

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



A.A. 2019-2020

1/58

<http://borghese.di.unimi.it>



Riassunto



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

A.A. 2019-2020

2/58

<http://borghese.di.unimi.it>



I vari tipi di apprendimento



$$\begin{array}{ll} x(t+1) = f[x(t), a(t)] & \text{Ambiente} \\ a(t) = \mathbf{g}[x(t)] & \text{Agente} \end{array}$$

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



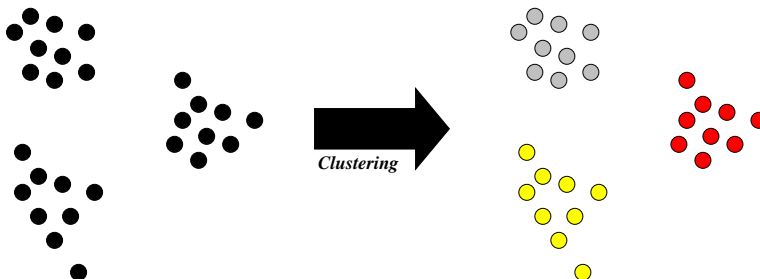
- I tipi di apprendimento
- **Il clustering e le feature**
- Clustering gerarchico
- Clustering partitivo: K-means
- Clustering partitivo: soft-clustering



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
 - ◆



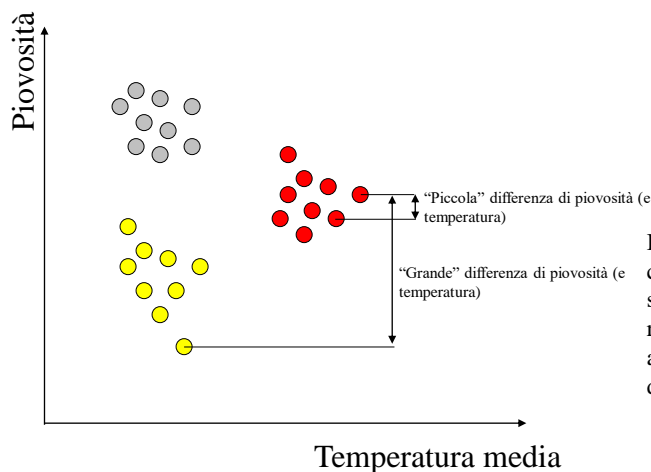
Novel name: **data mining**



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



Ricerca immagini su WEB.

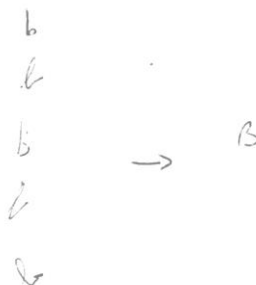


Clustering -> Indicizzazione



Clustering: definizioni

- **Pattern:** un singolo dato $\mathbf{X} = [x_1, x_2, \dots, x_D]$. 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, \dots, 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, $\{X_1 \dots X_N\} \in \mathbb{R}^D$
- Cluster $\{C_1 \dots C_M\} \rightarrow \{P_1 \dots P_M\} \in \mathbb{R}^D$

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.



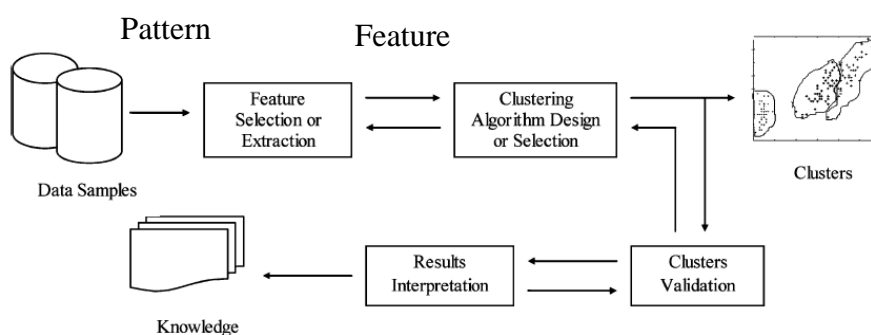
Tassonomia (sintetica) degli algoritmi di clustering



- Algoritmi gerarchici (agglomerativi, divisivi), e.g. **Hierarchical 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 Wunsch, 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 (gerarchico 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



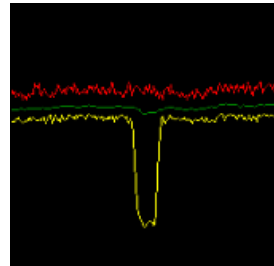
- Globali: livello di luminosità medio, varianza, contenuto in frequenza.....
- Feature locali



Features

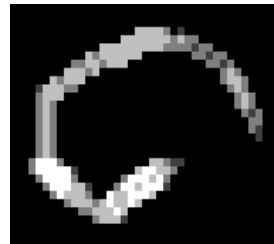


Macchie
dense



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

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...

$\text{dist}(\text{Roma}, \text{Milano}) = \text{dist}([17^\circ; 500\text{mm}], [13^\circ; 900\text{mm}]) = \dots$

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

Ha senso?



Normalizzazione feature



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

$$T_{\text{Max}} = 20^\circ \quad T_{\text{Min}} = 5^\circ \rightarrow T_{\text{Norm}} = (T - T_{\text{Min}}) / (T_{\text{Max}} - T_{\text{Min}})$$

$$P_{\text{Max}} = 1000\text{mm} \quad P_{\text{Min}} = 0\text{mm} \rightarrow P_{\text{Norm}} = (P - P_{\text{Min}}) / (P_{\text{Max}} - P_{\text{Min}})$$

$$\text{Roma}_{\text{Norm}} = [0.8 \ 0.5]$$

$$\text{Milano}_{\text{Norm}} = [0.53 \ 0.9]$$

$$\text{dist}(\text{Roma}_{\text{Norm}}, \text{Milano}_{\text{Norm}}) = ((0.8-0.53)^2 + (0.5-0.9)^2)^{1/2} = 0.4826$$

E' una buona scelta?



Altre funzioni di distanza



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

Altre metriche:

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



Riassunto



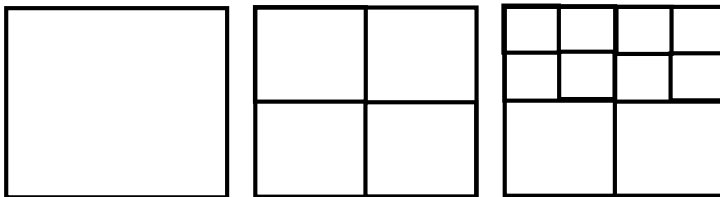
- I tipi di apprendimento
- Il clustering e le feature
- **Clustering gerarchico**
- Clustering partitivo: K-means
- Clustering partitivo: soft-clustering



Algoritmi gerarchici divisivi: QTD



- Quad Tree Decomposition;
- Suddivisione gerarchica dello spazio delle feature, mediante splitting dei cluster;
- Criterio di splitting (\sim 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):
 $\text{dist}(p1, p2) =$
 $\text{dist}([R1 \ G1 \ B1], [R2 \ G2 \ B2]) =$
 $\max(|R1-R2|, |G1-G2|, |B1-B2|).$



Algoritmi gerarchici: QTD



$$p1 = [0 \ 100 \ 250]$$

$$p2 = [50 \ 100 \ 200]$$

$$p3 = [255 \ 150 \ 50]$$

$$\begin{aligned} \text{dist}(p1, p2) &= \text{dist}([R1 \ G1 \ B1], [R2 \ G2 \ B2]) = \\ &= \max(|R1-R2|, |G1-G2|, |B1-B2|) = \max([50 \ 0 \ 50]) = 50. \end{aligned}$$

$$\text{dist}(p2, p3) = 205.$$

$$\text{dist}(p3, p1) = 255.$$



Algoritmi gerarchici: QTD



Critero 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

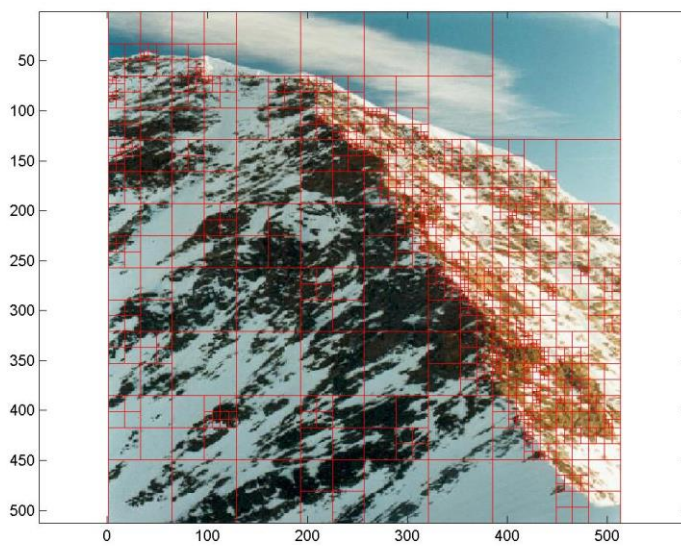


Original

Clusterized



QTD: Risultati





QTD: Risultati

Original

Clusterized



Hierarchical 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

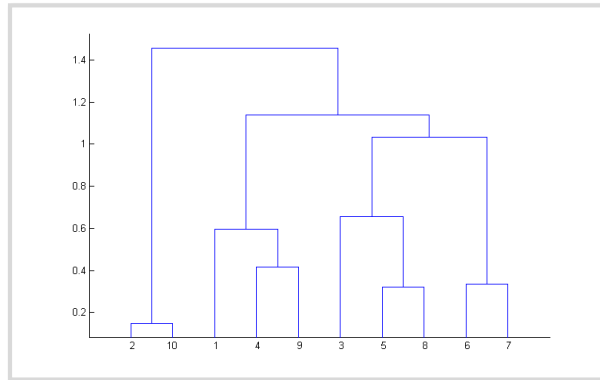


The 3 steps of agglomerative clustering



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

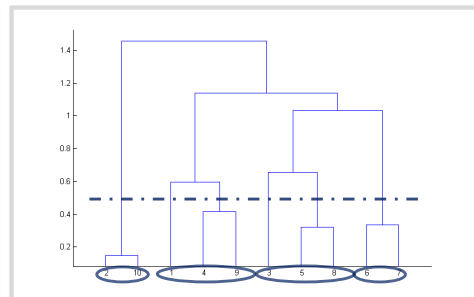
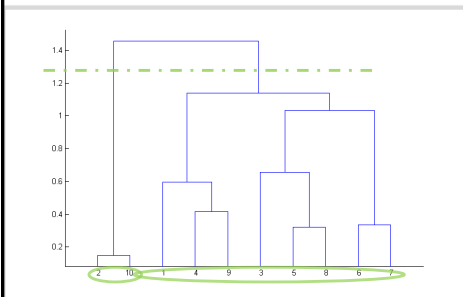
The clustering process is represented by a *dendrogram*



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:

- ⤷ $diss(i, j) = \frac{\sum_{x \in C_i} \sum_{y \in C_j} w_i w_j d(x, y)}{\sum_{x \in C_i} \sum_{y \in C_j} w_i w_j}$

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:
 - ⤷ $diss(i, j) = d(\text{center}_i, \text{center}_j)$, where each center_i is the average of the centers of the clusters composing C_i
 - ◆ Ward's: Method:
 - ⤷ $diss(i, j) = \text{increase in the total error sum of squares (ESS) due to the merging of } C_i \text{ and } C_j$
- 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:

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

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

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

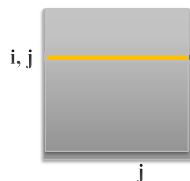
Spherical, compact clusters are obtained.

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	α_i	α_j	β	γ
Single Link.	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete Link.	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group Avg.	$n_i/(n_i+n_j)$	$n_j/(n_i+n_j)$	0	0
Weighted Avg.	$\frac{1}{2}$	$\frac{1}{2}$	0	0
Centroid	$n_i/(n_i+n_j)$	$n_j/(n_i+n_j)$	$-n_i n_j / (n_i + n_j)^2$	0
Median	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	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:
 - ◆ Independence 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
 - ◆ Ties can induce different clustering



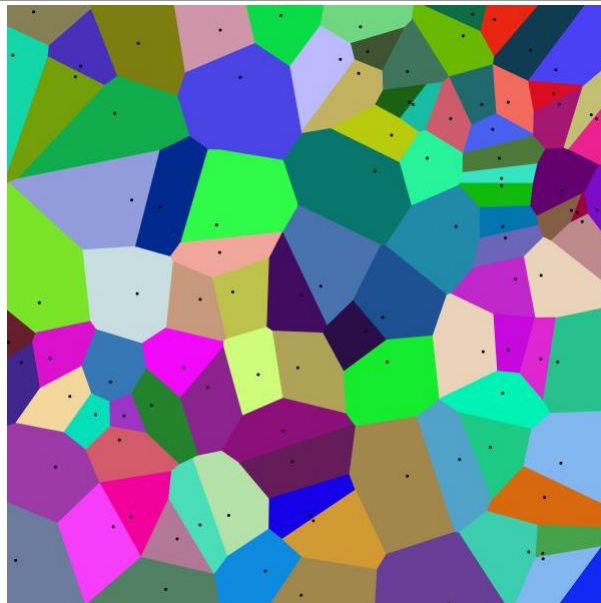
Riassunto



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



**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 $\mathbf{X}_1, \dots, \mathbf{X}_D$ i dati di addestramento, features (per semplicità, definiti in \mathbb{R}^2);
- Siano $\mathbf{C}_1, \dots, \mathbf{C}_K$ i *prototipi* di K cluster, definiti anch'essi in \mathbb{R}^2 ; ogni *prototipo* identifica il baricentro del cluster corrispondente;
- Lo schema di assegnamento 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.



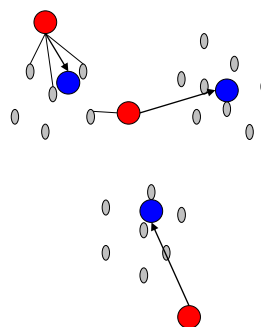
Algoritmo K-means



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 C_j : baricentro degli X_i classificati da 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 $C_k \in \mathbb{R}^N$. Ciascun cluster identifica una regione nello spazio, P_k .
- Valgono le seguenti proprietà:

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

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

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

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



Algoritmo K-means: dettaglio dei passi



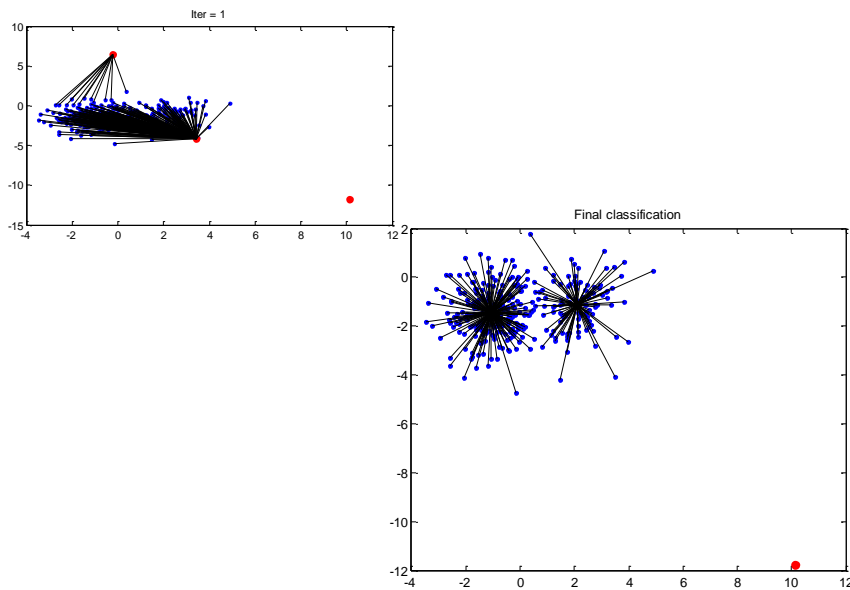
- Inizializzazione.
 - ◆ Posiziono in modo arbitrario o guidato i K centri dei cluster.

- 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, spostando quindi la posizione dei centri dei cluster.

- Condizione di uscita
 - I centri dei cluster non si spostano più.

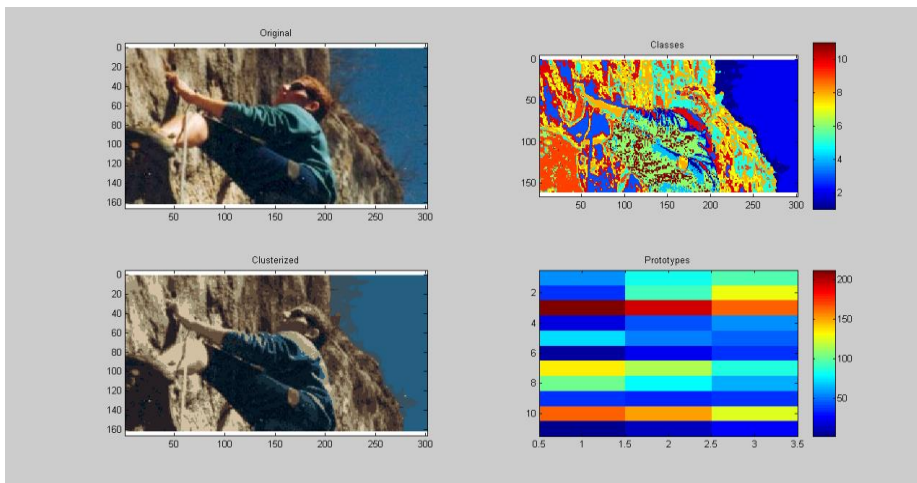


Bad initialization





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
- **Clustering partitivo: soft-clustering**



Principles of soft-clustering



- I centroidi vengono spostati e non posizionati
- Lo spostamento dei centroidi avviene analizzando iterativamente tutti i dati
- Lo spostamento viene ridotto via via che l'apprendimento procede



Competitive learning



- All'interazione k- esima, si presenti al sistema **un (1) dato**, \mathbf{X}_i ;
- **Aggiornamento di tutti i prototipi \mathbf{W}_j ("neuroni")**

- Generalized competitive Learning Rule:

$$\blacklozenge \Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$$

← AGGIORNAMENTO
PESI (POSIZIONE)
DEI NEURONI

$\Lambda_k(i,j)$ è una funzione "campo recettivo"

$$\blacklozenge \Lambda_k(i,j) = \exp(-\|\mathbf{X}_i - \mathbf{W}_j\|^2 / 2\sigma_k^2)$$

↻ (spazio dei dati)

$$\blacklozenge \Lambda_k(i,j) = \exp(-\|f(\mathbf{X}_i) - f(\mathbf{W}_j)\|^2 / 2\sigma_k^2)$$

↻ (spazio delle feature)

← FUNZIONI DI
VICINATO



Competitive learning



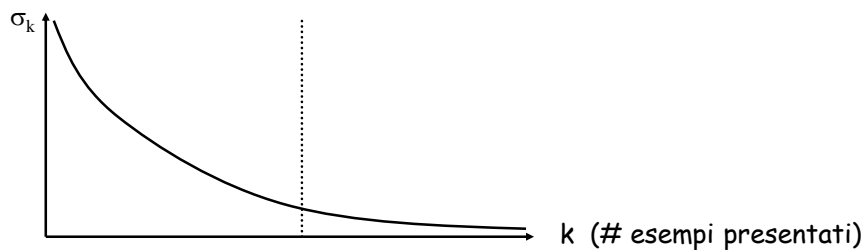
- All'interazione k- esima, si presenti alla rete **un (1)** dato \mathbf{X}_i ; al termine dell'apprendimento.
- Unità vincente (associazione):
 $j^* \text{ t.c. } \|\mathbf{W}_{j^*} - \mathbf{X}_i\| = \min_j \|\mathbf{W}_j - \mathbf{X}_i\|$

UNITA' VINCENTE

Anche qui viene indotta una tessellazione di Voronoj dallo spazio da tutte le unità vincenti.



Funzione di vicinato nel tempo

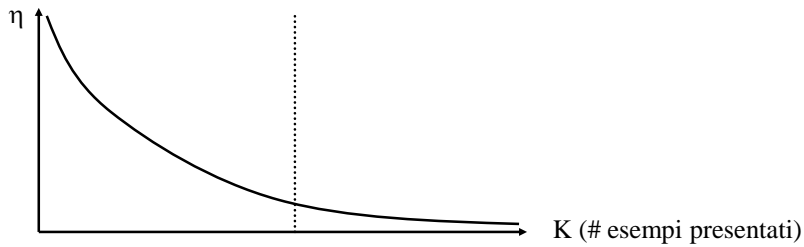


$$\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$$

Procedendo nell'addestramento della rete, un neurone perde la capacità di spostare i suoi vicini.



Learning rate nel tempo



$$\Delta W_j = \eta_k \Lambda_k(i,j) (X_i - W_j)$$

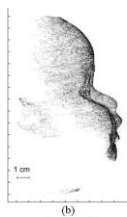
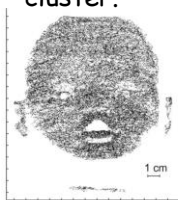
Procedendo nell'addestramento della rete, i pesi dei neuroni perdono la possibilità di muoversi => rete più stabile.



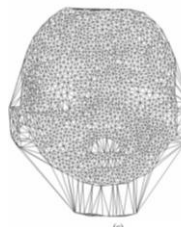
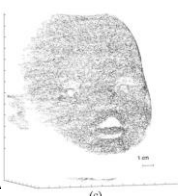
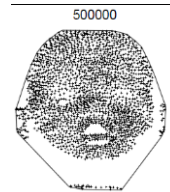
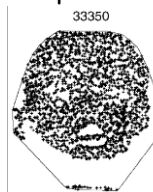
Competitive learning ("First search then converge")



- ORDERING PHASE:** η , σ grandi; ogni neurone può spostarsi molto verso l'ingresso X_i ; il neurone trascina con sé i vicini; in tale fase la rete si dispiega nello spazio R^N "spargendo" i suoi neuroni.
- TUNING PHASE:** η , σ piccoli; ogni neurone si muove da solo; è una fase di raffinamento in cui vengono raggiunti con precisione i centri dei cluster.



100,000 sampled points
reconstructed
with 2,000 RV





Soft-clustering

$$\Delta \mathbf{W}_j = \eta_k \Lambda_k(i,j) (\mathbf{X}_i - \mathbf{W}_j)$$

$\Lambda_k(i,j)$ è l'elemento chiave. I "Campi recettivi" dei diversi neuroni sono parzialmente sovrapposti.

In "Competitive clustering" $\Lambda_k(i,j)$ è una Gaussiana nello spazio dei dati $\underline{0}$ dei prototipi (mappe di Kohonen).

In "Neural-gas" $\Lambda_k(i,j)$ è una ranking function nello spazio dei dati e dei prototipi.

In "Fuzzy c-means" $\Lambda_k(i,j)$ è realizzata mediante membership function nello spazio dei dati.



I problemi del soft-clustering

Dead-units: sono centroidi che non vengono aggiornati da un certo passo, k , in poi.

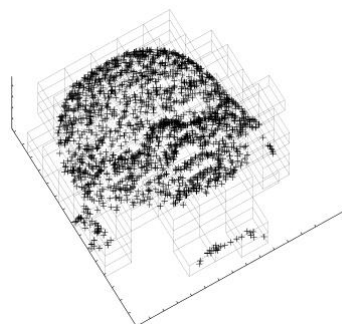
Inizializzazione guidata dai dati.

$$\rho_{Centroid} = \rho_{Data}^{\gamma} \quad \text{with} \quad \gamma = \frac{D}{D+2}$$

Partition of the input space and distribution of the number of centroids inside each box through a partitioning function:

$$M_k = M \frac{N_k^{\gamma}}{\sum_k N_k^{\gamma}}$$

Minimi locali.





Caratteristiche del soft-clustering



COMPETITIVE LEARNING. *Apprendimento competitivo. Dato un certo input, le unità **competono** tra loro per “aggiudicarsi” l’input.*

Questo meccanismo può essere hard. Nel caso estremo: “winner-take-all”, “spara” un solo neurone per volta (grandmother cell). Questo è l’approccio del K-means. Oppure può essere soft, le unità raggiungono un grado diverso di “vincita”.

Winner-take-all → hard approach

More than one winner → soft approach



Riassunto



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