# Apprendimento Non Supervisionato

*Unsupervised Learning*

# Supervisione nell'Apprendimento

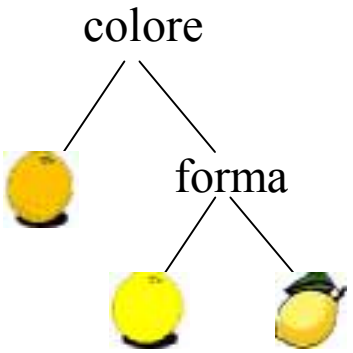
(arancio, rotondo, classe=  )

(giallo, lungo, classe=  )

(giallo, rotondo, classe=  )

(giallo, lungo, classe=  )

algoritmo di apprendimento
**supervisionato**



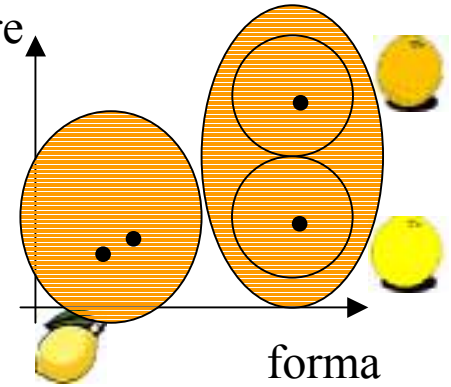

(arancio, rotondo)

(giallo, rotondo)

(giallo, rotondo)

(giallo, lungo)

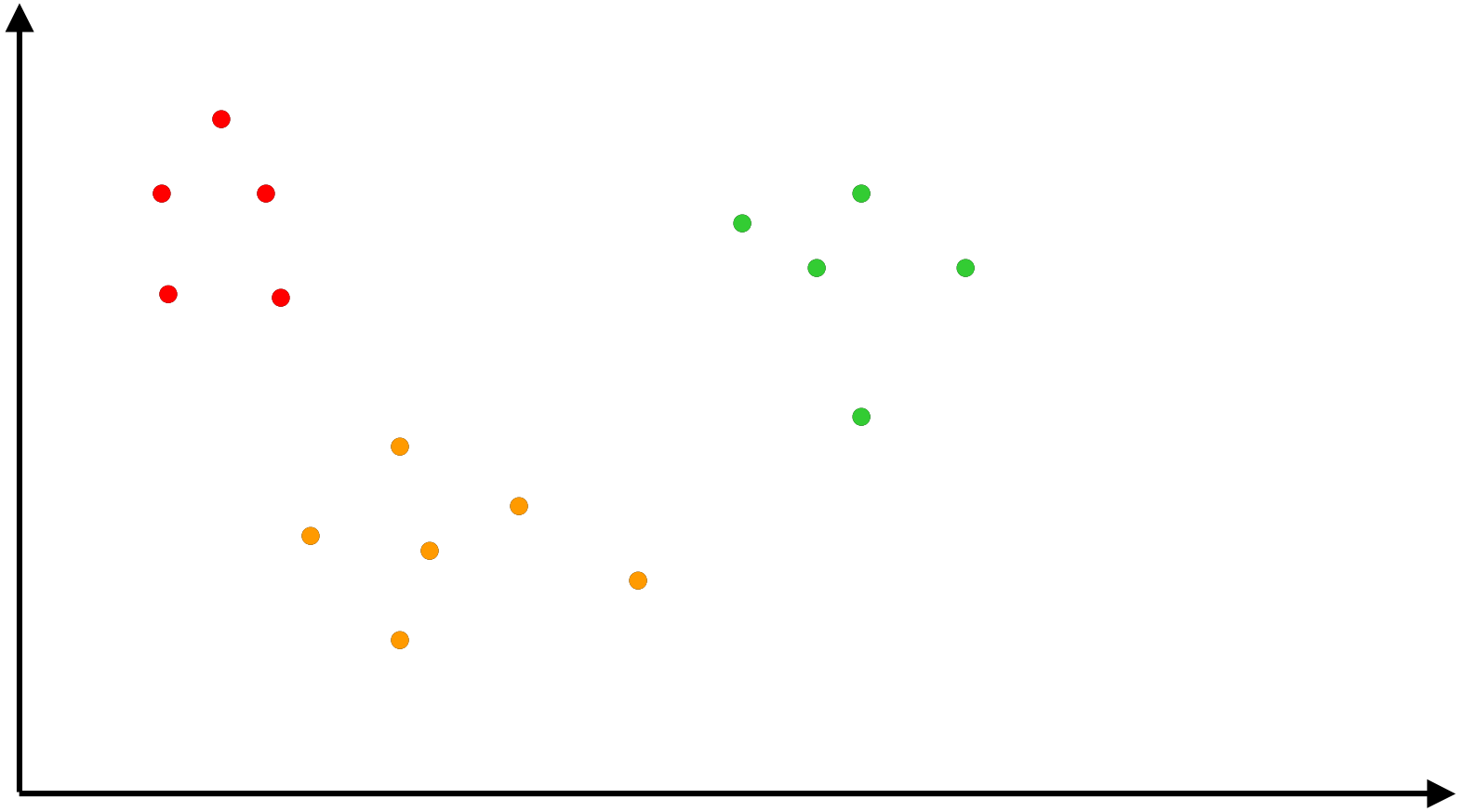
algoritmo di apprendimento
**non supervisionato**



# Clustering

- Suddivide esempi non etichettati in sottoinsiemi disgiunti (**cluster**), tali che:
  - Gli esempi in uno stesso gruppo sono “molto” simili
  - Gli esempi in gruppi diversi sono “molto” differenti
- Scopre **nuove categorie** in modo **non supervisionato** (a priori non vengono fornite etichette per le categorie)

# Clustering: un esempio

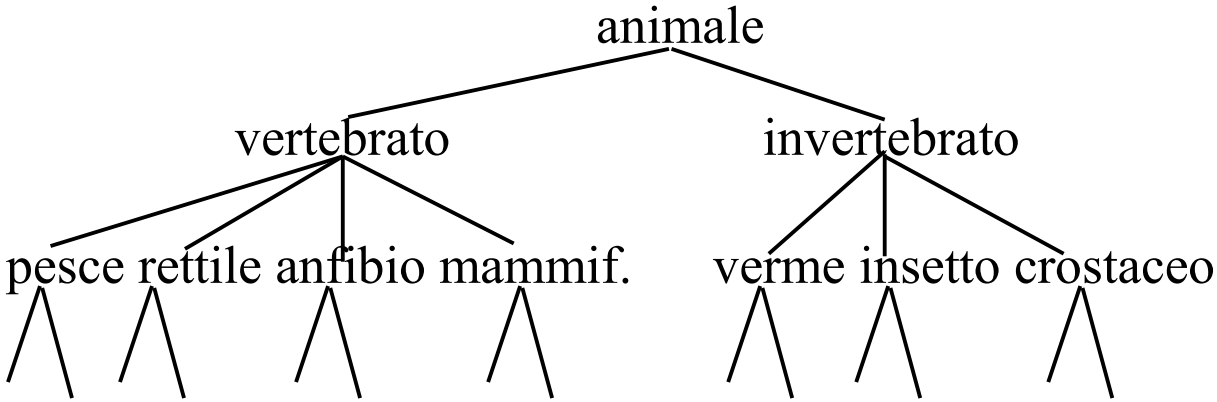


# Tipi di Clustering

- Clustering **gerarchico** (hierarchical clustering)
  - Formano cluster iterativamente utilizzando cluster precedentemente costituiti
- Clustering **partitivo** (partitional clustering)
  - Crea una sola partizione degli esempi in cluster minimizzando una certa funzione di costo

# Clustering Gerarchico

- Costruisce una tassonomia gerarchica ad albero a partire da un insieme di esempi non etichettati




- L'applicazione ricorsiva di un algoritmo di clustering può produrre un clustering gerarchico
- Distinguiamo due tipi di clustering gerarchico:
  - Agglomerativo (bottom-up)
  - Divisivo (top-down)

# Clustering Partitivo

- I metodi di clustering partitivo ottengono una singola partizione dei dati, invece di una struttura di clustering (es. albero di clustering)
- Richiedono di specificare il numero di cluster *k* desiderati
- Il numero di cluster *k* può essere determinato automaticamente generando esplicitamente clustering per diversi valori di k e scegliendo il miglior risultato secondo la funzione di valutazione del clustering

# Clustering Gerarchico Agglomerativo

- Assume l'esistenza di una **funzione di similarità** per determinare la similarità di due istanze
- Algoritmo:

  Parti con un cluster per ogni istanza

  **Finché** non c'è un solo cluster:

    Determina i due cluster $c_i$ e $c_j$ più simili

    Sostituisci $c_i$ e $c_j$ con un singolo cluster $c_i \cup c_j$
- La "storia" di fusione costituisce un albero binario o gerarchia di clustering (**dendrogramma**)

# Metriche per determinare la distanza

- Nota: se la distanza è normalizzata tra 0 e 1, la similarità $sim(x, y)$ è data da 1-$d(x, y)$
- Distanza euclidea (norma $L_2$):

$$L_2(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

- Norma $L_1$:

$$L_1(\vec{x}, \vec{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

- Cosine Similarity (trasformata in una distanza sottraendo da 1):

$$1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|} = 1 - \frac{\sum_{i=1}^{m} x_i y_i}{\sqrt{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2}}$$

# Misurare la Similarità tra Cluster

- Nel clustering gerarchico agglomerativo, utilizziamo una **funzione di similarità** che determina la similarità tra due istanze: $sim(x, y)$
- Come calcolare la similarità di due cluster $c_i$ e $c_j$ sapendo come calcolare la similarità tra due istanze nei due cluster?
  - **Single Link**: Similarità dei due membri più simili
  - **Complete Link**: Similarità dei due membri meno simili
  - **Group Average**: Similarità media tra i membri

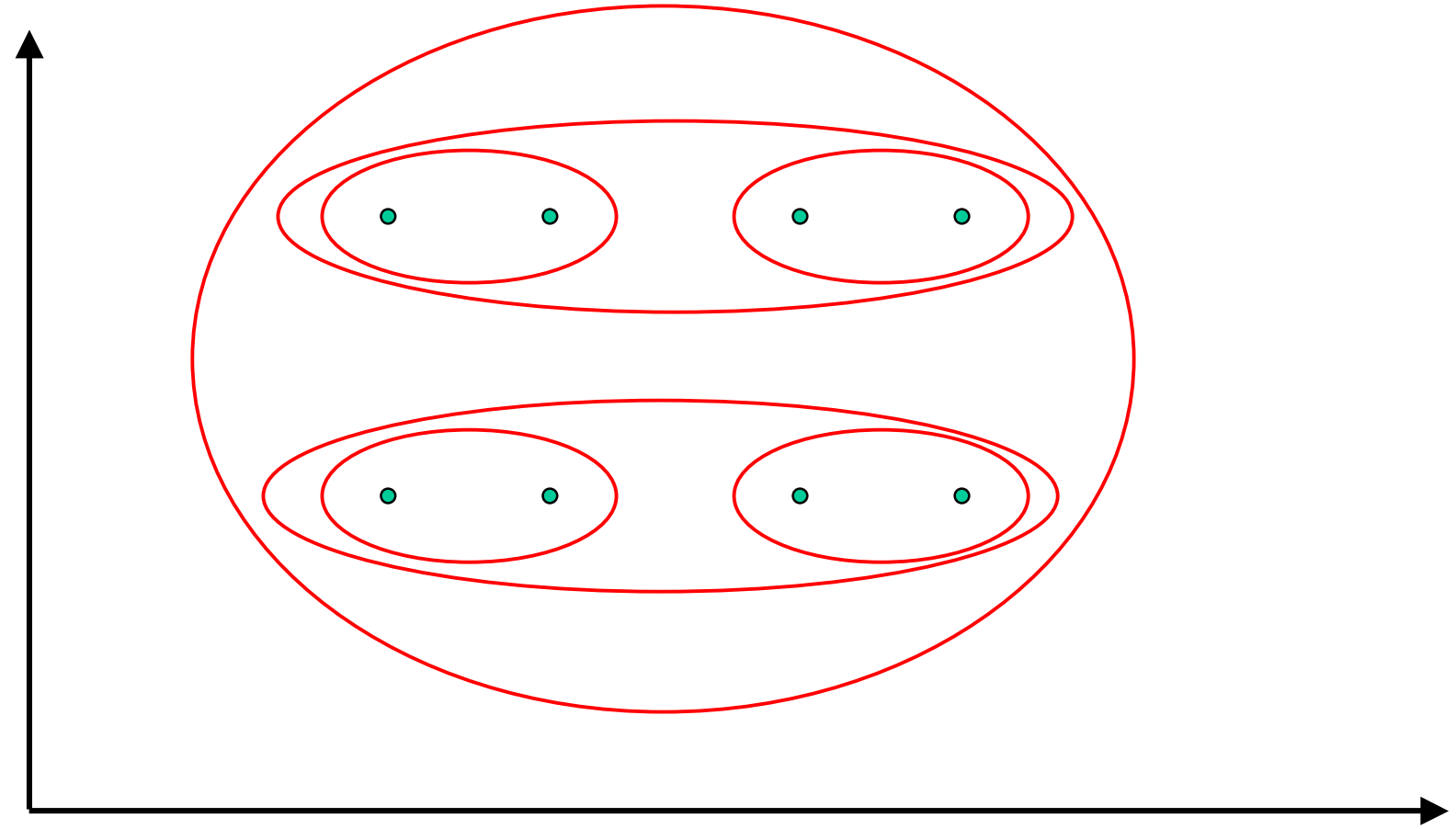
# Single Link Agglomerative Clustering

- Utilizziamo la similarità massima tra coppie di istanze:

$$sim(c_i, c_j) = \max_{\vec{x} \in c_i, \vec{y} \in c_j} sim(\vec{x}, \vec{y})$$

- A causa di un effetto concatenamento, può restituire cluster “lunghi e fini”
  - Adeguato in certi domini, come il raggruppamento di isole

# Esempio di Single Link



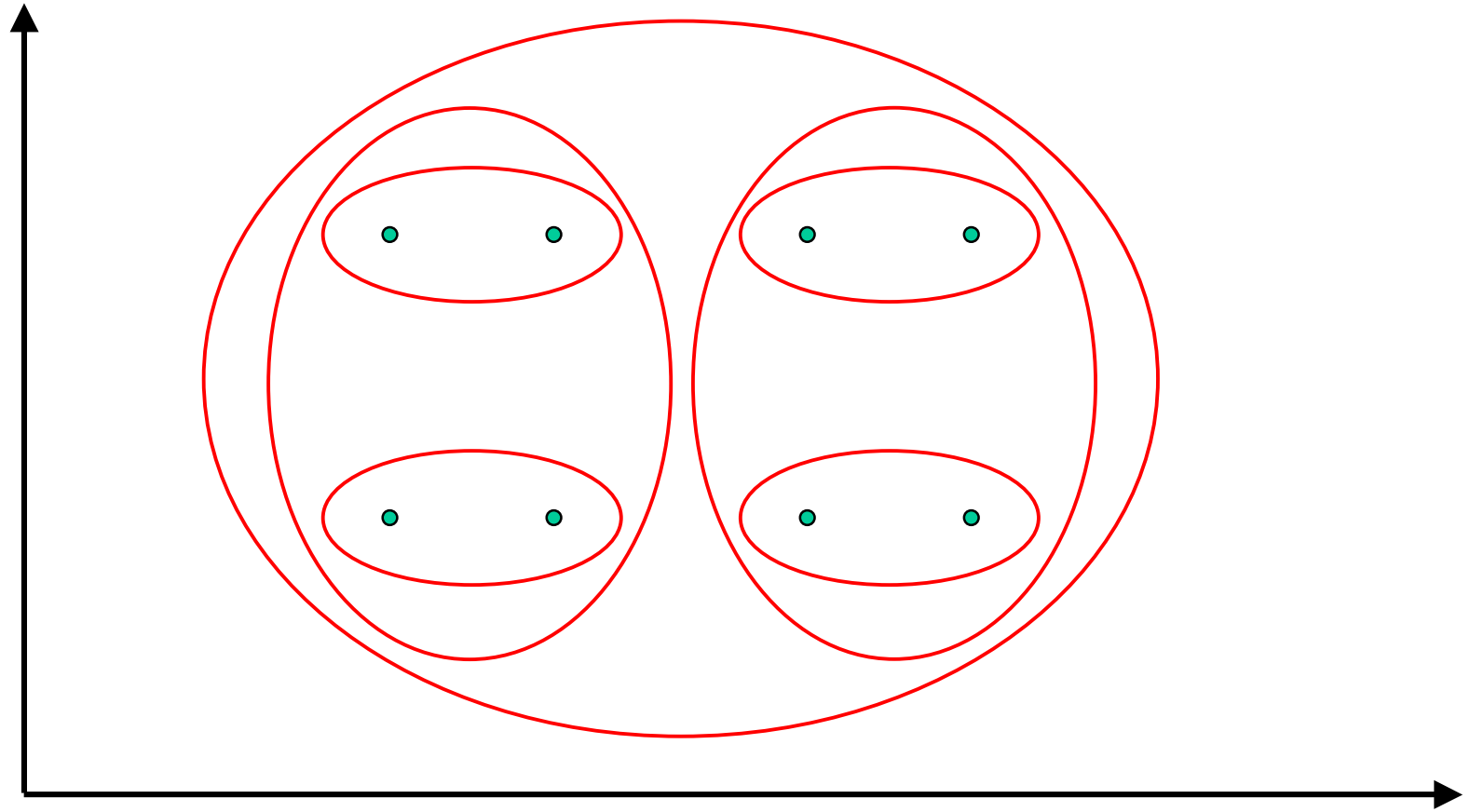
# Complete Link Agglomerative Clustering

- Basato sulla minima similarità tra coppie di istanze:

$$sim(c_i, c_j) = \min_{\vec{x} \in c_i, \vec{y} \in c_j} sim(\vec{x}, \vec{y})$$

- Crea cluster più sferici, normalmente preferibili

# Esempio di Complete Link



# Calcolare la Similarità tra Cluster

- Dopo aver fuso i cluster $c_i$ e $c_j$, la similarità del clustering ottenuto rispetto a un altro cluster arbitrario $c_k$ può essere calcolata come segue:
  - Single Link:

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

  - Complete Link:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$

# Group Average Agglomerative Clustering

- Per determinare la similarità tra $c_i$ e $c_j$ usa la similarità media su tutte le coppie nell'unione di $c_i$ e $c_j$.

$$sim(c_i, c_j) = \frac{1}{\left|c_i \cup c_j\right|\left(\left|c_i \cup c_j\right| - 1\right)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j): \vec{y} \neq \vec{x}} sim(\vec{x}, \vec{y})$$

- Compromesso tra single e complete link.
- Se si vogliono cluster più sferici e netti, si deve determinare la similarità media tra coppie ordinate di istanze nei due cluster (invece che tra coppie di istanze nell'unione)

# Clustering Partitivo

- Si deve fornire il numero desiderato di cluster *k*
- Si scelgono k istanze a caso, una per cluster, chiamate **semi (seeds)**
  - Si formano i *k* cluster iniziali sulla base dei semi
- Itera, riallocando tutte le istanze sui diversi cluster per migliorare il clustering complessivo
- Ci si ferma quando il clustering converge o dopo un numero prefissato di iterazioni

# k-Means

- Assume istanze a valori reali
- I cluster sono basati su **centroidi** o media dei punti in un cluster $c$:

$$\mu(\mathrm{c}) = \frac{1}{|c|}\sum_{\vec{x}\in c}\vec{x}$$

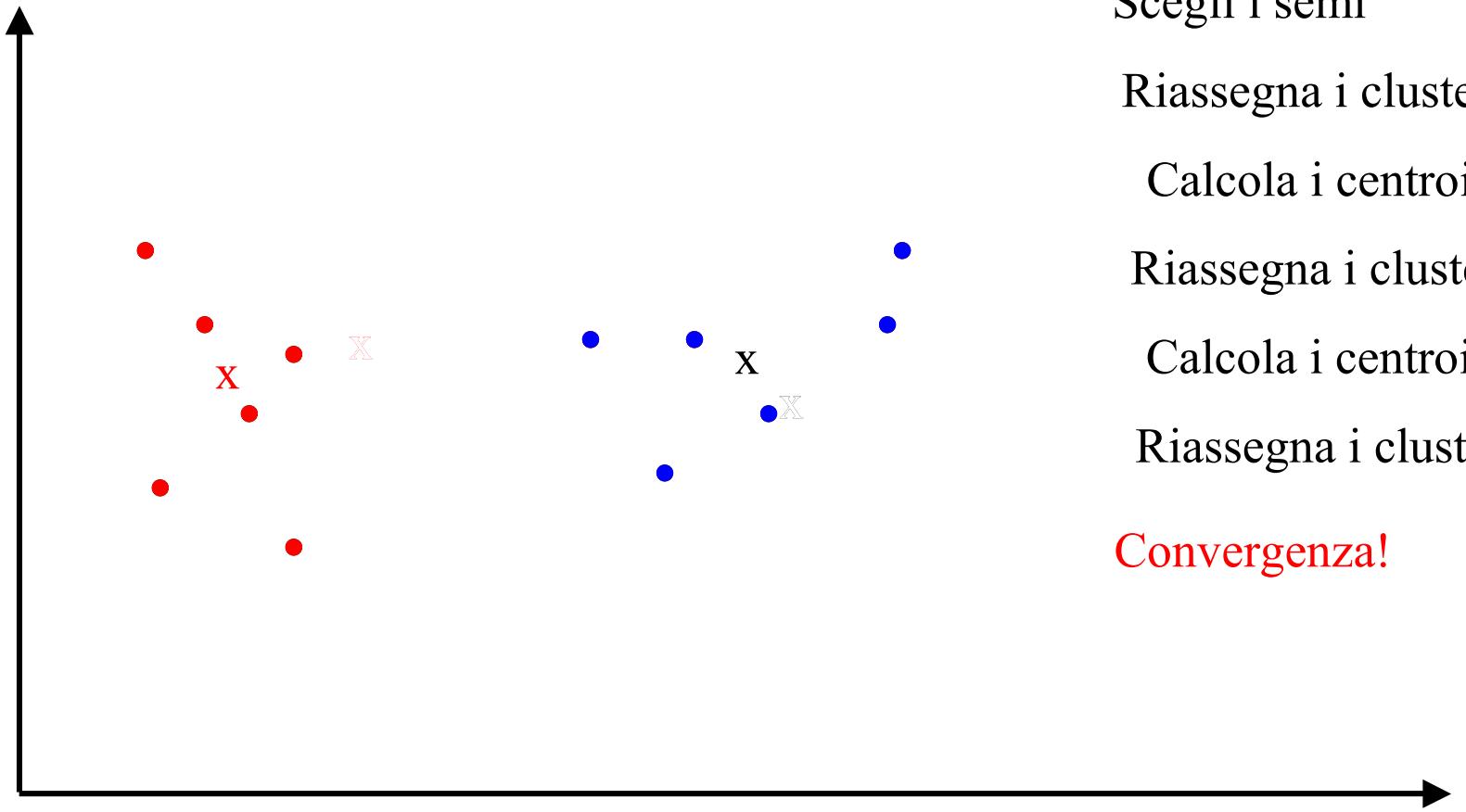
- Le istanze vengono riassegnate ai cluster sulla base della distanza rispetto ai centroidi dei cluster attuali

# Algoritmo k-means

**k-means**(distanza d, insieme delle istanze X)

**Seleziona** $k$ istanze a caso $\{s_1, s_2, \ldots, s_k\} \subseteq$ X come **semi**.
**Finché** clustering non converge o si raggiunge criterio di stop:
    Per ogni istanza $x \in$ X:
        **Assegna** $x$ al cluster $c_j$ tale che $d(x, s_j)$ è minimale
    **Aggiorna** i semi al centroide di ogni cluster, ovvero
    per ogni cluster $c_j$:
        $s_j = \mu(c_j)$

# k-Means: Esempio (k=2)



Scegli i semi

Riassegna i cluster

Calcola i centroidi

Riassegna i cluster

Calcola i centroidi

Riassegna i cluster

Convergenza!

# Obiettivo di k-Means

- L'obiettivo di k-means è di **minimizzare la somma del quadrato della distanza** di ciascun punto in X **rispetto al centroide del cluster** cui è assegnato:

$$\sum_{i=1}^{k} \sum_{x \in c_i} d(\vec{x}, \mu_i)^2$$

- Così come per gli algoritmi genetici, trovare il minimo globale è un problema NP-hard
- E' garantito che l'algoritmo k-means converga a un minimo locale

# Scelta dei Semi

- I risultati possono variare notevolmente sulla base della selezione dei semi
- Alcuni semi possono portare a un basso tasso di convergenza o a convergere su clustering sub-ottimali
- Si possono selezionare buoni semi usando euristiche o come risultato di un altro metodo

# Text Clustering

- I metodi di clustering possono essere applicati a documenti di testo in modo semplice
- Tipicamente, si rappresenta un documento mediante **vettori TF*IDF (term frequency*inverse document frequency) normalizzati** e si utilizza la **similarità del coseno**
- Applicazioni:
  - Durante la fase di recupero dei documenti di un sistema di Information Retrieval (IR), si possono fornire documenti nello stesso cluster di quello inizialmente recuperato per aumentare la recall del sistema
  - I risultati di un sistema di IR possono essere presentati per gruppi
  - Produzione automatizzata di tassonomie gerarchiche di documenti per scopi di nagiazione (stile Yahoo & DMOZ).

# Hard vs. Soft Clustering

- Tipicamente il clustering assume che ogni istanza sia assegnata a un solo cluster
  - Questo non permette di esprimere l'incertezza riguardo l'appartenenza di un'istanza a più cluster
- Il **soft clustering** fornisce una distribuzione di probabilità per ogni istanza rispetto all'appartenenza a ciascun cluster
  - Le probabilità di appartenenza di ogni istanza su tutti i cluster devono sommare a 1

# Fuzzy Clustering

- Approccio di soft clustering
- Si definisce un grado di appartenenza $u_i(x)$ dell'istanza $x$ all'$i$-esimo cluster, tale che:

$$\sum_{i=1}^{k} u_i(\vec{x}) = 1$$

  - dove k è il numero di cluster
- In **fuzzy k-means**, il centroide di un cluster $c_i$ è calcolato come la media di tutti i punti, ciascuno pesato rispetto al suo grado di appartenenza a $c_i$

$$\vec{\mu}(c_i) = \frac{\sum_{\vec{x} \in X} u_i(\vec{x})^m \vec{x}}{\sum_{\vec{x} \in X} u_i(\vec{x})^m}, \quad m > 1$$

  - m è **l'esponente di fuzzyness** (es. m=2)

# Fuzzy k-means

- Definiamo il grado di appartenenza di $x$ a $c_i$:

$$u_i(\vec{x}) = \frac{1}{\sum_{j=1}^{k} \left( \frac{d(\vec{x}, \vec{c_i})}{d(\vec{x}, \vec{c_j})} \right)^{\frac{2}{m-1}}}$$

- L'algoritmo è simile a k-means:

**Fuzzy k-means**(distanza d, insieme delle istanze X)

**Inizializza** i coefficienti $u_i(x)$ per ogni $i \in \{1, \ldots, k\}$ e $x \in X$
(casualmente o sulla base di un'applicazione di k-means)
**Finché** non si raggiunge criterio di stop:
**Calcola** i centroidi per ogni cluster $c_i$
**Aggiorna** i coefficienti $u_i(x)$ di $x$ di essere nel cluster $c_i$

# Problemi nell'Apprendimento Non Supervisionato

- Come valutare il clustering?
  - Valutazione interna:
    - Separazione netta dei cluster (ad es., l'obiettivo di k-means)
    - Corrispondenza con un modello probabilistico dei dati
  - Valutazione esterna
    - Confronta i cluster con etichette di classe note su dati di benchmark
- Clustering sovrapponibili
- Metodi semi-supervisionati
  - Si utilizzano pochi esempi annotati a mano e moltissimi esempi non annotati