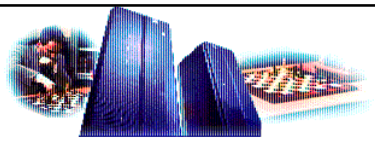
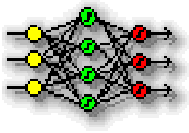




# Sistemi Intelligenti
# Learning and Clustering

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Informatica
alberto.borghese@unimi.it



## Riassunto

- I tipi di apprendimento
- Il clustering e le feature
- Clustering gerarchico
- Clustering partitivo: K-means

# I vari tipi di apprendimento

$x(t+1) = f[x(t), a(t)]$ Ambiente
$a(t) = \mathbf{g}[x(t)]$ Agente

**Supervisionato** (learning with a teacher). Viene specificato per ogni pattern di input, il pattern desiderato in output.

**Semi-Supervisionato**. Viene specificato solamente per **alcuni** pattern di input, il pattern desiderato in output.

**Non-supervisionato** (learning without a teacher). Estrazione di similitudine statistiche tra pattern di input. Clustering. Mappe neurali.

**Apprendimento con rinforzo** (reinforcement learning, learning with a critic). L'ambiente fornisce un'informazione puntuale, di tipo qualitativo, ad esempio success or fail.



# I gruppi di algoritmi

Clustering (data mining)

Classification

Predictive regression

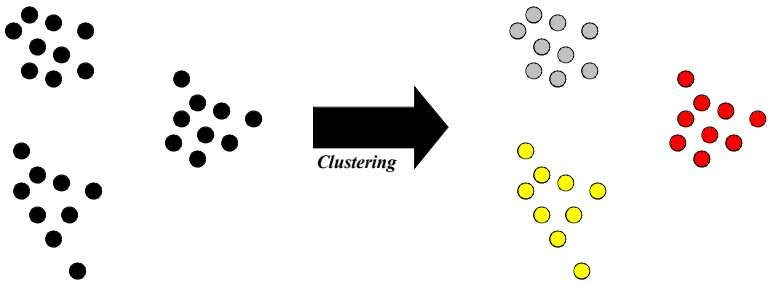
# Riassunto

- I tipi di apprendimento
- Il clustering e le feature
- Clustering gerarchico
- Clustering partitivo: K-means



# Clustering

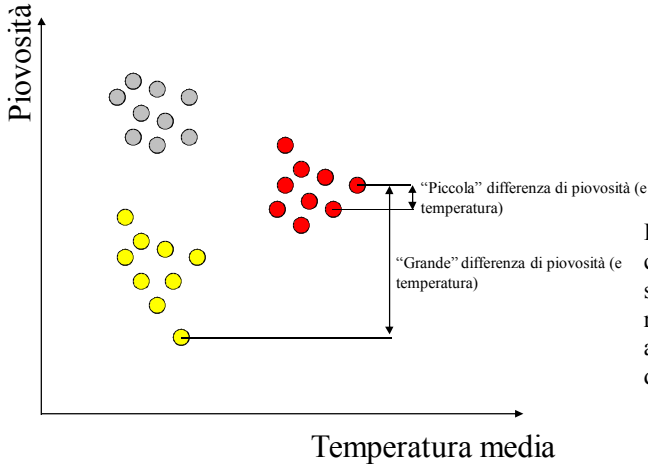
- Clustering: raggruppamento degli "oggetti" in cluster omogenee tra loro. Gli oggetti di un cluster sono più "simili" tra loro che a quelli degli altri cluster.
  - Raggruppamento per colore
  - Raggruppamento per forme
  - Raggruppamento per tipi
  - …..




Novel name: **data mining**

# Clustering

L'elaborazione verrà poi effettuata sui prototipi che rappresentano ciascun cluster.




I pattern appartenenti ad un cluster valido sono più simili l'uno con l'altro rispetto ai pattern appartenenti ad un cluster differente.



# Il clustering per…

… Confermare ipotesi sui dati (es. "E' possibile identificare tre diversi tipi di clima in Italia: mediterraneo, continentale, alpino…");

… Esplorare lo spazio dei dati (es. "Quanti tipi diversi di clima sono presenti in Italia? Quante sfere sono presenti in un'immagine?");

… Semplificare l'interpretazione dei dati ("Il clima di ogni città d'Italia è approssimativamente mediterraneo, continentale o alpino.").

…"Ragionare" sui dati o elaborare i dati in modo stereotipato.

# Esempio di clustering

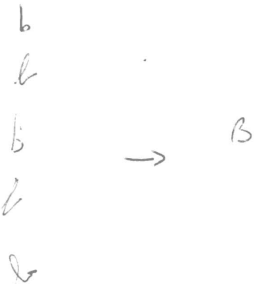
Clustering -> Indicizzazione

Ricerca immagini su WEB.



# Clustering: definizioni

- **Pattern**: un singolo dato $\mathbf{X} = [x_1, x_2, \ldots x_D]$. Il dato appartiene quindi ad uno spazio multi-dimensionale (D dimensionale), solitamente eterogeneo.
- **Feature**: le caratteristiche dei dati significative per il clustering, possono costituire anch'esso un vettore, il vettore delle feature: $f_1, f_2, \ldots f_M$. Questo vettore costituisce l'input agli algoritmi di clustering.



Inclinazione, occhielli, lunghezza, linee orizzontali, archi di cerchio ...

# Clustering: definizioni

- **D**: dimensione dello spazio dei pattern;
- **M**: dimensione dello spazio delle feature;
- **Cluster**: in generale, insieme che raggruppa dati simili tra loro, valutati in base alle feature;
- **Funzione di similarità o distanza**: una metrica (o quasi metrica) nello spazio delle feature, usata per quantificare la similarità tra due pattern.
- **Algoritmo**: scelta di come effettuare il clustering (motore di clustering).



# Clustering

- Dati, $\{X_1 \ldots X_N\} \in R^D$
- Cluster $\{C_1 \ldots C_M\} \rightarrow \{P_1 \ldots P_M\} \in R^D$

$\mathbf{P_j}$ is the prototype of cluster j and it represents the set of data inside its cluster.

To cluster the data:

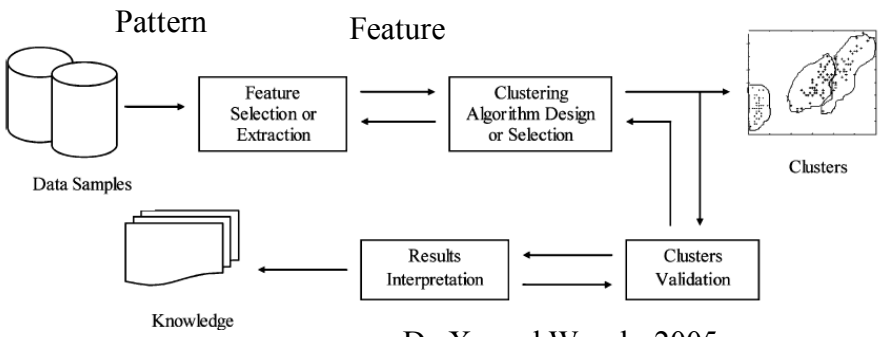
- The set of data inside each cluster has to be determined (the boundary of a cluster defined)
- The cluster boundaries are determined considering features associated to the data.

# Tassonomia (sintetica) degli algoritmi di clustering

- Algoritmi gerarchici (agglomerativi, divisivi), e.g. **Hirarchical clustering**.
- Algoritmi partizionali, hard: **K-means, quad-tree decomposition**.
- Algoritmi partizionali, **soft-clustering**: fuzzy c-mean, neural-gas, enhanced vector quantization, **mappe di Kohonen**.
- Algoritmi statistici: **mixture models**.



# Analisi mediante clustering




Da Xu and Wunsh, 2005

I cluster ottenuti sono significativi?
Il clustering ha operato con successo?

NB i cammini all'indietro consentono di fare la sintonizzazione dei diversi passi.

# Il clustering

Per una buona review: Xu and Wunsch, IEEE Transactions on Neural Networks, vol. 16, no. 3, 2005.

Il clustering non è di per sé un problema ben posto. Ci sono diversi gradi di libertà da fissare su come effettuare un clustering.

Rappresentazione dei pattern;

Calcolo delle feature;

Definizione di una misura di prossimità dei pattern attraverso le feature;

Tipo di algoritmo di clustering (gerachico o partizionale)

Validazione dell'output (se necessario) -> Testing.

Problema a cui non risponderemo: **quanti cluster**? Soluzione teorica (criterio di Akaike), soluzione empirica (growing networks di Fritzke).



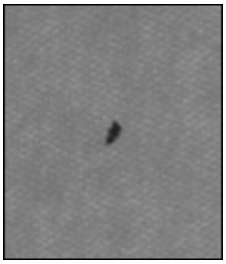
# Features

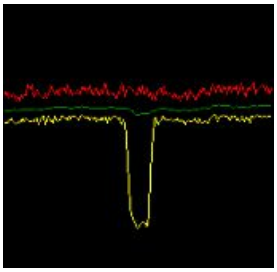
- Globali: livello di luminosità medio, varianza, contenuto in frequenza.....
- Feature locali

# Features

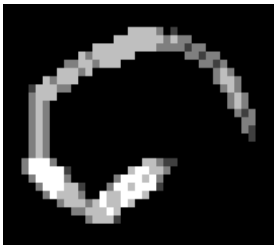
- *Località.*
- *Significatività.*
- *Rinoscibilità.*

Macchie dense





Fili





# Rappresentazione dei dati

- La similarità tra dati viene valutata attraverso le feature.
- Feature selection: identificazione delle feature più significative per la descrizione dei pattern.

  Esempio: descrizione del clima e della città di Roma.
  Roma è caratterizzata da: [17°; 500mm; 1.500.000 ab., 300 chiese]
- Quali feature scegliere?
- Come valutare le feature?
  - Analisi statistica del potere discriminante: correlazione tra feature e loro significatività.

# Similarità tra feature

- Definizione di una **misura di distanza tra due features**;

  Esempio:
  Distanza euclidea…
  dist (Roma, Milano) = dist ([17°; 500mm], [13°; 900mm]) = …

= … Distanza euclidea? = $((17-13)^2+(500-900)^2)^{1/2} = 400.02 \sim 400$

Ha senso?



# Normalizzazione feature

**E' necessario trovare una metrica corretta per la rappresentazione dei dati. Ad esempio, normalizzare le feature!**

$T_{Max}$= 20° $T_{Min}$= 5° → $T_{Norm} = (T - T_{Min})/(T_{Max} - T_{Min})$

$P_{Max}$= 1000mm $P_{Min}$= 0mm → $P_{Norm} = (P - P_{Min})/(P_{Max} - P_{Min})$

$Roma_{Norm}$ = [0.8 0.5]
$Milano_{Norm}$ = [0.53 0.9]

$dist(Roma_{Norm}, Milano_{Norm}) = ((0.8-0.53)^2+(0.5-0.9)^2)^{1/2} = 0.4826$

E' una buona scelta?

# Altre funzioni di distanza

- Mahalanobis:
  $dist(x,y)= (x_k-y_k)S^{-1}(x_k-y_k)$, con S matrice di covarianza.
  (Normalizzazione mediante covarianza)

Altre metriche:

- Distanza euclidea:
  $dist(x,y)=[\Sigma_{k=1..d}(x_k-y_k)^2]^{1/2}$
- Minkowski:
  $dist(x,y)=[\Sigma_{k=1..d}(x_k-y_k)^p]^{1/p}$
- Context dependent:
  $dist(x,y)= f(x, y, context)$

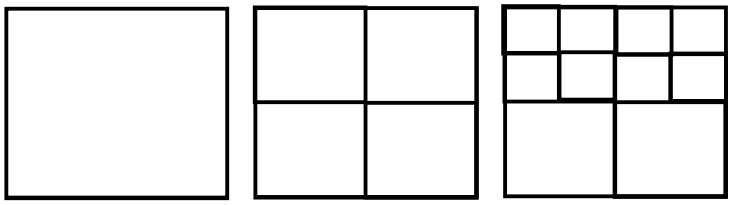


# Riassunto

- I tipi di apprendimento
- Il clustering e le feature
- Clustering gerarchico
- Clustering partitivo: K-means

# Algoritmi gerarchici divisivi: QTD

- Quad Tree Decomposition;
- Suddivisione gerarchica dello spazio delle feature, mediante splitting dei cluster;
- Criterio di splitting (~distanza tra cluster).





# Algoritmi gerarchici: QTD

- Clusterizzazione immagini RGB, 512x512;
- Pattern: pixel (x,y);
- Feature: canali R, G, B.
- Distanza tra due pattern (non euclidea):
  dist (p1, p2) =
  dist ([R1 G1 B1], [R2 G2 B2]) =
  max (|R1-R2|, |G1-G2|, |B1-B2|).

# Algoritmi gerarchici: QTD

p1 = [0 100 250]
p2 = [50 100 200]
p3 = [255 150 50]

dist (p1, p2) = dist ([R1 G1 B1], [R2 G2 B2]) =
max (|R1-R2|, |G1-G2|, |B1-B2|) = max([50 0 50]) = 50.

dist (p2, p3) = 205.

dist (p3, p1) = 255.



# Algoritmi gerarchici: QTD

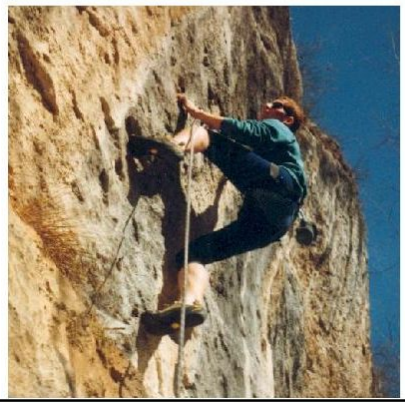
Criterio di splitting: se due pixel all'interno dello stesso cluster distano più di una determinata soglia, il cluster viene diviso in 4 cluster.

Esempio applicazione: segmentazione immagini, compressione immagini, analisi locale frequenze immagini…

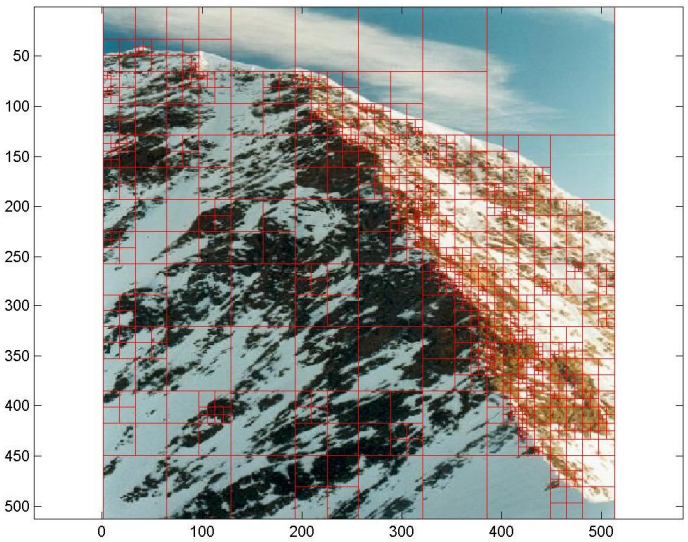
# QTD: Risultati

Original

Clusterized



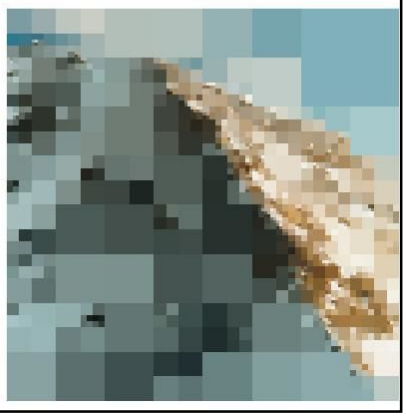
# QTD: Risultati



# QTD: Risultati

Original

Clusterized



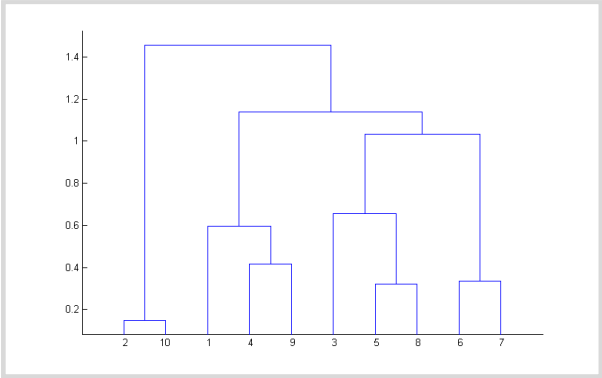
# Hierachical Clustering

- In brief, HC algorithms build a whole hierarchy of clustering solutions
  - Solution at level k is a *refinement* of solution at level k-1
- Two main classes of HC approaches:
  - Agglomerative: solution at level k is obtained from solution at level k-1 by merging two clusters
  - Divisive: solution at level k is obtained from solution at level k-1 by splitting a cluster into two parts
    - Less used because of computational load

# The 3 steps of agglomerative clustering

1. At start, each input pattern is assigned to a singleton cluster
2. At each step, the two *closest* clusters are merged into one
   - So the number of clusters is decreased by one at each step
3. At the last step, only one cluster is obtained

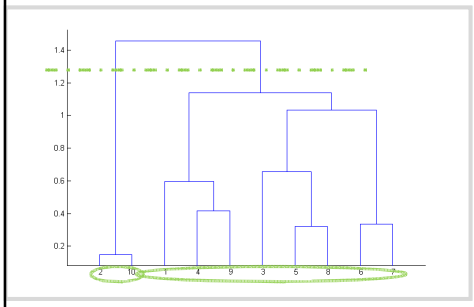
The clustering process is represented by a *dendrogram*

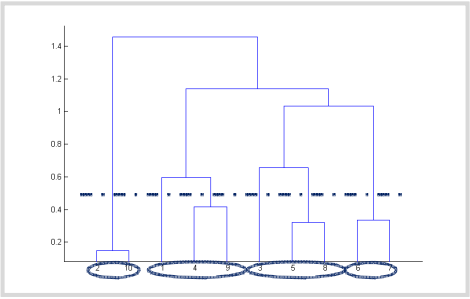





# How to obtain the final solution

- The resulting dendrogram has to be cut at some level to get the final clustering:
  - Cut criterion: number of desired clusters, or threshold on some features of resulting clusters





# Point-wise dissimilarity

- Different distances/indices of dissimilarity (***point wise***) …
  - E.g. euclidean, city-block, correlation…
- … and agglomeration criteria: Merge clusters $C_i$ and $C_j$ such that *diss(i, j)* is minimum (***cluster wise***)
  - Single linkage:
    - diss(i,j) = min d(x, y), where x is in $C_i$ , y in cluster $C_j$
  - Complete linkage:
    - diss(i,j) = max d(x, y), where x is in cluster i, y in cluster j
  - Group Average (GA) and Weighted Average (WA) Linkage:
    - diss(i j) = $\sum_{x\in C_i}\sum_{y\in C_j} w_i w_j d(x,y) \Big/ \sum_{x\in C_i}\sum_{y\in C_j} w_i w_j$

GA: $w_i = w_j = 1$
WA: $w_i = n_i$, $w_j = n_j$



# Cluster wise dissimilarity

- Other agglomeration criteria: Merge clusters $C_i$ and $C_j$ such that *diss(i, j)* is minimum
  - Centroid Linkage:
    - diss(i, j) = d($\mu_i$, $\mu_j$)
  - Median Linkage:
    - diss(i,j) = d($center_i$, $center_j$), where each $center_i$ is the average of the centers of the clusters composing $C_i$
  - Ward's: Method:
    - diss(i, j) = increase in the total error sum of squares (ESS) due to the merging of $C_i$ and $C_j$
- Single, complete, and average linkage: *graph methods*
  - *All points in clusters are considered*
- Centroid, median, and Ward's linkage: *geometric methods*
  - *Clusters are summed up by their centers*

# Ward's method

It is also known as minimum variance method.

Each merging step minimizes the increase in the total ESS:

$$ESS_i = \sum_{x \in C_i} (x - \mu_i)^2 \qquad ESS = \sum_i ESS_i$$

When merging clusters $C_i$ and $C_j$, the increase in the total ESS is:

$$\Delta ESS = ESS_{i,j} - ESS_i - ESS_j$$

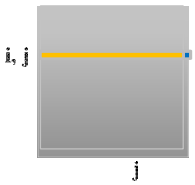
Spherical, compact clusters are obtained.

The solution at each level k is an approximation to the optimal solution for that level (the one minimizing ESS)



# How HC operates

- HC algorithms operate on a dissimilarity matrix:
  - For each pair of existant clusters, their dissimilarity value is stored
- When clusters $C_i$ and $C_j$ are merged, only dissimilarities for the new resulting cluster have to be computed
  - The rest of the matrix is left untouched



# The Lance-William recursive formulation

Used for iterative implementation. The dissimilarity value between newly formed cluster $\{C_i, C_j\}$ and every other cluster $C_k$ is computed as:

$$diss(k,(i,j)) = \alpha_i diss(k,i) + \alpha_j diss(k,j) + \beta diss(i,j) + \\ + \gamma |diss(k,i) - diss(k,j)|$$

Only values already stored in the dissimilarity matrix are used. Different sets of coefficients correspond to different criteria.

| Criterion | $\alpha_i$ | $\alpha_j$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|
| Single Link. | ½ | ½ | 0 | -½ |
| Complete Link. | ½ | ½ | 0 | ½ |
| Group Avg. | $n_i/(n_i+n_j)$ | $n_j/(n_i+n_j)$ | 0 | 0 |
| Weighted Avg. | ½ | ½ | 0 | 0 |
| Centroid | $n_i/(n_i+n_j)$ | $n_j/(n_i+n_j)$ | $-n_in_j/(n_i+n_j)^2$ | 0 |
| Median | ½ | ½ | - ¼ | 0 |
| Ward | $(n_i+n_k)/(n_i+n_j+n_k)$ | $(n_j+n_k)/(n_i+n_j+n_k)$ | $-n_k/(n_i+n_j+n_k)$ | 0 |



# Characteristics of HC

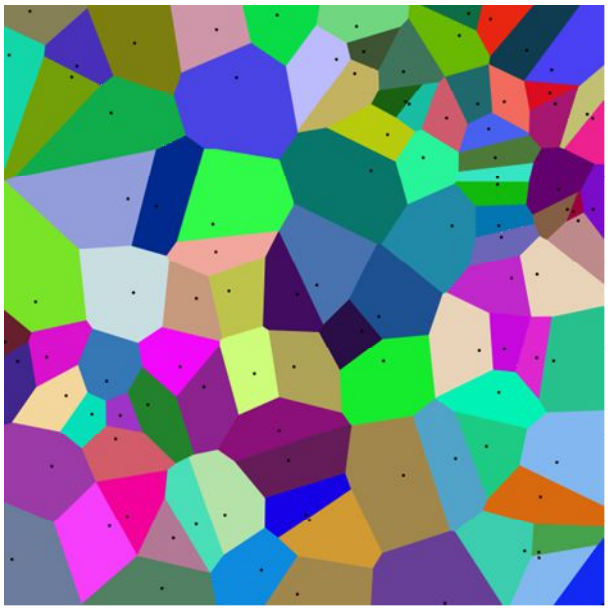
- Pros:
  - Indipendence from initialization
  - No need to specify a desired number of clusters from the beginning
- Cons:
  - Computational complexity at least $O(N^2)$
  - Sensitivity to outliers
  - No reconsideration of possibly misclassified points
  - Possibility of inversion phenomena and multiple solutions

# Riassunto

- I tipi di apprendimento
- Il clustering e le feature
- Clustering gerarchico
- Clustering partitivo: K-means



Risultato del clustering è un diagramma di Voronoj



I poligoni azzurri rappresentano i diversi cluster ottenuti. Ogni punto marcato all'interno del cluster (cluster center) è rappresentativo di tutti i punti del cluster

# K-means (partitional): framework

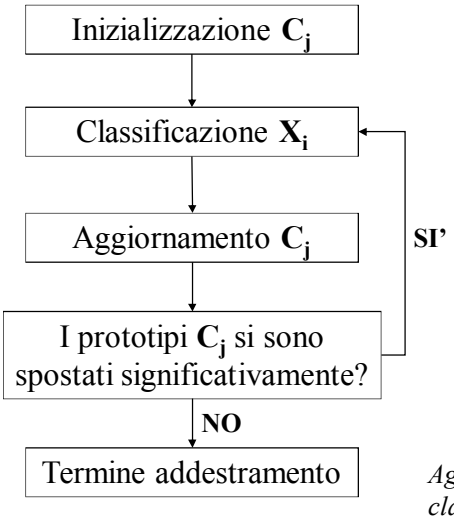
- Siano $\mathbf{X_1}$, …, $\mathbf{X_D}$ i dati di addestramento, features (per semplicità, definiti in $R^2$);
- Siano $\mathbf{C_1}$, …, $\mathbf{C_K}$ i *prototipi* di K cluster, definiti anch'essi in $R^2$; ogni *prototipo* identifica il baricentro del cluster corrispondente;
- Lo schema di classificazione adottato sia il seguente: "$\mathbf{X_i}$ appartiene a $\mathbf{C_j}$ se e solo se $\mathbf{C_j}$ è il *prototipo* più vicino a $\mathbf{X_i}$ (distanza euclidea)**";**
- L'algoritmo di addestramento permette di determinare le posizioni dei *prototipi* $\mathbf{C_j}$ mediante successive approssimazioni.

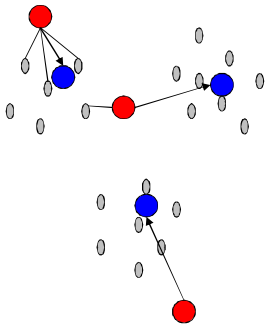


# Algoritmo K-means

L'obiettivo che l'algoritmo si prepone è di minimizzare la varianza totale intra-cluster. Ogni cluster viene identificato mediante un centroide o punto medio. L'algoritmo segue una procedura iterativa. Inizialmente crea K partizioni e assegna ad ogni partizione i punti d'ingresso o casualmente o usando alcune informazioni euristiche. Quindi calcola il centroide di ogni gruppo. Costruisce quindi una nuova partizione associando ogni punto d'ingresso al cluster il cui centroide è più vicino ad esso. Quindi vengono ricalcolati i centroidi per i nuovi cluster e così via, finché l'algoritmo non converge (Wikipedia).

# K-means: addestramento



*Aggiornamento* $\boldsymbol{C_j}$*: baricentro degli* $\boldsymbol{X_i}$ *classificati da* $\boldsymbol{C_j}$.



# Algoritmo K-means::formalizzazione

- Dati N pattern in ingresso $\{x_j\}$ e $C_k$ prototipi che vogliamo diventino i centri dei cluster, $x_j$ e $C_k \in R^N$. Ciascun cluster identifica una regione nello spazio, $P_k$.
- Valgono le seguenti proprietà:

$$\bigcup_{k=1}^{K} P_k = Q \supseteq R^D$$ I cluster coprono lo spazio delle feature

$$\bigcap_{k=1}^{K} P_k = \varnothing$$ I cluster sono disgiunti.

- $x_j \in C_k$ Se: $\left(\left|x_j - C_k\right|\right)^2 \leq \left(\left|x_j - C_l\right|\right)^2 \quad l \neq k$

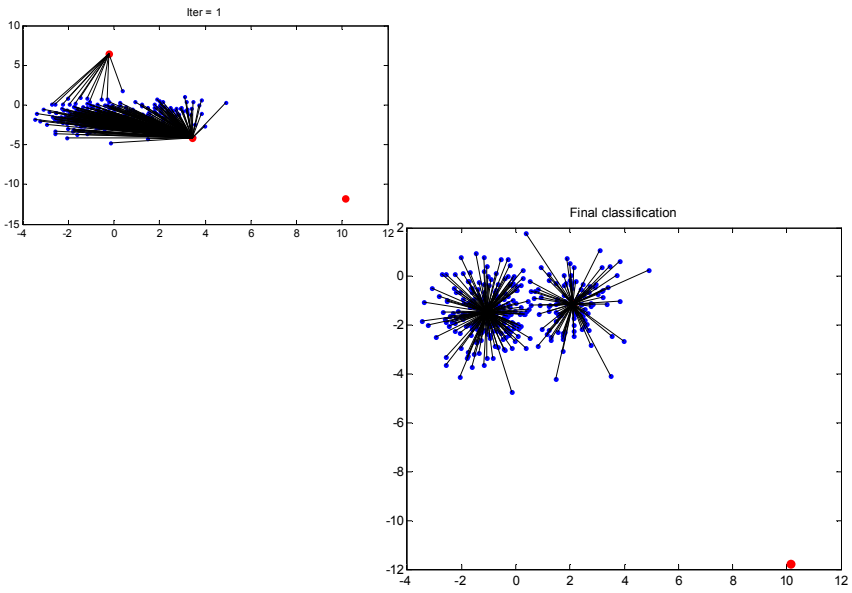
- La funzione obbiettivo viene definita come: $\sum_{i=1}^{K}\sum_{j=1}^{N}\left(\left|x_{j(k)} - C_k\right|\right)^2$

# Algoritmo K-means::dettaglio dei passi

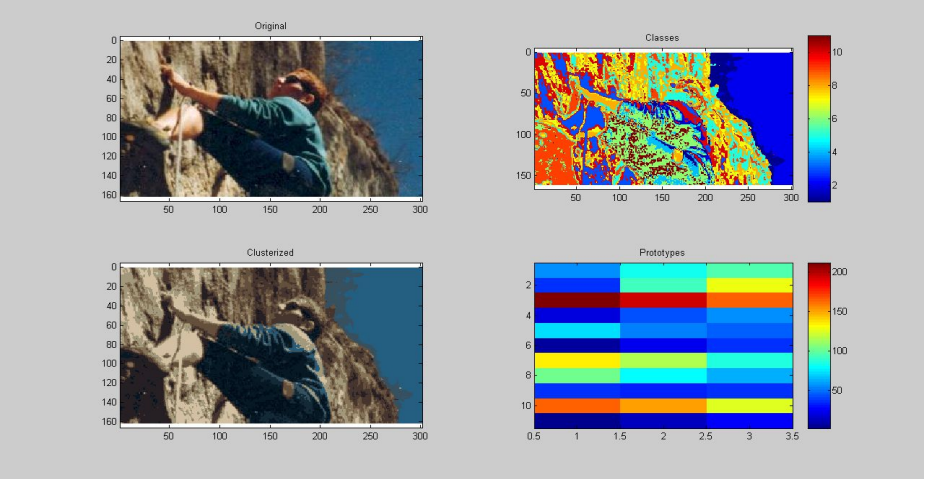
- Inizializzazione.
  - Posiziono in modo arbitrario o guidato i K centri dei cluster.
- Iterazioni
  - Assegno ciascun pattern al cluster il cui centro è più vicino, formando così un certo numero di cluster ($\leq$ K).
  - Calcolo la posizione dei cluster, $C_k$, come baricentro dei pattern assegnati ad ogni cluster, sposto quindi la posizione dei centri dei cluster.
- Condizione di uscita
  - I centri dei cluster non si spostano più.



# Bad initialization



# K-Means per immagine RGB




Da 255 colori a 33 colori



# Riassunto

- I tipi di apprendimento
- Il clustering e le feature
- Clustering gerarchico
- Clustering partitivo: K-means