


Table of contents

1. Introduction (Text:10.1)

2. Clustering: Basic Concepts

- 2.1. Quality of Clustering
- 2.2. Considerations
- 2.3. Major Approaches

3. Partitioning Methods (K-means) (Text:10.2)

- 3.1. Strength and Weakness
- 3.2. K-Medoids (PAM)
- 3.3. Exercise 

4. Hierarchical Clustering (AGNES and DIANA) (Text: 10.3)

- 4.1. Distance between Clusters
- 4.2. Dendrogram

5. Practical Exercises: K-means and Hierarchical Clustering

6. Density-Based Methods (DBSCAN) (Text: 10.4)

7. Practical Exercises: DBSCAN and K-means

8. Grid-Based Approach (Text 10.5)

9. Evaluation of Clustering (Text 10.6)

- 9.1. Assessing Clustering Tendency
- 9.2. Determine the Number of Clusters, k
- 9.3. Measure Clustering Quality

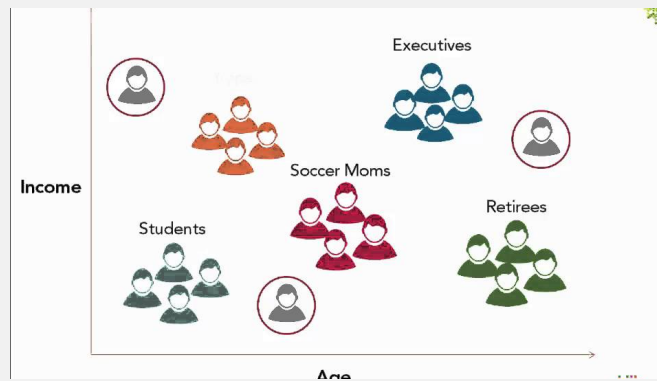
10. Quiz

1. Introduction (Text:10.1)

Most of this material is derived from the text, Han, Kamber and Pei, Chapter 10, or the corresponding powerpoint slides made available by the publisher. Where a source other than the text or its slides was used for the material, attribution is given. Unless otherwise stated, images are copyright of the publisher, Elsevier.

Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters. Clustering is usually used to understand the structure of a dataset, to inform more in-depth analysis and understanding later.

ACTION: Check out this video if you would like to see an example of how clustering might be applied to solve the problem of customer segmentation.



ACTION: You can watch this video lecture overview of the clustering topic if you find it helpful but all it covers is also in the written notes.



[Prerecorded lecture on Clustering](#)

2. Clustering: Basic Concepts

What is Cluster Analysis?

- Cluster: A collection of data objects
 - similar (or related) to one another within the same group
 - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or clustering, data segmentation, ...)
 - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters; discovering groups within the data
- **Unsupervised learning**: no predefined classes
- Typical applications
 - As a **stand-alone tool** to get insight into data distribution
 - As a **preprocessing step** for other algorithms

Clustering as a Preprocessing Tool

- Summarisation:
 - Preprocessing for regression, principal components analysis, classification, and association analysis
- Compression:
 - Image processing: vector quantisation
- Finding K-nearest Neighbours
 - Localising search to one or a small number of clusters
- [Outlier detection](#)
 - Outliers are often viewed as those “far away” from any cluster

Clustering for Data Understanding and Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

2.1. Quality of Clustering

What is Good Clustering?

- A good clustering method will produce high quality clusters
 - high intra-class similarity: cohesive within clusters
 - low inter-class similarity: distinctive between clusters
- The quality of a clustering method depends on
 - the similarity measure used by the method
 - its implementation, and
 - Its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric
 - Similarity is expressed in terms of a **distance function**, typically metric: $d(i, j)$
 - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
 - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
 - There is usually a separate "quality" function that measures the "goodness" of a cluster.
 - It is hard to define "similar enough" or "good enough"
 - The answer is typically highly subjective

2.2. Considerations

Algorithmic Considerations

- Partitioning criteria
 - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
 - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
 - Distance-based (e.g., Euclidean, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
 - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- Scalability
 - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
 - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
 - User may give inputs on constraints
 - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
 - Discovery of clusters with arbitrary shape
 - Ability to deal with noisy data
 - Incremental clustering and insensitivity to input order
 - High dimensionality

2.3. Major Approaches

Based on different approaches we can categorise known clustering algorithms into:

- Partitioning approach:
 - Construct various partitions and then evaluate them by some criterion, e.g., minimising the sum of square errors
 - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
 - Create a hierarchical decomposition of the set of data (or objects) using some criterion
 - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
 - Based on connectivity and density functions
 - Typical methods: DBSCAN (Density-based spatial clustering of applications with noise), OPTICS, DenClue
- Grid-based approach:
 - based on a multiple-level granularity structure
 - Typical methods: STING, WaveCluster, CLIQUE
- Model-based:
 - A model is hypothesised for each of the clusters and tries to find the best fit of that model to each other
 - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
 - Based on the analysis of frequent patterns
 - Typical methods: p-Cluster
- User-guided or constraint-based:
 - Clustering by considering user-specified or application-specific constraints
 - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
 - Objects are often linked together in various ways
 - Massive links can be used to cluster objects: SimRank, LinkClus

We will discuss some of major approaches in detail in the following.

3. Partitioning Methods (K-means) (Text:10.2)

Partitioning method

Partitioning a database D of n objects into a set of k clusters. The quality of cluster C_i can be measured by the within-cluster variation, which is the sum of squared distances between all objects in C_i and the centroid c_i , defined as:

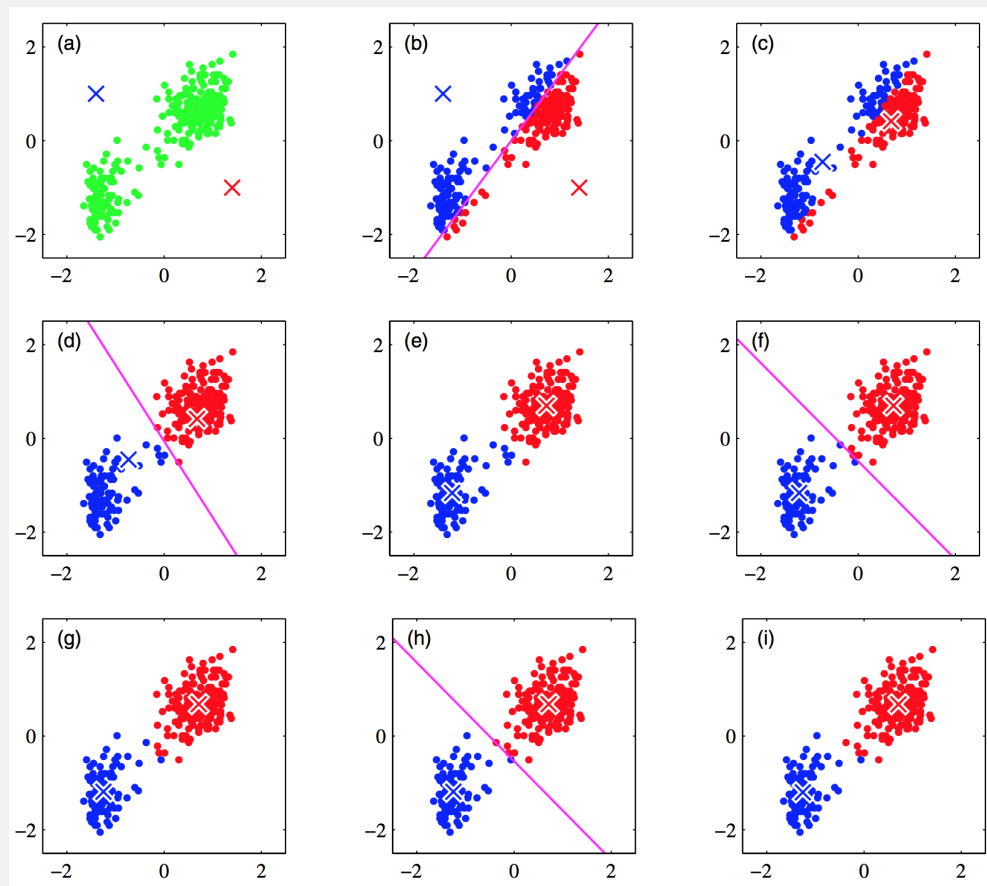
$$E = \sum_{i=1}^k \sum_{p \in C_i} (p - c_i)^2$$

- Given k , find a partition of k clusters that optimises the chosen partitioning criterion
 - Globally optimal: exhaustively enumerate all partitions
 - Heuristic methods: k-means and k-medoids algorithms
 - k-means: Each cluster is represented by the centre of the cluster
 - k-medoids or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster

K-means

- Given k , the k-means algorithm is implemented in four steps:
 1. Arbitrarily choose a centre of k clusters as the initial cluster centres
 2. Assign each object to the cluster to which the object is the most similar
 3. Update the cluster means, that is, calculate the mean value of the objects for each cluster
 4. Go back to Step 2, stop when the assignment does not change

Illustration of the K-means algorithm (from Pattern Recognition & Machine Learning, Bishop).



- (a) Green points denote the data set in a two-dimensional Euclidean space. The initial choices for centres for C_1 and C_2 are shown by the red and blue crosses, respectively.
- (b) Each data point is assigned either to the red cluster or to the blue cluster, according to which cluster centre is nearer. This is equivalent to classifying the points according to which side of the perpendicular bisector of the two cluster centres, shown by the magenta line, they lie on.
- (c) In the subsequent step, each cluster centre is re-computed to be the mean of the points assigned to the corresponding cluster.
- (d)–(i) show successive steps through to final convergence of the algorithm.

3.1. Strength and Weakness

- **Strength:**

- Computationally Efficient: time complexity is $O(tkn)$, where n is the number of objects, k is the number of clusters, and t is the number of iterations. Normally, k, t .
 - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$

- **Weakness**

- Need to specify k , the number of clusters, in advance
- Applicable only to objects in a continuous n -dimensional space
 - k-modes variant method for categorical data: replaces the mean value by the *mode* of a nominal attribute.
 - In comparison, k-medoids can be applied to a wide range of data
- Sensitive to noisy data and outliers
- Non deterministic algorithm. The final result depends on the first initialisation.
- Often terminates at a local optimum, rather than a global optimum.
- Not suitable for clusters with non-convex shapes

Noisy data point example

Consider six points in 1-D space having the values 1,2,3,8,9,10, and 25, respectively. Intuitively, by visual inspection we may imagine the points partitioned into the clusters {1,2,3} and {8,9,10}, **where point 25 is excluded** because it appears to be an **outlier**. How would k-means partition the values? If we apply k-means using $k = 2$,

- Case 1: partition values into {{1, 2, 3}, {8, 9, 10, 25}}. Within-cluster variation is $(1-2)^2 + (2-2)^2 + (3-2)^2 + (8-13)^2 + (9-13)^2 + (10-13)^2 + (25-13)^2 = 196$, given that the mean of cluster {1,2,3} is 2 and the mean of {8,9,10,25} is 13.
- Case 2: partition values into {{1, 2, 3, 8}, {9, 10, 25}}. Within-cluster variation is $(1-3.5)^2 + (2-3.5)^2 + (3-3.5)^2 + (8-3.5)^2 + (9-14.67)^2 + (10-14.67)^2 + (25-14.67)^2 = 189.67$, given that 3.5 is the mean of cluster {1, 2, 3, 8} and 14.67 is the mean of cluster {9, 10, 25}.

The latter partitioning has the lowest within-cluster variation; therefore, the k-means method assigns the value 8 to a cluster different from that containing 9 and 10 due to the outlier point 25. Moreover, the centre of the second cluster, 14.67, is substantially far from all the members in the cluster.

3.2. K-Medoids (PAM)

"How can we modify the k-means algorithm to diminish sensitivity to outliers?"

K-medoids

Instead of taking the mean value of the object in a cluster as a reference point, we can pick actual objects to represent the clusters.

The k-medoids method is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean.

Partitioning Around Medoids (PAM)

PAM algorithm is a popular realisation of k-medoids clustering.

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.

The quality of clustering can be measured by an absolute-error criterion (total cost):

$$E = \sum_{i=1}^k \sum_{p \in C_i} \text{dist}(p, o_i),$$

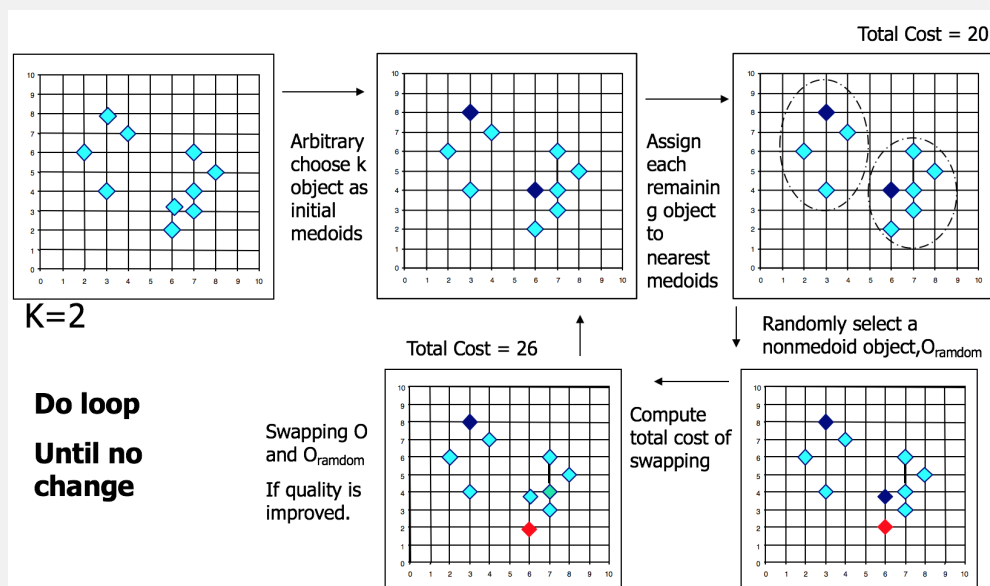
where E is the sum of the absolute error for all objects p in the dataset, and o_i is the representative object of C_i .

PAM works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)

Algorithm

1. Arbitrarily choose k objects in D as the initial representative objects
2. Assign each remaining object to the cluster with the nearest representative object
3. Randomly select a non-representative object, O_{random}
4. For each representative object O_j ,
 1. Compute the total cost of swapping representative object, O_j , with O_{random}
 2. If the swapping reduces the total cost, then swap O_j with O_{random} to form the new set of k representative objects
5. Repeat 2-4 until there is no change.

Illustration of k-medoids algorithm



3.3. Exercise

ACTION: Try out these exercises

 [Exercise: K-means clustering](#)

When you have had a go, you can check your answers against these worked answers:

 [Solution to Exercise: K-means clustering](#)

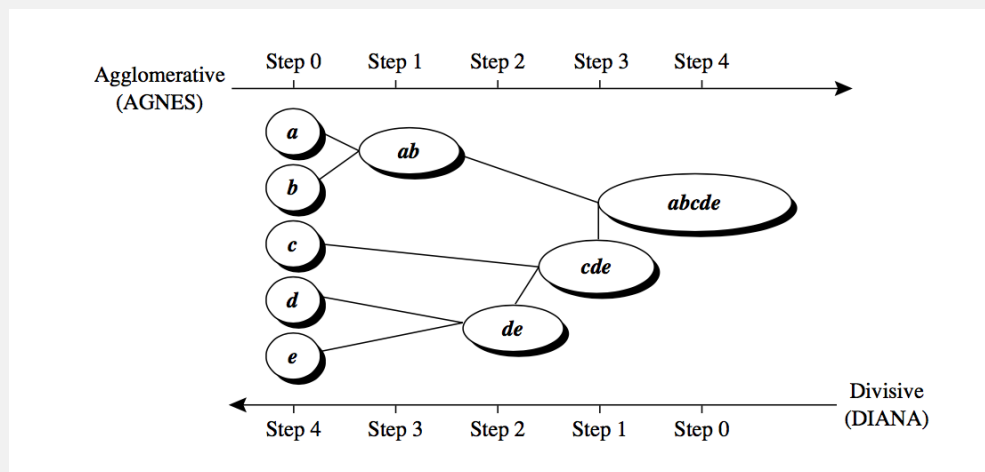
4. Hierarchical Clustering (AGNES and DIANA) (Text: 10.3)

Hierarchical clustering is a method of cluster analysis which seeks to **build a hierarchy of clusters**. Strategies for hierarchical clustering generally fall into two types:

- **Agglomerative**: This is a "bottom up" approach: each observation starts in its own cluster, and a pair of clusters is merged in each step of moving up the hierarchy.
- **Divisive**: This is a "top down" approach: all observations start in one cluster, and a cluster is split into two at each step of moving down the hierarchy.

In general, the merges and splits are determined in a greedy manner. The results of hierarchical clustering are usually presented in a [dendrogram](#).

Here is an example of the agglomerative and divisive hierarchical clustering approaches on data objects $\{a, b, c, d, e\}$.



Initially, the agglomerative method places each object into a cluster of its own. The clusters are then merged step-by-step according to some criterion. The merging process is repeated until all the objects are eventually merged to one cluster.

The divisive method proceeds in the opposite direction. All the objects are used to form one initial cluster. The cluster is split according to some principle. The splitting process repeats until each new cluster contains only a single object.

AGNES (AGglomerative NESTing)

- Uses the single-link method for determining the distance (dissimilarity) between clusters. Other methods can instead be applied, see [Distance between clusters](#), together with a [dissimilarity matrix](#)
- Merges nodes that have the least dissimilarity
- Go on until all nodes are in the same cluster

DIANA (DIvisive ANALysis)

- Inverse order of AGNES
- Go on until each distinct data object forms its own cluster

4.1. Distance between Clusters

Whether using an agglomerative method or a divisive method, a core need is to measure the distance between two clusters, C_i, C_j where each cluster is a set of objects.

- **Single link (minimum distance, nearest-neighbour clustering)**: smallest distance between an element in one cluster and an element in the other
 - i.e., $dist(C_i, C_j) = \min_{p \in C_i, q \in C_j} (|p - q|)$
- **Complete link (maximum distance)**: largest distance between an element in one cluster and an element in the other
 - i.e., $dist(C_i, C_j) = \max_{p \in C_i, q \in C_j} (|p - q|)$
- **Average (average distance)**: average distance between an element in one cluster and an element in the other
 - i.e., $dist(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} (|p - q|)$
- **Centroid**: distance between the centroids of two clusters
 - i.e., $dist(C_i, C_j) = (|c_i - c_j|)$
- **Medoid**: distance between the medoids of two clusters
 - i.e., $dist(C_i, C_j) = (|o_i - o_j|)$

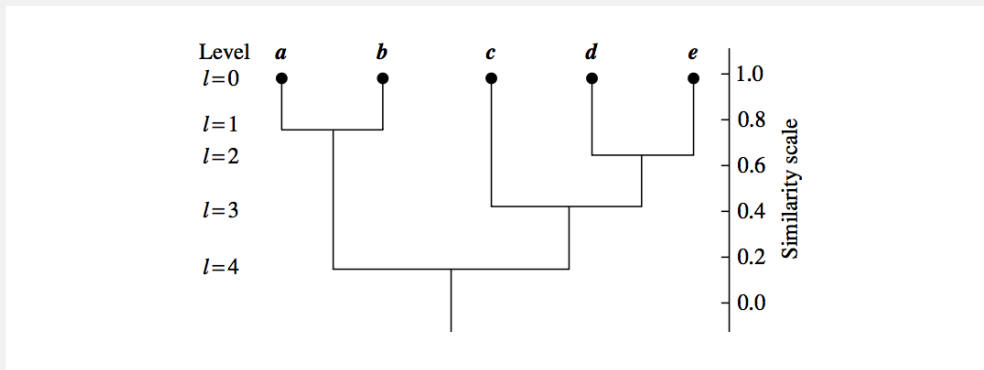
4.2. Dendrogram

A binary-tree-structured diagram, called a dendrogram, is commonly used to represent the process of hierarchical clustering. It shows how objects are grouped together (in an agglomerative method) or partitioned (in a divisive method) step-by-step. The similarity of the cluster pairs selected at the step of their agglomeration or division may be shown on a similarity scale.

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

The desired level is usually determined by selecting a threshold for similarity amongst clusters, but the desired number of clusters could be a factor too.

Here's an example of a dendrogram on data objects $\{a,b,c,d,e\}$



For example, by setting the similarity threshold to 0.5, one can obtain 3 clusters (a,b) , (c) , (d,e) from the dendrogram.

5. Practical Exercises: K-means and Hierarchical Clustering

ACTION: Have a go yourself with these R exercises! There is a video showing the mechanics to get you started, written instructions for you to work through, and separately some suggested solutions.

COMP3425/8410 K-Means Clustering in R



[Exercise: K-means and Hierarchical Clustering with R](#)



[Solution to Exercise: K-means and Hierarchical Clustering with R](#)

6. Density-Based Methods (DBSCAN) (Text: 10.4)

Density-Based Clustering

Model clusters as dense regions in the data space, separated by sparse regions. Does not attempt to assign every object to a cluster; many may be left out as "noise".

- Major features:
 - Discovers clusters of arbitrary shape
 - Partitioning and hierarchical methods are designed to find spherical-shaped (convex) clusters
 - Handles noise
 - One scan through the data only
 - Needs parameters to define threshold dense-ness (but not for the number of clusters)

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

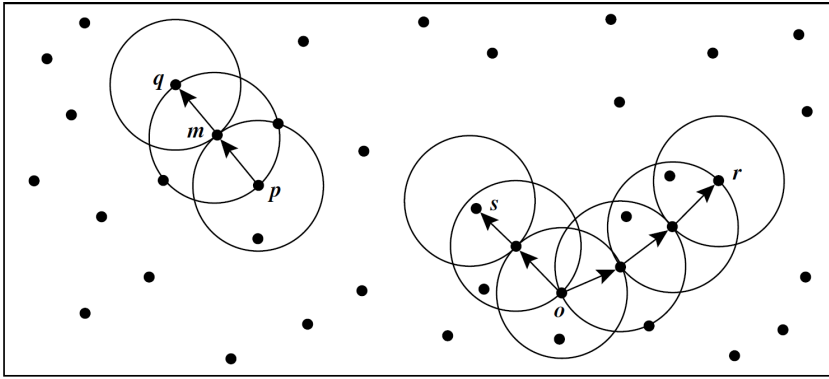
- **Density** of an object o : the number of objects close to o
- **Core** objects: Objects that have a dense neighbourhood
- DBSCAN: connects core objects and their neighbourhoods to form dense regions as cluster
- Two parameters:
 - ϵ : Maximum radius of the neighbourhood
 - *MinPts*: Minimum number of points in an ϵ -neighbourhood of that point
- $N_\epsilon(p)$: $\{q \in D \mid \text{dist}(p, q) \leq \epsilon\}$
 - Number of neighbourhood objects including p
 - If $N_\epsilon(p) \geq \text{MinPts}$, then p is core object
 - D is a data set.
- **Directly density-reachable**: A point p is directly density-reachable from a **core** point q if p is within the ϵ -neighbourhood of q
 - By definition, no points are *directly density-reachable* from a non-core point.
- **Density-reachable**: p is density-reachable from a **core** point q if there is a chain of objects p_1, p_2, \dots, p_n such that $p_1 = q$, $p_n = p$ and p_{i+1} is *directly density-reachable* from p_i with respect to ϵ and *MinPts*.
 - By definition, all the p_i s other than $p_n = p$ are core points
- **Density-connected**: Two objects p_1, p_2 are density-connected if
 - there is an object q such that both p_1 and p_2 are *density-reachable* from q with respect to ϵ and *MinPts*.
 - By definition, q must be a core point, and p_1 and p_2 must be in the neighbourhood of a core point, but may not be core points themselves.

Definition of Cluster in DBSCAN

A subset $C \subseteq D$ is a cluster if

- All points within the cluster C are **mutually density-connected**, and
- There is no point outside C that is **density-connected** to a point inside C .

Example of density-reachable and density-connected:



> Let ϵ be the radius of the circles and *MinPts* 3.

> m, p, o, r are core objects.

> Object q is directly density-reachable from m .

> Object m is directly density-reachable from p and vice versa.

> Object q is density-reachable from p because q is directly density reachable from m and m is directly density-reachable from p . However, p is not density reachable from q because q is not a core object.

> r and s are density-reachable from o

> o is density-reachable from r .

> o, r , and s are all density-connected.

DBSCAN algorithm

Algorithm: DBSCAN: a density-based clustering algorithm.

Input:

- D : a data set containing n objects,
- ϵ : the radius parameter, and
- $MinPts$: the neighborhood density threshold.

Output: A set of density-based clusters.

Method:

```
(1)  mark all objects as unvisited;  
(2)  do  
(3)      randomly select an unvisited object  $p$ ;  
(4)      mark  $p$  as visited;  
(5)      if the  $\epsilon$ -neighborhood of  $p$  has at least  $MinPts$  objects  
(6)          create a new cluster  $C$ , and add  $p$  to  $C$ ;  
(7)          let  $N$  be the set of objects in the  $\epsilon$ -neighborhood of  $p$ ;  
(8)          for each point  $p'$  in  $N$   
(9)              if  $p'$  is unvisited  
(10)                  mark  $p'$  as visited;  
(11)                  if the  $\epsilon$ -neighborhood of  $p'$  has at least  $MinPts$  points,  
                      add those points to  $N$ ;  
(12)                  if  $p'$  is not yet a member of any cluster, add  $p'$  to  $C$ ;  
(13)          end for  
(14)      output  $C$ ;  
(15)  else mark  $p$  as noise;  
(16) until no object is unvisited;
```

ACTION: Watch this video that shows how the DBSCAN algorithm works through to build the clusters in the diagram above.



[DBSCAN worked example](#)

7. Practical Exercises: DBSCAN and K-means

ACTION: Here are some practical experiments that look at differences between k-means and DBSCAN. There is a video showing the mechanics to get you started, written instructions for you to work through, and separately some suggested solutions.

COMP3425/8410 DBSCAN Clustering in R



 [Exercise: DBSCAN and K-means](#)

 [Solution to Exercise: DBSCAN and K-means](#)

8. Grid-Based Approach (Text 10.5)

Grid-based clustering

Space driven approach by partitioning the input space into cells.

Multiresolution grid data structure approach is used: quantize the object space into a finite number of cells that forms a grid structure.

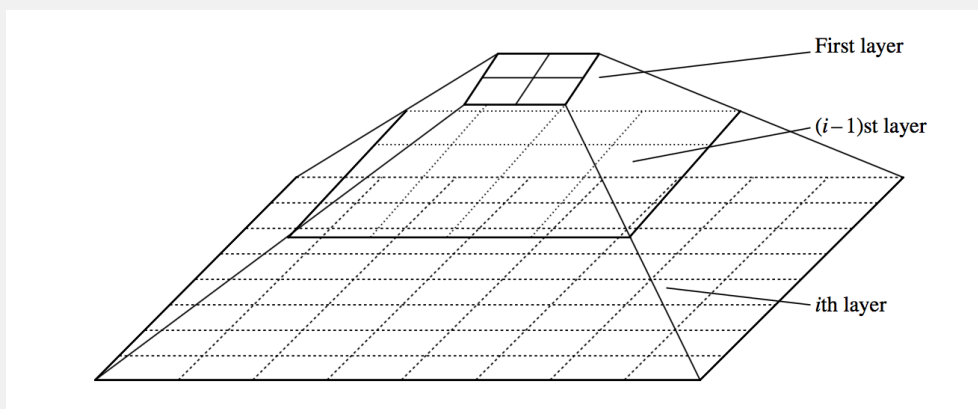
Fast processing time which is independent of the number of data objects.

Possible clusters are predefined by the design of the grid and therefore the defined allocation of objects to grid cells; the problem becomes to retrieve the clusters that satisfy a query that includes statistical properties of desirable clusters.

STING: STatistical INformation Grid

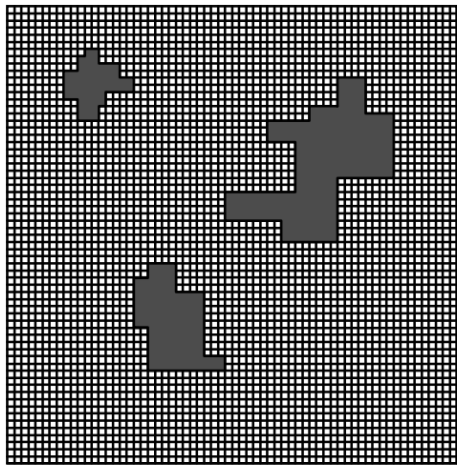
In STING, the input space is divided in a hierarchical way

- At the first layer, the input space is divided into some rectangular cells.
- Each cell at a high level is partitioned to form a number of cells at the next level.



> Hierarchical structure for STING clustering.

- Precomputed statistical parameters:
 - Statistical informations such as count, mean, maximum, and minimum, of each grid cell is precomputed and stored.
 - Statistics of the bottom level cells are directly computed from the data.
 - Statistics of a higher level cell can be computed based on lower-level cells
- Answer spatial data queries using precomputed statistics
 - Top down approach:
 - Start from a pre-selected layer - typically with a small number of cells
 - Compute the confidence interval reflecting the cell's relevance to the given query
 - Irrelevant cells are removed
 - Children of remaining cells will be examined
 - Repeat until the bottom layer is reached
 - The regions of relevant cells, which satisfy the query, are returned
 - Query example:
 - Select the maximal regions that have at least 100 houses per unit area and at least 70% of the house prices are above \$400K and with total area at least 100 units with 90% confidence



> Example of clustering result obtained from a query (Wang et al, 1997)

- Advantage
 - Grid-based computation is query-independent
 - Grid structure facilitates parallel processing and incremental updating
 - Computational efficiency: STING goes through the database once to compute the statistical parameters of the cells
- Disadvantage
 - Sensitive to bottom level granularity
 - If the granularity is very fine, the cost of processing will increase substantially
 - if the bottom level of the grid structure is too coarse, it may reduce the quality of cluster analysis
 - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

Q & A

- *I interpret "grid" as a 3D structure, where layers are 2D structures at different levels of the grid, implying that the layer only has information on two features.*

Some predefined number of dimensions are chosen in advance to aggregate over. Let's say 2 (and yes, that makes sense for the spatial data application, which is most usual, but not necessary). Let's say those 2 are "lat" and "long". Then each layer going up up the grid will partition lat and long into bigger and bigger intervals of space. The statistics stored at some cell in some layer correspond to the aggregate for a fixed number of cells in the layer below, where spatial resolution is finer. This is very different to a cuboid where each dimension is either aggregated or not (although it is close to a conceptual hierarchy where the hierarchy is structured in a very particular way). Aggregation happens here over pre-selected *intervals* of pre-selected dimensions only.

- *What exactly separates a layer from the layer directly above or below it?*

A pre-defined design decision: each higher up cell represents (in terms of the statistical data stored) all of its sub-intervals for each dimension at the level below.

- *Exactly what data is on a single layer of the grid?*

The statistical summaries are held in each cell which represents a fixed interval over each of the fixed dimensions selected. The statistical summaries are very particular measures (that include a statistical distribution) and these particular measures are required to give the advantages claimed.

- *How do we decide the partitioning between layers?*

What queries are you trying to optimise? You choose the dimensions; you choose the intervals at each level. 2D spatial dimensions are most common.

9. Evaluation of Clustering (Text 10.6)

"How can I evaluate whether the clustering results are good?"

The major tasks of clustering evaluation include the following:

- *Before you start:* Assess clustering tendency. Assess whether a non-random structure exists in the data. Clustering analysis on a data set is meaningful only when there is a non-random structure in the data.
- *Next:* Determine the number of clusters in a dataset. How many clusters are there to find? Like k-means, many methods require the number of clusters in advance as a parameter to the method.
- *After clustering:* Measure the clustering quality. There are various quality measures according to different criteria.

9.1. Assessing Clustering Tendency

Clustering tendency assessment determines whether a given data set has a non-random structure, which may lead to meaningful clusters.

- Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution
- Test spatial randomness by statistical test: **Hopkins Statistic**
 - Given a dataset D regarded as a sample of a random variable O , determine how far away O is from being uniformly distributed in the data space
 - Sample n points, p_1, \dots, p_n , uniformly from the **range of D** . For each p_i , find its nearest neighbour in $D : x_i = \min \text{dist}(p_i, v)$ where v in D
 - For example, if D consists of real valued observations whose minimum value is 0.5 and maximum value is 6.2, then p_i is a random value sampled uniformly between 0.5 and 6.2.
 - Sample n points, q_1, \dots, q_n , uniformly from $D (q_i \in D)$. For each q_i , find its nearest neighbour in $D - q_i : y_i = \min \text{dist}(q_i, v)$ where v in D and $v \neq q_i$
 - Unlike p_i , q_i is one of the existing values in D (i.e. $q_i \in D$).
 - Calculate the Hopkins Statistic:

$$H = \frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n x_i + \sum_{i=1}^n y_i}$$

- ◦ ▪ If D is uniformly distributed, $\sum x_i$ and $\sum y_i$ will be close to each other and H is close to 0.5.
- ◦ ▪ If D is highly skewed, H is close to 0
- ◦ If D is uniformly distributed then it contains no meaningful clusters.

9.2. Determine the Number of Clusters, k

A few general-purpose methods are given here, and each can be varied according to the clustering method, the cluster quality heuristic or domain knowledge. Note that the Elbow and Cross-validation methods require many clustering attempts with different numbers of clusters, so could be prohibitively expensive over the full data set. Consider selecting a random sample of the data for this purpose.

Empirical method

- Try number of clusters $\approx \sqrt{n/2}$ for a dataset of n points. Then each cluster would be expected to have $\sqrt{2n}$ points.

Elbow method

- As the number of clusters goes up, the within-cluster variance, that is the distances amongst points in the cluster (defined as the sum of squared distances between each object and the centroid) decreases to zero. So aim to choose a number that tends to reduce the sum of each within-cluster variance, but increasing the number any further would have only have marginal effect on the variance.
- Use the turning point in the curve of sum of within cluster variance w.r.t the number of clusters.

To implement:

- For many choices of $k > 0$ (in the extreme, $k = 1.., n$), execute the clustering with parameter k and calculate sum of within-cluster variances for that k. Plot each k against its sum. Choose the k corresponding to a notable bend in the curve to be the "right" number of clusters.



elbow method from
<http://www.sthda.com/english/articles/29-cluster-validation-essentials/96-determining-the-optimal-number-of-clusters-3-must-know-methods/>

Figure from www.sthda.com demonstrating the elbow method. In this case $k = 4$ would be a good choice.

Cross validation method

- Divide a given data set into m parts
- Use m – 1 parts to obtain a clustering model.
Use the remaining part to test the quality of the clustering.
 - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance. between all points in the test set and the closest centroids to measure how well the model fits the test set
- For several choices of $k > 0$, repeat it m times and compute the overall quality as the average for each of the m times.
Compare the overall quality measure w.r.t. different k's, and choose the number of clusters that corresponds to the k that has the best overall quality.

9.3. Measure Clustering Quality

- Two methods: extrinsic vs. intrinsic
- **Extrinsic**: supervised, i.e., the **ground truth is available**
 - Compare a clustering against the ground truth using certain clustering quality measure
 - Ex. BCubed precision and recall metrics
- **Intrinsic**: unsupervised, i.e., the **ground truth is unavailable**
 - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
 - e.g. Silhouette coefficient below is objectively quantitative, but subjective judgement can be just as useful.

Extrinsic Methods

To measure clustering quality, we need to define a score function $Q(C, C_g)$ for a clustering C and a ground truth clusters C_g .

Four criteria of a good score function:

- Cluster homogeneity: The more pure the clusters, the better the clustering.
- Cluster completeness: The counterpart of homogeneity. Any two objects belonging to the same category in the ground truth, should be assigned to the same cluster.
- Rag bag: Rag bag category: objects that cannot be merged with other objects. Putting a heterogeneous object into a pure cluster should be penalised more than putting it into a rag bag.
- Small cluster preservation: Splitting a small category into pieces is more harmful than splitting a large category into pieces.

BCubed precision and recall metrics satisfy the all four criteria.

- Let $C(o_i)$ be the cluster number of object o_i , $L(o_i)$ be the category of o_i given by the ground truth, and $cor(o_i, o_j)$ be 1 if $L(o_i) = L(o_j)$ and $C(o_i) = C(o_j)$, otherwise 0.
- **BCubed precision**: how many other objects in the same cluster belong to the same category as the object.

$$p = \left(\sum_{i=1}^n \frac{\sum_{o_j: i \neq j, C(o_i)=C(o_j)} cor(o_i, o_j)}{||\{o_j | i \neq j, C(o_i)=C(o_j)\}||} \right) \times \frac{1}{n}$$

- **BCubed recall**: how many objects of the same category are assigned to the same cluster.

$$r = \left(\sum_{i=1}^n \frac{\sum_{o_j: i \neq j, L(o_i)=L(o_j)} cor(o_i, o_j)}{||\{o_j | i \neq j, L(o_i)=L(o_j)\}||} \right) \times \frac{1}{n}$$

Intrinsic Methods

Two criteria for intrinsic method:

- How well the clusters are separated
- How compact the clusters are

1. Silhouette coefficient satisfies above two conditions.

Let C_i be the i th cluster from a clustering and let o be an object in the cluster C_i .

- *compactness*, $a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o, o')}{|C_i| - 1}$ where $o \in C_i$,
- *separation*, $b(o) = \min_{C_j: 1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|} \right\}$

$a(o)$ reflects the compactness of the cluster C_i , being the average distance of the object o in the cluster from every other object in the cluster. Low compactness is good.

$b(o)$ reflects the degree to which object o is separated from other clusters it does not belong to, being the average distance to all objects in the next-closest cluster. High separation is good.

The silhouette coefficient of o is then defined as:

- $$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}.$$

The value lies between -1 and 1. A negative value means o is closer to the objects in another cluster than to the objects in the same cluster in expectation, and this is normally undesirable.

To evaluate a particular cluster, average the silhouette coefficient for every object in the cluster. To evaluate a clustering, average the silhouette coefficient for every object in the dataset. Negative values are poor.

2. Visual inspection Plot the clusters (or a small random sample of the data instead of the whole dataset) in 2 dimensions (choose several pairs of dimensions for several plots, or choose pairs of dimensions that are expected to be important in the problem domain. You can plot in 3 dimensions if you prefer, but more than that and it gets very hard to inspect visually). Indicate the cluster membership by the colour coding of points on the plot. Do the clusters seem to be well separated and internally compact ?

3. Elbow method If you used the [elbow method](#) to choose an optimal number of clusters, was there a clear turning point in the plot, indicating that there is some inherent structural meaning to the k you chose?

10. Quiz

ACTION: Try the cluster analysis quiz

 [Quiz: Clustering](#)