Data Mining:
Concepts and Techniques

Cluster Analysis

Li Xiong

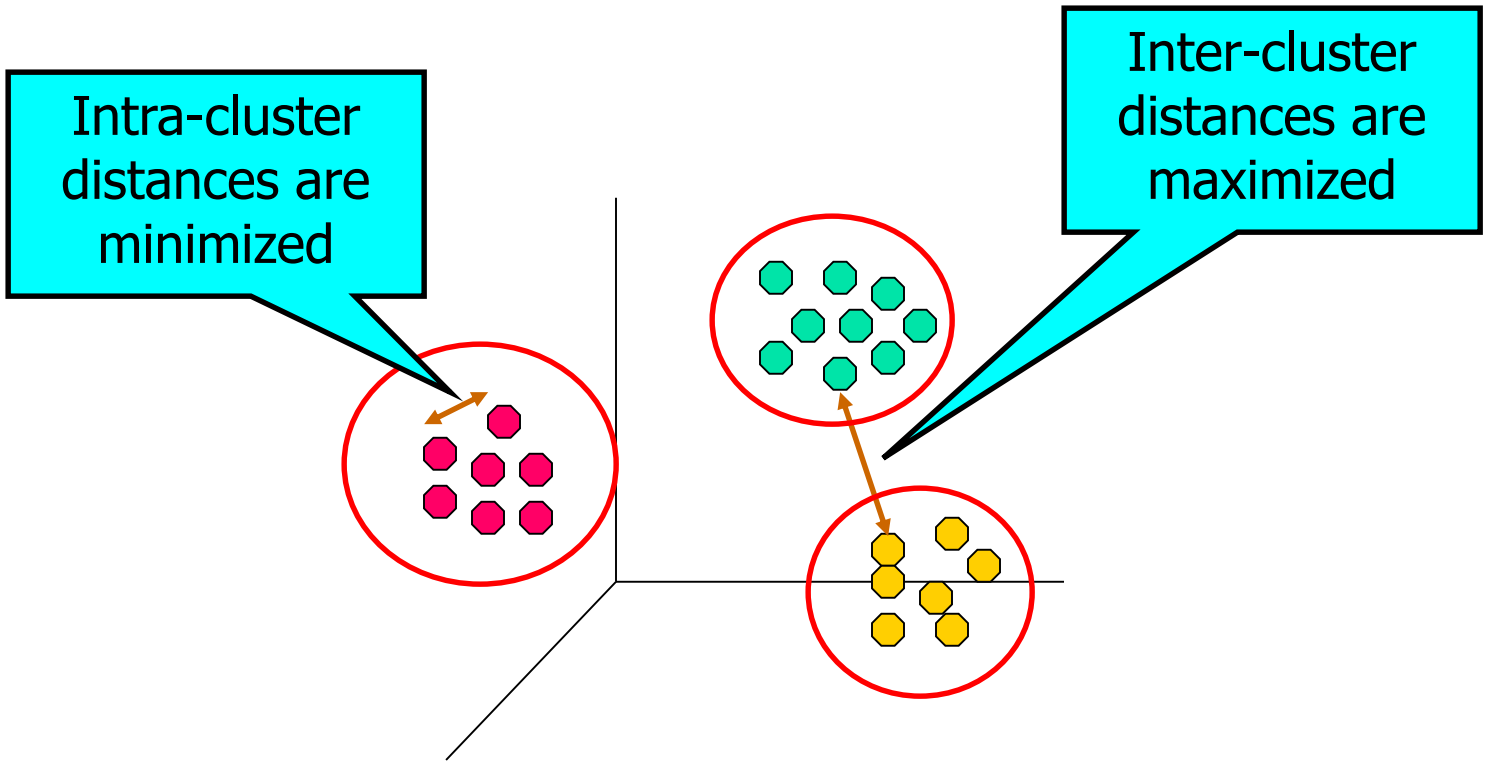
Slide credits: Jiawei Han and Micheline Kamber
Tan, Steinbach, Kumar

Cluster Analysis

- Basic Concepts
- Similarity and distances
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering

What is Cluster Analysis?

- Finding groups of objects (clusters) – given a notion of distance
  - Objects similar to one another in the same group
  - Objects different from the objects in other groups
- Unsupervised learning: no predefined classes



Machine Learning

- **Supervised:** Given input/output samples (X, y), we learn a function f such that y = f(X), which can be used on new data.
  - **Classification:** y is discrete (class labels).
  - **Regression:** y is continuous, e.g. linear regression.

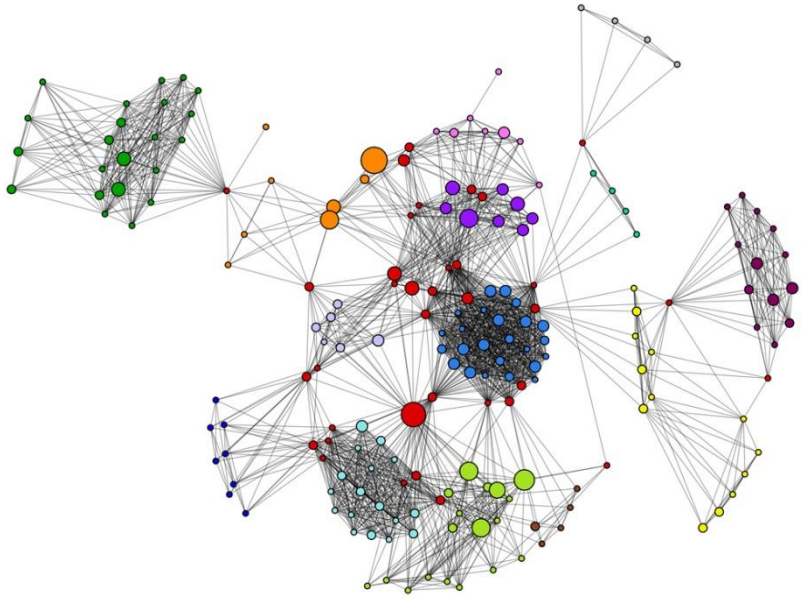
- **Unsupervised:** Given only samples X, we compute a function f such that y = f(X) is "simpler".
  - **Clustering:** y is discrete
  - **Dimension reduction:** y is continuous, e.g. matrix factorization

Applications of Cluster Analysis

- As a stand-alone tool to get insight into data distribution
  - Cluster into groups – automatic classification
  - Finding k-nearest neighbors
  - Outlier detection
- As a preprocessing step for other algorithms
  - Data cleaning: missing data, noisy data
  - Data reduction
  - Data discretization

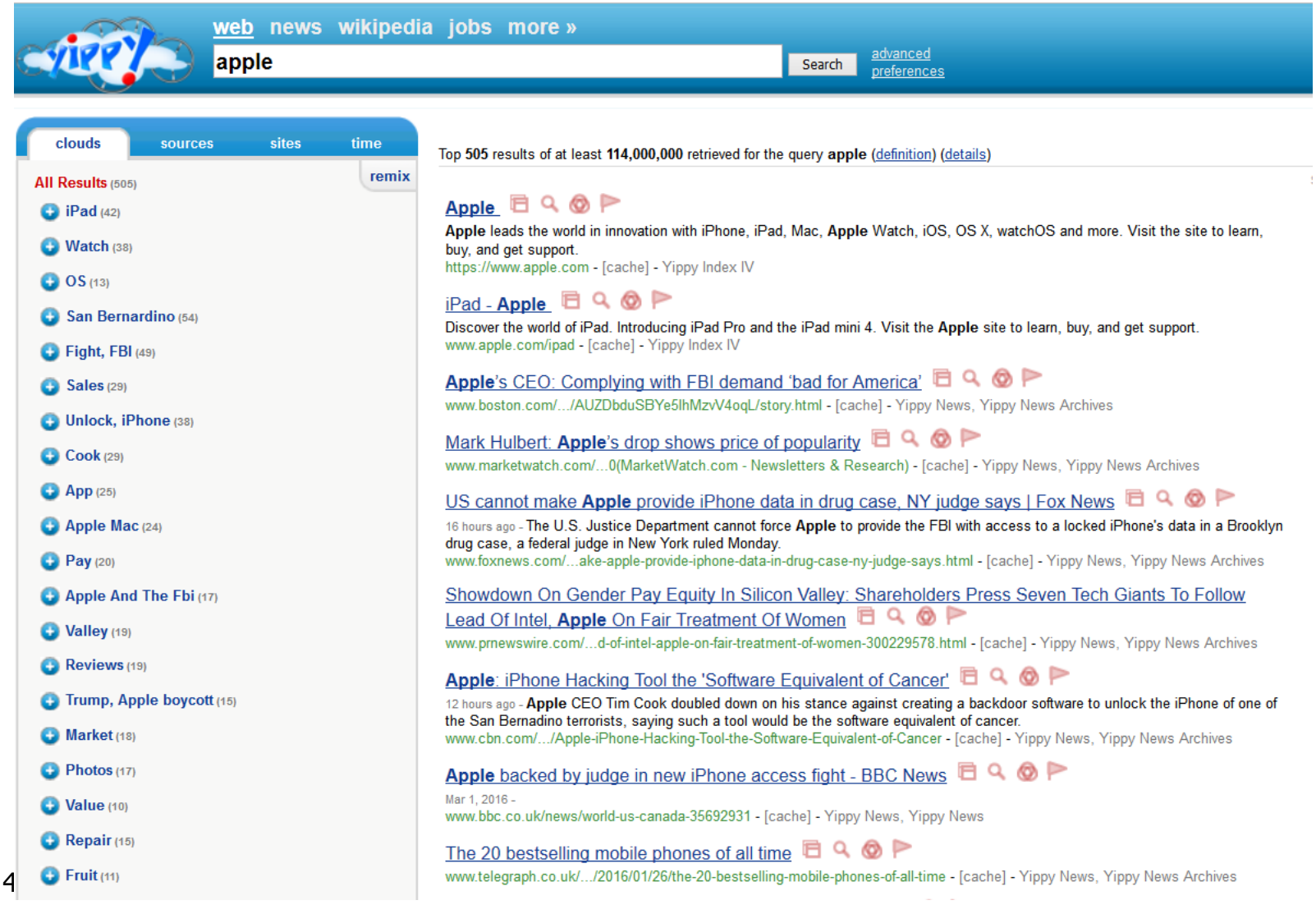
Clustering Applications

- Marketing research
- Social network analysis



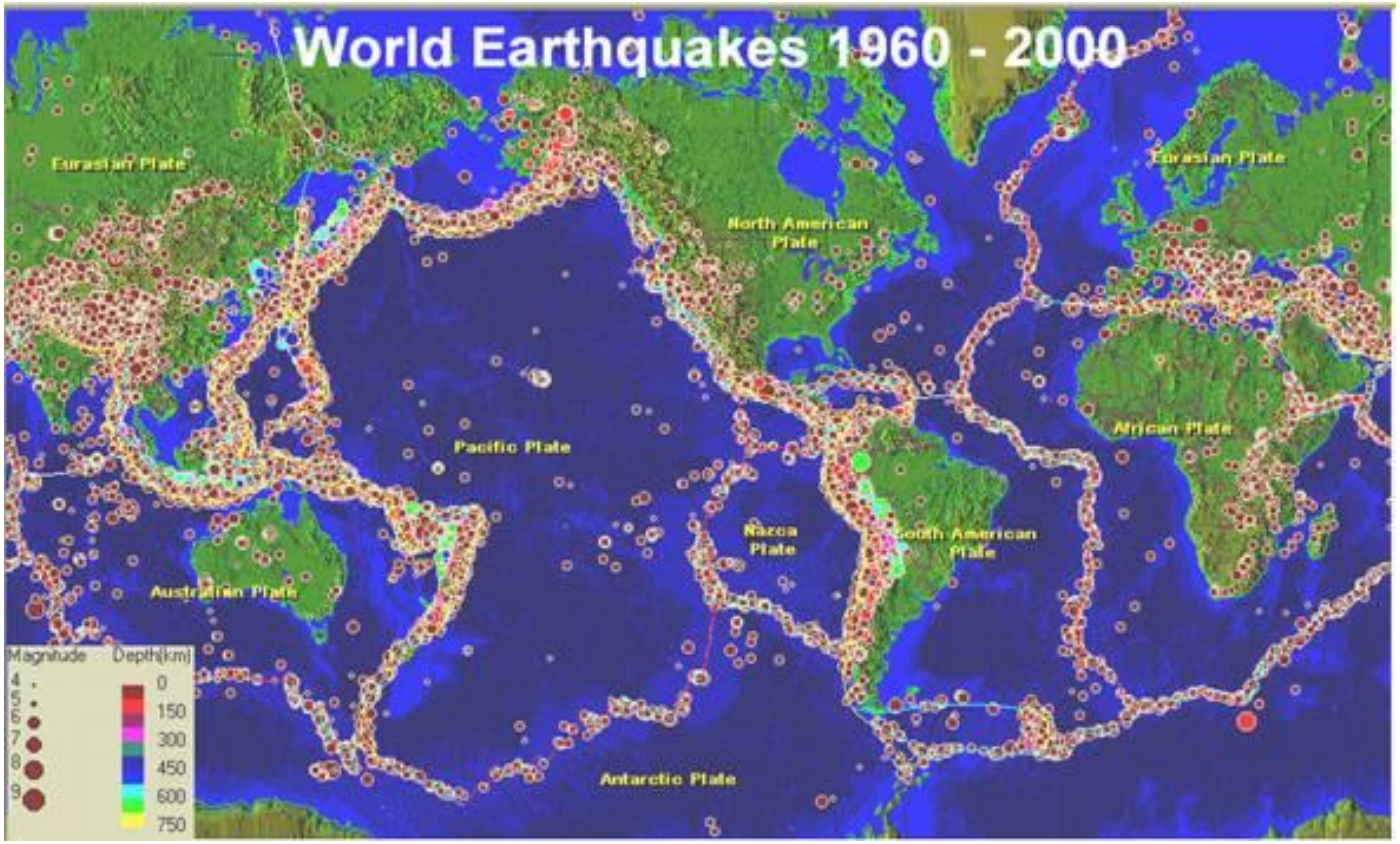
Clustering Applications

- WWW: Documents and search results clustering



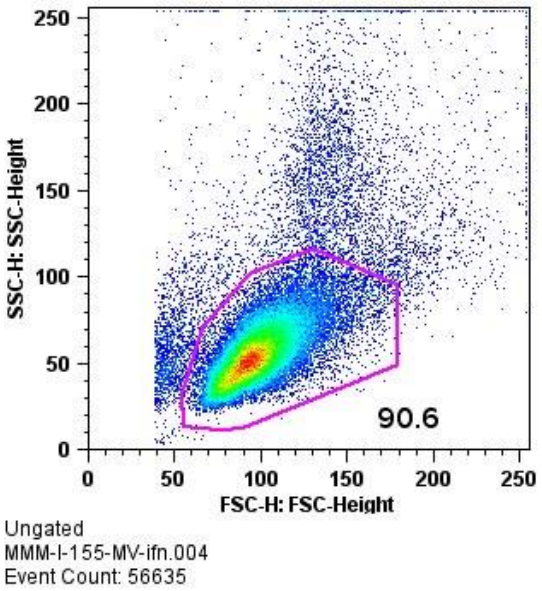
Clustering Applications

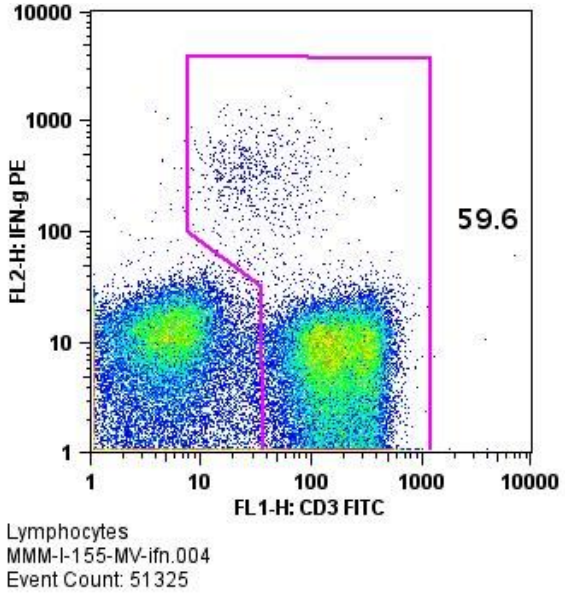
- Earthquake studies

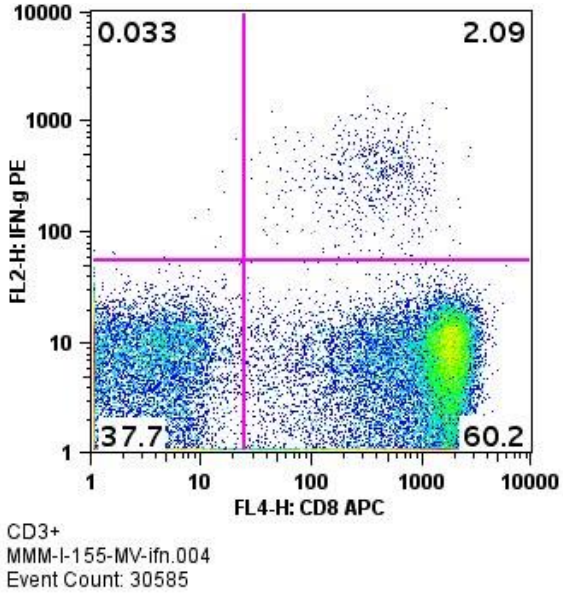


Clustering Applications

- Bioinformatics: microarray data, flow cytometry data analysis, …



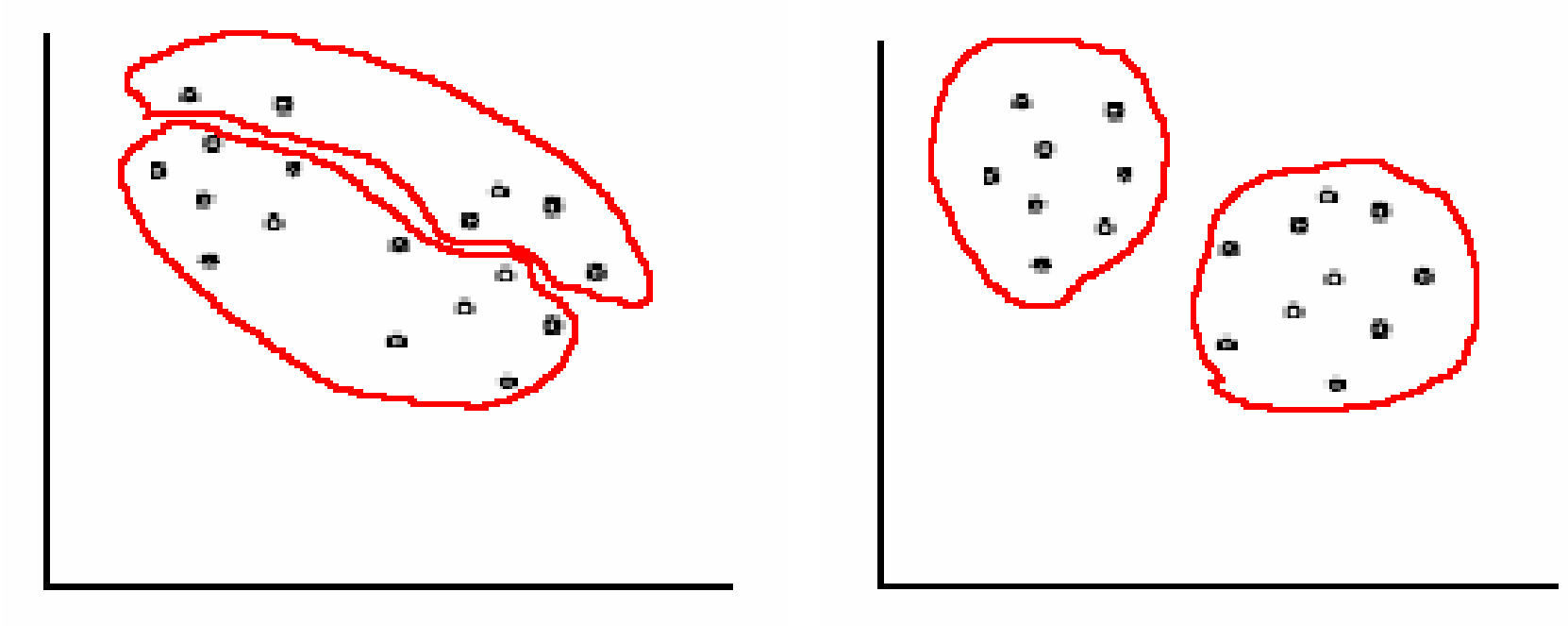




Challenges of Clustering

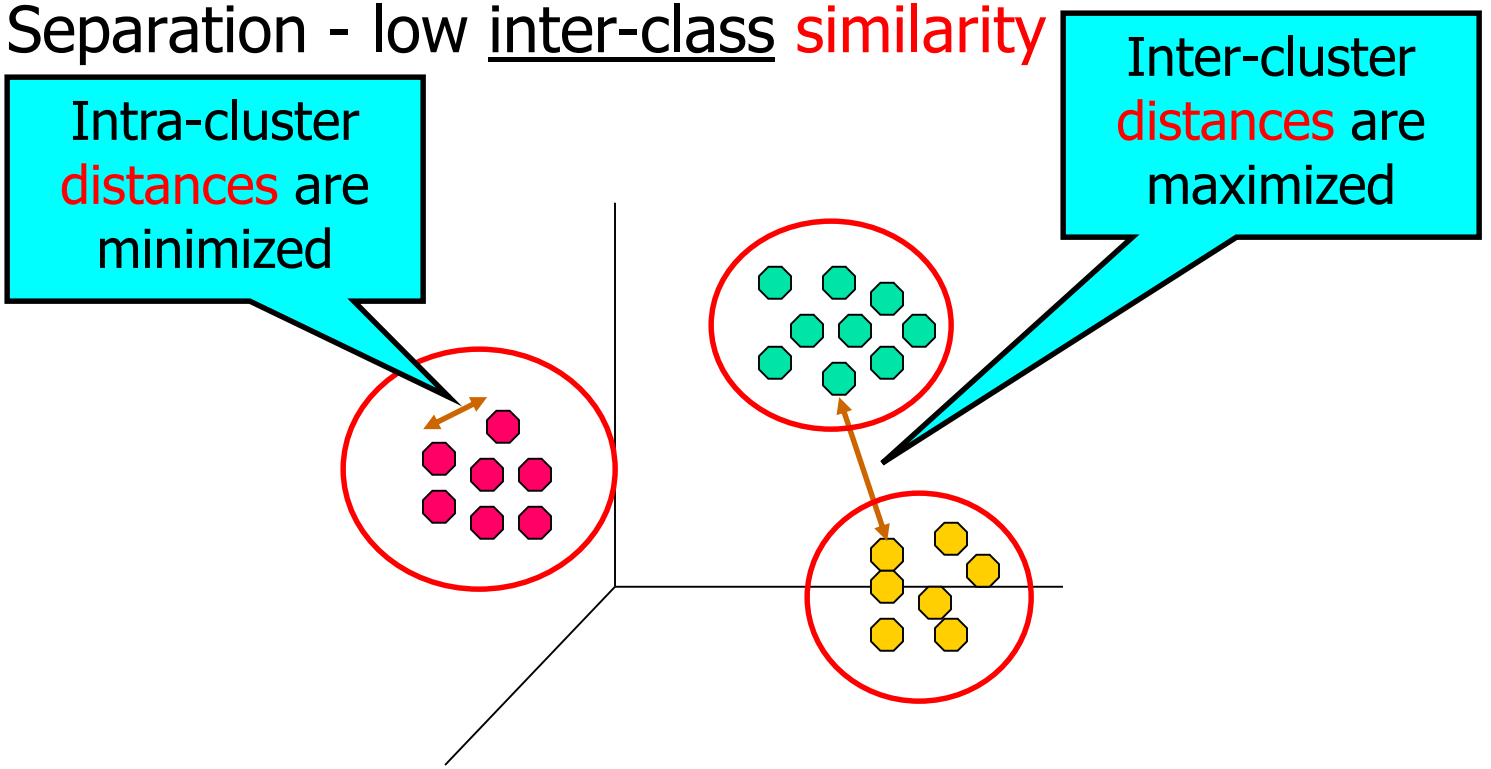
- Quality
  - Noise and outliers
  - High dimensionality
- Scalability
  - High dimensionality
  - Large data
- Usability
  - Minimal input parameters
  - User-specified constraints

Quality: What Is Good Clustering?



Quality: What Is Good Clustering?

- Agreement with "ground truth"
- A good clustering will produce high quality clusters with
  - Homogeneity - high intra-class similarity
  - Separation - low inter-class similarity



Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- **Similarity and distances**
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering

Similarity/distance between data objects

Data objects

- as points: distance between points
- as vectors: cosine between vectors
- as random variables: correlation
- as sets: Jaccard distance between sets
- as strings: Hamming distance

Distance between two data points

- Euclidean distance

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

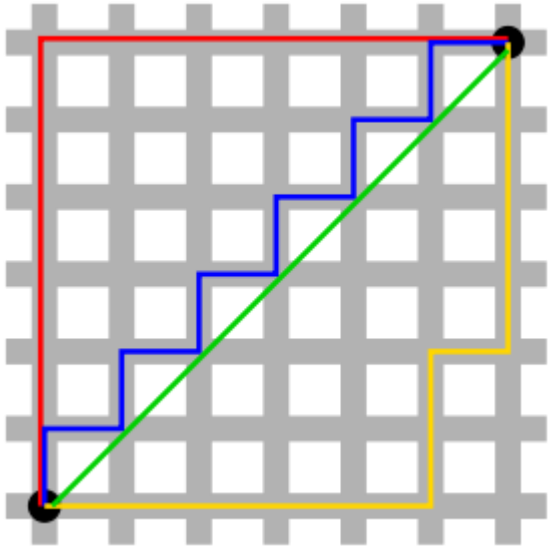
- Manhattan distance

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}^{1}-x_{j_p}|$$

- *Minkowski distance*

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$



Distance between two attributes values

- To compute $|x_{if} - x_{jf}|$
  - f is numeric (interval or ratio scale)
    - Scaling issues -> normalization
  - f is ordinal
    - Mapping by rank $z_{if} = \frac{r_{if} - 1}{M_f - 1}$
  - f is nominal
    - Mapping function

      $|x_{if} - x_{jf}| = 0$ if $x_{if} = x_{jf}$, or 1 otherwise
    - Hamming distance (edit distance) for strings

Normalization of attributes

- scaled attributes to fall within a small, specified range
- Min-max normalization: [min_A, max_A] to [new_min_A, new_max_A]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

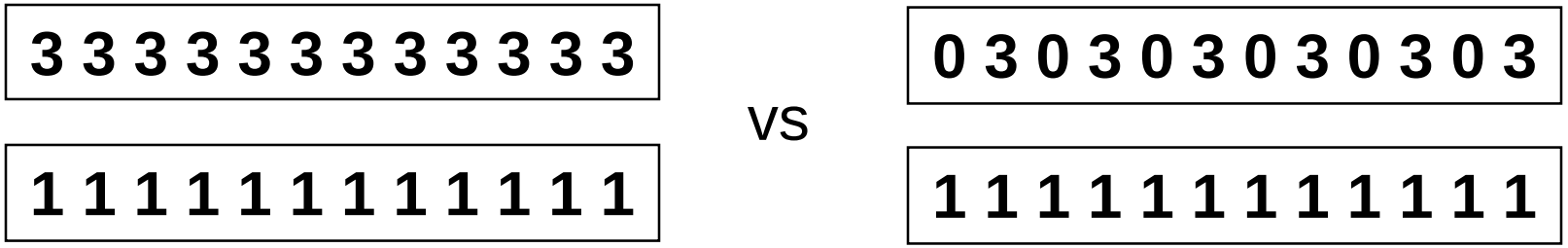
  - Ex. Let income [\$12,000, \$98,000] normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$
- Z-score normalization (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

Euclidean distance

- Euclidean distance may not be meaningful (counter intuitive) for high dimensional data, e.g. user movie ratings



Similarity/distance between data objects

Data objects

- as points: distance between points
- **as vectors: cosine between vectors**
- **as random variables: correlation**
- as sets: Jaccard distance between sets
- as strings: Hamming distance

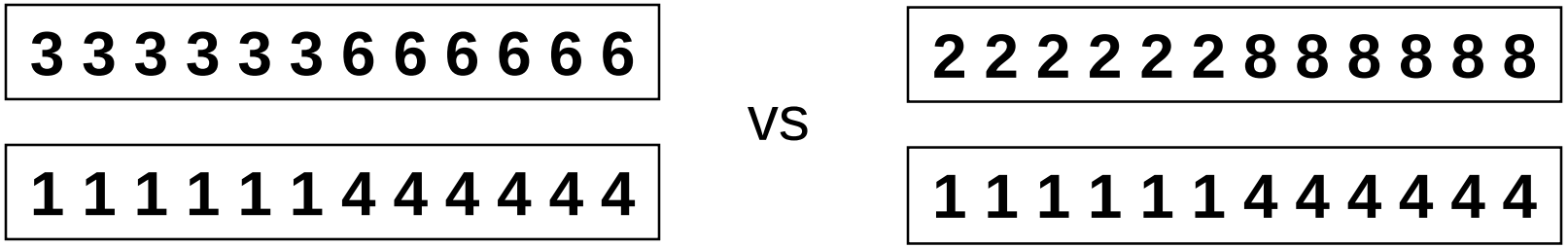
Cosine similarity between two vectors

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Cosine measure $\frac{X_i \bullet X_j}{\|X_i\| \cdot \|X_j\|}$
- From -1 to 1

Cosine similarity

- Consine similarity
  - Invariant to multiplicative scaling
  - Variant to additive scaling



Correlation between two random variables (numerical data)

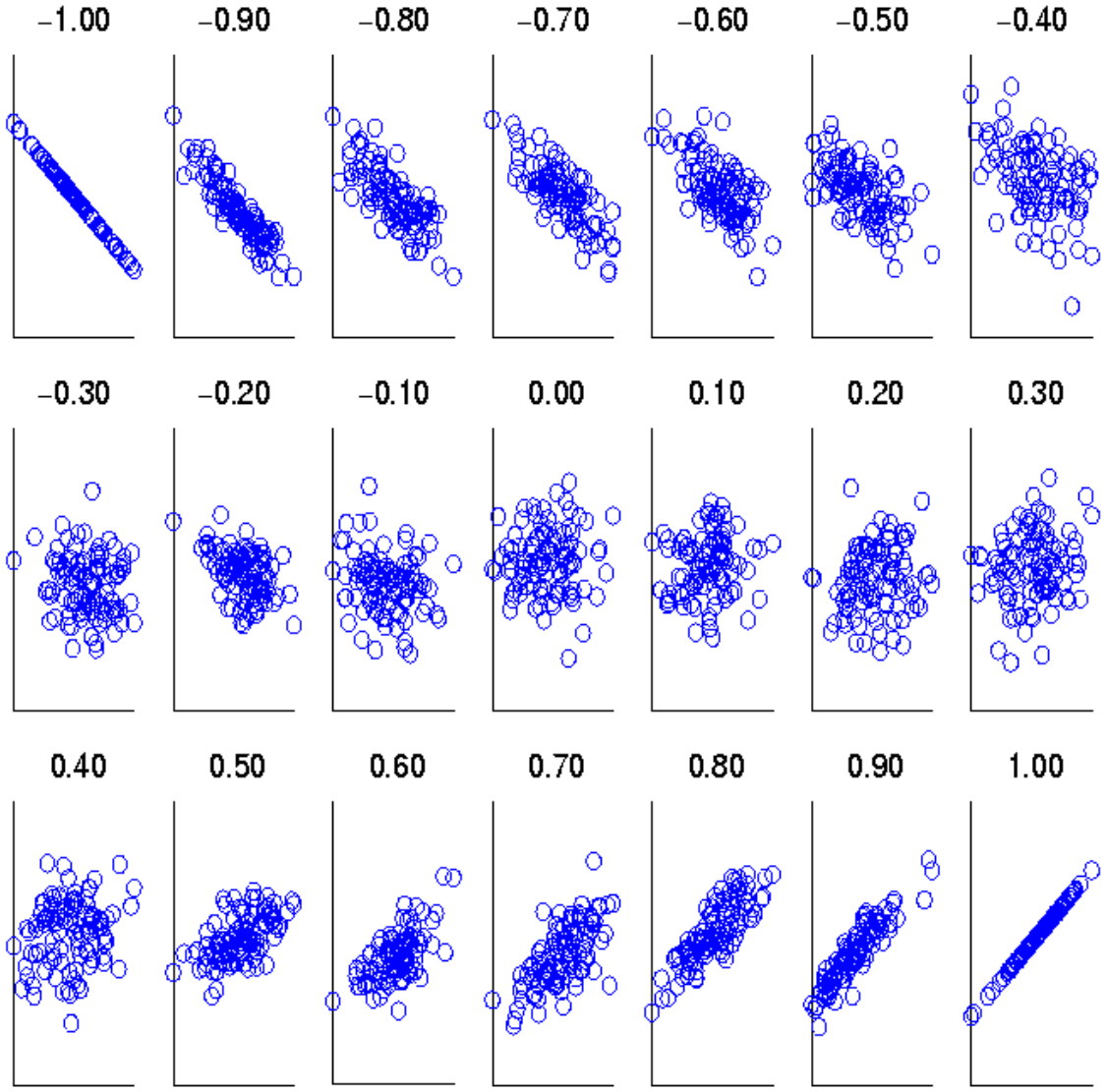
- Correlation coefficient (also called Pearson's product moment coefficient)

$$r_{A,B} = \frac{\sum (A - \overline{A})(B - \overline{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum (AB) - n\overline{A}\,\overline{B}}{(n-1)\sigma_A \sigma_B}$$

where n is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(AB)$ is the sum of the AB dot-product.

- $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's)
- $r_{A,B} = 0$: independent
- $r_{A,B} < 0$: negatively correlated

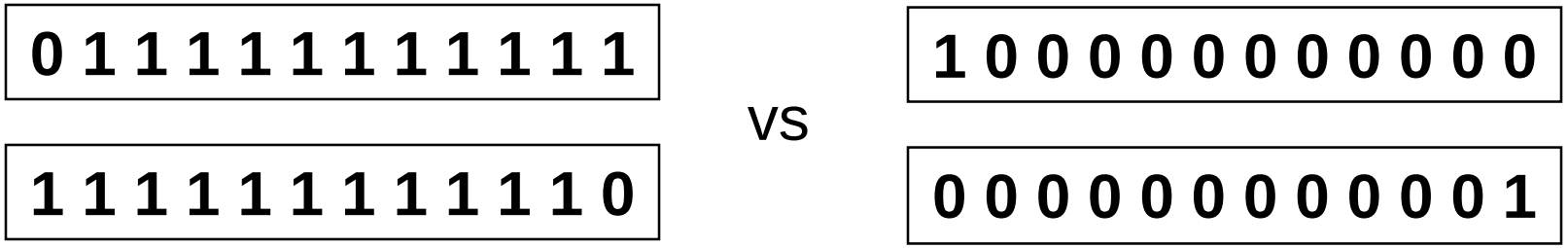
Visualization of Correlation



Scatter plots showing the Pearson correlation from –1 to 1.

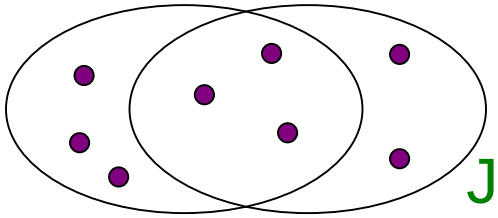
Data object at a set

- For transaction data, document data
  - Shared items are more important to consider



Jaccard distance between two sets

- The **Jaccard similarity** of two **sets** is the size of their intersection divided by the size of their union: $sim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$
- **Jaccard distance:** $d(C_1, C_2) = 1 - |C_1 \cap C_2| / |C_1 \cup C_2|$



3 in intersection
8 in union
Jaccard similarity= 3/8
Jaccard distance = 5/8

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Similarity and distances
- **Partitioning Methods**
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering

Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some "goodness" criterion
  - Typical methods: k-means, k-medoids
- Hierarchical approach:
  - Create a hierarchical decomposition of the objects
  - Typical methods: Diana, Agnes
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN
- Others

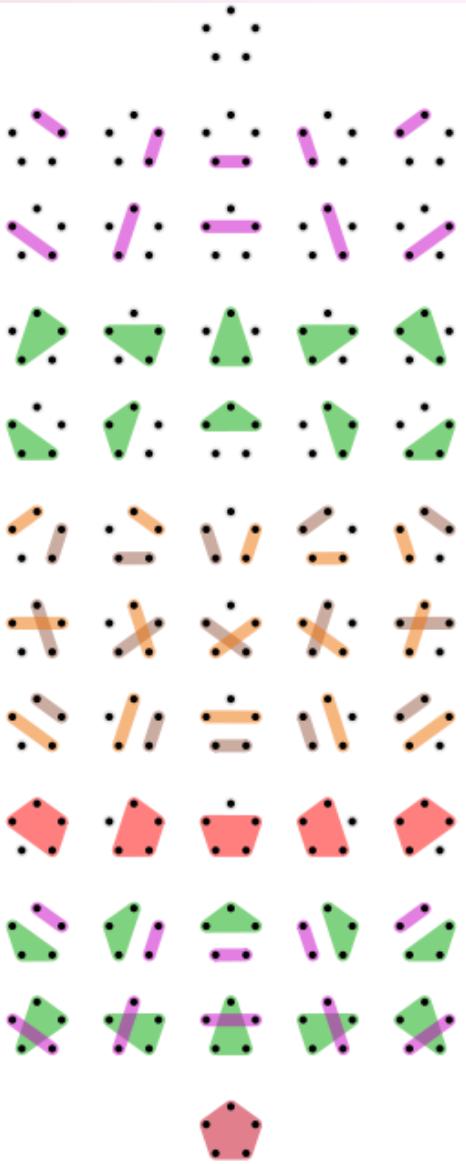
Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of ***n*** objects (into ***k*** clusters), s.t. intracluster similarity maximized and intercluster similarity minimized
- One objective: minimize the sum of squared distance from cluster centroid

$$\Sigma_{i=1}^{k}\Sigma_{p\in C_i}(p-m_i)^2$$

- How to find optimal partition?

Number of partitionings



Number of partitionings

- Stirling partition number – number of ways to partition n objects into k non-empty subsets

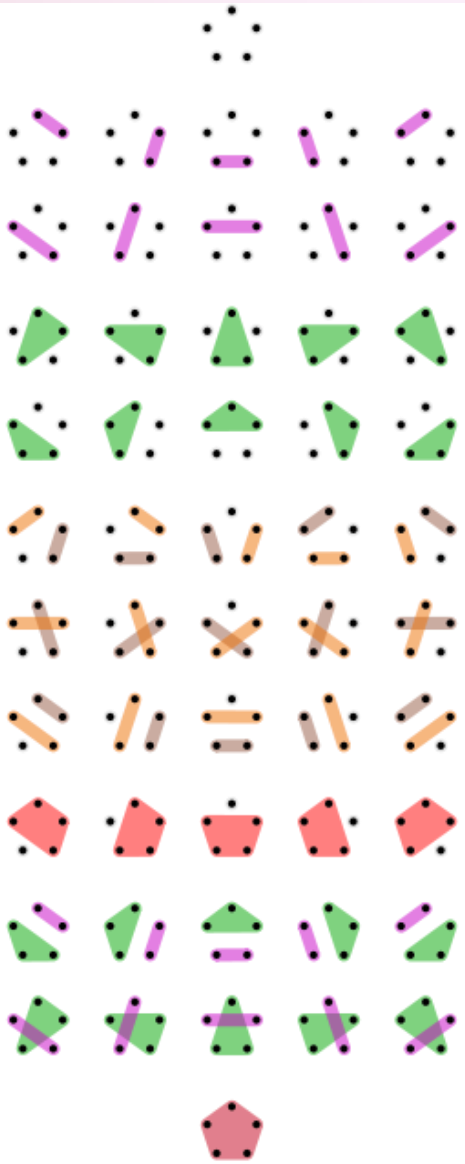
$$\left\{ {n+1 \atop k} \right\} = k \left\{ {n \atop k} \right\} + \left\{ {n \atop k-1} \right\}$$

(n= 5, k = 1, 2, 3, 4, 5): 1, 15, 25, 10, 1
(n=10, k = 1, 2, 3, 4, 5, …): 1, 511, 9330, 34105, 42525, …

- Bell numbers – number of ways to partition n objects

$$B_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\}.$$

(n = 0, 1, 2, 3, 4, 5, …): 1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, 678570, 4213597, 27644437, 190899322, 1382958545, 10480142147, 82864869804, 682076806159, 5832742205057, …



Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of ***n*** objects into ***k*** clusters, s.t. intracluster similarity maximized and intercluster similarity minimized
- One objective: minimize the sum of squared distance from cluster centroid

$$\Sigma_{i=1}^{k}\Sigma_{p\in C_i}(p-m_i)^2$$

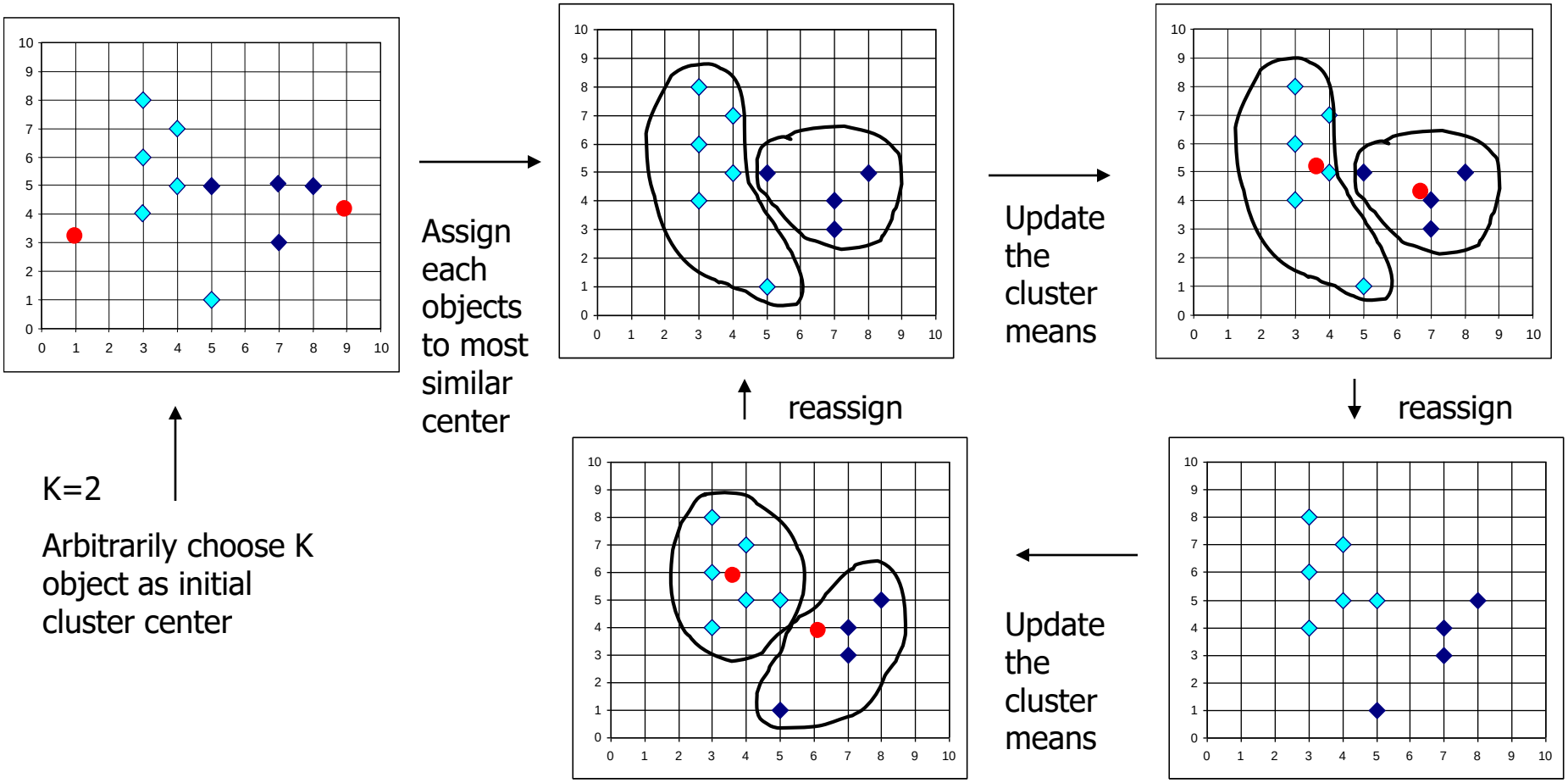
- Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (Lloyd'57, MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

K-Means Clustering: Lloyd Algorithm

- Given *k*, and randomly choose *k* initial cluster centers
- Partition objects into *k* nonempty subsets by assigning each object to the cluster with the nearest centroid
- Update centroid, i.e. *mean point* of the cluster
- Go back to Step 2, stop when no more new assignment and centroids do not change

The *K-Means* Clustering Method

- Example



K-means Clustering – Details

- Initial centroids are often chosen randomly
  - Example: Pick one point at random, then k-1 other points, each as far away as possible from the previous points
- The centroid is (typically) the mean of the points in the cluster.
- 'Nearest' is measured by Euclidean distance, cosine similarity, correlation, etc.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- Complexity is $O(tkn)$

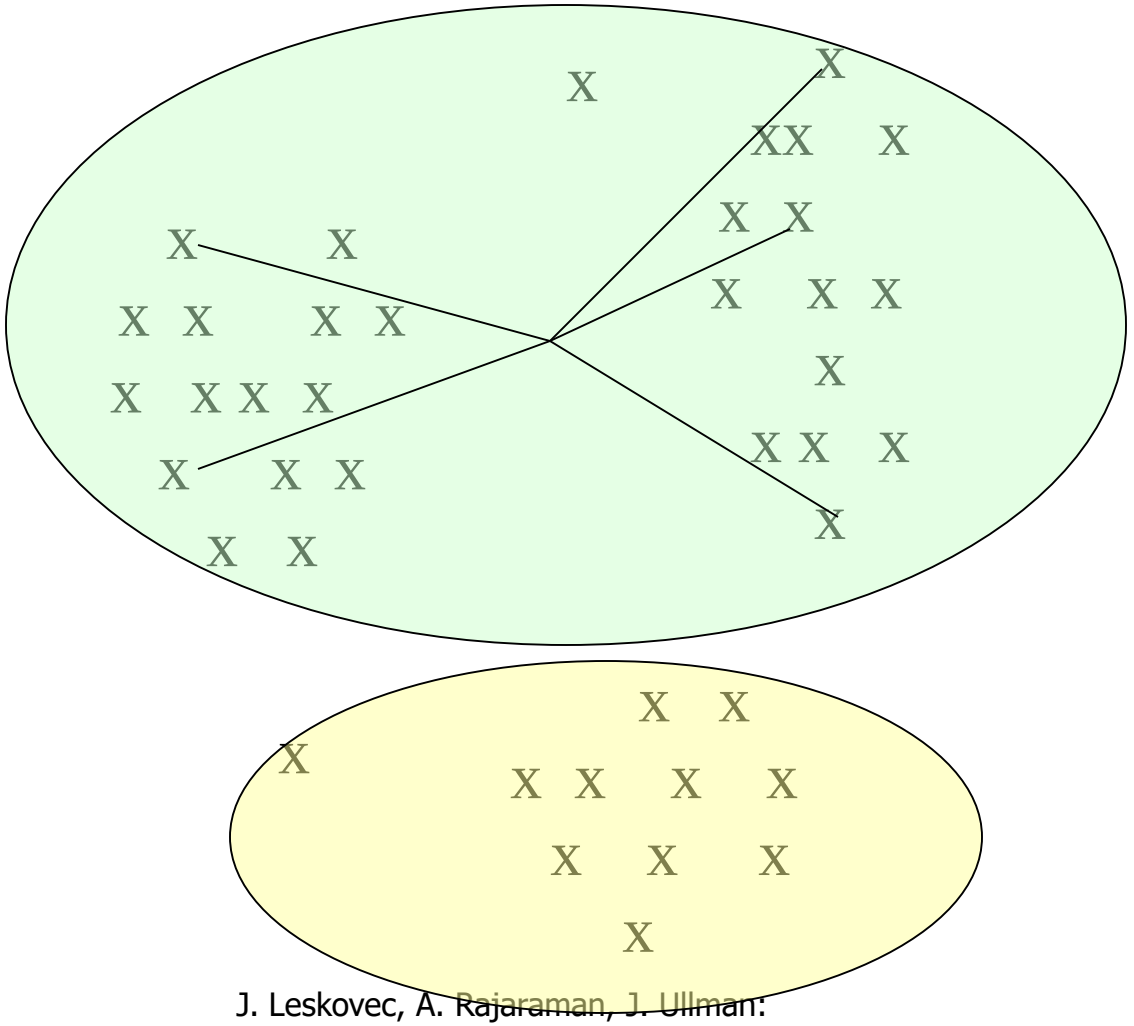
n is # objects, k is # clusters, and t is # iterations.

Comments on the *K-Means* Method

- Strength
  - Simple and works well for "regular" disjoint clusters
  - Relatively efficient and scalable (normally, k, $t << n$)
- Weakness
  - Need to specify *k,* the *number* of clusters, in advance
  - Depending on initial centroids, may terminate at a *local optimum*
  - Sensitive to noisy data and *outliers*
  - Not suitable for clusters of
    - Different sizes
    - Non-convex shapes

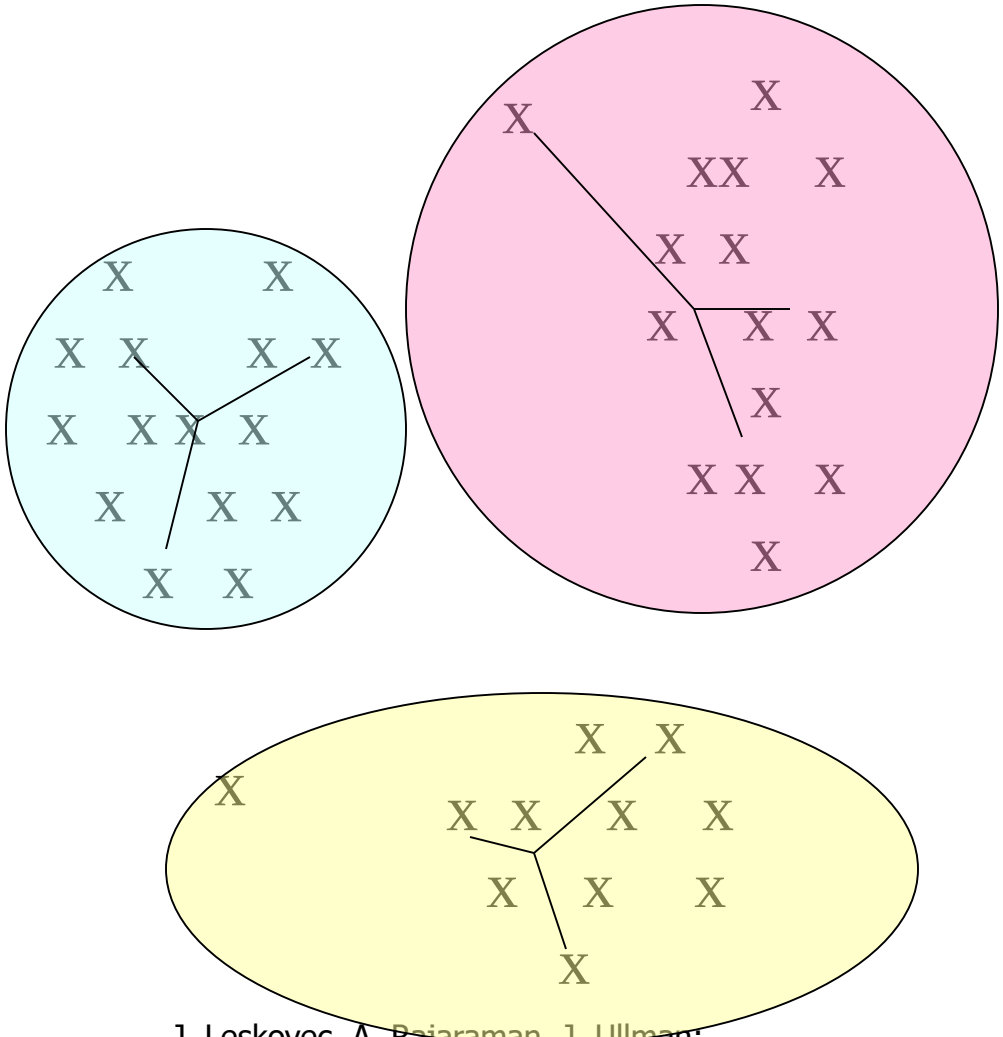
Example: Picking *k*

Too few;
many long
distances
to centroid.



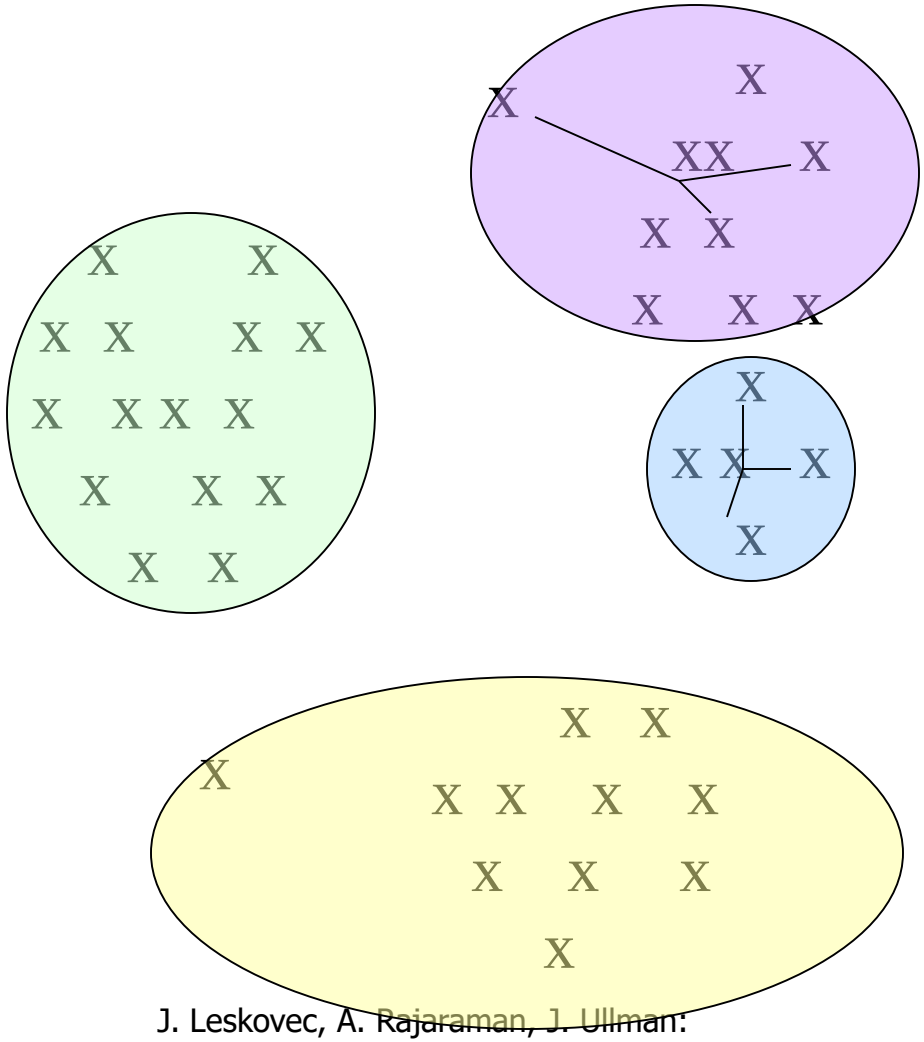
Example: Picking *k*

Just right; distances rather short.



Example: Picking *k*

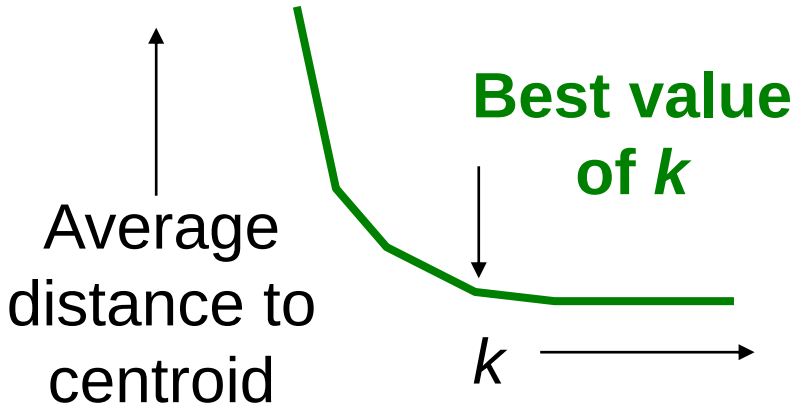
Too many;
little improvement in average distance.



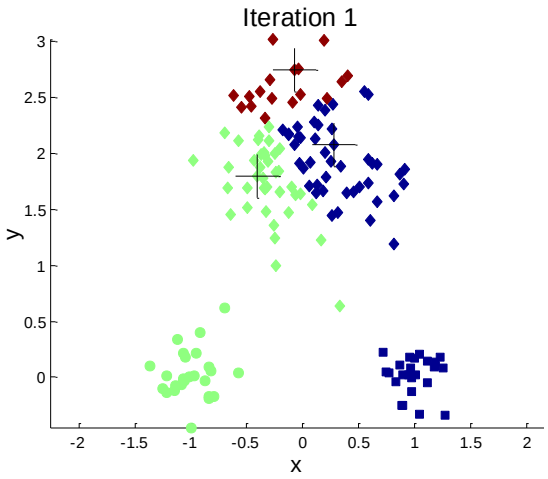
Getting the *k* right

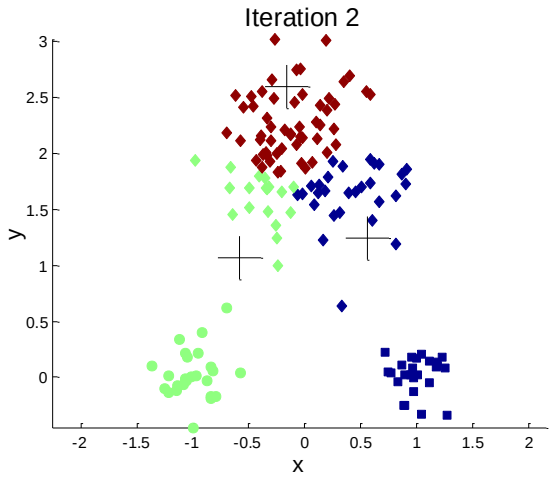
How to select *k*?

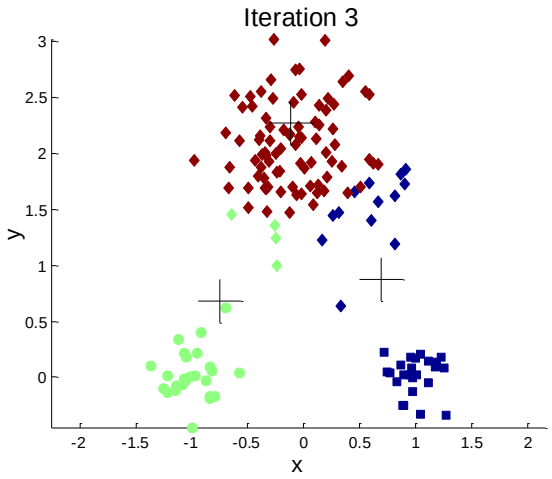
- Try different **k**, looking at the change in the average distance to centroid (or SSE) as **k** increases
- Average falls rapidly until right **k**, then changes little

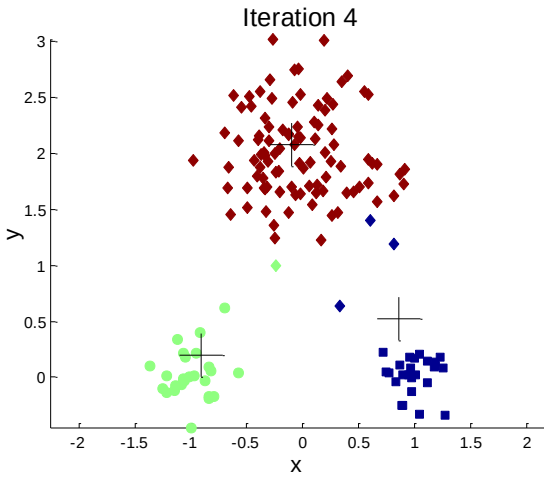


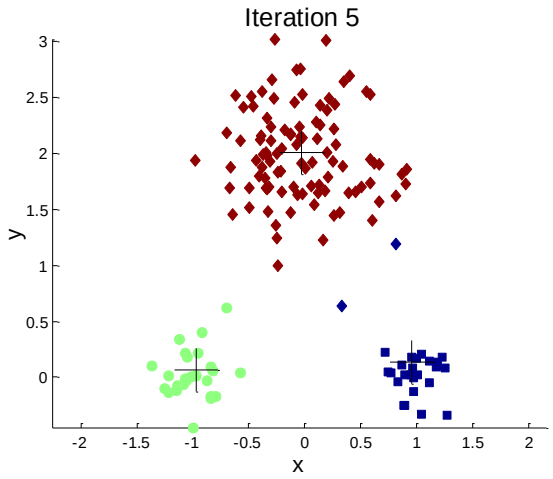
Importance of Choosing Initial Centroids – Case 1

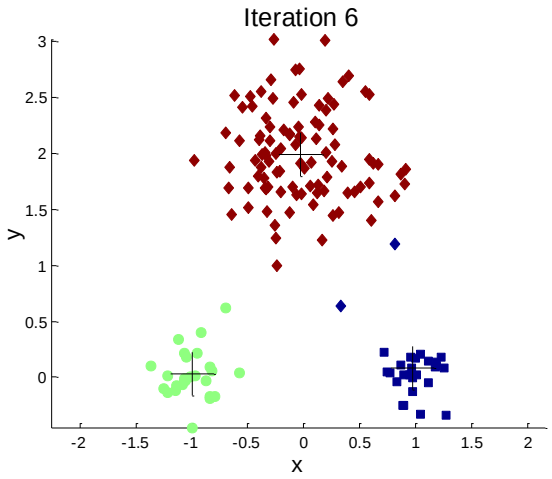




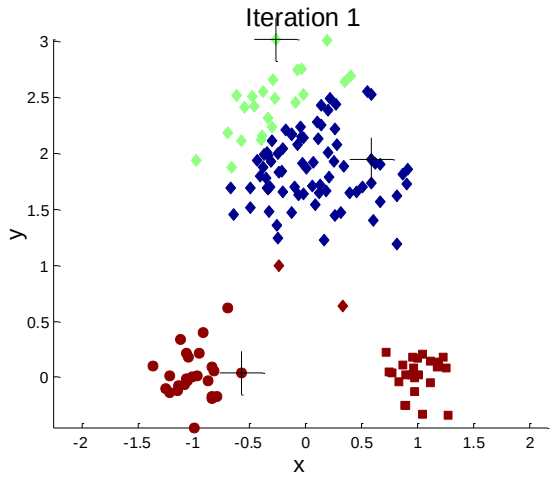


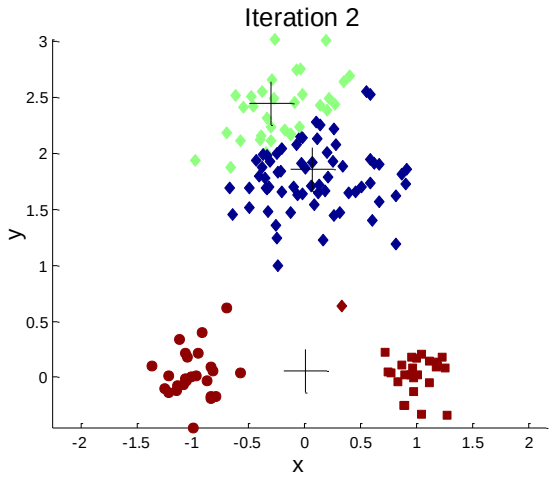


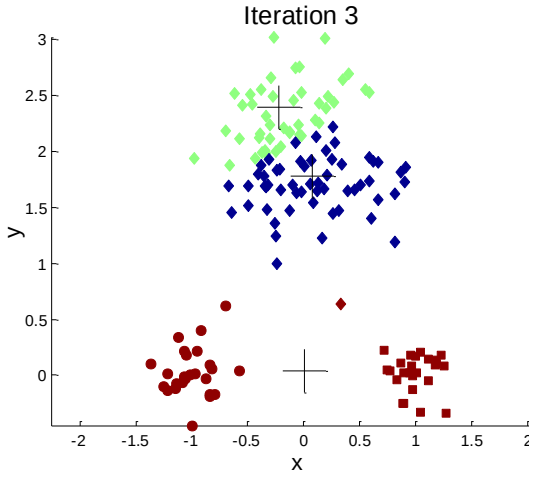


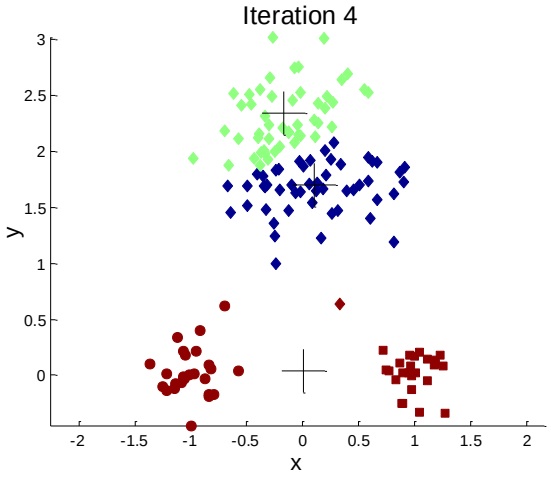


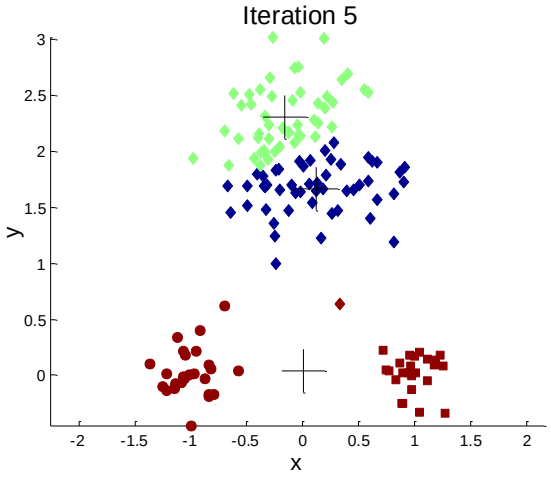
Importance of Choosing Initial Centroids – Case 2



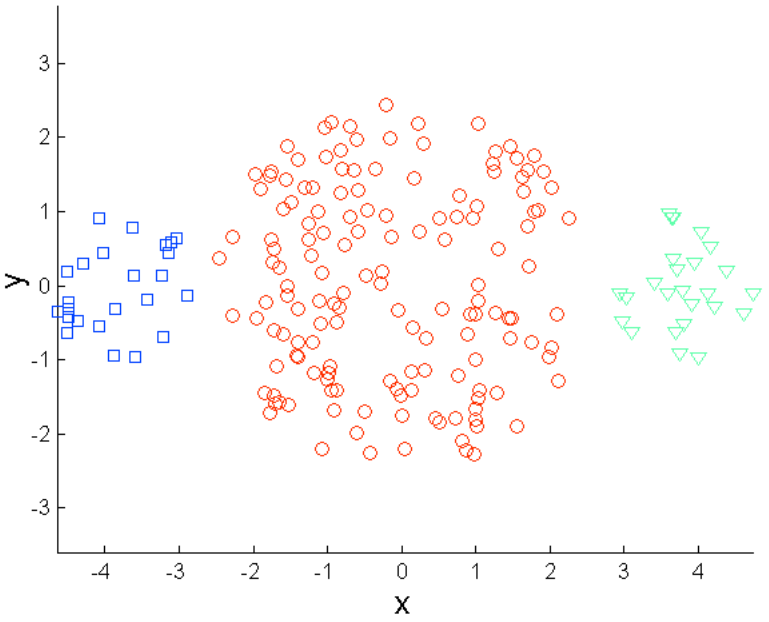




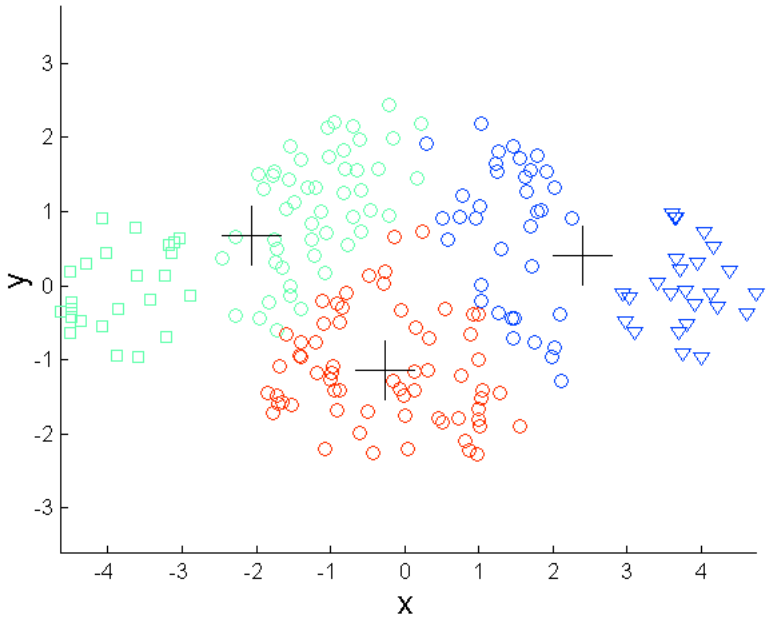




Limitations of K-means: Differing Sizes

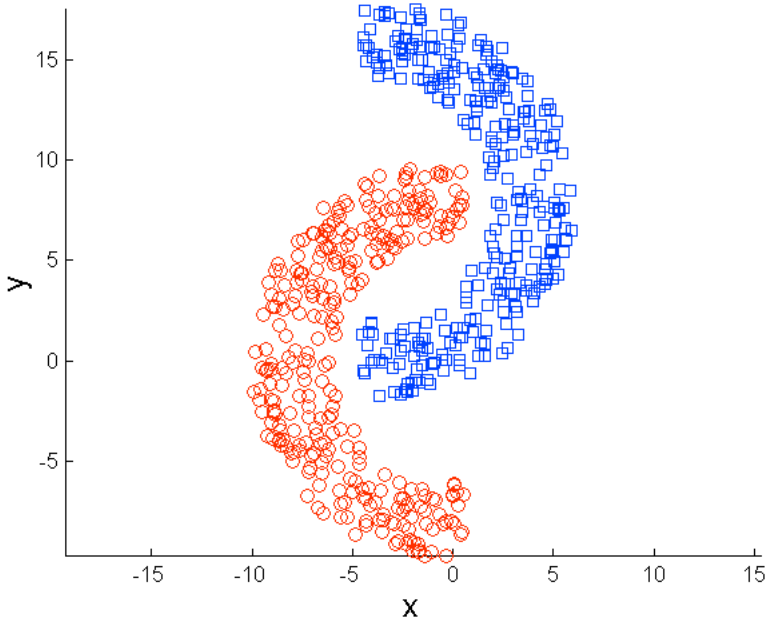


Original Points

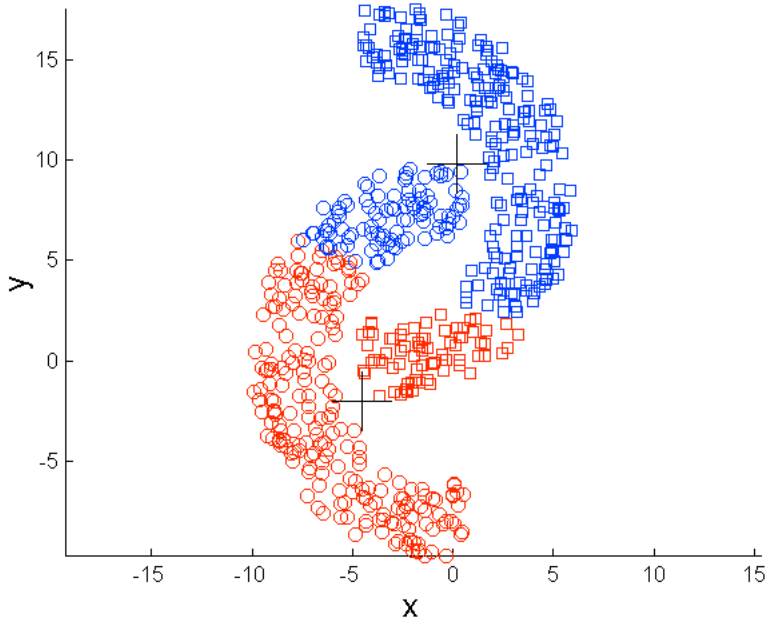


K-means (3 Clusters)

Limitations of K-means: Non-convex Shapes

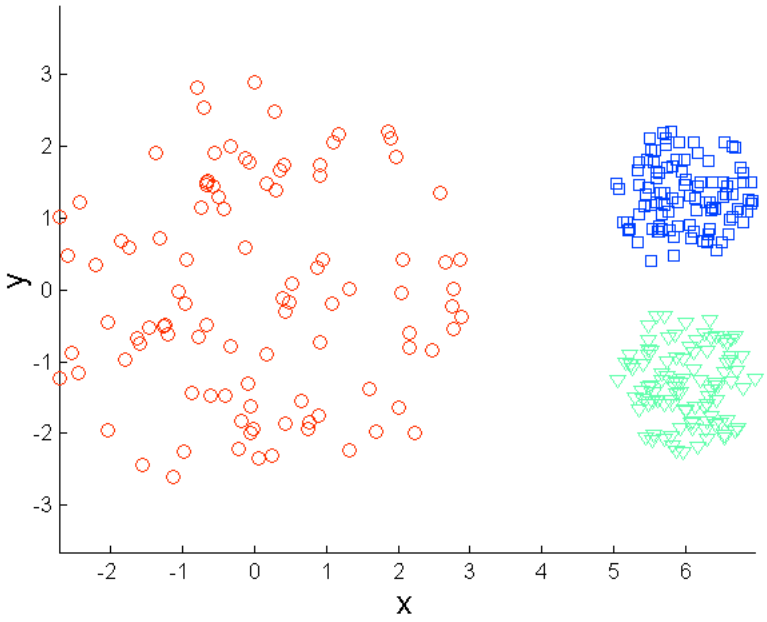


Original Points

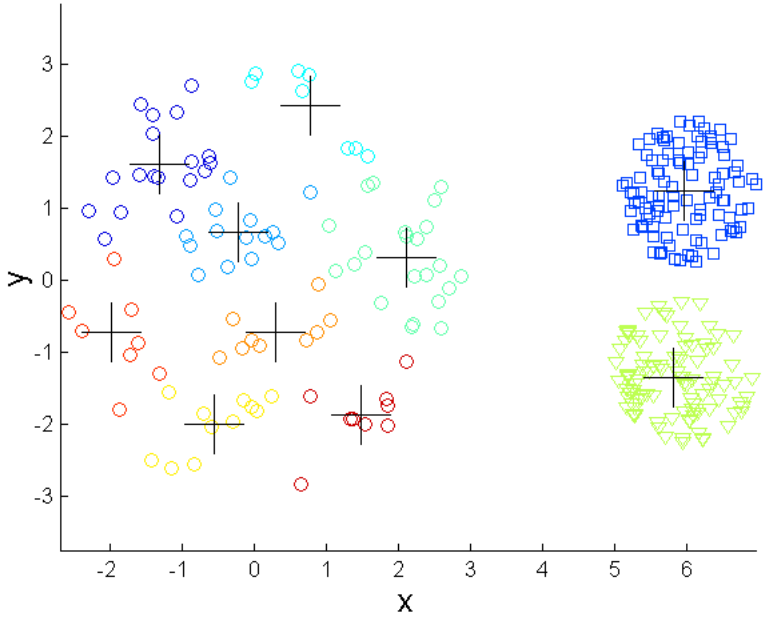


K-means (2 Clusters)

Overcoming K-means Limitations

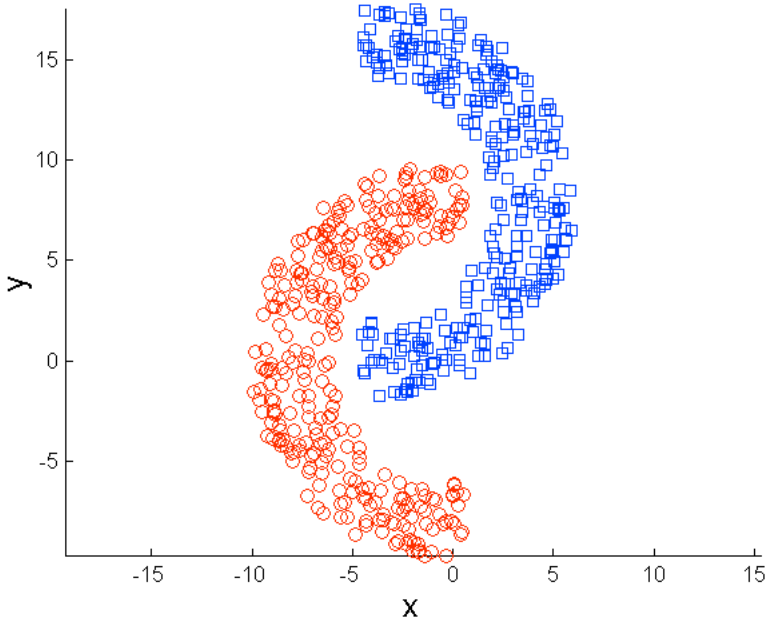


Original Points

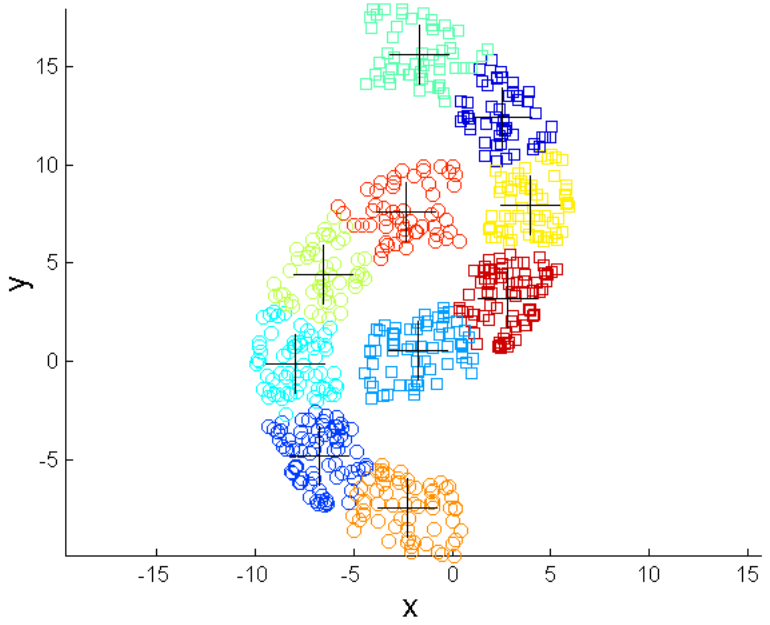


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

Assignment 3

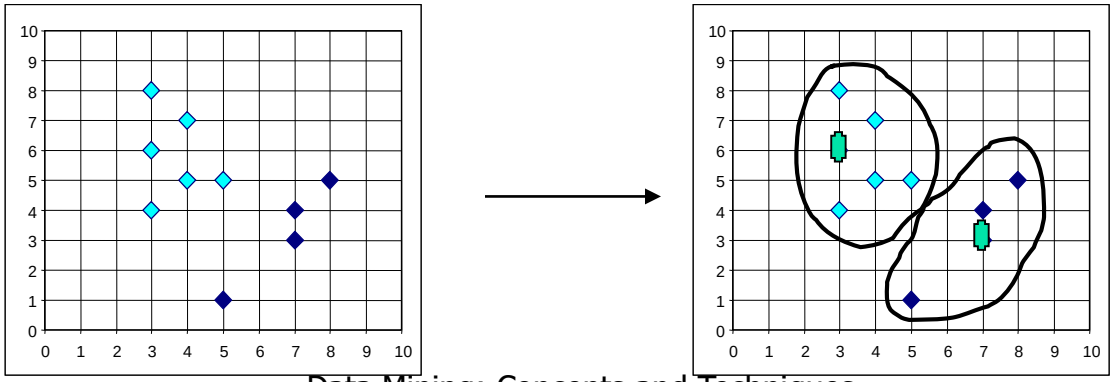
- Implement k-means clustering
- Evaluate the results

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - 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

K-Medoids Method

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the **mean** of the data.
- K-Medoids: Instead of using the **mean** as cluster representative, use **medoid**, the **most centrally located** object in a cluster.
- Possible number of solutions?

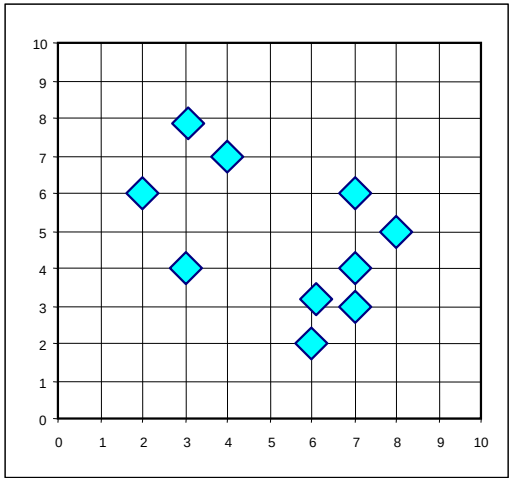


The *K-Medoids* Clustering Method

PAM (Partitioning Around Medoids) (Kaufman and Rousseeuw, 1987)

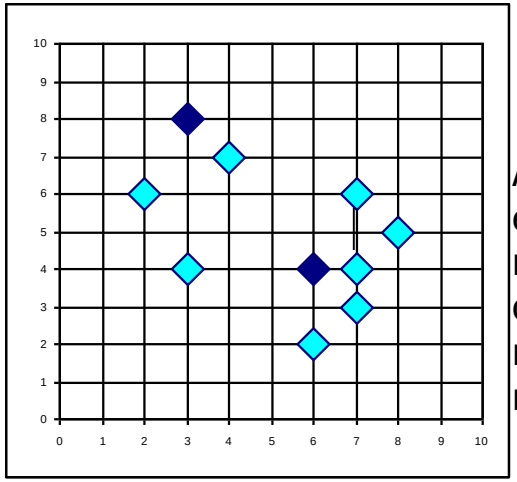
- Arbitrarily select k objects as medoid
- Assign each data object in the given data set to most similar medoid.
- For each nonmedoid object O’ and medoid object O
  - Compute total cost, S, of swapping the medoid object O to O’ (cost as total sum of absolute error)
- If min S<0, then swap O with O’
- Repeat until there is no change in the medoids.

A Typical K-Medoids Algorithm (PAM)



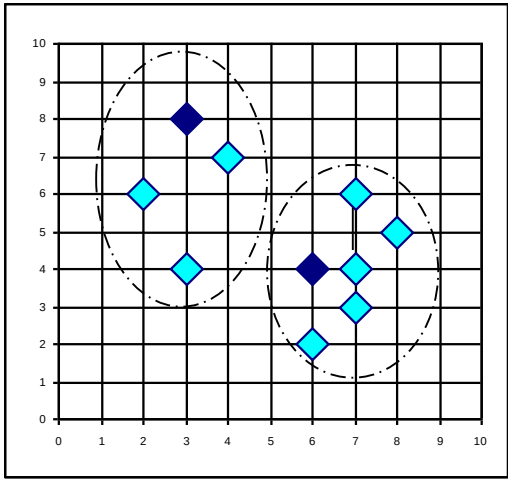
K=2

Arbitrary choose k object as initial medoids

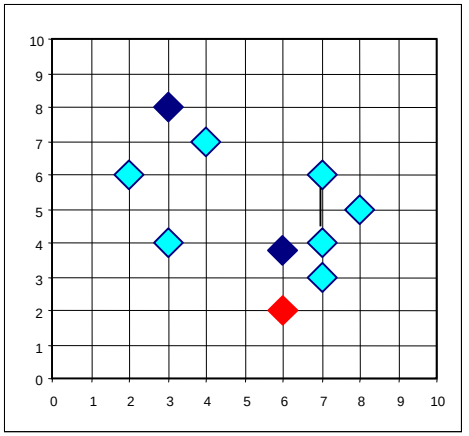


Assign each remaining object to nearest medoids

Total Cost = 20

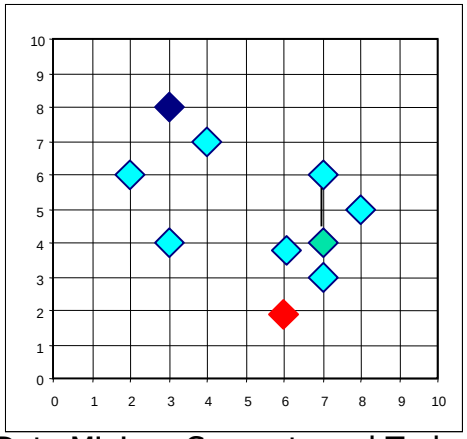


Select a nonmedoid object,O_{random}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

What Is the Problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - Complexity?

    n is # of data, k is # of clusters

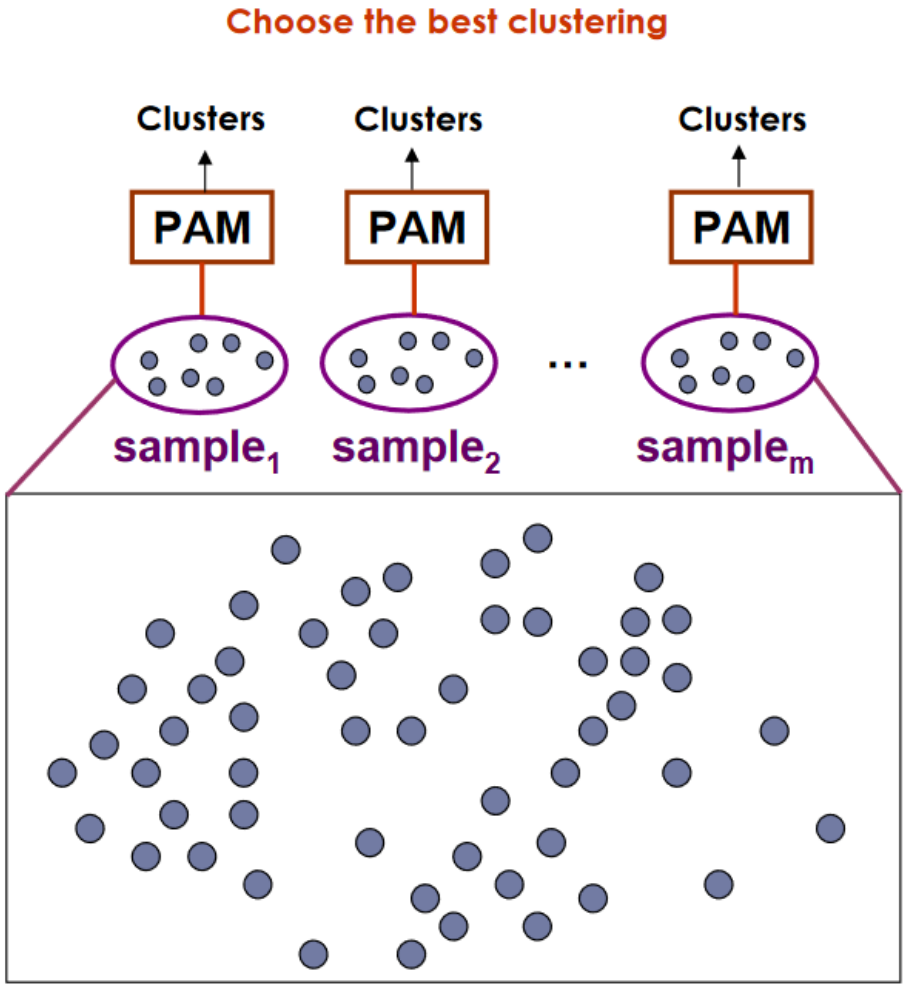
What Is the Problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - Complexity? $O(k(n-k)^2)$

    n is # of data, k is # of clusters

CLARA (Clustering Large Applications) (1990)

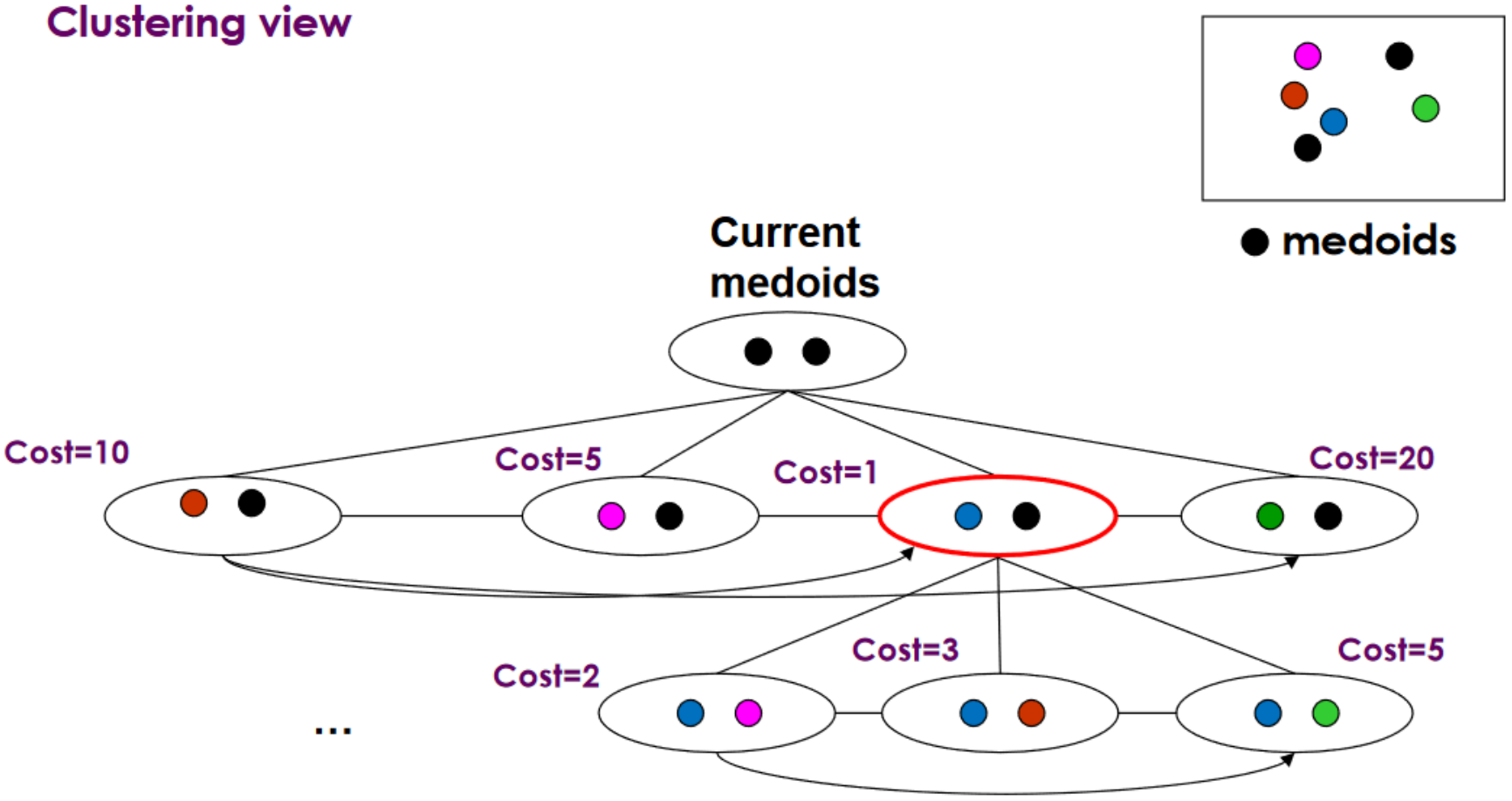
- *CLARA* (Kaufmann and Rousseeuw in 1990)
- Draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output



CLARANS ("Randomized" CLARA) *(1994)*

- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- The clustering process can be represented as searching a graph where every node is a potential solution, that is, a set of *k* medoids

Search graph



CLARANS ("Randomized" CLARA) *(1994)*

- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- The clustering process can be represented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
  - PAM examines all neighbors for local minimum
  - CLARA works on subgraphs of samples
  - CLARANS examines neighbors dynamically
    - Limit the neighbors to explore (*maxneighbor*)
    - If local optimum is found, start with new randomly selected node in search for a new local optimum (*numlocal*)

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Similarity and distances
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering

Cluster Evaluation

- Determine clustering tendency of data, i.e. distinguish whether non-random structure exists
- Determine correct number of clusters
- Evaluate the cohesion and separation of the clustering without external information
- Evaluate how well the cluster results are compared to externally known results
- Compare different clustering algorithms/results

Measures

- Unsupervised (internal): Used to measure the goodness of a clustering structure *without* respect to external information.
  - Sum of Squared Error (SSE)
- Supervised (external): Used to measure the extent to which cluster labels match externally supplied class labels.
  - Entropy
- Relative: Used to compare two different clustering results
  - Often an external or internal index is used for this function, e.g., SSE or entropy

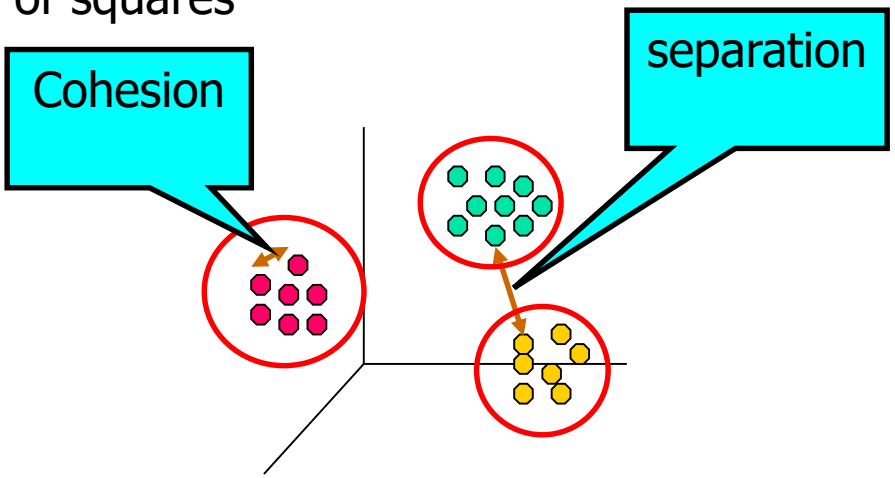
Internal Measures: Cohesion and Separation

- Cluster Cohesion: how closely related are objects in a cluster
- Cluster Separation: how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion: within cluster sum of squares (SSE)

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

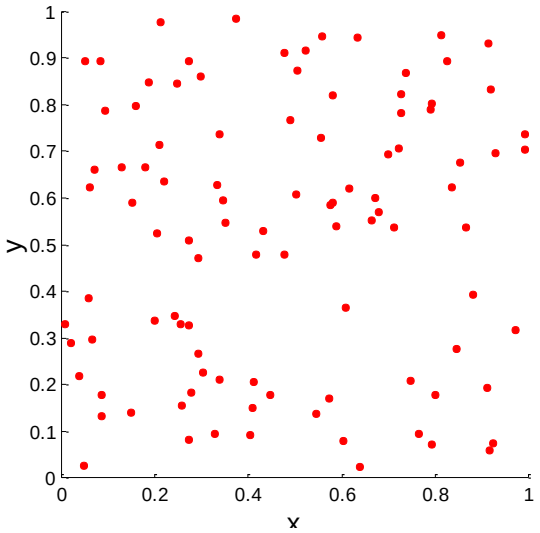
  - Separation: between cluster sum of squares

$$BSS = \sum_{i} \sum_{j} (m_i - m_j)^2$$

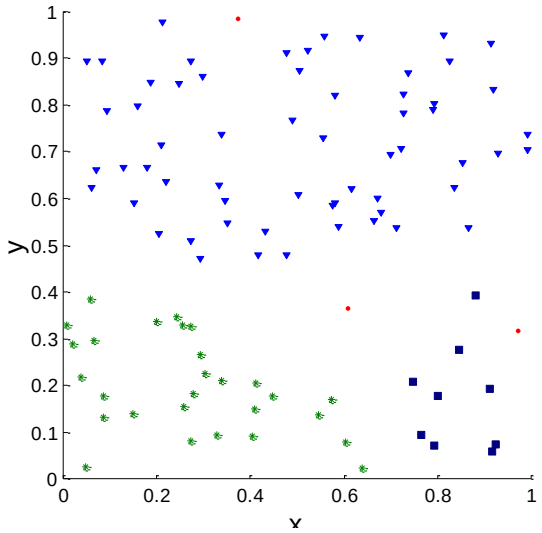


Cluster Validity: Clusters found in Random Data

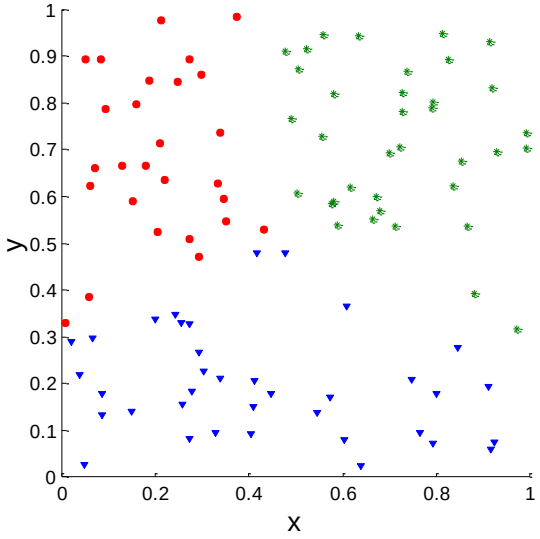
Random Points



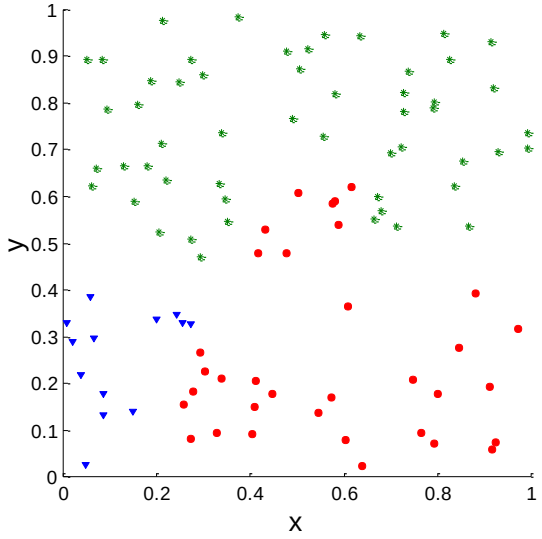
DBSCAN



K-means

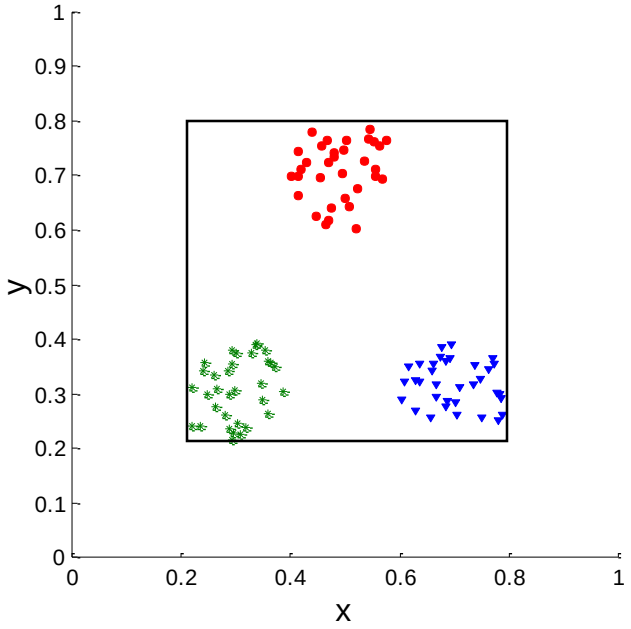


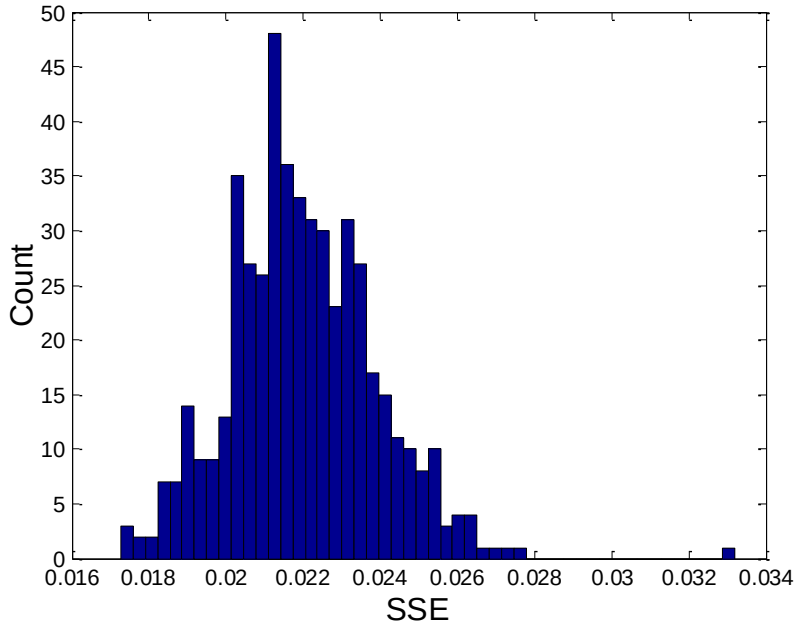
Complete Link



Internal Measures: Cluster Validity

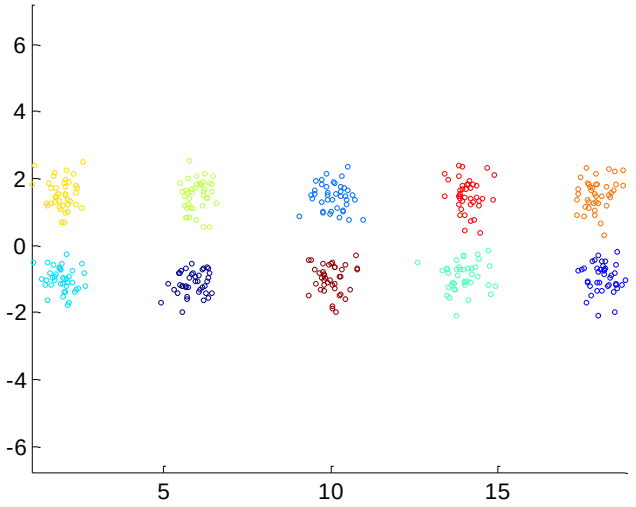
- Statistics framework for cluster validity
  - More "atypical" -> likely valid structure in the data
  - Use values resulting from random data as baseline
- Example
  - Clustering: SSE = 0.005
  - SSE of three clusters in 500 sets of random data points

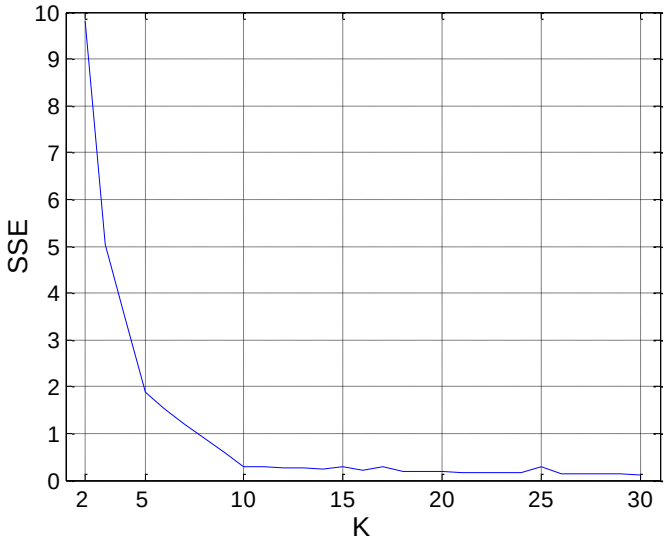




Internal Measures: number of clusters

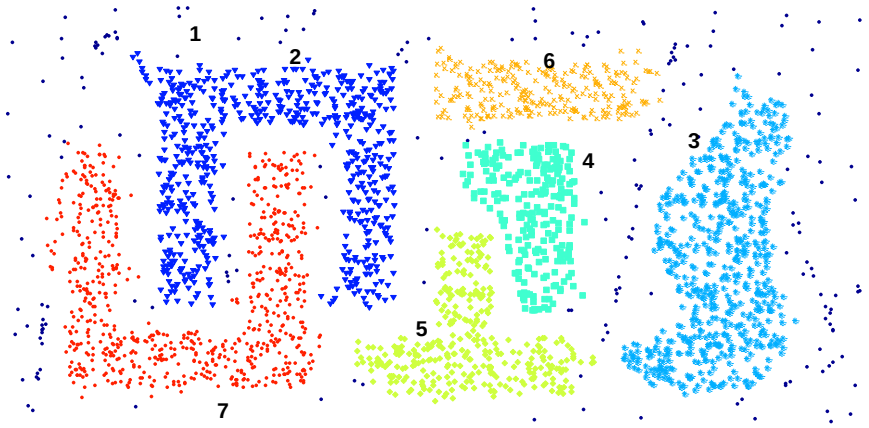
- Good for comparing two clusterings
- Can also be used to estimate the number of clusters
  - Elbow method: use turning point in the curve of SSE wrt # of clusters

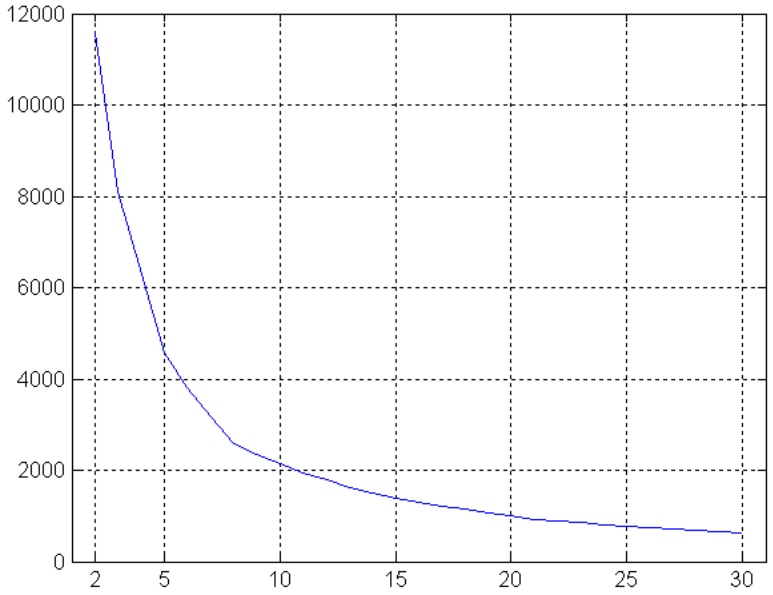




Internal Measures: Number of clusters

- Another example of a more complicated data set with varying number of clusters





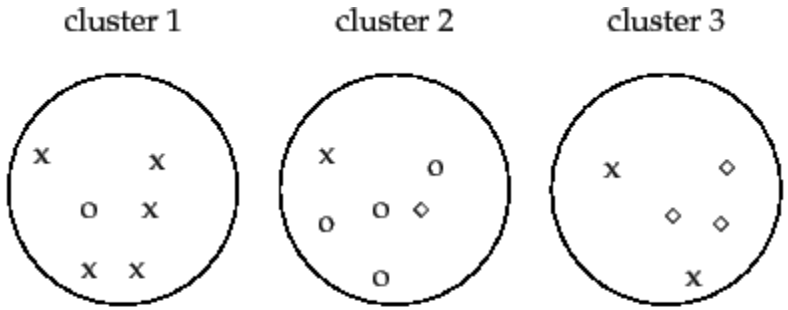
SSE of clusters found using K-means

External Measures

- Compare cluster results with "ground truth" or manually clustering
- Still different from classification measures
- Classification-oriented measures: entropy/purity based, precision and recall based
- Similarity-oriented measures: Jaccard scores

External Measures: Classification-Oriented Measures

- Entropy based measures: the degree to which each *cluster* consists of objects of a single class
- Purity: based on majority class in each cluster



► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ◇, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

External Measures: Classification-Oriented Measures

- BCubed Precision and recall: measures precision and recall associated with each *object*
  - Precision of an object: proportion of objects in the same cluster belong to the same category
  - Recall of an object: proportion of objects of the same category are assigned to the same cluster
  - Bcubed precision and recall are the average precision and recall of all objects

BCubed precision and recall

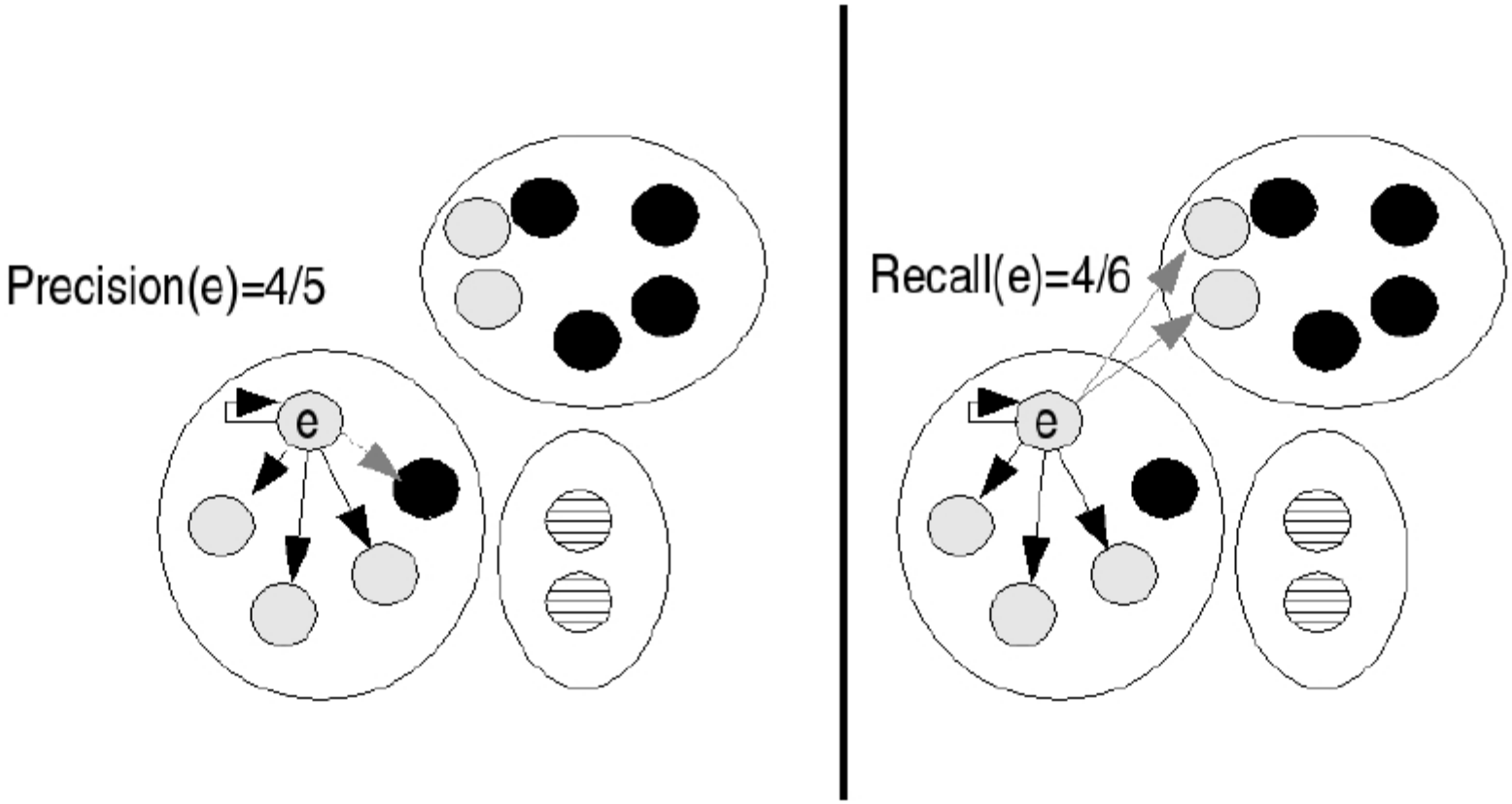


Figure 10: Example of computing the BCubed precision and recall for one item

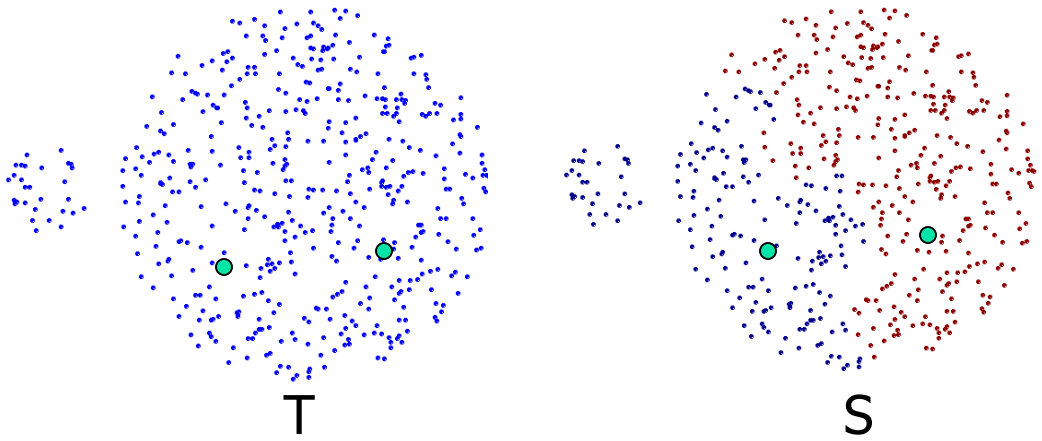
External Measure: Similarity-Oriented Measures

Given a reference clustering T and clustering S

- f_{00}: number of pair of points belonging to different clusters in both T and S
- f_{01}: number of pair of points belonging to different cluster in T but same cluster in S
- f_{10}: number of pair of points belonging to same cluster in T but different cluster in S
- f_{11}: number of pair of points belonging to same clusters in both T and S

$$Rand = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

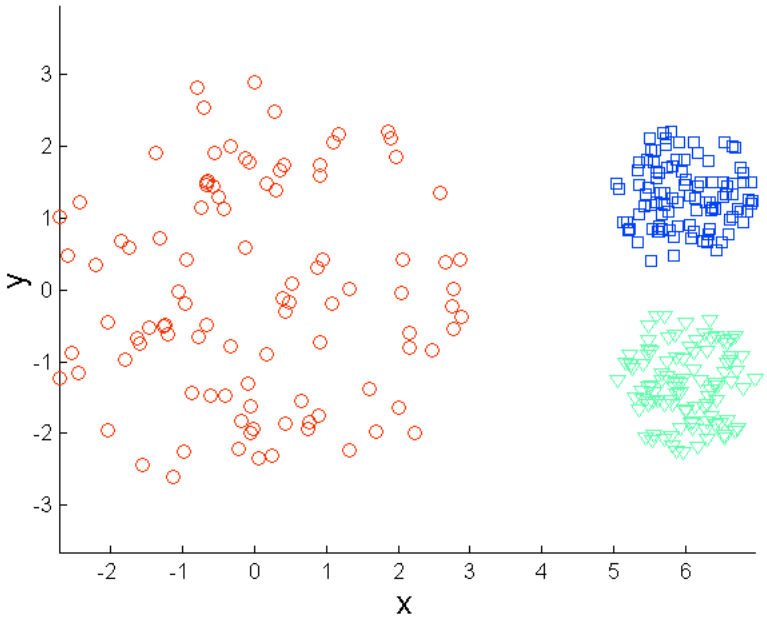
$$Jaccard = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$



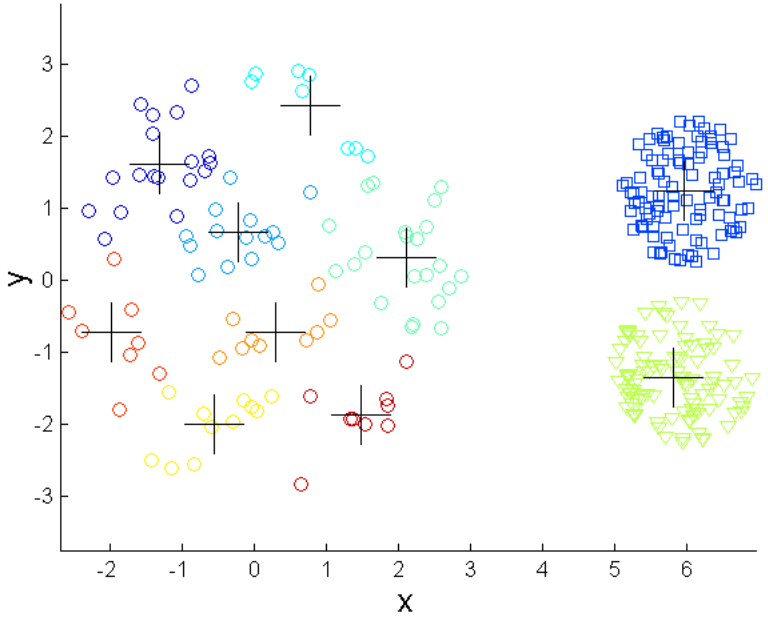
Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Similarity and distances
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering

Overcoming K-means Limitations

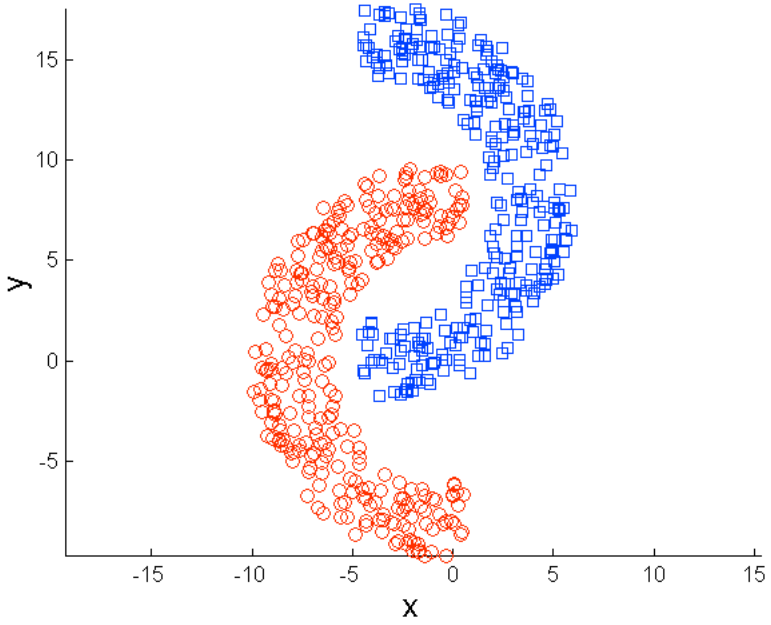


Original Points

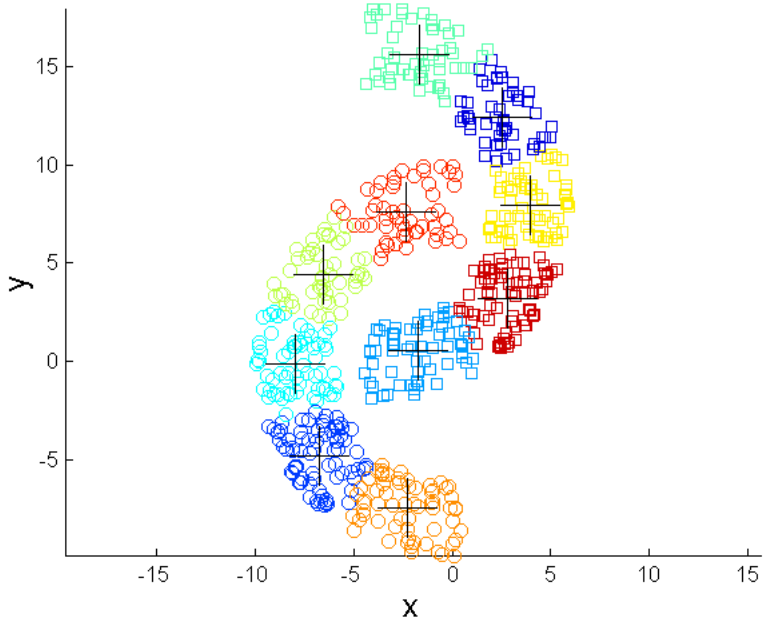


K-means Clusters

Overcoming K-means Limitations



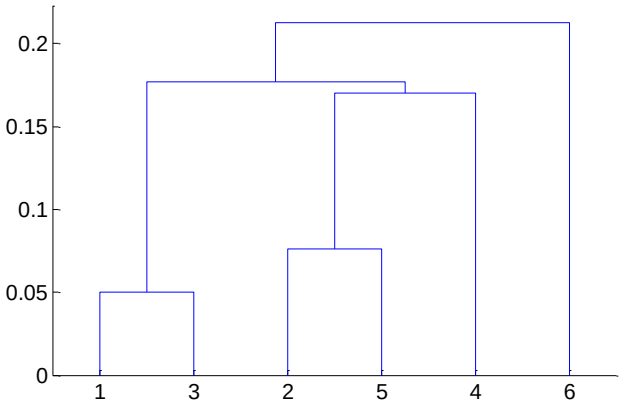
Original Points

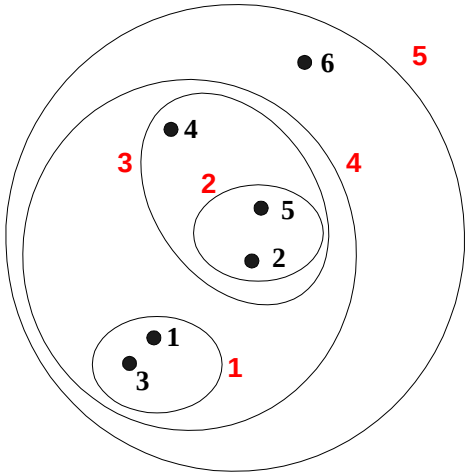


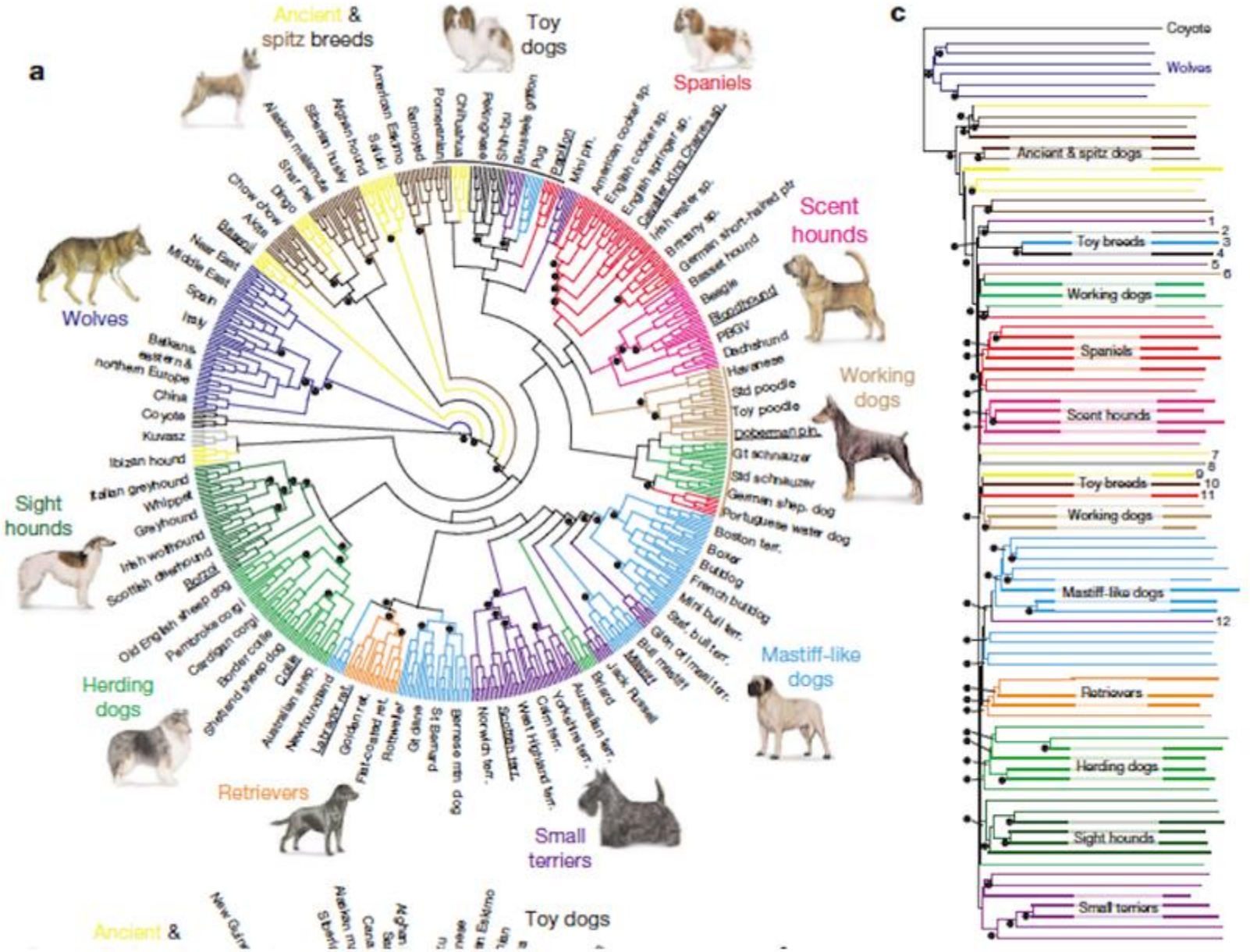
K-means Clusters

Hierarchical Clustering

- Produces a set of nested clusters
- Can be visualized as a dendrogram, a tree like diagram
  - Y-axis measures closeness
  - Clustering obtained by cutting at desired level
- Do not have to assume any particular number of clusters
- May correspond to meaningful taxonomies






a
Ancient & spitz breeds
Toy dogs
Spaniels
Scent hounds
Working dogs
Mastiff-like dogs
Small terriers
Retrievers
Herding dogs
Sight hounds
Wolves
Ancient & spitz dogs
Toy breeds
Working dogs
Spaniels
Scent hounds
Toy breeds
Working dogs
Mastiff-like dogs
Retrievers
Herding dogs
Sight hounds
Small terriers
Coyote
Wolves

Hierarchical Clustering

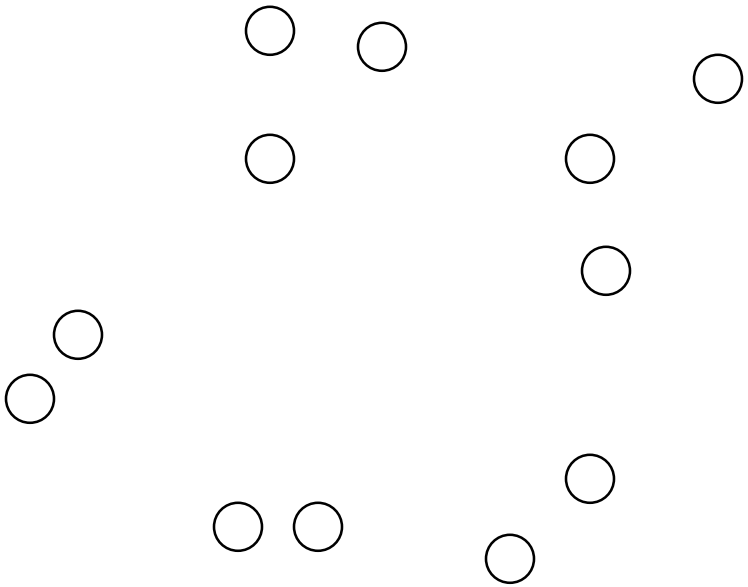
- Two main types of hierarchical clustering
  - Agglomerative (AGNES)
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive (DIANA)
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)

Agglomerative Clustering Algorithm

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

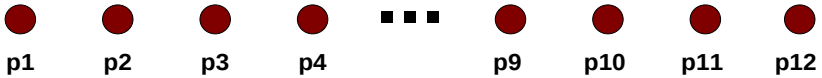
Starting Situation

- Start with clusters of individual points and a proximity matrix

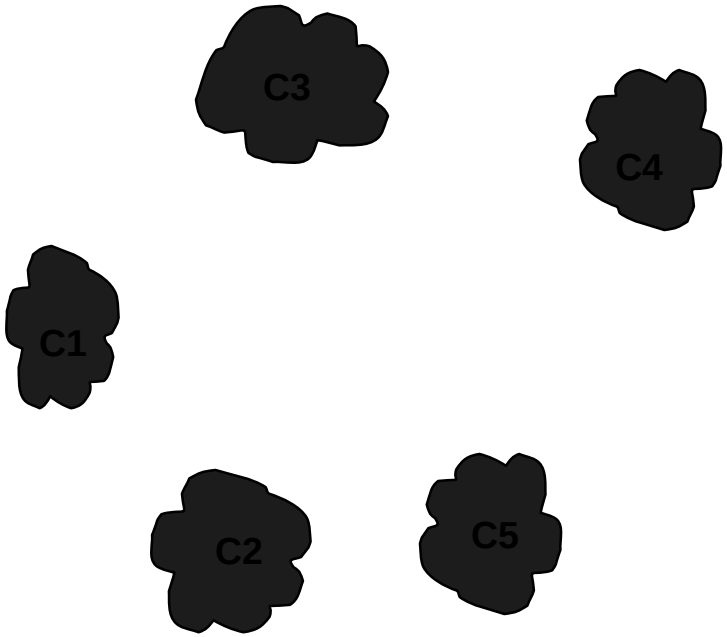


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

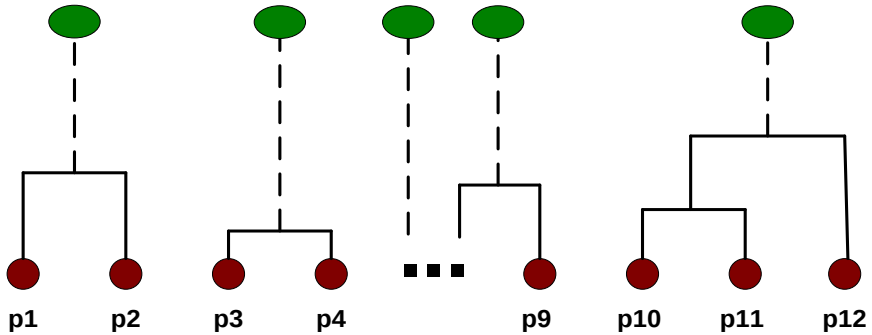


Intermediate Situation

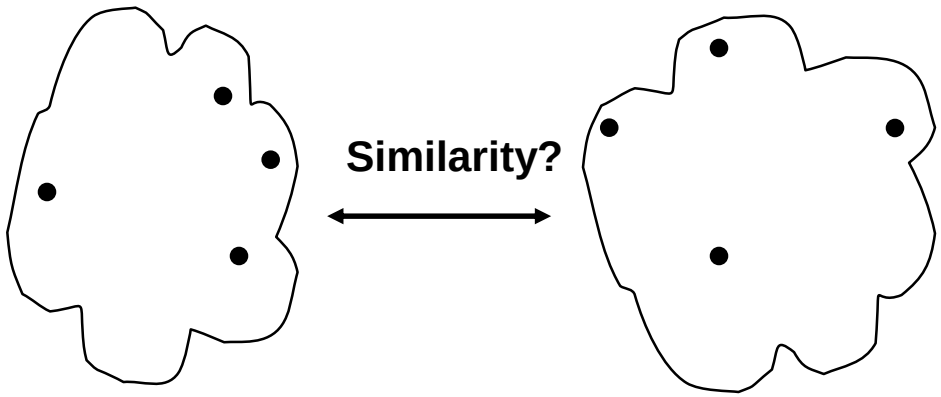


	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



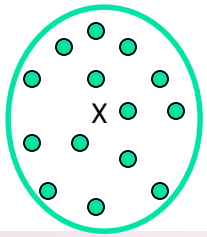
How to Define Inter-Cluster Similarity

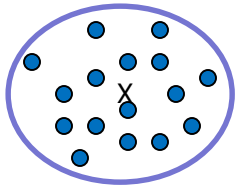


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

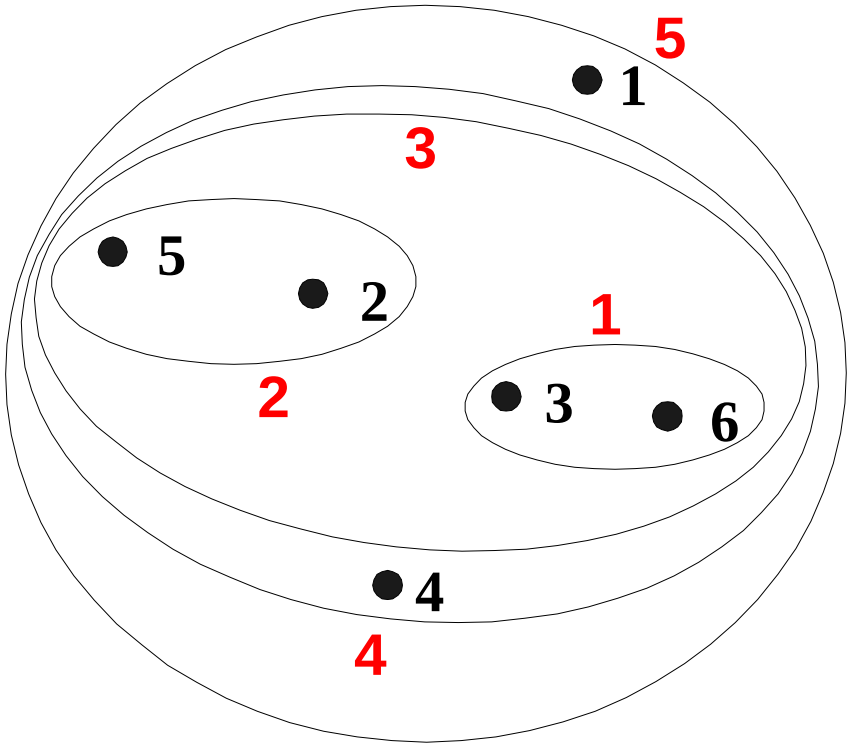
Distance between Clusters

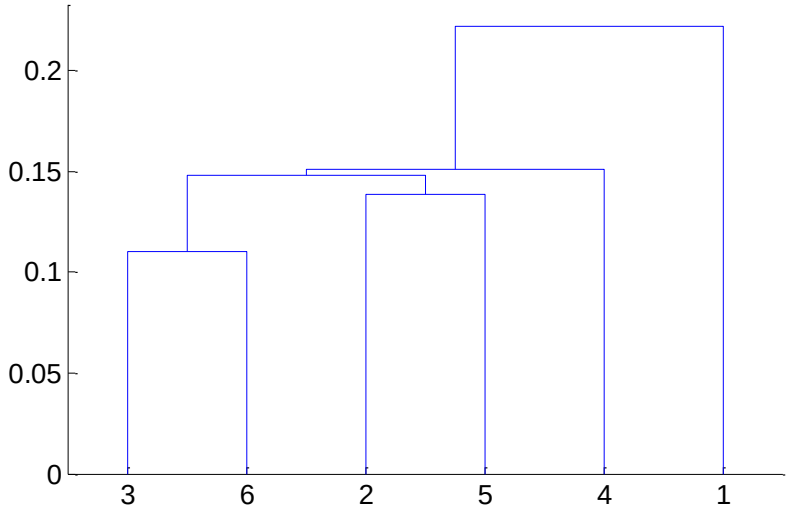




- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: avg distance 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

Hierarchical Clustering: MIN



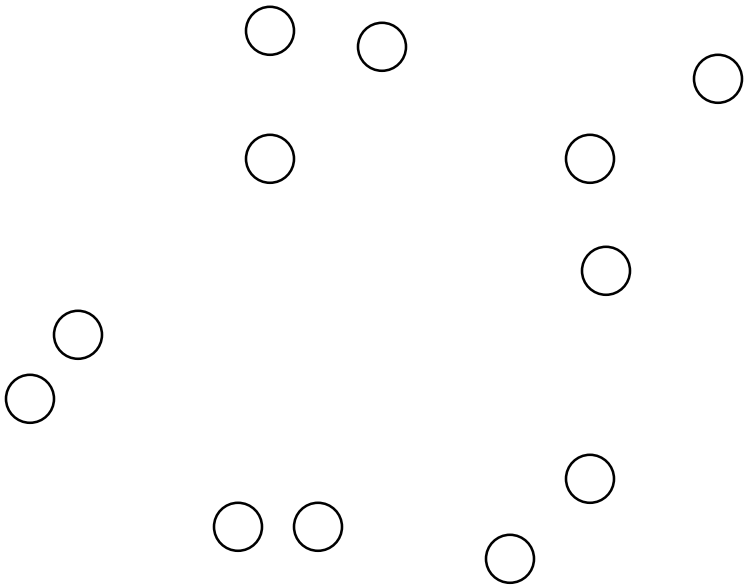


Nested Clusters

Dendrogram

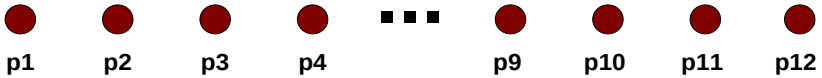
View points/similarities as a graph

- Start with clusters of individual points and a proximity matrix



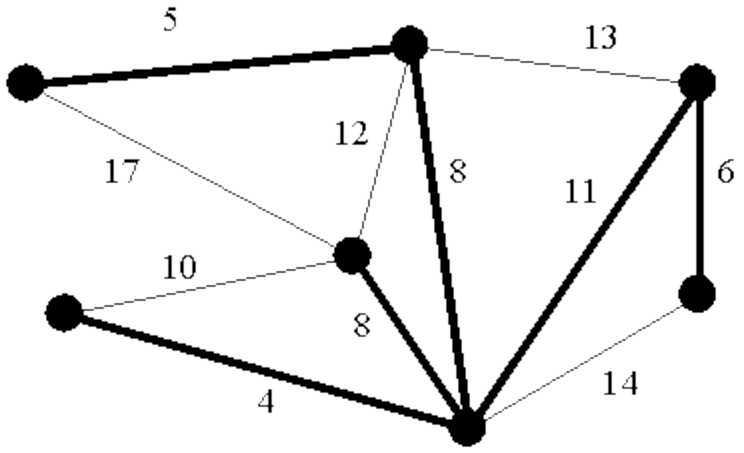
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

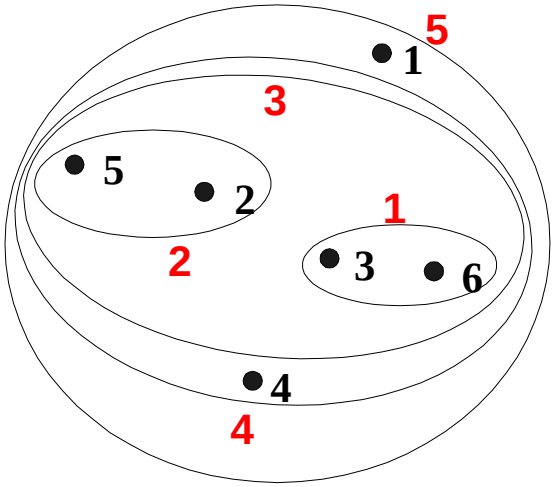


Single link clustering and MST (Minimum Spanning Tree)

- An aggolomerative algorithm using minimum distance (single-link clustering) essentially the same as Kruskal's algorithm for minimal spanning tree (MST)
- MST: a subgraph which is a tree and connects all vertices together that has the minimum weight
- Kruskal's algorithm: Add edges in increasing weight, skipping those whose addition would create a cycle
- Prim's algorithm: Grow a tree with any root node, adding the frontier edge with smallest weight

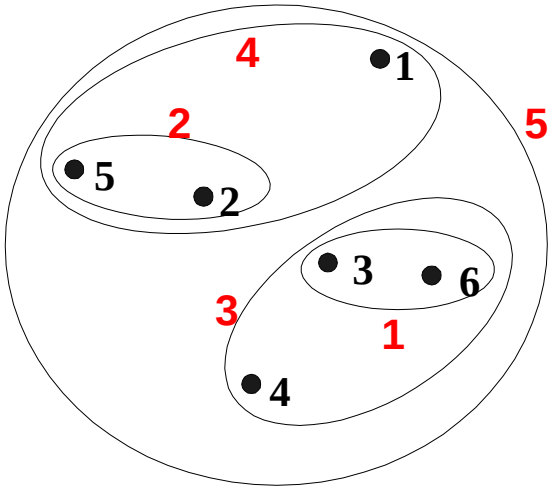


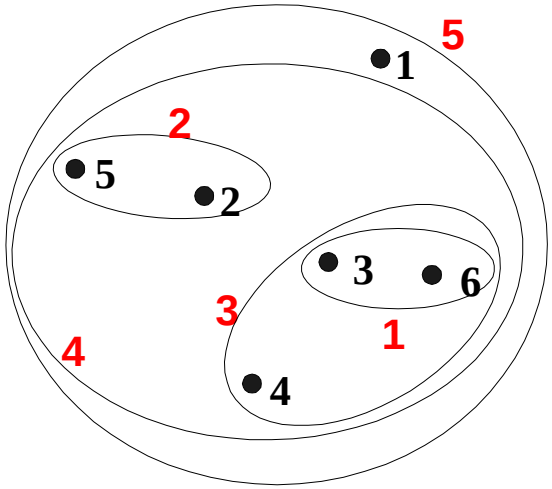
Min vs. Max vs. Group Average



MIN

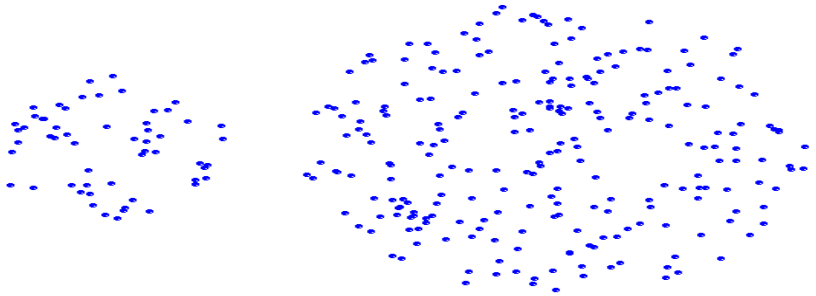
MAX



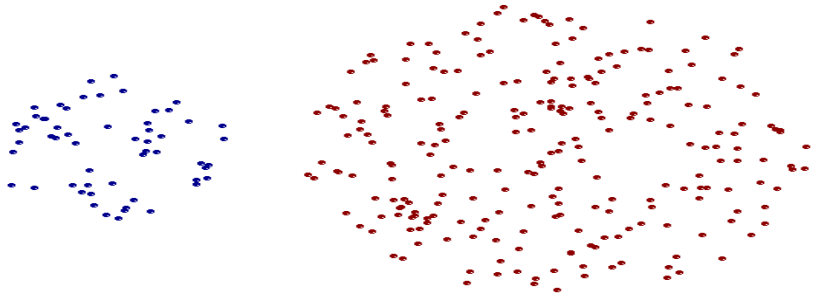


Group Average

Strength of MIN



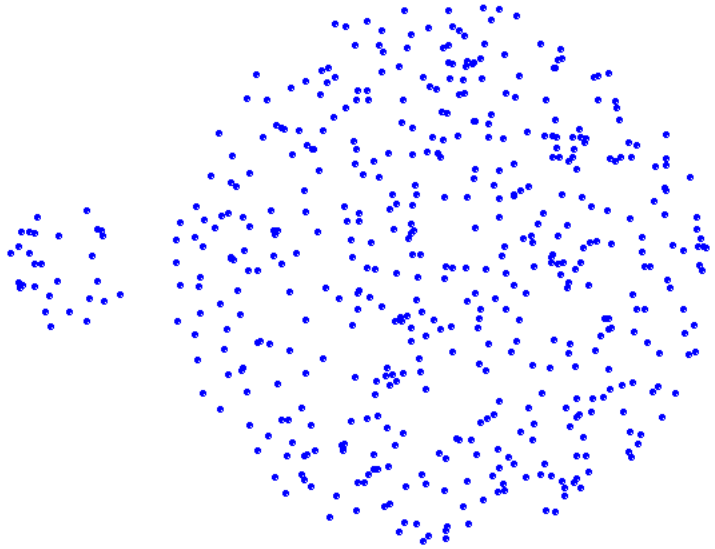
Original Points



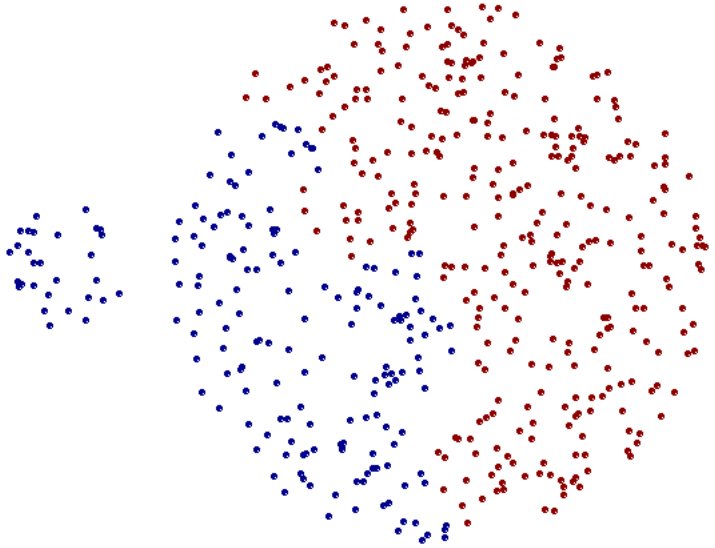
Two Clusters

- Can handle clusters with varying sizes
- Can also handle non-elliptical shapes

Limitations of MAX



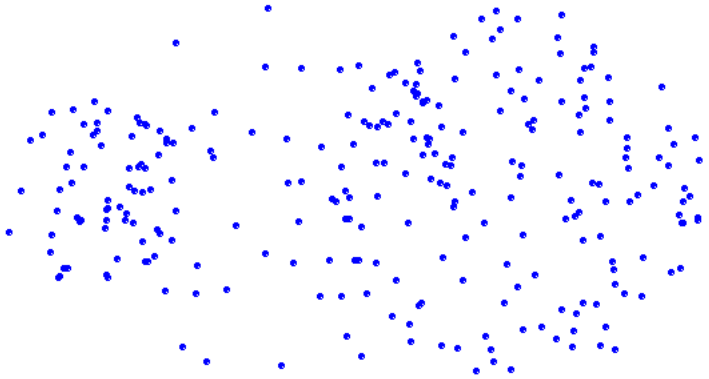
Original Points



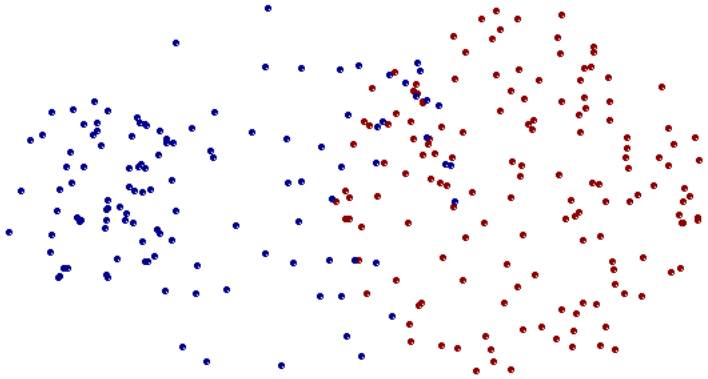
Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

Limitations of MIN



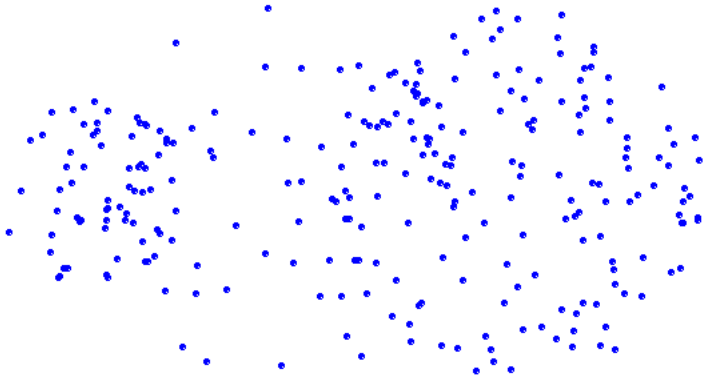
Original Points



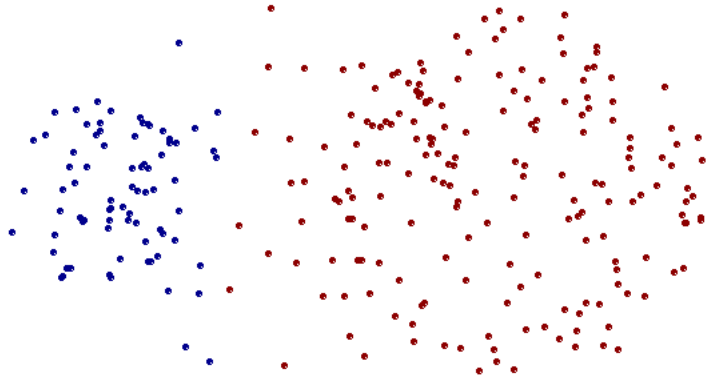
Two Clusters

- **Chaining phenomenon**
- **Sensitive to noise and outliers**

Strength of MAX



Original Points



Two Clusters

- **Less susceptible to noise and outliers**

Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

Hierarchical Clustering: Major Weaknesses

- Do not scale well (N: number of points)
  - Space complexity:
  - Time complexity:

Hierarchical Clustering: Major Weaknesses

- Do not scale well (N: number of points)
  - Space complexity: $O(N^2)$
  - Time complexity: $O(N^3)$

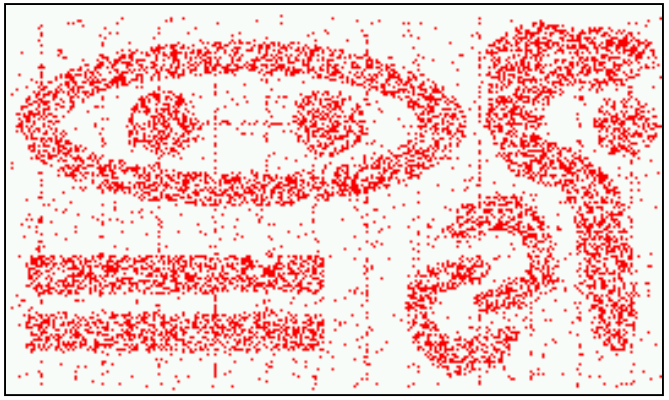
    $O(N^2 \log(N))$ for some cases/approaches

- Cannot undo what was done previously
- Quality varies in terms of distance measures
  - MIN (single link): susceptible to noise/outliers
  - MAX/GROUP AVERAGE: may not work well with non-globular clusters

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Similarity and distances
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering

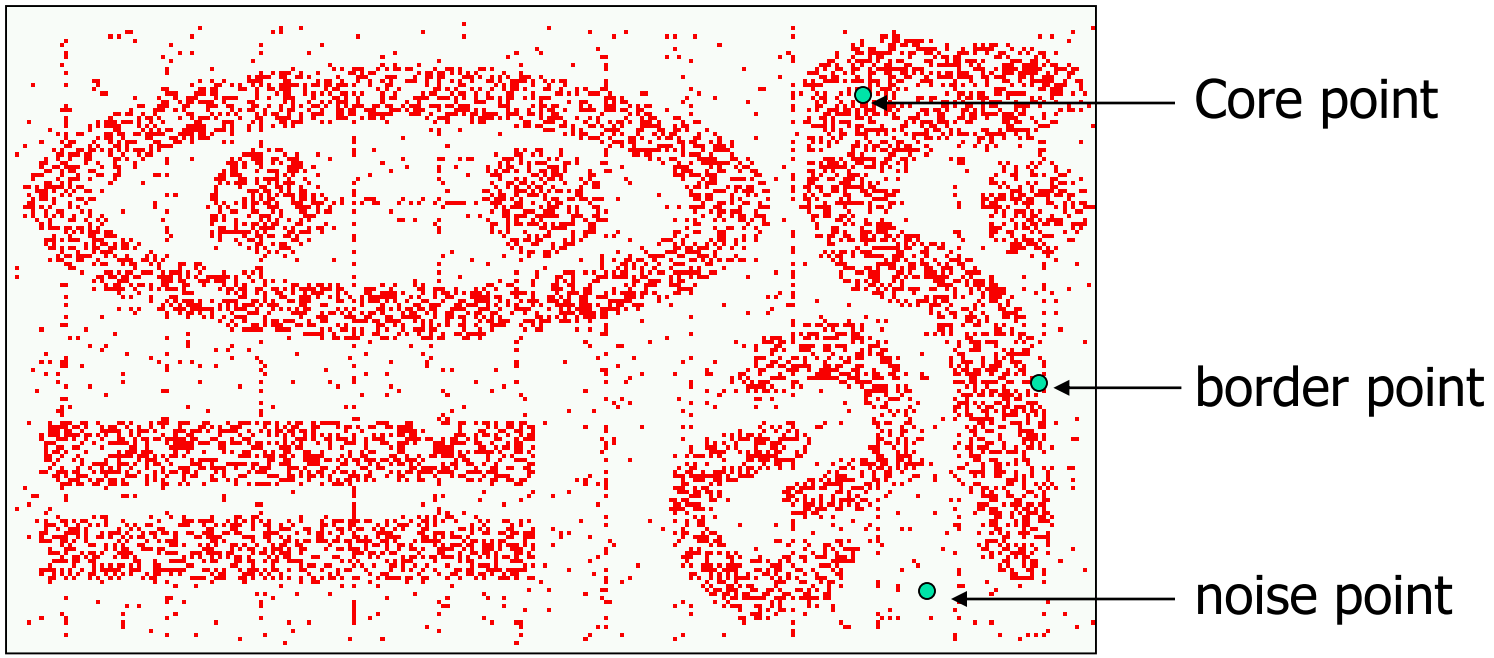
Density-Based Clustering Methods



- Clustering based on density
- Major features:
  - Clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

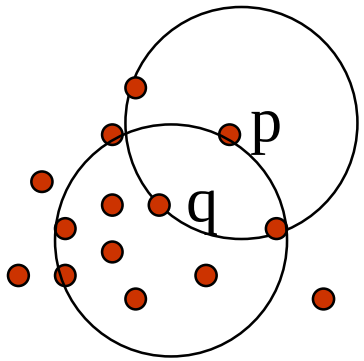
DBSCAN: Basic Concepts

- Density = number of points within a specified radius
- core point: has high density
- border point: has less density, but in the neighborhood of a core point
- noise point: not a core point or a border point.



DBScan: Definitions

- Two parameters*:*
  - *Eps*: radius of the neighbourhood
  - *MinPts*: Minimum number of points in an Eps-neighbourhood of that point
- $N_{Eps}(p)$: *{q belongs to D | dist(p,q) <= Eps}*
- core point: $|N_{Eps}(q)| >= MinPts$

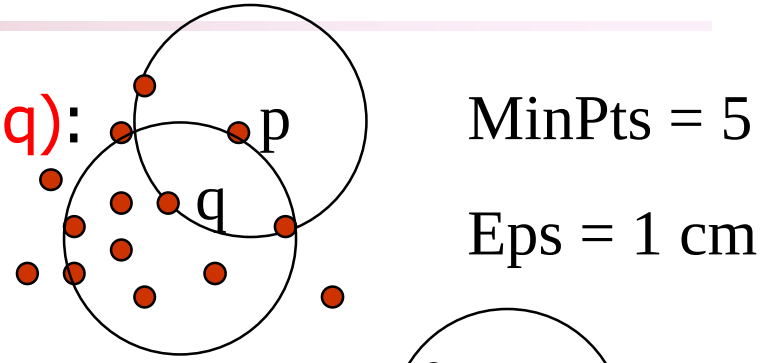


MinPts = 5

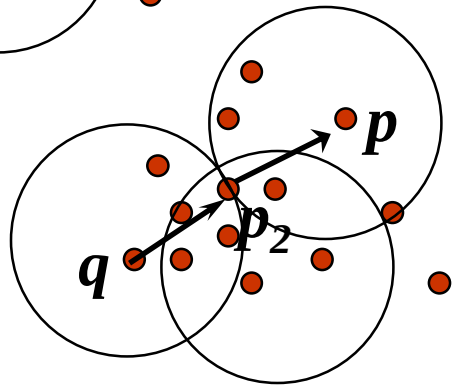
Eps = 1 cm

DBScan: Definitions

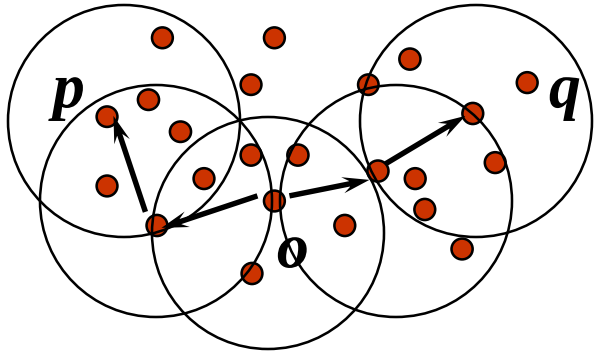
- Directly density-reachable (p from q): *p* belongs to $N_{Eps}(q)$



- Density-reachable (p from q): 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

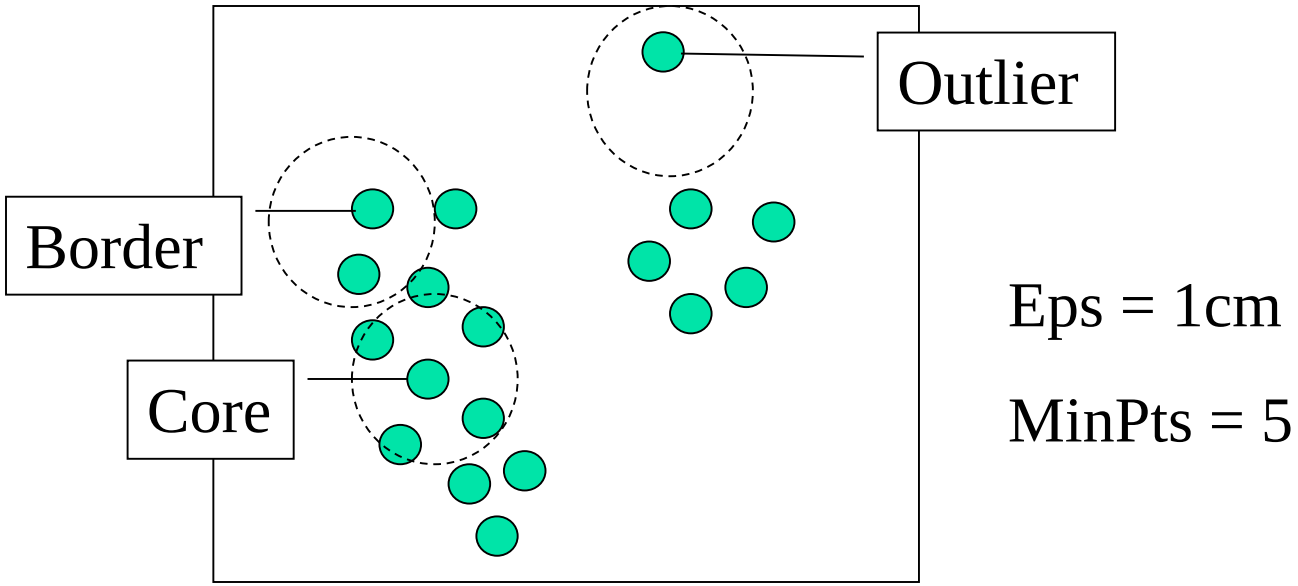


- Density-connected (p and q): if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



DBSCAN: Cluster Definition

- A *cluster* is defined as a maximal set of density-connected points



DBSCAN: The Algorithm

- Arbitrary select an unvisited point *p,* retrieve all neighbor points density-reachable from *p* w.r.t. *Eps* and *MinPts*
- If *p* is a core point, a cluster is formed, add all neighbors of p to the cluster, and recursively add their neighbors if they are a core point
- Otherwise, mark p as a noise point
- Continue the process until all of the points have been processed.
- Complexity: *$O(n^2)$. If a spatial index is used, O(nlogn)*

DBSCAN: Sensitive to Parameters

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

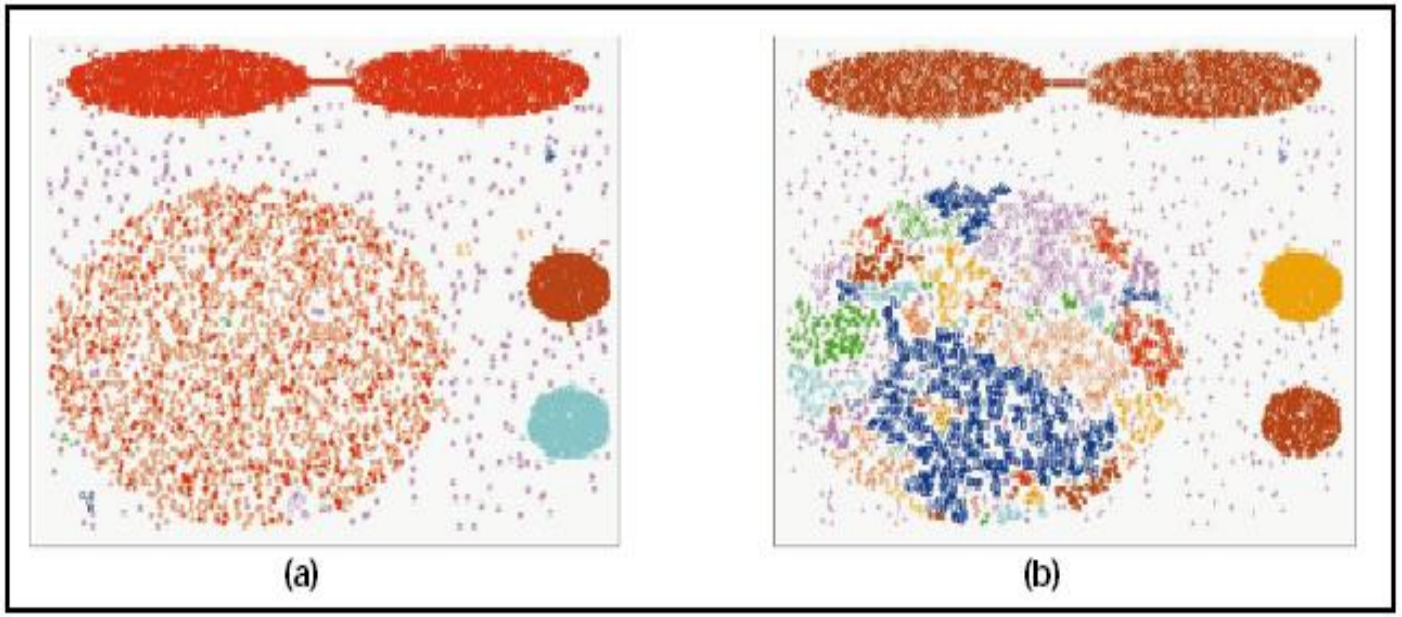
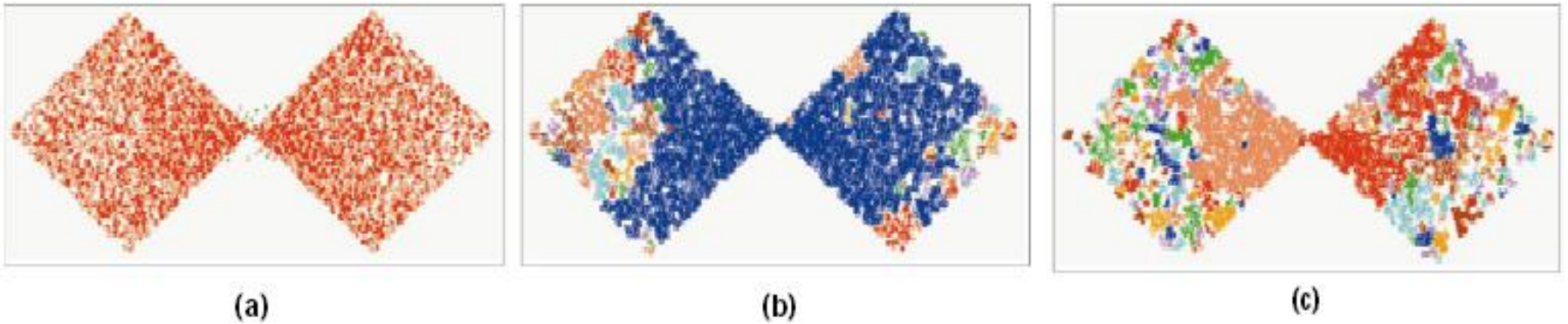
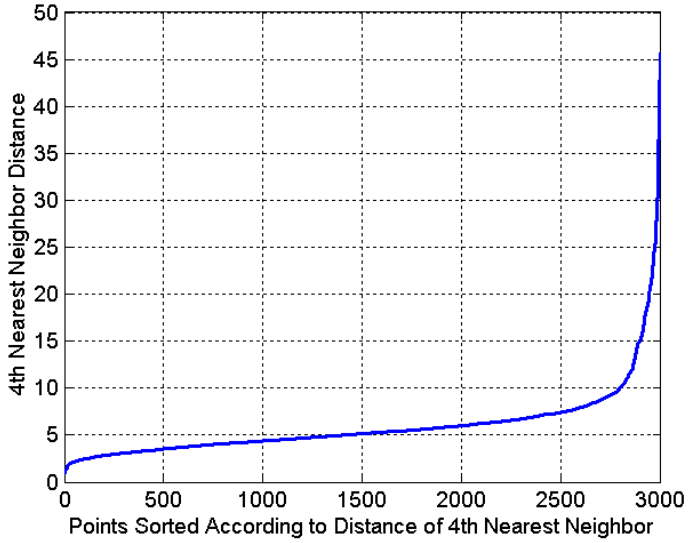


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



DBSCAN: Determining EPS and MinPts

- Basic idea (given MinPts = k, find eps):
  - For points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
  - Noise points have the k^{th} nearest neighbor at farther distance
- Plot sorted distance of every point to its k^{th} nearest neighbor

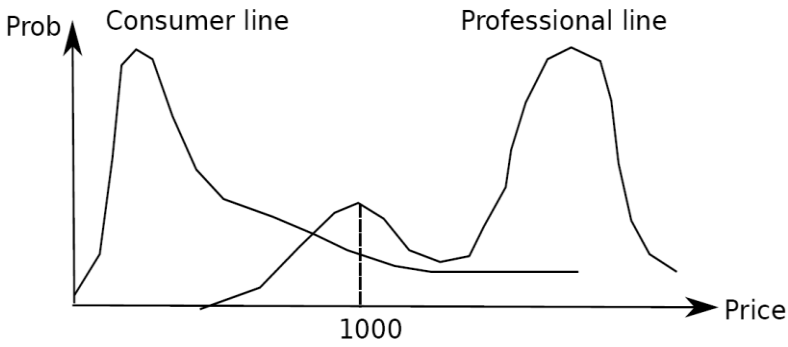


Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Similarity and distances
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering

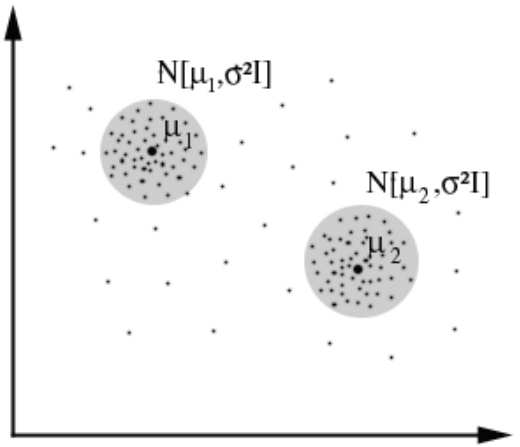
Probabilistic Model-Based Clustering

- Data are instances of underlying hidden categories
- Cluster analysis is to find hidden categories.
- A hidden category is a distribution over the data space, which can be mathematically represented using a probability density function (or distribution function).
- Ex. 2 categories for digital cameras sold
  - consumer line vs. professional line
  - density functions f_1, f_2 for C_1, C_2
  - obtained by probabilistic clustering



Clustering by Mixture Model

- A **mixture model** assumes data are generated by a mixture of probabilistic models
- Each cluster can be represented by a probabilistic model
  - e.g. a Gaussian (continuous) or a Poisson (discrete) distribution
- Data generation process: each observed object is generated independently
  - Choose a cluster, Cj, according to probabilities $\omega_1, \ldots, \omega_k$
  - Choose an instance of Cj according to its probability density function fj
- **Our task**: infer a set of *k* probabilistic models that is mostly likely to generate the data



Model-Based Clustering

- A set C of k probabilistic clusters $C_1, \ldots, C_k$ with probability density functions $f_1, \ldots, f_k$, respectively, and their probabilities $\omega_1, \ldots, \omega_k$.
- Probability of an object o generated by cluster C_j is $P(o|C_j) = \omega_j f_j(o)$
- Probability of o generated by the set of cluster $\boldsymbol{C}$ is $P(o|\boldsymbol{C}) = \sum_{j=1}^{k} \omega_j f_j(o)$
- Since objects are assumed to be generated independently, for a data set D = {o_1, …, o_n}, we have,

$$P(D|\boldsymbol{C}) = \prod_{i=1}^{n} P(o_i|\boldsymbol{C}) = \prod_{i=1}^{n} \sum_{j=1}^{k} \omega_j f_j(o_i)$$

- **Task**: Find a set C of k probabilistic clusters s.t. $P(D|\boldsymbol{C})$ is maximized

Univariate Gaussian Mixture Model

- O = {o_1, …, o_n} (n observed objects), Θ = {θ_1, …, θ_k} (parameters of the k distributions), and $P_j(o_i|\theta_j)$ is the probability that o_i is generated from the j-th distribution using parameter θ_j, we have

$$P(o_i|\Theta) = \sum_{j=1}^{k} \omega_j P_j(o_i|\Theta_j) \qquad P(\mathbf{O}|\Theta) = \prod_{i=1}^{n} \sum_{j=1}^{k} \omega_j P_j(o_i|\Theta_j)$$

- Univariate Gaussian mixture model
  - Assume the probability density function of each cluster follows a 1-d Gaussian distribution. Suppose that there are k clusters with 1/k prob.
  - The probability density function of each cluster are centered at μ_j with standard deviation σ_j, θ_j, = (μ_j, σ_j), we have

$$P(o_i|\Theta_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i-\mu_j)^2}{2\sigma^2}} \qquad P(o_i|\Theta) = \sum_{j=1}^{k} \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i-\mu_j)^2}{2\sigma^2}}$$

$$P(\mathbf{O}|\Theta) = \prod_{i=1}^{n} \sum_{j=1}^{k} \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(o_i-\mu_j)^2}{2\sigma^2}}$$

The EM (Expectation Maximization) Algorithm

- **The (EM) algorithm:** A framework to approach maximum likelihood or maximum a posteriori estimates of parameters in statistical models.
  - **Expectation-step** assigns objects to clusters according to the current clustering or parameters of probabilistic clusters
  - **Maximization-step** finds the new clustering or parameters that maximize the expected likelihood

Computing Mixture Models with EM

- Given n objects $O = \{o_1, \ldots, o_n\}$, we want to infer a set of parameters $\Theta = \{\theta_1, \ldots, \theta_k\}$ s.t.,$P(\mathbf{O}|\mathbf{\Theta})$ is maximized, where $\theta_j = (\mu_j, \sigma_j)$ are the mean and standard deviation of the j-th univariate Gaussian distribution
- We initially assign random values to parameters θ_j, then iteratively conduct the Expectation (E) and Maximization (M) steps until converge
- At the E-step, for each object o_i, calculate the probability that o_i belongs to each distribution,

$$P(\Theta_j|o_i, \mathbf{\Theta}) = \frac{P(o_i|\Theta_j)}{\sum_{l=1}^{k} P(o_i|\Theta_l)}$$

- At the M-step, adjust the parameters $\theta_j = (\mu_j, \sigma_j)$ so that the expected likelihood $P(\mathbf{O}|\mathbf{\Theta})$ is maximized

$$\mu_j = \sum_{i=1}^{n} o_i \frac{P(\Theta_j|o_i, \mathbf{\Theta})}{\sum_{l=1}^{n} P(\Theta_j|o_l, \mathbf{\Theta})} = \frac{\sum_{i=1}^{n} o_i P(\Theta_j|o_i, \mathbf{\Theta})}{\sum_{i=1}^{n} P(\Theta_j|o_i, \mathbf{\Theta})} \quad \sigma_j = \sqrt{\frac{\sum_{i=1}^{n} P(\Theta_j|o_i, \mathbf{\Theta})(o_i - u_j)^2}{\sum_{i=1}^{n} P(\Theta_j|o_i, \mathbf{\Theta})}}$$

The EM (Expectation Maximization) Algorithm

- The k-means algorithm has two steps at each iteration:
  - **Expectation Step** (E-step): Given the current cluster centers, each object is assigned to the cluster whose center is closest to the object: An object is *expected to belong to the closest cluster*
  - **Maximization Step** (M-step): Given the cluster assignment, for each cluster, the algorithm *adjusts the center* so that *the sum of distance* from the objects assigned to this cluster and the new center is minimized

Advantages and Disadvantages of Mixture Models

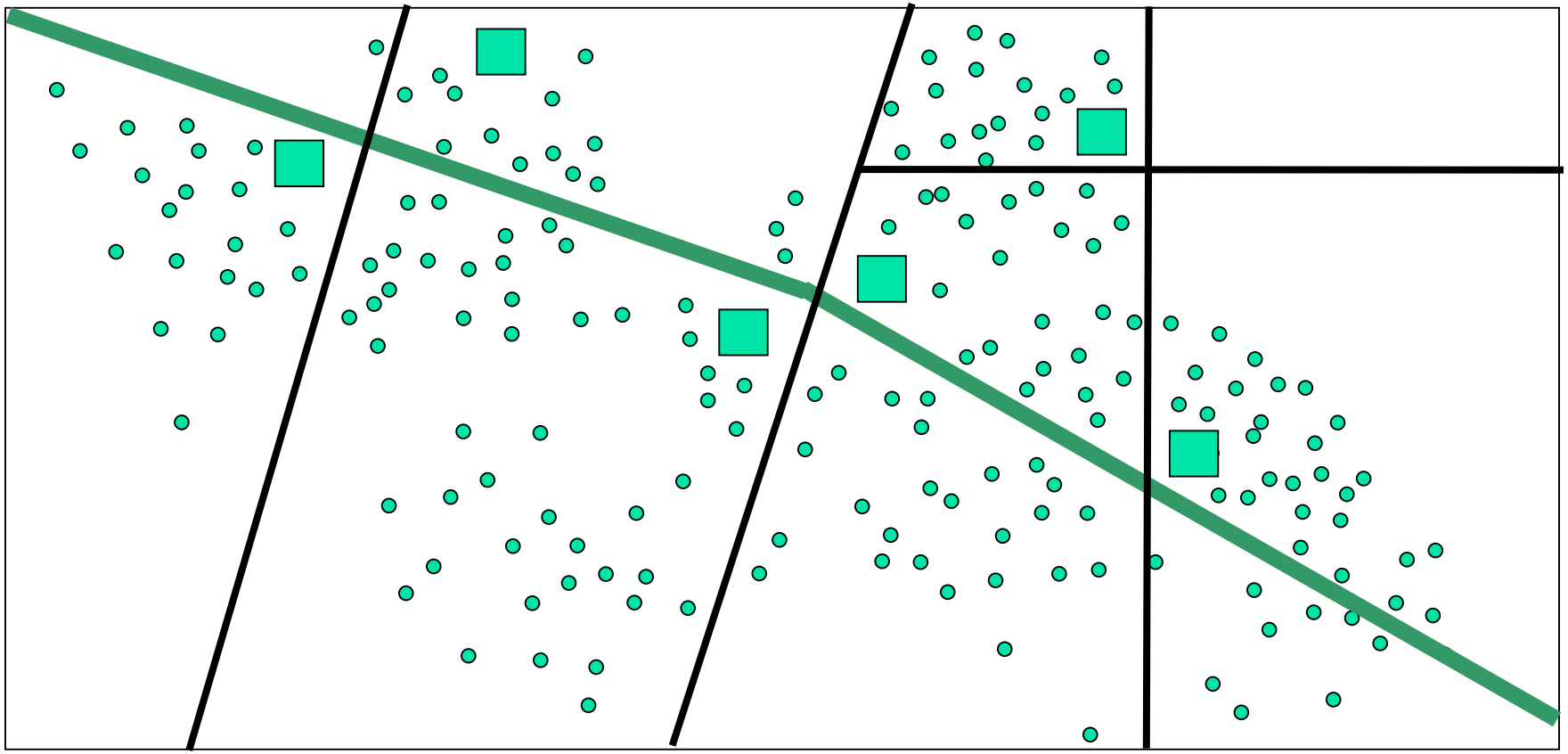
- Strength
  - Mixture models are more general than partitioning methods
  - Clusters can be characterized by a small number of parameters
  - The results may satisfy the statistical assumptions of the generative models
- Weakness
  - Converge to local optimal (overcome: run multi-times w. random initialization)
  - Computationally expensive if the number of distributions is large, or the data set contains very few observed data points
  - Need large data sets
  - Hard to estimate the number of clusters

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Similarity and distances
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Probabilistic Methods
- Evaluation of Clustering
- Clustering with constraint s

Why Constraint-Based Cluster Analysis?

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



Categorization of Constraints

- Constraints on instances: specifies how a pair or a set of instances should be grouped in the cluster analysis
  - Must-link vs. cannot link constraints
    - must-link(x, y): x and y should be grouped into one cluster
  - Constraints can be defined using variables, e.g.,
    - cannot-link(x, y) if dist(x, y) > d
- Constraints on clusters: specifies a requirement on the clusters
  - E.g., specify the min # of objects in a cluster, the max diameter of a cluster, the shape of a cluster (e.g., a convex), # of clusters (e.g., k)
- Constraints on similarity measurements: specifies a requirement that the similarity calculation must respect
  - E.g., driving on roads, obstacles (e.g., rivers, lakes)
- Issues: Hard vs. soft constraints; conflicting or redundant constraints

Constraint-Based Clustering Methods (I): Handling Hard Constraints

- Handling hard constraints: Strictly respect the constraints in cluster assignments
- How to handle must-link and cannot-link constraints in k-means?

Constraint-Based Clustering Methods (I): Handling Hard Constraints

- Handling hard constraints: Strictly respect the constraints in cluster assignments
- How to handle must-link and cannot-link constraints in k-means?
- Example: The COP-k-means algorithm
  - Generate super-instances for must-link objects
    - Compute the transitive closure of the must-link objects
    - Replace all objects in each subset by the mean
    - The super-instance also carries a weight, which is the number of objects it represents
  - Modified cluster assignment for cannot-link constraints
    - Modify the center-assignment process in k-means to a *nearest feasible center assignment*

Constraint-Based Clustering Methods (II): Handling Soft Constraints

- Treated as an optimization problem:
  - When a clustering violates a soft constraint, a penalty is imposed on the clustering
- Overall objective:
  - Optimizing the clustering quality, and minimizing the constraint violation penalty
- Ex. CVQE (Constrained Vector Quantization Error) algorithm
  - Objective function: Sum of distance used in k-means, adjusted by the constraint violation penalties

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, and model-based methods