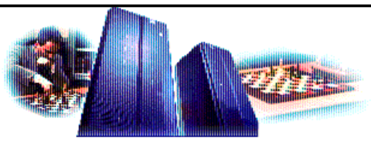
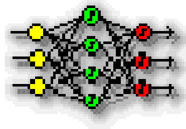




Clustering Gerarchico

Alberto Borghese

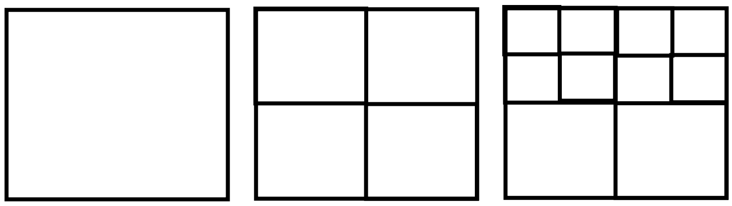
Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Scienze dell'Informazione
alberto.borghese@unimi.it

Riassunto

Hierarchical clustering

Algoritmi gerarchici: 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.

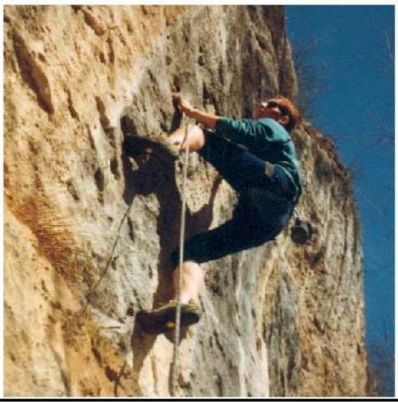
Algoritmi gerarchici: QTD

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…

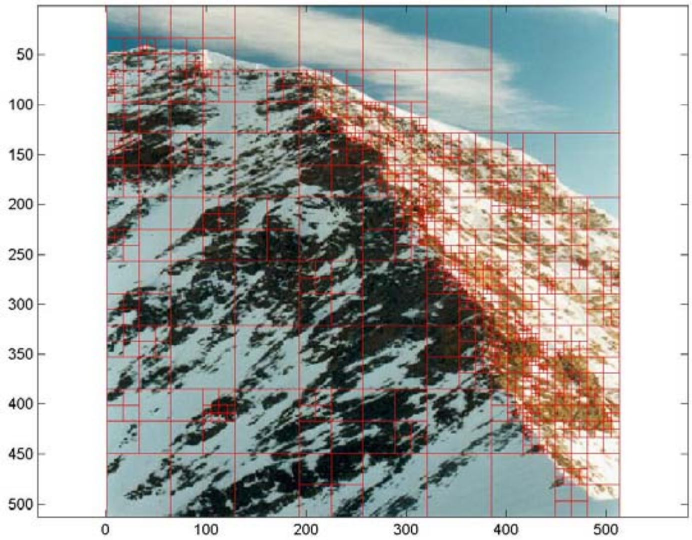
QTD: Risultati

Original



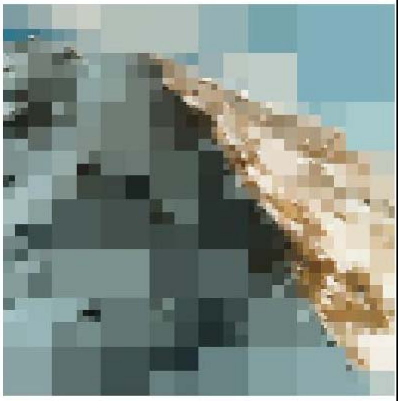
Clusterized

QTD: Risultati



A.A

QTD: Risultati



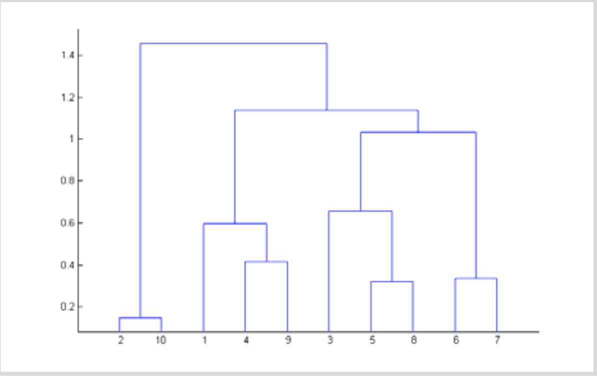
Hierachical Clustering

- 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

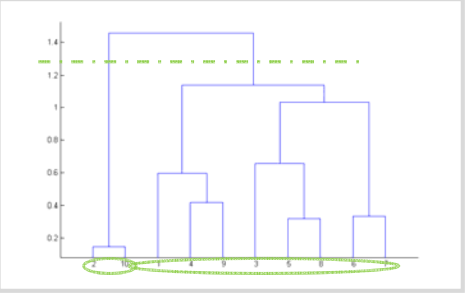
1. At start, each input pattern is assigned to a singleton cluster
2. At each step, the two *closest* clusters are merged into one
   - So the number of clusters is decreased by one at each step
3. At the last step, only one cluster is obtained

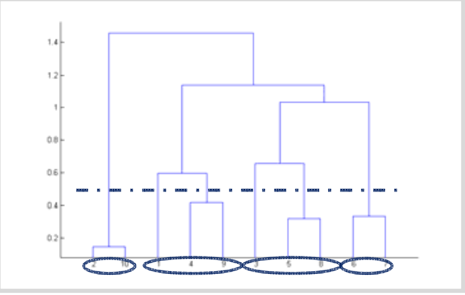
The clustering process is represented by a *dendrogram*



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





Dissimilarity criteria

- 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:

$$ESS_i = \sum_{x \in C_i} (x - \mu_i)^1 \qquad ESS = \sum_i ESS_i$$

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

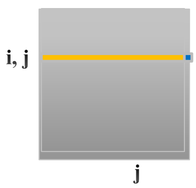
$$\Delta ESS = ESS_{i,j} - ESS_i - ESS_j$$

Spherical, compact clusters are obtained.

The solution at each level k is an <u>approximation</u> 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:

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

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

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

Hierarchical clustering