}_2(3) = \frac{1}{12} \{\bar{x}_9 + \bar{x}_{10} + \dots + \bar{x}_{20}\} = \begin{pmatrix} 7.67 \\ 7.33 \end{pmatrix}$$

7. È ancora $\bar{z}_J(3) \neq \bar{z}_J(2)$, si torna al passo 2

8. Si ottiene lo stesso risultato dell'iterazione precedente, per cui

$$S_1(4) = S_1(3) \quad \text{e} \quad S_2(4) = S_2(3)$$

9. Idem come operazione precedente

10. Ora è $\bar{z}_J(4) = \bar{z}_J(3)$, l'algoritmo finisce.

Si osservi che nel caso più generale non è possibile una ispezione visiva (dimensioni > 2) e non è possibile valutare a priori k : si procede per tentativi, con vari valori di k .

Algoritmo Isodata

L'algoritmo Isodata (Iterative Self-Organizing Data Analysis Technique A¹) è simile alla procedura K-Means: si aggiungono in più delle procedure euristiche (importante, tra le altre, quella detta di *split and merge*).

All'inizio si fissa un numero di cluster iniziali N_c (non necessariamente quello che si desidera come definitivo) e i loro centri (arbitrari) $\bar{z}_1, \bar{z}_2, \dots, \bar{z}_{N_c}$

¹ La A finale è aggiunta per comodità di pronuncia

Passo 1) Si specificano i seguenti parametri di processo:

K numero dei centri di cluster desiderati

\mathcal{G}_N numero di campioni con cui confrontare il numero di campioni in un cluster

\mathcal{G}_S parametro di deviazione standard

\mathcal{G}_c parametro di aggregazione di un blocco

L massimo numeri di coppie di centri di cluster che possono essere aggregate

I numero di iterazioni permesse

Passo 2) Si distribuiscono gli N campioni $\{\bar{x}_1, \dots, \bar{x}_N\}$ tra gli attuali centri di cluster in base alla relazione: $\bar{x} \in S_j$ se $\|\bar{x} - \bar{z}_j\| < \|\bar{x} - \bar{z}_i\|$, con $i = 1, 2, \dots, N_c; i \neq j$

Passo 3) Si scaricano i cluster con meno di \mathcal{G}_N campioni; formalmente:
Se per un j è $N_j < \mathcal{G}_N$, allora si elimina S_j e si riduce N_C di 1.

Passo 4) Si ricalcolano i centri dei cluster \bar{z}_j , $j = 1, 2, \dots, N_c$, ponendoli uguali alla media dei campioni in ciascun cluster,

$$\bar{z}_j = \frac{1}{N_j} \sum_{\bar{x} \in S_j} \bar{x}, \quad \text{con } j = 1, 2, \dots, N_c$$

Passo 5) Si calcola la distanza media

$$\bar{D}_j = \frac{1}{N_j} \sum_{\bar{x} \in S_j} \|\bar{x} - \bar{z}_j\| \quad \text{con } j = 1, 2, \dots, N_c$$

Passo 6) Si calcola la media pesata delle distanze medie, cioè

$$\overline{D} = \frac{1}{N} \sum_{j=1}^{N_c} N_j \overline{D}_j$$

Passo 7)

- a) Se questa è l'ultima iterazione porre $\mathcal{G}_c = 0$ e andare al passo 11.
- b) Se $N_c \leq k/2$ (pochi cluster rispetto al previsto), andare al passo 8 (fase di *split*).
- c) Se questa è un'iterazione pari oppure $N_c \geq 2k$ (troppi cluster), andare al passo 11 (fase di *merge*); altrimenti continuare.

Passo 8) Calcolare il vettore di deviazioni standard $\bar{\sigma}_j = (\sigma_{1j}, \sigma_{2j}, \dots, \sigma_{nj})'$ per ciascun cluster, secondo la relazione

$$\sigma_{ij} = \sqrt{\frac{1}{N_j} \sum_{\bar{x} \in S_j} (x_{ik} - z_{ij})^2}, i = 1, 2, \dots, n, j = 1, 2, \dots, N_c$$

dove :

- n è la dimensionalità dei vettori \bar{x}
- x_{ik} è la i -esima componente del k -esimo campione di S_j
- z_{ij} è la i -esima componente del j -esimo centro di cluster

In pratica, il vettore rappresenta la deviazione standard dei campioni in lungo gli assi principali delle coordinate.

Passo 9) Si cerca la componente massima di ciascun $\bar{\sigma}_j$, con $j = 1, 2, \dots, N_c$ e si denota come $\sigma_{j \max}$

Passo 10) Se per ogni $\sigma_{j \max}$ ci si trova di fronte a questa situazione:

$$\sigma_{j \max} > \mathcal{I}_S \text{ e } \begin{cases} a) \bar{D}_j > \bar{D} \text{ e } N_j > 2(\mathcal{I}_N + 1) \\ \text{oppure} \\ b) N_C \leq K / 2 \end{cases}$$

si spezza \bar{z}_j in due cluster \bar{z}_j^+ e \bar{z}_j^- .

Il nuovo centro di cluster \bar{z}_j^+ è ricavato da \bar{z}_j sommando una data quantità γ_j alla componente di \bar{z}_j che corrisponde alla massima componente di $\bar{\sigma}_j$;

per \bar{z}_j^- invece γ_j viene sottratto. γ_j in pratica viene calcolato come $\gamma_j = k\sigma_{j_{\max}}$ con $0 \leq k \leq 1$.

L'obiettivo è quello di creare una perturbazione, cioè una differenza sensibile nella distanza tra un generico campione e i due nuovi centri di cluster, senza tuttavia modificare la disposizione degli altri cluster in modo apprezzabile. Se avviene lo “splitting”, si va al passo 2 altrimenti si continua al passo 14;

(è evidente che la \bar{z}_j originaria scompare, N_C aumenta di 1)

Passo 10bis) Vai a 14)

Passo 11) Si calcolano le distanze a coppie D_{ij} tra i centri dei cluster:

$$D_{ij} = \|\bar{z}_i - \bar{z}_j\|, \quad i = 1, 2, \dots, N_C - 1 \text{ e } j = 1, 2, \dots, N_C$$

Passo 12) Si confrontano le distanze D_{ij} con il parametro \mathcal{G}_c . Si ordinano le L più piccole distanze che sono minori di \mathcal{G}_c in ordine crescente: $[D_{i_1 j_1}, D_{i_2 j_2}, \dots, D_{i_L j_L}]$ (si ricordi che L è il numero massimo di centri di cluster che si possono aggregare).

Passo 13) A partire dalla distanza più piccola $D_{i_e j_e}$ si aggregano i centri a coppie \bar{z}_{i_e} e \bar{z}_{j_e} (a condizione che \bar{z}_{i_e} e \bar{z}_{j_e} non siano già stati usati per una aggregazione).

In pratica da \bar{z}_{i_e} e \bar{z}_{j_e} si crea un nuovo centro come somma pesata dei due (i pesi sono il numero di campioni in ogni cluster)

$$\bar{z}_e^* = \frac{1}{N_{i_e} + N_{j_e}} [N_{i_e} \cdot (\bar{z}_{i_e}) + N_{j_e} \cdot (\bar{z}_{j_e})]$$

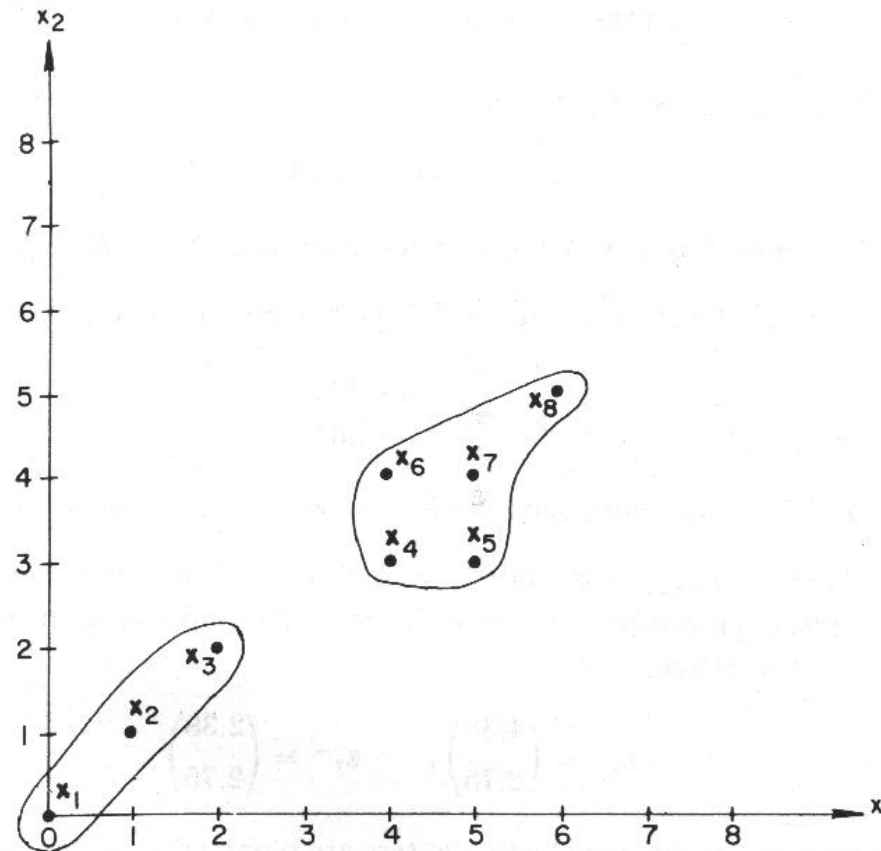
(Praticamente \bar{z}^* è il baricentro)

Si cancellano quindi \bar{z}_{ie} e \bar{z}_{je} , riducendo N_C di 1. Si osservi che, poiché ogni centro di cluster è usato al più una volta, non è detto che si riescano a fare L aggregazioni.

Passo 14) Se questa è l'ultima iterazione, fine. Altrimenti andare al passo 2 (è previsto che si possa andare anche al passo 1 per modificare qualche parametro, se l'utente lo ritiene opportuno per una migliore adattatività).

Esempio numerico

Con i dati dell'esempio precedente ($N=8$, $n=2$)



Si può partire con $N_C=1$ e $\bar{z}_1 = (0,0)'$ e i parametri scelti come:

$$k = 2, \mathcal{G}_N = 1, \mathcal{G}_S = 1, \mathcal{G}_C = 4, L = 0, I = 4$$

e si può verificare che si raggiungono risultati molto vicini ai precedenti (si ha uno “splitting” ... si suggerisce di ripetere i passi dell'algoritmo per esercizio).