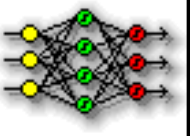


Sistemi Intelligenti
Clustering

Alberto Borghese

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

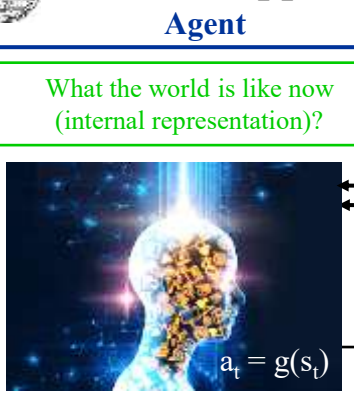
1

Riassunto

- Il clustering e le feature
- Clustering statistico
- Clustering gerarchico
- Clustering partitivo hard

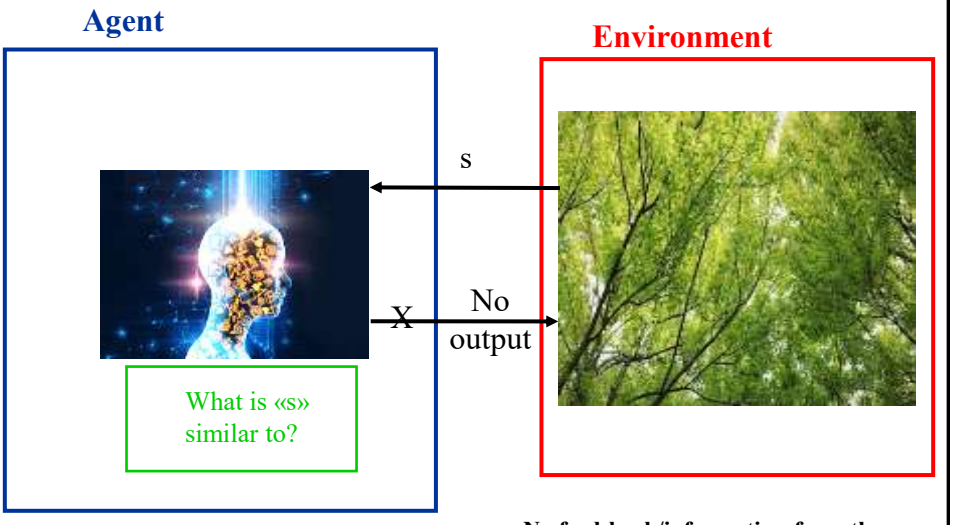
2

Apprendimento con rinforzo



L'ambiente fornisce un'informazione **puntuale**, di tipo **qualitativo**, ad esempio success or fail.

Apprendimento non supervisionato



No feed-back/information from the environment

Il clustering per…

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

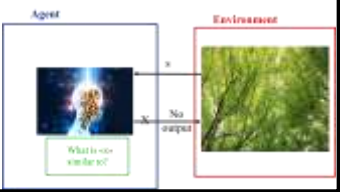
… Esplorare lo spazio dei dati (es. “Quanti tipi diversi di clima sono presenti in Italia? Quante sfere sono presenti in un’immagine?”);

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

…“Ragionare” sui dati mediante tecniche di Intelligenza Artificiale classica.

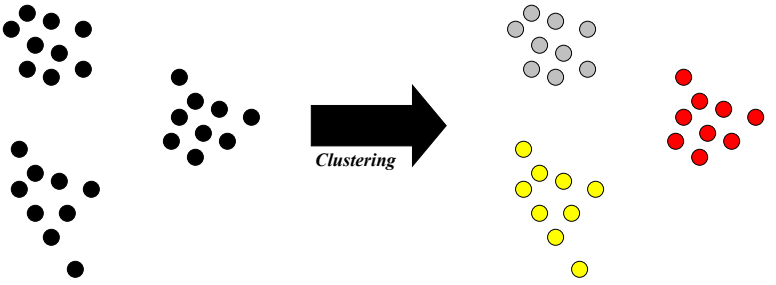
Il clustering è la funzionalità concettualmente più “semplice” di un agente, non richiede feed-back dall’ambiente, ma è in realtà molto “difficile”…

Raggruppamento di input simili.



Clustering

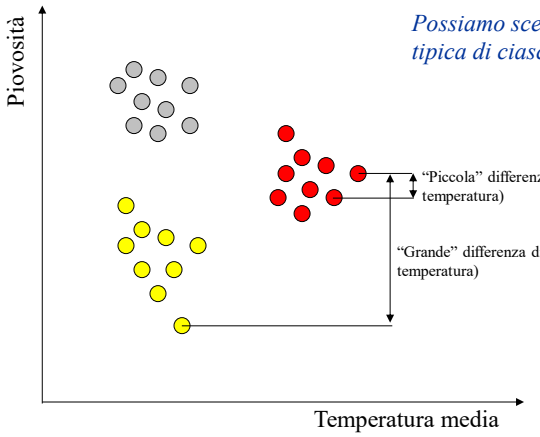
- Clustering: raggruppamento degli “oggetti” in **raggruppamenti omogenei tra loro**: **cluster**. Gli oggetti di un cluster sono più “simili” tra loro che a quelli degli altri cluster.
  - Raggruppamento per colore
  - Raggruppamento per forme
  - Raggruppamento per tipi
  - Raggruppamento per posizione
  - …..



Novel name: **data mining**

Prototipi e cluster

Ciascun cluster può essere rappresentato da un **prototipo** o dagli **elementi stessi** del cluster. L'elaborazione verrà poi effettuata sui **prototipi** che rappresentano ciascun cluster.



Possiamo scegliere una piovosità e temperatura tipica di ciascun raggruppamento

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

Esempio di clustering

Clustering -> Indicizzazione

E.g. immagini della regione di Bosa in Sardegna.

Sono immagini della stessa zona?

Ricerca immagini su WEB.

Clustering: features

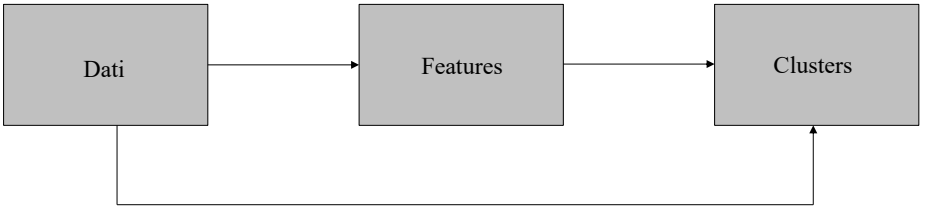
- **Pattern**: un singolo dato $\mathbf{X} = \{x_1, x_2, \ldots x_D\}$. Il dato appartiene quindi ad uno spazio multi-dimensionale (D dimensionale), solitamente eterogeneo (e.g. il livello di grigio dei pixel di un'immagine).

Feature: le caratteristiche dei dati significative per il clustering, possono costituire anch'esso un vettore multi-dimensionale (M-dimensionale), il vettore delle feature: $F = \{f_1, f_2, \ldots f_M\}$. Questo vettore costituisce l'input agli algoritmi di clustering.

E.g. Inclinazione, occhielli, lunghezza, linee orizzontali, archi di cerchio ... => Sistemi di riconoscimento della scrittura. **Feature eterogenee tra loro.**

Features

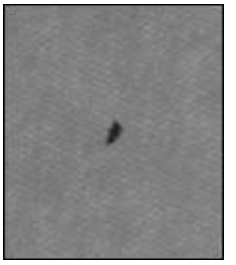
- Globali: livello di luminosità medio, varianza, contenuto in frequenza.....
- **Feature locali**

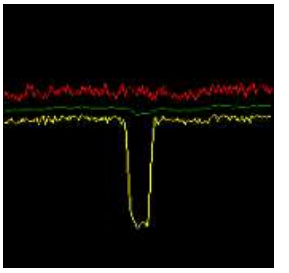


Feature locali

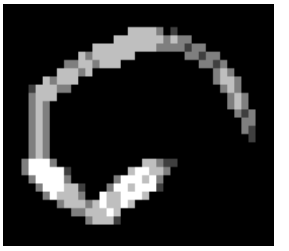
- *Località.*
- *Significatività.*
- *Rinoscibilità.*

Macchie dense





Fili



Total quality control assessment through boosting (Fomasi, Borghese 2015)

Clustering: definizioni

- **Pattern:** D - dimensione dello spazio dei pattern $\{x_i\}$ presentati dall'ambiente;
- **Feature:** M - dimensione dello spazio delle feature $\{f_j\}$;
- **Cluster**: in generale, insieme che raggruppa dati simili tra loro, valutati in base alle feature o ai dati stessi;
- **Funzione di similarità o distanza**: una metrica (o quasi metrica) nello spazio delle feature/dati, usata per quantificare la similarità tra due pattern.
- **Algoritmo**: scelta di come effettuare il clustering (motore di clustering).

Tante possibilità di scelta ➔ risultati diversi…

Feature selection

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

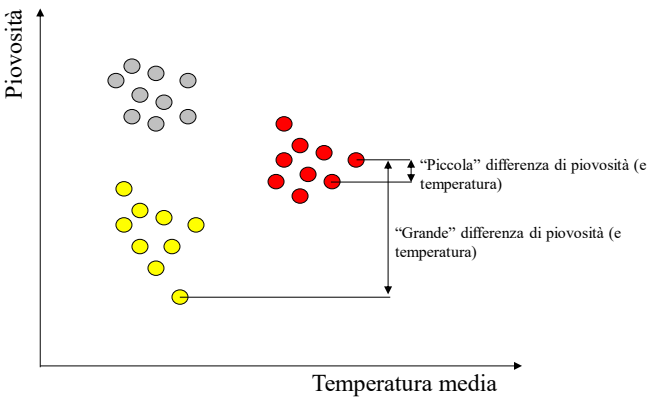
  Esempio: descrizione del **clima** della città di Roma mediante feature. Roma è caratterizzata da: [17°; 500mm; 1.500.000 ab., 300 chiese]
- Quali feature scegliere?
- Come valutare le feature?
  - Analisi statistica del potere discriminante: correlazione tra feature e loro significatività per il clustering.

Similarità tra feature / dati

- Definizione di una **misura di dissimilarità (distanza) tra due features / dati**;

  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 / dati

E' necessario trovare una metrica corretta per la rappresentazione. Per 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 → $P_{Norm} = (P - P_{Min})/(P_{Max} - P_{Min})$

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

$dist(Roma_{Norm}, Milano_{Norm}) = ((0.8-0.53)^2+(0.5-0.9)^2)^{½} = 0.4826 > 0.4 = 0.9 – 0.5$

E' una distanza compresa tra 0 (valore minimo) e 1 (valore massimo)

E' una buona scelta?

Normalizzazione su base statistica

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

Esempio:
P(x,y) ={2, 1; 2,2, 8; -2,4 -9; -3, -7,2; -1 -13; -1,23, 12; 1, 15,5; 1,12, 4; 0,22, -13; 1,4, -13}

La varianza è: Var = {3,37401, 120,3868}

La distanza al quadrato tra due punti consecutivi sarà:
Dist ={**49,04**, 3,6, 625,053, 132,2644. **1,3924**}

La distanza al quadrato normalizzata tra due punti consecutivi normalizzata secondo Malhanobis sarà:
Dist_norm su x e su y = {0,011855, 0,407021; 0,106698, 0,026913; 0,015679, **5,1916;** 0,004269, 1,0985; 0,412684, 0}
Dist_norm tra coppie = {**0,418877**, 0,133611, 5,20728, 1,10281, **0,412684**}

Possibilità di individuare gli outlier.

I punti 1 e 2 e 9 e 10 hanno distanze normalizzate simili, pur avendo distanze assolute molto diverse.

Altre metriche

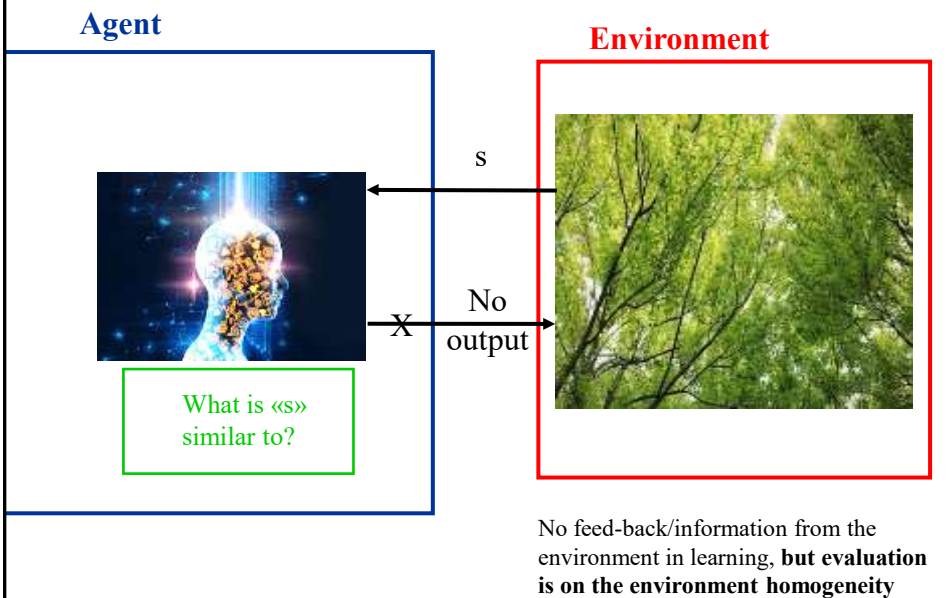
Altre metriche:

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

All'aumentare di p aumenta il peso degli outliers,
per p -> ∞ dist(x,y) = max(x,y)
per p=1 Manhattan or city-block distance
per p=2 Distance Euclidea

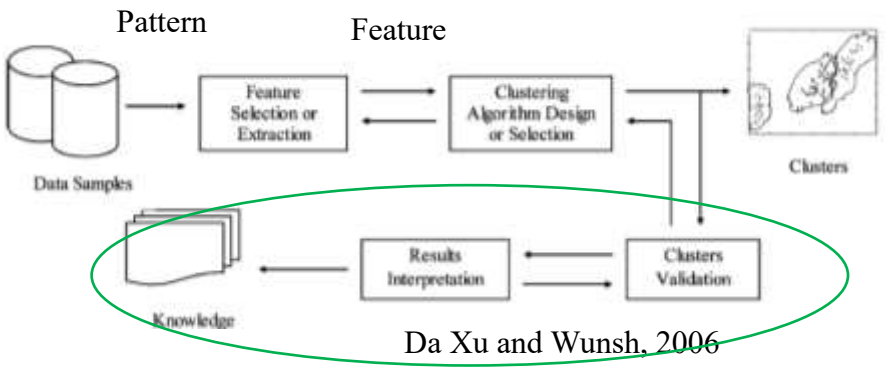
- Context dependent e funzioni di dissimilarità:
  diss(x,y)= f(x, y, context)

Apprendimento non supervisionato



No feed-back/information from the environment in learning, **but evaluation is on the environment homogeneity**

Clustering valutazione



Da Xu and Wunsh, 2006

I cluster ottenuti sono significativi?
Il clustering ha operato con successo?
Non è sufficiente che il codice funzioni in modo corretto!!!

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

Il clustering - riferimenti

Per una buona review: Xu and Wunsch, IEEE Transactions on Neural Networks, vol. 16, no. 3, 2005, Wiley Publisher 2008.

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, 1974 and followers, e.g. Vrieze 2012), soluzione empirica (growing networks di Fritzke 1996, Marsland 2002).

Tassonomia (sintetica) degli algoritmi di clustering

- Algoritmi **gerarchici** (agglomerativi, divisivi), e.g. **Hirarchical clustering**. Vengono partizionati (suddivisi in cluster) i dati forniti dall'ambiente. **Data-based**.
- Algoritmi **partizionali,** viene indotta una partizione nello spazio dei dati, ogni partizione può essere rappresentata da un **prototipo**. **Region-based.**
  - **hard-clustering**: **K-means, quad-tree decomposition**.
  - **soft-clustering**: competitive clustering, fuzzy c-mean, neural-gas, enhanced vector quantization, **mappe di Kohonen**.
- Algoritmi statistici: **mixture models** (misture statistiche: combinazione lineare di densità di probabilità).
- Algoritmi «black box» risolvono il problema dell'estrazione delle feature e del clustering in un unico passo (e.g. Deep NN). Tuttavia rimane il problema di quali feature vengano estratte, come vengano utilizzate…. («opening the box»).

Riassunto

- Il clustering e le feature
- Clustering statistico
- Clustering gerarchico
- Clustering partitivo hard

Esempio di clustering mediante mixture models

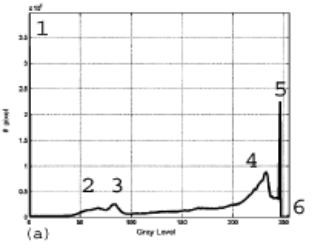
Radiografie cefaliche hanno facilmente problem di sovra-sotto esposizione.

Acquisizione su 12 bit e rappresentazione su 8 bit.

Tessuti molli e rigidi non sono facilmente visibili.

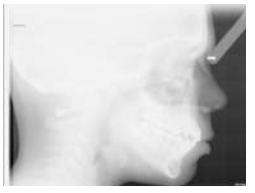
Immagine dopo clusterizzazione e filtraggio

Mixture models all'opera



Mixture di 3 Gaussiane:
- 2,3 picchi del soft tissue filtering
- 4 picco del bone tissue

Fitting del mixture model per trovare una buona soglia per separare soft and bone tissue



Selective gain applied to the two populations (gamma correction)

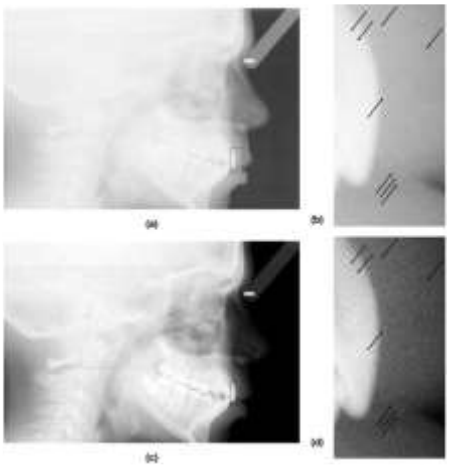
For more details: Frosio, Ferrigno, Borghese «Soft tissue filtering». IEEE Trans. Med. Imag., 2006; N.A. Borghese and I. Frosio (2006), "Soft Tissue Filtering in Medical Images", European Patent EP 1,624,411 A2, 8th February 2006.

Other mixtures: impulsive noise removal

Mixuture of Poisson noise + quantization noise (impulsive)

Pulse identification and correction through median filtering

Automatic gain estimate



For more details: Frosio, Borghese «Statistical Based Impulsive Noise Removal in Digital Radiography». IEEE Trans. Med. Imag., 2009.

25

Riassunto

- Il clustering e le feature
- Clustering statistico
- Clustering gerarchico
- Clustering partitivo hard

Hierachical Clustering

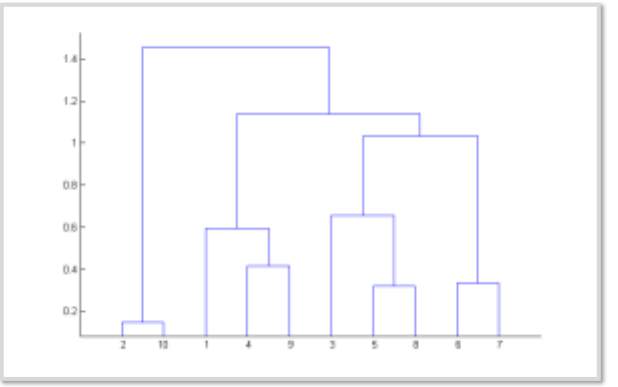
- In brief, HC algorithms build a whole **hierarchy** of clustering solutions (rappresentato da un albero)
  - Solution at level k is a *refinement* of solution at level k-1
  - Recursive subdivision / merging of cluster
- Two main classes of HC approaches:
  - **Agglomerative** solution at level k is 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 remains

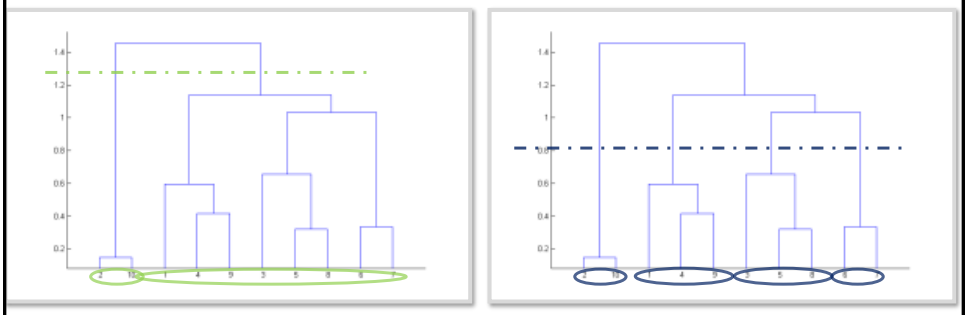
The clustering process is represented by a *dendrogram (albero)*

From as many clusters as the number of data to 1 cluster



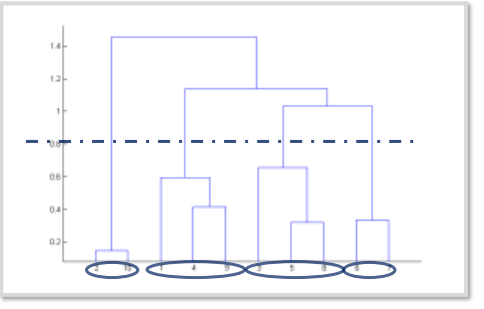
How to obtain the final solution

- The resulting dendrogram has to be cut at some level to get the final clustering:
  - Cut criterions
    - number of desired clusters,
    - threshold on some features computed on the current clusters
    - threshold on split criterion (e.g. standard deviation intra-cluster)



Merging clusters

- Each cluster is characterize by its elements
- Computation of the intra-class and inter-cluster dissimilarity
- Choose **2 clusters** to be merged that have minimal inter-cluster dissimilarity (maximum inter-cluster similarity)



Criterion to choose clusters to be merged

- Different indexes of **dissimilarity, d(x,y)** …
  - E.g. Euclidean, city-block, correlation, Malhanobis, (***point wise***)…
- … and agglomeration criteria: Merge clusters C_i and C_j such that *diss(i, j)* is minimum (***cluster wise***)

Dissimilarity computed on data:

- Single linkage:
  - diss(i,j) = min **d(x, y),** where x is in cluster C_i , y in cluster C_j
- Complete linkage:
  - diss(i,j) = max **d(x, y),** where x is in cluster C_i , y in cluster C_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: $w_i = n_i$, $w_j = n_j$

Dissimilarity measures computed on prototypes

Dissimilarity computed on cluster **prototypes**:

- Centroid Linkage:
  - diss(i, j) = $d(\mu_i, \mu_j)$ where μ_i is the centroid of cluster C_i, μ_j of cluster C_j
- Median Linkage:
  - diss(i,j) = $d(center_i, center_j)$, where $center_i$ is the median of the elements of 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 (minimum increase in variance)

- 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*

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:

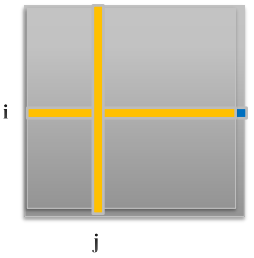
$$diss(k,(i,j)) = \alpha_i diss(k,i) + \alpha_j diss(k,j) + \beta diss(i,j) + \gamma |diss(k,i) - diss(k,j)|$$

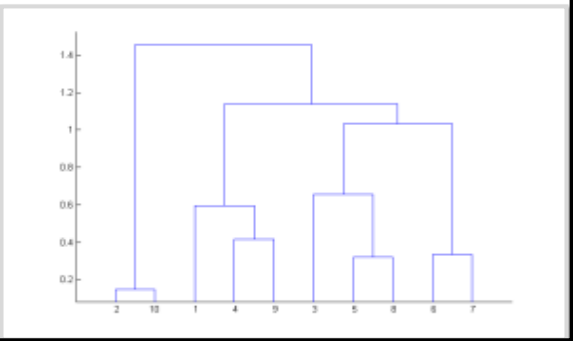
Only values already stored in the dissimilarity matrix are used. Different sets of coefficients correspond to different criteria.

Criterion	α_i	α_j	β	γ
Single Link.	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

How HC operates

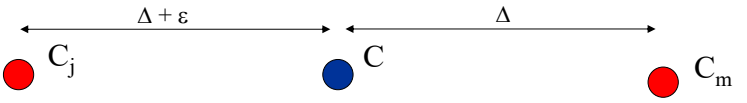
- HC algorithms operate on a dissimilarity matrix:
  - For each pair of existent clusters, their dissimilarity value is stored (e.g. minimum, maximum, increase in ESS…)
- When clusters C_i and C_j are merged, only dissimilarities for the new resulting cluster have to be computed
  - The rest of the matrix is left untouched





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



Which is the most adequate assignment of C?

Applicazione alla genetica

- Dati: interrelazione proteina-proteina nel database: BIOGRID.
- Identificazione dei cluster di proteine con funzionalità collegate
- A seconda dell'ordine di presentazione dei dati sono stati ottenuti 4 diversi dendrogrammi: 2 con 9 cluster e 2 con 10 cluster.
- Sono poi stati identificati i GO (Geno Ontology) terms that share the same functionality.

Analisi della soluzione

- Cluster costituiti rispettivamente da: {233, 233, 248, 249} Gene Ontology (GO) terms.
- Il numero di GO term differenti nelle diverse soluzioni sono rappresentati in tabella:
- Soluzioni diverse anche dal punto di vista funzionale!

Additional functional classes present in *S*4 are characterized by BP involved in the structural organization of cellular components and by its related anabolic/catabolic processes and are not present in the other solutions.

	S_1	S_2	S_3	S_4
S_1	-	0	5	23
S_2	0	-	5	23
S_3	20	20	-	20
S_4	39	39	20	-

For more details: Cattinelli, Valentini, Paulesu, Borghese. «A Novel Approach to the Problem of Non-uniqueness of the Solution in Hierarchical Clustering» IEEE Trans. NN-LS, 2013.

Algoritmi gerarchici divisivi: QTD (Quad Tree Decomposition)

Quad è una struttura regolare (quadrato) che contiene i dati che vengono analizzati. Si parte da un quad che contiene tutti i dati e che funge da unico cluster.

- Suddivisione gerarchica dello spazio delle feature, mediante suddivisionee regolare (**splitting**) dei cluster;
- Criterio di splitting (**criterio di similarità** intra e inter cluster).
- Si crea un alberto di quad.

Per spazi tridimensionali si ha octree decomposition, e decomposizione in iper-cubi.

Esempio di algoritmi gerarchici: QTD

- Clusterizzazione immagini RGB, 512x512;
- Pattern: pixel (x,y);
- Feature: canali [R, G, B] per ogni pixel.
- Definizione della distanza tra due pattern (massima sui tre canali):
  dist(p1, p2) =
  dist ([R1 G1 B1], [R2 G2 B2]) =
  max (|R1-R2|, |G1-G2|, |B1-B2|).

Splitting

In un quad, i pixel assumuno 3 valori RBG pari a:
p1 = [0 100 250]
p2 = [50 100 200]
p3 = [255 150 50]
dobbiamo suddividere il quad?

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.

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

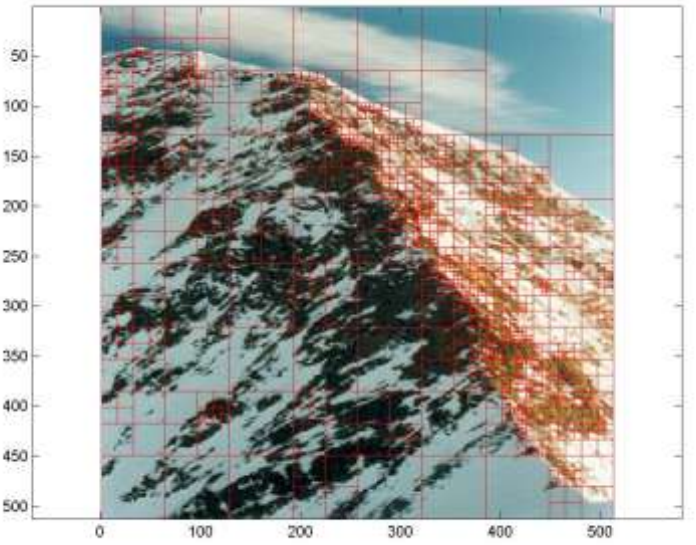
Esempio applicazione: segmentazione immagini, compressione immagini, analisi locale frequenze immagini…

QTD: Risultati

Original

Clusterized

QTD: Risultati



41

QTD: Risultati

Original

Clusterized

42

Riassunto

- Il clustering e le feature
- Clustering statistico
- Clustering gerarchico
- Clustering partitivo hard

Clustering partitivo

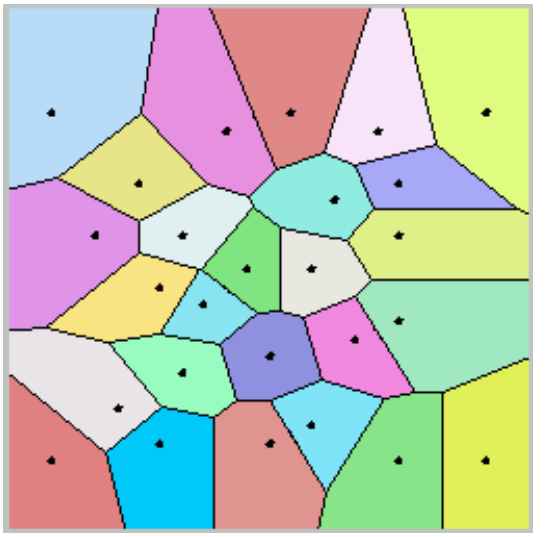
- Dati, $\{X_1 \ldots X_N\} \in R^D$
- Cluster $\{C_1 \ldots C_M\} \rightarrow \{P_1 \ldots P_M\} \in R^D$

$\mathbf{P_j}$ is the **prototype** of cluster j, it belongs to the same D-dimensional space of the data 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 as the data that are most similar among them (the boundary of a cluster is implicitly defined)
- Data can be analysed through their features.

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 oppure le feature estratte (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 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 la posizioni dei *prototipi* $\mathbf{C_j}$ mediante successive approssimazioni (iterazioni)

Algoritmo K-means

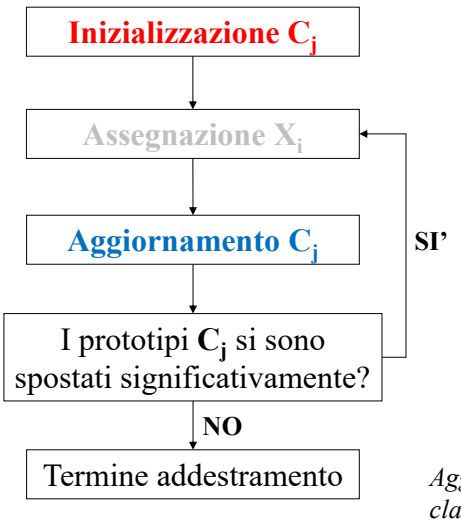
L'obiettivo che l'algoritmo si prepone è di **minimizzare la varianza totale intra-cluster**.

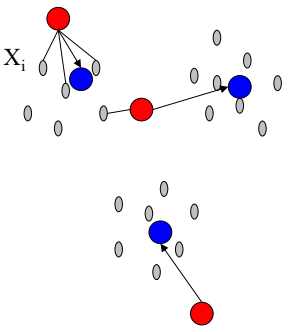
Ogni cluster viene identificato mediante un centroide o punto medio.

L'algoritmo segue una procedura iterativa.

- Inizialmente crea K partizioni e assegna ad ogni partizione i punti d'ingresso. Quindi calcola il centroide di ogni partizione.
- Costruisce quindi una nuova partizione dove i punti all'interno di ogni partizione sono più vicini al prototipo di quella partizione che a quelli delle altre.
- Quindi vengono ricalcolati i centroidi a partire dai dati nelle nuove partizioni.
- Finché i prototipi non subiscono più spostamenti (convergenza)

K-means: addestramento





Aggiornamento C_j*: baricentro degli* X_i *classificati da* C_j.

Clustering partitivo

- 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:

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

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

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

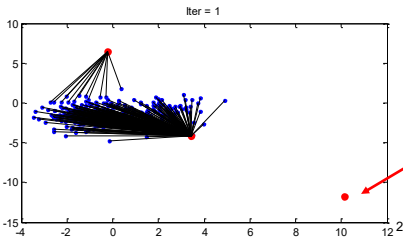
- La funzione obbiettivo soddisfa le proprietà degli spazi normati e metrici. Viene definita in generale come:

$$\sum_{i=1}^{K}\sum_{j=1}^{N}\left(\left|x_{j(k)} - C_k\right|\right)^2$$

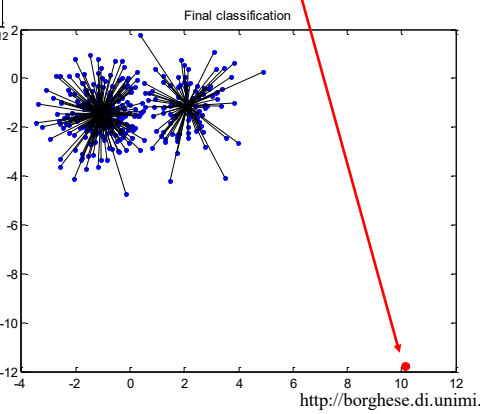
Algoritmo K-means::dettaglio dei passi

- 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 (≤ 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



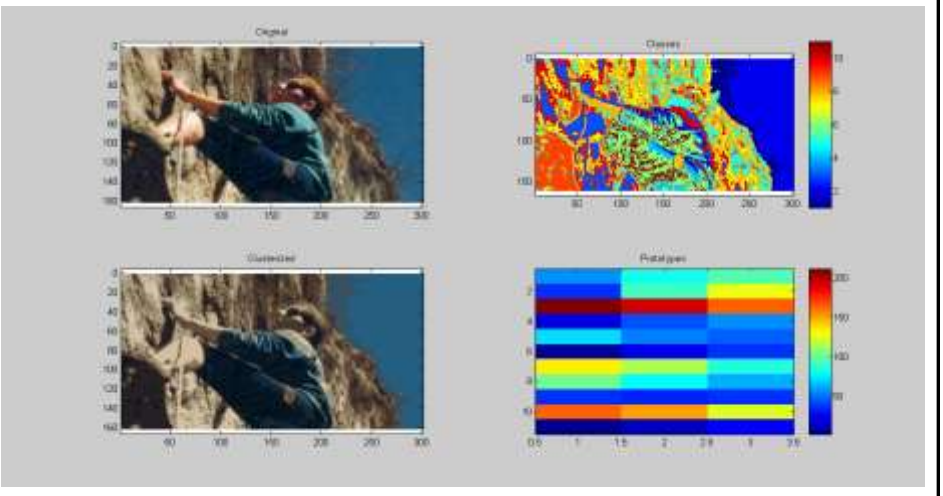
Dead units



Le dead units non hanno una posizione legata ai dati e non partecipano alla rappresentazione dei dati, anzi la disturbano.

Le dead unit devono essere eliminate (e così il cluster ad esse associato)

K-Means per immagine (clustering delle feature: colore RGB)



Da 255 colori a 33 colori

Riassunto

- Il clustering e le feature
- Clustering statistico
- Clustering gerarchico
- Clustering partitivo hard