COMP3420: Advanced Databases and Data Mining

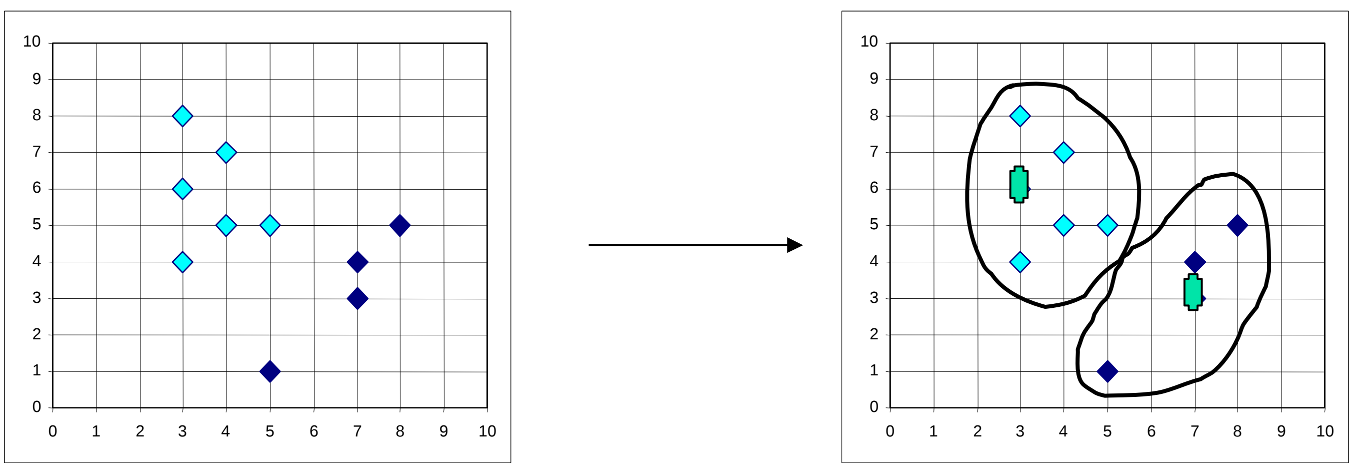
Advanced cluster analysis

Lecture outline

- The problems with *k-means* clustering
- *k-medoids* clustering
- *CLARA* and *CLARANS*
- Hierarchical clustering
  - Dendrograms
- Density-based clustering methods
- Grid-and model-based clustering
  - Self-organising maps (SOM)
- Clustering high-dimensional data
  - The *curse of dimensionality*
- Constraint-based clustering

What is the problem of the *k-means* method?

- The *k-means* algorithm is sensitive to outliers
- Since an object with an extremely large value may substantially distort the distribution of the data
- *K-medoids:* Instead of taking the mean value of the data object in a cluster as a reference point, medoids can be used, which are the most centrally located data objects in a cluster



The *k-medoids* clustering method

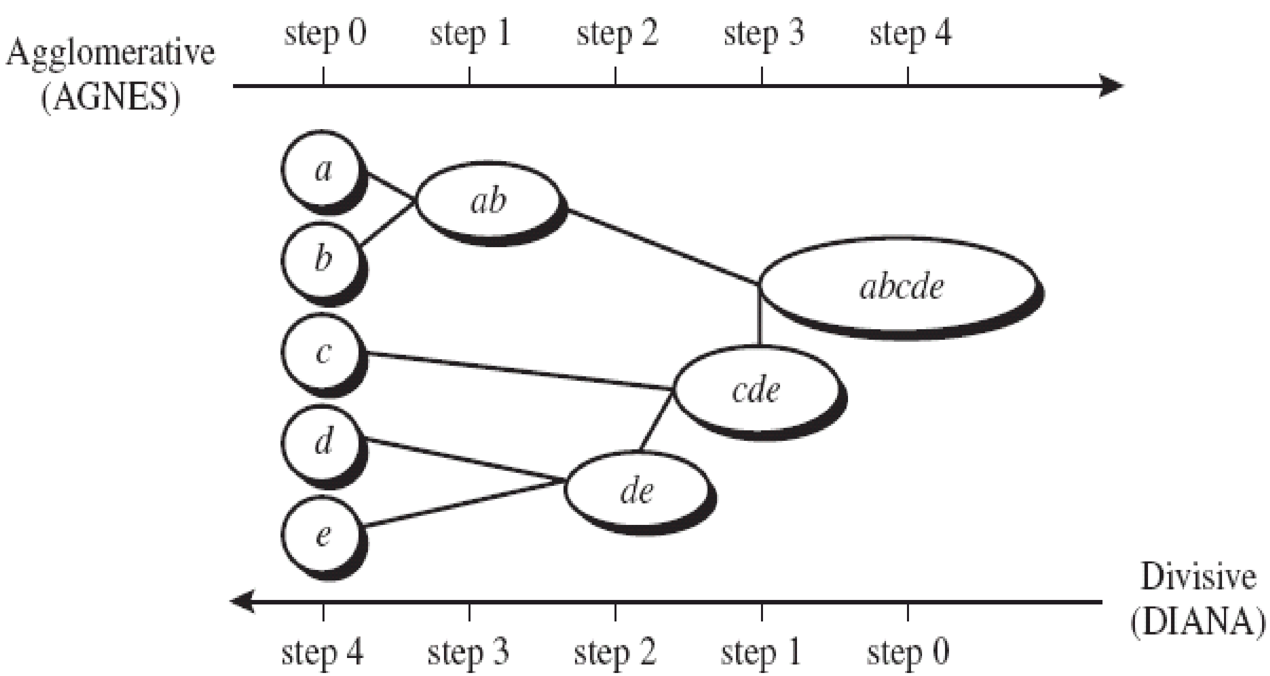
- Find representative data objects, called *medoids*, in clusters
  - *PAM* (Partitioning Around Medoids, 1987)
- Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
- *PAM* works effectively for small data sets, but does not scale well for large data sets
  - Complexity is $O(k(n-k)^2)$ for each iteration, with n the number of data objects and k the number of clusters
- Sampling based methods
  - *CLARA* (Kaufmann & Rousseeuw, 1990)
  - *CLARANS* (Ng & Han, 1994): Randomised sampling

CLARA and *CLARANS*

- CLARA (Clustering LARge Applications)
  - Draws multiple samples of the data set, applies PAM on each sample and gives the best clustering as output
  - Strength: Can deal with larger data sets
  - Weakness: Efficiency depends on the sample size, a good sample based clustering might not necessarily represent a good clustering of the whole data set
- *CLARANS* ('Randomised' *CLARA*)
  - Draws sample of neighbours dynamically
  - Is more efficient and scalable than both *PAM* and *CLARA*
  - Clustering process can be represented as searching a graph, where every node is a potential solution (i.e. a set of k-medoids)
  - If a local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum

Hierarchical clustering

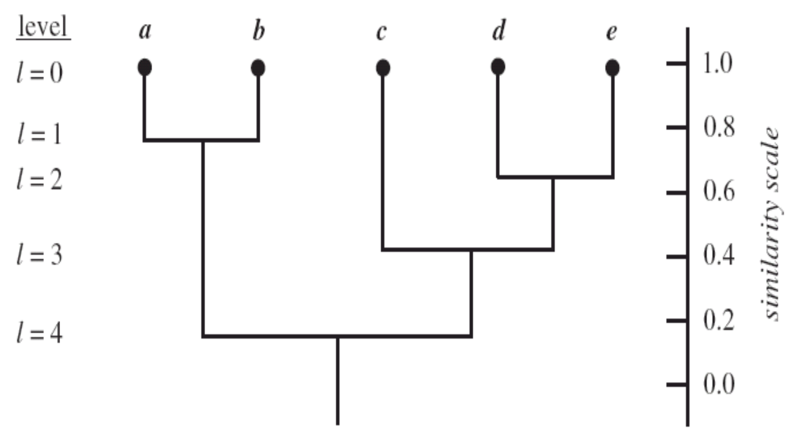
- Use a *distance matrix* as clustering criteria. This method does not require the number of clusters *k* as an input, but needs a termination condition



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Dendrogram

- Shows how the clusters are merged
- Decompose data objects into several levels of nested partitionings (tree of clusters), called a *dendrogram*
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

AGNES and *DIANA*

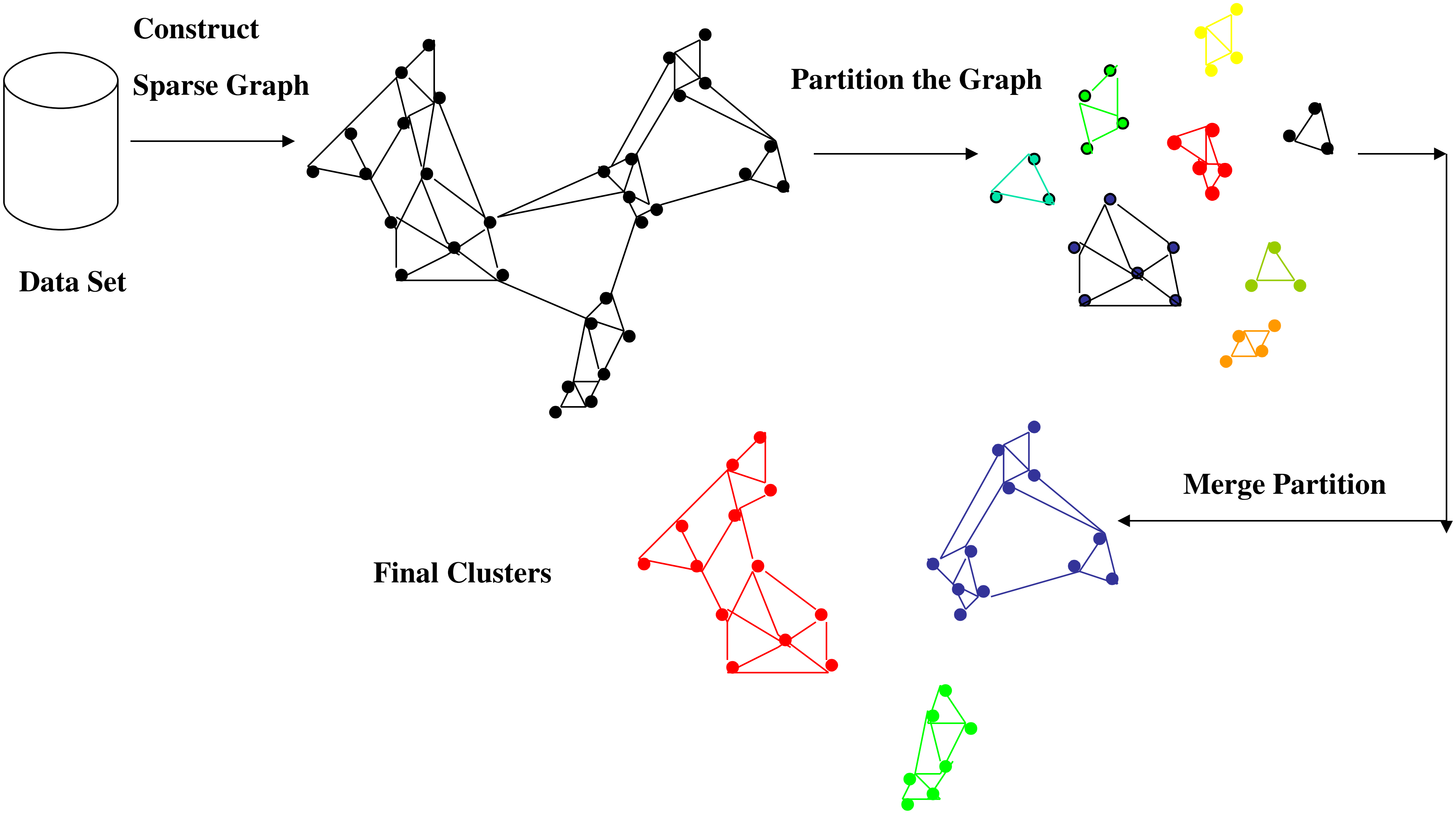
- *AGNES* (AGglomerative NESting)
  - Uses the single-link method and dissimilarity matrix
  - Merges nodes that have the least dissimilarity
  - Go on until all nodes are in the same cluster

- *DIANA* (DIvisive ANAlysis)
  - Inverse order of *AGNES*
  - At the end each data object forms its own cluster

Other hierarchical clustering methods

- Major weakness of agglomerative clustering methods
  - They do not scale well: time complexity of at least $O(n^2)$, where *n* is the number of data objects
  - Can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - *BIRCH (Balanced Iterative Reducing and Clustering)* (1996): uses CF-tree (clustering feature) and incrementally adjusts the quality of sub-clusters, scales linearly with (single data scan), but can handle only numerical data
  - *ROCK (RObust Clustering using linKs)* (1999): clustering categorical data by neighbor and link analysis, uses links to measure similarity/proximity, not distance based, uses sampling
  - *CHAMELEON (Hierarchical Clustering using Dynamic Modeling)* (1999): two clusters are merged if their interconnectivity is high and they are close together, based on a graph partitioning algorithm

CHAMELEON framework



Source: Han and Kamber, DM Book, 2^{nd} Ed. (Copyright © 2006 Elsevier Inc.)

CHAMELEON examples

Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

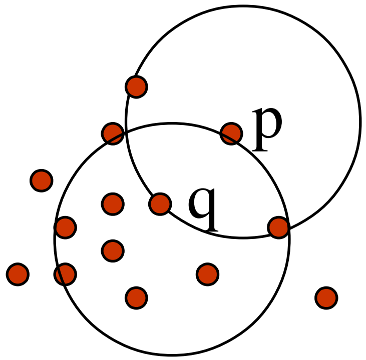
Density-based clustering methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan through data
  - 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) (more grid-based)

Density-based clustering: Basic concepts

- Two parameters:
  - *Eps* (epsilon): Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- $N_{eps}(p)$: $\{q \text{ belongs to } D \mid dist(p,q) <= Eps\}$
- Directly density-reachable: A point *p* is directly density-reachable from a point *q* with respect to *Eps* and *MinPts,* if
  - *p* belongs to $N_{Eps}(q)$
  - Core point condition:

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

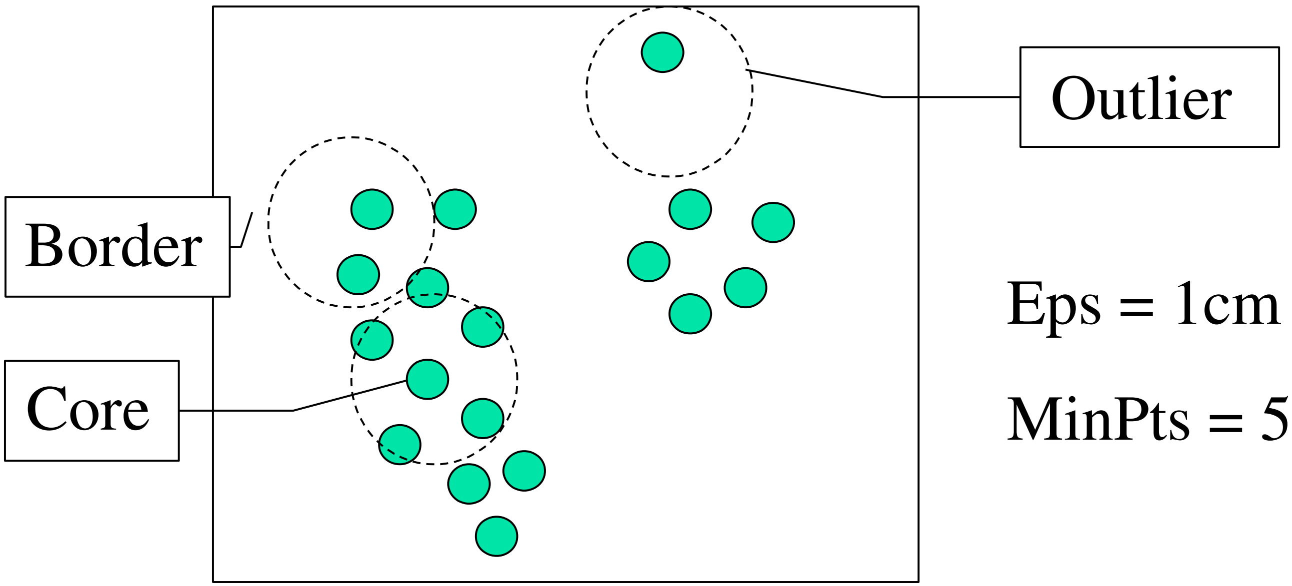


MinPts = 5

Eps = 1 cm

DBSCAN

- Density-Based Spatial Clustering of Applications with Noise
- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximum set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise

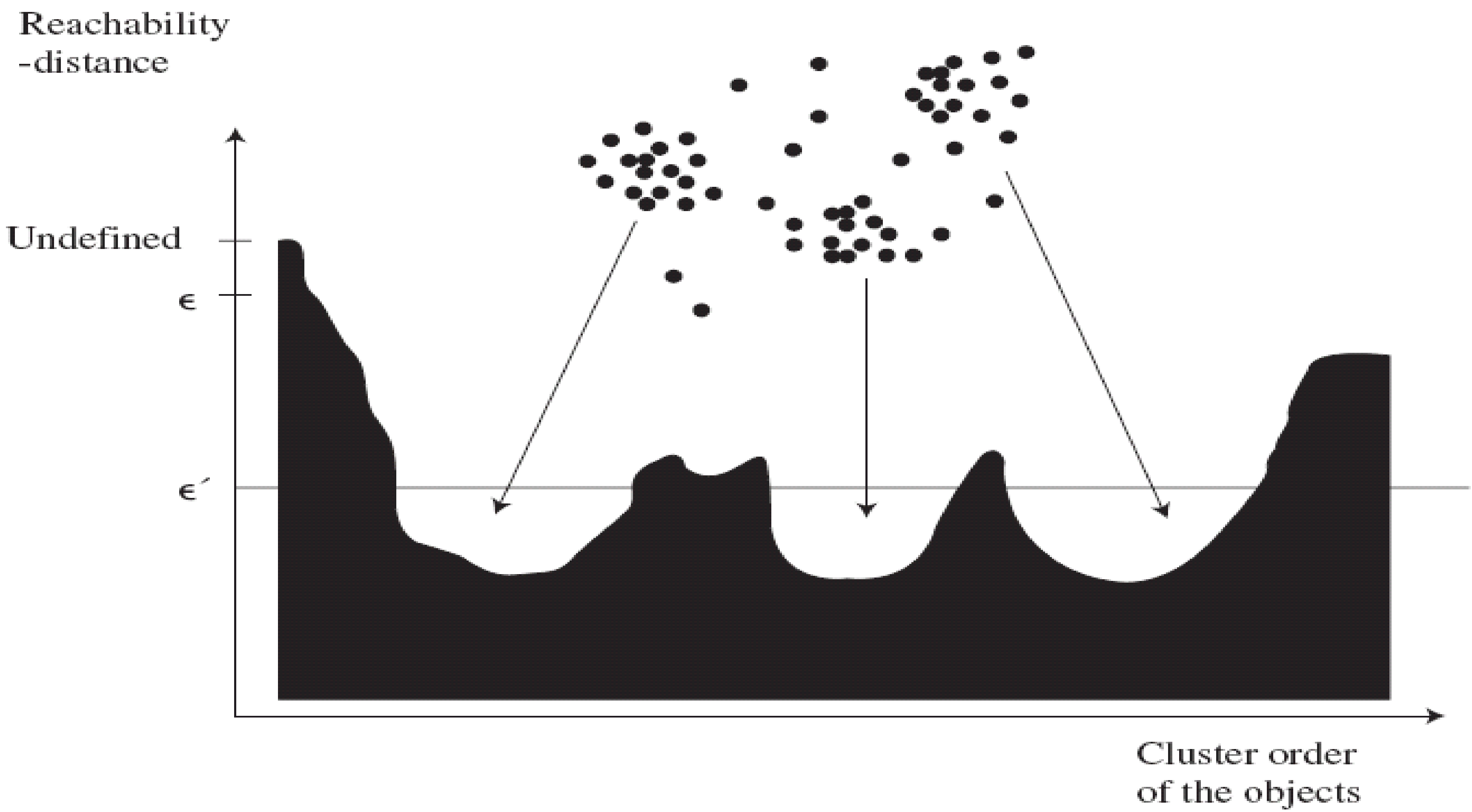


Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

OPTICS

- *O*rdering *P*oints *To* *I*dentify the *C*lustering *S*tructure
- Produces a special order of the database with respect 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 visualisation techniques

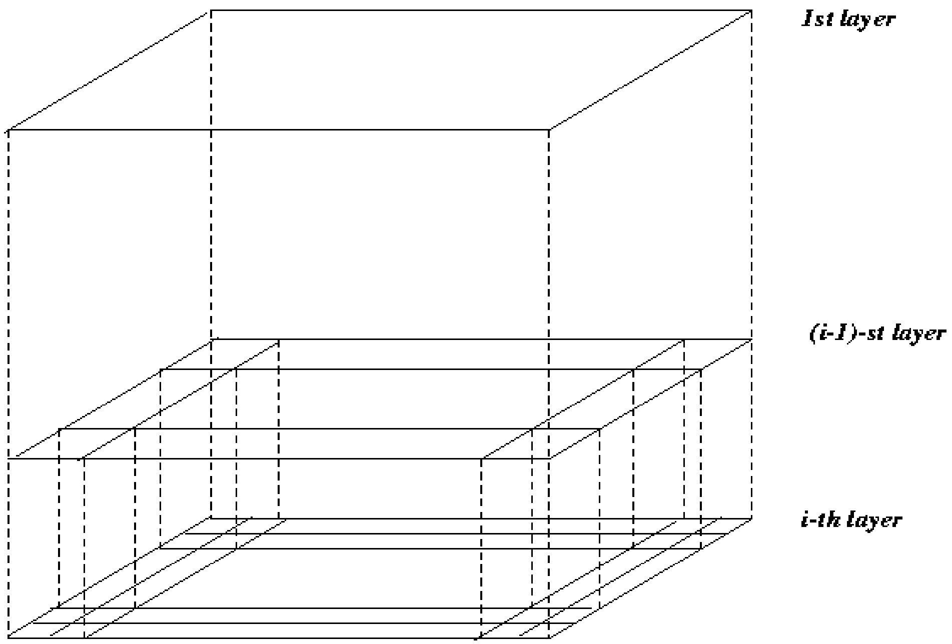
Cluster ordering in *OPTICS*



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Grid-based clustering methods

- Using multi-resolution grid data structure
- Several interesting methods (*STING, WaveCluster, CLIQUE*)



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

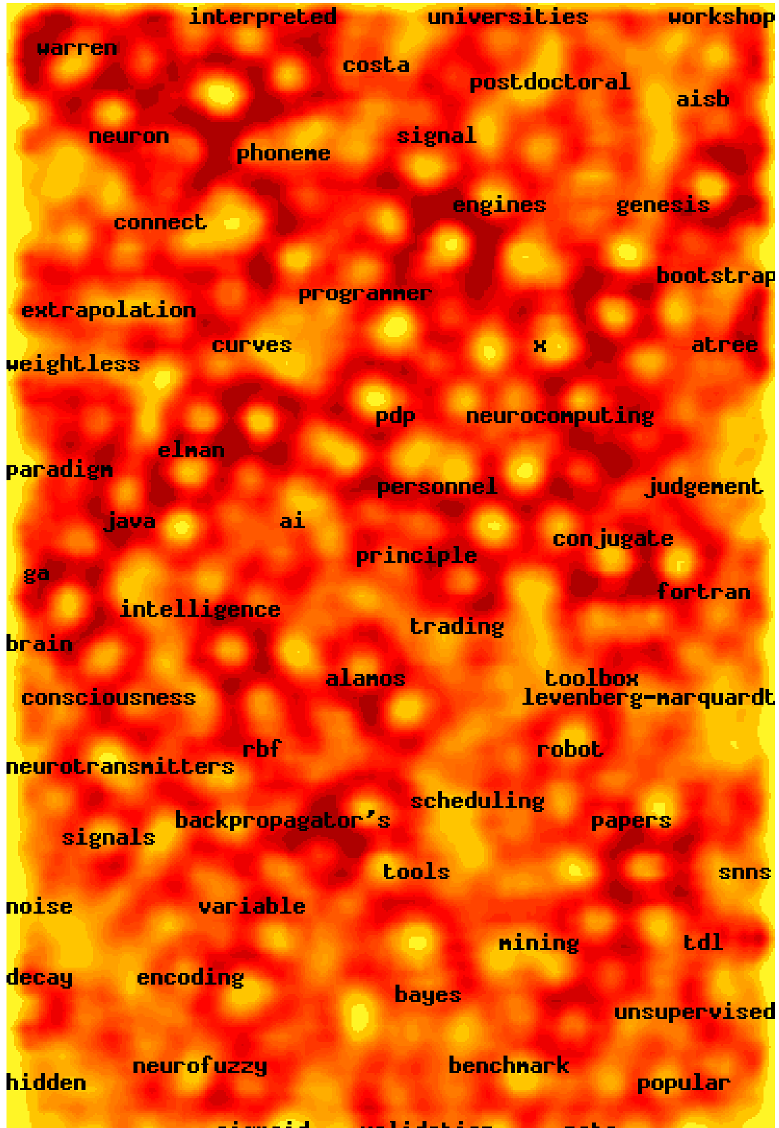
Model-based clustering

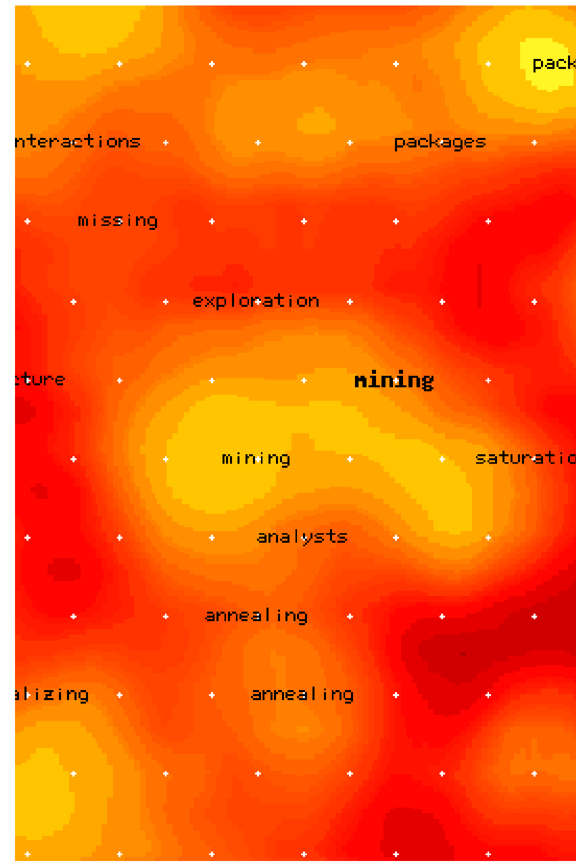
- Attempt to optimise the fit between the given data and some mathematical model
  - Based on the assumption: data are generated by a mixture of underlying probability distributions

- Typical methods
  - Statistical approach: *EM* (Expectation maximisation, a statistical variation of *k-means*), *AutoClass*
  - Machine learning approach: *COBWEB*, *CLASSIT*
  - Neural network approach: *SOM* (Self-Organizing Map), represent each cluster as an exemplar, acting as a “prototype” of the cluster, useful for visualising high-dimensional data in 2- or 3-D space

Web document clustering using *SOM*

- The result of *SOM* clustering of 12,088 Web articles
- The picture on the right: drilling down on the keyword "mining"
- Based on *websom.hut.fi* Web page





Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

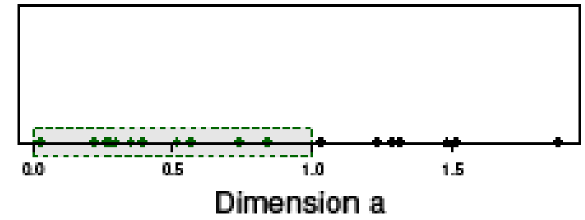
Clustering high-dimensional data

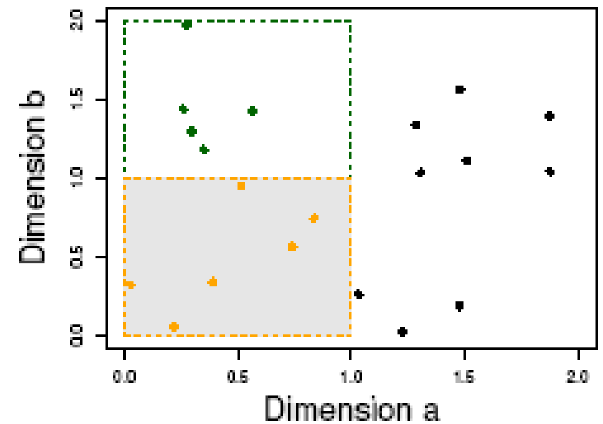
- Many applications: text documents, DNA micro-array data
- Major challenges:
  - Many irrelevant dimensions may mask clusters
  - Distance measure becomes meaningless—due to equi-distance
  - Clusters may exist only in some subspaces
- Methods
  - Feature transformation: only effective if most dimensions are relevant
  - PCA (principal component analysis) & SVD (singular value decomposition) useful only when features are highly correlated/redundant
  - Feature selection: wrapper or filter approaches, useful to find a subspace where the data have nice clusters
  - Subspace-clustering: find clusters in all the possible subspaces (*CLIQUE*, *ProClus*, and frequent pattern-based clustering)

The *curse of dimensionality*

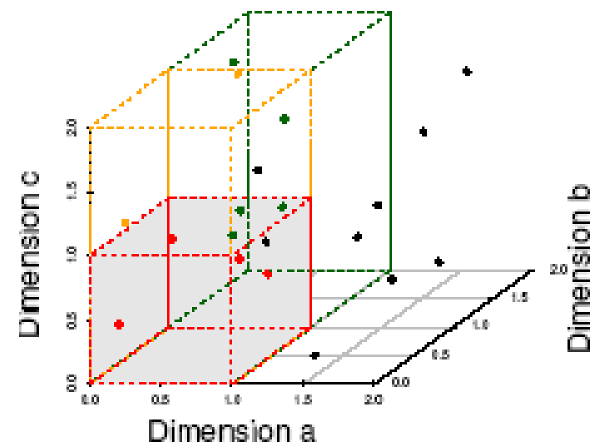
(graphs adapted from Parsons et al. KDD Explorations 2004)

- Data in only one dimension is relatively packed
- Adding a dimension “stretches” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless—due to equi-distance





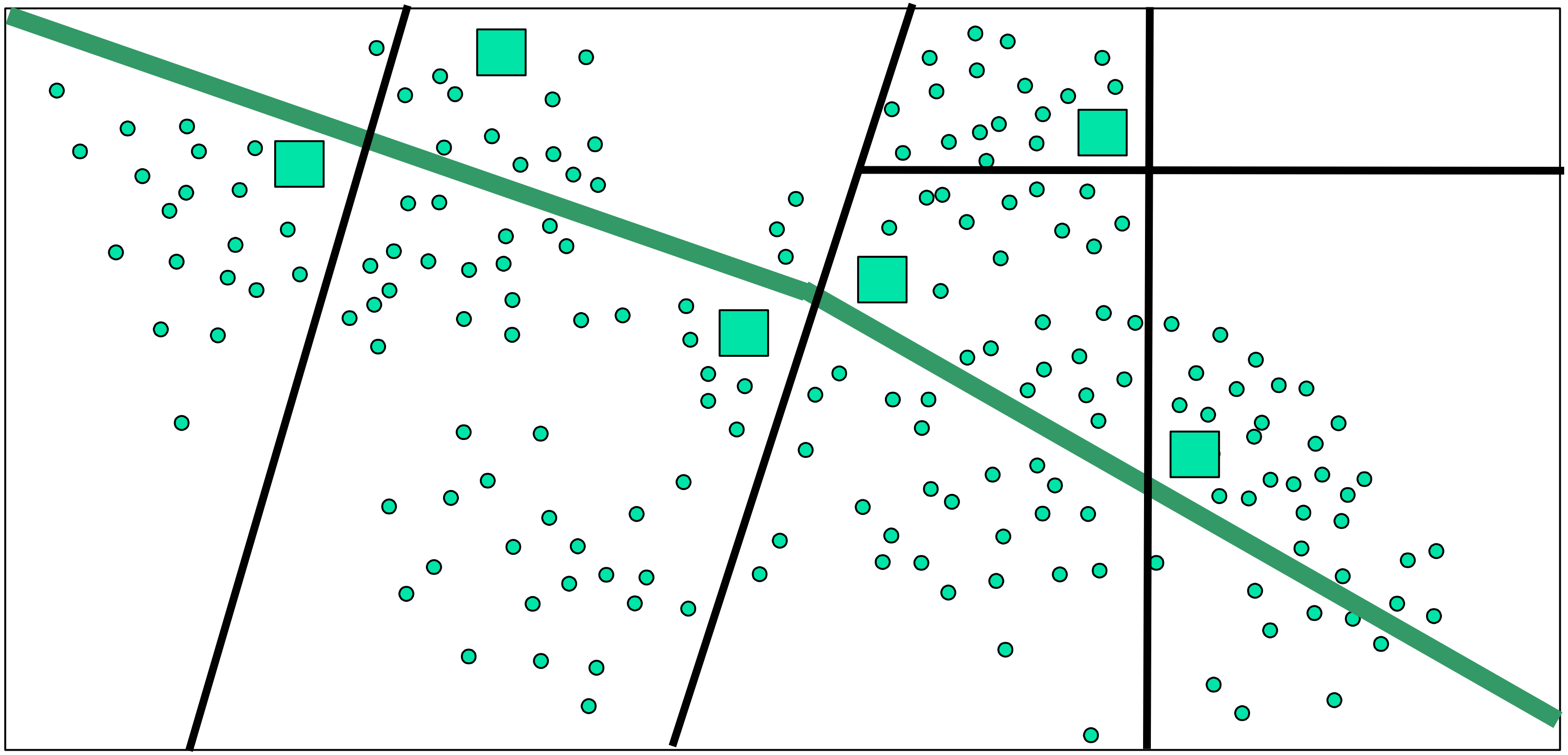
(b) 6 Objects in One Unit Bin



(c) 4 Objects in One Unit Bin

Why constraint-based cluster analysis?

- Need user feedback: Users know their applications best
- Less parameters but more user-desired constraints
  - For example, an ATM allocation problem: obstacle & desired clusters



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Review question

- Even when different distance measures are used, a given clustering algorithm will always produce the same clusters on the same data set.

Yes or No?

- Any clustering algorithm will find the true clusters in a data set.

Yes or No?

What now... things to do

- Read Section 2.4 and Chapters 10 and 11 in text book
- **Assignment 1 due tomorrow 5 pm**
- Assignment 2 will be released during the semester break – start working on it early!

Have a great semester break! See you back on 19 April