Data Clustering Analysis, from simple groupings to scalable clustering with constraints

Tutorial Notes: PAKDD 2002

Dr. Osmar R. Zaïane & Andrew Foss

University of Alberta
Canada

Objectives

- Learn basic techniques for data clustering.
- Understand the issues and the major challenges in clustering large data sets in multi-dimensional spaces.
- Discuss the issues of validating clustering results.

Data Clustering Outline

- What is cluster analysis and what do we use it for?
  - Motivation and Applications
- What are the important issues?
  - Categorical/Numerical data
  - Similarity/Distance measures
  - Noise and Outliers
- Are there different approaches to data clustering?
  - Partitioning/Hierarchical/Density-based/Grid-based…
- What are the other major clustering issues?
  - Multi-resolution detection
  - High-dimensionality
  - Clustering with constraints
  - Cluster validation

Data Clustering Outline

- What is cluster analysis and what do we use it for?
- What are the important issues?
- Are there different approaches to data clustering?
- What are the other major clustering issues?

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 sizable 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 close together in the sky and seem to have common properties (as distance).

➔ **A cluster is a closely-packed group (of people or things).**

What is Clustering in Data Mining?

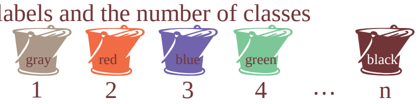
Clustering is a process of partitioning a set of data (or objects) in a set of meaningful sub-classes, called **clusters**.

- 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

What Is Good Clustering?

- A good clustering method will produce high quality clusters in which:
  - the **intra-class** (that is, 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
- Handles high dimensionality
- Interpretability and usability

Applications of Clustering

- 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
  - Separate objects from background and identify them
  - Remove noise
    - e.g. OCR, identify volcanoes on Venus, etc.
- Economic Science (especially market research)
  - e.g. Market basket analysis

More Applications

- WWW:
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns
- Outlier detection
  - Detecting credit card fraud, etc.

Further Examples of 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.

Data Clustering Outline

- What is cluster analysis and what do we use it for?
- What are the important issues?
- Are there different approaches to data clustering?
- What are the other major clustering issues?

Types of Data

- Numerical
  - Generally can be represented in a Euclidean Space.
  - Distance can be easily computed as a numerical difference.
- Categorical
  - A metric space may not be definable for this.
  - Distance has to defined in terms of similarity.

Handling Categorical Data: Similarity Measures

- Jaccard and Dice (functionally equivalent)
  - Jaccard $\frac{|X \cap Y|}{|X \cup Y|}$ Dice $\frac{2|X \cap Y|}{|X| + |Y|}$
- Overlap $\frac{|X \cap Y|}{\min(|X|, |Y|)}$
- Cosine $\frac{|X \cap Y|}{\sqrt{|X| \times |Y|}}$
- Other methods consider the sequence order.

Sequence Aware Similarity

- Compare the two sequences
  - Seq 1 : 1234
  - Seq 2 : 124
  - Score 1100 Result 2/4
- Can add spaces
  - Seq 1 : 1234
  - Seq 2 : 12_4
  - Score 1101 Result 3/4
- Optionally weight different levels.

Sequence Aware Similarity

- Can compute a value for a group of sequences using Dynamic Programming.
- Applications
  - Protein/DNA sequence alignment
  - Web log analysis
  - etc.

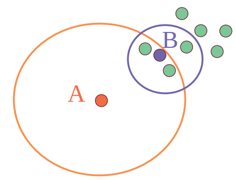
Noise and Outlier Detection

- Any clustering method, such as TURN*, AUTOCLUS and DBSCAN, that can differentiate remote points from internal ones or define very small unexpected clusters, can be used to find outliers.
- Algorithms that require the input of *k* are generally unsuitable.

Noise and Outlier Detection

- Most algorithms regard being an outlier as a binary property. Breunig et al. (SIGMOD 2000) defined a *local outlier factor (LOF).*
- This is a weighting for each point, its degree of being an outlier. *LOF* is based on MinPts, the number of points in the near neighbourhood.

LOF Explained



- Set MinPts = 4
- The 4-distance of A (d4, red) is greater than that of any of the others in its 4-neighborhood (e.g B).
- The ratio $\sum_{i=1}^{4} d4_i \;/\; 4d4_A$ gives a measure of its being an outlier.

Data Clustering Outline

- What is cluster analysis and what do we use it for?
- What are the important issues?
- Are there different approaches to data clustering?
- What are the other major clustering issues?

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 method**
  - **grid-based method**
  - **model-based method**

Partitioning
Hierarchical
Density-based
Grid-based
Model-based

Five Categories of Clustering Methods

- **Partitioning algorithms**: Construct various partitions and then evaluate them by some criterion.
- **Hierarchy algorithms**: Create a hierarchical decomposition of the set of data (or objects) using some criterion. There is an agglomerative approach and a divisive approach.
- **Density-based**: based on connectivity and density functions.
- **Grid-based**: 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.

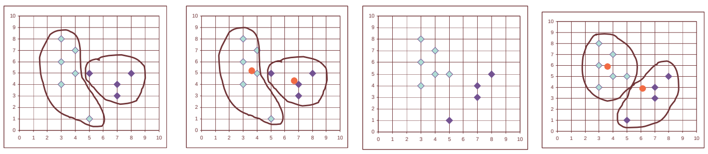
Partitioning Algorithms: Basic Concept

Partitioning
Hierarchical
Density-based
Grid-based
Model-based

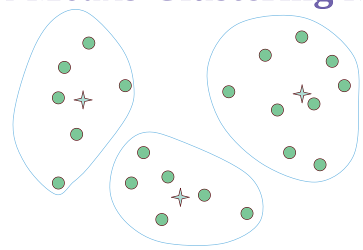
- **Partitioning method:** Construct a partition of a database ***D*** of *N* objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion.
  - Global optimal: exhaustively enumerate all partitions.
  - Heuristic methods: *k-means* 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.

The *K-Means* Clustering Method

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

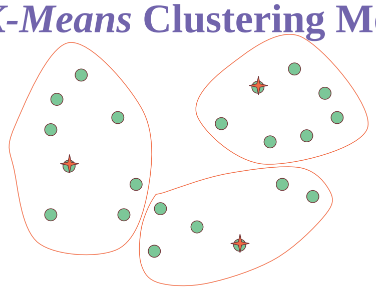


The *K-Means* Clustering Method



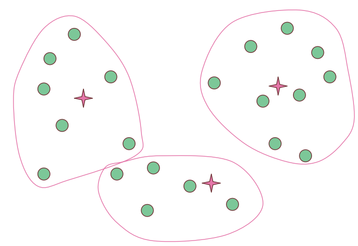
1.Select *k* nodes at random,
2.Assign all nodes to *k* clusters based on nearness.

The *K-Means* Clustering Method



1.Select *k* nodes at random,
2.Assign all nodes to *k* clusters based on nearness,
3. Calculate new means and reassign all nodes.

The *K-Means* Clustering Method



1.Select *k* nodes at random,
2.Assign all nodes to *k* clusters based on nearness,
3. Calculate new means and reassign all nodes.

Iterate until the criterion function converges
(e.g. squared error *node - mean* for all *k* clusters)

Comments on the *K-Means* Method

- Strength of the *k-means*:
  - *Relatively efficient*: $O(tkN)$, where N is # of objects, k is # of clusters, and t is # of iterations. Normally, $k, t << N$.
  - Often terminates at a *local optimum*.
- Weakness of the *k-means*:
  - 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.
- Use a representative point rather than a abstract cluster centre: *k-mediods*
- Handling categorical data: *k-modes* (Huang,1998)
- A mixture of categorical and numerical data: *k-prototype* method (Huang, 1997)

The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
  - To achieve this goal, only the definition of distance from any two objects is needed.
- *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 S+.
- Use real object to represent the cluster.
  1. Select ***k*** representative objects arbitrarily.
  2. For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}.
     - If $TC_{ih} < 0$, ***i*** is replaced by ***h***.
  3. Then assign each non-selected object to the most similar representative object.
  4. Repeat steps 2-3 until there is no change.

$O(k(N-k)^2)$

CLARA (Clustering Large Applications)

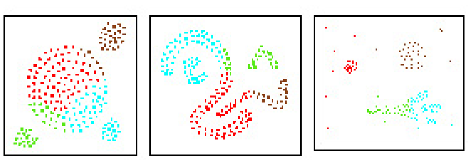
- *CLARA* (Kaufmann and Rousseeuw in 1990)
- Built in statistical analysis packages, such as S+.
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output.
- Strength of *CLARA*:
  - deal with larger data sets than *PAM*.
- Weakness of *CLARA*:
  - 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.

$O(kS^2+k(n-k))$

CLARANS ("Randomized" CLARA)

- *CLARANS* (A Clustering Algorithm based on Randomized Search) by Ng and Han in 1994.
- CLARANS draws sample of neighbours 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*.
- Focusing techniques and spatial access structures may further improve its performance (Ester et al.'95).

CLARANS Clustering Examples



From http://db.cs.sfu.ca/GeoMiner/survey/html/node9.html

The *K-Modes* Clustering Method

- K-means for categorical data.
- Uses a simple matching dissimilarity measure defined as the total mismatches of the corresponding attribute of 2 objects.
- Defines a mode of a set of objects $X = \{x_1, x_2, \dots x_N\}$ as a vector $Q = [q_1, q_2, \dots q_m]$ that minimises

$$D(X, Q) = \sum_{i=1}^{N} d(x_i, Q)$$

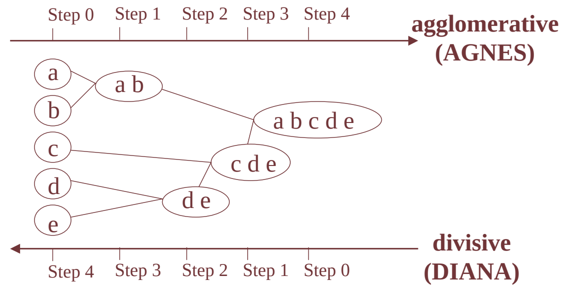
where $d(x_i, Q)$ is the disimilarity between x_i and Q.
- k-modes also favours 'spherical' clusters.

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.



Generic Methods for Computing Distance Between Clusters

- Single link
- Complete link
- Average link

The distance between one cluster and another cluster is equal to the shortest distance from any member of one cluster to any member of the other cluster. For categorical data, use the greatest similarity from any member of one cluster to any member of the other cluster.

Generic Methods for Computing Distance Between Clusters

- Single link
- Complete link
- Average link

In *complete-link* clustering (also called the *diameter* or *maximum* method), the distance between one cluster and another cluster is equal to the greatest distance from any member of one cluster to any member of the other cluster.

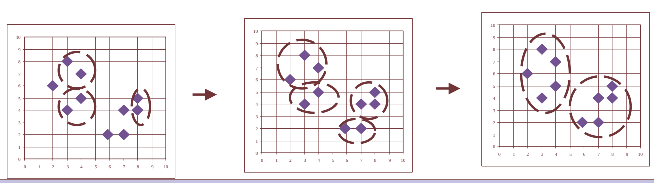
Generic Methods for Computing Distance Between Clusters

- Single link
- Complete link
- Average link

In *average-link* clustering, the distance between one cluster and another cluster is equal to the average distance from any member of one cluster to any member of the other cluster.

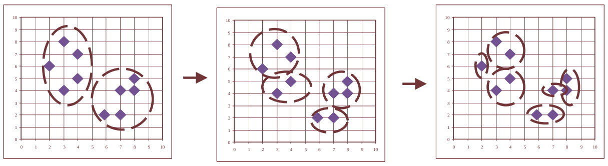
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, such as S+.
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, such as S+.
- Inverse order of AGNES.
- Eventually each node forms a cluster on its own.



PCA Partitioning

- Proposed by Moore et al. 1997
  - Cut the distribution with a hyperplane at the arithmetic mean normal to the principal direction (direction of maximum variance).
  - Repeat as often as desired.
  - Uses a scatter value, measuring the average distance from the nodes in a cluster to the mean, to determine next cluster to split.

More on Hierarchical Clustering

- Major weakness of 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 clustering with distance-based method:
  - 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.
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling.

BIRCH (1996)

- 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.
- *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

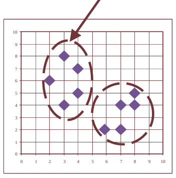
Clustering Feature: $CF = (N, \vec{LS}, SS)$

N: **Number of data points**

LS: $\Sigma^N_{i=1}=\vec{X_i}$

SS: $\Sigma^N_{i=1}=\vec{X_i}^2$

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



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

Clustering Categorical Data: ROCK

- ROCK: Robust Clustering using linKs, by S. Guha, R. Rastogi, K. Shim (ICDE '99).
  - Use links to measure similarity/proximity
  - Not distance-based
  - Computational complexity: $O(n^2 + nm_m m_a + n^2 \log n)$
- Basic ideas:
  - Similarity function and neighbours: $Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$

    Let T_1 = {1,2,3}, T_2={3,4,5}

$$Sim(T1, T2) = \frac{|\{3\}|}{|\{1,2,3,4,5\}|} = \frac{1}{5} = 0.2$$

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)
  - TURN*: Foss and Zaïane (2002)

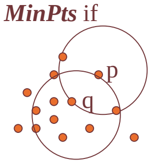
DBSCAN: A Density-Based Clustering

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

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

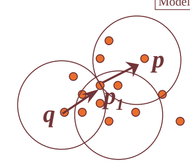


MinPts = 5

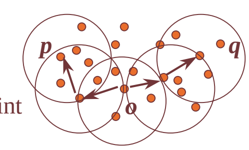
$\mathcal{E}$ = 1 cm

Density-Based Clustering: Background

- Density-reachable:
  - A point *p* is density-reachable from a point *q* wrt. ε, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



- Density-connected
  - A point *p* is density-connected to a point *q* wrt. ε, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. ε and *MinPts*.

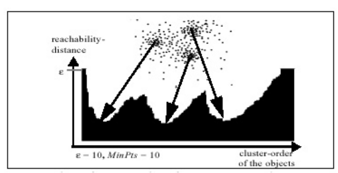


OPTICS: A Cluster-Ordering Method

- OPTICS: Ordering Points To Identify the Clustering Structure
  - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99).
  - Extensions to DBSCAN.
  - Produces a special order of the database with regard to its density-based clustering structure.
  - This cluster-ordering contains information equivalent to the density-based clusterings corresponding to a broad range of parameter settings.
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure.
  - Can be represented graphically or using visualization techniques.

OPTICS

- A motivation is to discover an optimum value of ε the reachability-distance as the clustering result is very sensitive to ε.
- OPTICS provides a view of the results for varying ε up to a limit.
- Creates a 2D plot allowing identification of clustering at different resolution levels.
- This is a plot of the reachability-distance (RD) for every object for ε = 10, MinPts = 10. If RD > ε, it is classed as 'Undefined'.



TURN*

(Foss and Zaiane, 2002)

Partitioning
Hierarchical
Density-based
Grid-based
Model-based

- TURN* contains several sub-algorithms.
- TURN-RES computes a clustering of spatial data at a given resolution based on a definition of ‘close’ neighbours: $d_i - d_j <= 1.0$ for two points *i, j* and a local density t_i based on a point’s distances to its nearest axial neighbours: $t_i = \Sigma_{d=0}^{D} f(d_d)$ for dimensions *D*.
- This density based approach is fast, identifies clusters of arbitrary shape and isolates noise.

TURN* Component Algorithms

Partitioning
Hierarchical
Density-based
Grid-based
Model-based

- TURN* wrapper algorithm finds the starting resolution and calls the other algorithms as needed as it scans over the range for which $k \geq 1$.
- TURN-RES generates both a clustering result and certain global statistics, especially a total density – the sum of the point local densities over all points clustered, excluding outliers.
- TurnCut finds the areas of interest in this graph using double differencing on the change values.

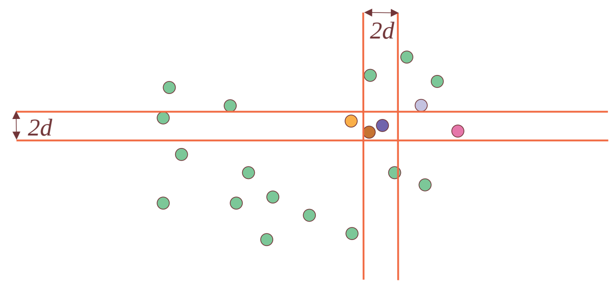
TURN*

Partitioning
Hierarchical
Density-based
Grid-based
Model-based

- A clustering result will be found over a certain range of resolutions. Outside of that there is either one cluster (S_1) or every point is classified as noise (S_∞).
- TURN* first searches for S_∞ and then scans towards S_1 using TURN-RES to cluster until a clustering optimum is reported by TurnCut assessing the global cluster features collected at each resolution by TURN-RES.
- First TURN-RES is explained….

Defining Neighbours

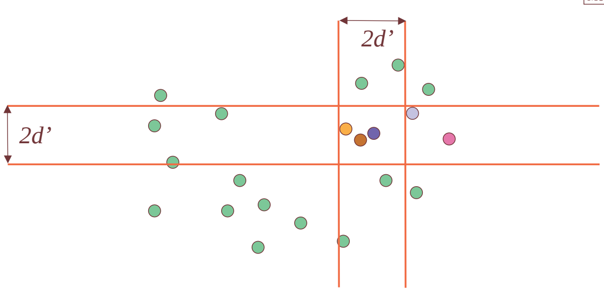
Partitioning
Hierarchical
Density-based
Grid-based
Model-based



A resolution is defined by a distance *d* along each dimensional axis. At this resolution the brown and pink points are nearest neighbours of the blue point along the vertical dimensional axis.

Defining Neighbours

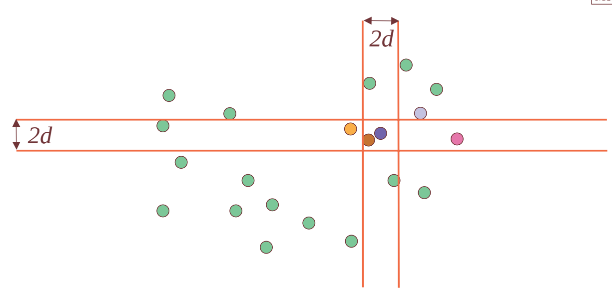
Partitioning
Hierarchical
Density-based
Grid-based
Model-based



At coarser resolution *d’* the silver point now replaces the pink as the right nearest neighbour of the blue point along the vertical dimensional axis.

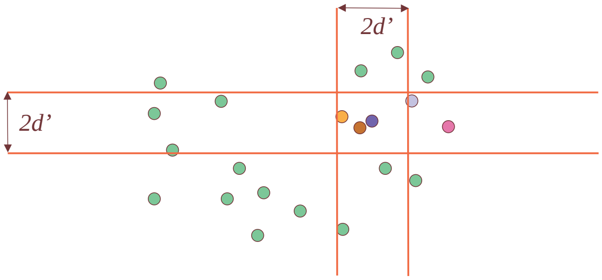
Defining Close Neighbours

Partitioning
Hierarchical
Density-based
Grid-based
Model-based



At resolution *d* the brown is a close neighbour of the blue point but the pink point is not close: *dist > d* along the vertical dimensional axis.

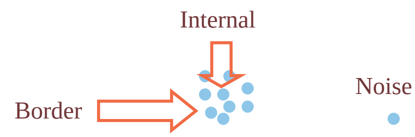
Defining Close Neighbours



At resolution *d'* both the brown and the orange points are close neighbours of the blue point.

TURN-RES

A local density for each point is computed that identifies if a point is 'internal' to a cluster. A border point is not internal but has 'close' neighbours that are and thus gets included in the cluster.



How TURN-RES Clusters

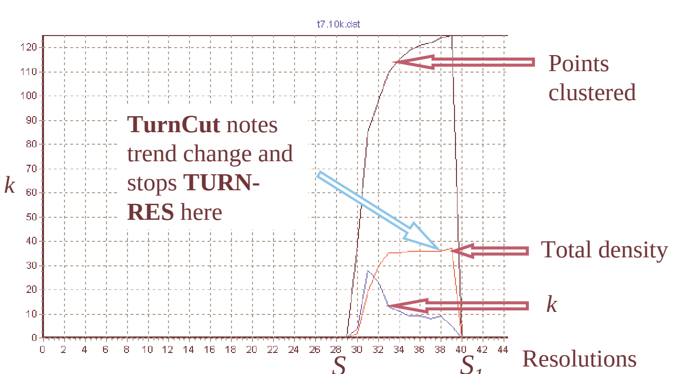
Type	Classified as	Close Neighbour?	Clustered
Interior	Internal	Yes	Yes
Border	External	Yes	Yes
Distant	External	No	No - Noise

All close neighbours to internal points are included into the cluster which pulls in the boundary points without the cluster extending out beyond the boundaries into noisy areas.

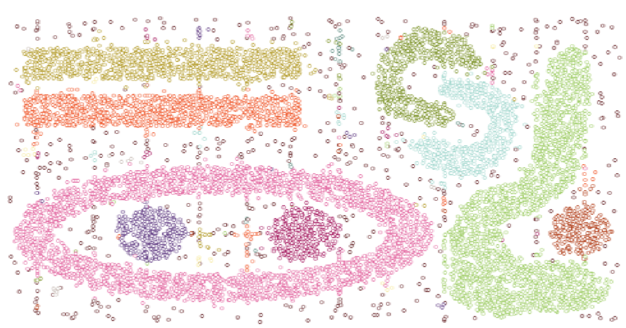
TurnCut and Differencing

- Single and double differencing is used in time series analysis to render a series stationary.
- Depends on the series having 'plateaus' with steps/transitions in between.
- It is a kind of high-pass filter that can reveal underlying trends.
- Here, we discover areas of stability ('plateaus') in the clustering (if any exist).

TurnCut Example

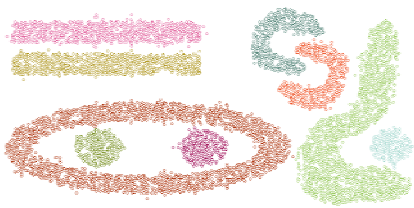


TURN* Example



TURN* used to Remove Noise

TURN* identifies outliers and small clusters. These can then be removed or separated from the rest of the clustered points.



Grid-Based Clustering Methods

- 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).

CLIQUE (1998)

- CLIQUE (Clustering In QUEst) by Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
- Automatic subspace clustering of high dimensional data
- CLIQUE can be considered as both density-based and produces a grid-like result.
- Input parameters:
  - size of the grid and a global density threshold
- It *partitions* an *m*-dimensional data space into non-overlapping rectangular units. This is done in 1-D for each dimension.
- 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.

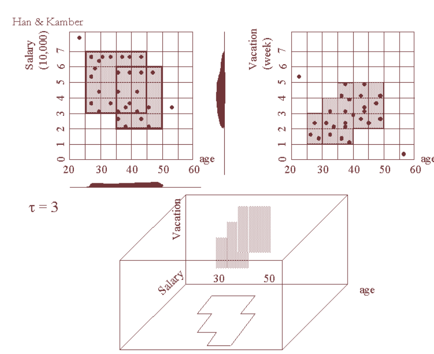
CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition after projection onto each dimension.
- Identify the subspaces that contain clusters, using the DNF expression, prune dimensions without dense units.
- 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.

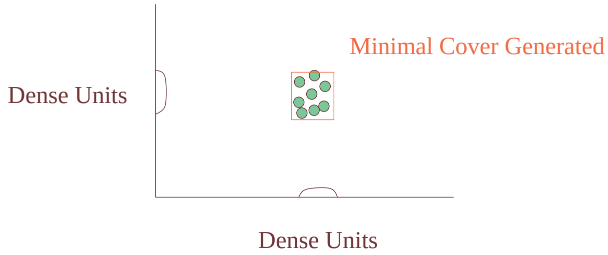
CLIQUE Pluses and Minuses

- Pluses
  - Automatically finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces.
  - Insensitive to input order, no assumption of any canonical distribution.
  - Scales linearly with input and well with increasing dimensionality.
- Minuses
  - Accuracy may suffer due to simplicity of the method.
  - Can easily find clusters that don't exist.

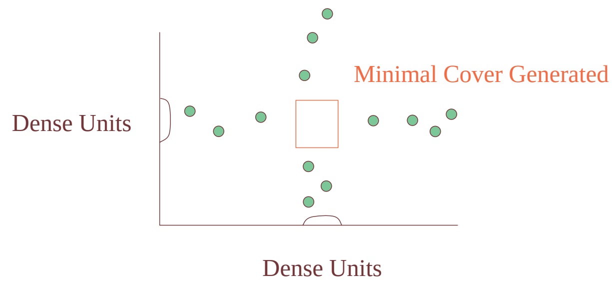
CLIQUE Example



CLIQUE Works



CLIQUE Fails



Model-Based Clustering Methods

Partitioning
Hierarchical
Density-based
Grid-based
Model-based

- 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.
- 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 (cont.)

Partitioning
Hierarchical
Density-based
Grid-based
Model-based

- 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.
- 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.

Some Important Algorithms Applied to a Complex 2D Spatial Dataset

(from the CHAMELEON Paper)

Best parameters chosen after many runs where needed.

K-means, ROCK, CURE and CHAMELEON

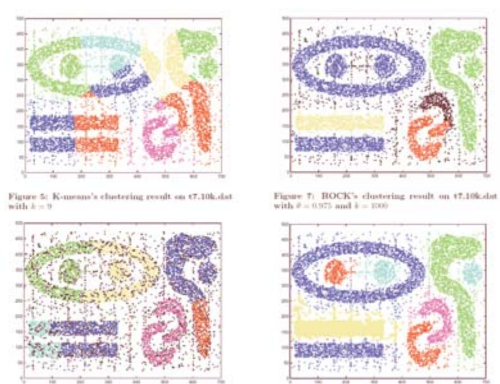


Figure 5: K-means's clustering result on t7.10k.dat with $k = 9$

Figure 7: ROCK's clustering result on t7.10k.dat with $\theta = 0.975$ and $k = 1000$

Figure 6: CURE's clustering result on t7.10k.dat with $k = 9$, $\alpha = 0.3$, and number_of_representative_points = 10

Figure 8: CHAMELEON's clustering result on t7.10k.dat with nl_closest_neighbor = 10, MinSize = 2.5% and $k = 9$

DBSCAN, WaveCluster and TURN*

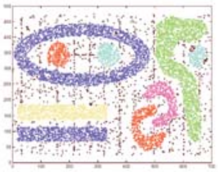


Figure 9: DBSCAN's clustering result on t7.10k.dat with $\epsilon = 5.9$ and MinPts = 4

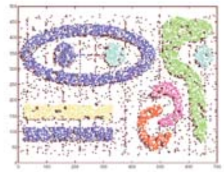


Figure 11: WaveCluster's clustering result on t7.10k.dat with resolution = 5 and $\tau = 1.5$

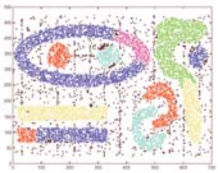


Figure 10: DBSCAN's clustering result on t7.10k.dat with $\epsilon = 5.5$ and MinPts = 4

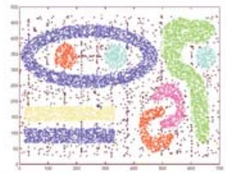


Figure 2: *TURN**'s clustering result on t7.10k.dat before cleaning

Tabular Comparison

Algorithm	Clustering time (secs)	Complexity	Memory Usage
K_means	8.44	$O(N)$	5.5MB
CURE	155.59	$\geq O(N^2)$	4.6MB
ROCK	526.19	$> O(N^2)$	1.145GB
CHAMELEON	1667.86	$> O(NlogN)$	8.6MB
DBSCAN	10.53	$O(NlogN)$	1.4MB
WaveCluster	0.82	$O(N)$	0.8MB
TURN-RES	0.26	$O(NlogN)$	1.4MB
*TURN**	3.90	$O(NlogN)$	1.4MB

Table 1: Clustering Speed and Memory Size Results upon a data set with 10,000 data points

Data Clustering Outline

- What is cluster analysis and what do we use it for?
- What are the important issues?
- Are there different approaches to data clustering?
- What are the other major clustering issues?

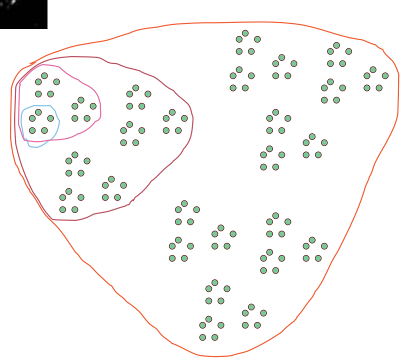
Multi-resolution
High dimension
Constraints
Validation

Resolution
a Key Issue in Clustering

- All algorithms face it.
- Very few papers discuss it.
- Usually dealt with by setting parameters.
- Trying to find the best parameters misses the point - useful information may be derived from many different settings.
- There may be a 'natural' best resolution or local areas may have different key resolutions.

Resolution

Multi-resolution
High dimension
Constraints
Validation



How many clusters?

As you zoom in and out the view changes.

Multi-resolution
High dimension
Constraints
Validation

Resolution
Some Approaches

- WaveCluster
- Optics
- TURN*

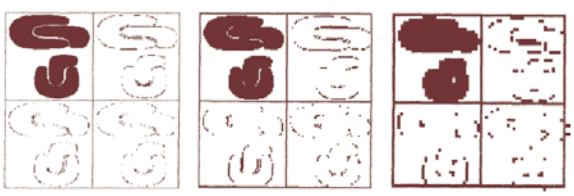
WaveCluster

Uses image processing techniques to find dense and sparse areas
- Apply grid to data.
- Apply noise filter to accentuate differences between grid nodes.
- Use wavelet transform to identify overall, horizontal and vertical high frequency components (cluster borders).
- Stitch adjacent dense areas defined by borders together into clusters.

Procedure

- Applies a grid reducing *N* data points to *n* grid points – key to fast processing.
- Applies a filter to reduce noise and sharpen boundaries.
- Apply wavelet transform on the reduced (feature) space.
- Find the connected components (clusters) in the subbands.
- The complexity *C* is *O(N)* if the whole grid can be represented in memory.

WaveCluster



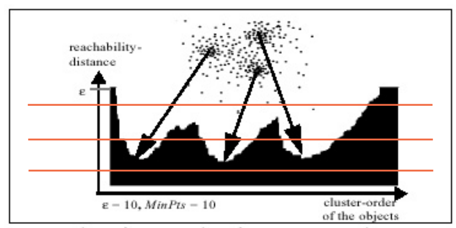
Three different resolution levels as seen by WaveCluster.

WaveCluster increasingly downsamples, omitting grid rows, so precision is traded for speed.

OPTICS

- Creates a 2D plot allowing identification of clustering at different resolution levels.

1 Cluster
3 Clusters
No Clusters

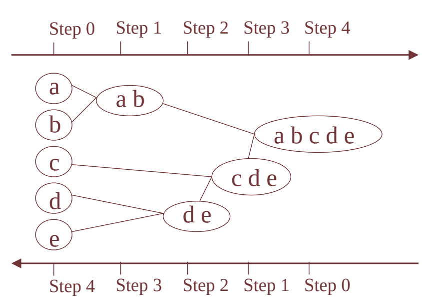


Proc. ACM SIGMOD'99 Int. Conf. on Management of Data, Philadelphia PA, 1999.

TURN*

- Finds the range of resolutions in which $k \geq 1$, after discarding outliers.
- Scans across this range detecting areas of stability representing 'interesting' clustering results and collecting clustering data and global clustering statistics.
- The data can be used to merge clusters of different densities or to manually review clustering at different resolution levels.

Many Algorithms Generate a Dendogram



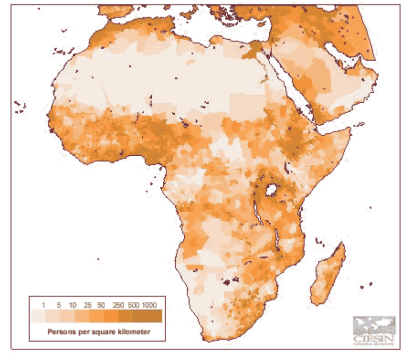
Cluster Tree / Dendogram

- Represents clustering at different resolution levels.
- Little discussion about significance of step sizes – few algorithms give any control.
- Is there an ‘optimum’ cluster result? Some authors assume so.
- Others combine different levels – ‘fuzzy’ clustering.
- Closely associated with Cluster Validation.
- Can we find an optimum algorithmically? (TURN*)

Fuzzy Clustering

Lets us combine areas of different features (whatever the algorithm uses to differentiate the clusters).

E.g. neighbouring areas of differing population density.



Fuzzy Clustering

- Usually seen as a weighting of partial cluster membership.
- Can also be seen as a ‘flattening’ or alternative representation of a dendogram.
- ‘Flattening’ causes a loss of information regarding the transformation between resolution levels which may be used in Cluster Validation.

Scaling to VLDB

- All algorithms discussed here are for data mining and thus intended to scale to handle VLDBs. However, hierarchical algorithms such as CHAMELEON, ROCK and CURE don’t scale well.
- Grid based methods are very effective because they condense the data.
- Methods such as DBSCAN and TURN* also scale well and compete with WaveCluster, etc. without the risks of condensation.

Scaling to High Dimensionality

- As the number of dimensions D increases, data gets sparser and any clustering effect is reduced.
- For $D > 16$ strategies for finding the near neighbours, such as indexed trees (e.g. SR-tree), fail and the computational complexity goes to $O(N^2)$.
- The length of each point’s description increases requiring more memory or I/O.

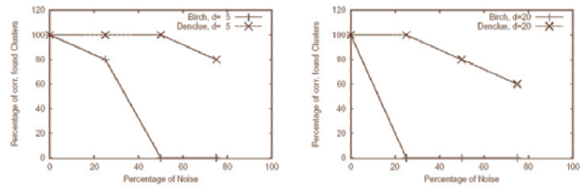
Some Solutions

- Grid based approaches reduce N or the search space
  - DENCLUE
  - OptiGrid (Hinneburg and Keim, 1999)
- Other methods attempt to discard the dimensions with the least contribution to the clustering
  - Singular Value Decomposition (SVD) - usually applied globally (e.g. Thomasian et al., 1998)
  - ORBCLUS - local dimensional reduction using SVD (Aggarwal and Yu, 2002)

DENCLUE

- Has been shown to generalise STING and WaveCluster
- Essentially DENCLUE models the overall point density analytically as the sum of influence functions of the data points. Clusters are identified by determining density-attractors and clusters of arbitrary shape are described by a simple equation based on the overall density function.
- For high-dimensional data, DENCLUE uses a grid-based approach but only stores the grid-cells which contain data points.
- DENCLUE connects all neighbouring populated grid cells of a highly-populated grid cell.

Comparison of DENCLUE & BIRCH on Noisy Data



(a) Dimension 5 (b) Dimension 20

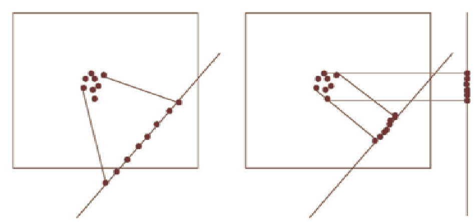
BIRCH has problems building its CF tree in the presence of noise. DENCLUE is good for low noise.

OptiGrid

(Hinneburg and Keim, VLDB 1999)

- Finds optimal grid partitioning in high dimensions by determining the best partitioning hyperplane in each dimension using contracting projections of the data.
- If a good partitioning hyperplane can not be found the dimension is discarded.
- Main challenge is finding the best projections.
- Uses axes parallel projections – max. error is $(1/2)^{d+1}$ theoretically (under assumptions).
- This is shown to be efficient and effective for normally distributed data with noise.

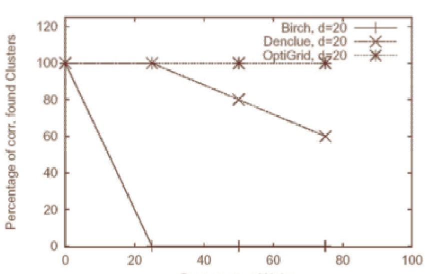
Contracted Projections



(a) general (b) contracting

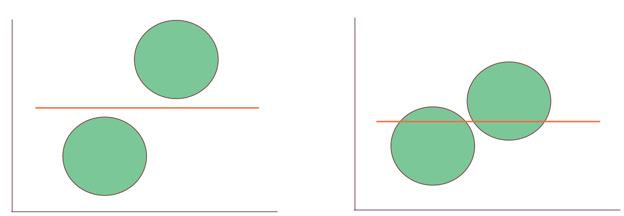
The cut is made at the density minimum on the contracted projection.

D = 20 Result for Normally Distributed Data with Noise



Axis Parallel Projections

Works for Has Problems with



Multi-resolution
High dimension
Constraints
Validation

Singular Value Decomposition

- The *d x d* covariance matrix is constructed. Each entry *(i, j)* is the covariance between dimensions *i* and *j*.
- The eigenvectors are found defining an orthonormal system along which the second order correlations are removed. The eigenvectors with the largest eigenvalues are chose to represent the data.

Multi-resolution
High dimension
Constraints
Validation

ORBCLUS

(Aggarwal and Yu, 2002)

- Clusters an initial sample creating a chosen number of clusters k_c and then applies SVD to each cluster.
- Points are added to the cluster to which they are 'closest'.
- Clusters are merged reducing k_c until the target k is achieved.
- Small clusters are discarded as outliers.
- Vulnerable to the initial sampling and clustering which has to be done with full-dimensionality.

Multi-resolution
High dimension
Constraints
Validation

Constraint Based Clustering

- Bridges (connecting points or sides of a polygon).
- Walls.
- Adding features to the constraints (bridge length, wall size, etc.).

Multi-resolution
High dimension
Constraints
Validation

Algorithms

AUTOCLUST+ (Estivill-Castro and Lee, 2000)
Based on graph partitioning

COD-CLARANS (Tung, Hou, and Han, 2001)
Based on CLARANS – partitioning

DBCluC (Zaiane and Lee, 2002)
Based on DBSCAN – density based

Multi-resolution
High dimension
Constraints
Validation

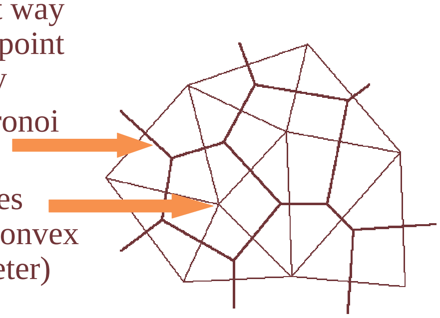
AUTOCLUST+

- Builds a Delaunay diagram for clustering data points – more scaleable and efficient than COD-CLARANS.
- Based on AUTOCLUST which is parameter free (though has built in parameters).
- Edges that cross obstacles are removed.
- Does not consider 'bridges'.
- O(Nlog(N)) complexity.

Multi-resolution
High dimension
Constraints
Validation

Delauney Triangulation

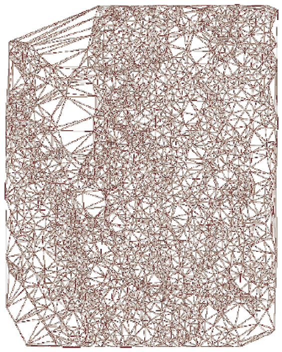
- An efficient way of defining point connectivity
- Dual to Voronoi diagrams
- Trianagulates within the convex hull (perimeter)



Multi-resolution
High dimension
Constraints
Validation

Delauney Triangulation

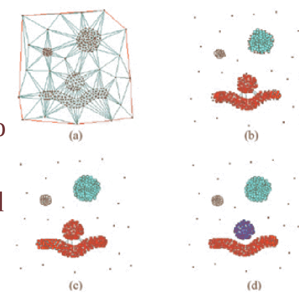
- All co-circularity is removed (not more than 3 points in any circum-circle)
- Most equilateral of triangulations
- Edge length can show cluster boundaries



Multi-resolution
High dimension
Constraints
Validation

AUTOCLUST uses Delauney Triangulation

- Create Delauney Diagram (a)
- Remove 'long' and 'short' edges (b)
- Cleanup using 'short' edges to classify doubtful points (c)
- k-edge local mean length used to remove additional 'long' edges (d)



Multi-resolution
High dimension
Constraints
Validation

AUTOCLUST

- Internal Parameters
  - Definition of 'long' and 'short'
    - Uses local mean edge length (LM) and a global standard deviation of edge length measure (MSD)
    - Picks a formula using these, e.g.
      - Long edge e, where $\text{length}_e > LM + MSD$
  - Choice of *k* (= 2 in paper)

Multi-resolution
High dimension
Constraints
Validation

COD-CLARANS

- Builds visibility graph to find the shortest distance between points. If an edge crosses an obstacle, it is removed.
- This preprocessing to the clustering is very expensive $O(N^2)$.
- CLARANS has its own limitations as discussed.

Multi-resolution
High dimension
Constraints
Validation

DBCluC

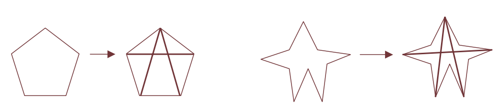
- Models obstacles and 'bridges' with polygons – a flexible approach
- Prunes the search space by reducing the polygons to a minimum set of lines
- Runs DBSCAN while prohibiting it to cross these lines
- Complexity O(nlog(n))

Multi-resolution
High dimension
Constraints
Validation

Modeling Constraints

– Polygon Reduction Algorithm

1. Convex Test - Determine if a polygon is a convex or a concave.
   - A polygon is *Concave* if ∃ a concave point in the polygon.
   - A polygon is *Convex* if ∀ points are convex points.
   - Convex - $\lceil n/2 \rceil$ obstruction lines*.
   - Concave – The number of obstruction lines depends on a shape of a given polygon.

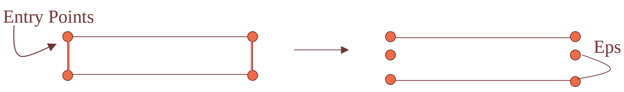


5 → 3 10 → 3

Modeling Crossings

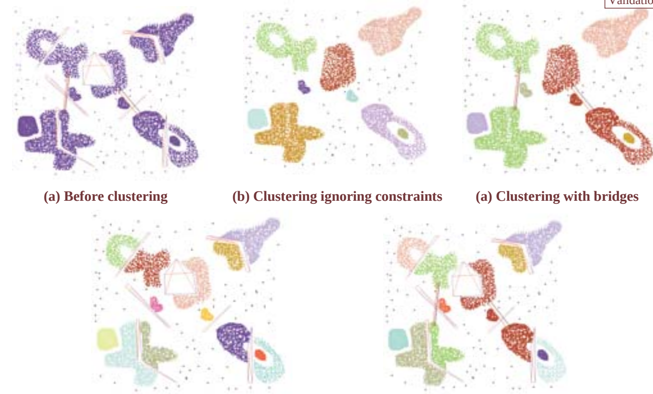
Multi-resolution
High dimension
Constraints
Validation

- Crossing Modeling
  - Objective
    - Connectivity functionality.
    - Control Flow of data.
  - A polygon with *Entry Points* and *Entry Edge*.
  - Defined by users or applications



Performance

Multi-resolution
High dimension
Constraints
Validation



(a) Before clustering (b) Clustering ignoring constraints (a) Clustering with bridges

(a) Clustering with obstacles (a) Clustering with obstacles and bridges

Cluster Validation

Multi-resolution
High dimension
Constraints
Validation

How good are the results from clustering algorithms?

Some Approaches

Multi-resolution
High dimension
Constraints
Validation

- Visual
- Statistical e.g. using cross-validation / Monte Carlo
- Based on internal cluster feature changes as resolution changes / dendogram traversal.

Mapping to 2D – for Visual Assesment

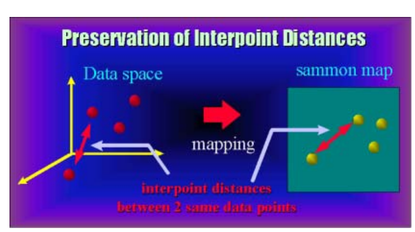
Multi-resolution
High dimension
Constraints
Validation

- Sammon
- SOM
- FastMap
- OPTICS
- Resolution problem

Sammon Mapping

Multi-resolution
High dimension
Constraints
Validation

- Non-linear mapping algorithm proposed by John Sammon in 1969.



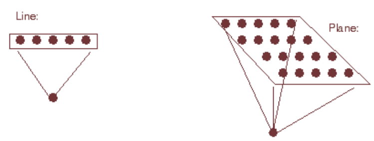
Sammon, cont.

- Uses a steepest descent algorithm to minimise the 'Sammon stress' which is a measure of the error based on the change in the interpoint distance differences $O(N^2)$.
- Starts from random distribution of the points.
- High cost can be reduced using Conjugate Gradient approach (for example).

Sammon, cont.

- Efficient in identifying hyper-spherical, hyper-ellipsoidal clustering, whose geometric relationships are obviously governed by the inter-point distances, but…
- Have to start over if any new points added.
- May stop on local minima but can make several runs.

Kohonen SOM (1981)



- Uses a Neural Net to map the data onto 2D.

SOM, cont.

- Finds the winning node for input vector x:
- $|\omega_i - x| \le |\omega_k - x| \qquad \forall k$
- Updates weights using a neighbourhood function
- It has value 1 when i=k and falls off with the distance $|r_k - r_i|$ between units i and k in the output array

$$w_k(new) = w_k(old) + \mu\, \aleph(i,k)\,(x - w_k)$$

SOM, cont.

Units close to the winner as well as the winner itself, have their weights updated appreciably while weights associated with far away output nodes do not change significantly. It is here that the topological information is supplied. Nearby units receive similar updates and thus end up responding to nearby input patterns.

FastMap

(Faloutsos and Lin, SIGMOD 1995)

- Projects objects and candidate clusters onto a 2D space for visual analysis.
- Based on the principle that a cluster that is separated in 2D (or 3D) will be separate in higher dimensions.
- Uses pivot points for the projection, looks for points as far separated as possible.

Disadvantages of Visual Assessment

- Visual assessment is subjective.
- Depends on the resolution presented to the viewer (as already seen).
- Projection onto a lower dimensional space can always obscure certain structures.

Statistical

- Various approaches
  - Internal
    - Assumes an intuition about the clustering outcome, e.g. value of cluster number *k*.
  - External
    - Uses some internal feature for comparison, e.g. the proximity matrix P.
  - Relative
    - evaluate which of several results arising from different parameter settings (e.g. *k*) give the best result. Have to characterize the clustering result in a way that tells us about the quality of the clustering.

Statistical, cont.

Internal and External approaches use Monte Carlo methods

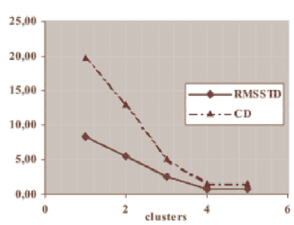
- Many synthetic datasets are generated and clustered and the outcome distribution tells us if the actual clustering is significantly different from chance.
- Depends on the validity of the assumptions made. E.g. *k* (External) or construction of P and statistic used for comparison (Internal).

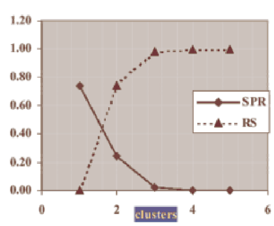
Relative

- The Relative approach depends on the characterization of data. In short, the validation method such as the computation of Root-mean-square standard deviation, a measure of homogeneity, is compared with the clustering algorithms own measure.
- Methods proposed to date fail with non-spherical clusters. May be inferior to the algorithm they test.

Relative, cont.

- Often looks for a ‘knee’ in a graph to show the ‘right’ value of *k*.





Monte Carlo Cross-Validation

- Smyth (KDD ’96) introduced this approach for data sets where a likelihood function such as Gaussian mixture models can be defined.
- An EM algorithm is used to define *nc* for $nc = 1$ to c_{max} clusters and a log likelihood *L* is calculated for each *nc*.
- This is repeated *m* times over *m* randomly chosen train/test partitions to computer posterior probablities *p* for each *nc* looking for *p* close to 1.
- This depends on estimating certain parameters for the data set.
- Uses a variant of k-means and finds the best *k*.

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, TURN*
  - Grid-based: STING, WaveCluster.
  - Model-based: Autoclass, Denclue, Cobweb.
- Current clustering techniques may not address all the requirements adequately (and concurrently).
- Large number of dimensions and large number of data items.
- Strict clusters vs. overlapping clusters.
- Clustering with constraints.
- Cluster validation.

References

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'98)*, pages 94-105, 1998.
- A.K.Jain, M.N.Murty, and P.J.Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264-323, 1999.
- N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. 1990 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'90)*, pages 322-331, 1990.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining (KDD'96)*, pages 226-231, 1996.
- A. Foss, W. Wang, and O. R. Zaiane. A non-parametric approach to web log analysis. In *Proc of Workshop on Web Mining in First International SIAM Conference on Data Mining (SDM2001)*, pages 41-50, Chicago, April 2001.
- A. Foss, and O. R. Zaiane. TURN* - Unsupervised Clustering of Spatial Data. Submitted to *SIGKDD*, 2002.
- S. Guha, R. Rastogi, and K. Shim. ROCK: a robust clustering algorithm for categorical attributes. In *15th Int'l Conf. on Data Eng.*, 1999.
- J. Han and M. Kamber. *Data Mining, Concepts and Techniques*. Morgan Kaufmann, 2001.
- J. Moore, S. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher, Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering, I7th Workshop on Information Technologies and Systems, Dec 97.
- G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGraw Hill, New York, 1982.
- A. Thomasian, V. Castelli and C.-S. Li. Clustering and Singular Value Decomposition for Approximate Indexing in High Dimensional Spaces. In *Proc. Conf. Information and Knowledge Management*, 1998.
- K.V.R Kanth, D. Agarwal and A. Singh. Dimensionality Reduction for Similarity Searching in Dynamic Databases. *Proc. ACM SIGMOD Conf.*, 1998.
- J. Han, M. Kamber, and A. K. Tung. Spatial clustering methods in data mining: A survey. In H. Miller and J. Han, editors, *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis, 2001.
- C.C. Aggarwal and P.S. Yu. Redefining Clustering for High-Dimensional Applications. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 2, 2002.
- A. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

- A. Hinneburg and D.A. Keim. Optimal Grid-Clustering: Toward Breaking the Curse of Dimensionality in High-Dimensional Clustering. Proc. 25th VLDB Conf., 1999.
- A. Hinneburg and D.A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. *Proc. 4rd Int. Conf. on Knowledge Discovery and Data Mining*, AAAI Press, 1998.
- M. V. M. Halkidi, Y. Batistakis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3), December 2001.
- M.Ankerst, M.Breunig, H.-P. Kriegel, and J.Sander. OPTICS: Ordering points to identify the clustering structure. In *Proc. 1999 ACM-SIGMOD Conf. on Management of Data (SIGMOD'99)*, pages 49-60, 1999.
- T. Masters. *Neural, Novel & Hybrid Algorithms for Time Series Prediction*. John Wiley and Sons, 1995.
- B. Matichuk and O. R. Zaiane. Unsupervised classification of sound for multimedia indexing. In *First Intl. Workshop on Multimedia Data Mining (MDM/KDD'2000)*, pages 31-36, Boston, MA, August 2000.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. 1994 Int. Conf. Very Large Data Bases*, pages 144-155, Santiago, Chile, September 1994.
- E. Pauwels, P. Fiddelaers, and L. V. Gool. DOG-based unsupervized clustering for CBIR. *Second International Conference on Visual Information Systems*, pages 13-20, 1997.
- G. Piatetsky-Shapiro, U. Fayyad, and P. Smith. Data mining to knowledge discovery: An overview. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1-35. AAAI/MIT Press, 1996.
- K. S. S. Guha, R. Rastogi. CURE: An ecient clustering algorithm for large databases. In *SIGMOD'98*, Seattle, Washington, 1998.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: a multi-resolution clustering approach for very large spatial databases. In *24th VLDB Conference*, New York, USA, 1998.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. A wavelet-based clustering approach for spatial data very large databases. *The International Journal Very Large Databases*, 8(4):289-304, February 2000.
- W.Wang, J.Yang, and R.Muntz. Sting: A statistical information grid approach to spatial data mining. *Proc. 1997 Int. Conf. Very Large Data Bases (VLDB'97)*, pages 186-195, 1997.
- O. Zamir and O. Etzioni. Web document clustering: feasibility demonstration. In *Proc. ACM SIGIR* 1998.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *1996 ACM SIGKDD Int. Conf. Managament of Data*, pages 103-114, June 1996.
- Z.X. Huang. Clustering Large Data Sets with Mixed Numeric and Categorical Values. In *Porc. Of the First Pacifi-Asia Conference on Knowledge-Discovery and DataMining*, 1997

- L.Kaufman and P.J.Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis.* John Wiley \& Sons, 1990.
- Raymond T. Ng and Jiawei Han. Efficient and Effective Clustering Methods for Spatial Data Mining. *VLDB 1994*: 144-155.
- Sudipto Guha, Rajeev Rastogi, Kyuseok Shim: CURE: An Efficient Clustering Algorithm for Large Databases. *SIGMOD 1998*: 73-84.
- R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *SIGMOD 1998*: 94-105.
- W. Wang, Yang, R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data Mining. *VLDB 1997*.
- Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. *Proc. of the Int. Conf. on Pattern Recognition*. 1996:101-105. CRPC-TR93358
- T. Kohenen. Self-Organizing Maps. *Springer-Verlag*, Berlin, Germany. 1995.
- D. Fisher. Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning.* 1987. 2:139--172.
- P. Cheeseman, J. Stutz. Bayesian Classification (AutoClass): Theory and Results. *Advances in Knowledge Discovery and Data Mining*. 1996.
- J.D. Banfield and A.F. Raftery. Model-based Gaussian and non Gaussian Clustering. *Biometrics* 49, pp 803-821.
- J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. Web page categorization and feature selection using association rule and principal component clustering. In 7th Workshop on Information Technologies and Systems, Dec 1997.
- AUTOCLUST+: Automatic Clustering of Point-Data Sets in the Presence of Obstacles,
  V. Estivill-Castro and I. Lee, Proceedings of the International Workshop on Spatal, Temporal and Spatio-Temporal Data Mining, Springer-Verlag, 131 - 144, 2000.
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, 32(8):68-75, August 1999.
- A. K. H. Tung, J. Hou, and J. Han. Spatial Clustering in the Presence of Obstacles. *Proc. 2001 Int. Conf. on Data Engineering (ICDE'01)*, Heidelberg, Germany, April 2001.
- C. Faloutsos and K. Lin. Fastmap: a fast algorithm for indexing, datamining and visualisation of traditional and multimedia datasets. In *Proc. of ACM-SIGMOD, pp 163-174, 1995.*
- P. Smyth. Clustering using monte-carlo cross-validation. In *KDD-96*, pages 126--133, 1996.