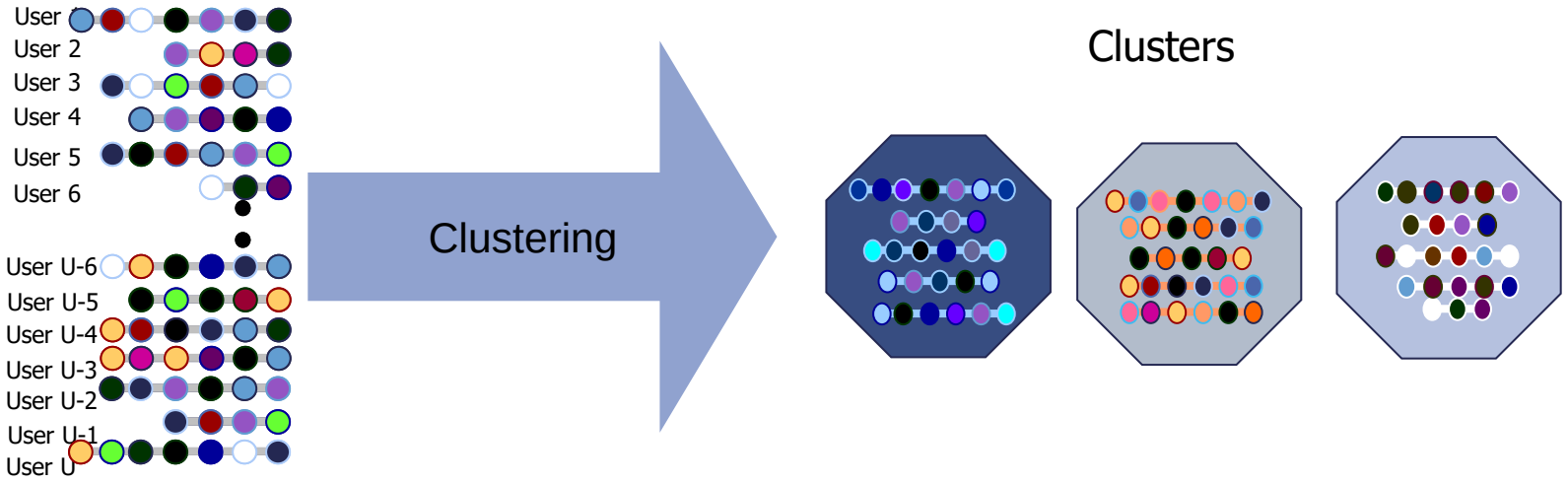
Clustering Analysis

Fernando Calderon
Department of Computer Science and Information Engineering
Fu Jen Catholic University
fhcalderon87@gmail.com

Clustering

- Group data into clusters
  - Similar to the objects within the same cluster
  - Dissimilar to the objects in other clusters
  - No predefined classes (unsupervised classification)



What is not Cluster Analysis?

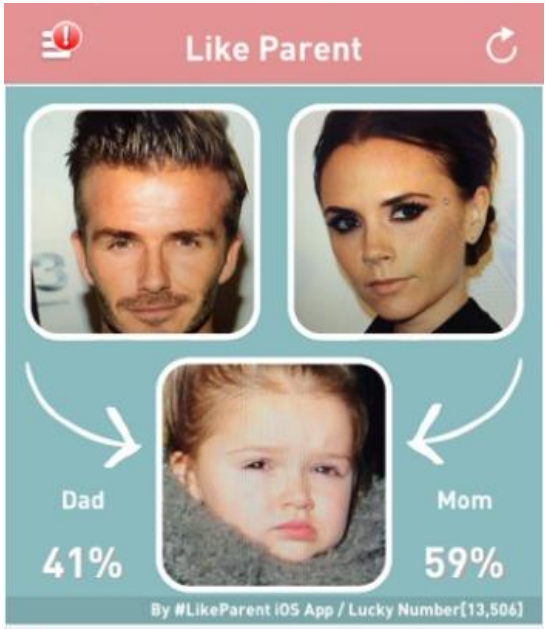
- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically
- Results of a query
  - Groupings are a result of an external specification
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical

Good Clustering

- Good clustering (produce high quality clusters)
  - Intra-cluster similarity is high
  - Inter-cluster similarity is low
- Quality factors
  - Similarity measure and its implementation
  - Definition and representation of cluster chosen
  - Clustering algorithm

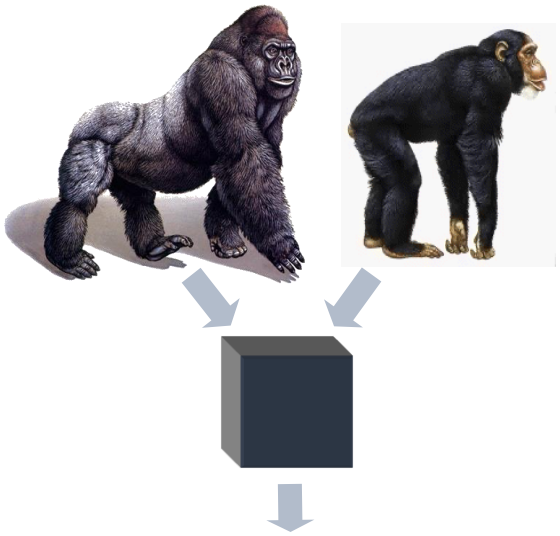
What is Similarity?

- Similarity is hard to define, but…
- "*We know it when we see it*" (??)

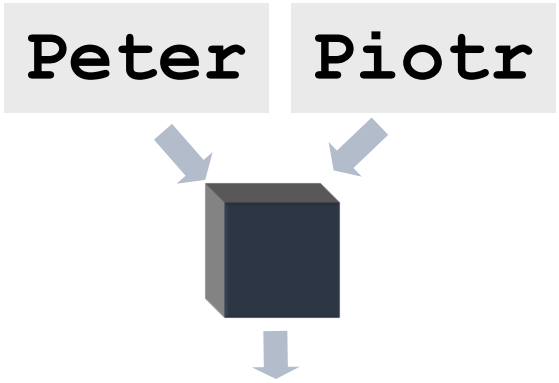


Defining Distance Measures

- **Definition**: Let O_1 and O_2 be two objects from the universe of possible objects. The distance between O_1 and O_2 is a real number denoted by $D(O_1,O_2)$



0.23



3

342.7

Requirements of Clustering

- Scalability
- Insensitivity to order of input records
- Ability to deal with noisy data
- High dimensionality

practical

- Discovery of clusters with arbitrary shape
  - not only sphere

optional

- Dealing with different types of attributes
- Minimal requirements for domain knowledge to input design parameters
- Constraint-based clustering
- Interpretability and usability

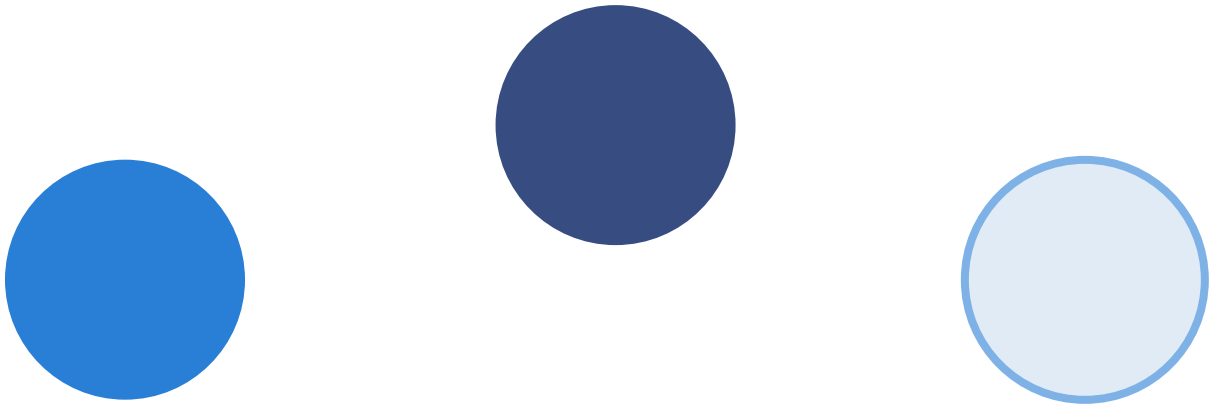
research challenges

Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual
- Described by an Objective Function

Types of Clusters: Well-Separated

- Well-Separated clusters:
  - Points in a cluster are closer than to others not in the cluster



3 well-separated clusters

Types of Clusters: Center-Based

- Center-based
  - Objects in a cluster are closer (more similar) to the “center”
  - The center of a cluster is often a:
    - Centroid: The average of all the points in the cluster
    - Medoid: The most representative point of a cluster

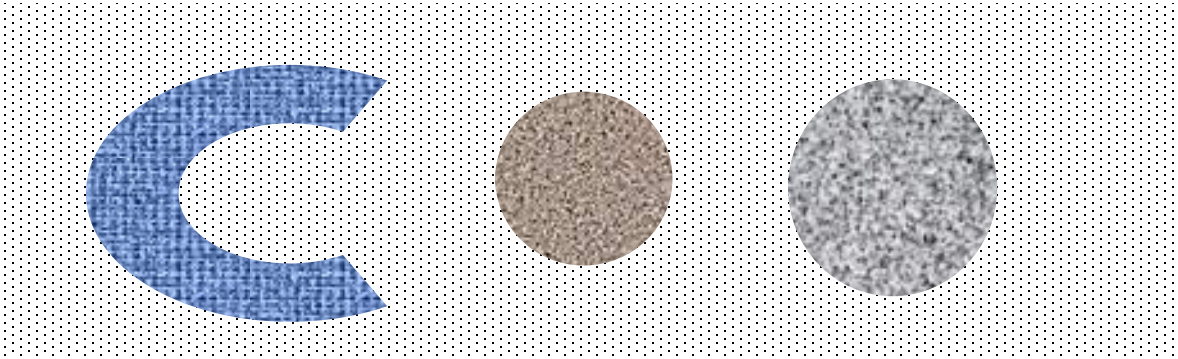
4 center-based clusters

Types of Clusters: Contiguity-Based

- Contiguous cluster (Nearest neighbor or transitive)
  - Points in a cluster is closer to one or more other points in the cluster

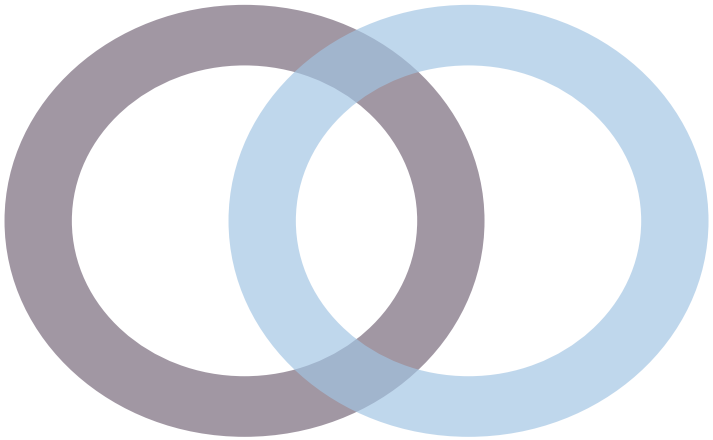
Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region from other regions of high density
    - Separated by low-density regions
  - Used when
    - The clusters are irregular or intertwined
    - Noise and outliers are present



Types of Clusters: Conceptual Clusters

- Shared property or conceptual clusters
  - Clusters share some common property



2 Overlapping Circles

Types of Clusters: Objective Function

- Clusters defined by an objective function
  - Finds clusters that minimize or maximize an objective function
  - Naïve approaches:
    - Enumerate all possible ways
    - Evaluate the `goodness' of each potential set of clusters

    →NP Hard

  - Can have global or local objectives
    - Hierarchical clustering algorithms typically have local objectives
    - Partitioned algorithms typically have global objectives

Five Categories of Clustering Methods

- Partitioning algorithms
  - Construct various partitions
  - Evaluate partitions by some criterion
- Hierarchy algorithms
  - Create a hierarchical decomposition using some criterion
- Density-based
  - Based on connectivity and density functions
- Grid-based
  - Based on a multiple-level granularity structure
- Model-based
  - A model is hypothesized for each of the clusters
  - Find the best fit of that model to each other

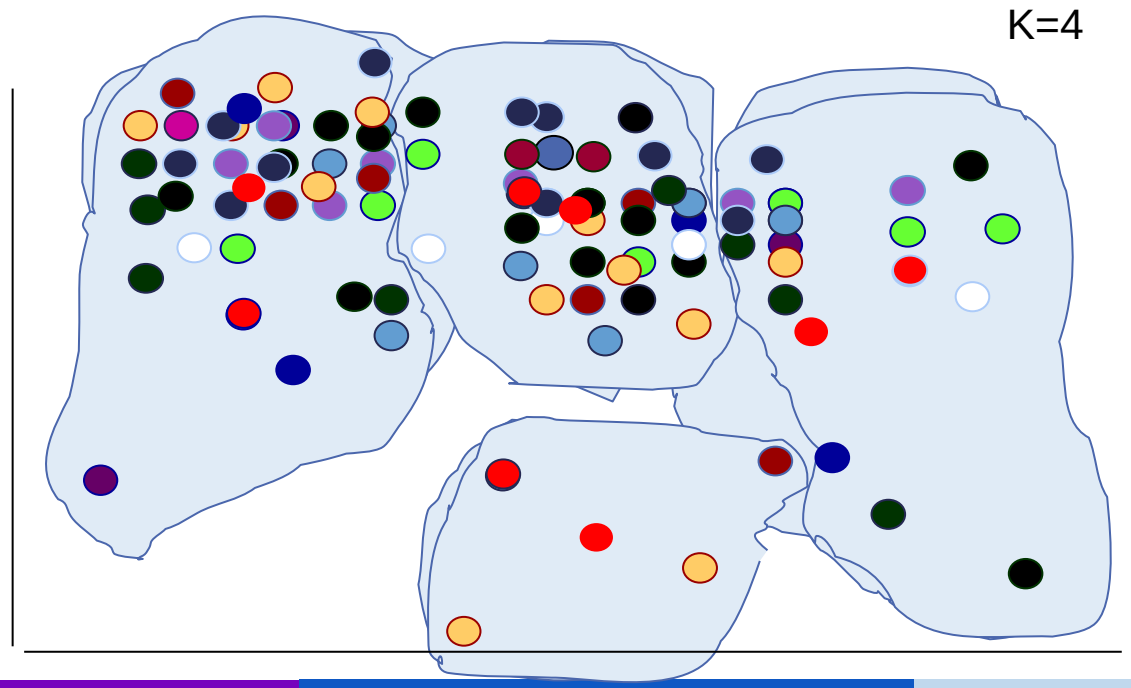
Partition-based Clustering

Partitioning Algorithms: Basic Concept

- Given a *k*, find a partition of *k* clusters that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions.
  - Heuristic methods.
    - k-means: each cluster is represented by the center of the cluster
    - k-medoids or PAM (Partition Around Medoids) : each cluster is represented by one of the objects in the cluster.

K-Means Clustering Algorithm

- Algorithm:
  - Randomly initialize k cluster means
  - Iterate:
    - Assign each object to the nearest cluster mean
    - Recompute cluster means
  - Stop when clustering converges

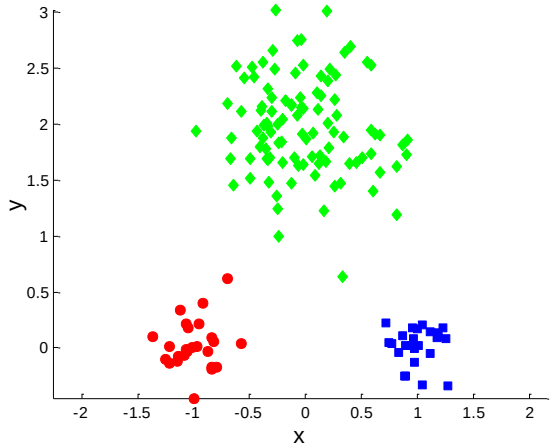


K-means Clustering – Details

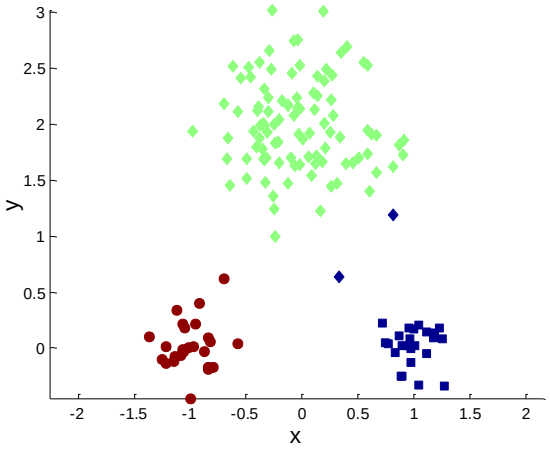
- Clusters produced vary from one run to another
- ‘Closeness’ is measured
  - E.g., by Euclidean distance, cosine similarity, and correlation
- Most of the convergence happens in the first few iterations
  - Stopping condition can be changed to ‘Until few changes’
- Complexity is $O(n \times K \times I \times d)$
  - n = number of points, K = number of clusters,

    I = number of iterations, d = number of attributes

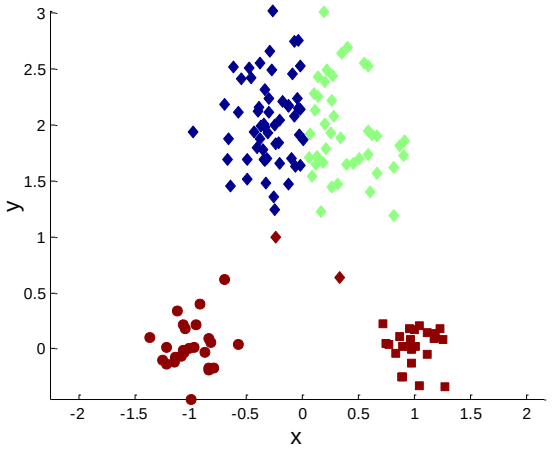
Two different K-means Clusterings



Original Points



Optimal Clustering



Sub-optimal Clustering

Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

    - x is a data point in cluster C_i ; m_i is the representative point for cluster C_i
    - Show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error

➢ One easy way to reduce SSE is to increase K (# clusters)

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- Postprocessing
- Bisecting K-means
  - Not as susceptible to initialization issues

Bisecting K-means

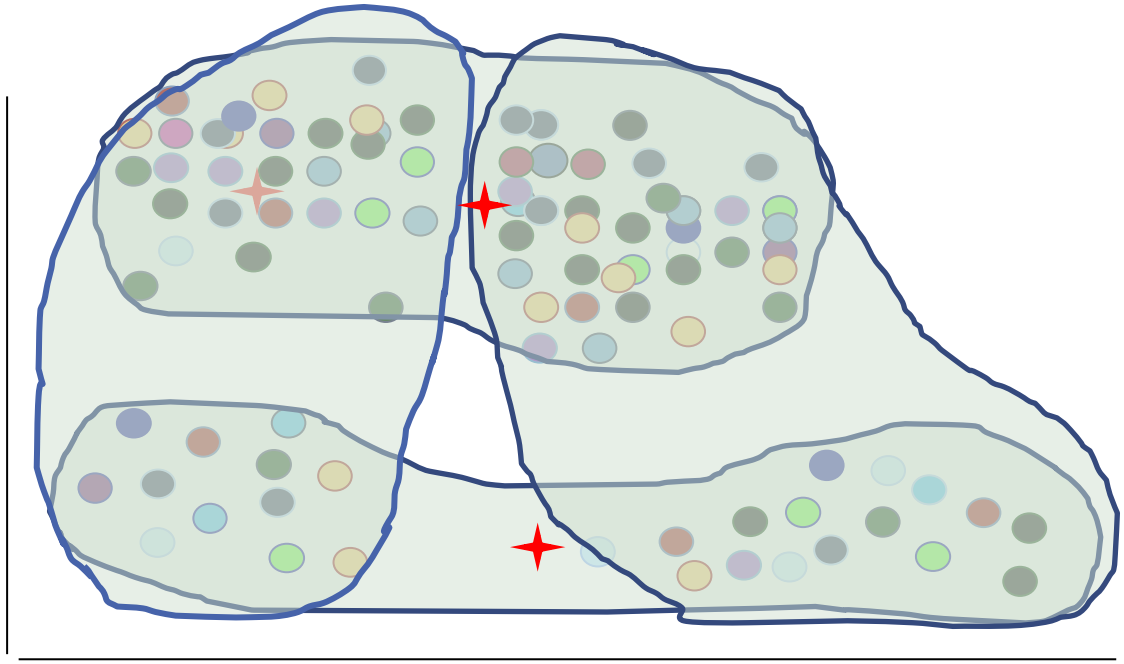
- Bisecting K-means algorithm
  - Variant of K-means that can produce a partitioned or a hierarchical clustering

Algorithm 3 Bisecting K-means Algorithm.

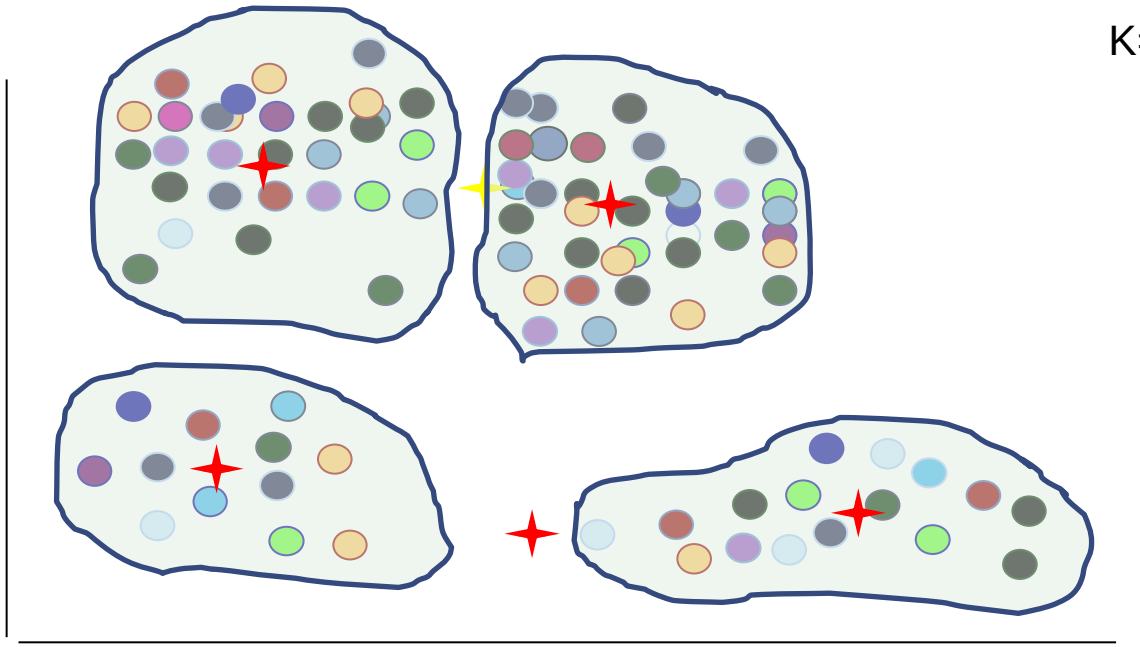
1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains K clusters

Bisecting K-means Example

K=4



Bisecting K-means Example



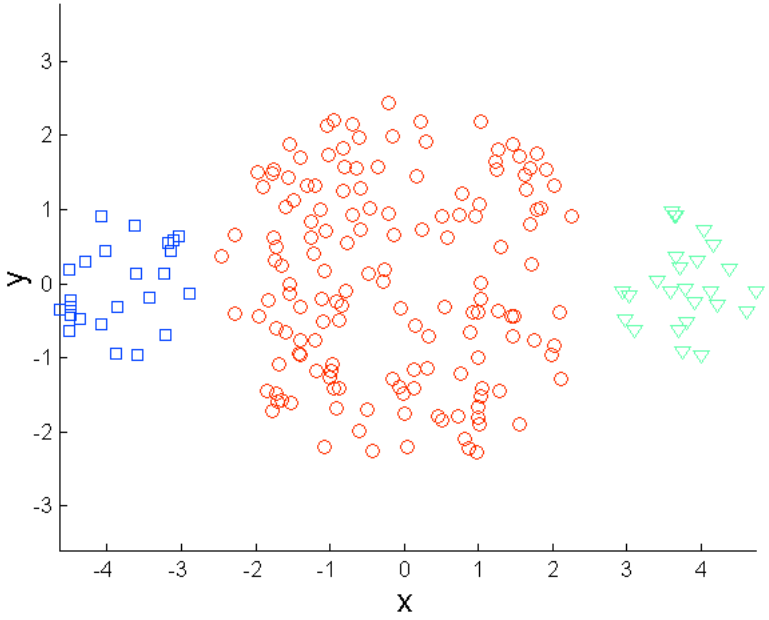
Produce a hierarchical clustering based on the sequence of clusterings

Limitations of K-means

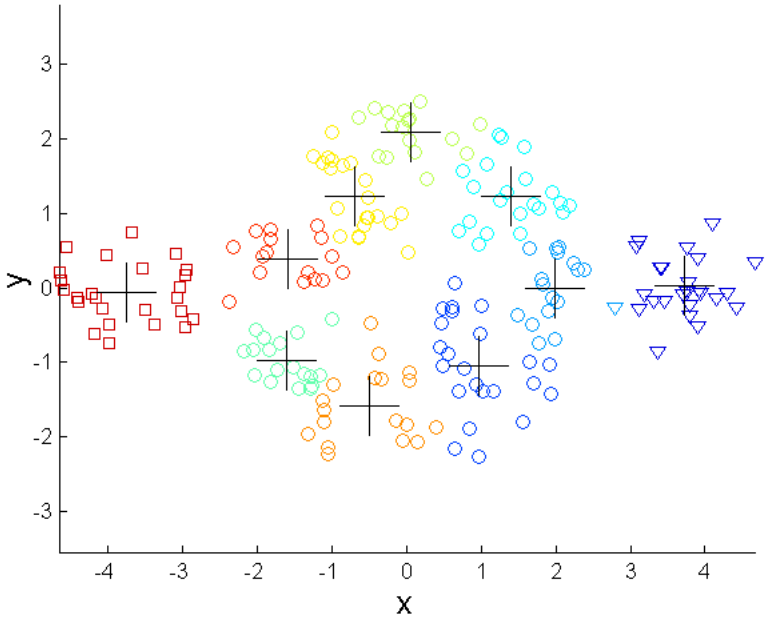
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers

Limitations of K-means: Differing Sizes



Original Points

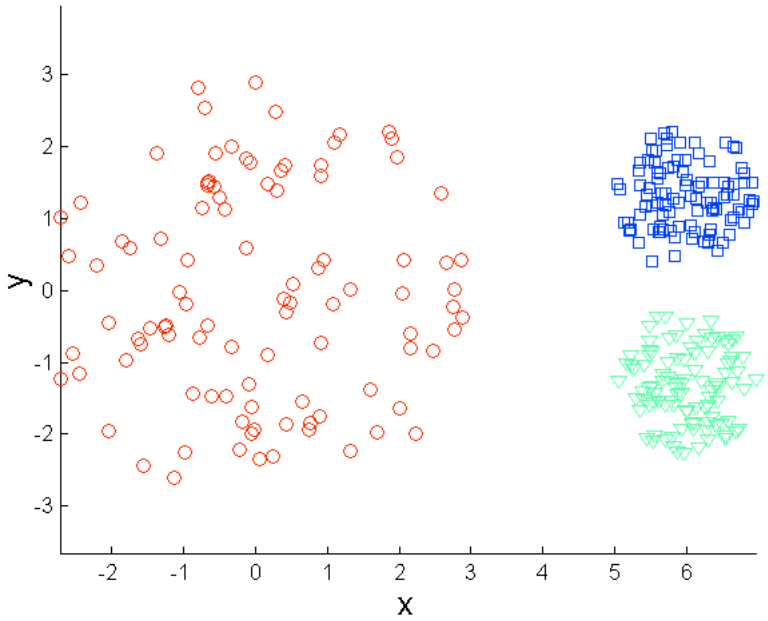


K-means (10 Clusters)

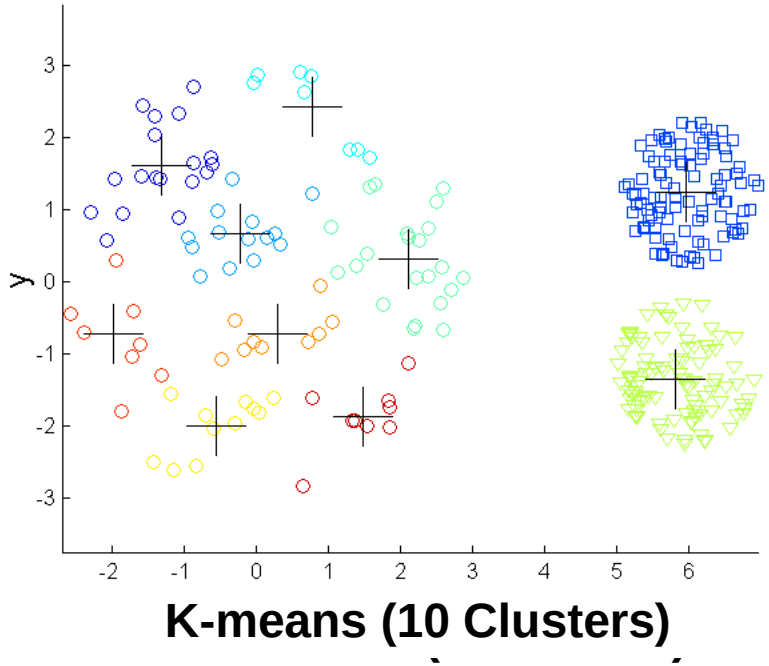
One solution is to use many clusters.

Find parts of clusters, but need to put together

Limitations of K-means: Differing Density



Original Points

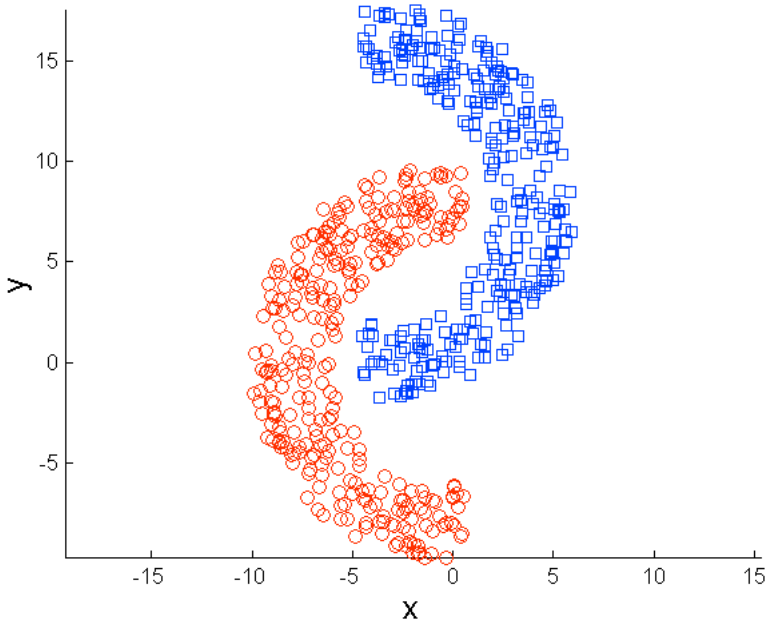


K-means (10 Clusters)

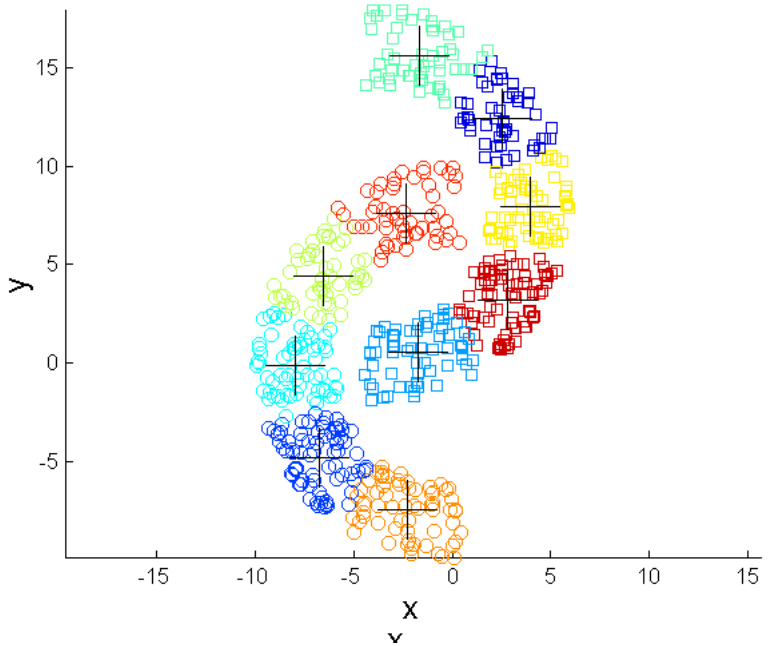
One solution is to use many clusters.

Find parts of clusters, but need to put together

Limitations of K-means: Non-globular Shapes



Original Points



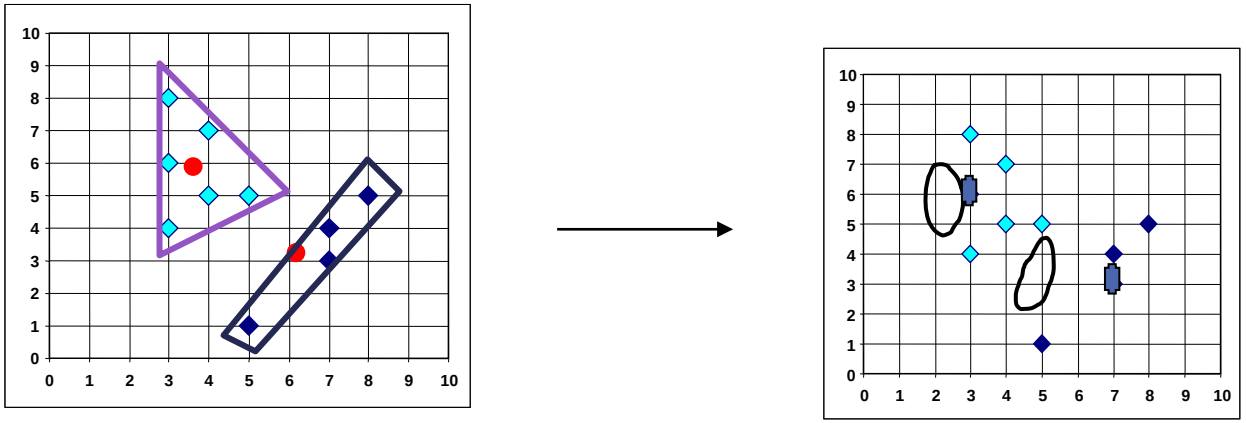
K-means (10 Clusters)

One solution is to use many clusters.

Find parts of clusters, but need to put together

K-Means vs. K-Medoids

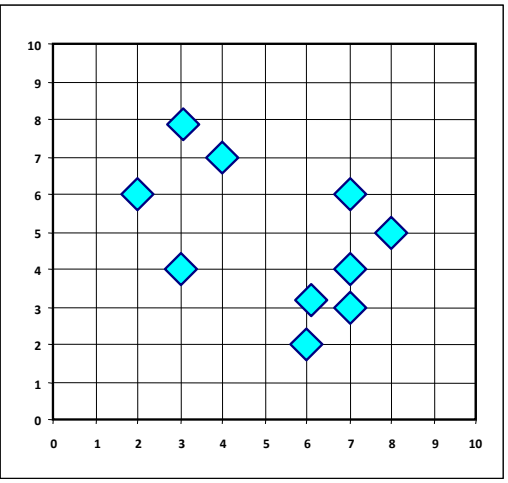
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most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



The *K-Medoids* Clustering Method

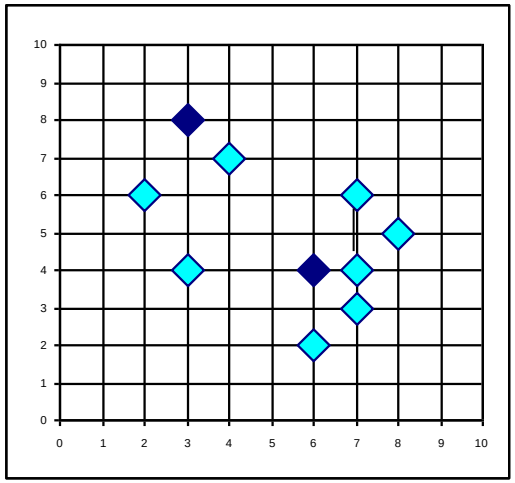
- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - Approach:
    - Starts from an initial set of medoids
    - Iteratively replaces one of the medoids by one of the non-medoids
    - If it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

Typical K-medoids algorithm (PAM)

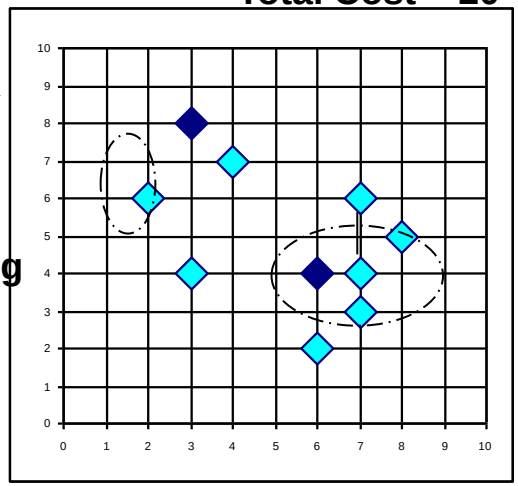


K=2

Arbitrary choose k object as initial medoids

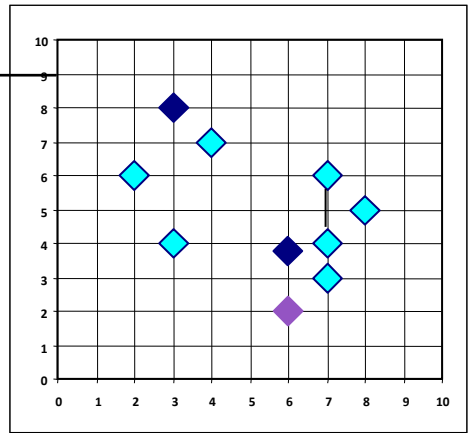


Assign each remaining object to nearest medoids

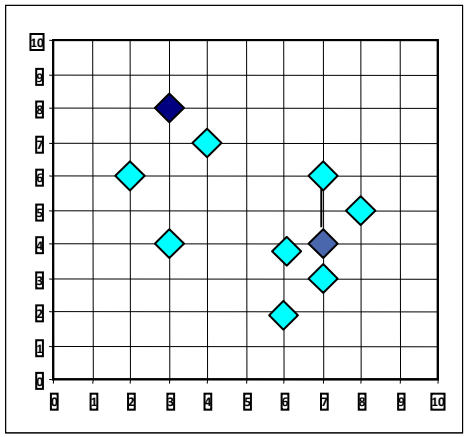


Total Cost = 20

Randomly select a nonmedoid object,O_{ramdom}



Compute total cost of swapping



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

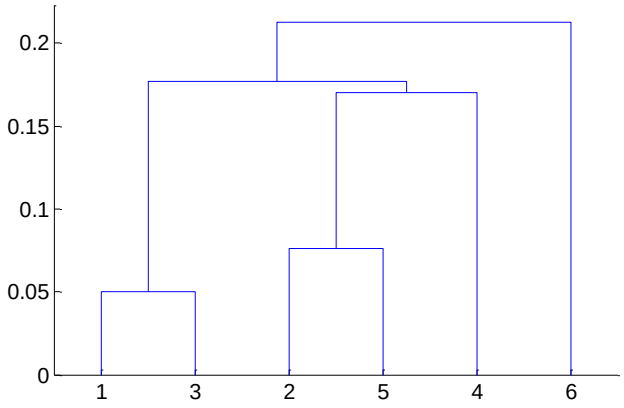
CLARA (Clustering Large Applications)

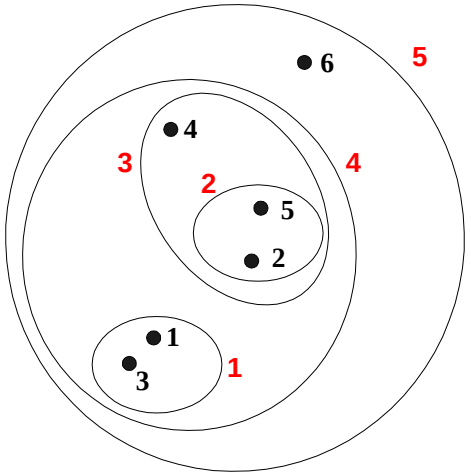
- Partition for large database
- Steps
  - Draws *multiple samples* of the data set
  - Applies *PAM* on each sample
  - Gives the best clustering as the output
- Strength
  - Deal with larger data sets than *PAM*
- Weakness
  - Efficiency depends on the sample size
  - Good clustering on samples => good clustering on whole data set ?

Hierarchical Clustering

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Approaches of Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - 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:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

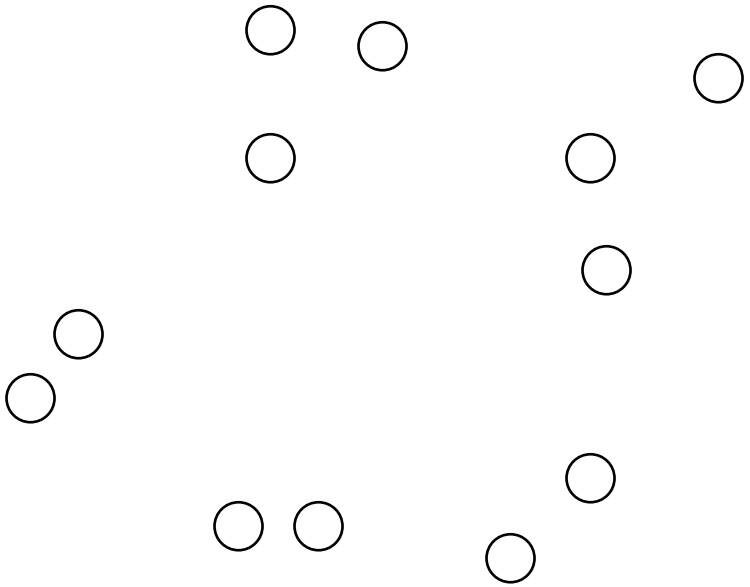
Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. Repeat
     1. Merge the two closest clusters
     2. Update the proximity matrix
  4. Until only a single cluster remains

➢ Key operation: the computation of the proximity of two clusters

Starting Situation

- Start with clusters of individual points and a proximity matrix

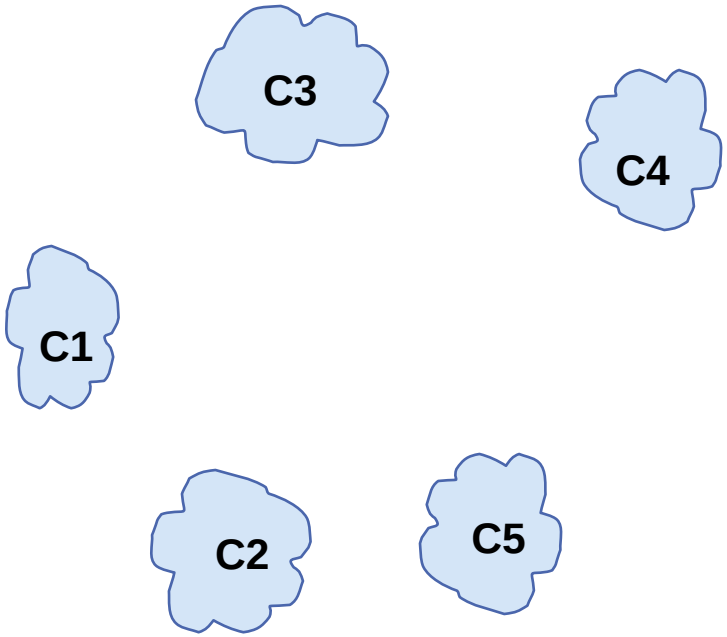


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

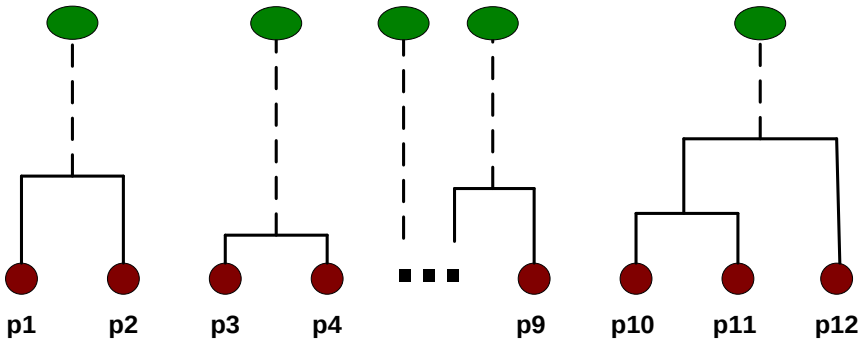
Intermediate Situation

- After some merging steps, we have some clusters



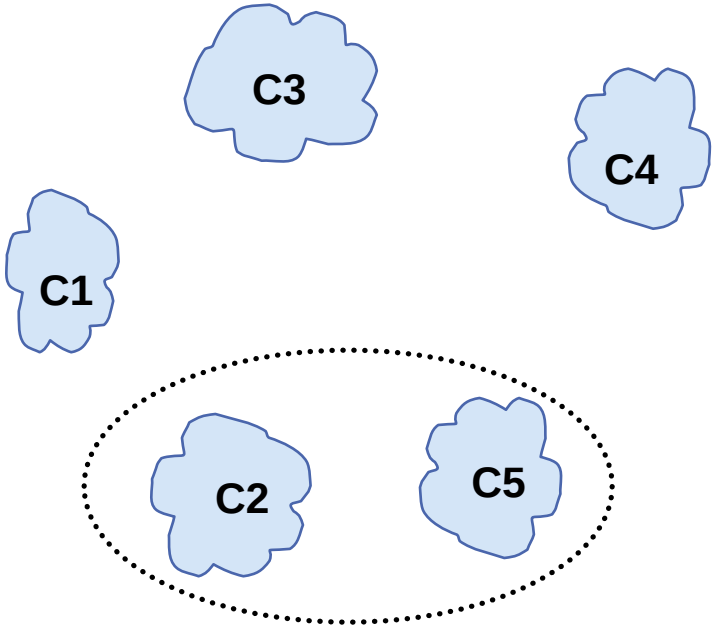
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



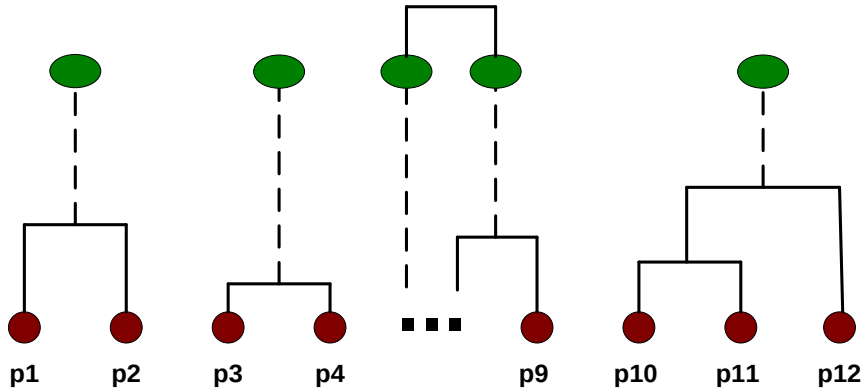
Intermediate Situation (Contd.)

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix



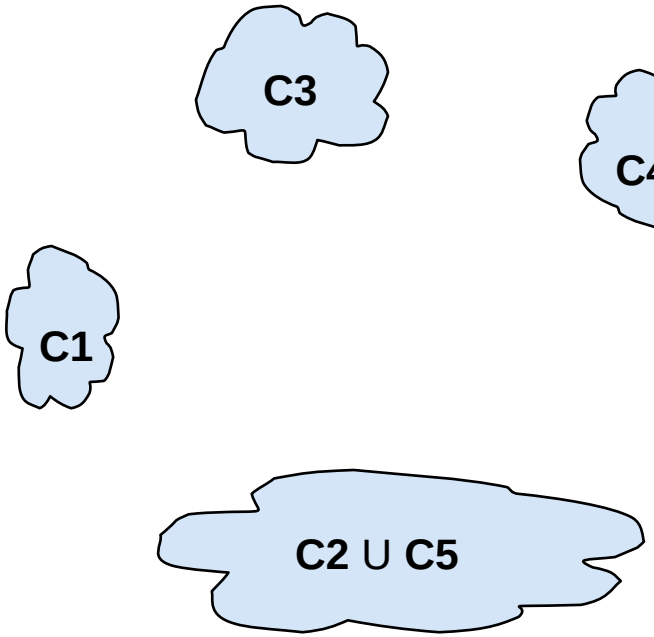
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



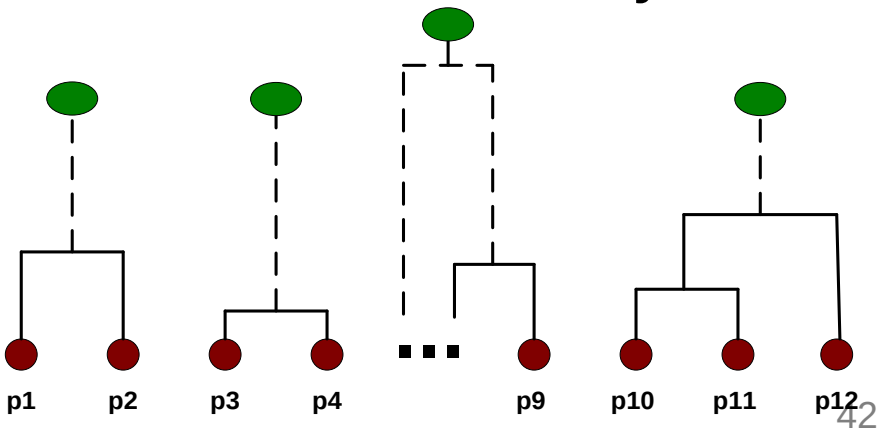
After Merging

- The question is “How do we update the proximity matrix?”

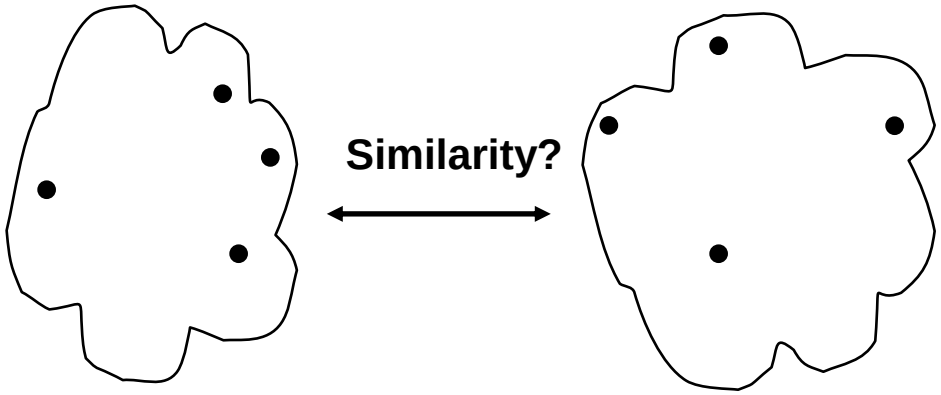


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity



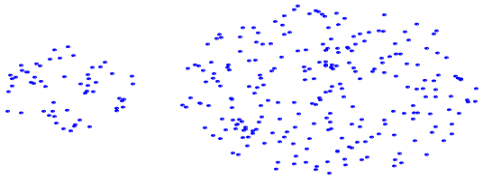
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

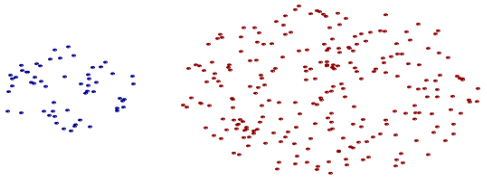
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

MIN

- Strength:
  - Can handle non-elliptical shapes

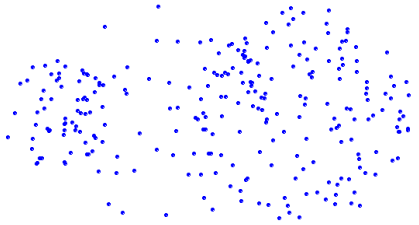


Original Points

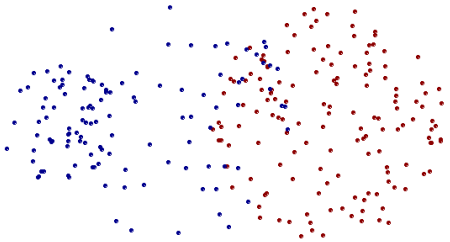


Two Clusters

- Limitations:
  - Sensitive to noise and outliers



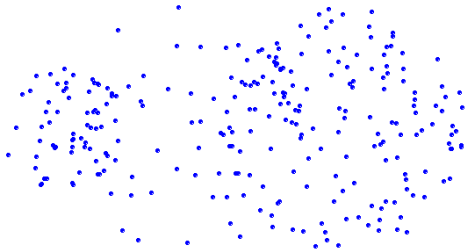
Original Points



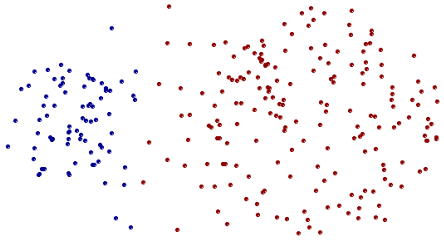
Two Clusters

MAX

- Strength:
  - Less susceptible to noise and outliers

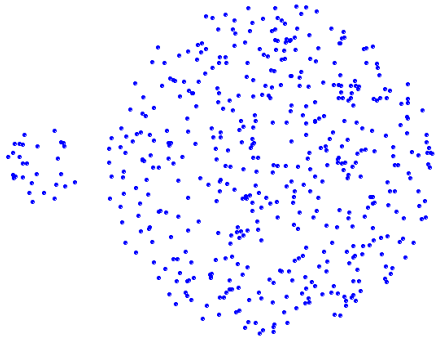


Original Points

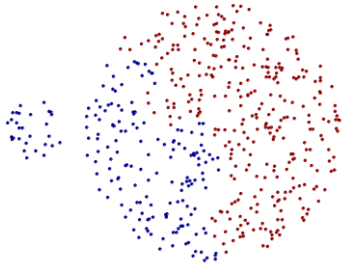


Two Clusters

- Limitations:
  - Tends to break large clusters
  - Biased towards globular clusters

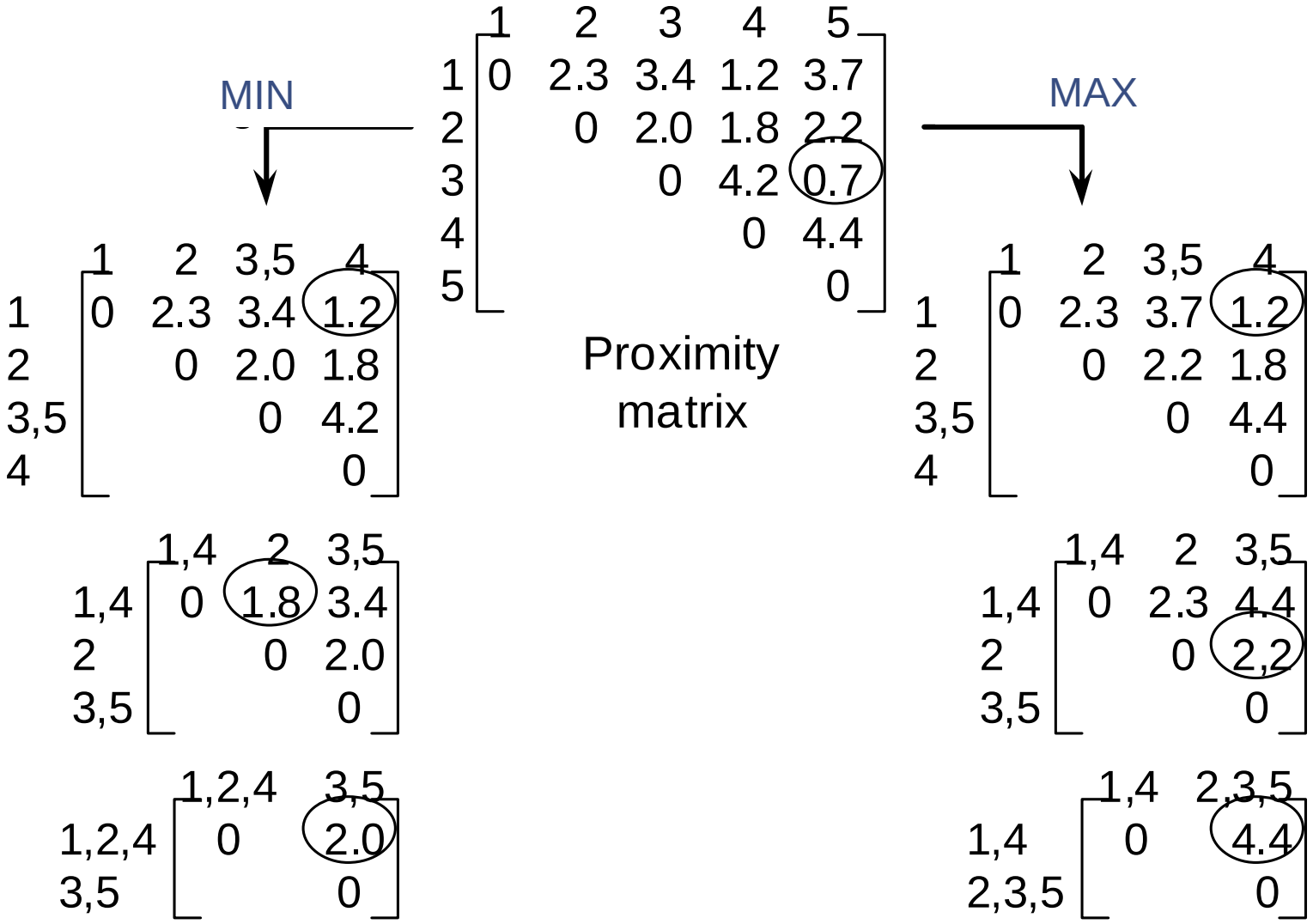


Original Points



Two Clusters

Proximity Matrix Examples



Group Average

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters

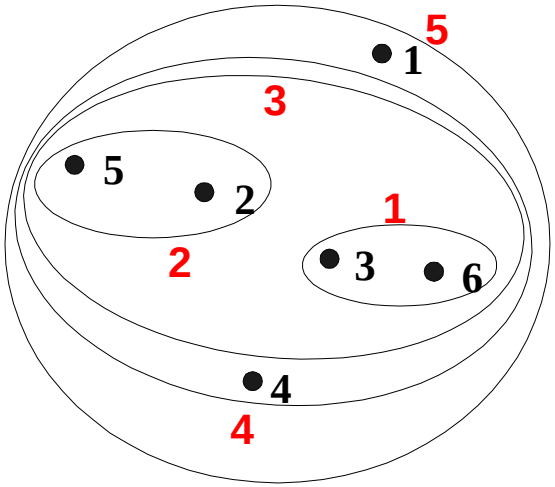
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Compromise between MIN and MAX
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

Ward's Method

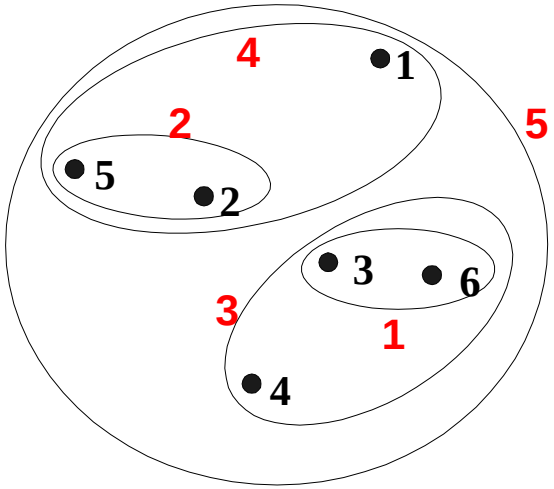
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters

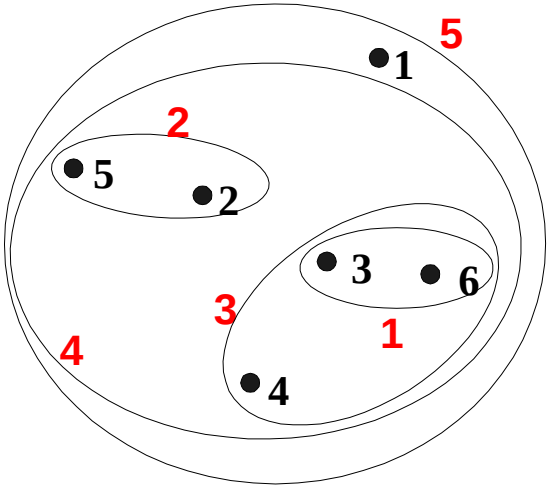
Hierarchical Clustering: Comparison



MIN

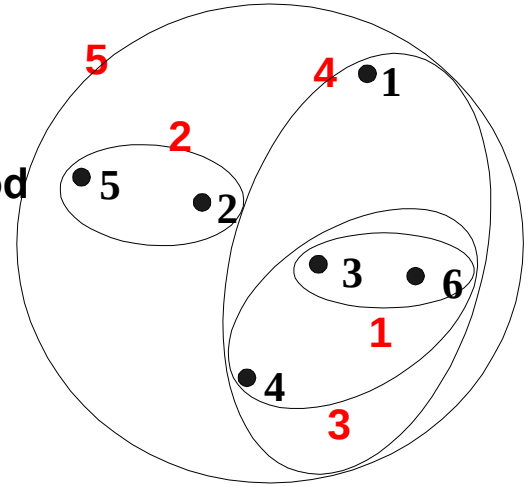
MAX





Group Average

Ward's Method

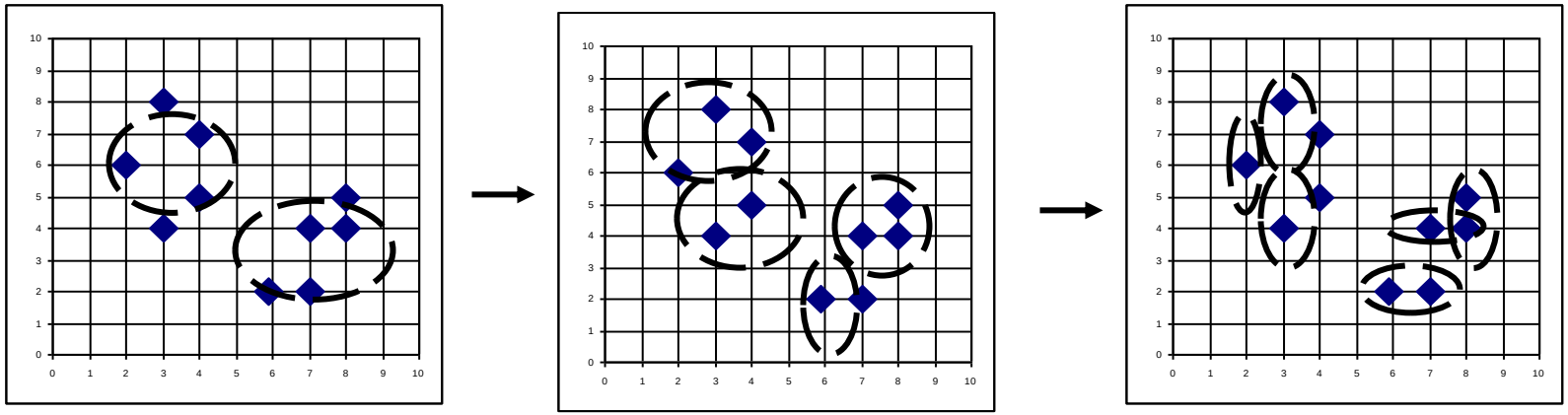


Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

1: Compute a minimum spanning tree for the proximity graph.
2: **repeat**
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: **until** Only singleton clusters remain



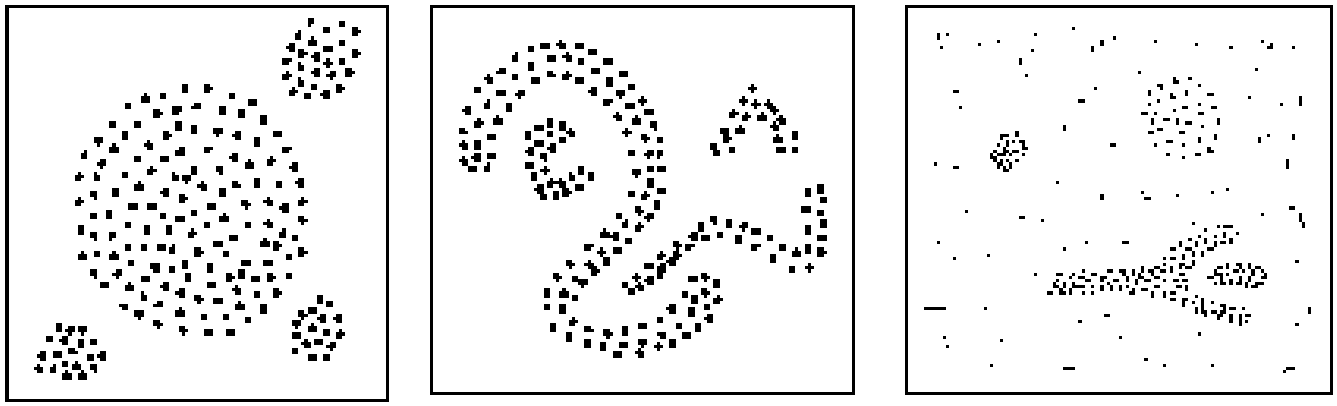
Time and Space Requirements

- $O(N^2)$ space since it uses the proximity matrix.
  - N is the number of points.
- $O(N^3)$ time in many cases
  - There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches
- Once a decision is made to combine two clusters, it cannot be undone

Density-Based Clustering

Density-Based Clustering

- Clustering based on density (local cluster criterion), such as density-connected points
- Each cluster has a considerable higher density of points than outside of the cluster



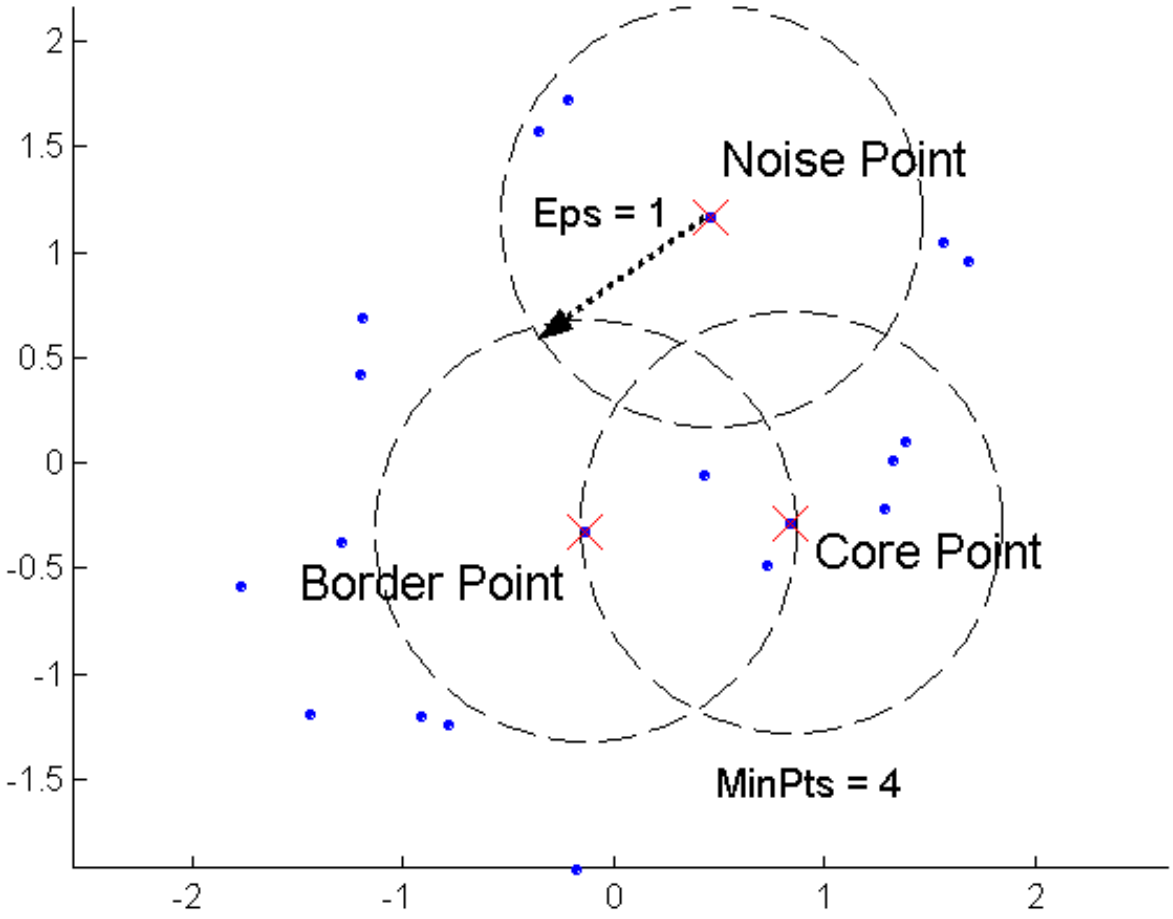
Density-Based Clustering Methods

- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Approaches
  - DBSCAN (KDD’96)
  - OPTICS (SIGMOD’99).
  - DENCLUE (KDD’98)
  - CLIQUE (SIGMOD’98)

DBSCAN

- Density = number of points within a specified radius (Eps)
- A point is a core point if it has more than a specified number of points (MinPts) within Eps
  - These are points that are at the interior of a cluster
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
- A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points



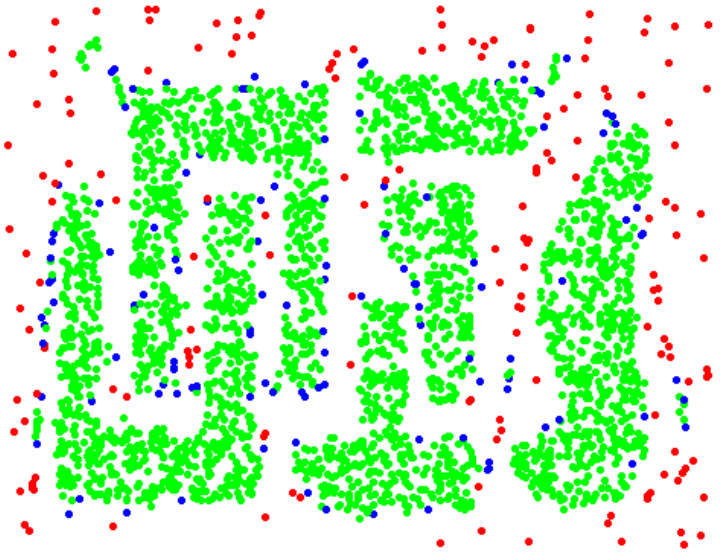
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
end for
```

DBSCAN Examples

Original Points

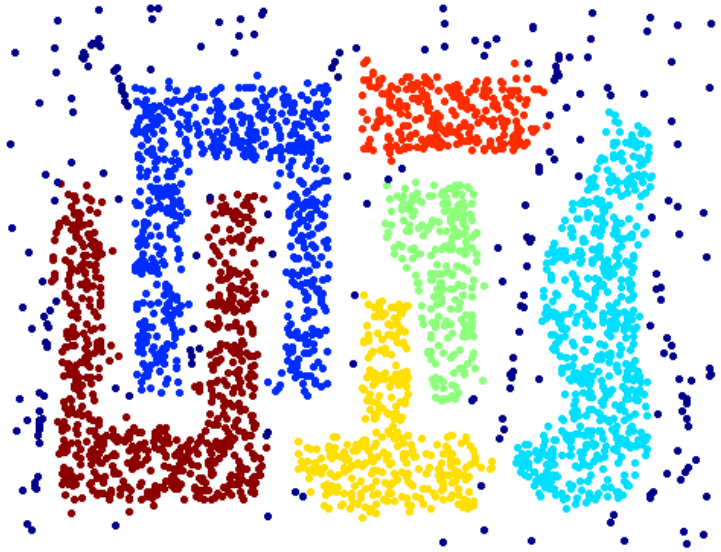


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

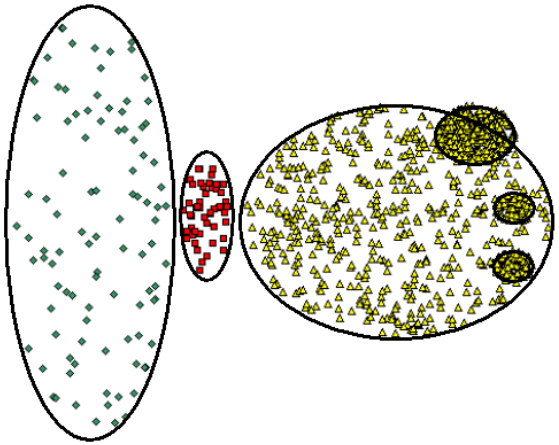
Original Points



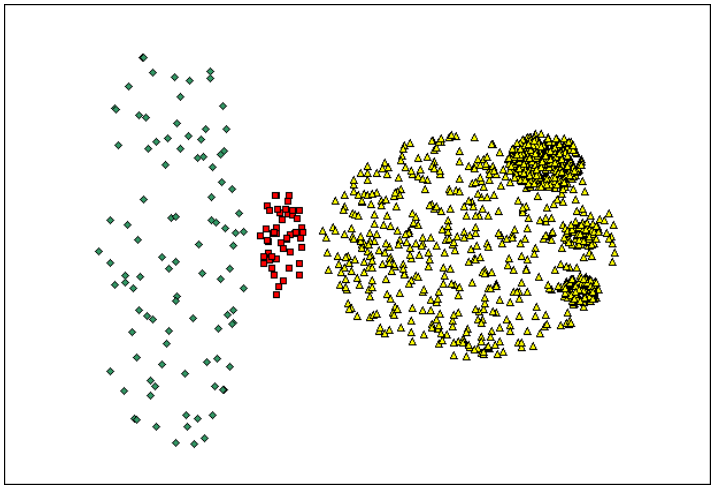
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

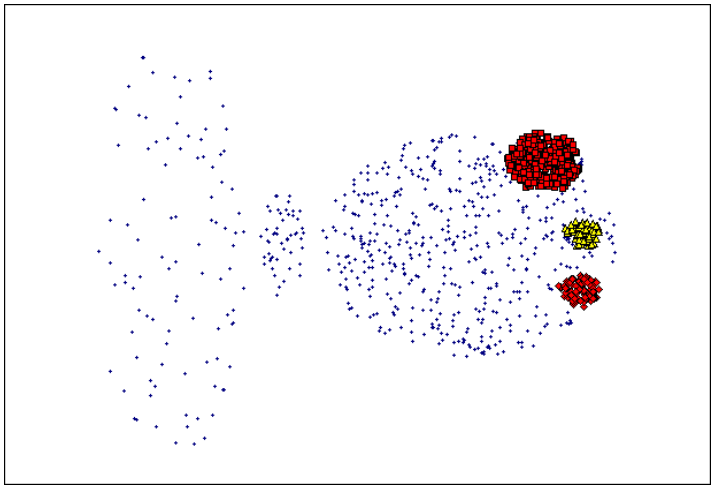
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).

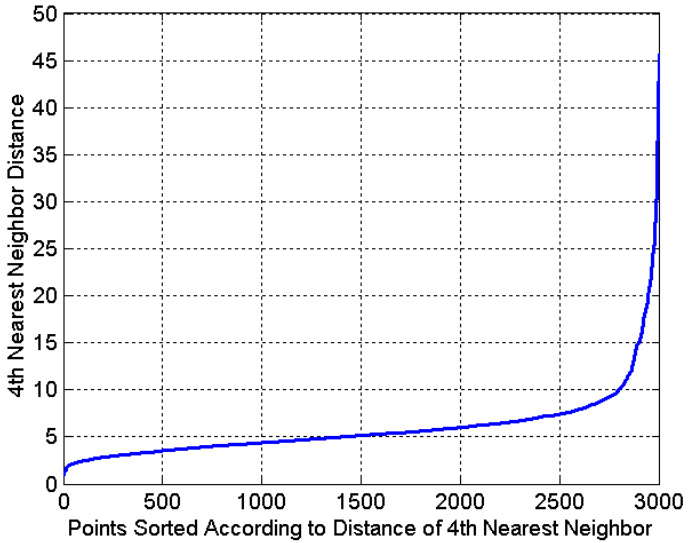


(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data

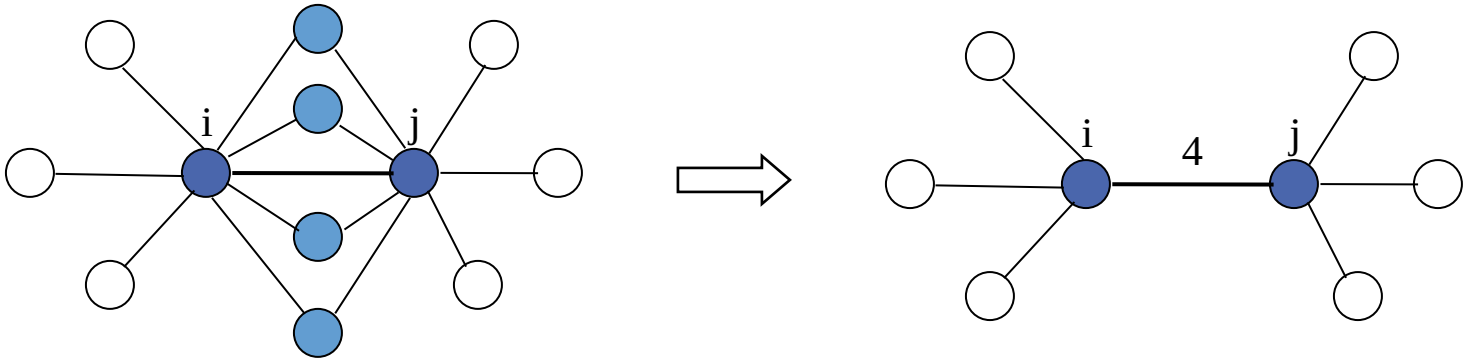
Determining EPS and MinPts

- Idea is that 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
- So, plot sorted distance of every point to its k^{th} nearest neighbor



Shared Nearest Neighbor Approach

- SNN graph: the weight of an edge is the number of shared neighbors between vertices given that the vertices are connected



SNN Clustering Algorithm

1. **Compute the similarity matrix**
   This corresponds to a similarity graph with data points for nodes and edges whose weights are the similarities between data points

2. **Sparsify the similarity matrix by keeping only the *k* most similar neighbors**
   This corresponds to only keeping the *k* strongest links of the similarity graph

3. **Construct the shared nearest neighbor graph from the sparsified similarity matrix.**
   At this point, we could apply a similarity threshold and find the connected components to obtain the clusters (Jarvis-Patrick algorithm)

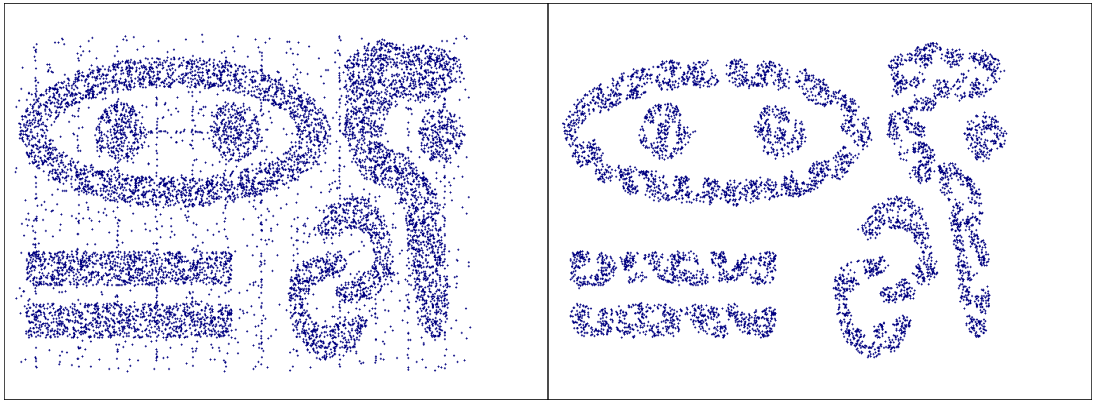
4. **Find the SNN density of each Point.**
   Using a user specified parameters, *Eps*, find the number points that have an SNN similarity of *Eps* or greater to each point. This is the SNN density of the point

SNN Clustering Algorithm (Contd.)

5. **Find the core points**
   Using a user specified parameter, *MinPts*, find the core points, i.e., all points that have an SNN density greater than *MinPts*
6. **Form clusters from the core points**
   If two core points are within a radius, *Eps*, of each other they are place in the same cluster
7. **Discard all noise points**
   All non-core points that are not within a radius of *Eps* of a core point are discarded
8. **Assign all non-noise, non-core points to clusters**
   This can be done by assigning such points to the nearest core point

(Note that steps 4-8 are DBSCAN)

SNN Density



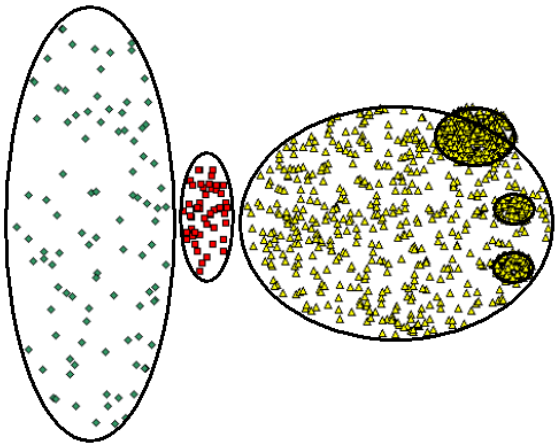
a) All Points

b) High SNN Density

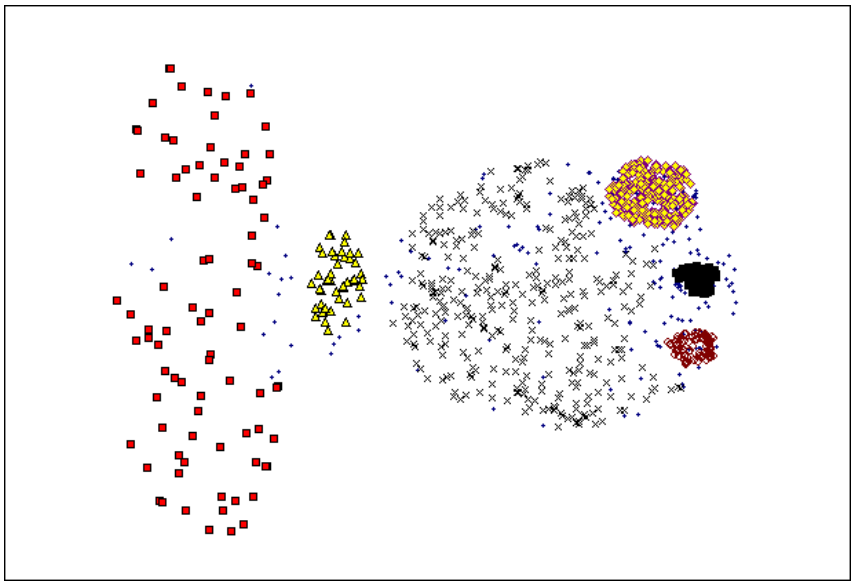
c) Medium SNN Density

d) Low SNN Density

SNN Density



Original Points



SNN Clustering

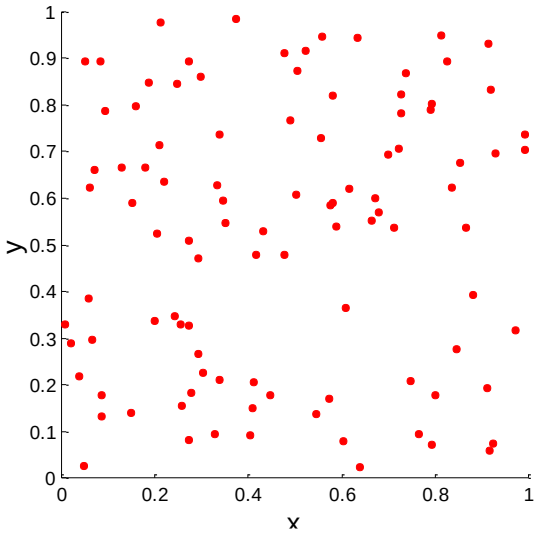
Cluster Validity

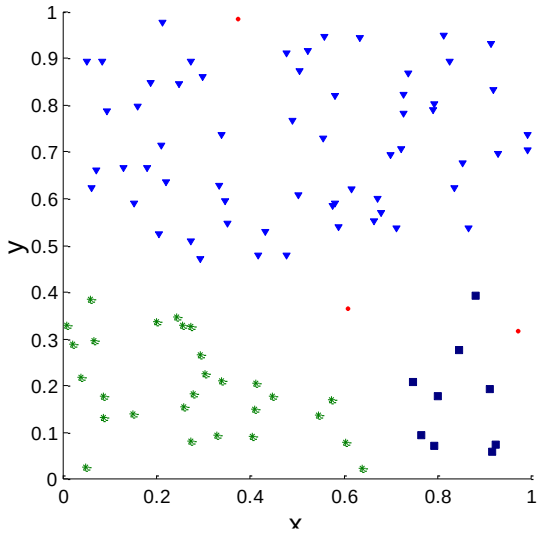
Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?
- But "clusters are in the eye of the beholder"!
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Clusters found in Random Data

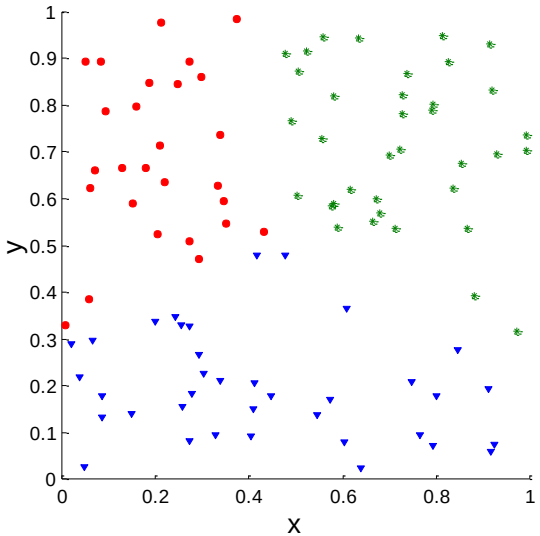
Random Points

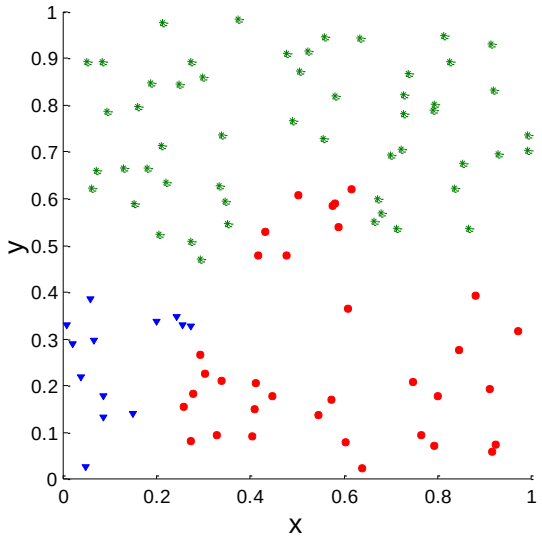




DBSCAN

K-means





Complete Link

Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data
   - i.e., distinguishing whether non-random structure actually exists in the data
2. Comparing the results of a cluster analysis to externally known results e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information. - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

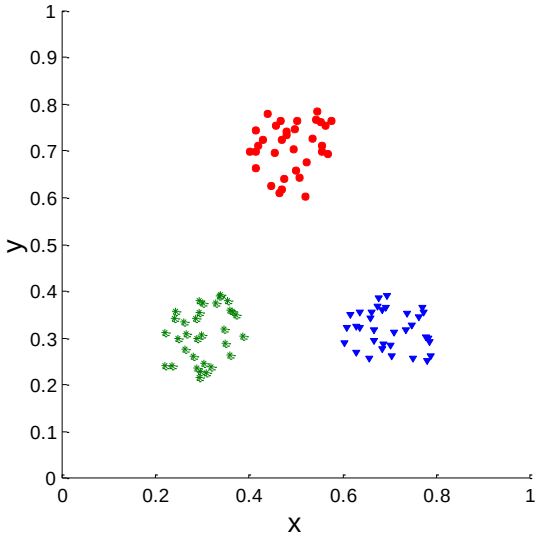
- Numerical measures applied to judge various aspects of cluster validity, are classified into the following three types
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels, e.g., Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information, e.g., Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusters
- Sometimes are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measuring Cluster Validity Via Correlation

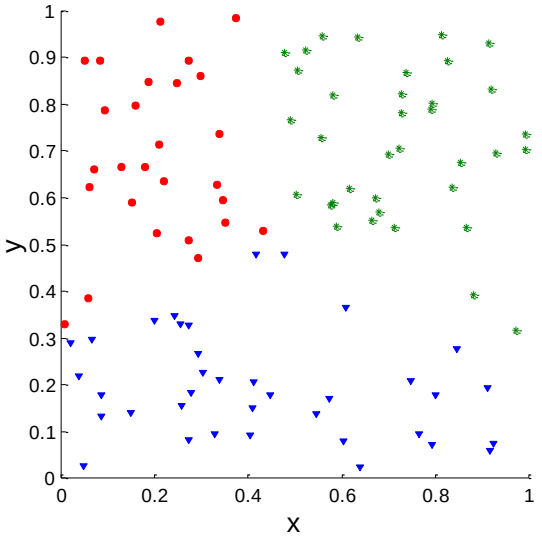
- Two matrices
  - Proximity Matrix
  - “Incidence” Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clustering of the following two data sets



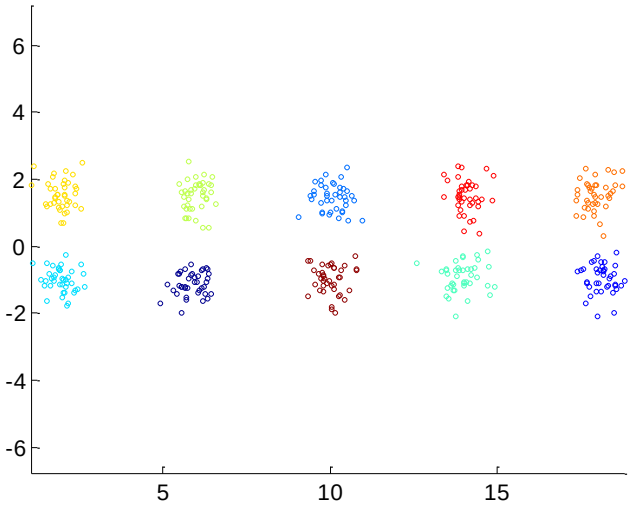
Corr = -0.9235

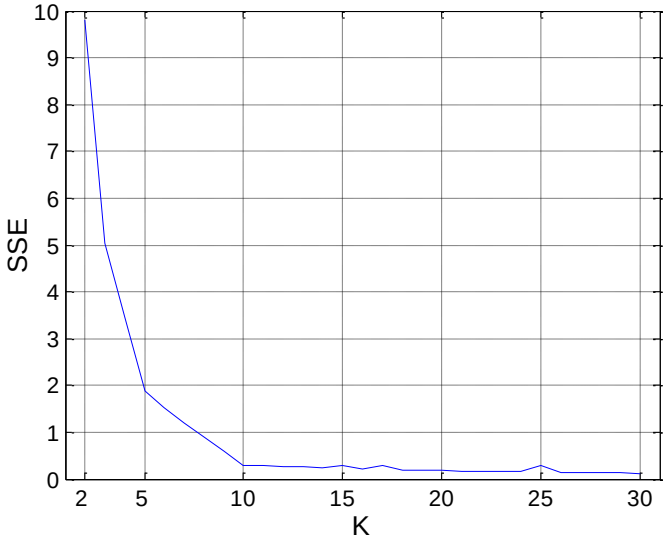


Corr = -0.5810

Internal Measures: SSE

- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
- SSE is good for comparing two clusters
- Can also be used to estimate the number of clusters





Internal Measures: Cohesion and Separation

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

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

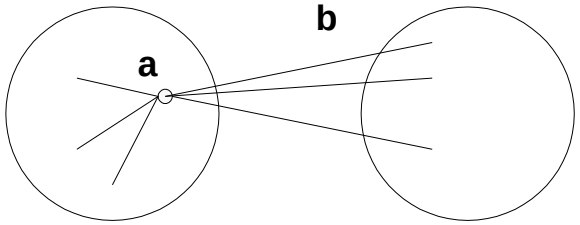
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

    - Where $|C_i|$ is the size of cluster i

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by
    s = 1 – a/b if a < b, (or s = b/a - 1 if a ≥ b, not the usual case)



  - Typically between 0 and 1.
  - The closer to 1 the better.
- Can calculate the Average Silhouette width for a cluster or a clustering

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes

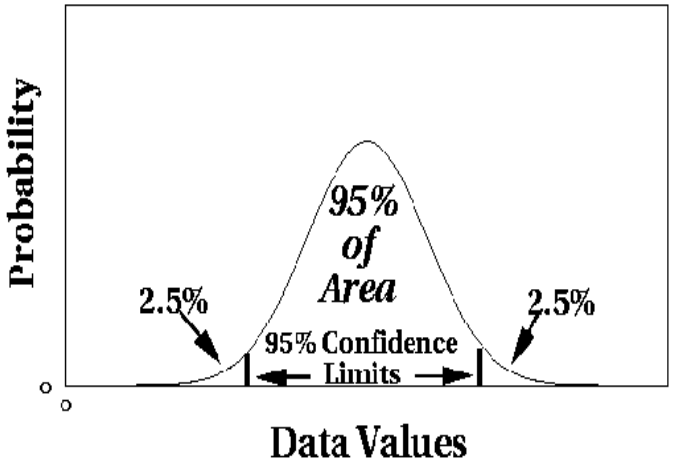
Outlier Analysis

Outlier Analysis

- Outliers
  - set of objects dissimilar from the remainder of the data
  - e.g. Sports: Michael Jordon
- Problem: given data points, find top n outlier
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Medical analysis
- Approaches
  - statistics-based
  - distance-based

Statistical Approaches

- Assumption
  - A model underlying distribution that generates data set
  - E.g. normal distribution
- Use discordancy tests depending on
  - Data distribution
  - Distribution parameter (e.g., mean, variance)
  - Number of expected outliers
- Drawbacks
  - Most tests are for single attribute
  - Unknown data distribution



Grubbs' Test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
  - H_0: There is no outlier in data
  - H_A: There is at least one outlier
- Grubbs' test statistic:

$$G = \frac{\max\left|X - \overline{X}\right|}{s}$$

- Reject H_0 if:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N,N-2)}}{N-2+t^2_{(\alpha/N,N-2)}}}$$

Likelihood Approach

- Data distribution, D = (1 – λ) M + λ A
- M is a probability distribution estimated from data
  - Can be based on any modeling method (naïve Bayes, maximum entropy, etc)
- A is initially assumed to be uniform distribution
- Likelihood at time t:

$$L_t(D) = \prod_{i=1}^{N} P_D(x_i) = \left((1-\lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right) \left(\lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right)$$

$$LL_t(D) = |M_t| \log(1-\lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

Limitations of Statistical Approaches

- Most of the tests are for a single attribute
- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution

Distance-Based Outliers

- Data is represented as a vector of features
- Three major approaches
  - Nearest-neighbor based
  - Density based
  - Clustering based

Limitations of KNN Approaches

- Proximity-based approaches typically take $O(m^2)$ time
- The approach is sensitive to the choice of parameters
- It cannot handle data sets with region of widely differing densities
  - Because it use global thresholds

Limitations of Density-based Approaches

- These approaches typically take $O(m^2)$ time
- The approach is sensitive to the choice of parameters
- It is difficult to select the values of upper and lower bounds

Limitations of Clustering-based Approaches

- The set of outliers can be heavily dependent upon the number of clusters
- The quality of outliers is heavily impacted by the quality of clusters
  - Each clustering algorithm is suitable only for a certain type of data
  - The clustering algorithm needs to be chosen carefully