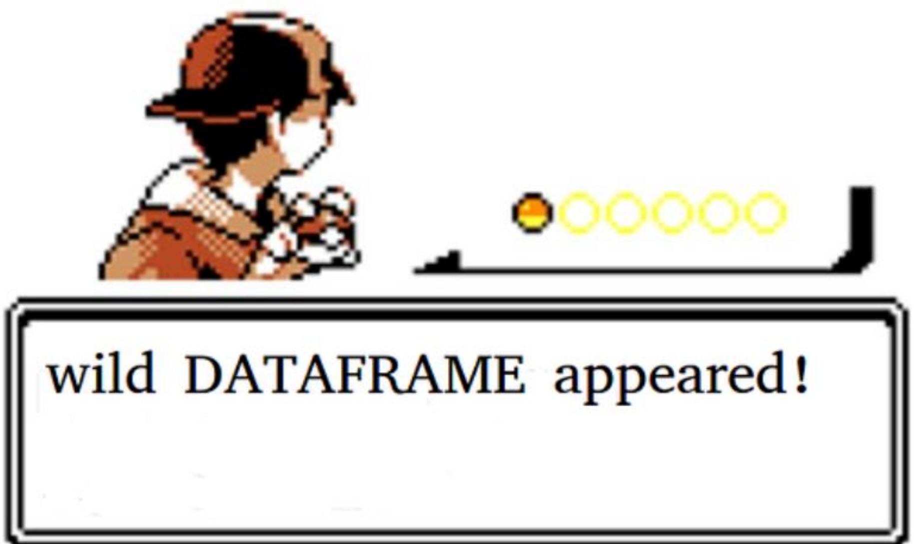
CISC 372
Clustering

	name	age	state	num_children	num_pets
0	john	23	iowa	2	0
1	mary	78	dc	2	4
2	peter	22	california	0	0
3	jeff	19	texas	1	5
4	bill	45	washington	2	0
5	lisa	33	dc	1	0



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
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- 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 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
- Earthquake studies: Observed earthquake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

Quality: 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 result 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

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, 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

Considerations for Cluster Analysis

- 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., Euclidian, 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

Distance Measures for Different Kinds of Data

Discussed in KNN model

- Numerical (interval)-based:
  - x1=(1,2), x2=(3,5)

  - Manhattan (L_1-norm)
  - Manhattan distance = (3-1) + (5-2) = 5

  - Euclidean (L_2-norm)
  - Euclidean distance = $\sqrt{(3-1)^2 + (5-2)^2} = 3.61$

Distance Measures for Different Kinds of Data

- Binary variables:
  - symmetric vs. asymmetric (Jaccard coeff.)

	Object *j*			
Object *i*		1	0	Sum
	1	*q*	*r*	*q*+*r*
	0	*s*	*t*	*s*+*t*
	Sum	*q*+*s*	*r*+*t*	*p*

- For symmetric binary variables, e.g., gender.
  d(i,j) = (r+s)/(q+r+s+t)
- For asymmetric binary variables, e.g., fever.
  d(i,j) = (r+s)/(q+r+s)

	Object j			
Object i		1 (M/Y)	0 (F/N)	Sum
	1 (M/Y)	q	r	$q+r$
	0 (F/N)	s	t	$s+t$
	Sum	$q+s$	$r+t$	p

Sym: $d(i,j) = (r+s)/(q+r+s+t)$

Name	Gender	Adult	Student
Paul	M	Y	N
John	M	Y	N
Irene	F	N	Y
Peter	M	N	Y

- Symmetric: d(Paul,John) = 0/3 = 0
- Symmetric: d(Paul,Irene) = 3/3 = 1
- Symmetric: d(Paul,Peter) = 2/3 = 0.67

	Object *j*			
Object *i*		1 (M/Y)	0 (F/N)	Sum
	1 (M/Y)	*q*	*r*	*q*+*r*
	0 (F/N)	*s*	*t*	*s*+*t*
	Sum	*q*+*s*	*r*+*t*	*p*

Sym: d(i,j) = (r+s)/(q+r+s+t)

Asym: d(i,j) = (r+s)/(q+r+s)

Name	Gender	Fever	Zika	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	Y	N	N	N
Mary	F	Y	N	Y	N	Y	N
Jim	M	Y	Y	N	N	N	N

- Symmetric: d(Jack,Mary) = 1/1=1
- Asymmetric: d(Jack,Mary) = (0+1)/(2+0+1) = 1/3 = 0.3
- Asymmetric: d(Jack,Jim) = (1+1)/(1+1+1) = 2/3 = 0.67
- Asymmetric: d(Mary,Jim) = (1+2)/(1+1+2) = 3/4 = 0.75
- Both: d(Jack,Mary) = (1+0+1)/(1+2+0+1) = 2/4 = 0.5

Centroid, Radius and Diameter of a Cluster (for numerical data sets) - Skip

- Centroid: the "middle" of a cluster

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

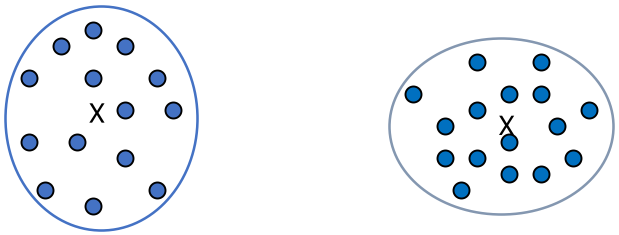
- Radius: square root of average distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$

Distance between Clusters



- **Single link:** smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- **Complete link:** largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- **Average:** avg distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$
- **Centroid:** distance between the centroids of two clusters, i.e., $dist(K_i, K_j) = dist(C_i, C_j)$
- **Medoid:** distance between the medoids of two clusters, i.e., $dist(K_i, K_j) = dist(M_i, M_j)$
  - Medoid: a chosen, centrally located object in the cluster

Chapter 10. Cluster Analysis

1. What is Cluster Analysis?
2. A Categorization of Major Clustering Methods
3. Partitioning Methods
4. Hierarchical Methods
5. Density-Based Methods
6. Clustering High-Dimensional Data
7. Summary

Major Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing 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, ROCK, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster, HFTC, FIHC

Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. A Categorization of Major Clustering Methods
3. Partitioning Methods
4. Hierarchical Methods
5. Density-Based Methods
6. Clustering High-Dimensional Data
7. Summary

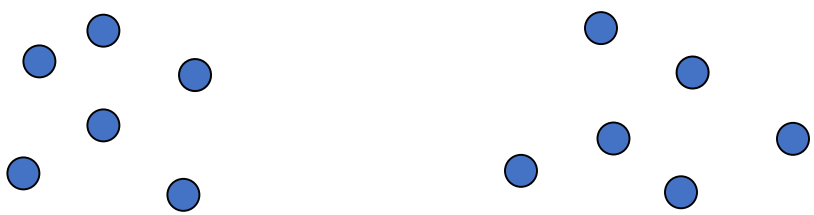
Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (dist(p, c_i))^2$$

- Given *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, Lloyd'57/'82): 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

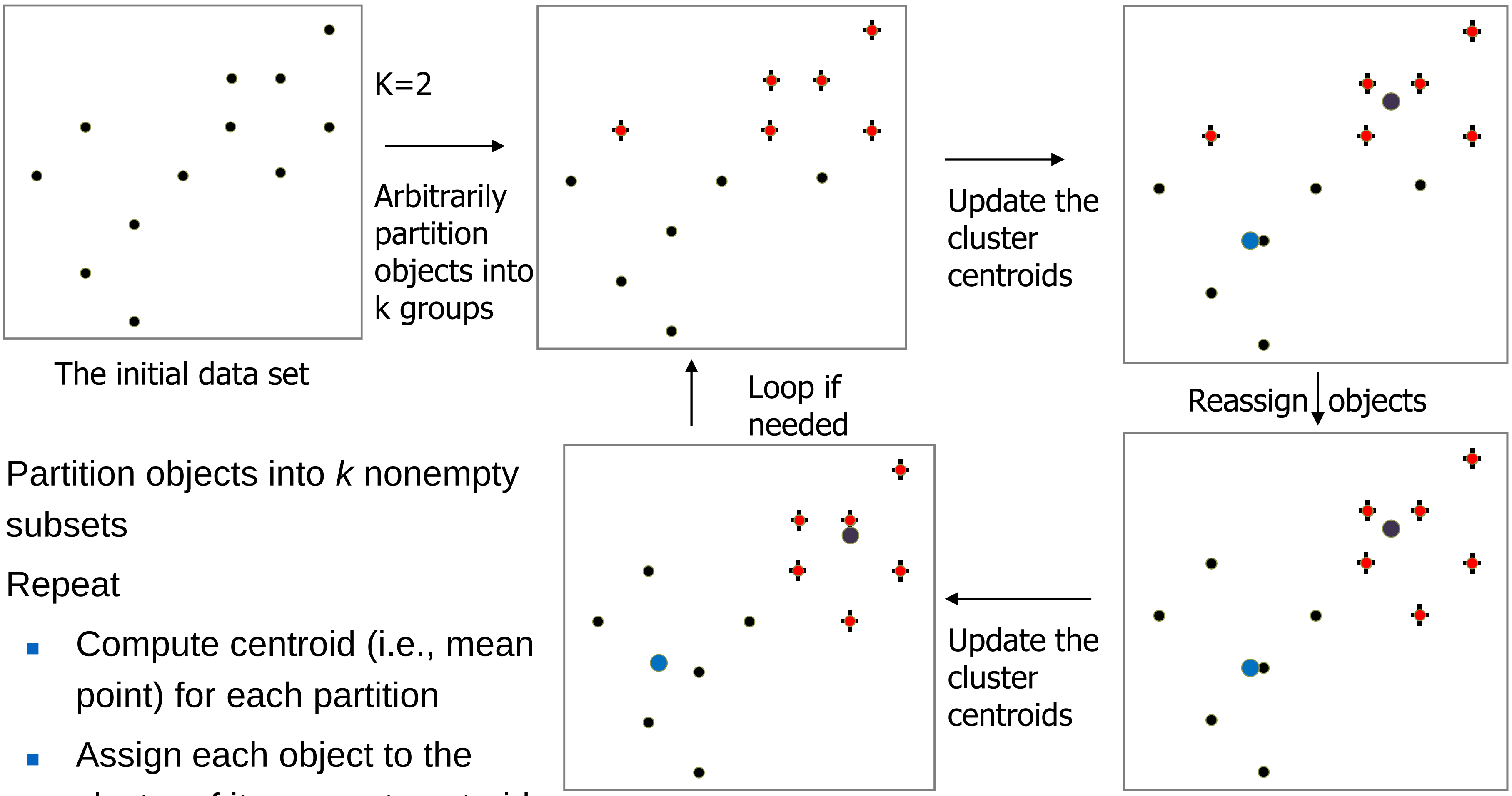
Partitioning Algorithms: Basic Concept



The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  1. Partition objects into *k* nonempty subsets
  2. Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *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 the assignment does not change
- Demo: http://util.io/k-means

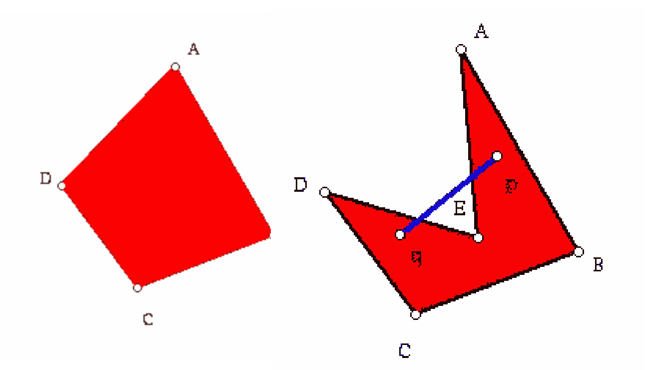
An Example of *K-Means* Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Comments on the *K-Means* Method

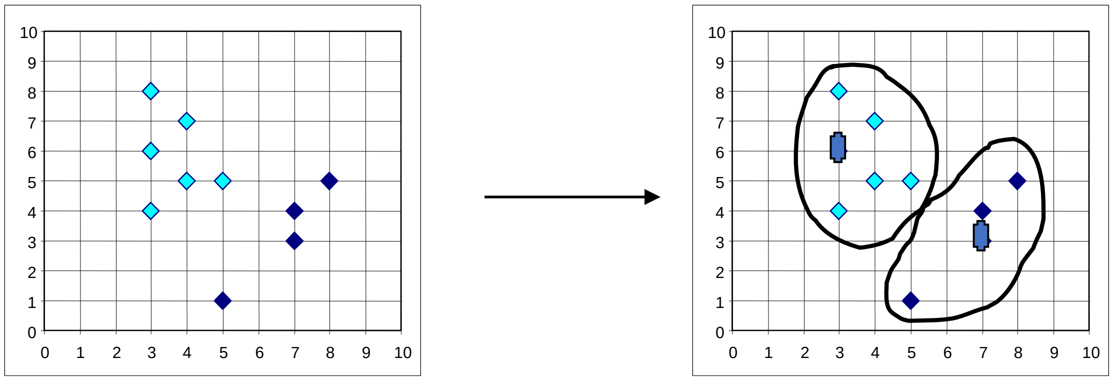
- Strength: *Efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
- Comment: Often terminates at a *local optimal*



- Weakness
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

What Is the Problem of the K-Means Method?

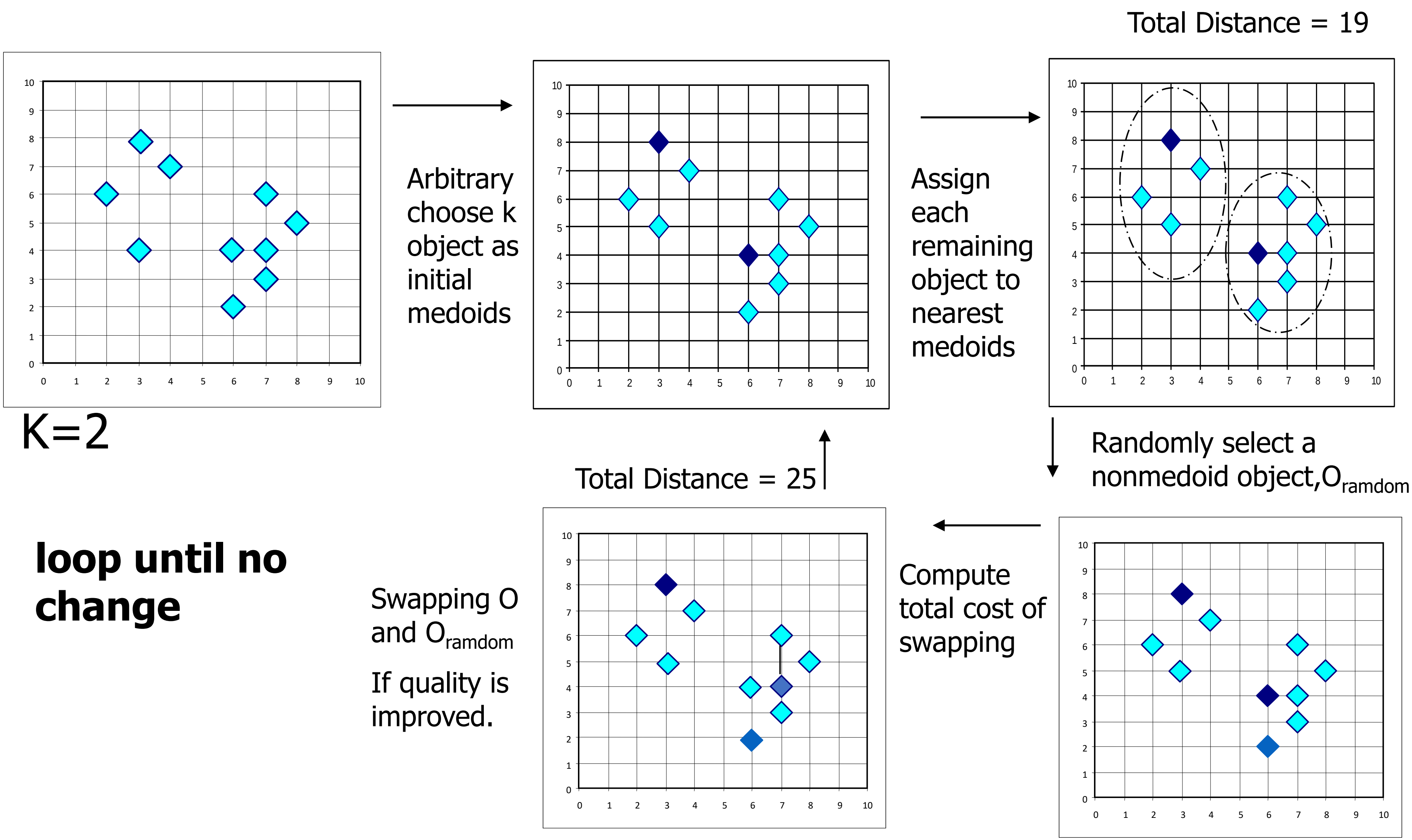
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally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



PAM (Partitioning Around Medoids) (1987)

- Find *representative* objects, called <u>medoids</u>, in clusters
- Use real object to represent the cluster
  - arbitrarily select ***k*** representative objects
  - repeat
    - assign each remaining object to nearest representative object o_j
    - randomly select a non-representative object o_{random}
    - compute the total cost, *TC*, of swapping o_j with o_{random}
    - if $TC < 0$, ***i*** is replaced o_j by o_{random}
  - until there is no change

PAM: A Typical K-Medoids Algorithm



What Is the Problem with PAM?

- PAM 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
- PAM works efficiently for small data sets but **does not scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

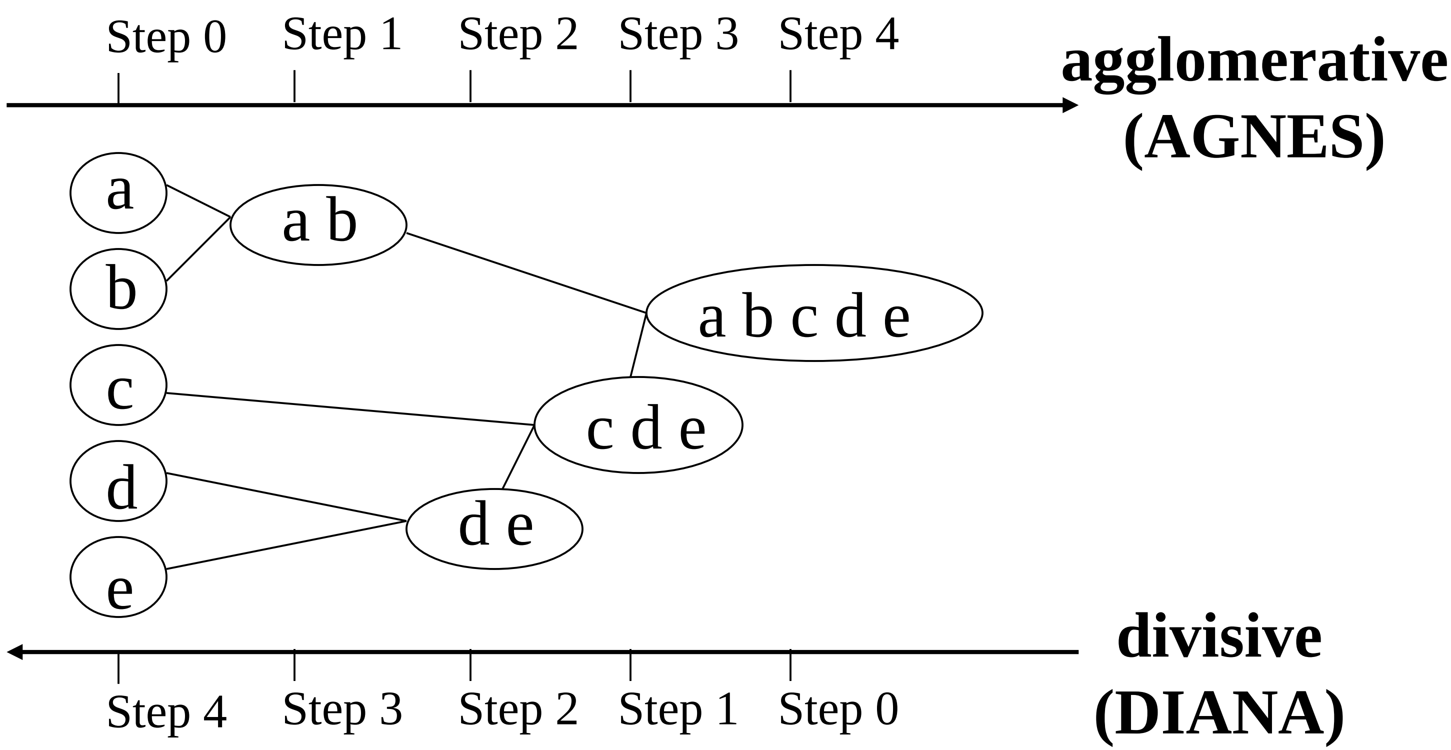
  where n is # of objects, k is # of clusters

Chapter 10. Cluster Analysis

1. What is Cluster Analysis?
2. A Categorization of Major Clustering Methods
3. Partitioning Methods
4. Hierarchical Methods
5. Density-Based Methods
6. Clustering High-Dimensional Data
7. Summary

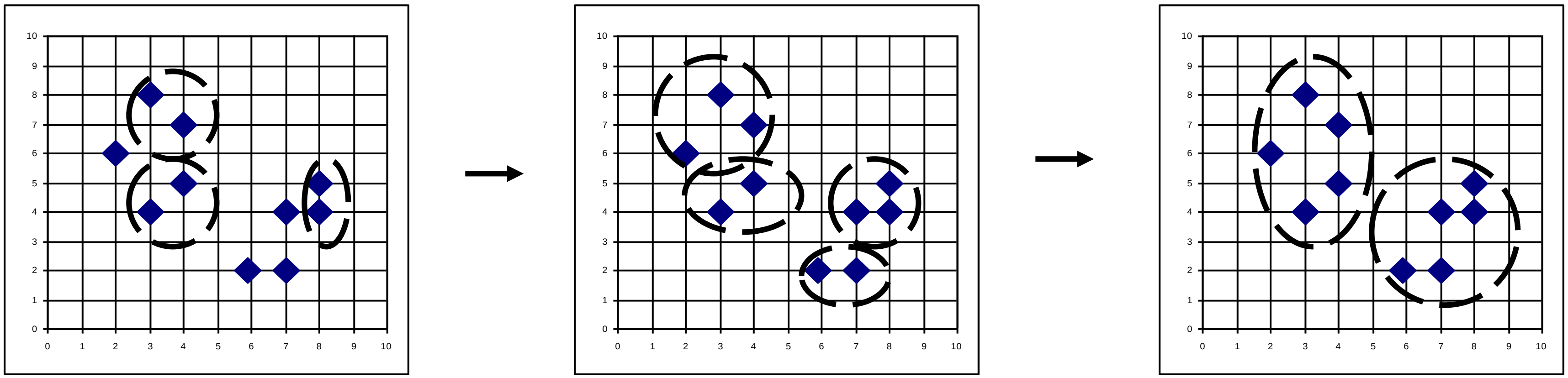
Hierarchical Clustering

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



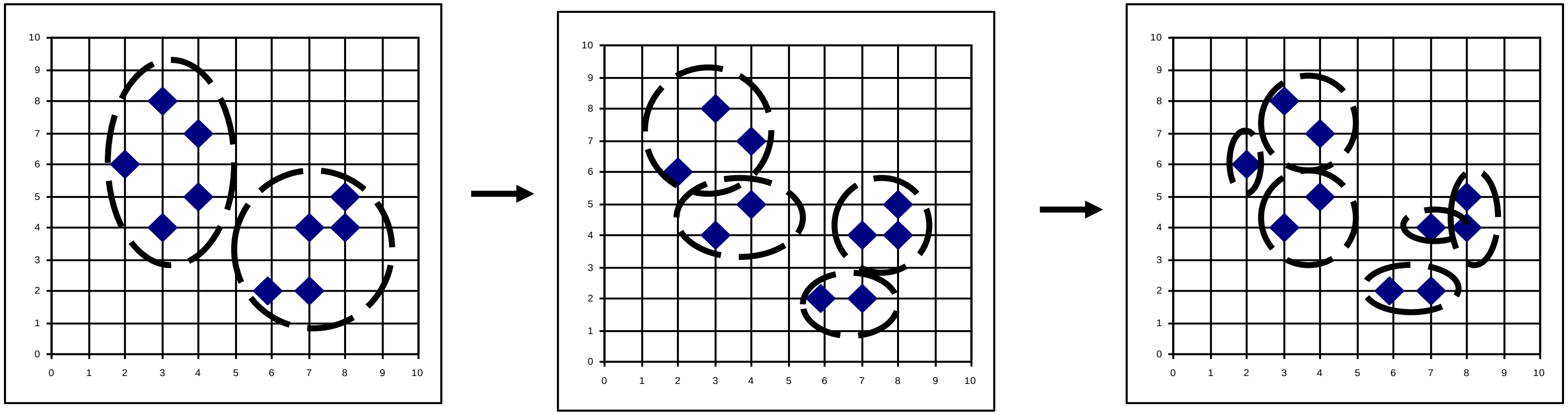
AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix
- Merge <u>clusters</u> that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster

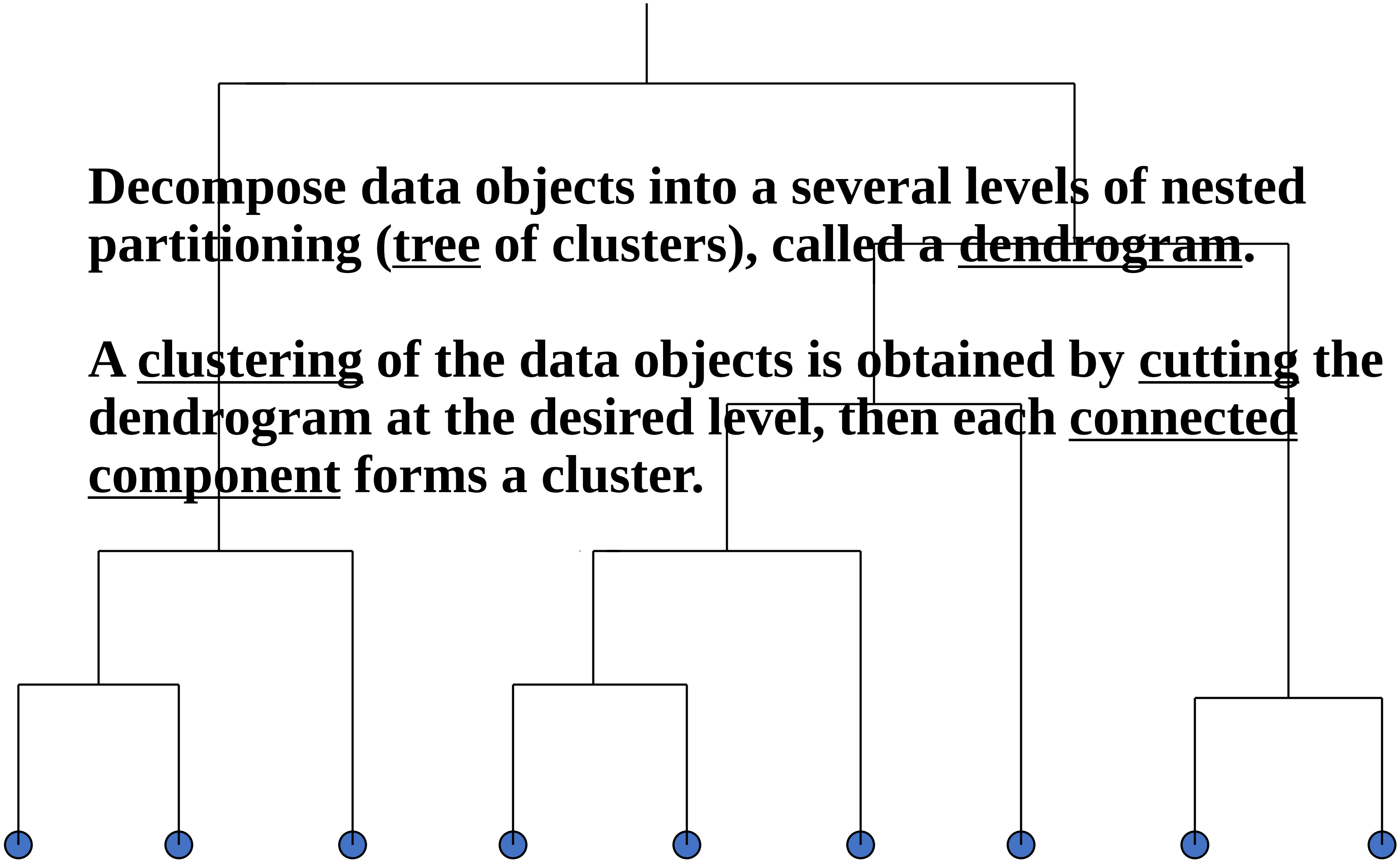


DIANA (Divisive Analysis)

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



Dendrogram: Shows How the Clusters are Merged



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

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

Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - <u>Do not scale</u> well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - Can never undo what was done previously
- (Refernece) Integration of hierarchical & distance-based clustering
  - <u>BIRCH (1996)</u>: uses CF-tree and incrementally adjusts the quality of sub-clusters
  - <u>ROCK (1999</u>): clustering categorical data by neighbor and link analysis
  - <u>CHAMELEON (1999)</u>: hierarchical clustering using dynamic modeling

Chapter 10. Cluster Analysis

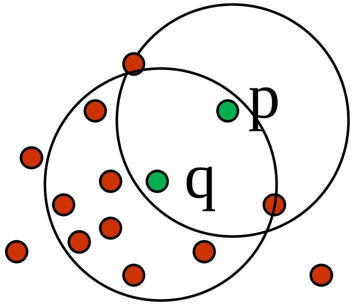
1. What is Cluster Analysis?
2. A Categorization of Major Clustering Methods
3. Partitioning Methods
4. Hierarchical Methods
5. Density-Based Methods
6. Clustering High-Dimensional Data
7. Summary

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) ← This one only…
  - 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*:*
  - ε: Maximum radius of the neighbourhood
  - *MinPts*: Minimum number of points in an ε-neighbourhood of that point
- $N_\varepsilon(q)$: $\{p \mid dist(p,q) <= \varepsilon\}$
- q is a core point if $|N_\varepsilon(q)| >= MinPts$
- Directly density-reachable: A point p is directly density-reachable from a point q with respect to ε and *MinPts* if
  - p belongs to $N_\varepsilon(q)$ and
  - q is a core point.

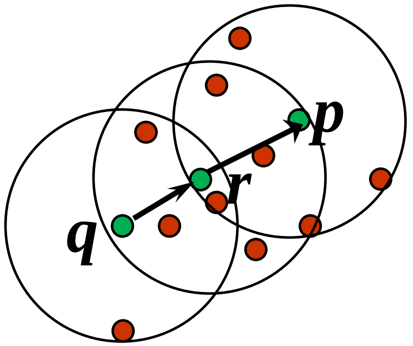


$MinPts = 5$

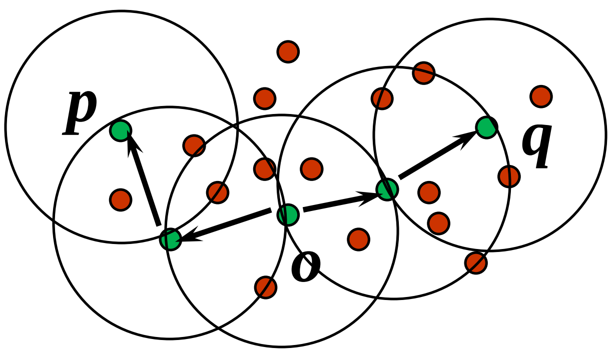
$\varepsilon = 1$ cm

Density-Reachable and Density-Connected

- Density-reachable:
  - A point p is density-reachable from a point q w.r.t. 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 w.r.t. ε, *MinPts* if there is a point o such that both p and q are density-reachable from o.

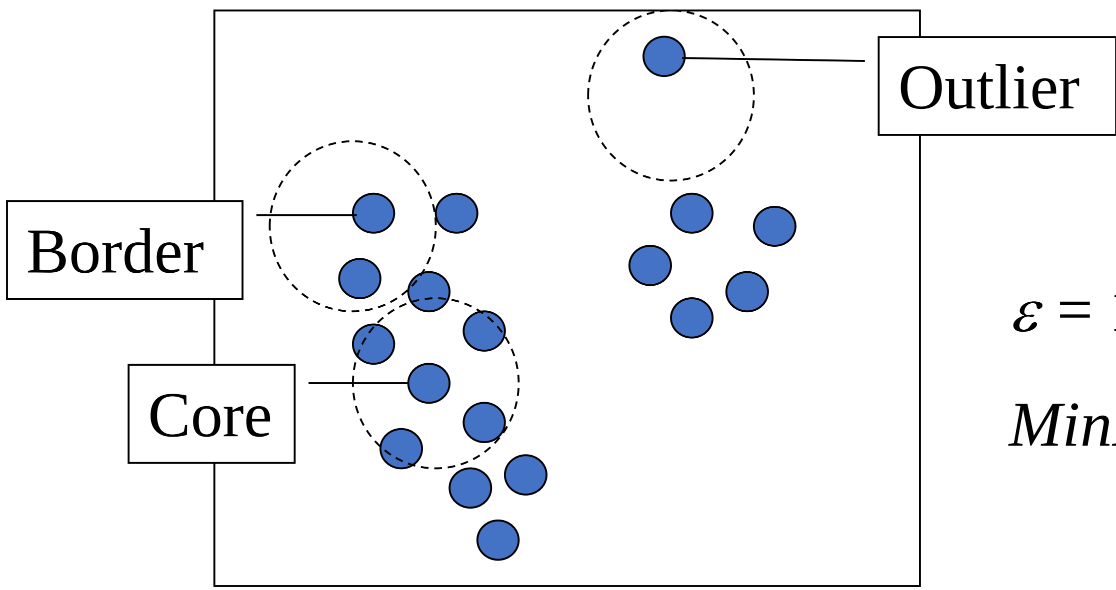


DBSCAN: The Algorithm

- Mark all objects as “unvisited”
- Arbitrarily select a point *p*, and mark it as “visited”.
- If *p* is not a core point, no points are density-reachable from *p*. So, it is a border point or an outlier.
- If *p* is a core point, form a new cluster *C* for *p*. For each “unvisited” neighbour *p′* of *p*, if *p′* is a core point, add *p′*s neighbours to *C,* and mark it as “visited”. Continue expanding *C* until *C* can no longer be expanded.
- Select another unvisited object from the remaining ones.

DBSCAN: Density Based Spatial Clustering of Applications with Noise

- 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



ε = 1cm

$MinPts = 4$

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

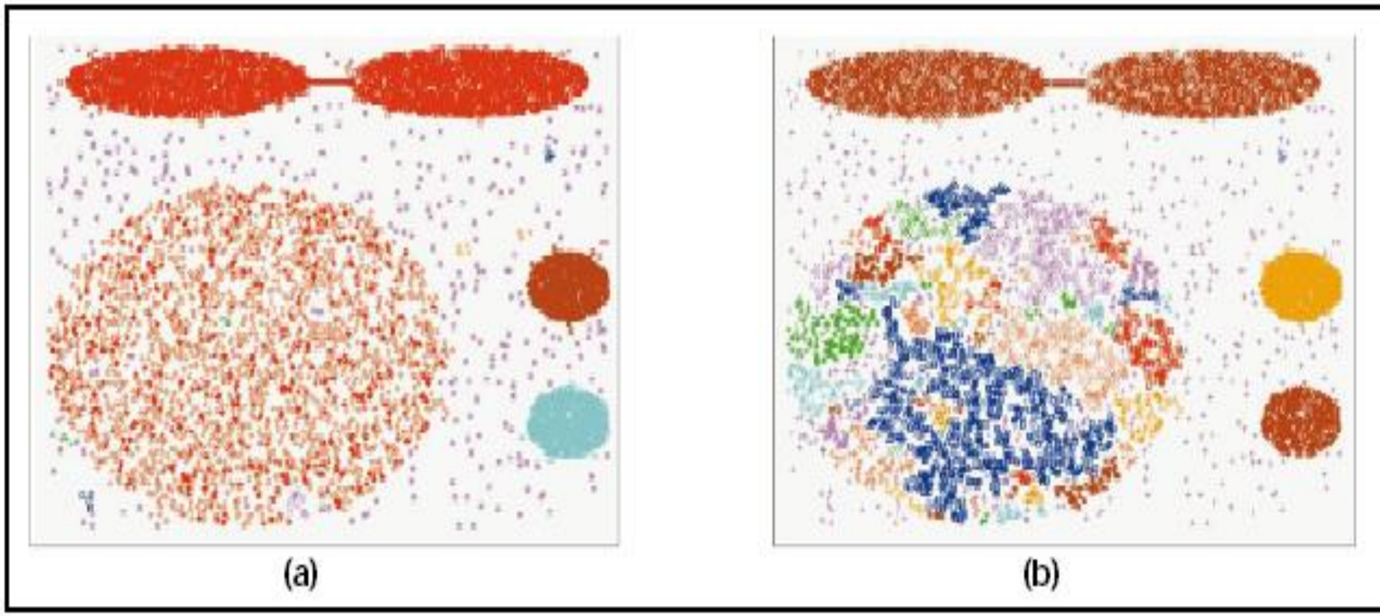
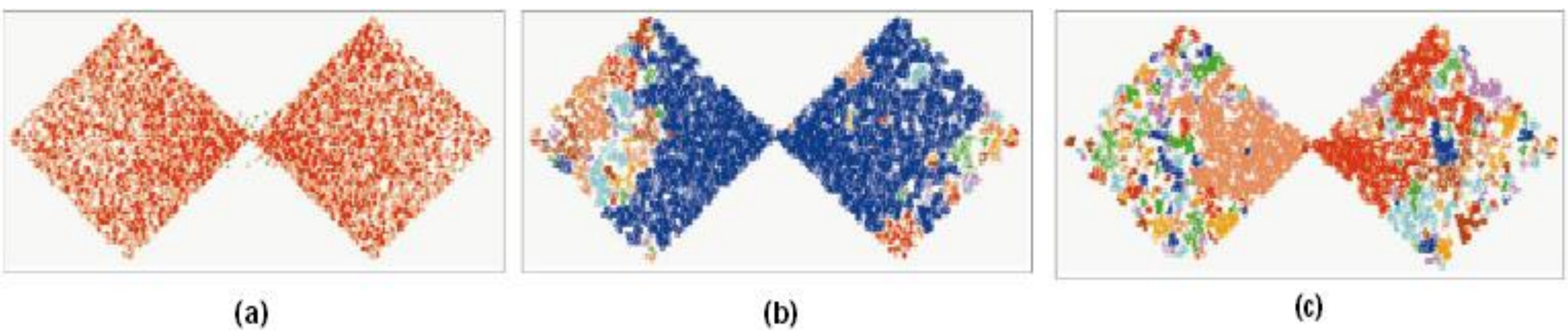


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBScan Advantages

- DBScan does not require you to know the number of clusters in the data a priori.

- DBScan does not have a bias towards a particular cluster shape or size.

- DBScan is resistant to noise and provides a means of filtering for noise if desired.

DBScan Disadvantages

- DBScan does not respond well to high dimensional data. As dimensionality increases, so does the relative distance between points making it harder to perform density analysis.

- DBScan does not respond well to data sets with varying densities.

Chapter 10. Cluster Analysis

1. What is Cluster Analysis?
2. A Categorization of Major Clustering Methods
3. Partitioning Methods
4. Hierarchical Methods
5. Density-Based Methods
6. Clustering High-Dimensional Data
7. Summary

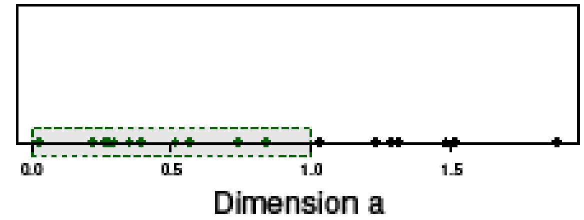
Clustering High-Dimensional Data

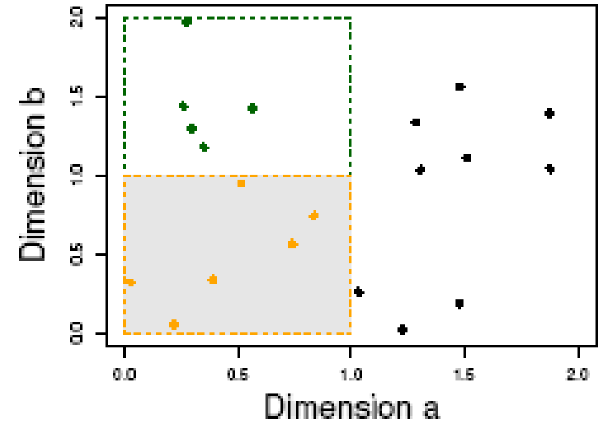
- Clustering high-dimensional data
  - Many applications: text documents, DNA micro-array data
  - Major challenges:
    - Many irrelevant dimensions may mask clusters
    - Distance measure becomes meaningless—due to equi-distance
    - Clusters may exist only in some subspaces

The Curse of Dimensionality

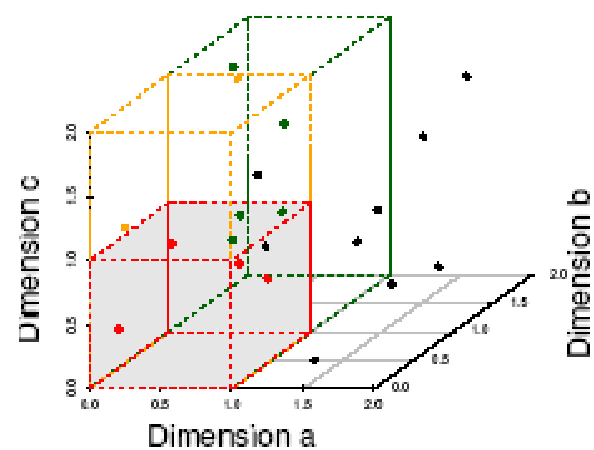
(graphs adapted from Parsons et al. KDD Explorations 2004)

- Data in only one dimension is relatively packed
- Adding a dimension “stretch” 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

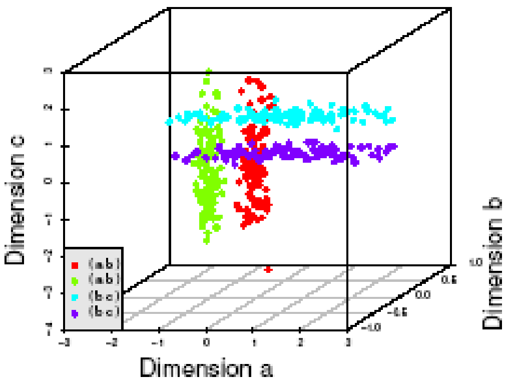




(b) 6 Objects in One Unit Bin



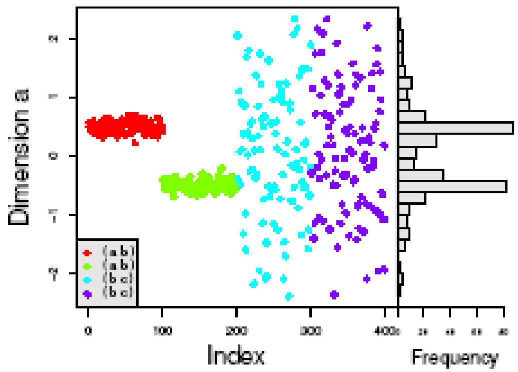
(c) 4 Objects in One Unit Bin



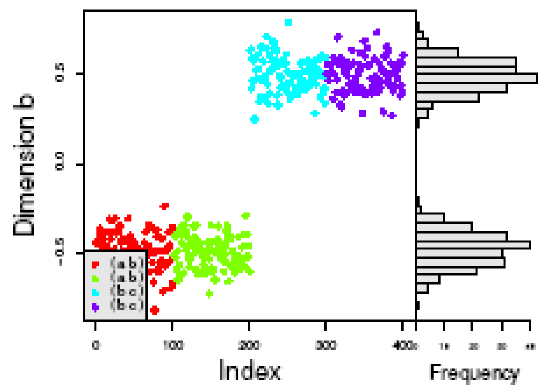
Why Subspace Clustering?

(adapted from Parsons et al. SIGKDD Explorations 2004)

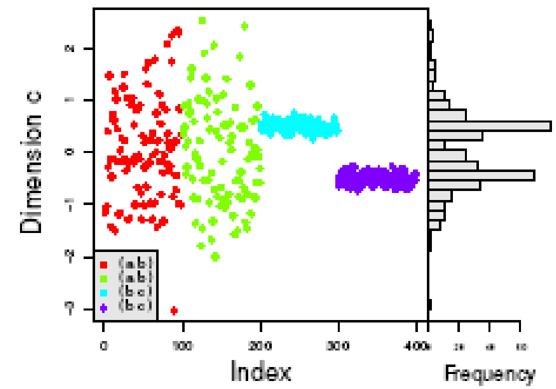
- Clusters may exist only in some subspaces
- Subspace-clustering: find clusters in all the subspaces



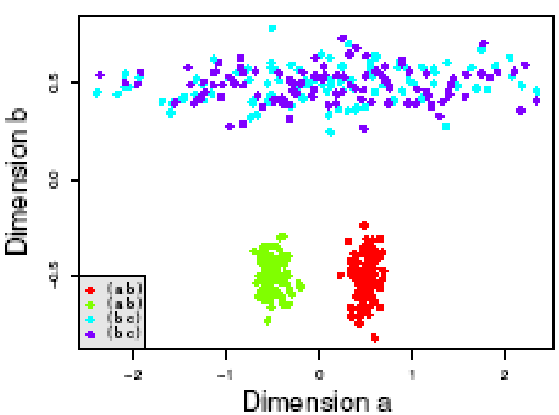
(a) Dimension a



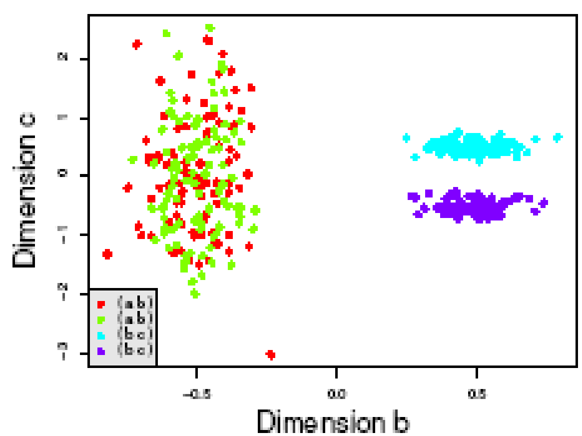
(b) Dimension b



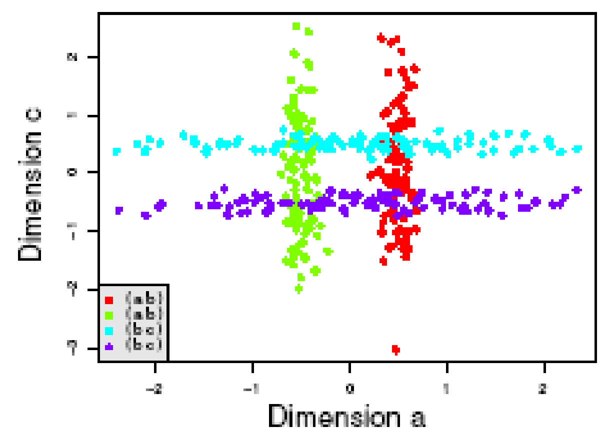
(c) Dimension c



(a) Dims a & b



(b) Dims b & c



(c) Dims a & c

Subspace Clustering (example)

	Apple	Orange	Banana	Microsoft	Window
Doc1	1	1	1		
Doc2	1			1	1
Doc3	1		1		
Doc4	1			1	

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- There are still lots of research issues on cluster analysis

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- Beil F., Ester M., Xu X.: "Frequent Term-Based Text Clustering", KDD'02
- M. M. Breunig, H.-P. Kriegel, R. Ng, J. Sander. LOF: Identifying Density-Based Local Outliers. SIGMOD 2000.
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. VLDB'98.
- V. Ganti, J. Gehrke, R. Ramakrishan. CACTUS Clustering Categorical Data Using Summaries. KDD'99.

References (2)

- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB’98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. In ICDE'99, pp. 512-521, Sydney, Australia, March 1999.
- A. Hinneburg, D.l A. Keim: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD’98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.
- G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. COMPUTER, 32(8): 68-75, 1999.
- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB’98.

References (3)

- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- L. Parsons, E. Haque and H. Liu, Subspace Clustering for High Dimensional Data: A Review, SIGKDD Explorations, 6(1), June 2004
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition,.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB’98.
- A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-Based Clustering in Large Databases, ICDT'01.
- A. K. H. Tung, J. Hou, and J. Han. Spatial Clustering in the Presence of Obstacles, ICDE'01
- H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by pattern similarity in large data sets, SIGMOD’ 02.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB’97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : An efficient data clustering method for very large databases. SIGMOD'96.
- Xiaoxin Yin, Jiawei Han, and Philip Yu, “LinkClus: Efficient Clustering via Heterogeneous Semantic Links”, in Proc. 2006 Int. Conf. on Very Large Data Bases (VLDB'06), Seoul, Korea, Sept. 2006.