CLUSTERING ESSENTIALS

Hari Sundaram

hs1@illinois.edu

http://sundaram.cs.illinois.edu

adapted from slides by Jiawei Han and Kevin Chang

BASIC CONCEPTS

Partitioning Hierarchical Density-Based

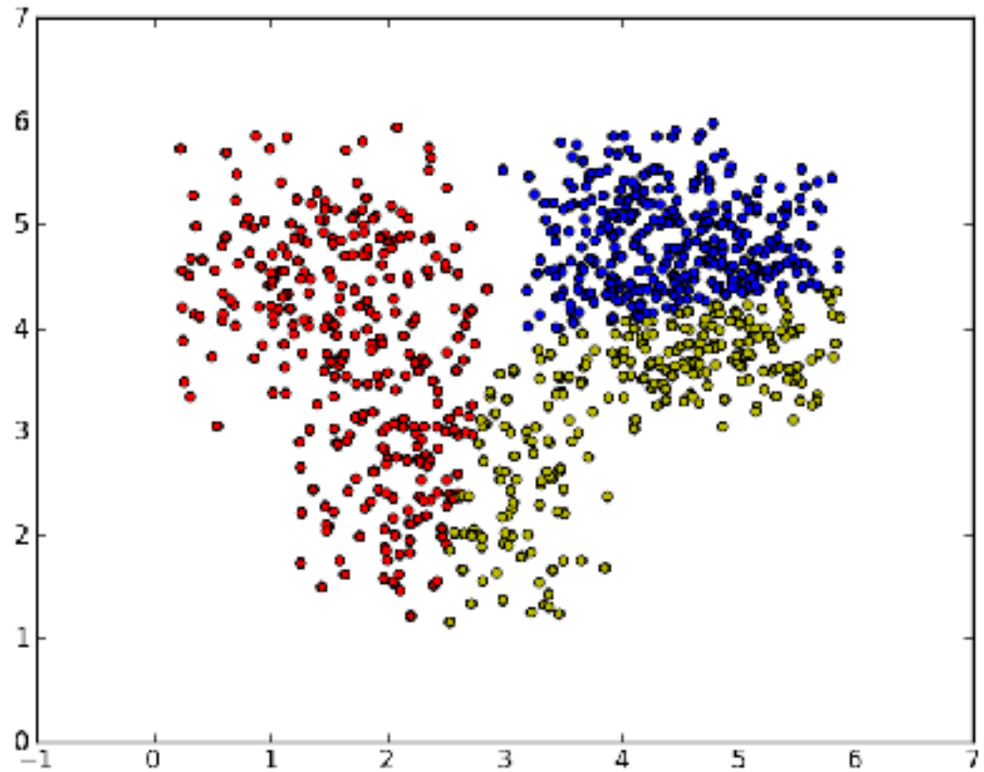
Grid-Based Evaluation Summary

AKA Unsupervised learning: no predefined classes (i.e., learning by observations vs. learning by examples: supervised)

preprocessing step

insights into the data

A collection of data objects similar (or related) to one another within the same group dissimilar (or unrelated) to the objects in other groups



what is a cluster?

Data reduction

Summarization:
Preprocessing for regression, PCA, classification, and association analysis

Compression: Image processing: vector quantization

Outlier detection

applications

Hypothesis generation and testing

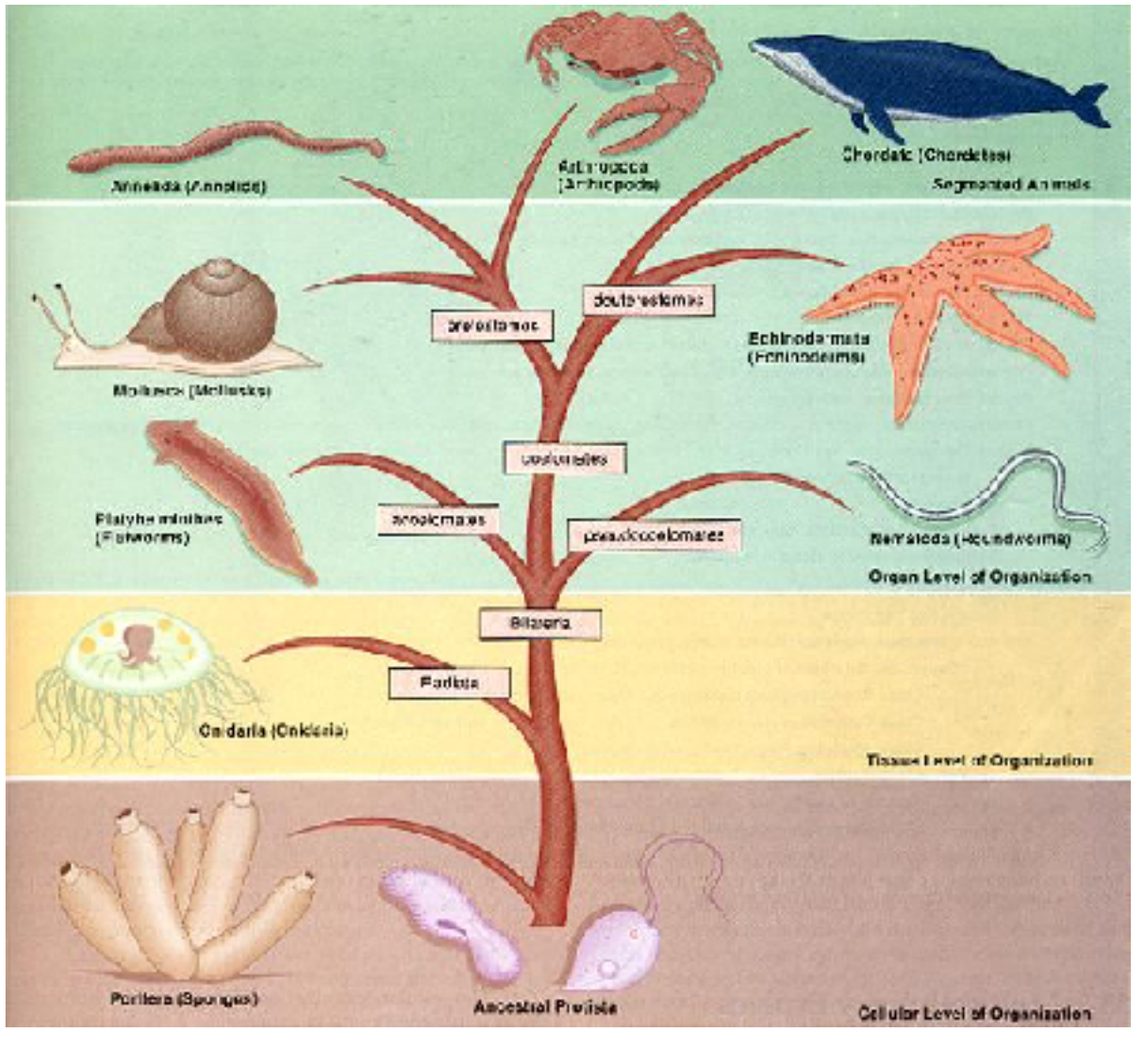
Finding K-nearest Neighbors

Localizing search to one or a small number of clusters

Prediction

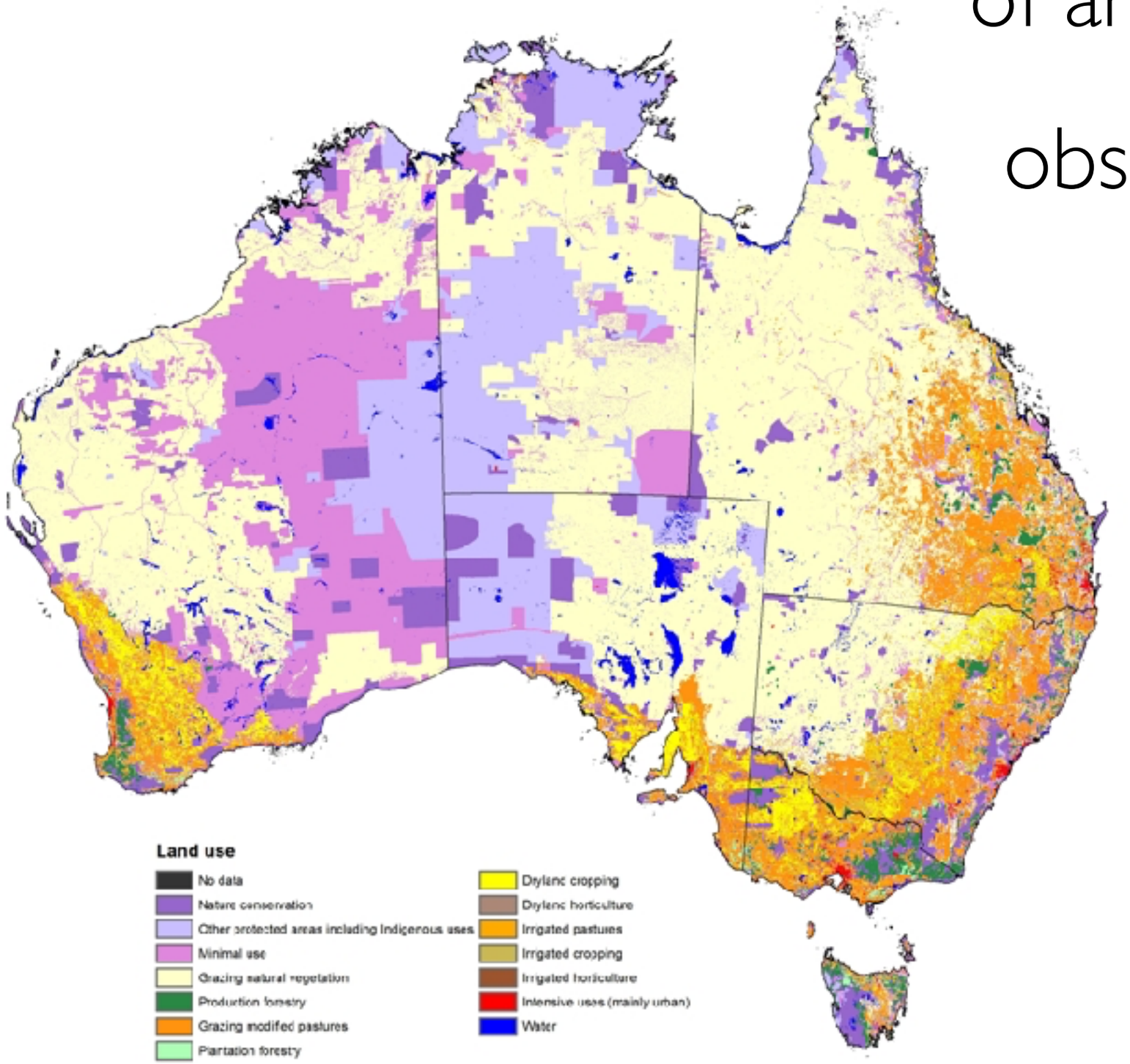
Cluster & find characteristics/patterns for each group

Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species



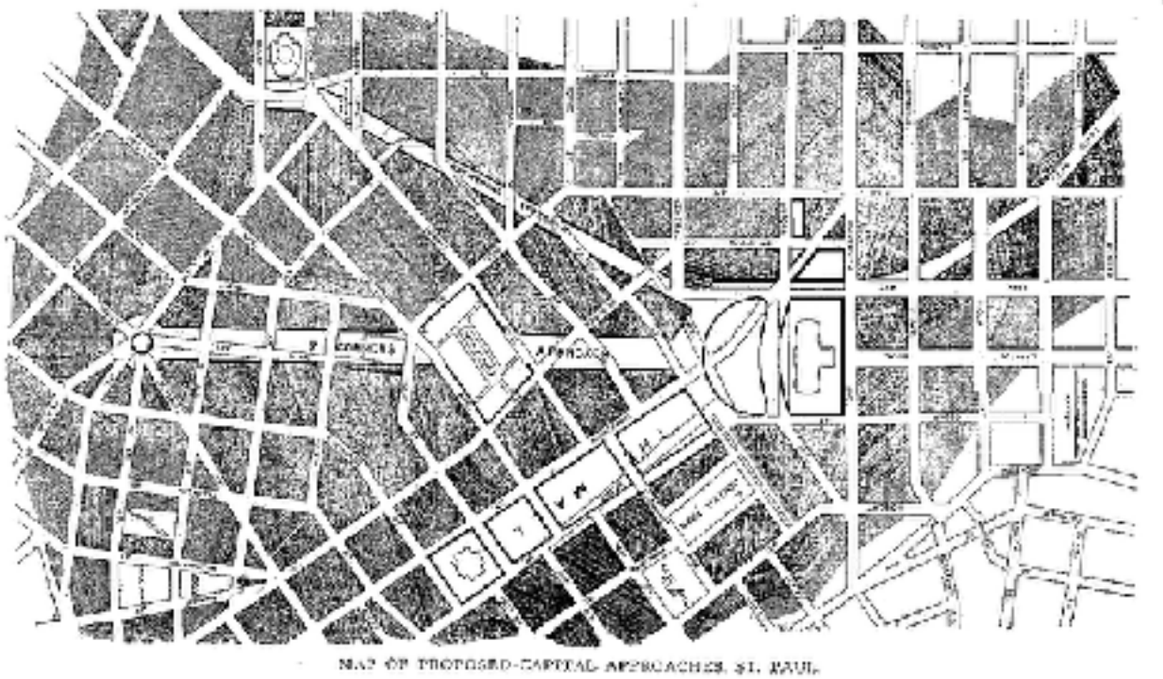
Information **retrieval:** document clustering

Land use: Identification of areas of similar land use in an earth observation database



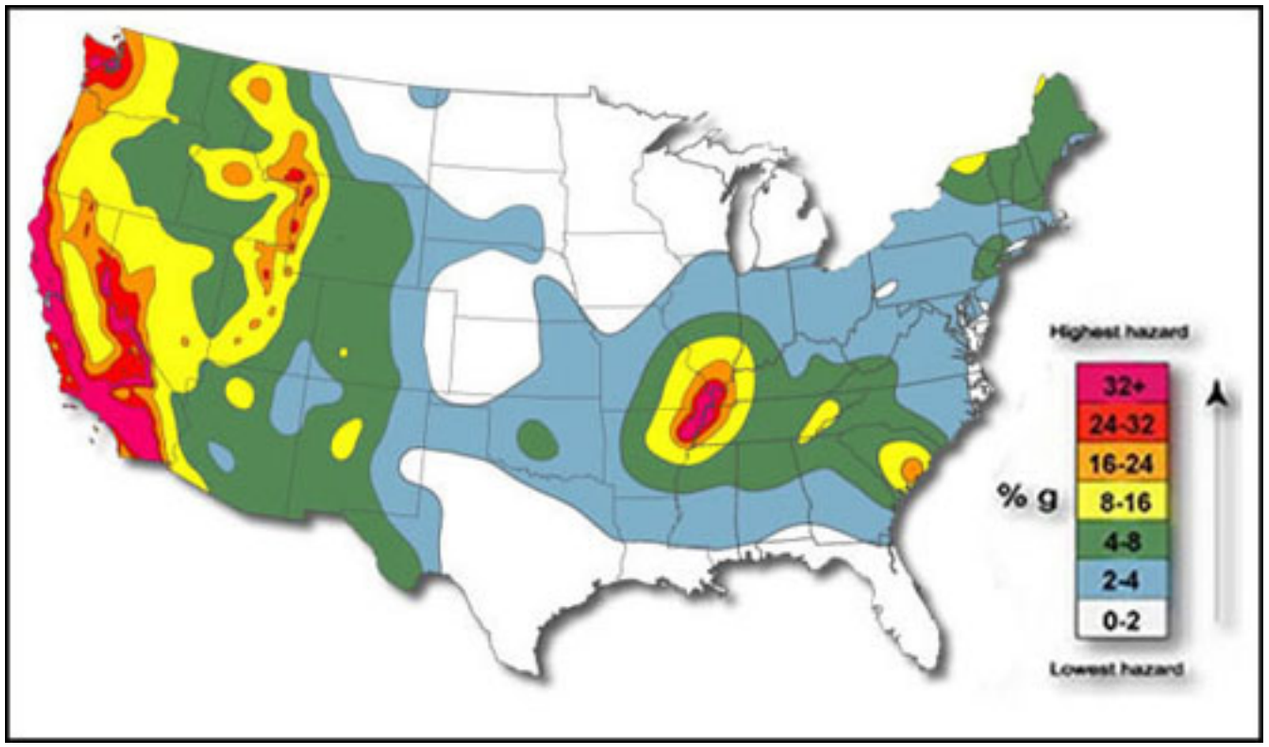
Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

City-planning: Identifying groups of houses according to their house type, value, and geographical location



MAP OF PROPOSED CAPITAL APPROACHES, ST. PAUL.

Earthquake studies: Observed earthquake epicenters should be clustered along continent faults



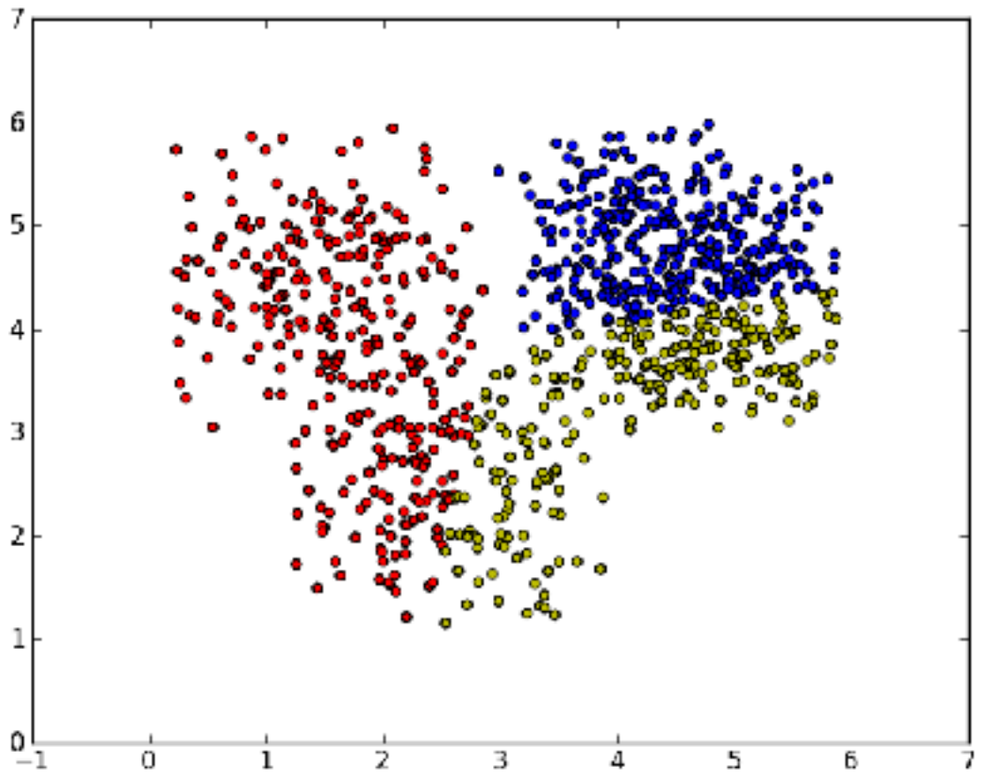
Climate: understanding earth climate, find patterns of atmospheric and ocean

Economic Science: market research

Feature selection

Proximity measure

Clustering criterion



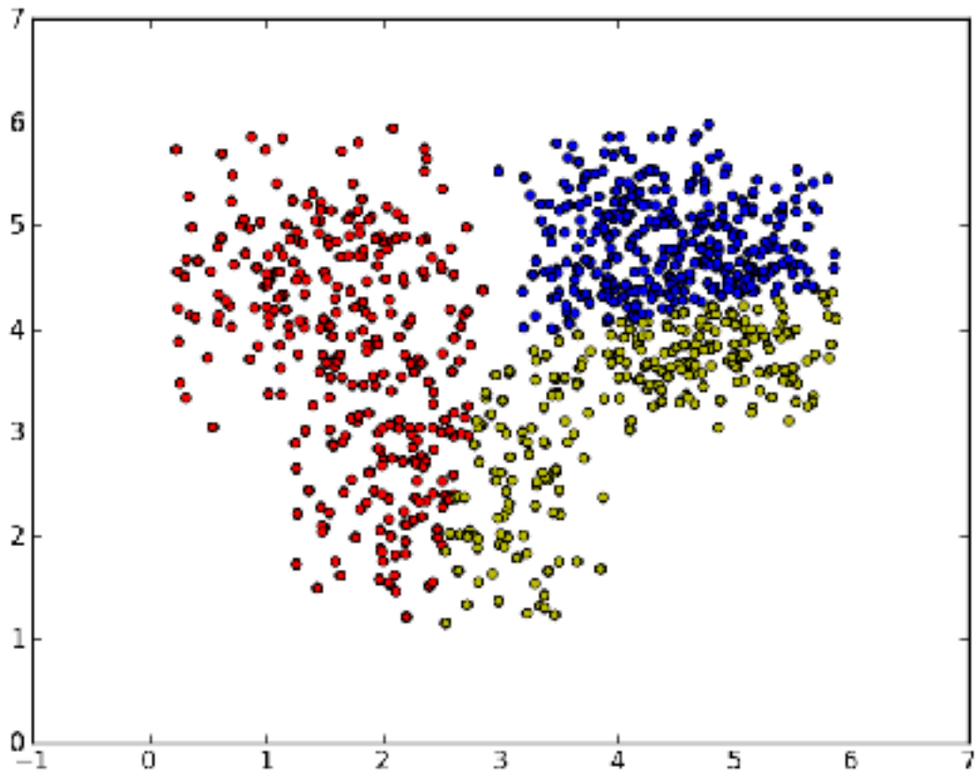
Algorithms

Validation

Interpretation

Integration

A good clustering method will produce high quality clusters

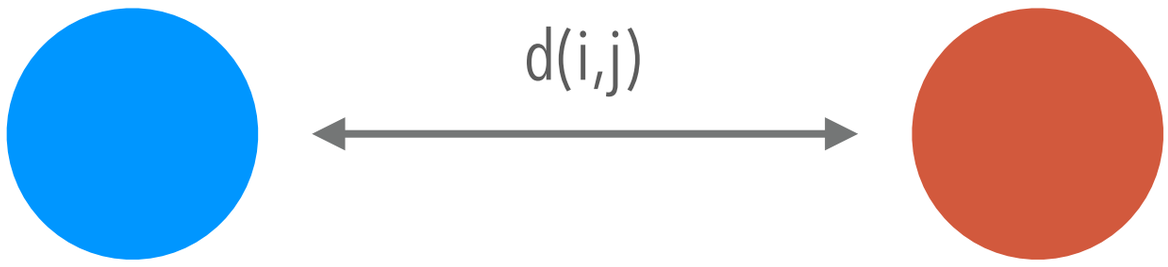


high intra-class similarity: cohesive within clusters

low inter-class similarity: distinctive between clusters

The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables

Dissimilarity/ Similarity metric



Scalability

Discovery of clusters with arbitrary shape

Ability to deal with different types of attributes

Ability to deal with noisy data

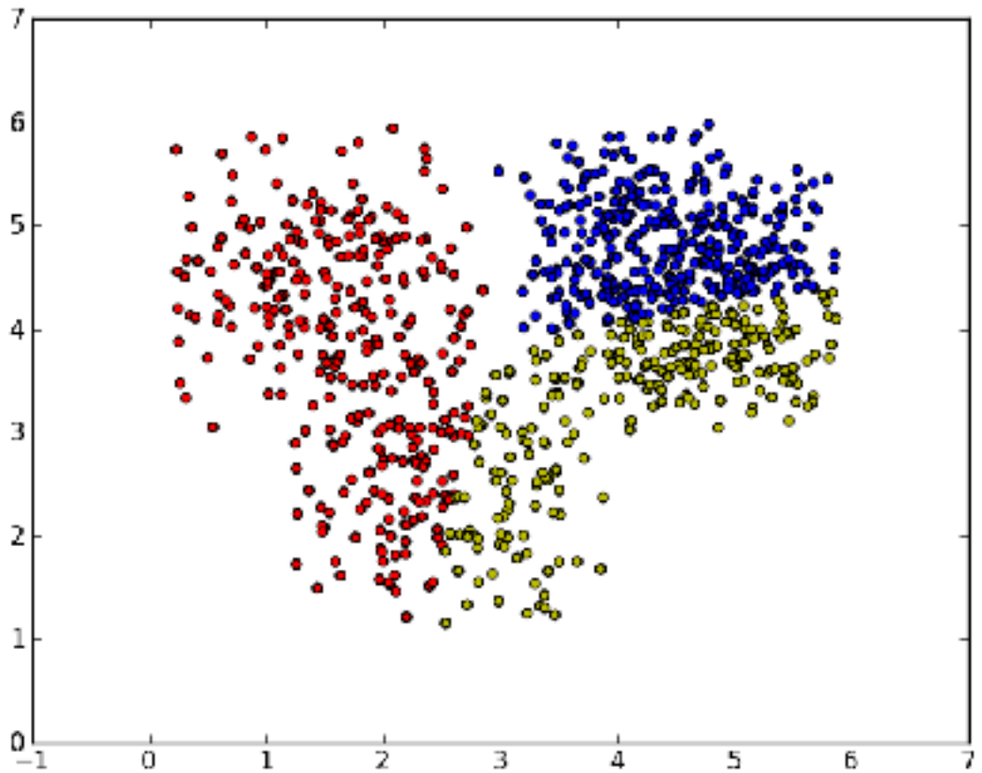
requirements, challenges

Constraint-based clustering

Incremental clustering and insensitivity to input order

Interpretability

High dimensionality



MAJOR APPROACHES

Partitioning approach:

Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors

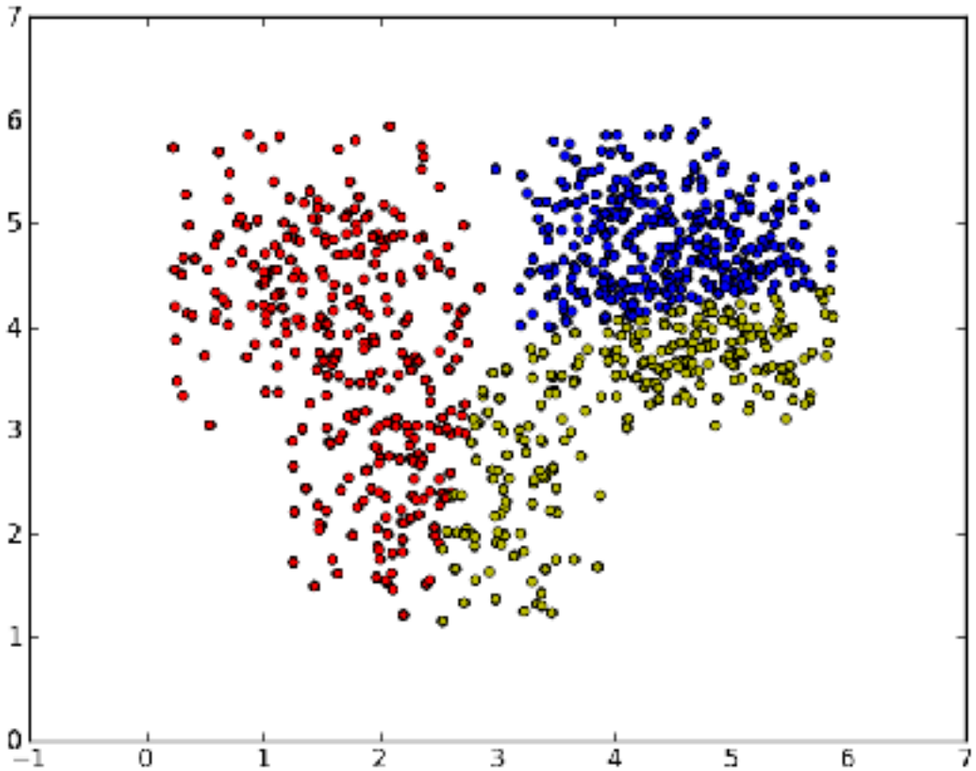
Typical methods: k-means, k-medoids, CLARANS

Hierarchical approach:

Create a hierarchical decomposition of the set of data (or objects) using some criterion

Typical methods: Diana, Agnes, BIRCH, CAMELEON

MAJOR APPROACHES



Density-based approach:

- Based on connectivity and density functions
- Typical methods: DBSACN, OPTICS, DenClue

Grid-based approach:

- based on a multiple-level granularity structure
- Typical methods: STING, WaveCluster, CLIQUE

PARTITION-BASED APPROACHES

PARTITION METHODS

cluster center

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

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)

Given k, find a partition of k clusters that optimizes the chosen partitioning criterion

PARTITION METHODS

cluster center

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

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

k-means

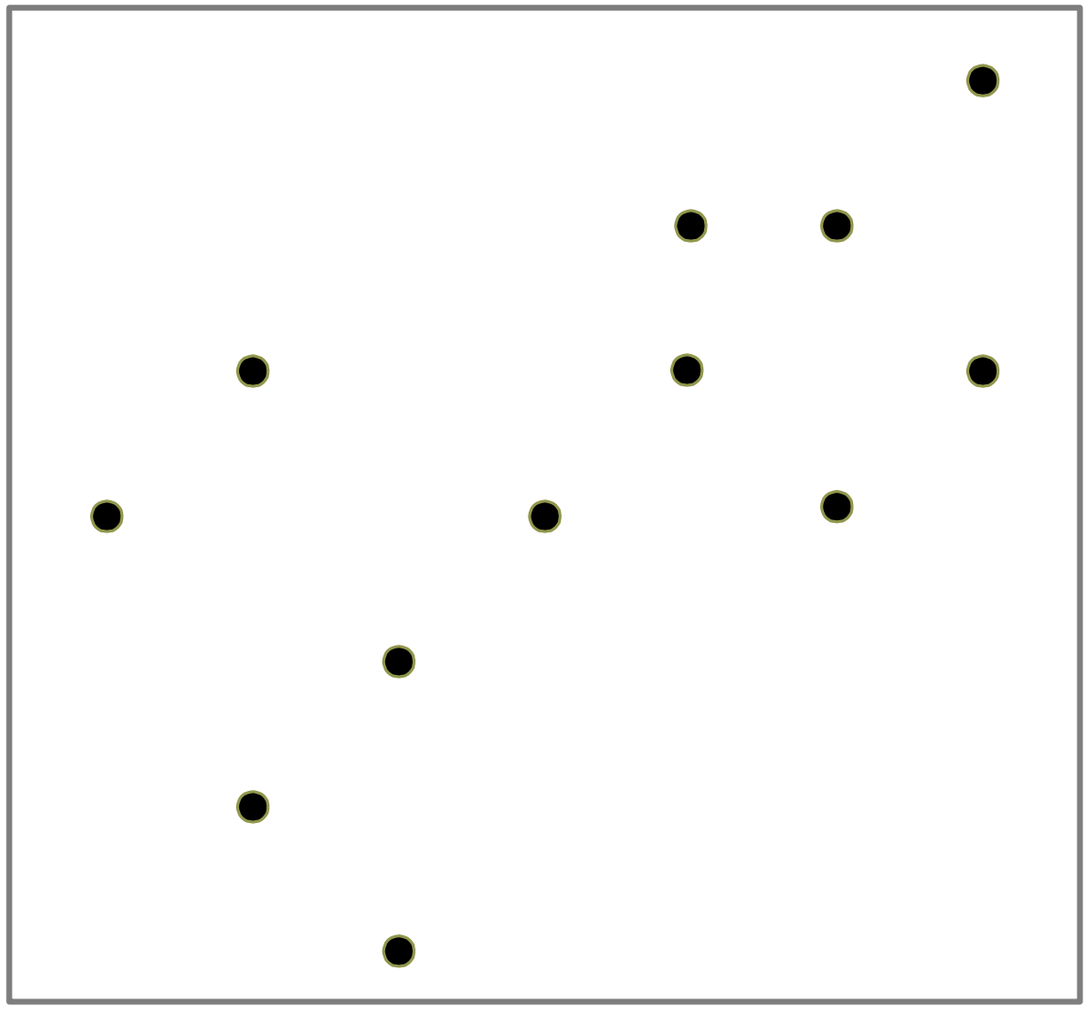
1 Partition objects into k nonempty subsets

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)

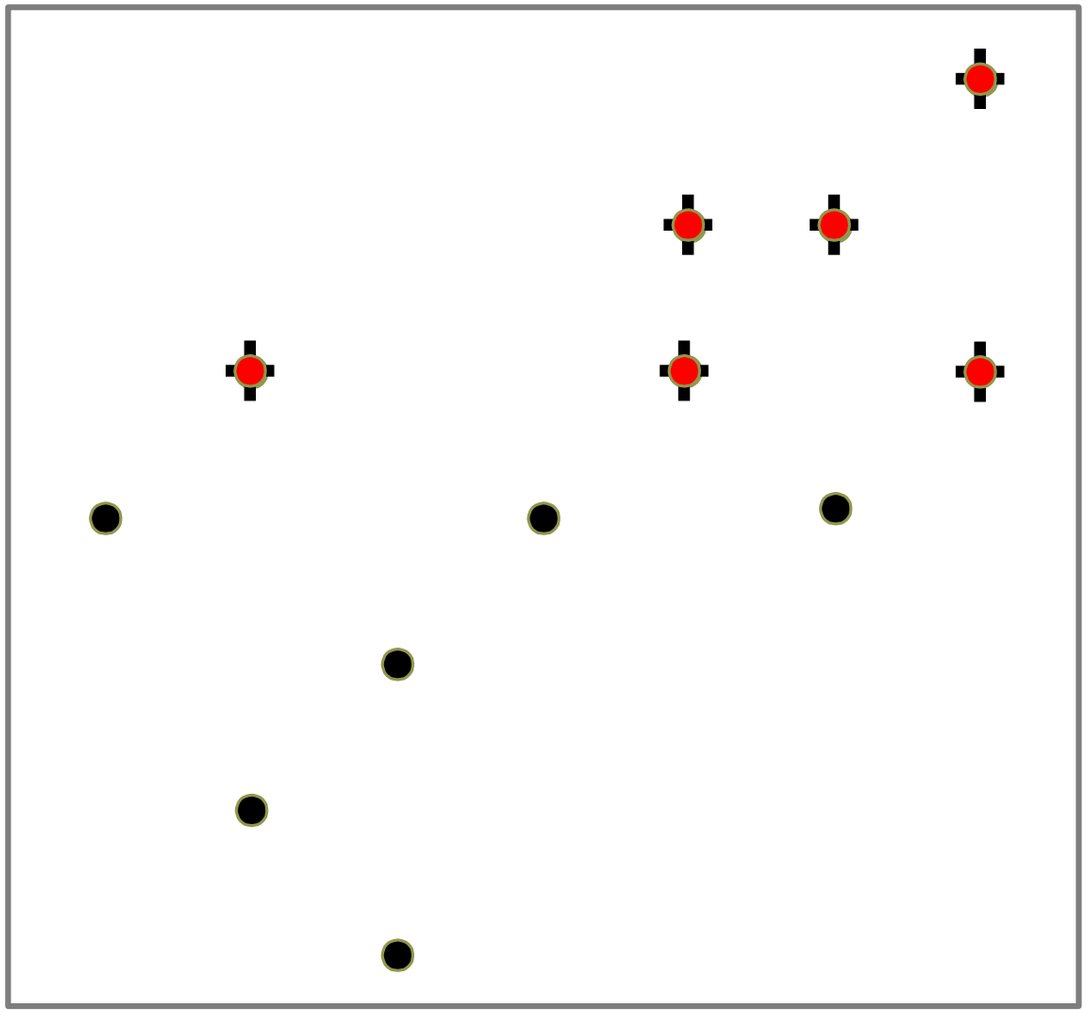
3 Assign each object to the cluster with the nearest seed point

4 Go back to Step 2, stop when the assignment does not change

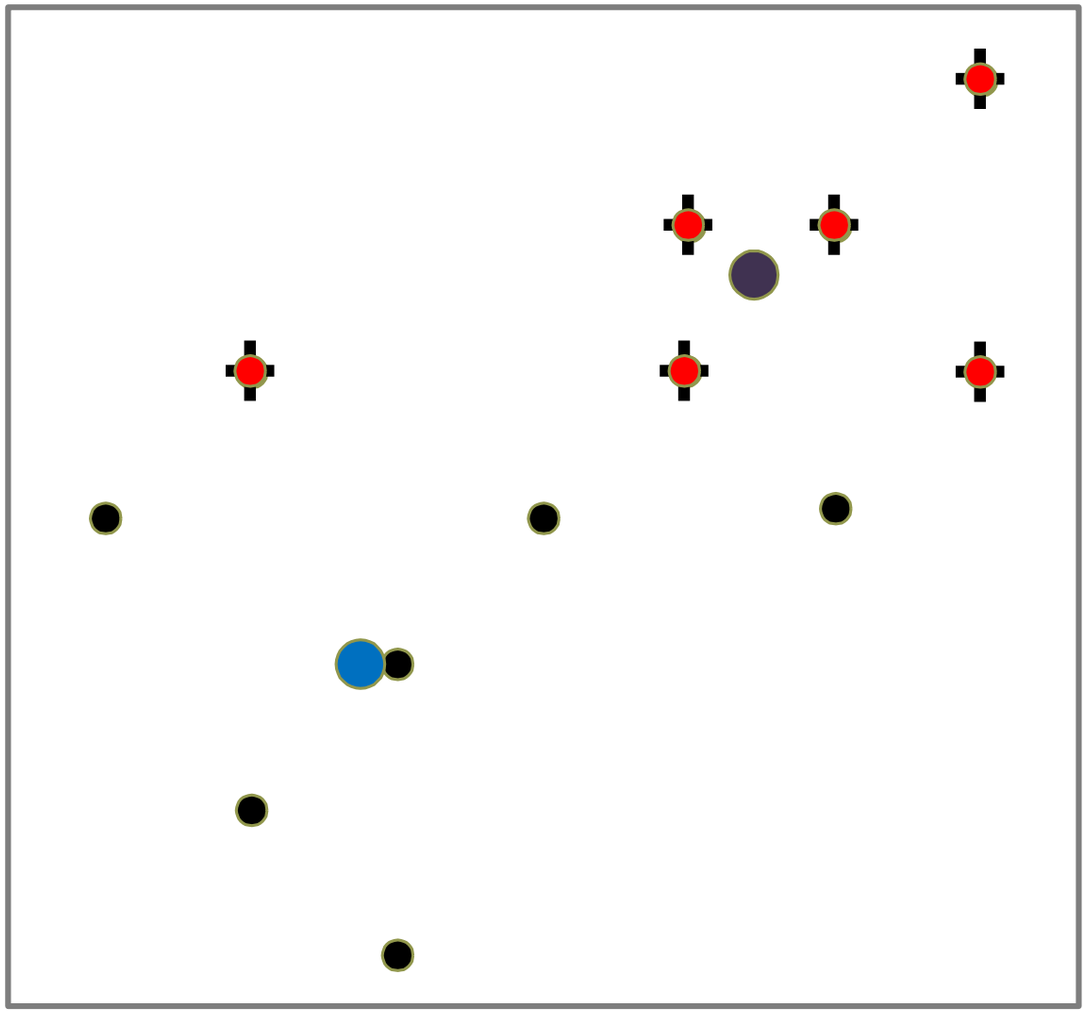
decide on k



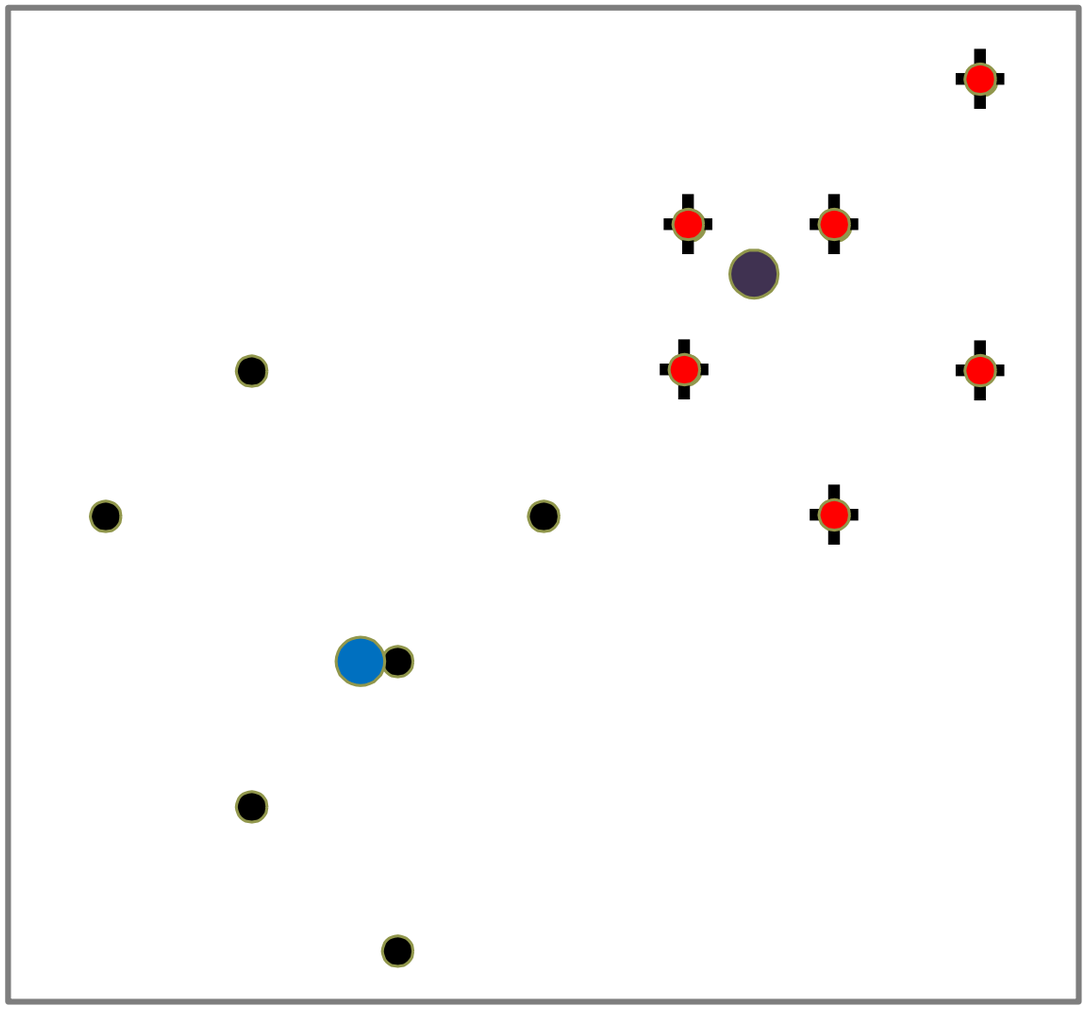
random assignment



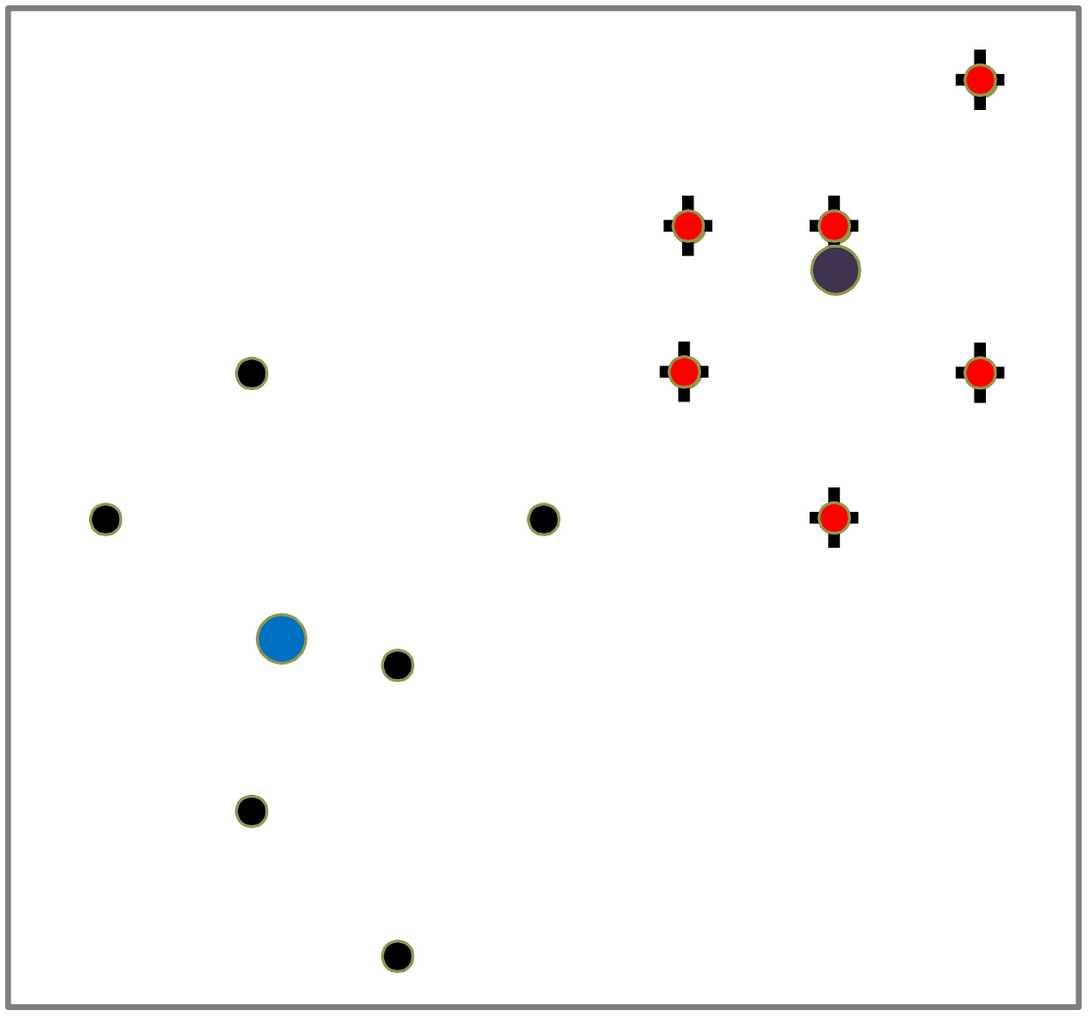
update cluster centroids



reassign objects



update cluster centroids

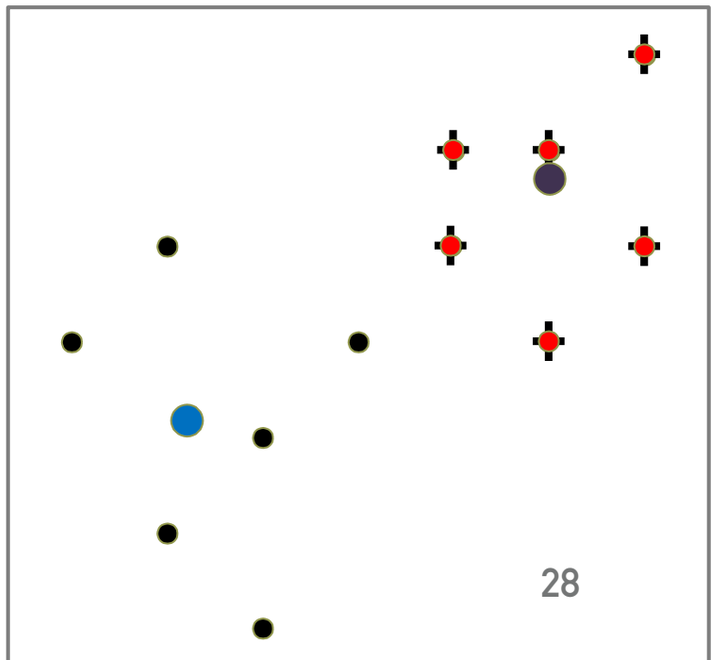


Loop as necessary

Efficient: **O**(tkn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t ≪ n.

k-means strengths

PAM: **O**(k(n-k)2), CLARA: **O**(ks^2 + k(n-k))



Often terminates at a local optimal

Not suitable to discover clusters with non-convex shapes

Applicable only to objects in a continuous n-dimensional space

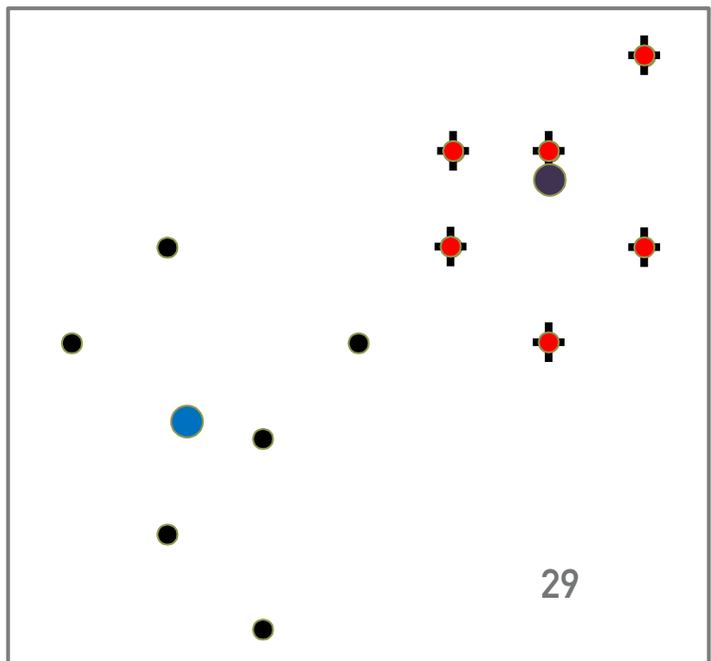
k-means weaknesses

Using the k-modes method for categorical data

In comparison, k-medoids can be applied to a wide range of data

how to specify k?

Sensitive to noisy data and outliers



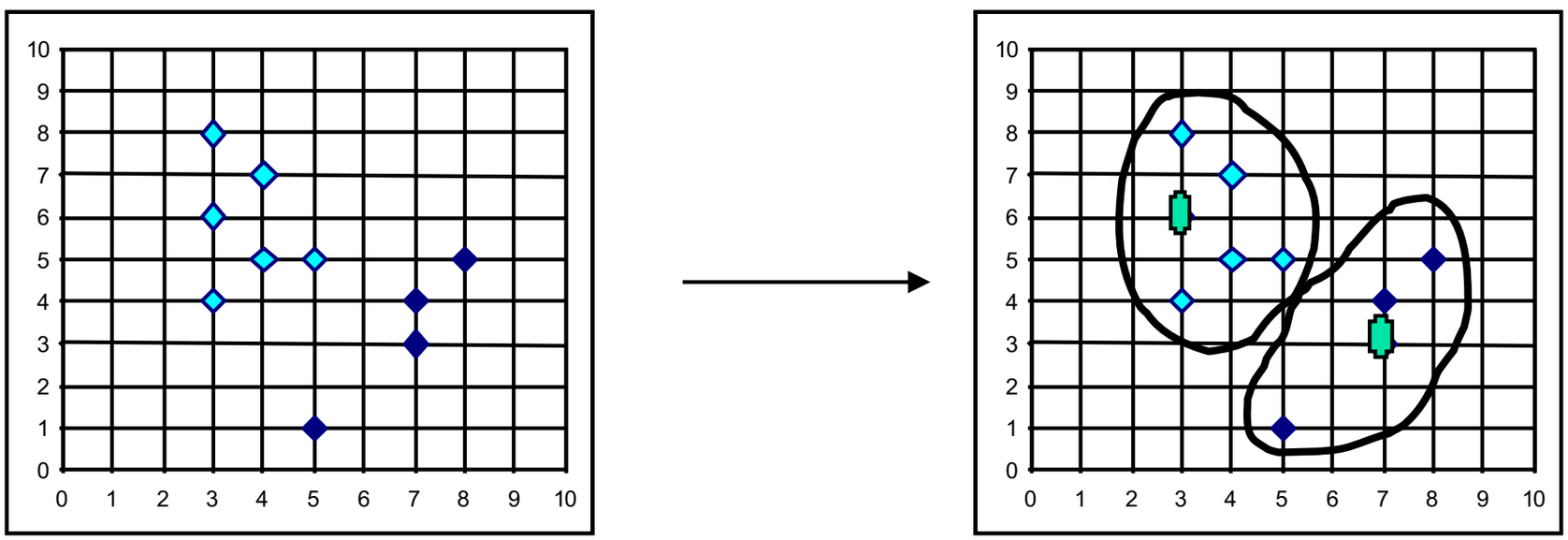
what is the computational complexity of determining the **global optimum?**

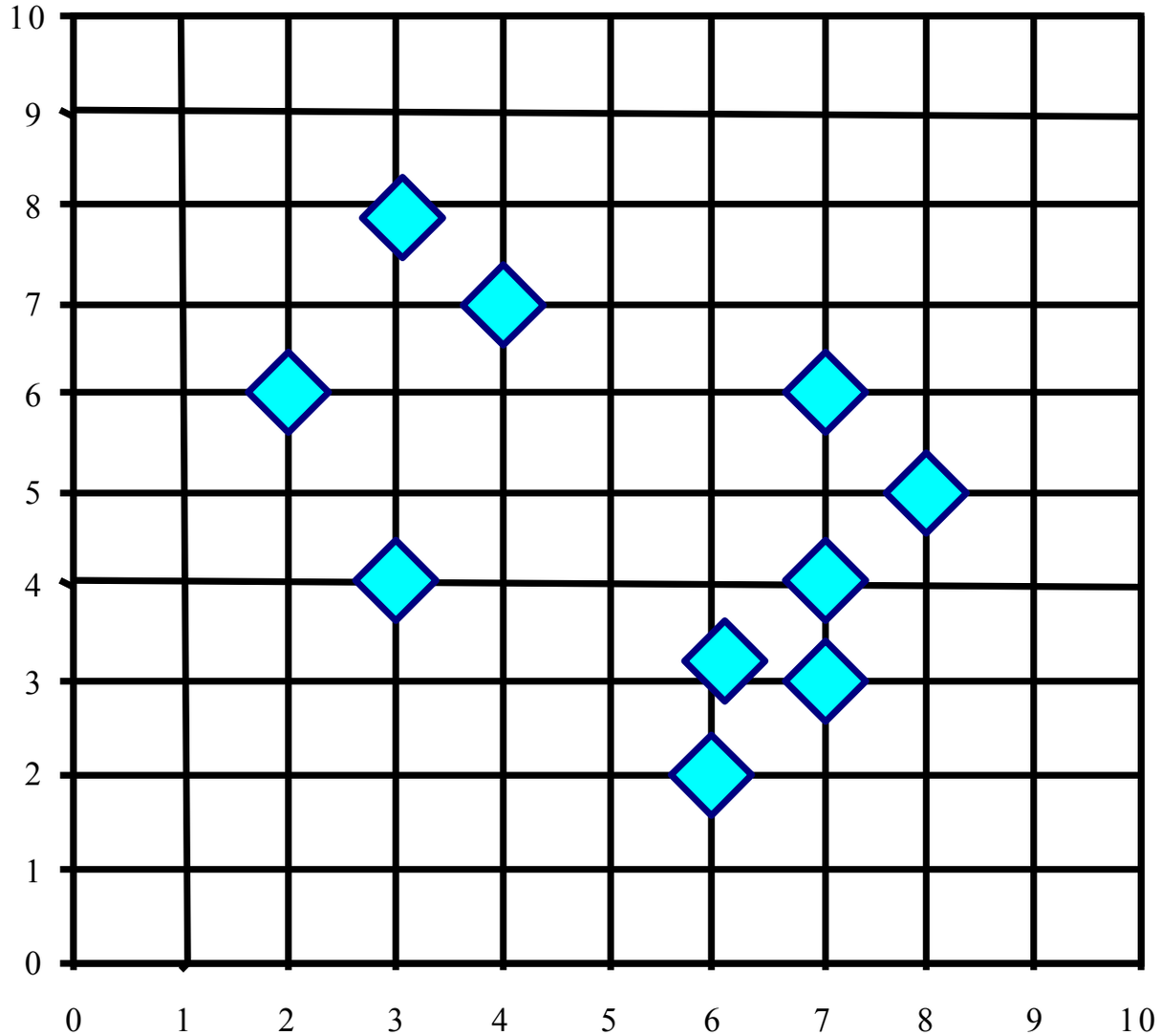
VARIANTS

- Principal Differences
  - Selection of the initial k means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: k-modes
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: k-prototype method

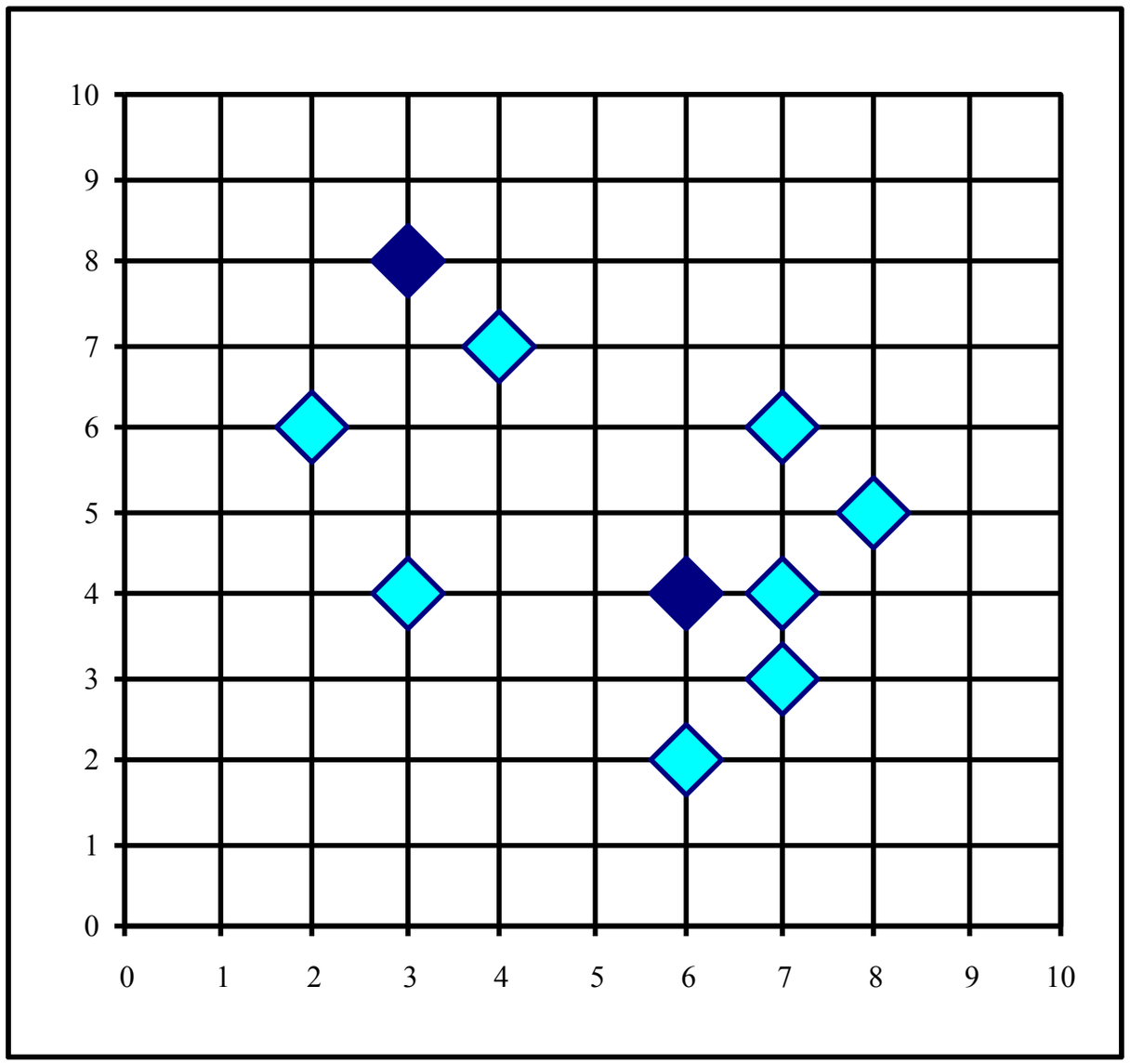
why is **k-means** so sensitive to noise?

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

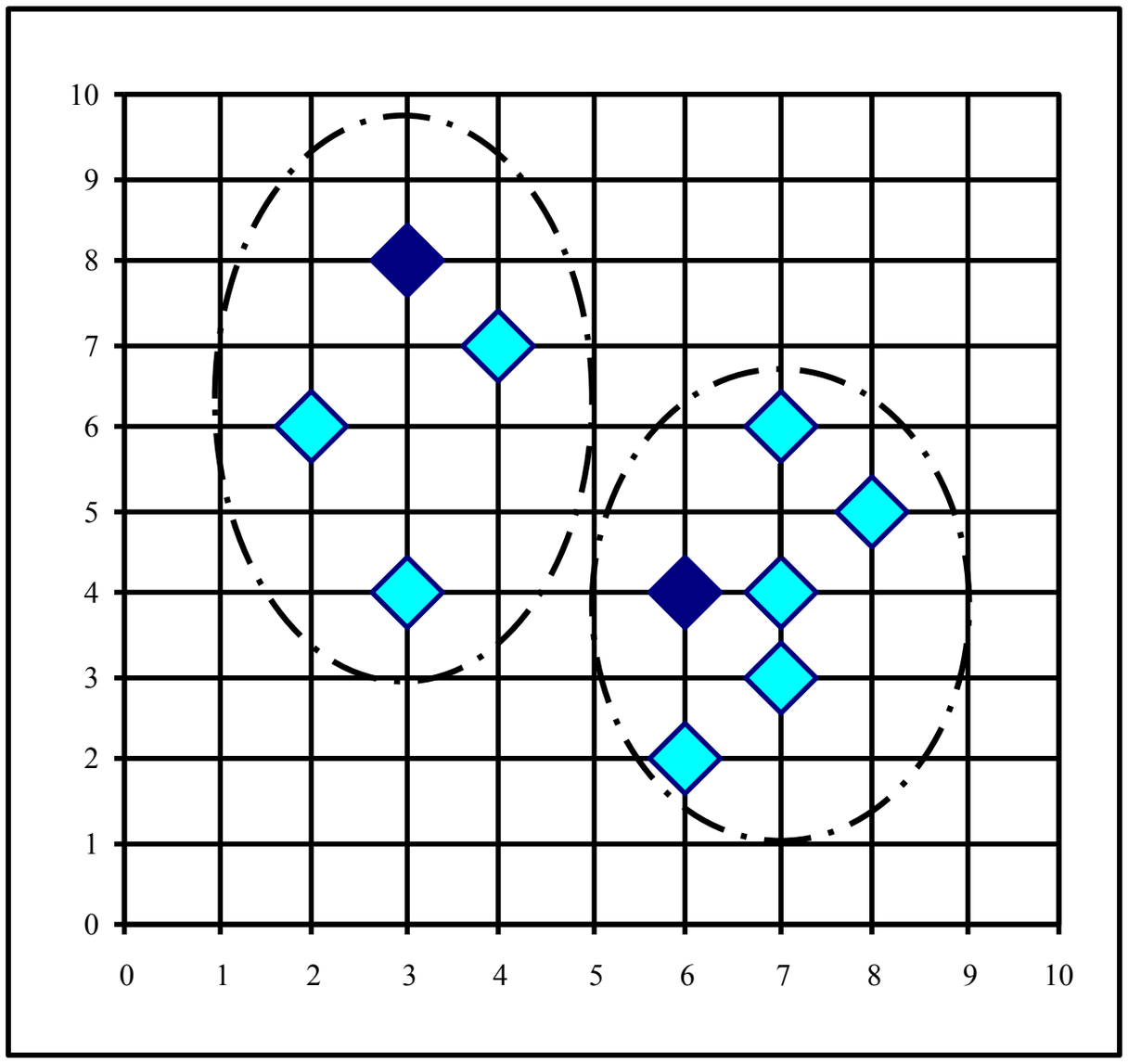


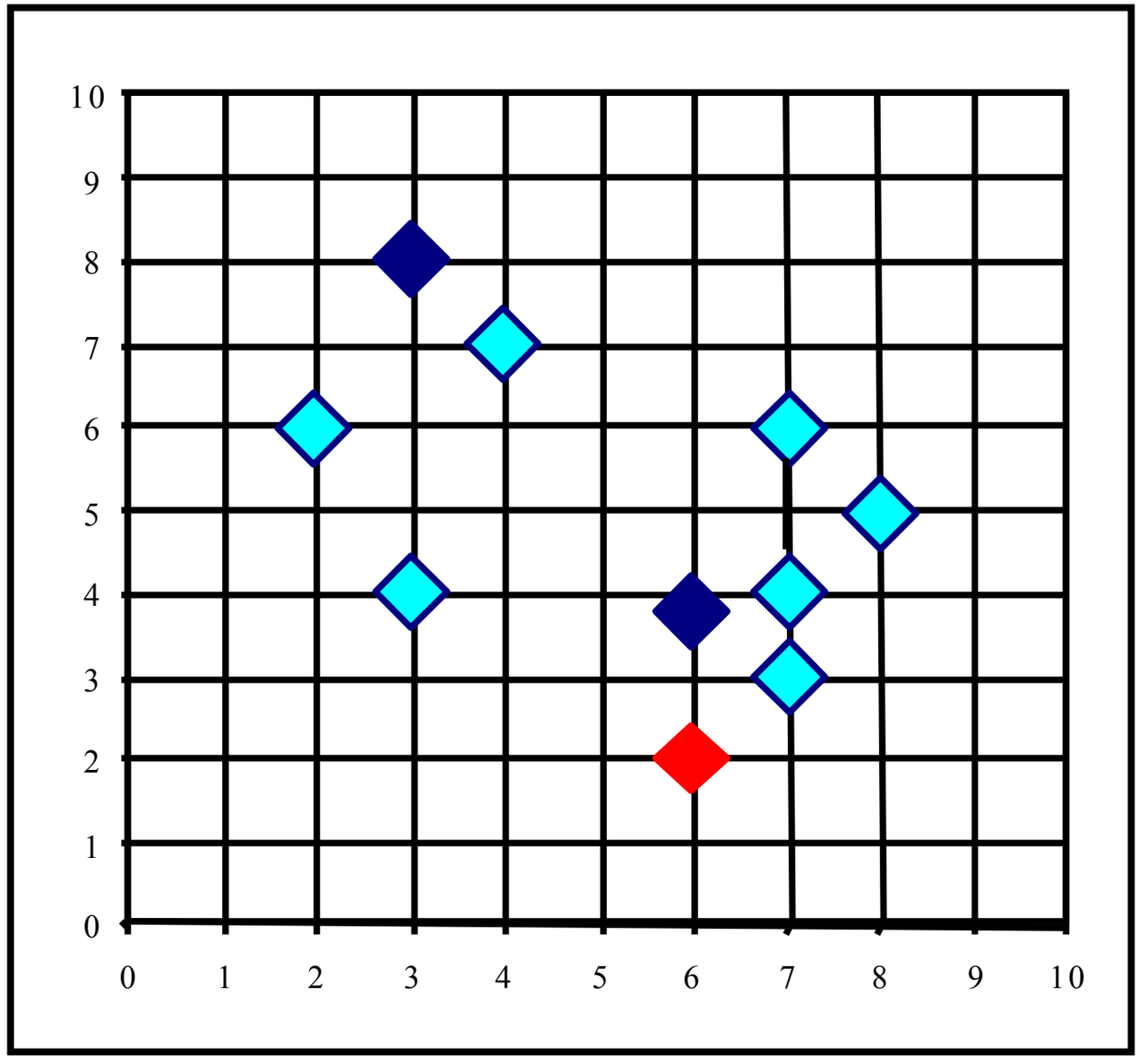

10
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9
10

Arbitrary choose k object as initial medoids



Assign each remaining object to nearest medoids



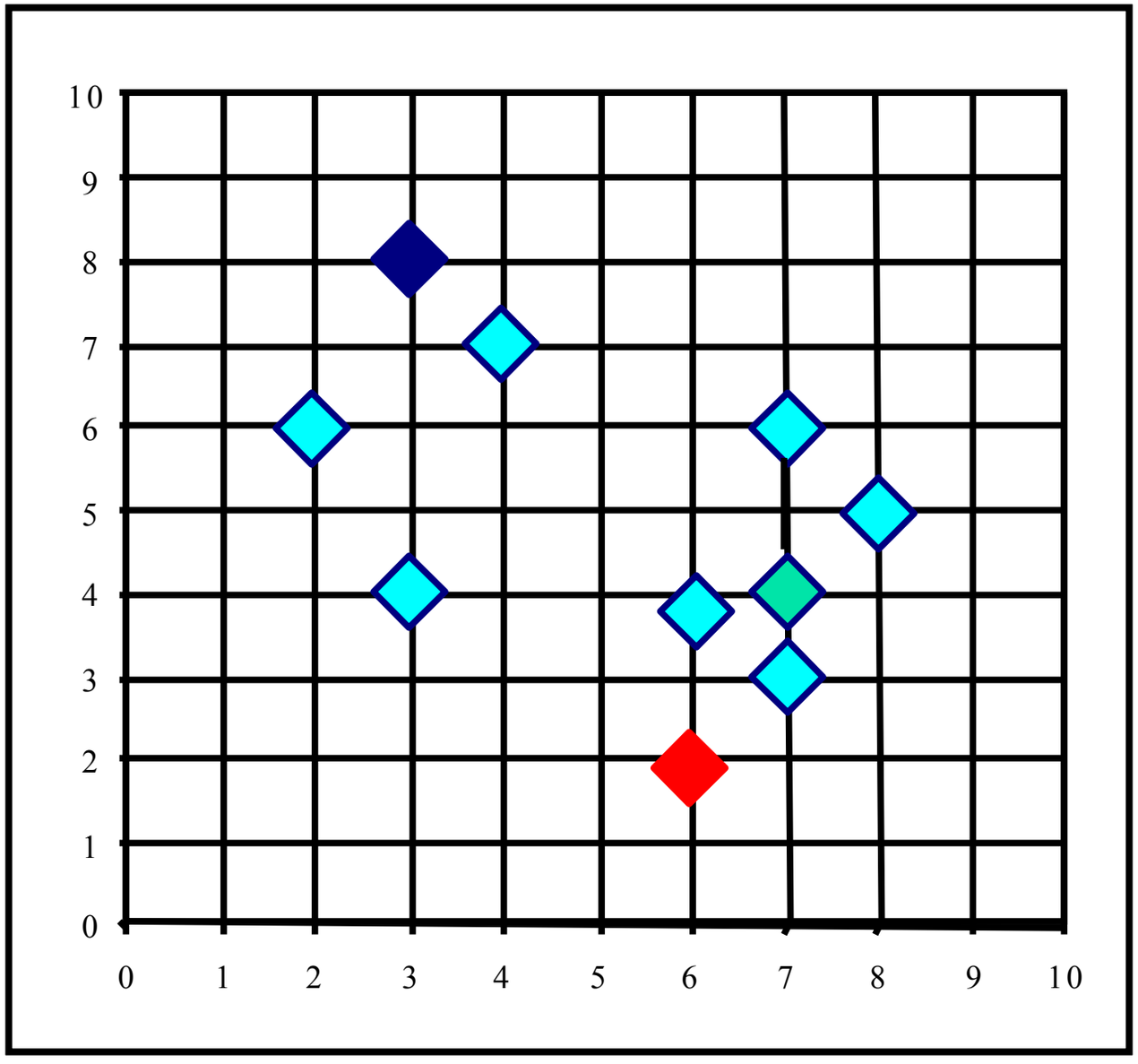


select a non-mediod object; all replacements are tried

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

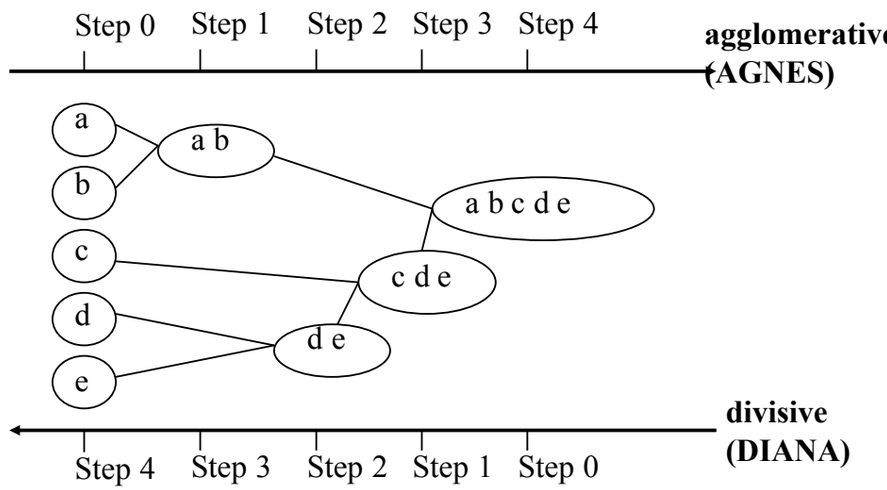
CLARA (Kaufmann & Rousseeuw, 1990): PAM on samples

CLARANS (Ng & Han, 1994): Randomized re-sampling



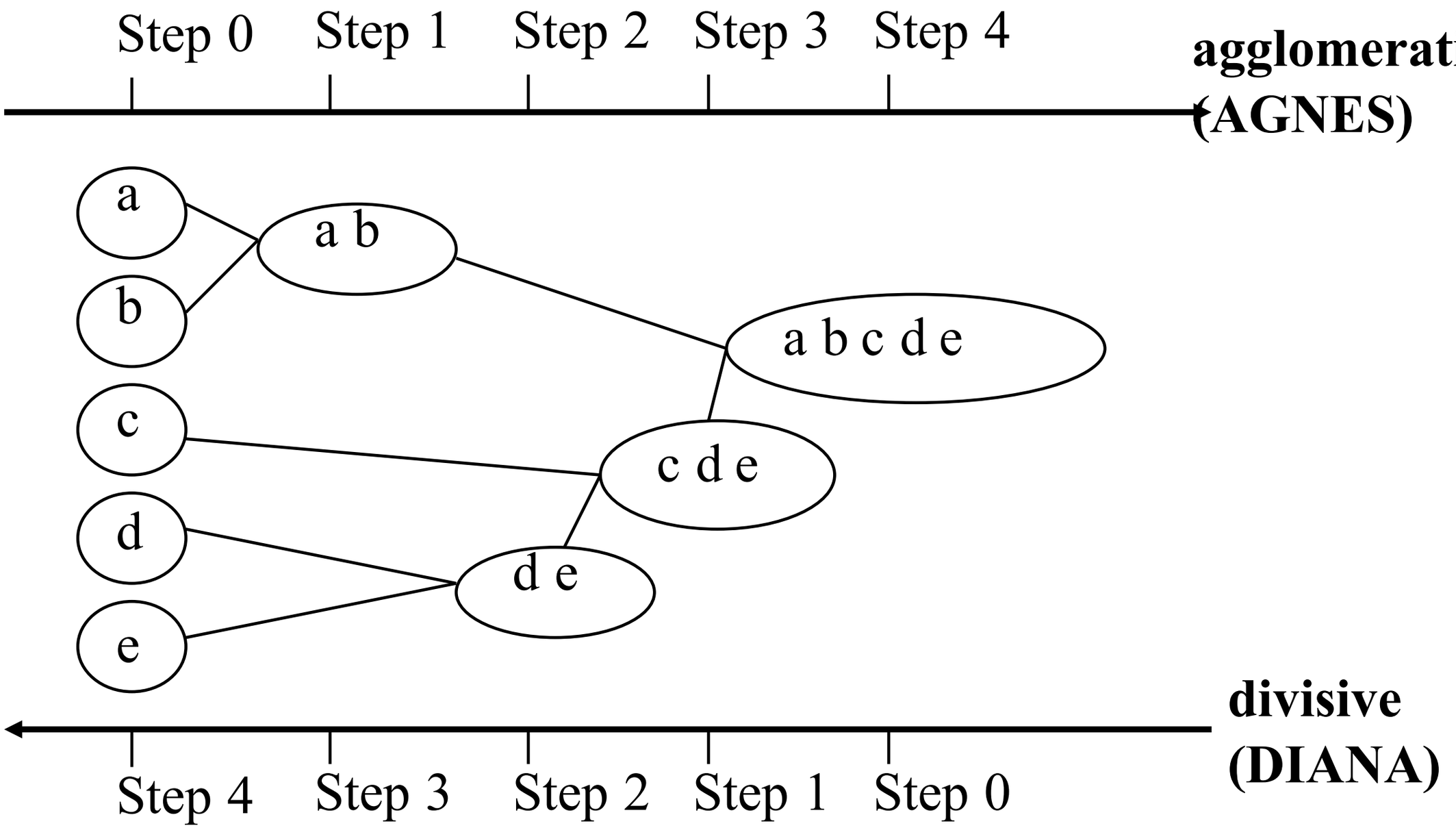
if the swap results in a better clustering, keep

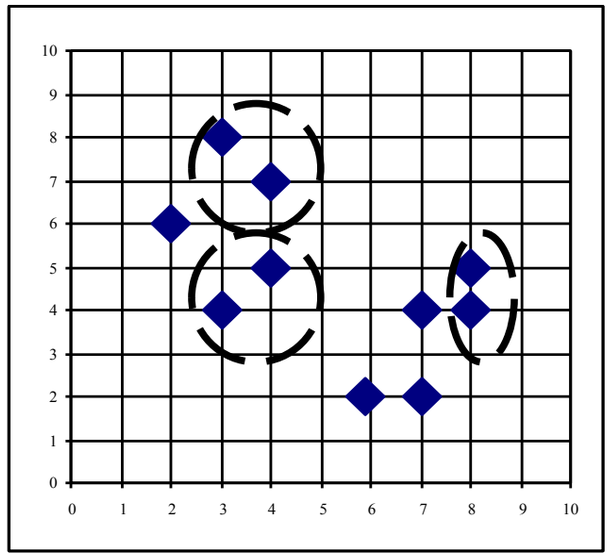
HIERARCHICAL APPROACHES

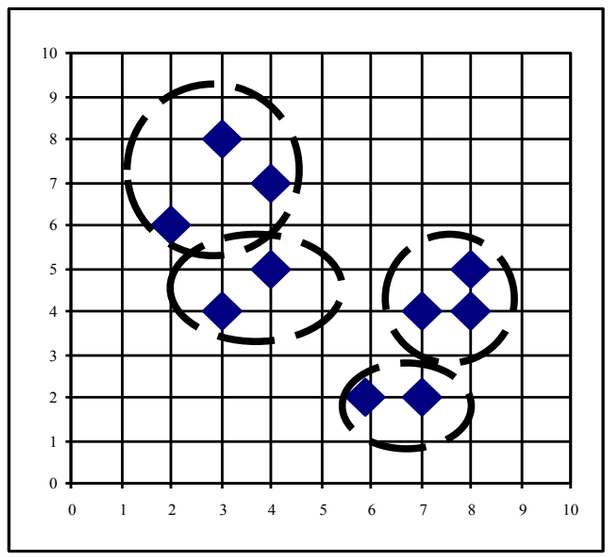


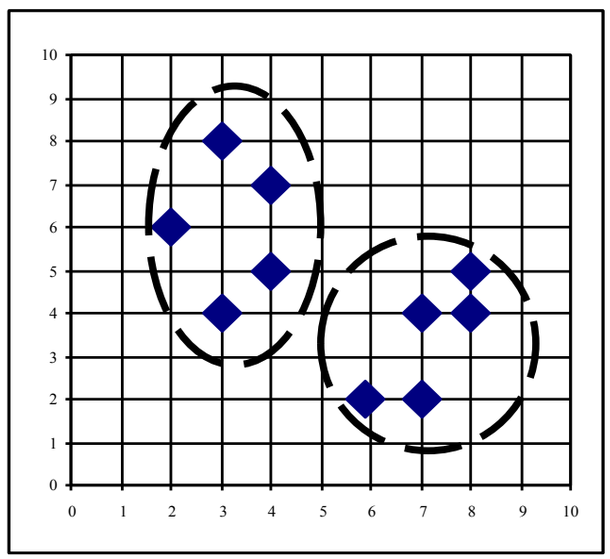
hierarchical approaches

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


Step 0
Step 1
Step 2
Step 3
Step 4
agglomerative
(AGNES)
a
b
c
d
e
a b
d e
c d e
a b c d e
divisive
(DIANA)
Step 4
Step 3
Step 2
Step 1
Step 0







AGNES

Introduced in Kaufmann and Rousseeuw (1990)

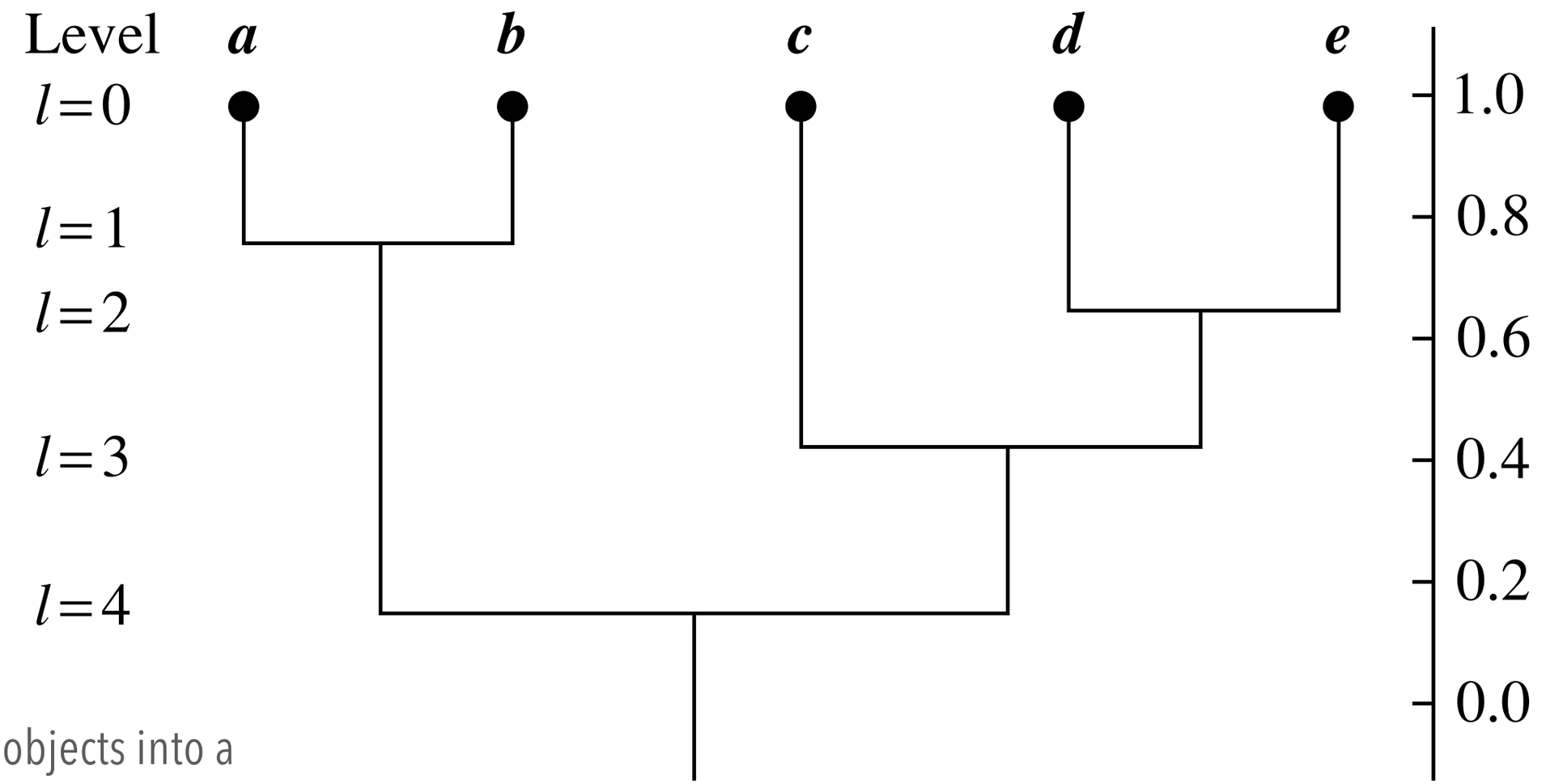
Implemented in statistical packages, e.g., Splus

Use the single-link method and the dissimilarity matrix

Merge nodes that have the least dissimilarity

Go on in a non-descending fashion

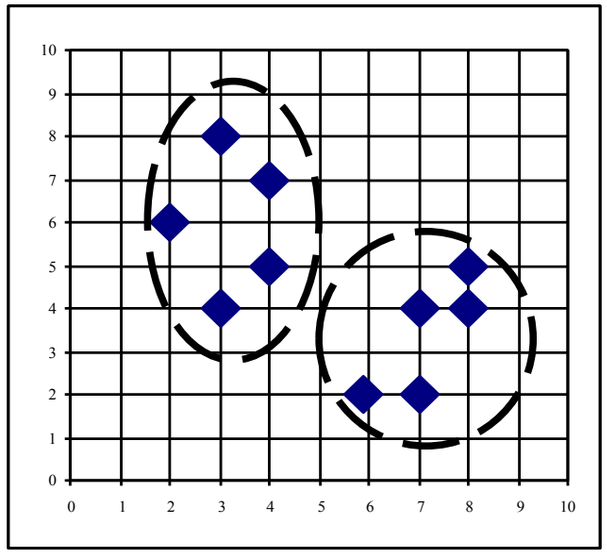
Eventually all nodes belong to the same cluster

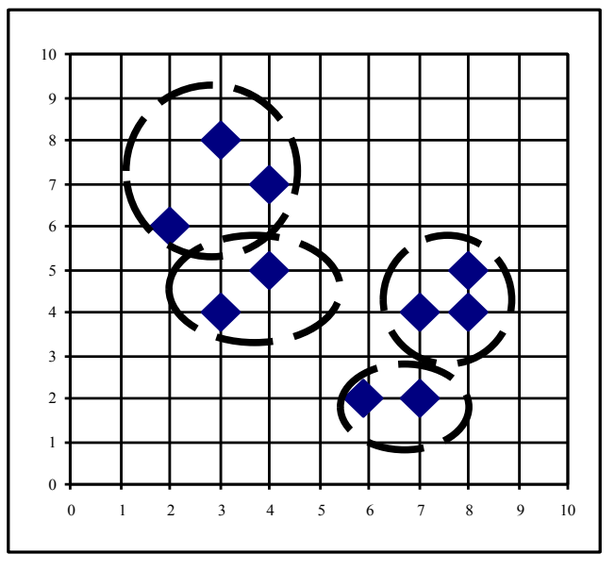


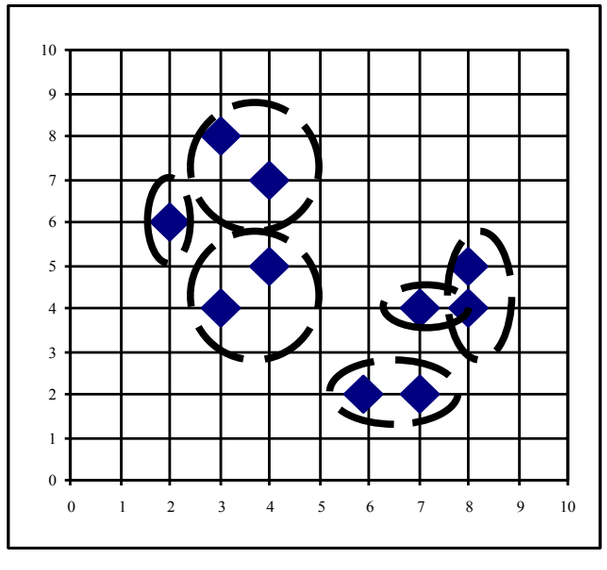
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

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

dendograms







DIANA

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

Minimum distance: $dist_{min}(C_i, C_j) = \min_{\boldsymbol{p} \in C_i, \boldsymbol{p}' \in C_j} \{|\boldsymbol{p} - \boldsymbol{p}'|\}$

Maximum distance: $dist_{max}(C_i, C_j) = \max_{\boldsymbol{p} \in C_i, \boldsymbol{p}' \in C_j} \{|\boldsymbol{p} - \boldsymbol{p}'|\}$

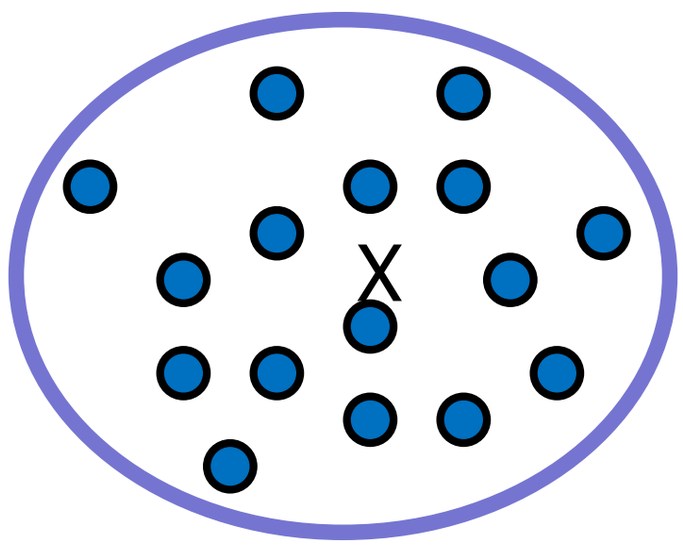
Mean distance: $dist_{mean}(C_i, C_j) = |\boldsymbol{m_i} - \boldsymbol{m_j}|$

Average distance: $dist_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{\boldsymbol{p} \in C_i, \boldsymbol{p}' \in C_j} |\boldsymbol{p} - \boldsymbol{p}'|$

distance measures

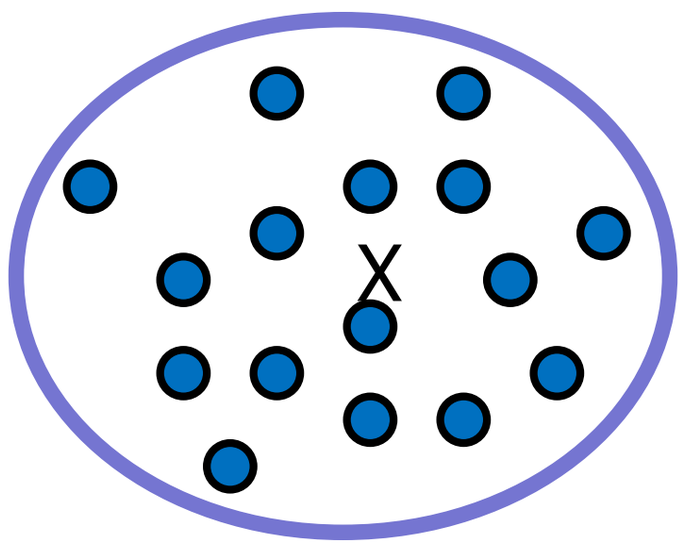
$$C_m = \frac{\sum_{i=1}^{N} t_{ip}}{N}$$

centroid

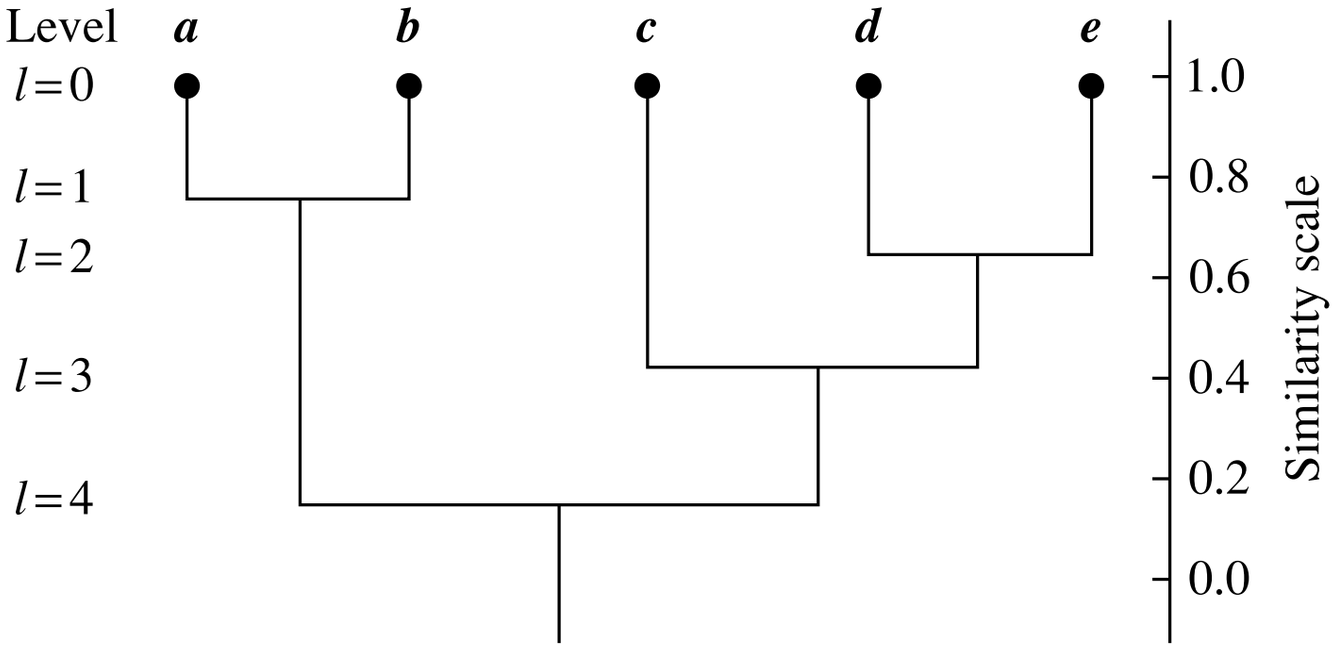


$$R_m = \sqrt{\frac{\sum_{i=1}^{N}(t_{ip} - c_m)^2}{N}}$$

radius



WEAKNESSES, EXTENSIONS



Can never undo what was done previously

Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects

Integration of hierarchical & distance-based clustering

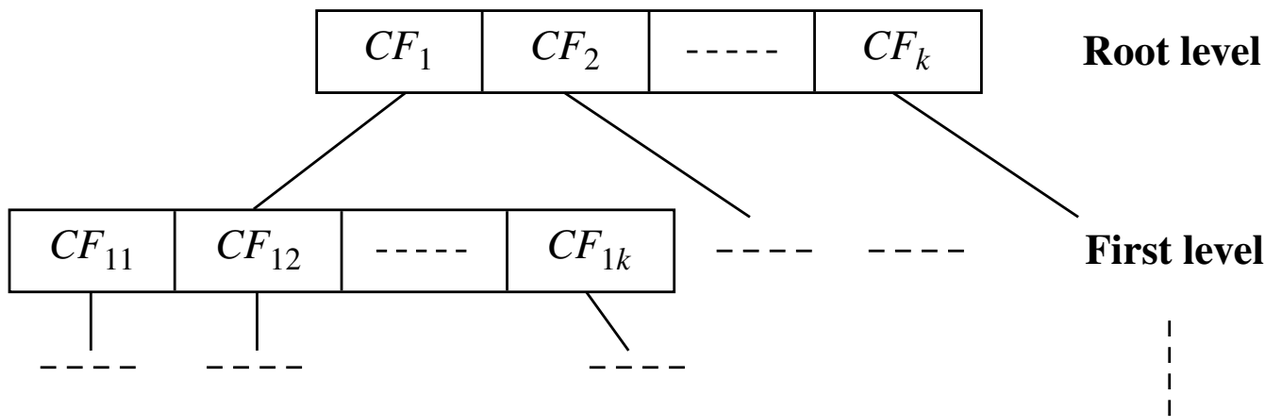
- BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
- CHAMELEON (1999): hierarchical clustering using dynamic modeling

BIRCH

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

key idea: use summary statistics

CF = (N, LS, SS)

$$x_0 = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{LS}{n}$$

number of items

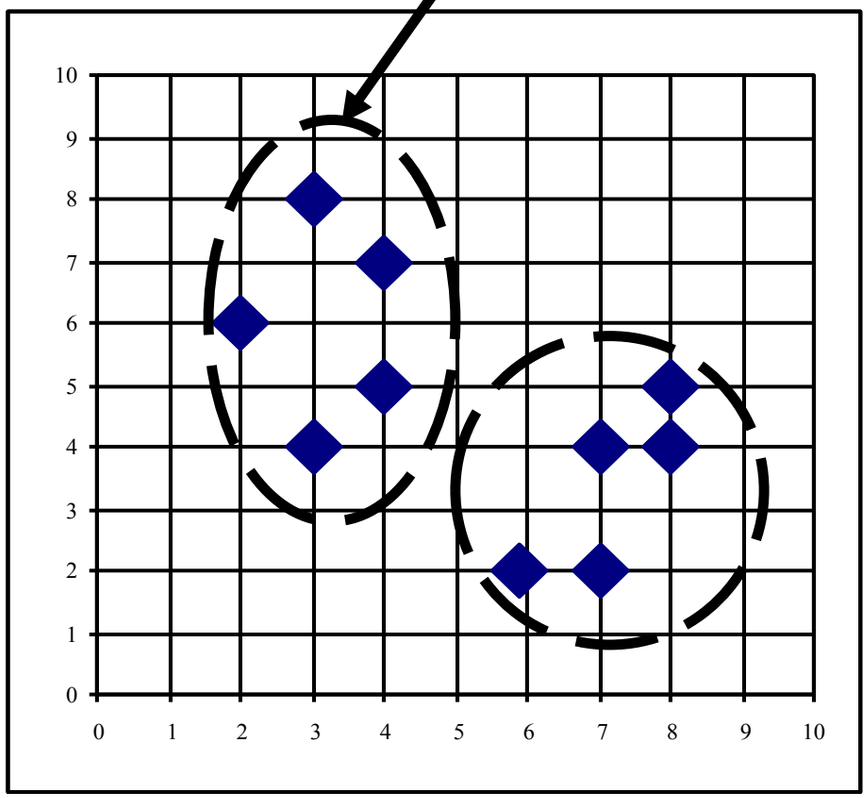
CF=(N, LS, SS)

linear sum sum of squares

$$R = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_0)^2}{n}} = \sqrt{\frac{nSS - 2LS^2 + nLS}{n^2}},$$

$$D = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2}{n(n-1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}}.$$

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



(3,4)

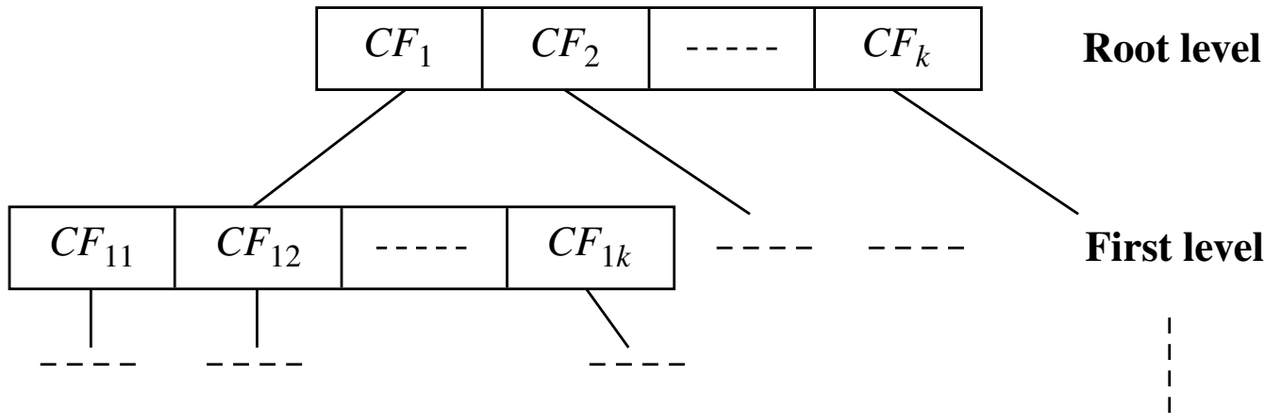
(2,6)

(4,5)

(4,7)

(3,8)

CLUSTERING FEATURE



Clustering feature:

- Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
- Registers crucial measurements for computing cluster and utilizes storage efficiently

A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering

- A nonleaf node in a tree has descendants or "children"
- The nonleaf nodes store sums of the CFs of their children

A CF tree has two parameters

- Branching factor: max # of children
- Threshold: max diameter of sub-clusters stored at the leaf nodes

BIRCH ALGORITHM

For each point in the input

Find closest leaf entry

Add point to leaf entry and update CF

If entry diameter > max_diameter, then split leaf, and possibly parents

Algorithm is O(n)

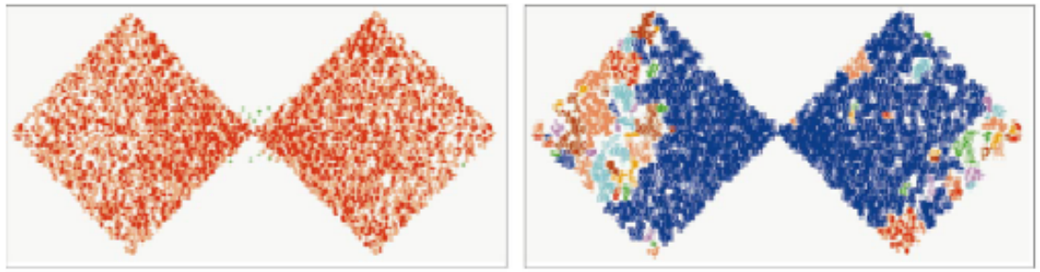
Concerns

Sensitive to insertion order of data points

Since we fix the size of leaf nodes, so clusters may not be so natural

Clusters tend to be spherical given the radius and diameter measures

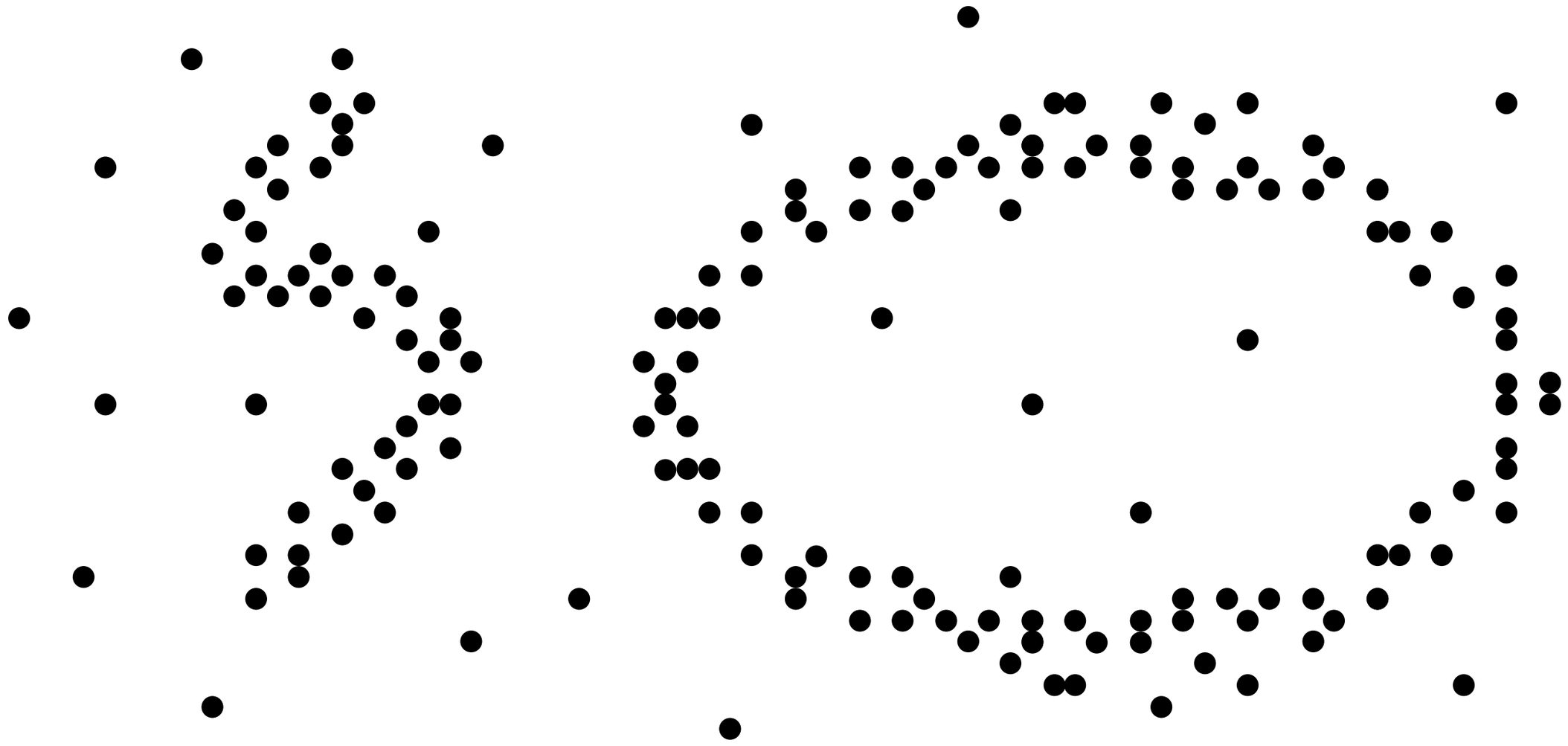
DENSITY BASED ALGORITHMS



Basic Concepts Partition Evaluation

Grid-Based Hierarchical Summary

Clustering based on density (local cluster criterion), such as density-connected points



Discover clusters of arbitrary shape

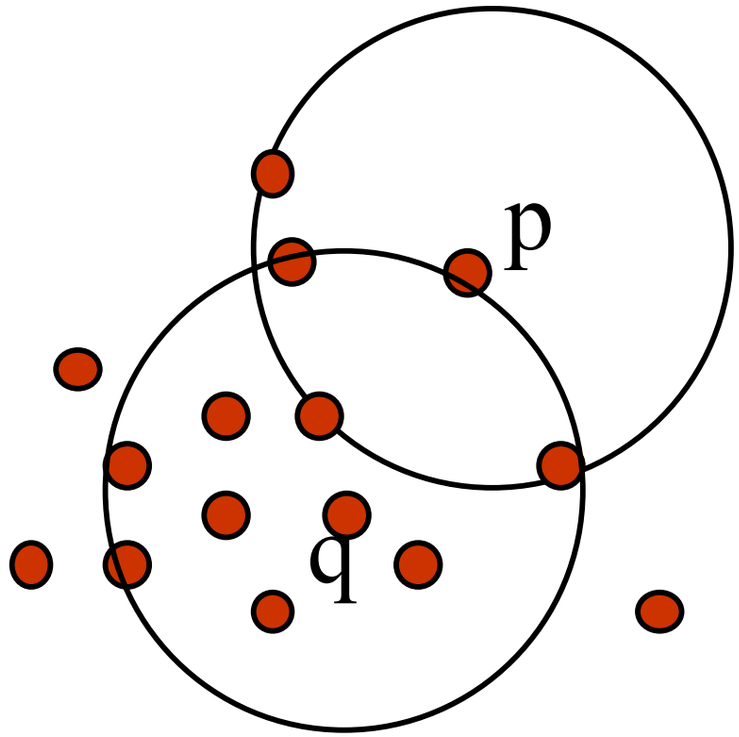
Handle noise

One scan

Need density parameters as termination condition

DBSCAN

Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996, August. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).



MinPts = 5

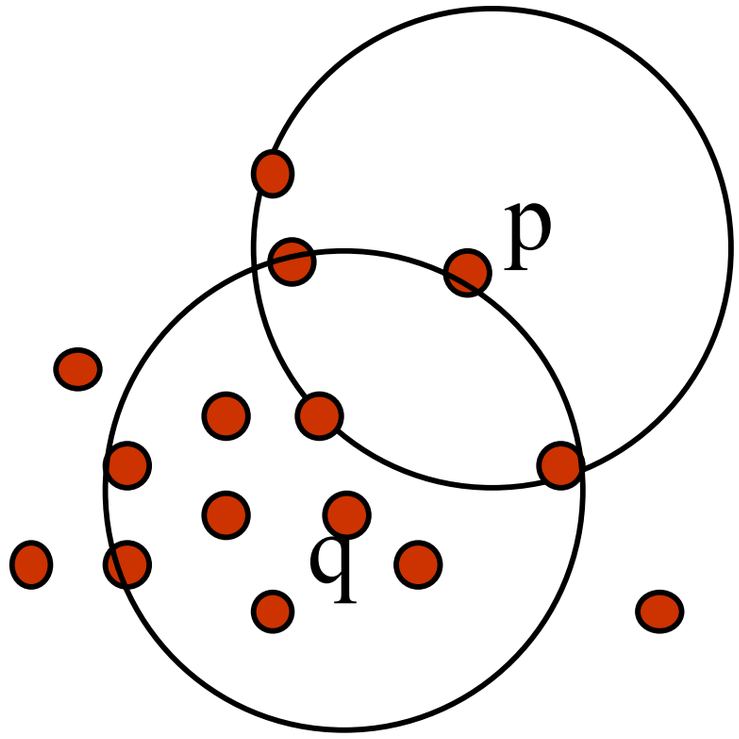
Eps = 1 cm

N_{Eps}(q): {p belongs to D | dist(p,q) ≤ Eps}

basic concepts

Eps: Maximum radius of the neighborhood

MinPts: Minimum number of points in an Eps-neighborhood of that point



MinPts = 5

Eps = 1 cm

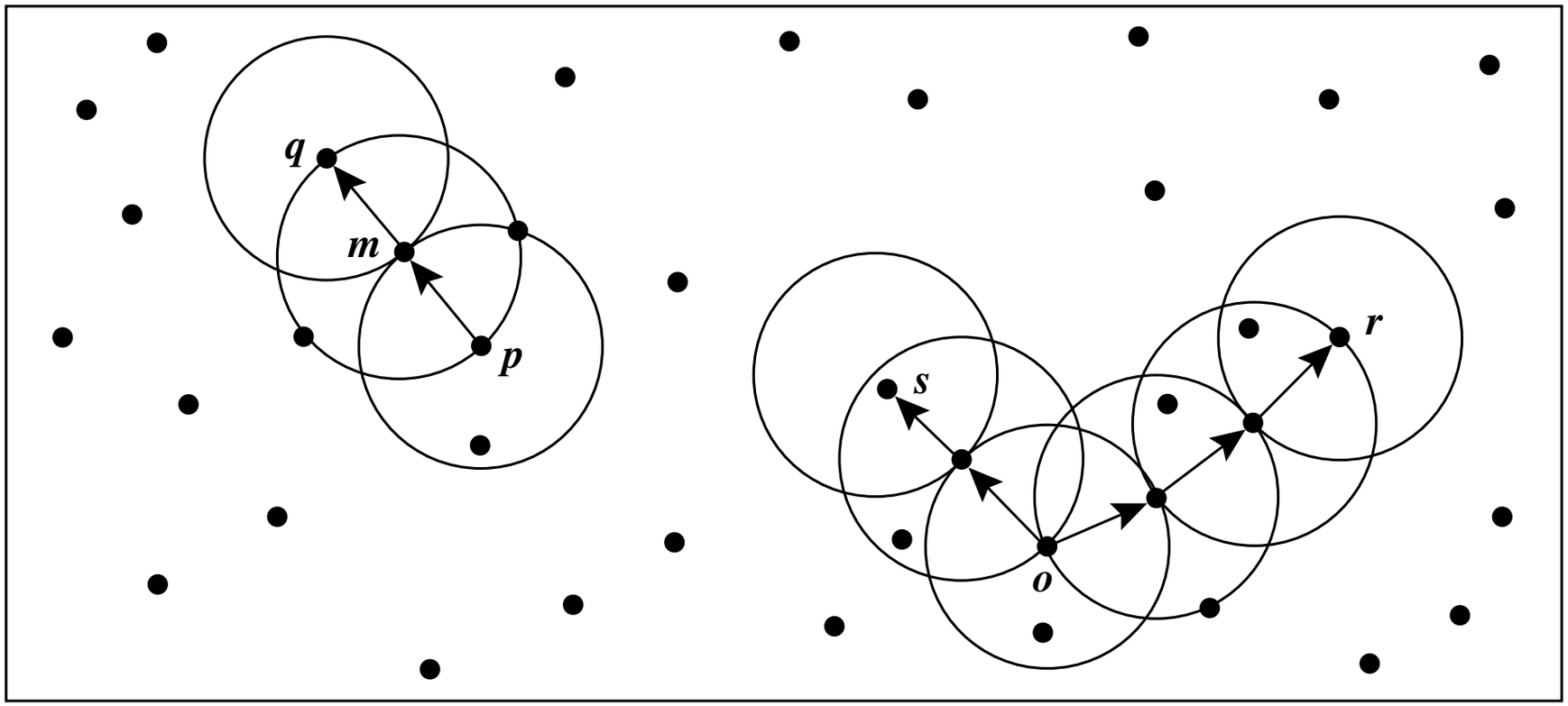
$N_{Eps}(q)$: {p belongs to D | dist(p,q) ≤ Eps}

core point, reachability

$|N_{Eps}(q)| \geq MinPts$

Directly density-reachable: A point p is directly density-reachable from a point q with respect to Eps, MinPts if p belongs to $N_{Eps}(q)$

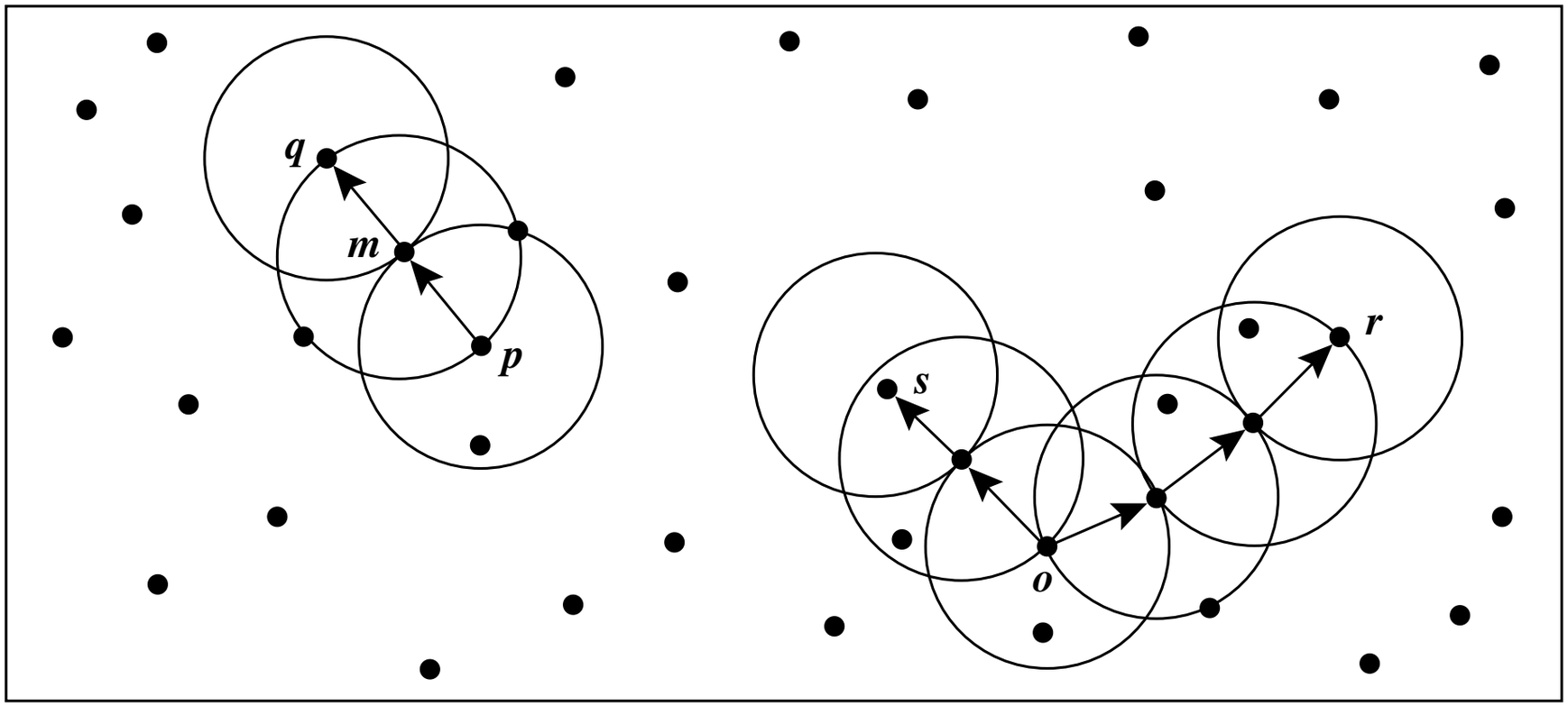
MinPts=3



density reachable

A point p is density-reachable from a point q with respect to Eps, 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

MinPts=3

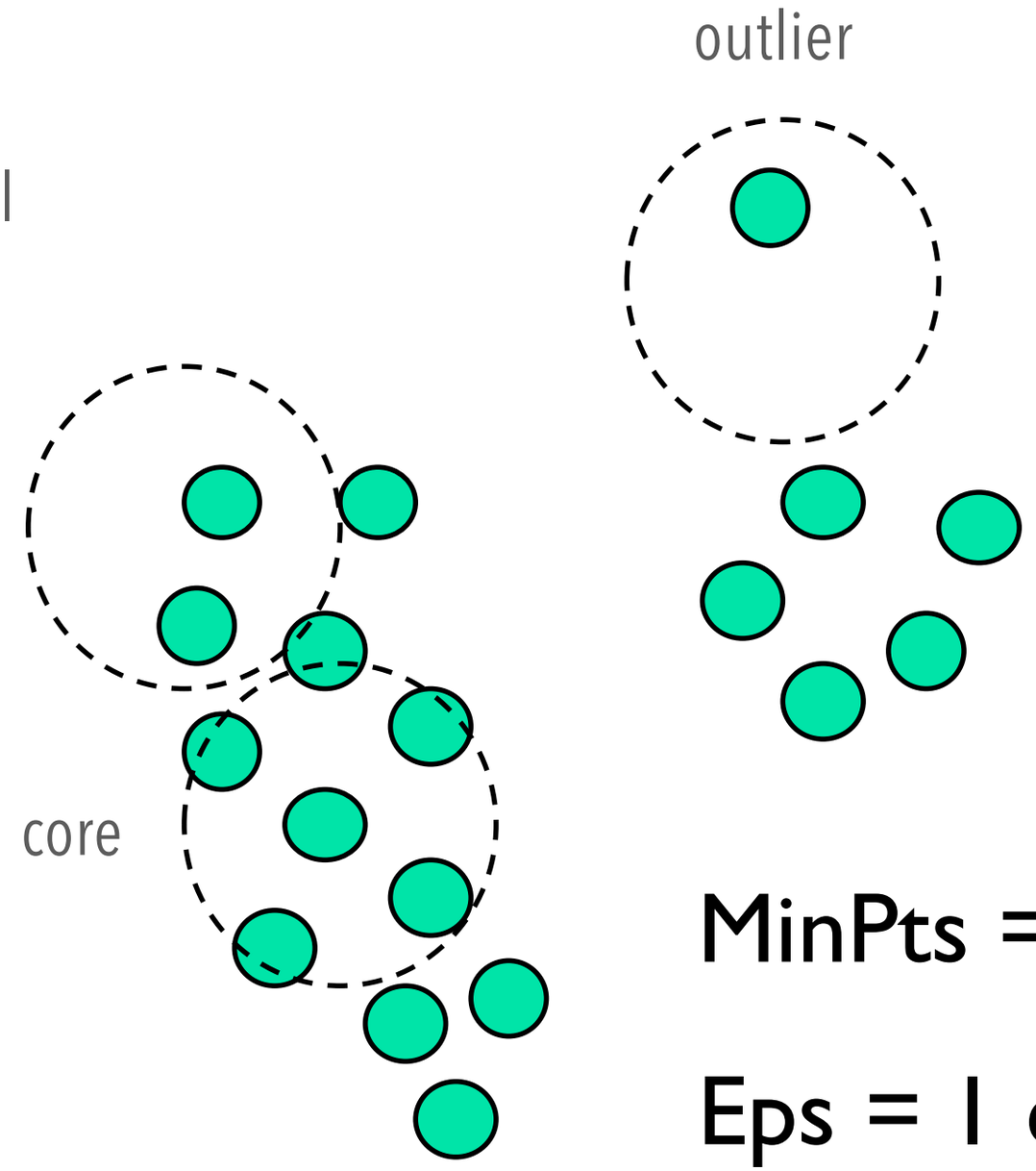


density connected

A point p is density-connected to a point q with respect to Eps, MinPts if there is a point o such that both, p and q are density-reachable from o with respect to Eps and MinPts

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

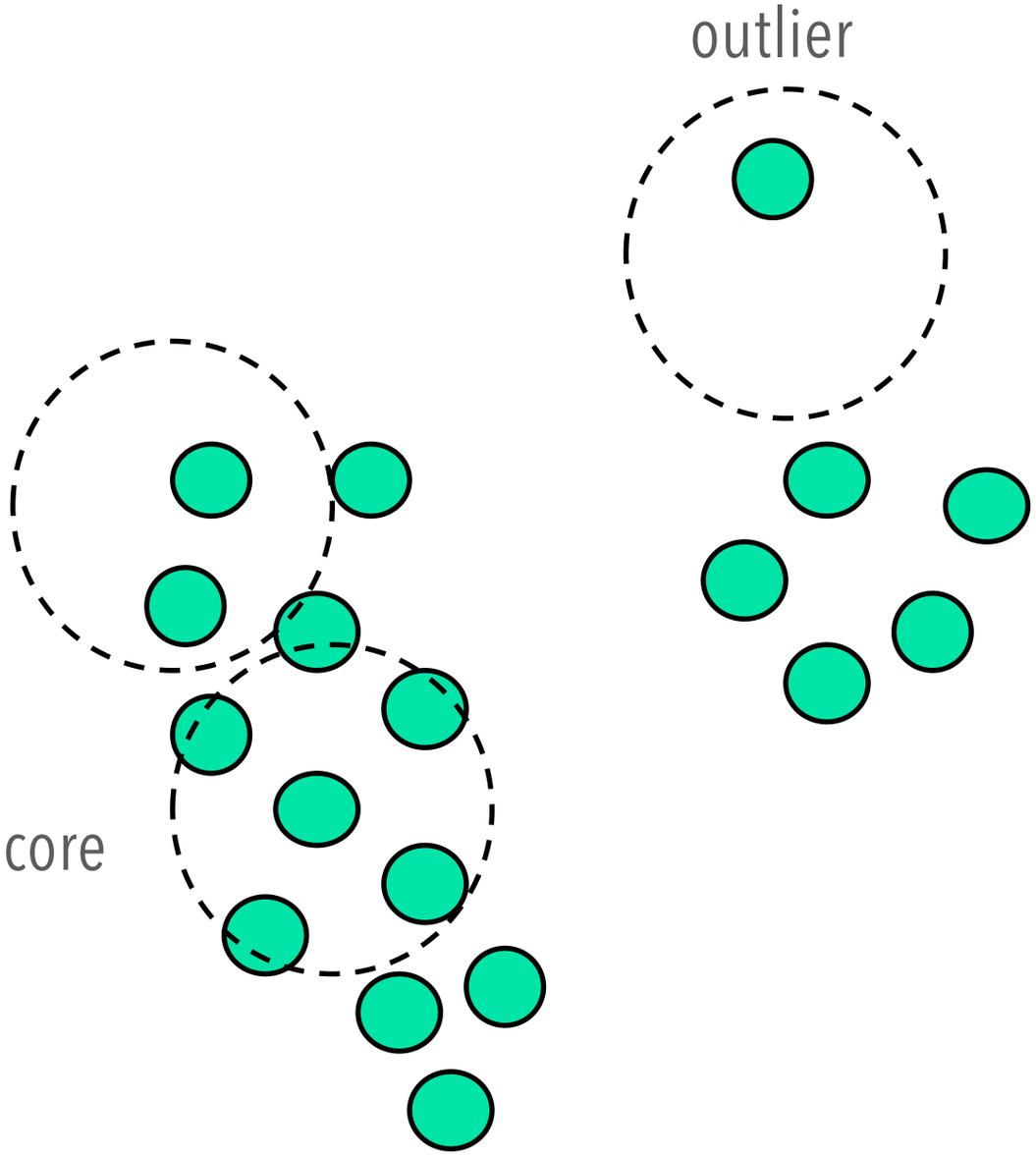


MinPts = 5

Eps = 1 cm

MinPts = 5

Eps = 1 cm



DBSCAN

Arbitrary select a point p

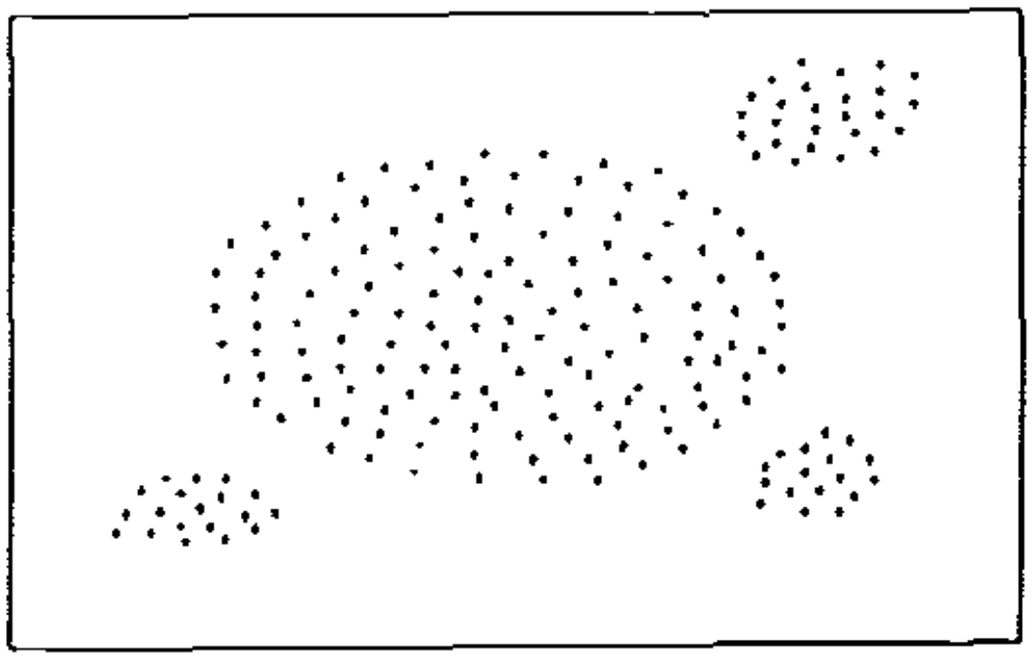
Retrieve all points density-reachable from p with respect to Eps and MinPts

If p is a core point, a cluster is formed

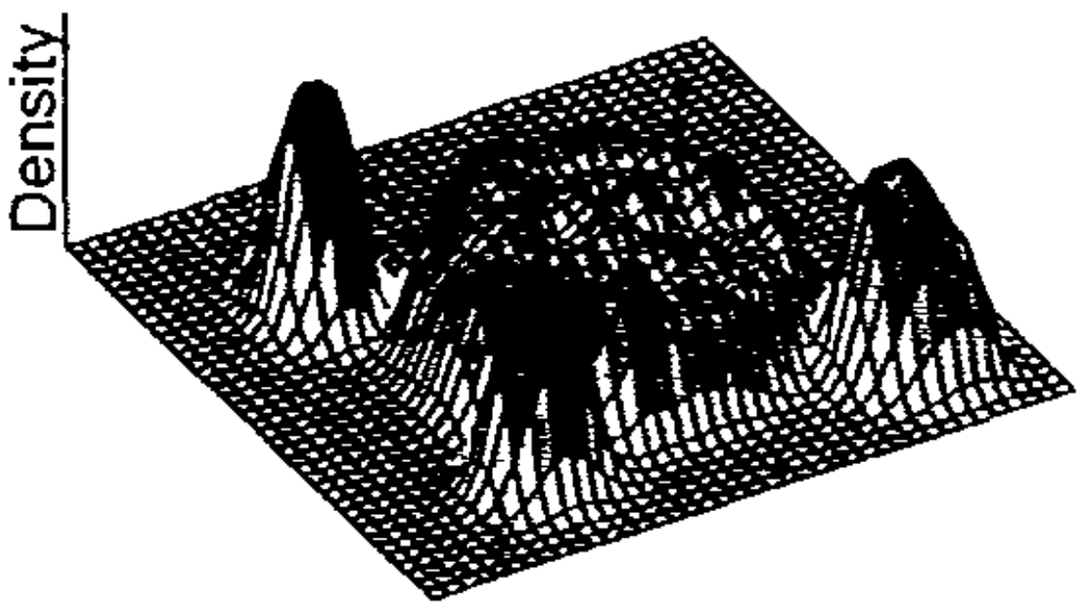
- All neighborhood points are candidates
- Add all unvisited neighborhood points to the cluster
- if a neighborhood point is a core point, add its neighborhood as candidates
- If a neighborhood point p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database

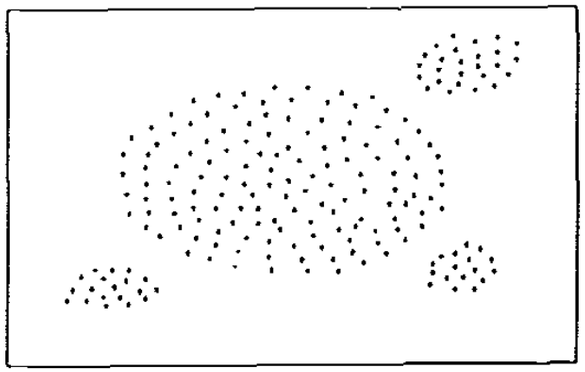
Continue the process until all of the points have been processed

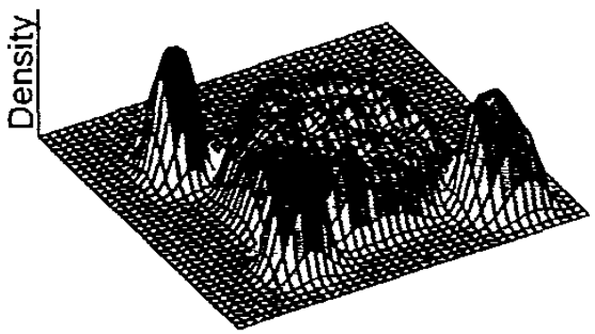
If a spatial index is used, the computational complexity of DBSCAN is O(n log n), where n is the number of database objects. Otherwise, the complexity is $O(n^2)$

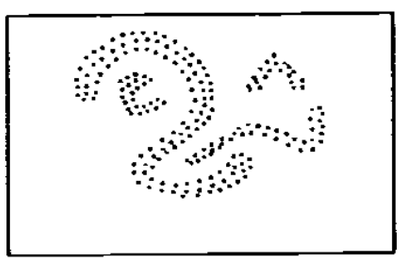


DENCLUE

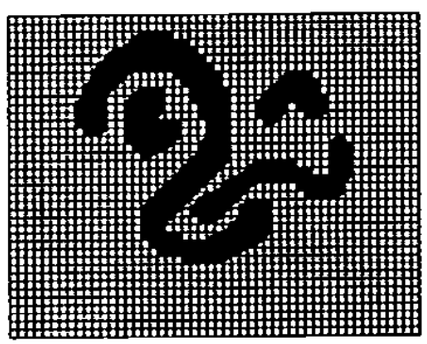








(a) DBSCAN



(b) DENCLUE

KEY IDEAS

Uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure

Influence function: describes the impact of a data point within its neighborhood

Overall density of the data space can be calculated as the sum of the influence function of all data points

Clusters can be determined mathematically by identifying density attractors

Density attractors are local maximal of the overall density function

Center defined clusters: assign to each density attractor the points density attracted to it

Arbitrary shaped cluster: merge density attractors that are connected through paths of high density (> threshold)

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

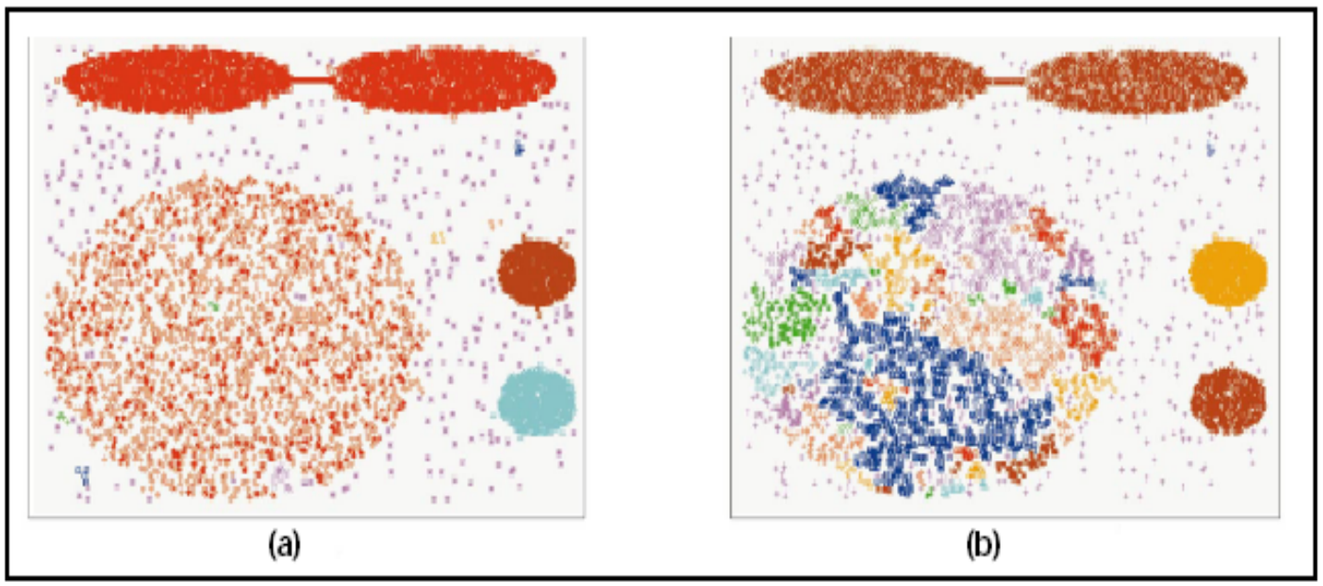
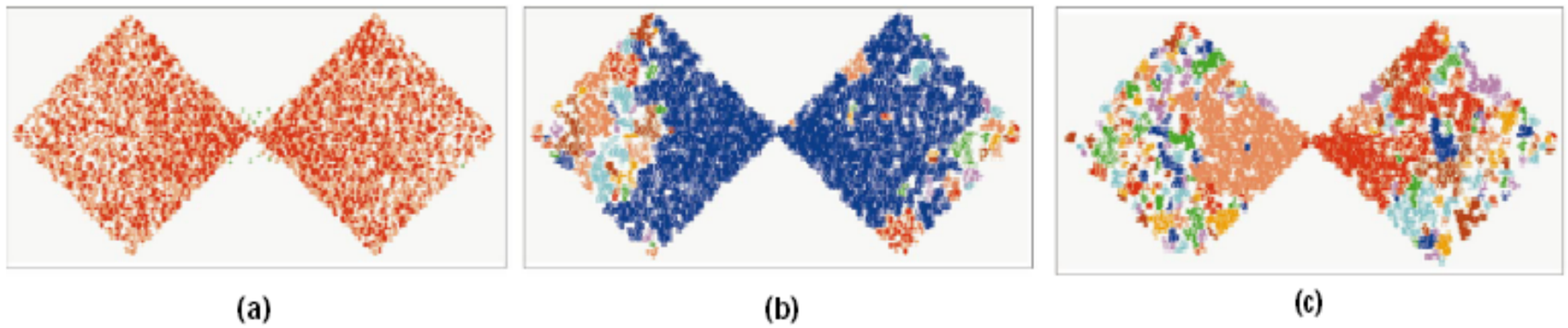


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



(a) (b) (c)

GRID BASED ALGORITHMS

Basic Concepts Partition Density

Evaluation Hierarchical Summary

CLIQUE

Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space

CLIQUE can be considered as both density-based and grid-based

It partitions each dimension into the same number of equal length interval

It partitions an m-dimensional data space into non-overlapping rectangular units

A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter

A cluster is a maximal set of connected dense units within a subspace

MAJOR STEPS

Partition the data space and find the number of points that lie inside each cell of the partition.

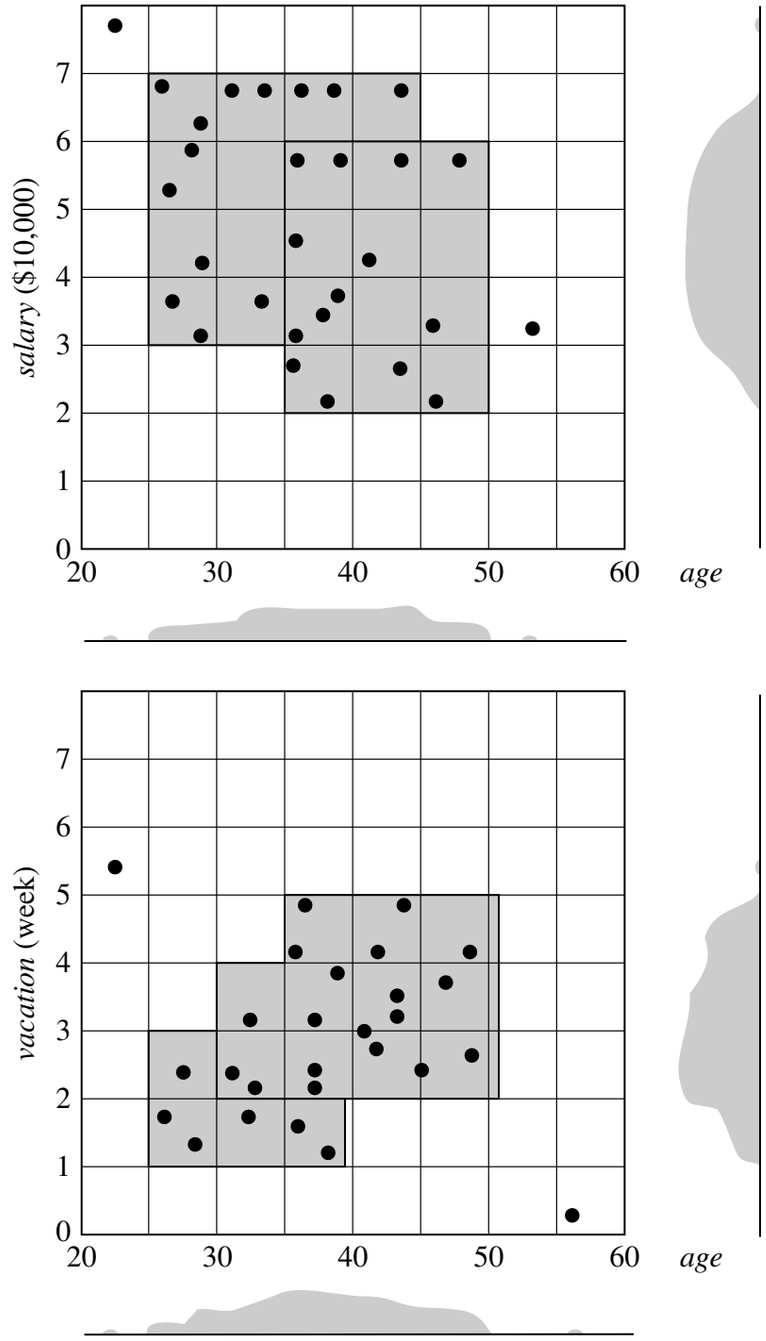
Identify the subspaces that contain clusters using the Apriori principle

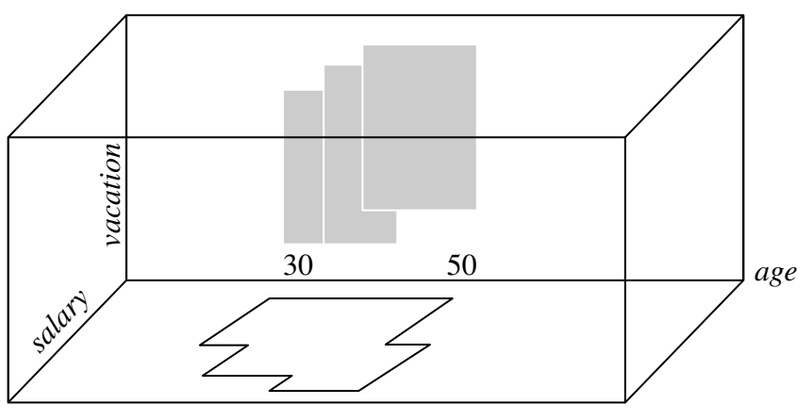
Identify clusters

- Determine dense units in all subspaces of interests
- Determine connected dense units in all subspaces of interests.

Generate minimal description for the clusters

- Determine maximal regions that cover a cluster of connected dense units for each cluster
- Determination of minimal cover for each cluster


salary ($10,000)
0
1
2
3
4
5
6
7
20
30
40
50
60
age
vacation (week)
0
1
2
3
4
5
6
7
20
30
40
50
60
age

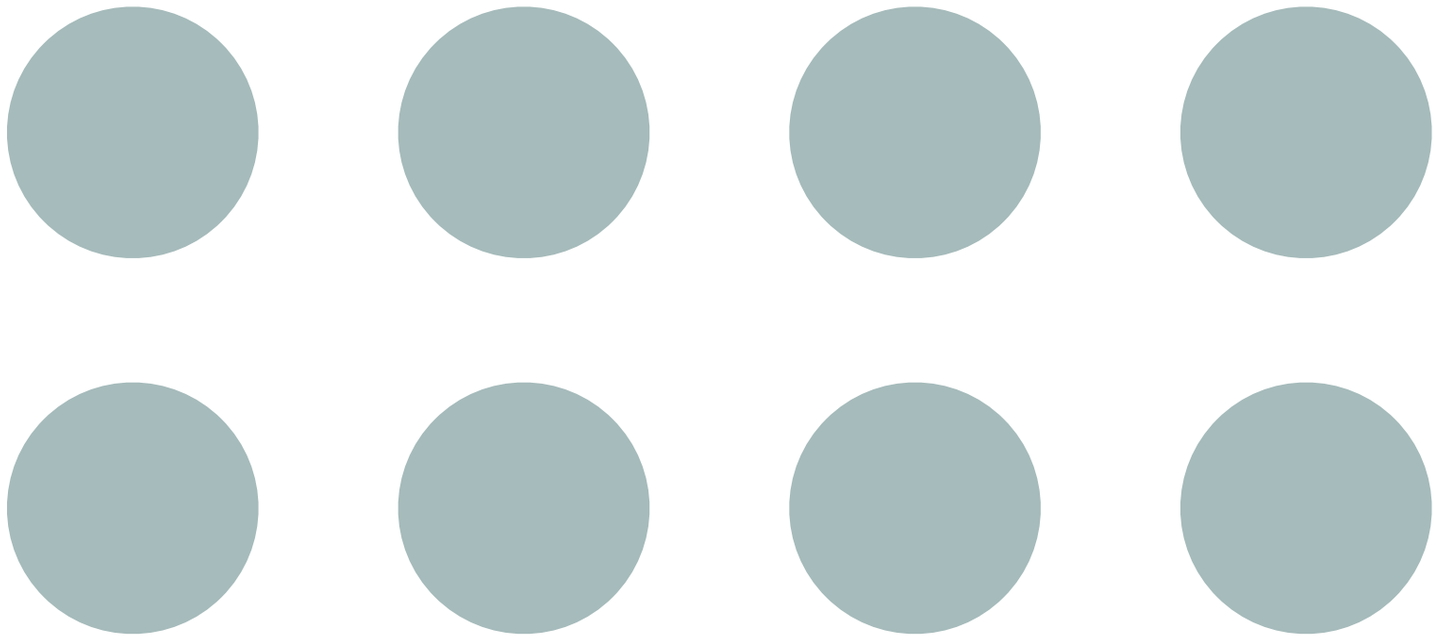

vacation
salary
30
50
age

"Your evaluation is based on the next 30 seconds. Go!"

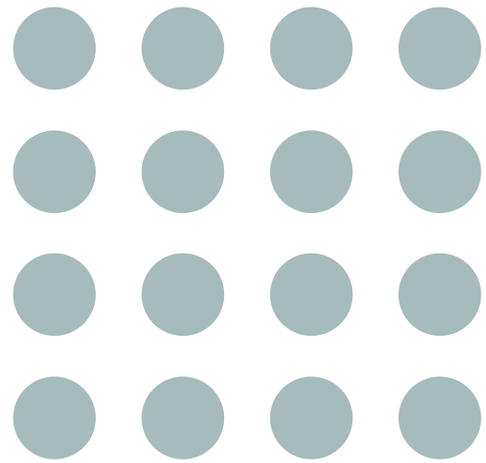
EVALUATION

Basic Concepts Partition Density-Based

Grid-Based Hierarchical Summary



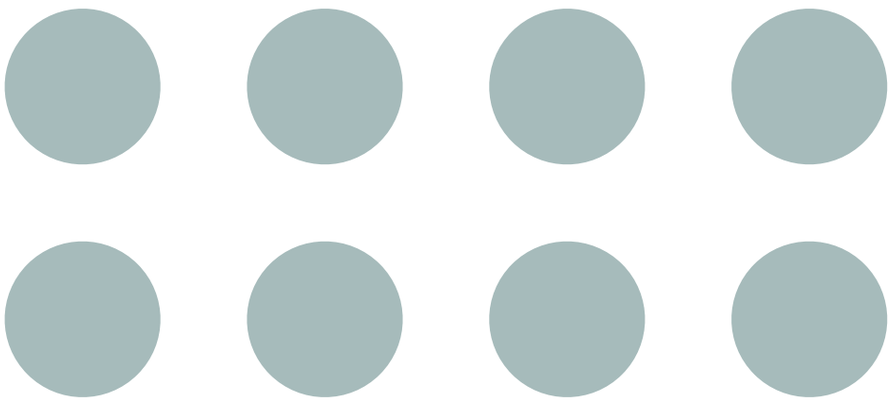
Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution

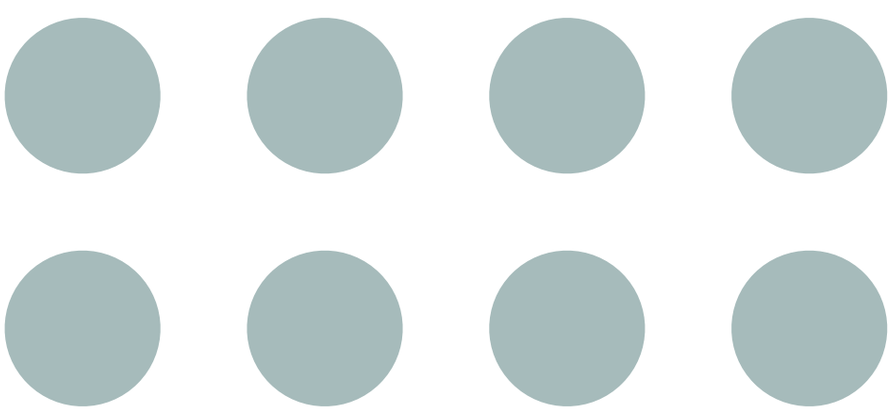


assessing **clustering tendency**

Test spatial randomness by statistic test: Hopkins Static

ASSESSING **CLUSTERING TENDENCY**





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

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, $p_1, \ldots, p_n$, uniformly at random. For each p_i, find its nearest neighbor in D: x_i = min{dist (p_i, v)} where v in D

Sample n points, $q_1, \ldots, q_n$, uniformly from D. For each q_i, find its nearest neighbor in D – {q_i}: y_i = min{dist (q_i, v)} where v in D and v ≠ q_i

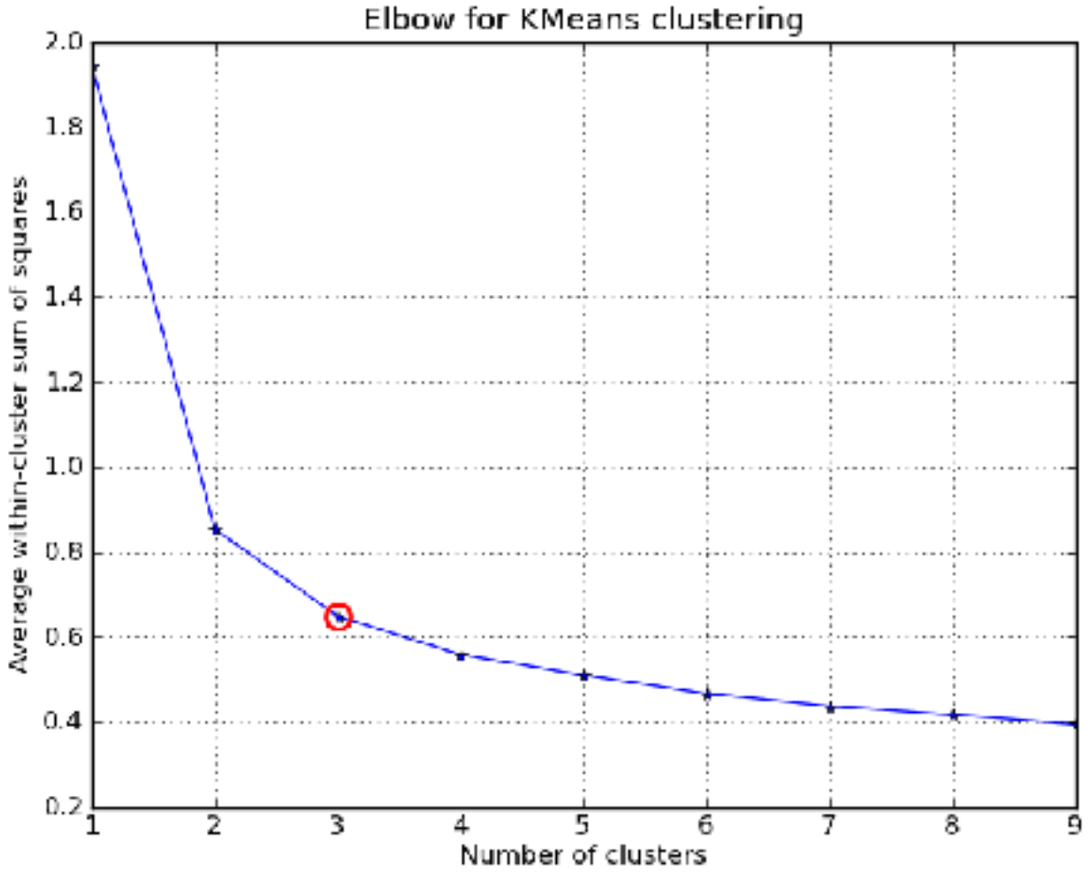
If D is uniformly distributed, $\sum x_i$ and $\sum y_i$ will be close to each other and H is close to 0.5. If D is skewed, H is close to 0

how many **clusters?**

empirical

of clusters: k ≈√n/2 for
a dataset of n points,
e.g., n = 200, k = 10

elbow method



Use the turning point in the curve of sum of within cluster variance with respect to the number of clusters

Divide a given data set into m parts

Cross validation

Use m – 1 parts to obtain a clustering model

Use the remaining part to test the quality of the clustering

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 any k > 0, repeat it m times, compare the overall quality measure with respect to different k's, and find number of clusters that fits the data the best

external quality measure

supervised, employ criteria not inherent to the dataset

Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure

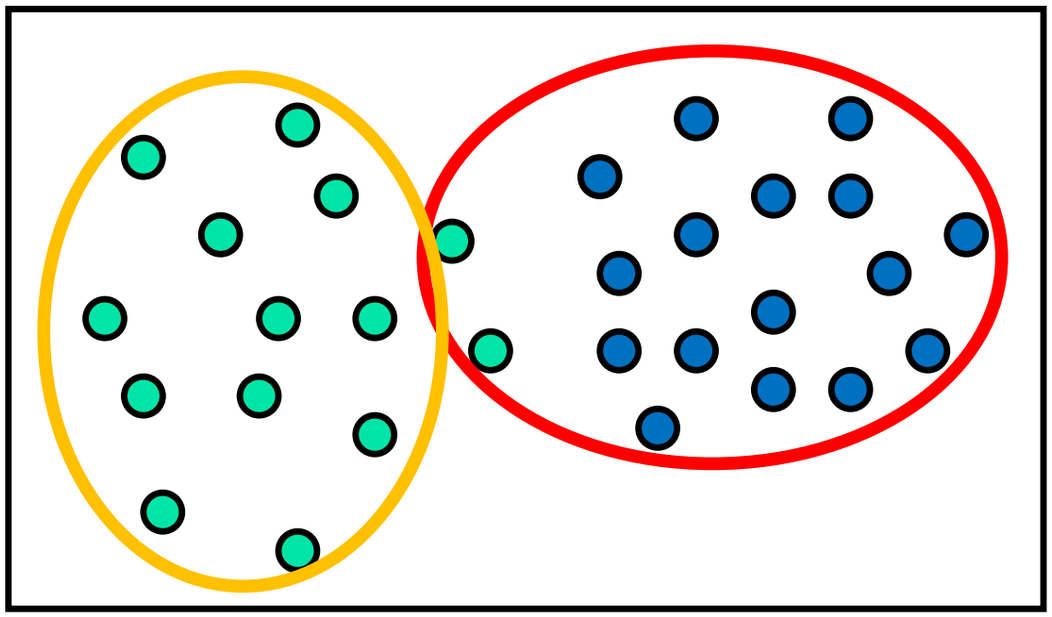
internal quality measure

unsupervised, criteria derived from data itself

Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are, e.g., Silhouette coefficient

relative

directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm



EXTERNAL METHODS

Clustering quality measure: Q(C, T), for a clustering C given the ground truth T

Q is good if it satisfies the following four essential criteria

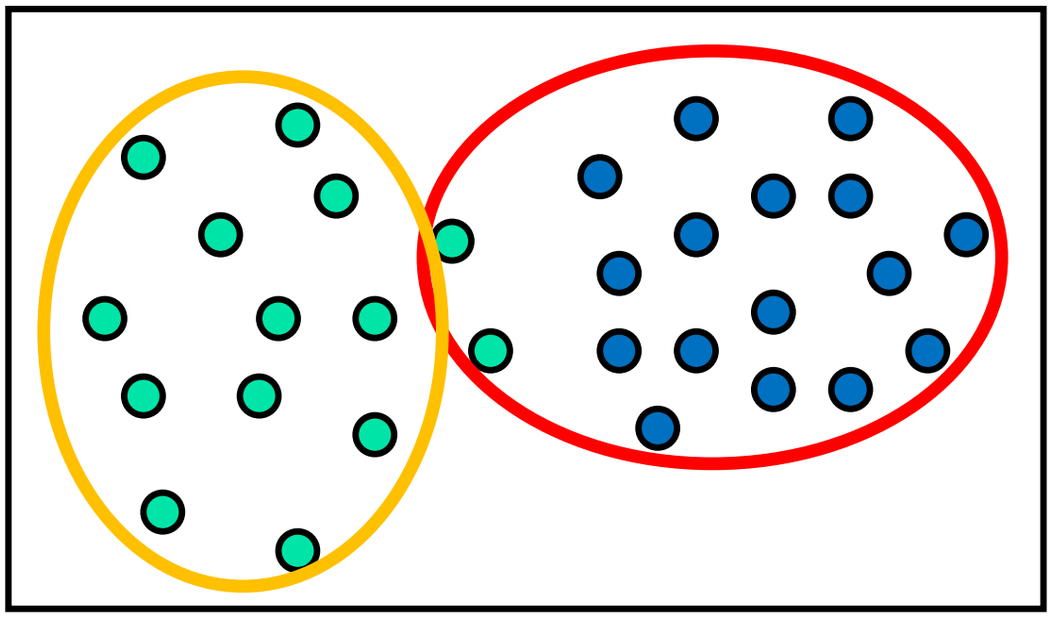
Cluster homogeneity: the purer, the better

Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster

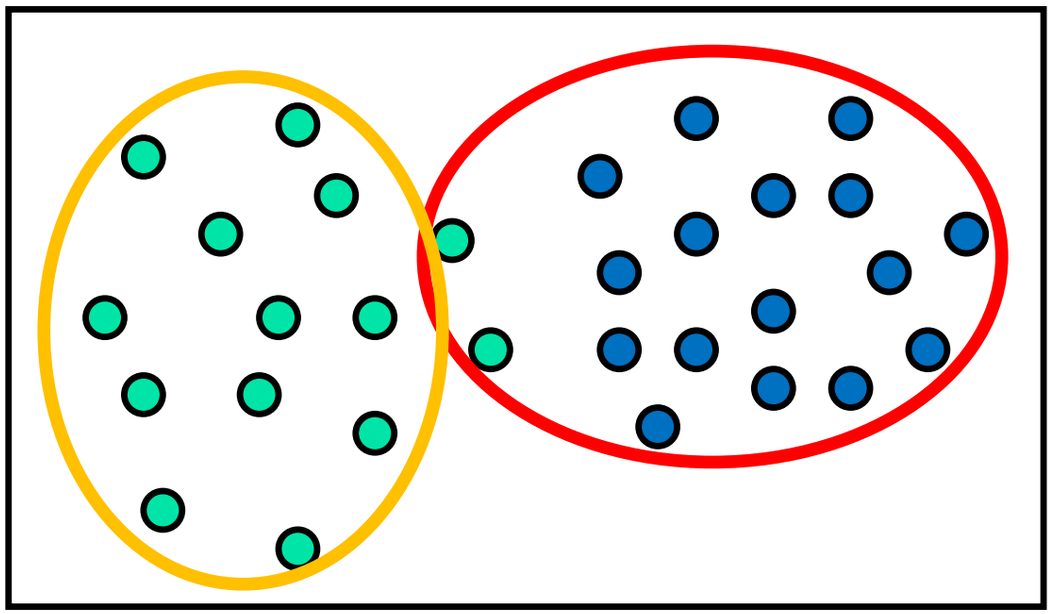
Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a rag bag (i.e., "miscellaneous" or "other" category)

Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

MANY EXTERNAL MEASURES



- Matching-based measures
  - Purity, maximum matching, F-measure
- Entropy-Based Measures
  - Conditional entropy, normalized mutual information (NMI), variation of information
- Pair-wise measures
  - Four possibilities: True positive (TP), FN, FP, TN
  - Jaccard coefficient, Rand statistic, Fowlkes-Mallow measure
- Correlation measures
  - Discretized Huber static, normalized discretized Huber static



conditional entropy

$$H(C) = -\sum_{i=1}^{r} p_{C_i} \log p_{C_i} \qquad p_{C_i} = \frac{n_i}{n}$$

entropy of partitioning T

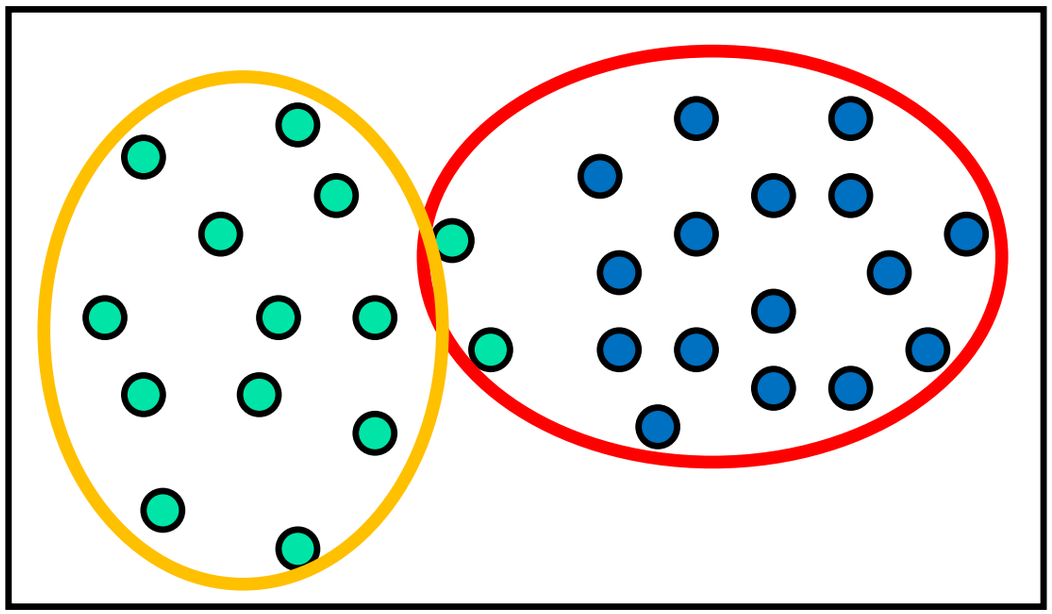
$$H(T) = -\sum_{i=1}^{k} p_{T_i} \log p_{T_i}$$

partitioning T with respect to cluster C_i:

$$H(T|C_i) = -\sum_{j=1}^{k} \frac{n_{i,j}}{n_i} \log \frac{n_{i,j}}{n_i}$$

partitioning T with respect to clustering C

$$H(T|C) = -\sum_{i=1}^{r} \frac{n_i}{n} H(T|C_i) = -\sum_{i=1}^{r} \sum_{j=1}^{k} p_{i,j} \log \frac{p_{i,j}}{p_{C_i}}$$



conditional entropy

The more a cluster's members are split into different partitions, the higher the conditional entropy

For a perfect clustering, the conditional entropy value is 0, where the worst possible conditional entropy value is log k

mutual information

quantify the amount of shared information between the clustering C and partitioning T

$$I(C,T) = H(T) - H(T|C)$$

$$I(C,T) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{i,j} \log\left(\frac{p_{i,j}}{p_{C_i} p_{T_j}}\right)$$

mutual information

quantify the amount of shared information between the clustering C and partitioning T

$$NMI(C,T) = \frac{I(C,T)}{\sqrt{H(C)H(T)}}$$

normalized **mutual** information

quantify the amount of shared information between the clustering C and partitioning T