Clustering Techniques

Marco BOTTA
Dipartimento di Informatica
Università di Torino
botta@di.unito.it
www.di.unito.it/~botta/didattica/clutering.html

Data Clustering Outline

- What is cluster analysis ?
- What do we use clustering for ?
- Are there different approaches to data clustering ?
- What are the major clustering techniques ?
- What are the challenges to data clustering ?

Clustering 3

What is a Cluster *?*

- According to the Webster dictionary:
  - a number of similar things growing together or of things or persons collected or grouped closely together: BUNCH
  - two or more consecutive consonants or vowels in a segment of speech
  - a group of buildings and esp. houses built close together on a sizeable tract in order to preserve open spaces larger than the individual yard for common recreation
  - an aggregation of stars , galaxies, or super galaxies that appear together in the sky and seem to have common properties (as distance)
- A cluster is a closely-packed group (of things or people)

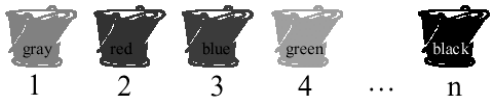
Clustering 4

What is Clustering in Data Mining?

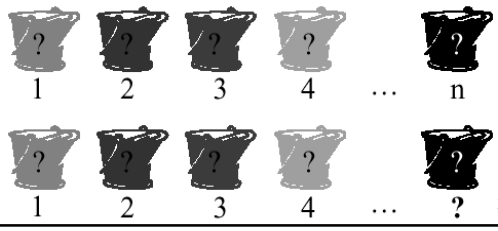
- Clustering is a process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called clusters.
  - Helps users understand the natural grouping or structure in a data set.
- Cluster: a collection of data objects that are “similar” to one another and thus can be treated collectively as one group.
- Clustering: unsupervised classification: no predefined classes.

Supervised and Unsupervised

- Supervised Classification = Classification
  - We know the class labels and the number of classes



- Unsupervised Classification = Clustering
  - We do not know the class labels and may not know the number of classes



ica

What Is Good Clustering?

- A good clustering method will produce high quality clusters in which:
  - the intra-class (that is, intra intra-cluster) similarity is high.
  - the inter-class similarity is low.
- The quality of a clustering result also depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.
- The quality of a clustering result also depends on the definition and representation of cluster chosen.

Requirements of Clustering in Data Mining

- Scalability
- Dealing with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Interpretability and usability.

Applications of Clustering

- Clustering has wide applications in
  - Pattern Recognition
  - Spatial Data Analysis:
    - create thematic maps in GIS by clustering feature spaces
    - detect spatial clusters and explain them in spatial data mining.
  - Image Processing
  - Economic Science (especially market research)
  - WWW:
    - Document classification
    - Cluster Weblog data to discover groups of similar access patterns

Examples of Clustering Applications

- *Marketing*: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.
- *Land use*: Identification of areas of similar land use in an earth observation database.
- *Insurance*: Identifying groups of motor insurance policy holders with a high average claim cost.
- *City-planning*: Identifying groups of houses according to their house type, value, and geographical location.
- *Earthquake studies*: Observed earthquake epicenters should be clustered along continent faults.

Major Clustering Techniques

- Clustering techniques have been studied extensively in:
  - Statistics, machine learning, and data mining

  with many methods proposed and studied.
- Clustering methods can be classified into 5 approaches:
  - partitioning algorithms
  - hierarchical algorithms
  - density-based
  - grid-based
  - model-based method

Clustering 11

Five Categories of Clustering Methods

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchical algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based algorithms: based on connectivity and density functions
- Grid-based algorithms: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Marco Botta, Dip Informatica

Clustering 12

Partitioning Algorithms: Basic Concept

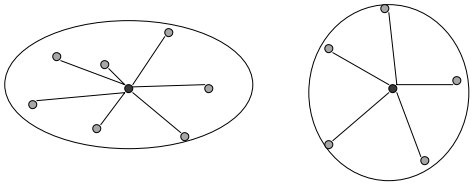
- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

Marco Botta, Dip Informatica

Optimization problem

- The goal is to optimize a score function
- The most commonly used is the square error criterion:

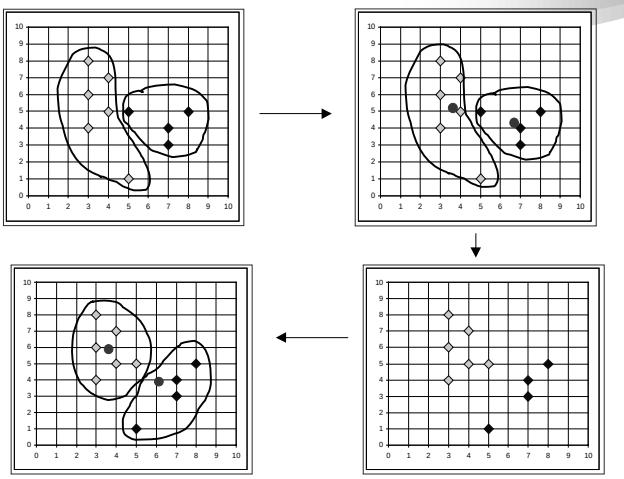
$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left\| p - m_i \right\|^2$$



The K-Means *Clustering Method*

- Given *k*, the *k-means* algorithm is implemented in 4 steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
  - Assign each object to the cluster with the nearest seed point.
  - Go back to Step 2, stop when no more new assignment.

The K-Means *Clustering Method*



Comments on the K-Means *Method*

- Strength
  - *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k*, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the K-Means *Method*

- A few variants of the k-means which differ in:
  - Selection of the initial k means.
  - Dissimilarity calculations.
  - Strategies to calculate cluster means.
- Handling categorical data: k-modes (Huang'98):
  - Replacing means of clusters with modes.
  - Using new dissimilarity measures to deal with categorical objects.
  - Using a frequency-based method to update modes of clusters.
  - A mixture of categorical and numerical data: k-prototype method.

The K-Medoids *Clustering Method*

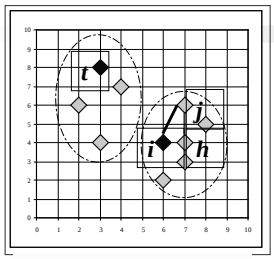
- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

PAM (Partitioning Around Medoids)

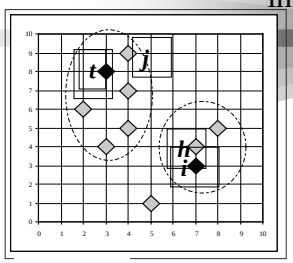
- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  - For each pair of ***i*** and ***h***,
    - If $TC_{ih} < 0$, ***i*** is replaced by ***h***
    - Then assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until there is no change

PAM Clustering: Total swapping cost

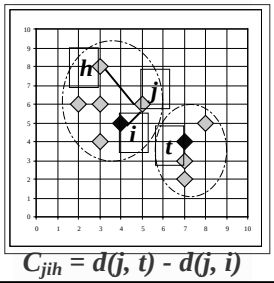
$TC_{ih}=\sum_j C_{jih}$



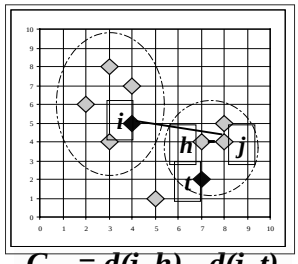
$C_{jih} = d(j, h) - d(j, i)$



$C_{jih} = 0$



$C_{jih} = d(j, t) - d(j, i)$



$C_{jih} = d(j, h) - d(j, t)$

CLARA *(Clustering Large Applications)*

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S+
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

CLARANS *("Randomized" CLARA)*

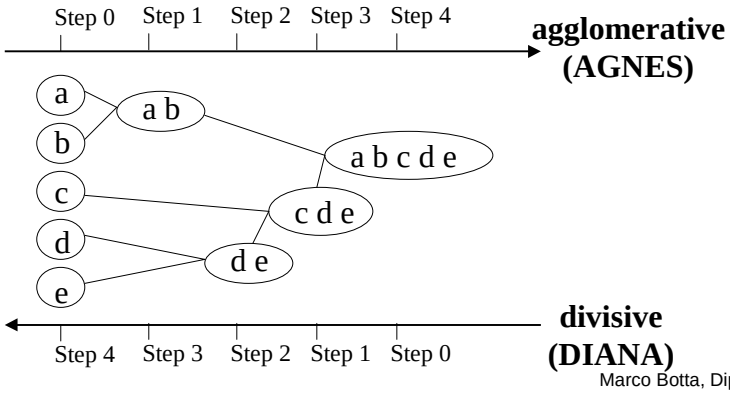
- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum
- It is more efficient and scalable than both *PAM* and *CLARA*

Two Types of Hierarchical Clustering Algorithms

- Agglomerative (bottom-up): merge clusters iteratively.
  - start by placing each object in its own cluster.
  - merge these atomic clusters into larger and larger clusters.
  - until all objects are in a single cluster.
  - Most hierarchical methods belong to this category. They differ only in their definition of between-cluster similarity.
- Divisive (top-down): split a cluster iteratively.
  - It does the reverse by starting with all objects in one cluster and subdividing them into smaller pieces.
  - Divisive methods are not generally available, and rarely have been applied.

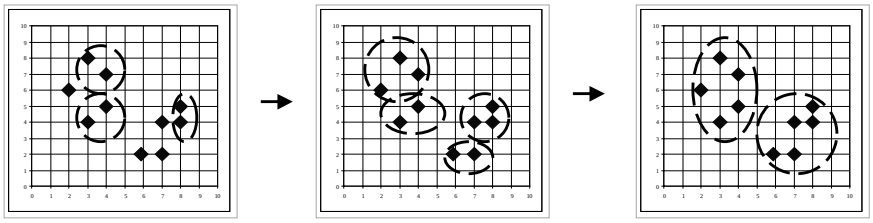
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



AGNES (Agglomerative Nesting)

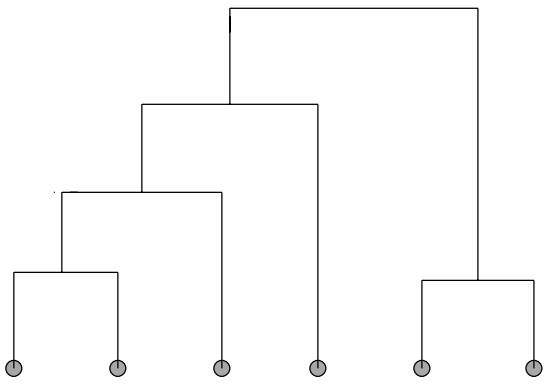
- Agglomerative, Bottom-up approach
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



A *Dendrogram* Shows How the Clusters are Merged Hierarchically

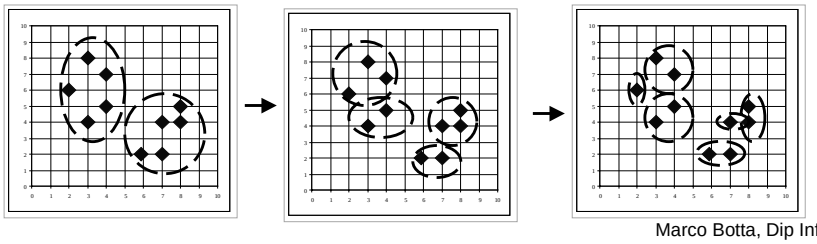
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.



DIANA (Divisive Analysis)

- Top-down approach
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



More on Hierarchical Clustering Methods

- Major weakness of vanilla agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction

BIRCH

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD'96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

BIRCH

- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record.

Clustering Feature Vector

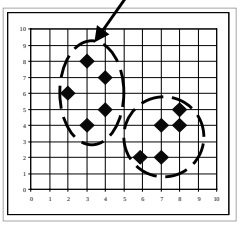
Clustering Feature: ***CF = (N, $\overrightarrow{LS}$, SS)***

N: **Number of data points**

LS: $\sum_{i=1}^{N} \overrightarrow{X_i}$

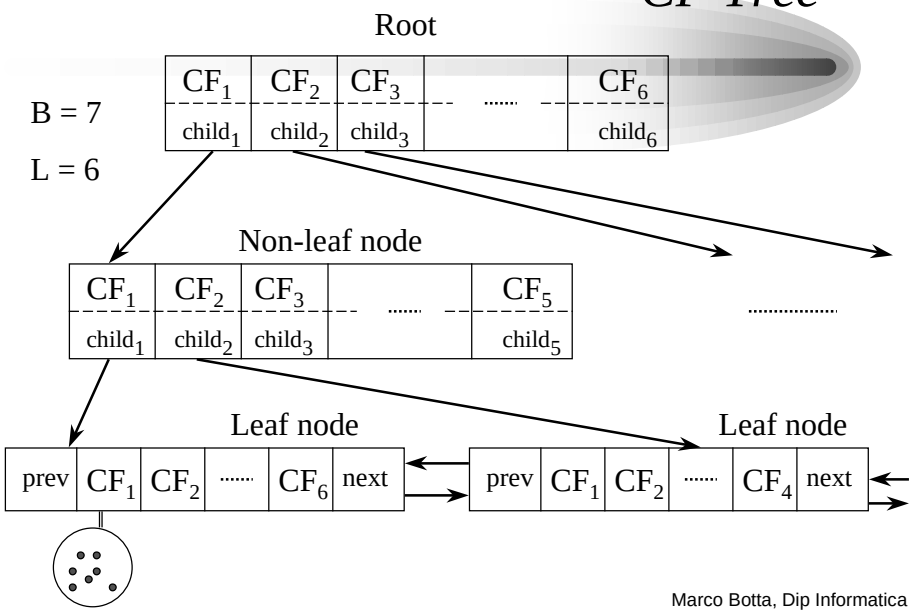
SS: $\sum_{i=1}^{N} \overrightarrow{X_i^2}$

CF = (5, (16,30),(54,190))

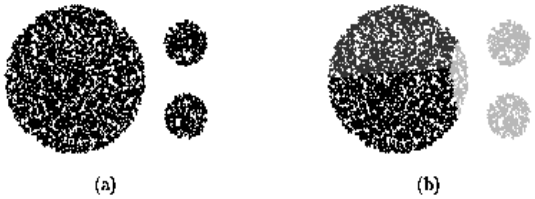


(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF Tree

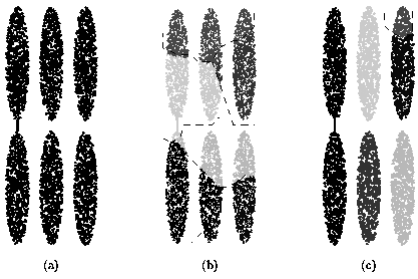


CURE (Clustering Using REpresentatives)



- CURE: proposed by Guha, Rastogi & Shim, 1998
  - Stops the creation of a cluster hierarchy if a level consists of *k* clusters
  - Uses multiple representative points to evaluate the distance between clusters, adjusts well to arbitrary shaped clusters and avoids single-link effect

Drawbacks of Distance-Based Method



- Drawbacks of square-error based clustering method
  - Consider only one point as representative of a cluster
  - Good only for convex shaped, similar size and density, and if *k* can be reasonably estimated

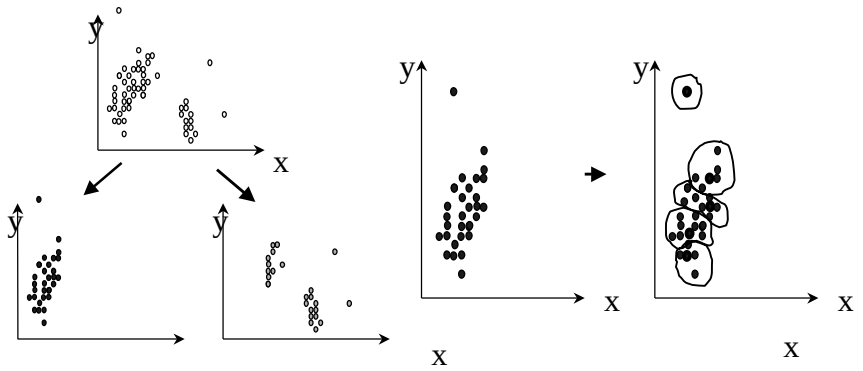
Cure: The Algorithm

- Draw random sample *s*.
- Partition sample to *p* partitions with size *s/p*
- Partially cluster partitions into *s/pq* clusters
- Eliminate outliers
  - By random sampling
  - If a cluster grows too slow, eliminate it.
- Cluster partial clusters.
- Label data in disk

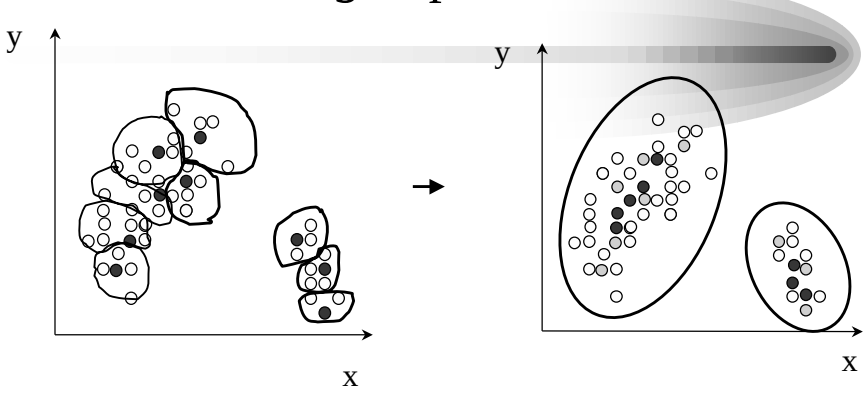
Data Partitioning and Clustering

- s = 50
- p = 2
- s/p = 25

- s/pq = 5



Cure: Shrinking Representative Points



- Shrink the multiple representative points towards the gravity center by a fraction of α.
- Multiple representatives capture the shape of the cluster

Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

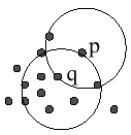
DBSCAN: A Density-Based Clustering

- DBSCAN: Density Based Spatial Clustering of Applications with Noise.
  - Proposed by Ester, Kriegel, Sander, and Xu (KDD'96)
  - Relies on a density-based notion of cluster: A cluster is defined as a maximal set of density- connected points
  - Discovers clusters of arbitrary shape in spatial databases with noise

Density-Based Clustering: Background

- Two parameter***s:***
  - ***Eps***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: ***{q belongs to D \| dist(p,q) <= Eps}***
- Directly density-reachabl**e:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** if
  - 1) ***p*** belongs to $N_{Eps}(q)$
  - 2) core point condition:

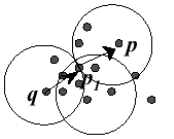
    $|N_{Eps}(q)| >= MinPts$

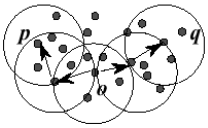


MinPts = 5
Eps = 1 cm

Density-Based Clustering: Background

- Density-reachable:
  - A point *p* is density-reachable from a point *q* wrt. *Eps*, *MinPts* if there is a chain of points p_1, …, p_n, p_1 = q, p_n = *p* such that $pi+_1$ is directly density-reachable from p_i
- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



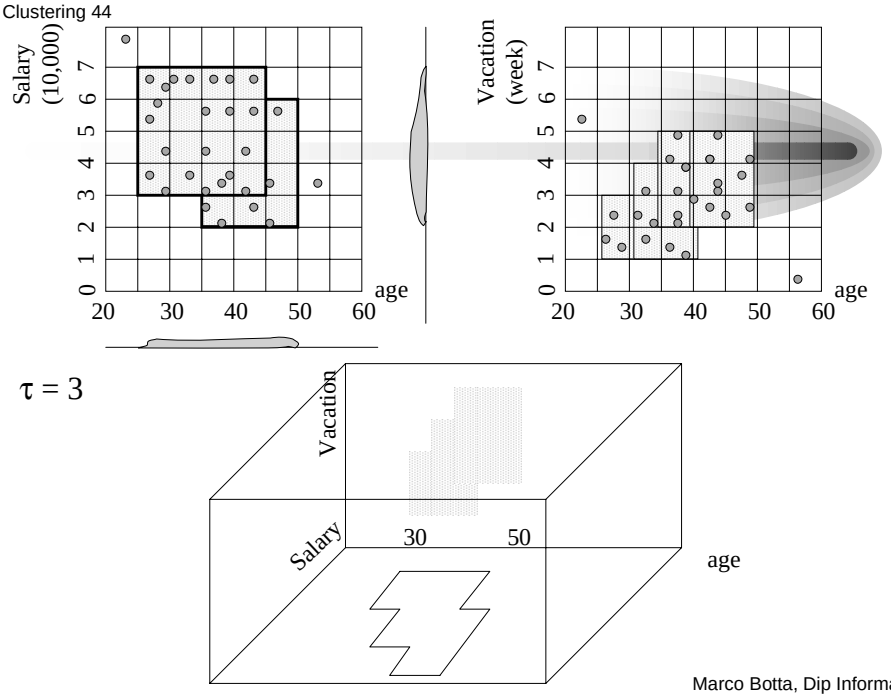


CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length interval
  - It partitions an m-dimensional data space into non-overlapping rectangular units
  - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - A cluster is a maximal set of connected dense units within a subspace

CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters:
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster



Strength and Weakness of CLIQUE

- Strength
  - It *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - It is *insensitive* to the order of records in input and does not presume some canonical data distribution
  - It scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

Grid-Based Clustering Method

- Grid-based clustering: using multi-resolution grid data structure.
- Several interesting studies:
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - BANG-clustering/GRIDCLUS (Grid-Clustering) by Schikuta (1997)
  - WaveCluster (a multi-resolution clustering approach using wavelet method) by Sheikholeslami, Chatterjee and Zhang (1998)
  - CLIQUE (Clustering In QUEst) by Agrawal, Gehrke, Gunopulos, Raghavan (1998).

Model-Based Clustering Methods

- Use certain models for clusters and attempt to optimize the fit between the data and the model.
- Neural network approaches:
  - The best known neural network approach to clustering is the SOM (*self-organizing feature map*) method, proposed by Kohonen in 1981.
  - It can be viewed as a nonlinear projection from an m-dimensional input space onto a lower-order (typically 2-dimensional) regular lattice of cells. Such a mapping is used to identify clusters of elements that are similar (in a Euclidean sense) in the original space.

Model-Based Clustering Methods

- Machine learning: probability density-based approach:
  - Grouping data based on probability density models: based on how many (possibly weighted) features are the same.
  - COBWEB (Fisher'87) Assumption: The probability distribution on different attributes are independent of each other --- This is often too strong because correlation may exist between attributes.

Model-Based Clustering Methods

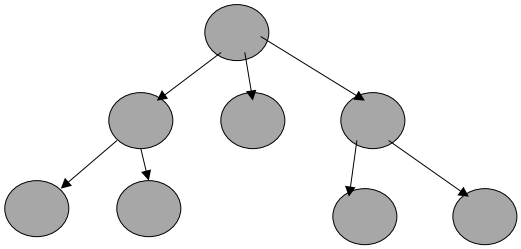
- Statistical approach: Gaussian mixture model (Banfield and Raftery, 1993): A probabilistic variant of k-means method.
  - It starts by choosing k seeds, and regarding the seeds as means of Gaussian distributions, then iterates over two steps called the estimation step and the maximization step, until the Gaussians are no longer moving.
  - Estimation: calculating the responsibility that each Gaussian has for each data point.
  - Maximization: The mean of each Gaussian is moved towards the centroid of the entire data set.

Model-Based Clustering Methods

- Statistical Approach: AutoClass (Cheeseman and Stutz, 1996): A thorough implementation of a Bayesian clustering procedure based on mixture models.
  - It uses Bayesian statistical analysis to estimate the number of clusters.

Clustering Simbolico: COBWEB

- Gerarchia di categorie (classi)
- Concetti probabilistici
- Algoritmo incrementale



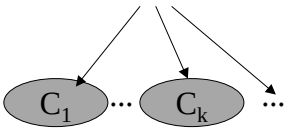
L'Algoritmo di COBWEB

Caratterizzato da:

- Una misura euristica di valutazione della rete di concetti.
- La rappresentazione dello stato del sistema
- Operatori usati per la costruzione della gerarchia di concetti.
- La strategia di controllo.

La Misura di Valutazione

- Dato un insieme A di attributi categorici
- Dato un insieme di Classi C



Intra-class similarity:

$$P(A_i = V_{ij} \mid C_k)$$

Inter-class dissimilarity:

$$P(C_k \mid A_i = V_{ij})$$

....La Misura di Valutazione...

- Data una partizione su n classi, si definisce una misura di valutazione:

$$\sum_{k=1}^{n} \sum_{i} \sum_{j} P(A_i = V_{ij}) P(C_k \mid A_i = V_{ij}) P(A_i = V_{ij} \mid C_k)$$

- Dal teorema di Bayes:

$$P(A_i = V_{ij}) P(C_k \mid A_i = V_{ij}) = P(C_k) P(A_i = V_{ij} \mid C_k)$$

$$\Downarrow$$

$$\sum_{k=1}^{n} P(C_k) \sum_{i} \sum_{j} P(A_i = V_{ij} \mid C_k)^2$$

.....La Misura di Valutazione

- Notiamo che:

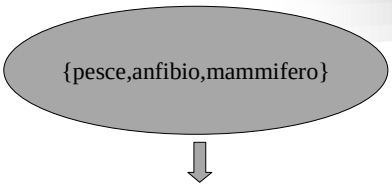
$$\sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2$$

è il numero atteso di attributi che si possono predire correttamente

- Category utility:

$$CU = \frac{\sum_{k=1}^{n} P(C_k)[\sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2]}{n}$$

Rappresentazione dei Concetti



copertura	scaglie[0.33], pelle umida[0.33], pelo[0.33]
camere cardiache	due[0.33, tre[0.33], quattro[0.33]
temperatura	non-regolata[0.67], regolata[0.33]
inseminazione	esterna[0.67], interna[0.33]

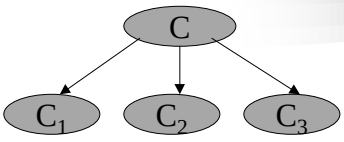
$$P(A_i = V_{ij} \mid C_k) = \frac{\#\, N\ di\ volte\ A_i = V_{ij}}{\#\, N\ di\ \ volte\ A_i\ e'\ stato\ osservato}$$

Operatori

- Classificazione di un oggetto rispetto ad una classe esistente.
- Creazione di una nuova classe
- Fusione di due classi
- Split di una classe in due nuove classi

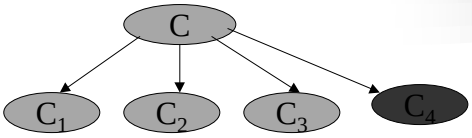
Operatori: classificazione

- Data:



- Un nuovo oggetto viene assegnato alla classe per cui la CU risultante è massima

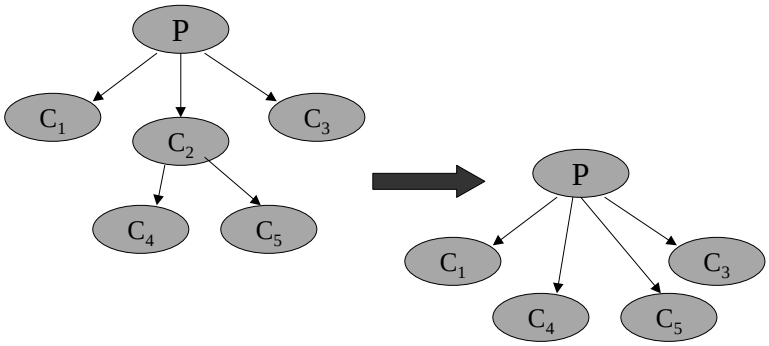
Operatori: creazione



- Se la CU è più alta di quella ottenuta assegnando la nuova istanza ad una classe esistente, la creazione può venire confermata

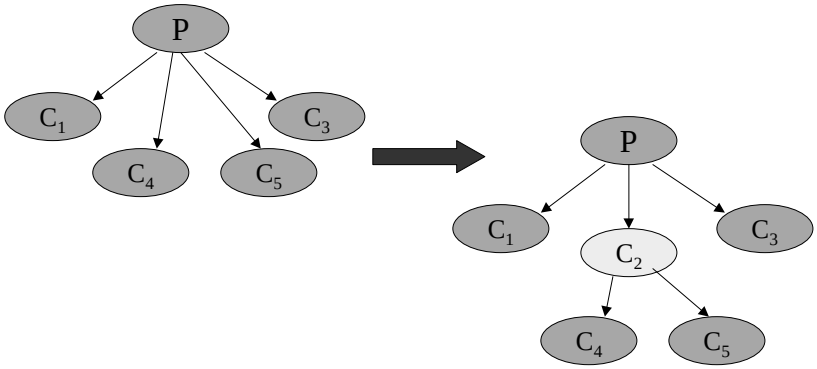
Operatori:split

- La condizione è che la CU dei figli di P cresca



Operatori: fusione

- La condizione è che la CU dei figli di P cresca



Strategia di Controllo

- Function COBWEB(Object, Root)

  1) Update counts of the root

  2) IF Root is leaf

  THEN Return the expanded leaf to accommodate Object

  ELSE Find the child of Root that best hosts Object and perform one of the following:

  a) Consider creating a new class and do so if appropriate

  b) Consider node merging and do so if appropriate, and call COBWEB(Object, Merged node)

  c) Consider node splitting and do so if appropriate, and call COBWEB(Object, Root)

  d) IF none of the above (a,b, or c) were performed THEN call COBWEB(Object, Best child of Root)

Complessità

- Sia:
  - B il branching factor medio nell'albero
  - A il numero di attributi
  - V il numero medio di valori per attributo
  - LogBn la profondita' stimata dell'albero
- Per ogni figlio di un nodo C il calcolo della CU è O(BAV)
- Essendo ripetuto per tutti i B figli il costo totale è O(B2AV)
- Moltiplicando per la profondità dell'albero:

$$Complessita` = O(B^2 \log_B n \times AV)$$

Utilizzazione

- Categorizzazione
- Classificazione come un decision tree

Problems and Challenges

- Considerable progress has been made in scalable clustering methods:
  - Partitioning: k-means, k-medoids, CLARANS
  - Hierarchical: BIRCH, CURE
  - Density-based: DBSCAN, CLIQUE, OPTICS
  - Grid-based: STING, WaveCluster.
  - Model-based: Autoclass, Denclue, Cobweb.
- Current clustering techniques do not address all the requirements adequately (and concurrently).
- Large number of dimensions and large number of data items.
- Strict clusters vs. overlapping clusters.