Cluster Analysis: Concepts and Algorithms (Unsupervised Learning)

XX-161-A-21 – Big Data Analytics

Dr. Jim Scrofani
jwscrofa@nps.edu

Topics at a Glance

- Introduction/Motivation
- Taxonomy of Cluster Analysis
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Validation

Approaches to Learning

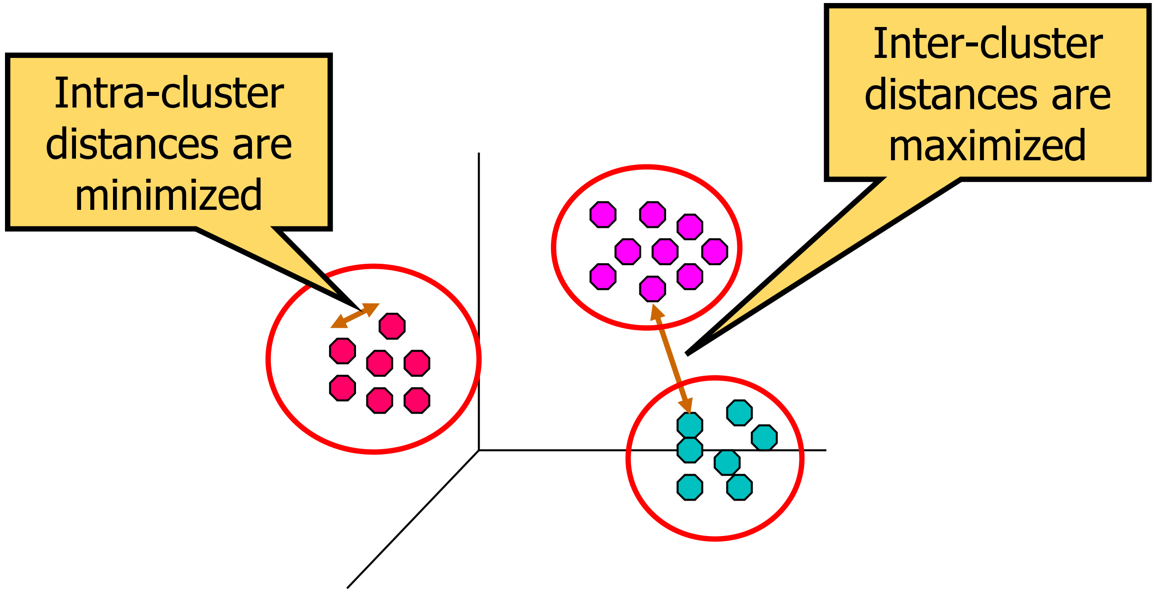
- Supervised Learning
  - Used to estimate an unknown (input, output) mapping from known (input, output) samples
  - "Supervised" – output values for training samples are known, i.e., provided by a teacher
  - Common approaches
    - Classification
    - Regression

- Unsupervised Learning
  - Used to estimate an unknown (input, output) mapping from input samples only
  - There is no teacher
  - Common approaches
    - Distribution estimation
    - Discover structure in the data ("clusters")

No teacher, no labels, no target

Cluster Analysis

- *Cluster analysis* or *data segmentation* -- Grouping or segmenting a collection of objects into subsets or ***clusters*** such that those within each cluster are more related (more similar) to one another than objects assigned to different clusters



Purpose of Cluster Analysis

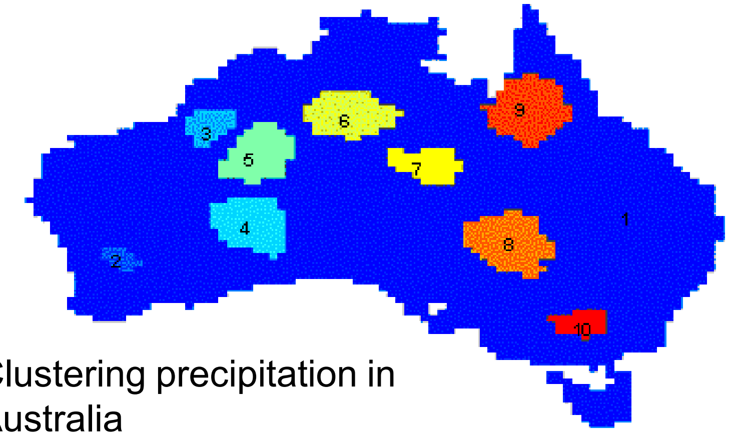
- Understanding
  - Classes or conceptually meaningful groups of objects share common characteristics
  - Play an important role in how people analyze and describe the world

- Utility
  - Abstraction from individual data objects to clusters in which those objects reside
    - Summarization
    - Compression
    - Efficiently finding Nearest Neighbors

Purpose of Cluster Analysis

- Understanding
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations
- Utility
  - Reduce the size of large data sets

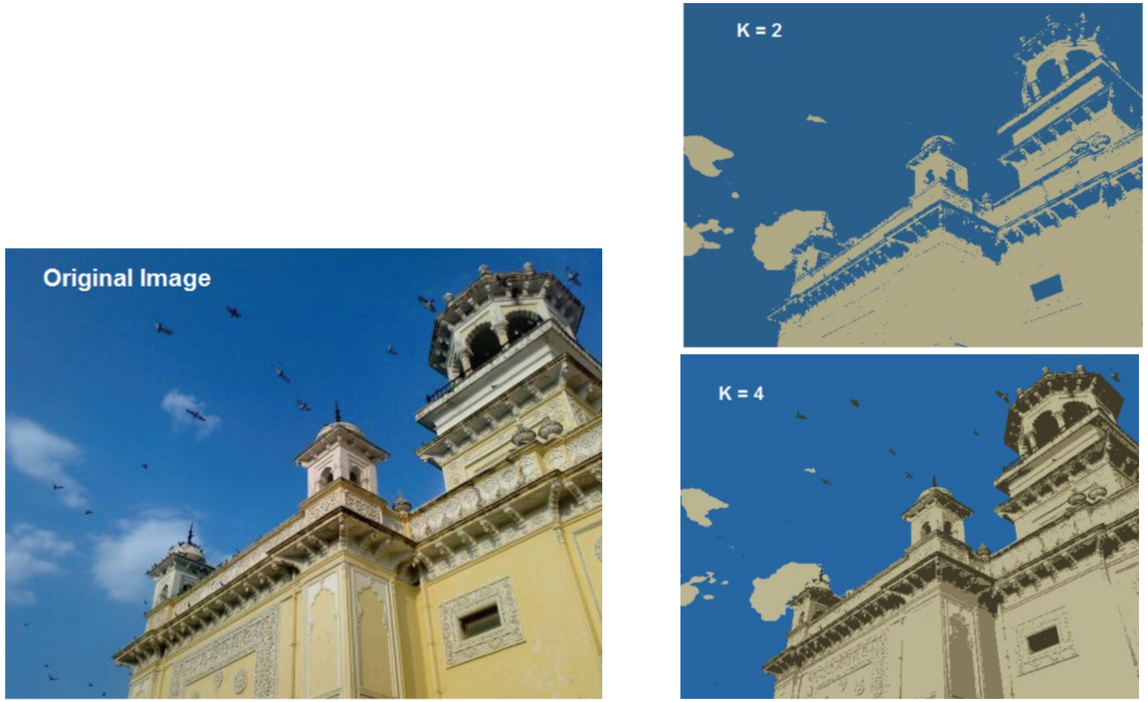
	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP



Clustering precipitation in Australia

Application Examples

- A stand-alone tool: exploratory data analysis
- A preprocessing step for other algorithms
- Pattern recognition, spatial-temporal data analysis, image processing
- Text analytics
- Anomaly detection



Application Examples

- Keystroke dynamics used to authenticate users

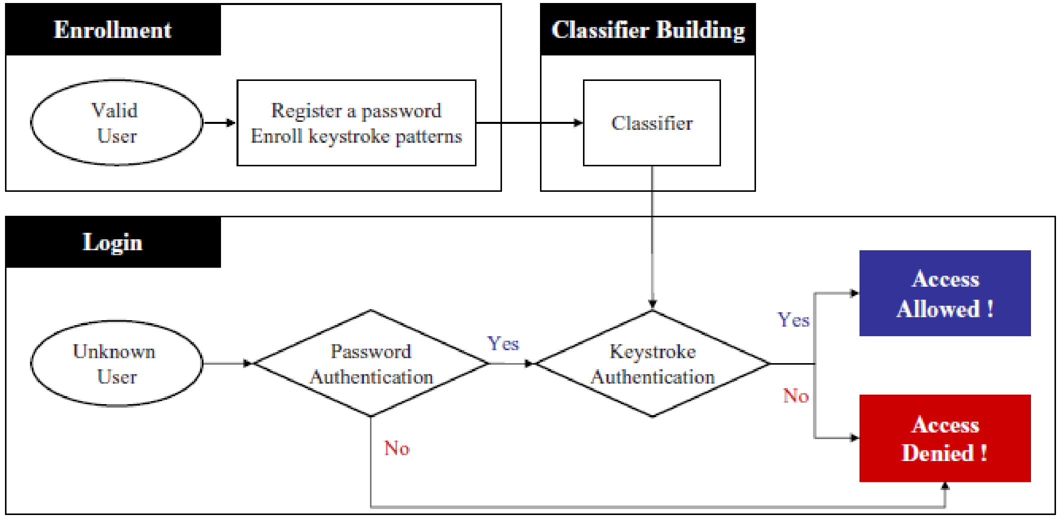


Fig. 1. Keystroke dynamics based authentication process

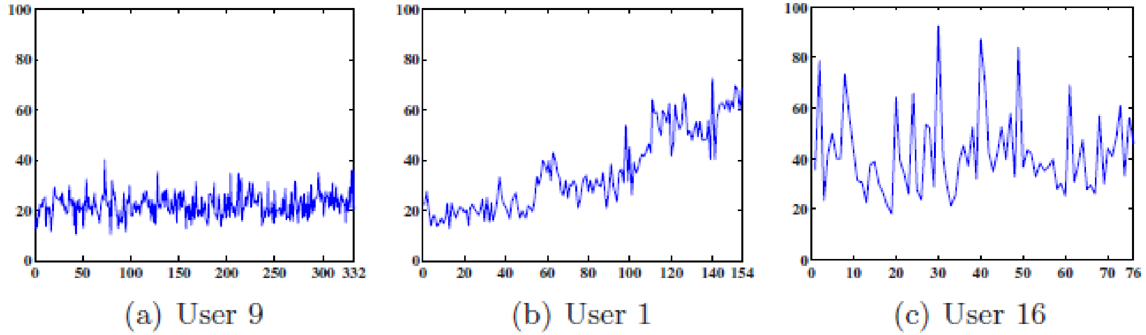
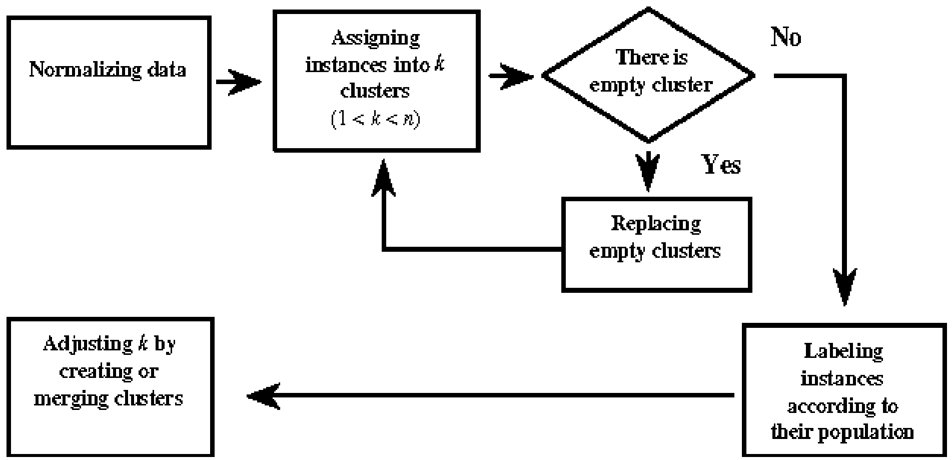


Fig. 2. Distance between login pattern and mean enroll pattern

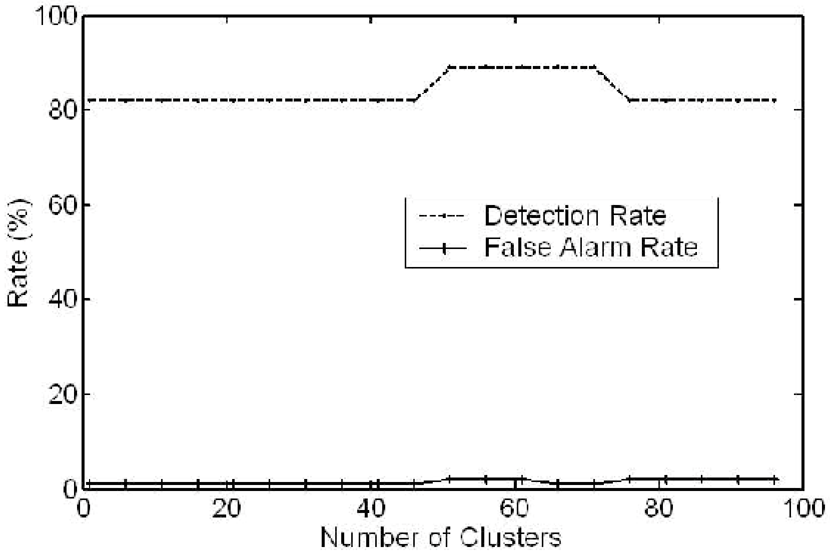
Kang, Pilsung, Seongseob Hwang and Sungzoon Cho. "Continual Retraining of Keystroke Dynamics Based Authenticator." *ICB* (2007).

Application Examples

- Detection of intrusion using cluster analysis



Y-means Algorithm



Y-means with different initial number of clusters

Guan, Ghorbani, Belacel. “Y-Means: A Clustering Method for Intrusion Detection.” *CCECE* (2003).

<u>What is NOT Cluster Analysis?</u>

- Supervised Learning/Classification
  - Requires class labels
- Simple Segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of external specification

- Clustering uses <u>only the data</u>!
  - Unlabeled data

Similarity

- How do we measure similarity/proximity/dissimilarity/distance between objects/data?
- Refer back to lecture on Data Concepts
- Examples
  - Minkowski distance: Manhattan distance, Euclidean distance, etc.
  - Jaccard index for binary data
  - Gower's distance for mixed data (ratio/interval and nominal)
  - Correlation coefficient as similarity between variables

Dissimilarities Based on Attributes

Given measurements x_i and x_j on p attributes $f = 1, 2, \ldots, p$.

$$D(x_i, x_j) = \sum_{f=1}^{p} d^{(f)}(x_i, x_j),$$

is the dissimilarity between object i and j.

Dissimilarities Based on Attributes

- Quantitative variables

$$d(x_i, x_j) = g(x_i, x_j), \qquad \text{e.g., } g(x_i, x_j) = (x_i - x_j)^2,\ |x_i - x_j|, \text{etc.}$$

- Ordinal variables

replace M original values with $\frac{k - 1/2}{M}, k = 1, 2, \ldots, M$, then use quantitative techniques.

- Nominal variables

$$d(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \neq x_j \\ 0 & \text{if } x_i = x_j \end{cases} \quad .$$

Dissimilarities Based on Attributes

- Mixed attributes: *Gower Distance*
- Use distance measure between 0 and 1 for each variable
- Aggregate:

$$D_{\text{Gower}}(i,j) = \frac{\sum_f \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_f \delta_{ij}^{(f)}},$$

where $\delta_{ij}^{(f)} = 1$ if measurements x_i and x_j for the fth variable are non-missing; 0 otherwise.

- Use appropriate dissimilarity measure for each attribute

Notions of a Cluster Can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Same dataset

Types of Clusterings

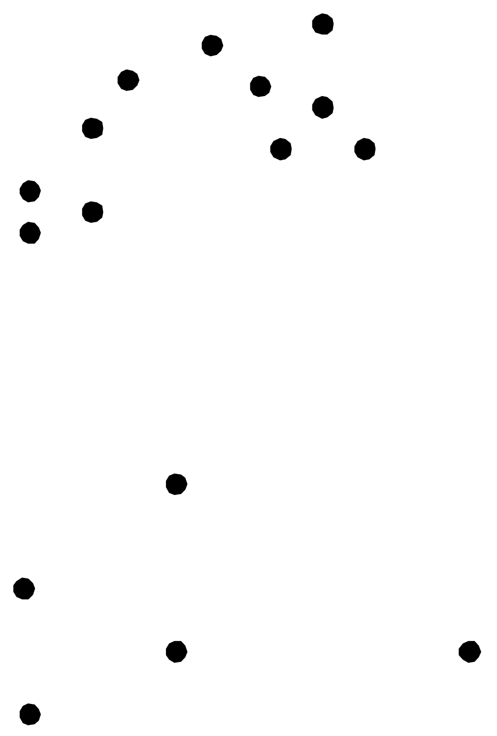
- A *clustering* is a set of clusters

- Important distinction between *hierarchical* and *partitional* sets of clusters

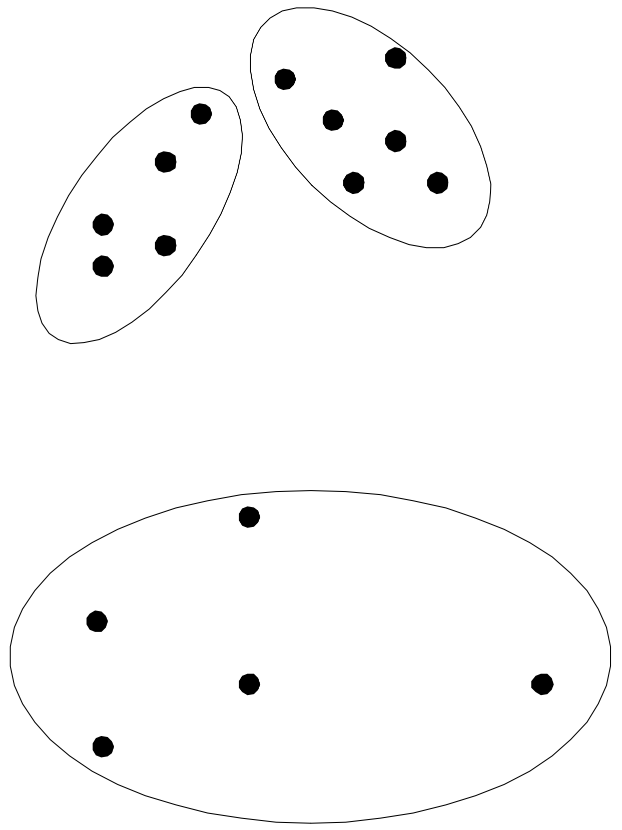
- *Partitional* Clustering
  - Division of data objects into non-overlapping subsets (clusters) such that each data *object is in exactly one subset*

- *Hierarchical* clustering
  - A set of nested *clusters organized as a hierarchical* tree

Partitional Clustering

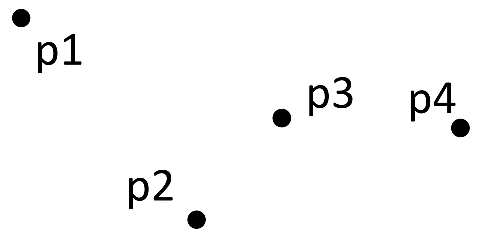


Original Points

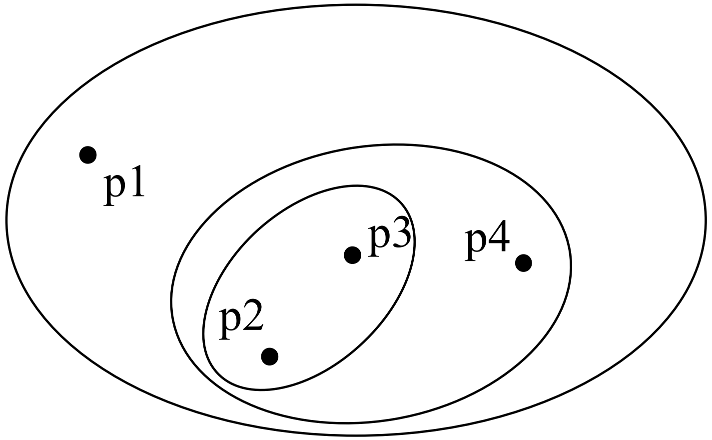


A Partitional Clustering

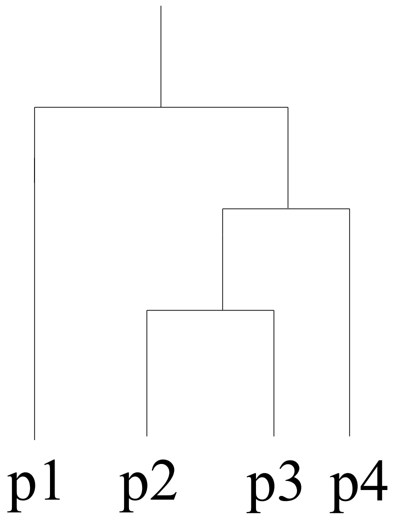
Hierarchical Clustering



Original Points



Nested Cluster Diagram



Dendrogram

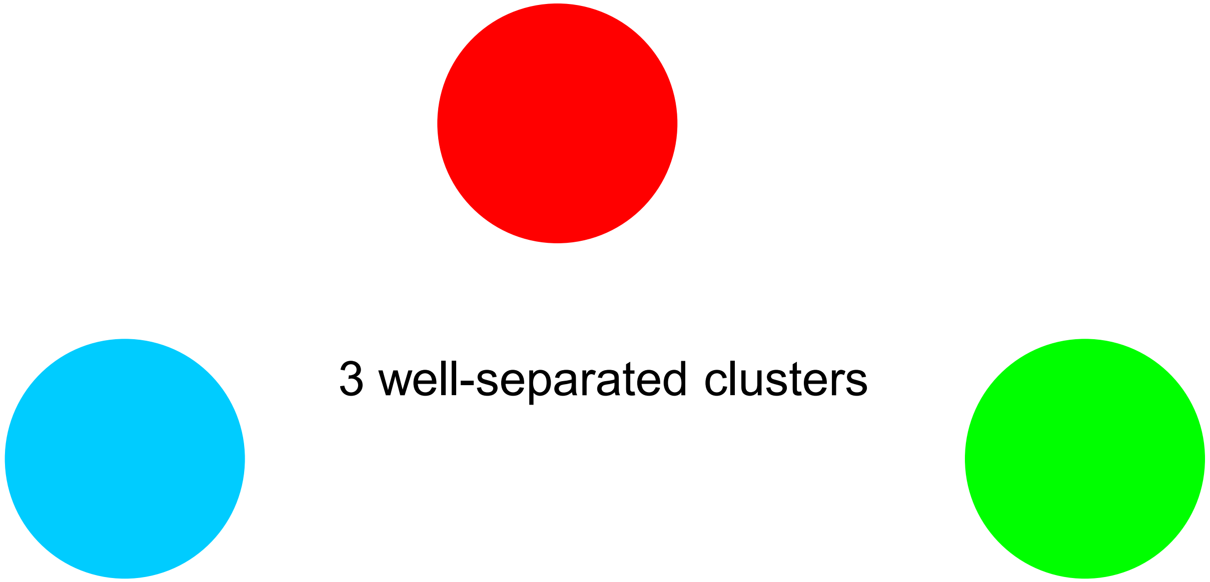
Hierarchical clustering of four points shown as a nested cluster diagram and a dendrogram.

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters
  - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
- Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster of widely different sizes, shapes, and densities

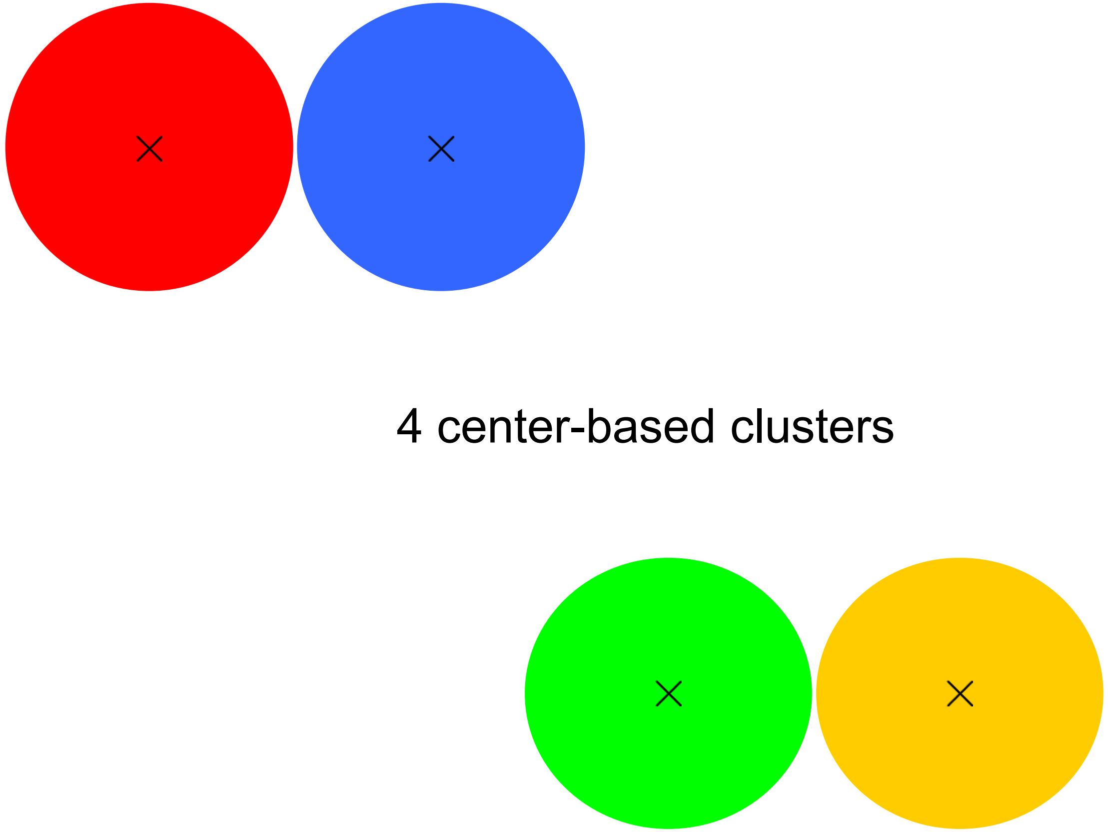
Types of Clusters

- Well-Separated Clusters:
  - Set of objects such that any object in a cluster is closer (or more similar) to every other object in the cluster than to any object not in the cluster



Types of Clusters

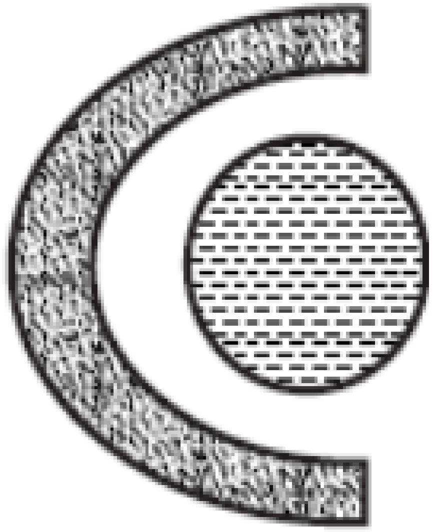
- Center-based Clusters:
  - Set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a *centroid*, the average of all the points in the cluster, or a *medoid*, the most "representative" point of a cluster



4 center-based clusters

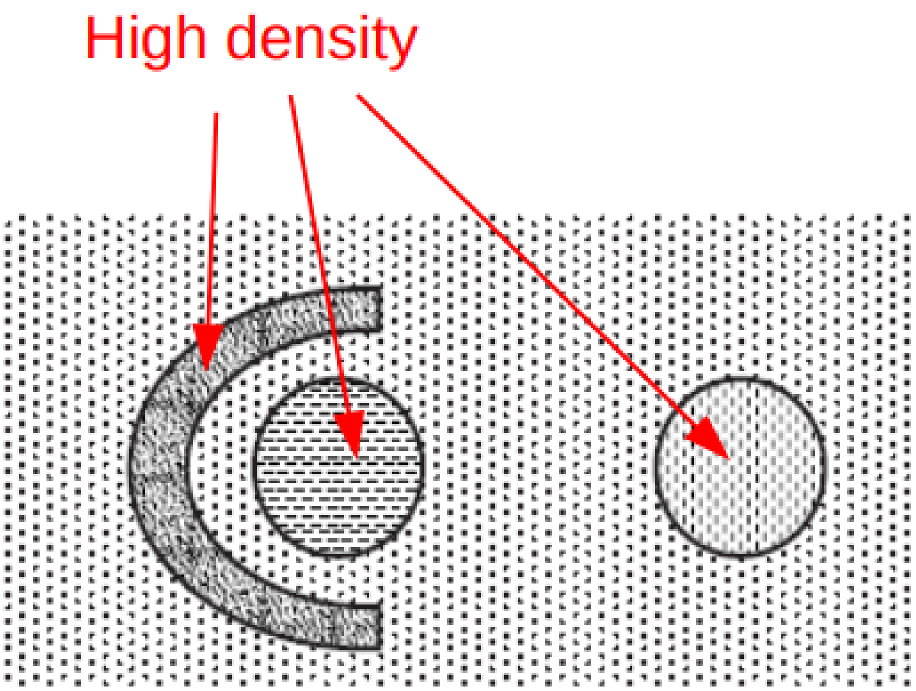
Types of Clusters

- Contiguous Cluster (Nearest neighbor or Transitive)
  - Set of objects such that an object in a cluster is closer (or more similar) to one or more other objects in the cluster than to any object not in the cluster



Types of Clusters

- Density-based Cluster
  - Dense region of objects that is surrounded by a region of low density
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present



Considerations Regarding Input Data

- Sparseness
- Attribute type
- Type of Data
- Dimensionality
- Noise and Outliers
- Type of Distribution

→Conduct preprocessing and select the appropriate dissimilarity or similarity measure

→Determine the objective of clustering and choose the appropriate method

Clustering Algorithms

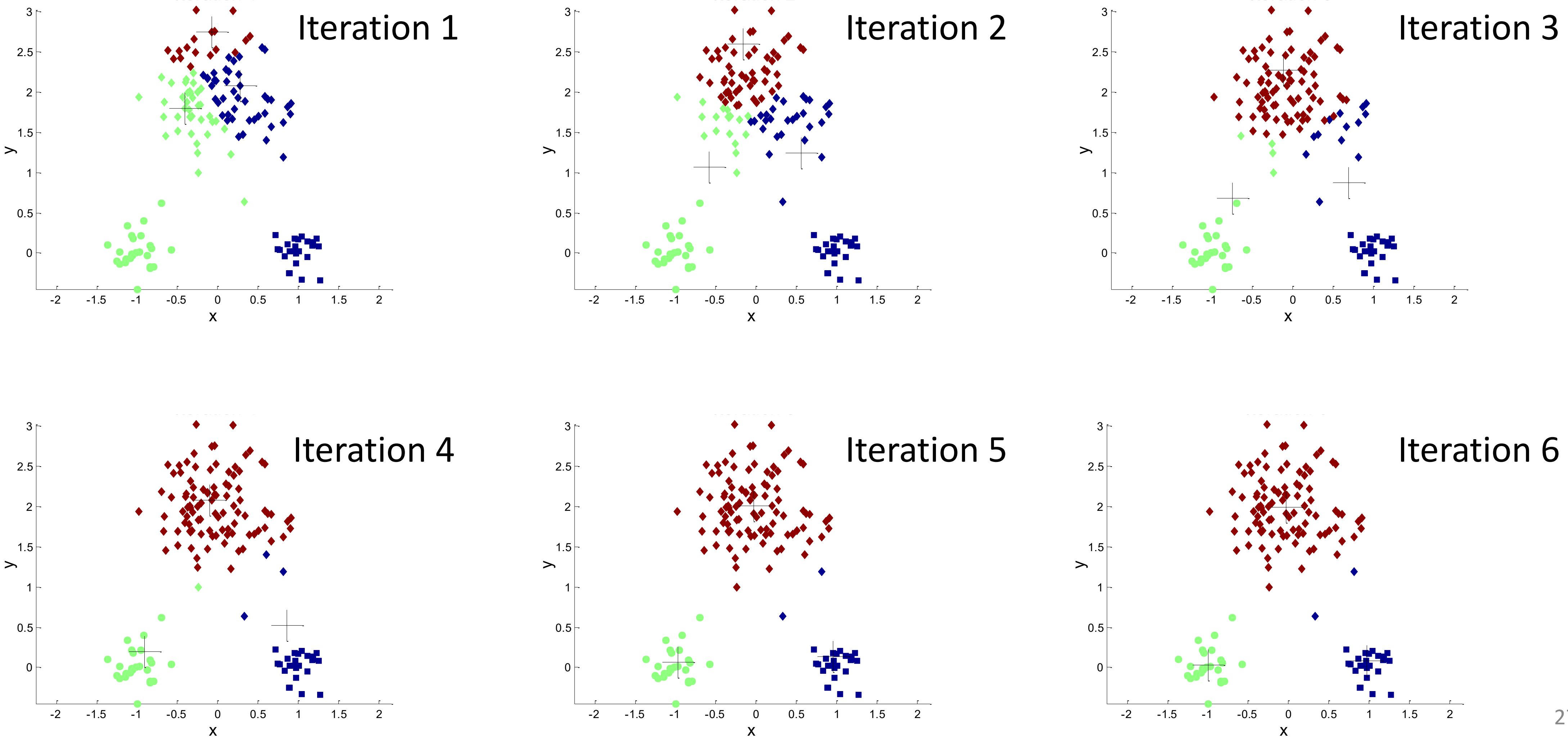
- K-means Clustering and its variants
- Hierarchical Clustering
- Density-based Clustering

K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a *centroid* (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is straightforward

Select K points as the initial centroids.
repeat
  Form K clusters by assigning all points to the closest centroid.
  Recompute the centroid of each cluster.
until The centroids don't change

Using K-means, $K=3$



K-means Clustering -- details

- Initial centroids are often chosen randomly
  - Clusters produced vary from one run to another
- The centroid is (typically) the mean of the points in the cluster
- 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above
- 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 $\mathcal{O}(nkdi)$

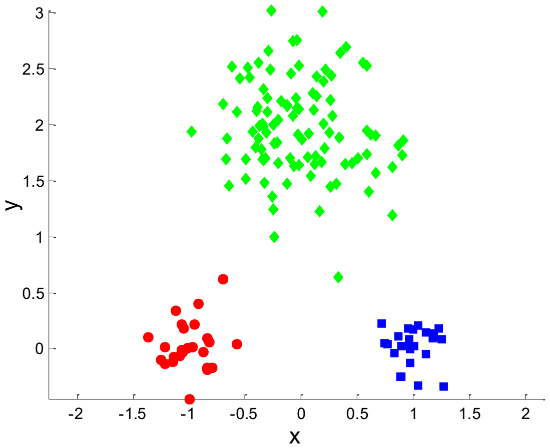
$$\text{where}$$

$$n = \text{number of points}, \; k = \text{number of clusters},$$

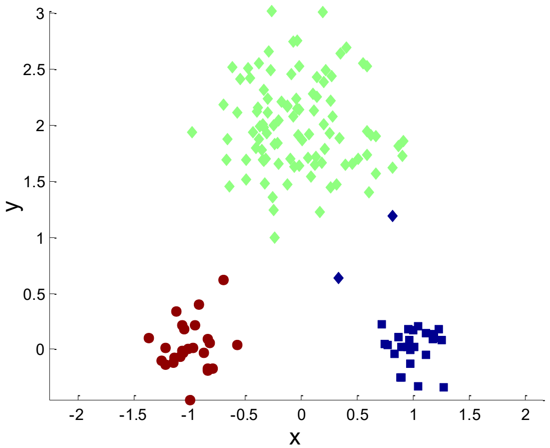
$$d = \text{number of attributes}, \; i = \text{number of iterations}$$

Two Different K-means Clusterings

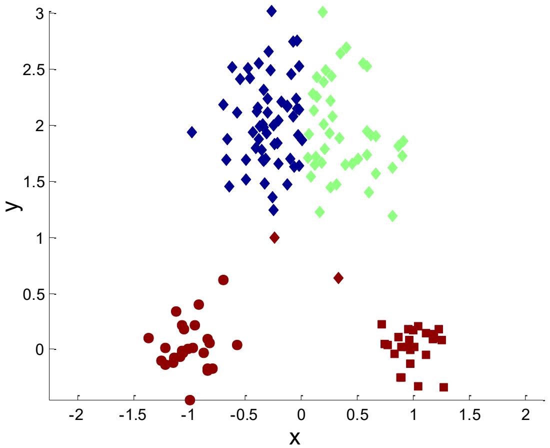
Different initial centroids



Original Points



Optimal Clustering



Sub-optimal Clustering

Evaluating K-means Clustering

- 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 over all clusters

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - m_i||^2$$

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

Use for Pre- / Post-Processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers

- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE
  - Can use these steps during the clustering process
    - e.g., ISODATA clustering algorithm

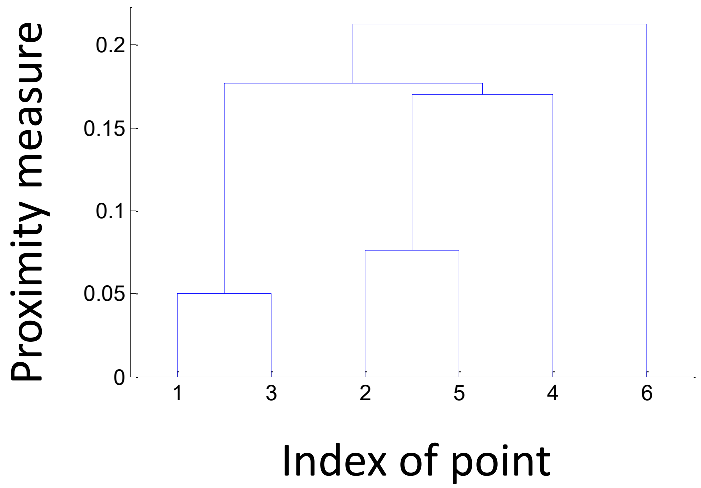
Limitations of K-means

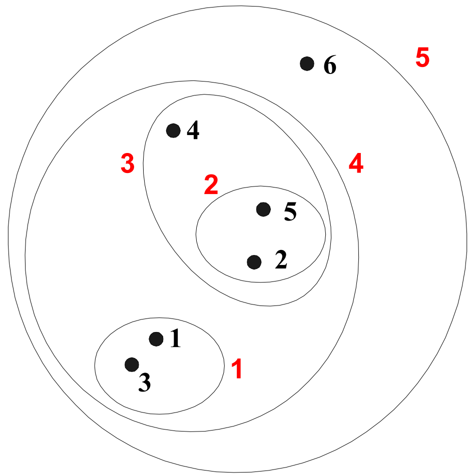
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers

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

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 than divisive hierarchical clustering techniques

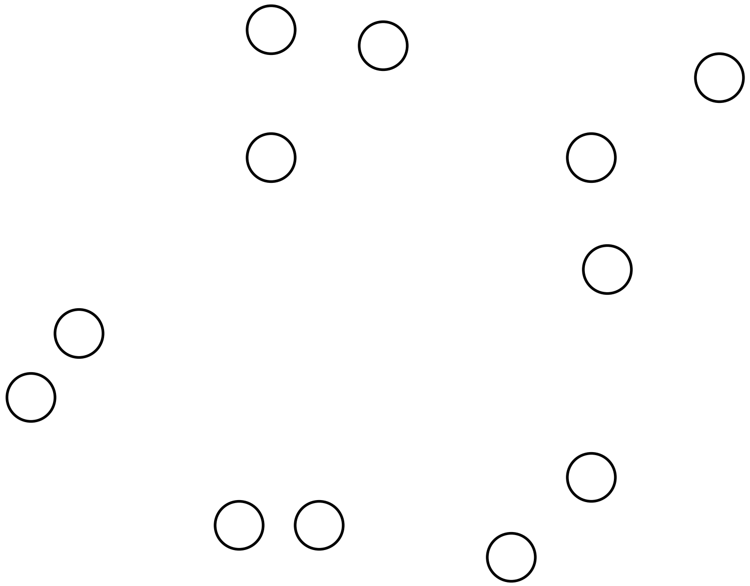
Algorithm 1: Agglomerative Clustering Algorithm

```
Compute the proximity matrix;
Let each data point be a cluster;
repeat
    Merge the two closest clusters;
    Update the proximity matrix;
until only a single cluster remains;
```

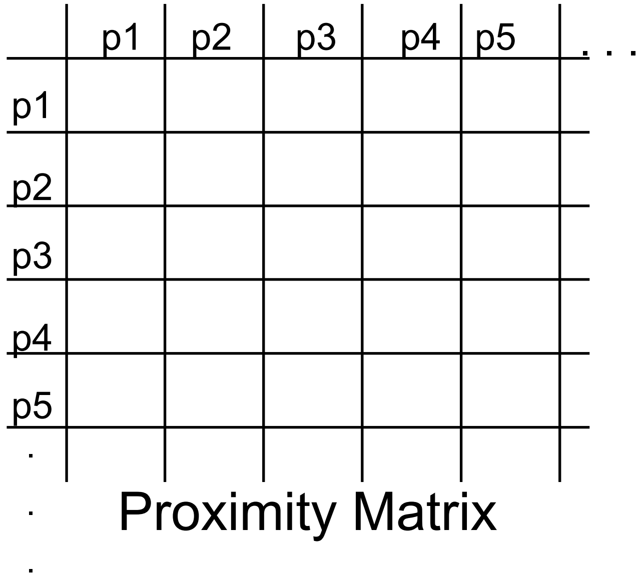
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

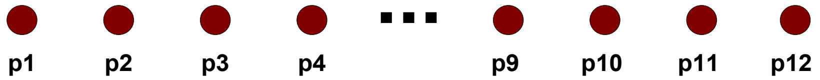
- Start with clusters of individual points and a proximity matrix



Individual Points

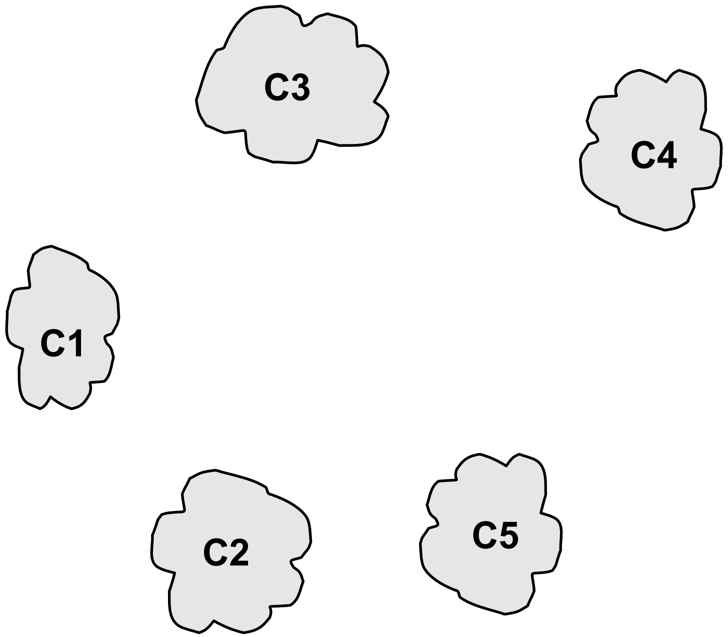


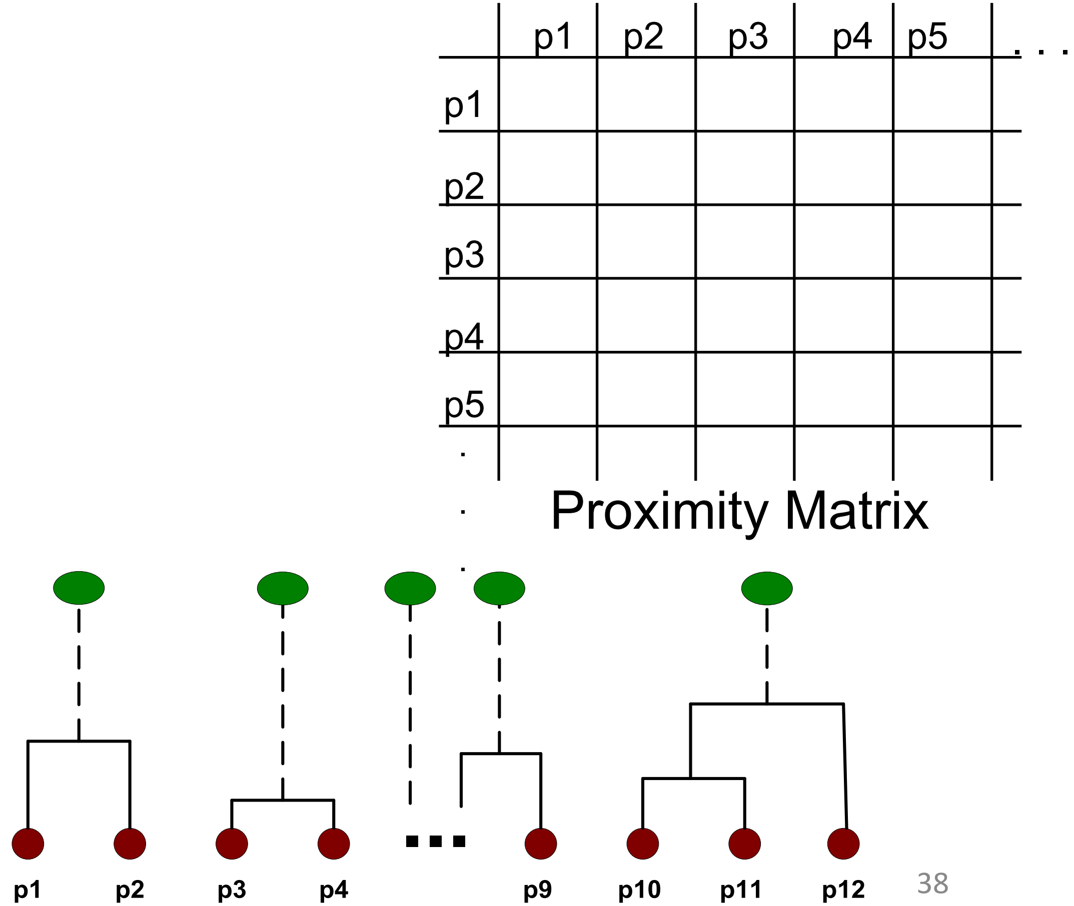
Proximity Matrix



Intermediate Situation

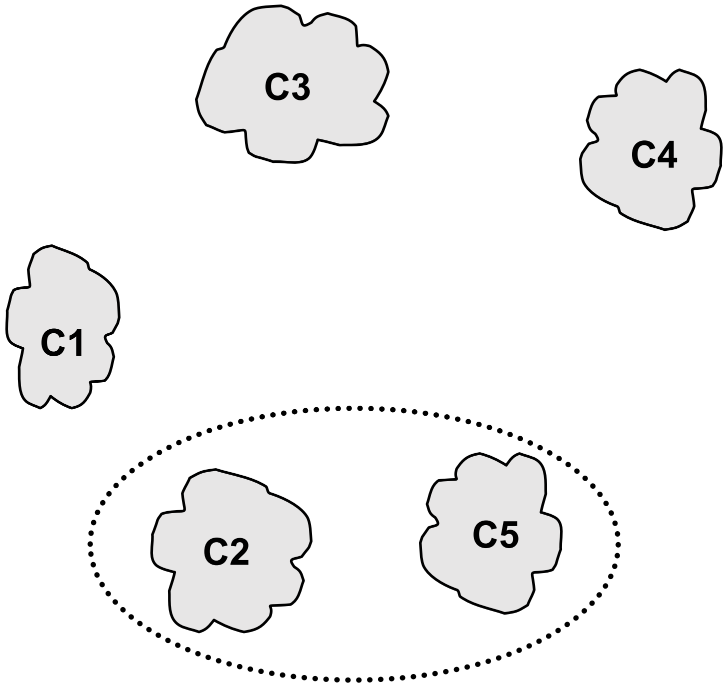
- After some merging steps, we have some clusters

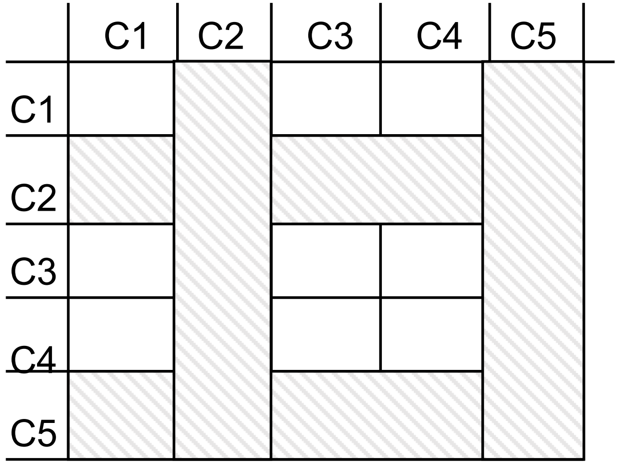




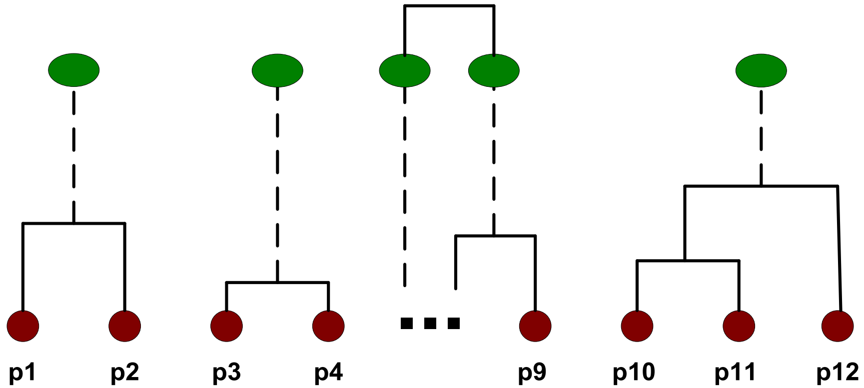
Intermediate Situation

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



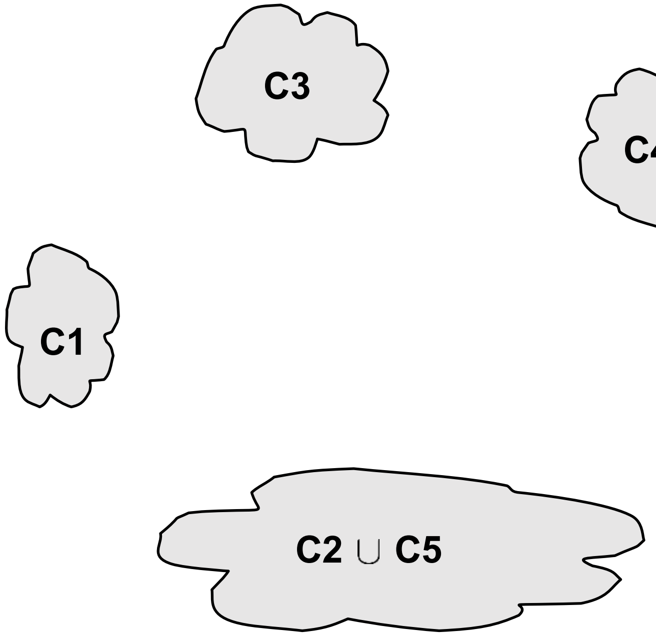


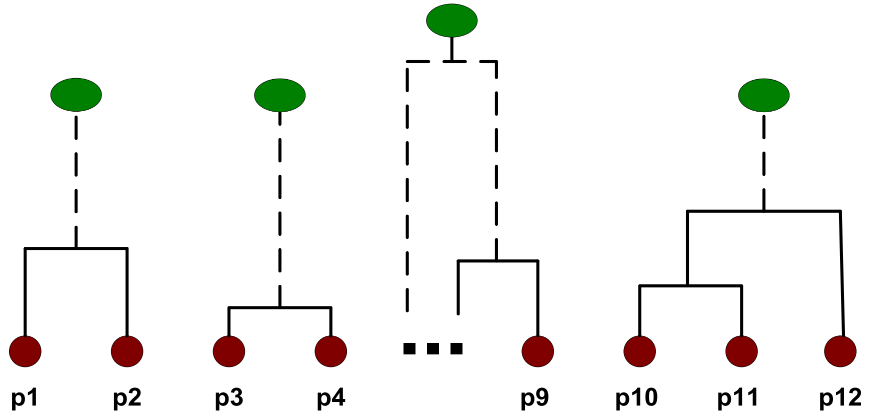
Proximity Matrix



After Merging

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

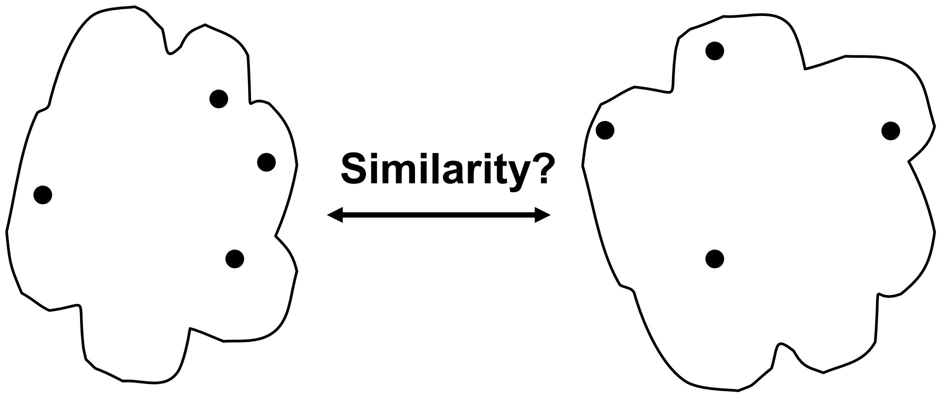




	C1	C2 ∪ C5	C3	C4
C1		?		
C2 ∪ C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix

How to Define Inter-Cluster Similarity

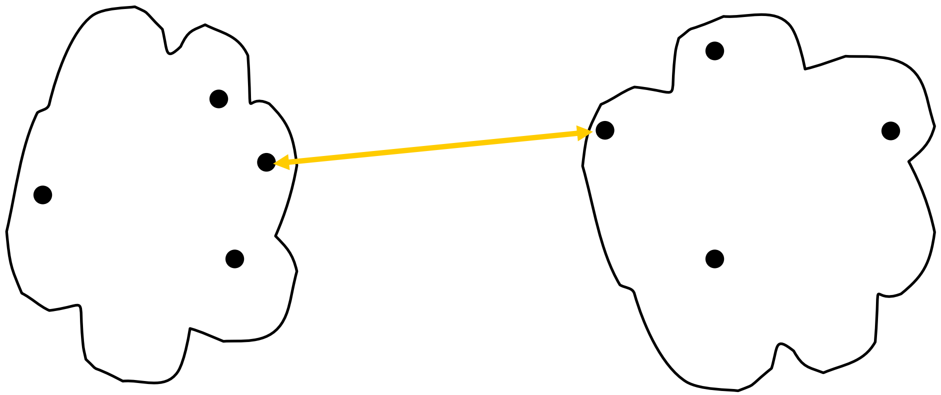


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

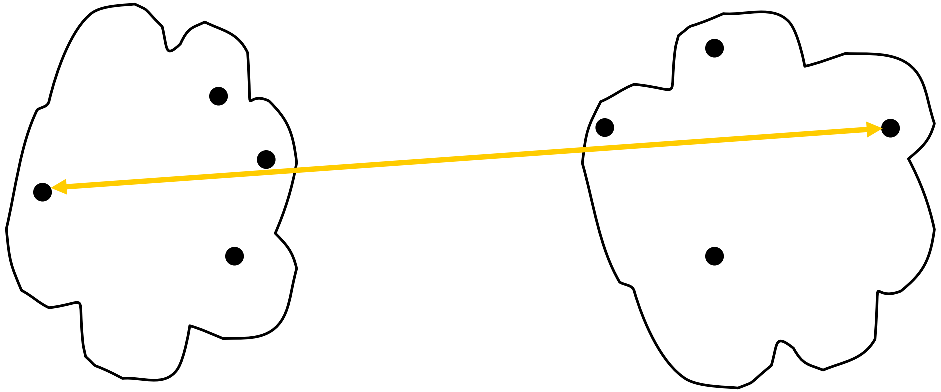


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

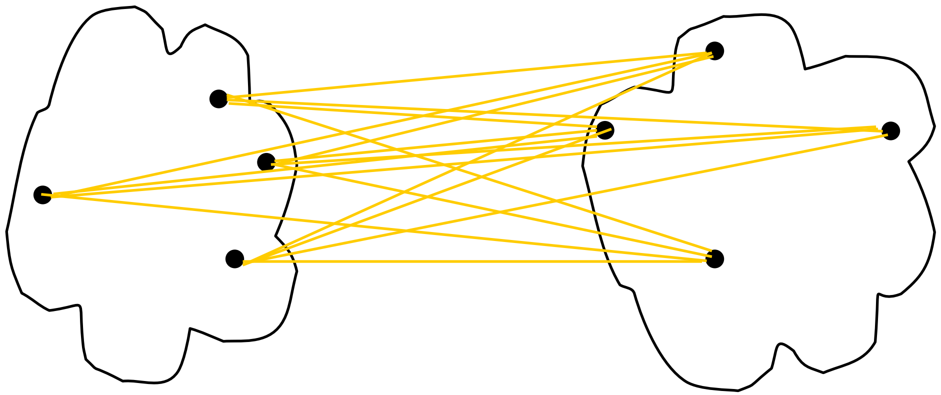


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

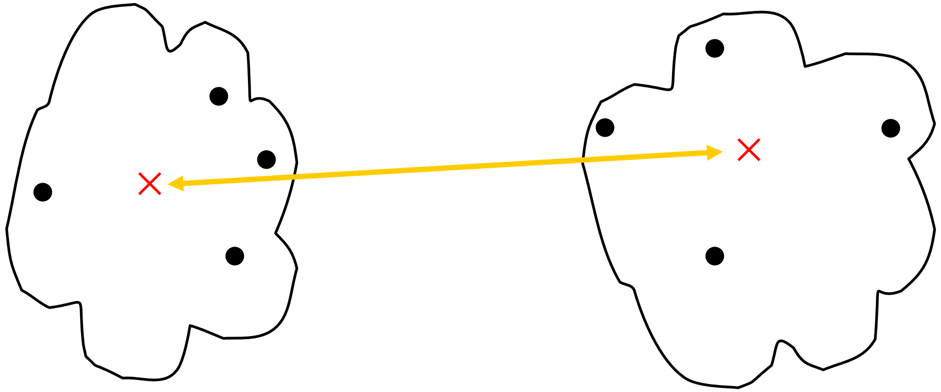


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity

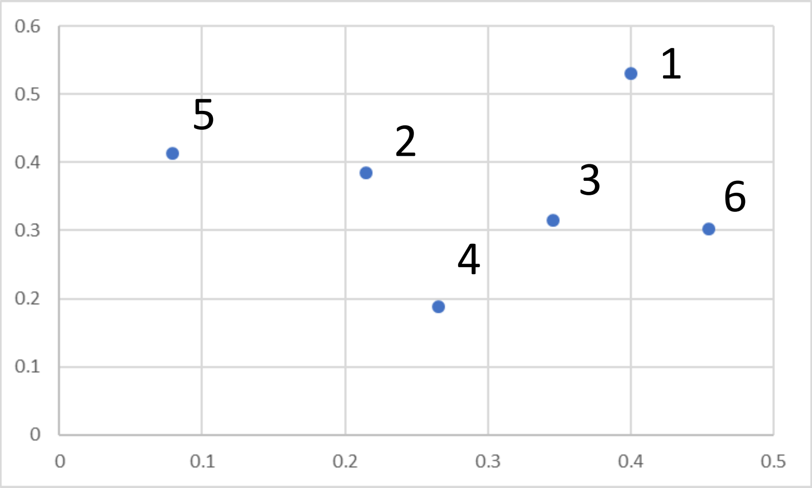


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

Example



Point	x-coordinate	y-coordinate
p1	0.4005	0.5306
p2	0.2148	0.3854
p3	0.3457	0.3156
p4	0.2652	0.1875
p5	0.0789	0.4139
p6	0.4548	0.3022

x, y coordinates of six points

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

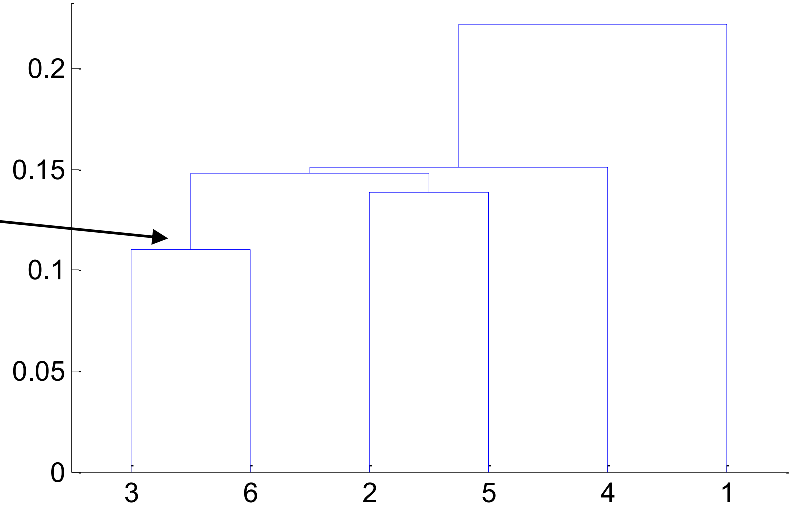
Euclidean distance matrix for six points

Cluster Similarity: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by *one link* in the proximity matrix

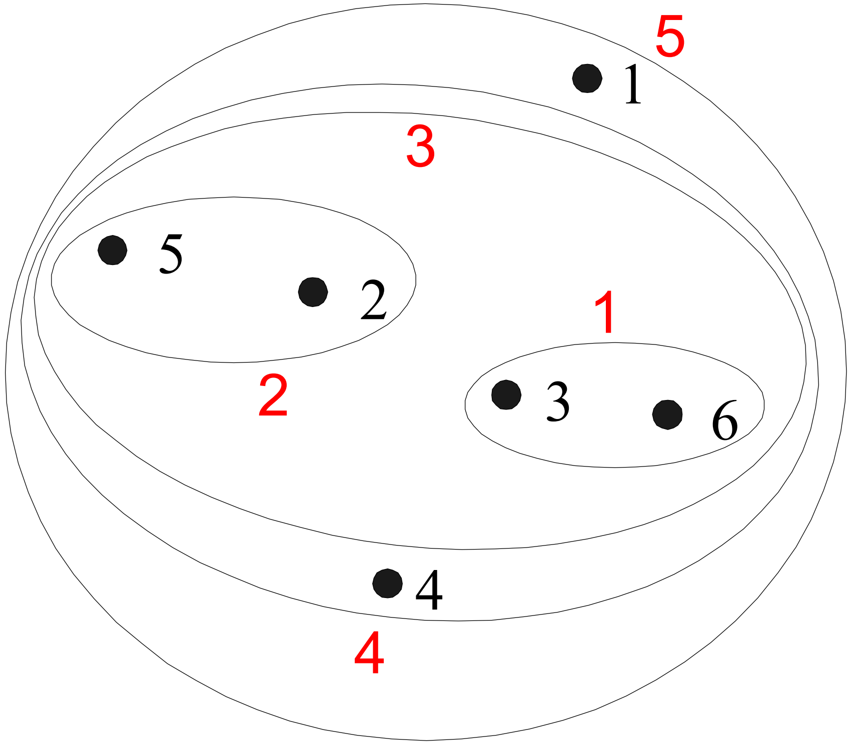
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Proximity Matrix

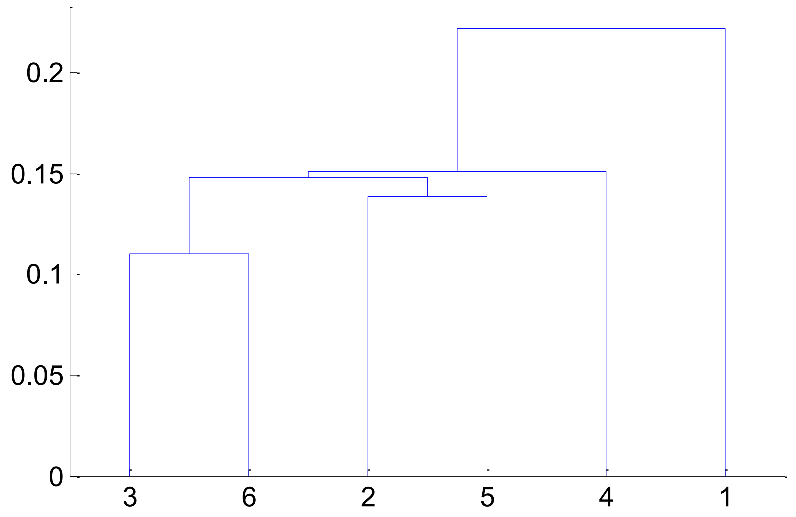


$$dist(\{3,6\},\{2,5\}) = \min(dist(3,2), dist(6,2), dist(3,5), dist(6,5))$$

Hierarchical Clustering: MIN



Good at handling non-elliptical shapes, but is sensitive to noise and outliers



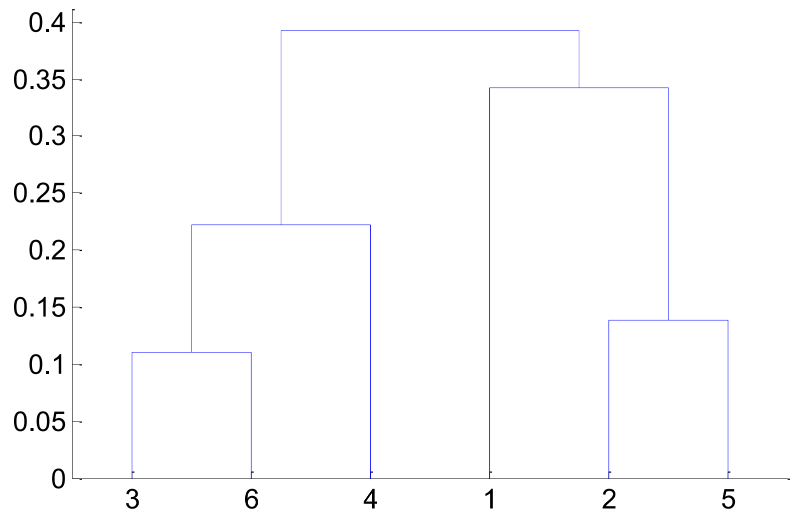
Nested Clusters

Dendrogram

Cluster Similarity: MAX or Complete Linkage

- Similarity of two clusters is based on the two most least (most distant) points in the different clusters
  - Determined by all pair of points in the two clusters

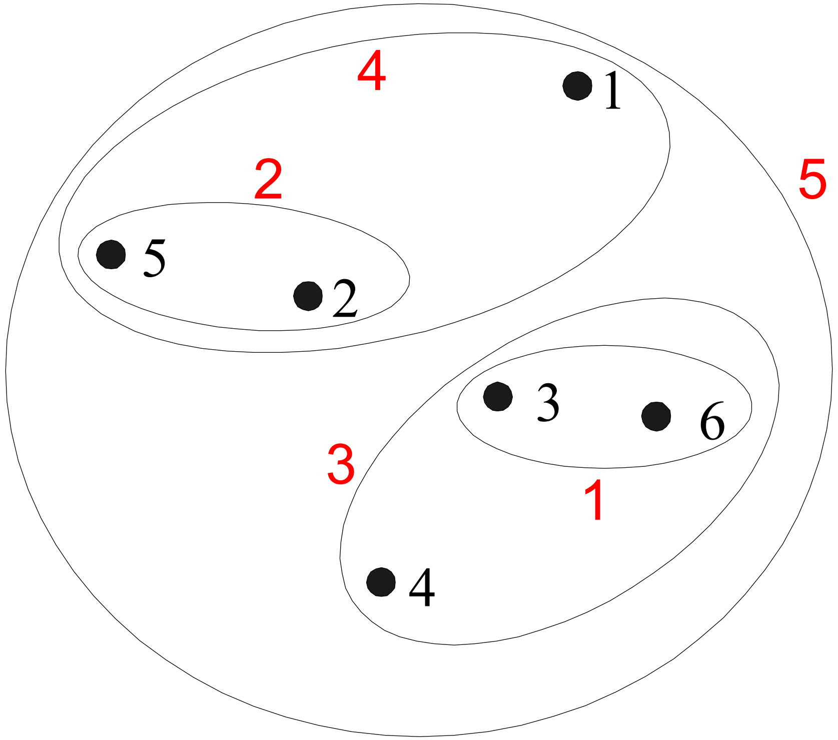
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



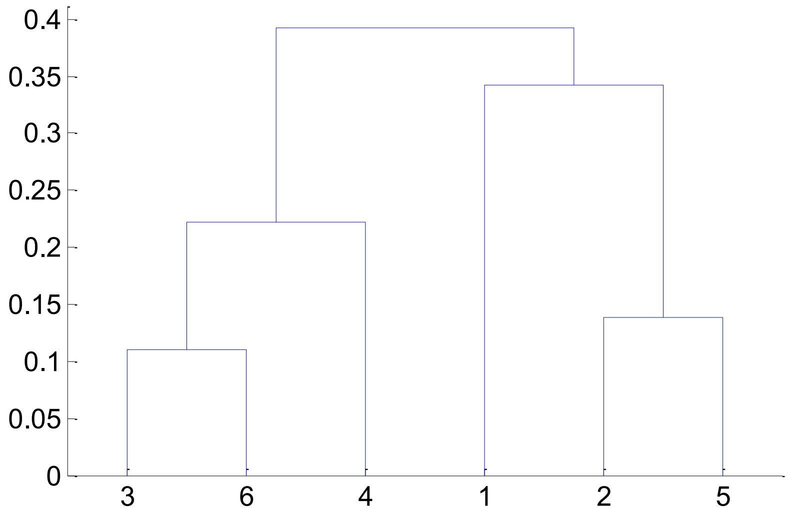
$$dist(\{3,6\},\{4\}) = \max\left(dist(3,4), dist(6,4)\right) = 0.22$$
$$dist(\{3,6\},\{2,5\}) = \max\left(dist(3,6), dist(6,2), dist(3,5), dist(6,5)\right) = 0.39$$
$$dist(\{3,6\},\{1\}) = \max\left(dist(3,1), dist(6,1)\right) = 0.23$$

Hierarchical Clustering: MAX

Less susceptible to noise and outliers, but can break large clusters and favors globular shapes



Nested Clusters



Dendrogram

Cluster Similarity: Group Average

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

$$\text{proximity}(C_i, C_j) = \frac{\sum_{p_i \in C_i, p_j \in C_j} \text{proximity}(p_i, p_j)}{|C_i||C_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

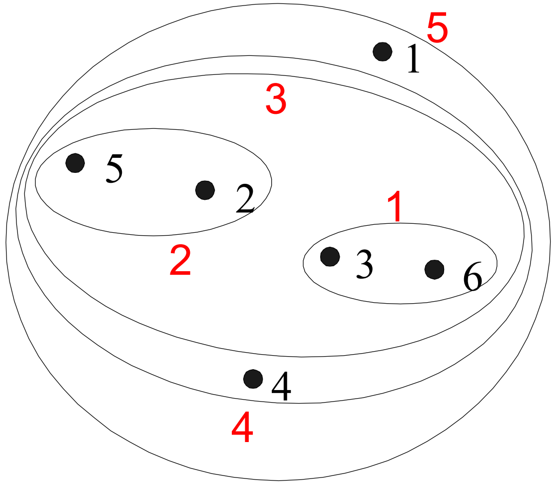
Intermediate approach between single and complete link

Cluster Similarity: 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
- Hierarchical analogue of K-means
  - Can be used to initialize K-means

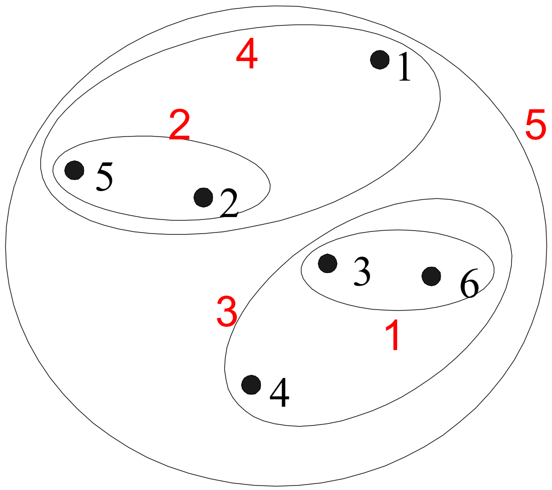
Less susceptible to noise and outliers, biased towards globular clusters

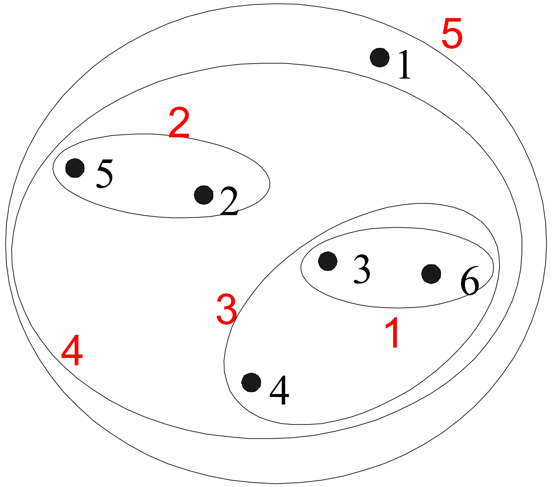
Hierarchical Clustering: Comparison



MIN

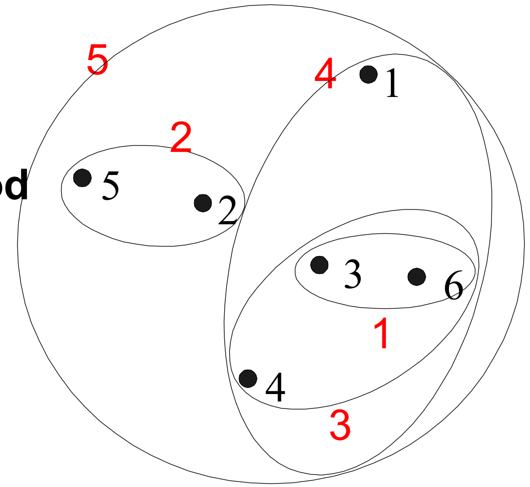
MAX



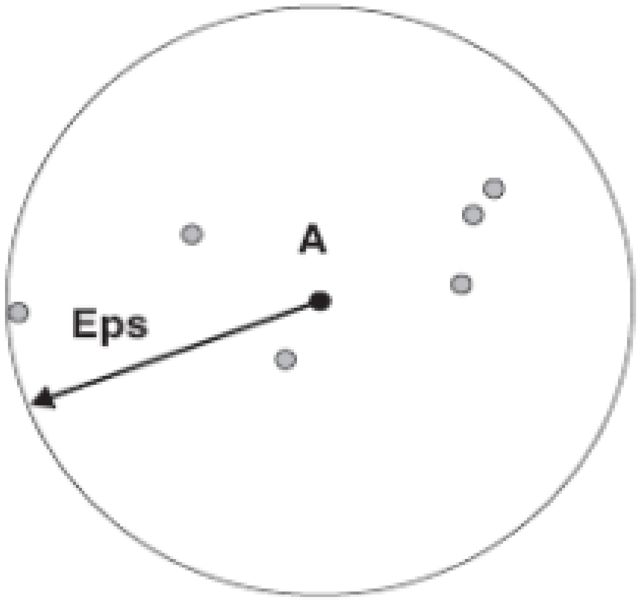


Group Average

Ward's Method



DBSCAN



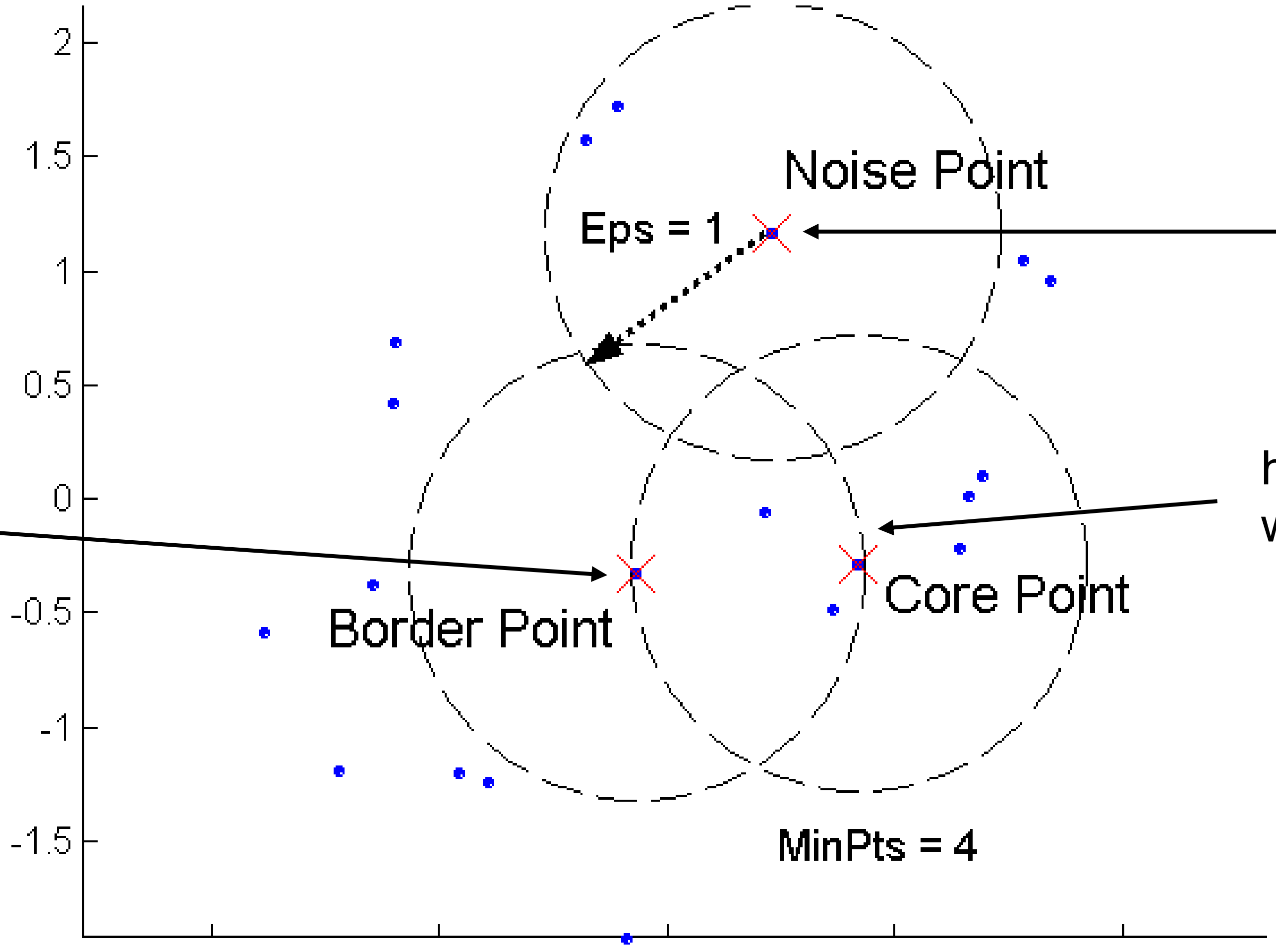
Density = 7 points

Density = number of points within a specified radius (Eps)

DBSCAN

- DBSCAN is a density-based algorithm
  - *Density* is the 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* points 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



not a core point nor a border point

has fewer than *4* points within *Eps*, but is in the neighborhood of a core point

has more than 4 points within *Eps*

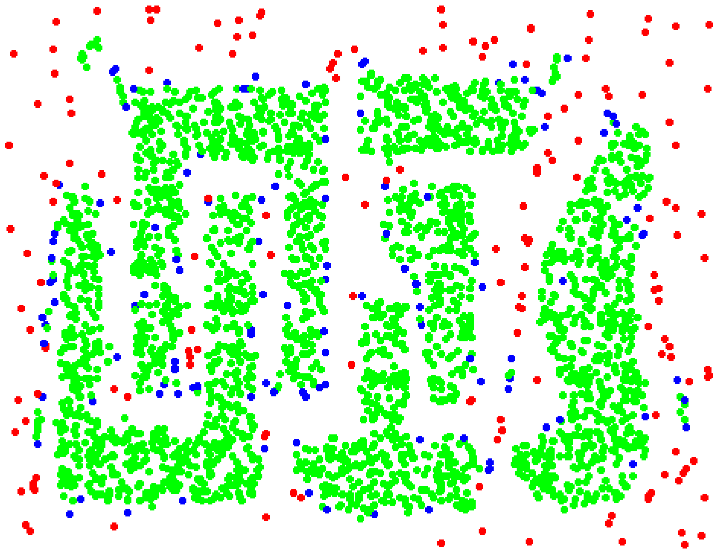
DBSCAN Algorithm

Algorithm 1: DBSCAN Clustering Algorithm

1. Label all points as core, border, or noise;
2. Eliminate noise points;
3. Put an edge between all core points that are within Eps of each other;
4. Make each group of connected core points into a separate cluster;
5. Assign each border point to one of the clusters of its associated core points;

DBSCAN: Core, Border, and Noise Points

Original Points

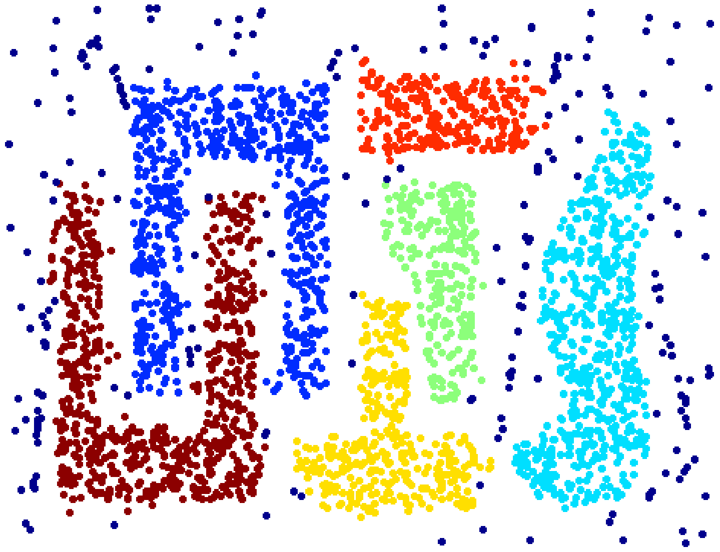


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

Original Points



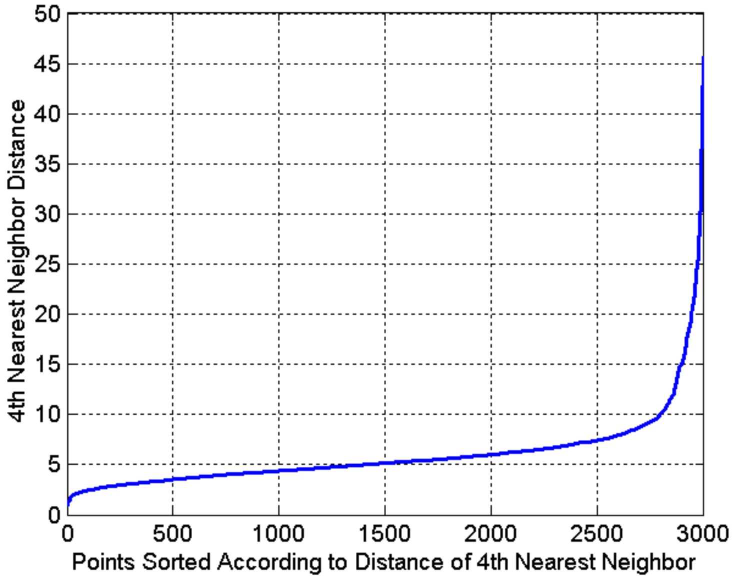
Clusters

- Works well for: Noisy data, clusters of different shapes and sizes
- Does not work well for: Varying densities, high-dimensional data

DBSCAN: 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

Take k as the MinPts parameter, break point as Eps



$k = 4$

Cluster Validation

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall, etc.

- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- Why validation?
  - To avoid finding clusters formed by chance
  - To compare clustering algorithms
  - To compare clusters

Aspects of Cluster Validation

- Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data
- Determining the correct number of clusters
- Comparing the results of a cluster analysis to *ground truth* (externally known results)
  - Externally provided class labels
  - External Indices
- Evaluating the quality of clusters **without** reference to external information
  - Use only the data
  - Internal Indices
- Determining the reliability of clusters
  - To what confidence level, the clusters are not formed by chance
  - Statistical framework

External Indices – Comparing to Ground Truth

- Comparing the results of a cluster analysis to ground truth (externally known results)
  - Externally provided class labels
  - External Indices

Similarity-based Measures

- Premise: any two objects that are in the same cluster should be in the same class and *vice versa*

- Assess cluster validity by comparing two similarity matrices
  - Class similarity matrix
  - Ideal cluster similarity matrix

Similarity-based Measures

- **Notation**

  N is the number of objects in dataset,

  O_i is the ith object,

  $L = \{L_1, \dots, L_s\}$ is the set of ground truth labels or classes,

  $C = \{C_1, \dots, C_t\}$ is the set of clusters reported by clustering algorithm.

- **Similarity matrices (**$L =$ **class similarity,** $C =$ **cluster similarity)**

  $N \times N$, both rows and columns correspond to objects,

  $L_{ij} = 1$, if O_i and O_j belong to the same *ground truth* class in L; $L_{ij} = 0$ otherwise,

  $C_{ij} = 1$, if O_i and O_j belong to the same cluster in C; $C_{ij} = 0$ otherwise,

Similarity-based Measures

- Example:

$$L_1 = \{O_1, O_2\} \text{ and } L_2 = \{O_3, O_4, O_5\}$$
$$C_1 = \{O_1, O_2, O_3\} \text{ and } C_2 = \{O_4, O_5\}$$

Point	O_1	O_2	O_3	O_4	O_5
O_1	1	1	0	0	0
O_2	1	1	0	0	0
O_3	0	0	1	1	1
O_4	0	0	1	1	1
O_5	0	0	1	1	1

L : Class similarity matrix.

Point	O_1	O_2	O_3	O_4	O_5
O_1	1	1	1	0	0
O_2	1	1	1	0	0
O_3	1	1	1	0	0
O_4	0	0	0	1	1
O_5	0	0	0	1	1

C : Ideal cluster similarity matrix.

Similarity-based Measures

- Take correlation of similarity matrices as a measure of cluster validity
  - Hubert's Gamma Statistic

  - In our example, correlation between the entries in the matrices is 0.359

- Use measures for binary similarity to measure cluster validity
  - Rand statistic
  - Jaccard statistic

Binary Similarity Measures

- A pair of data objects falls into one of the following categories:

$$
\begin{aligned}
f_{00} &= \text{number of pairs of objects having a different class and a different cluster,}\\
f_{01} &= \text{number of pairs of objects having a different class and the same cluster,}\\
f_{10} &= \text{number of pairs of objects having the same class and a different cluster,}\\
f_{11} &= \text{number of pairs of objects having the same class and the same cluster.}
\end{aligned}
$$

- Simple matching coefficient (Rand statistic) and Jaccard coefficient are the most widely used cluster validity measures:

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

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

Example: Rand and Jaccard

Point	O_1	O_2	O_3	O_4	O_5
O_1	1	1	1	0	0
O_2	1	1	1	0	0
O_3	1	1	1	0	0
O_4	0	0	0	1	1
O_5	0	0	0	1	1

C : Ideal cluster similarity matrix.

	Same Cluster	Different Cluster
Same Class	$f_{11} = 2$	$f_{10} = 2$
Different Class	$f_{01} = 2$	$f_{00} = 4$

Two-way contingency table.

Point	O_1	O_2	O_3	O_4	O_5
O_1	1	1	0	0	0
O_2	1	1	0	0	0
O_3	0	0	1	1	1
O_4	0	0	1	1	1
O_5	0	0	1	1	1

L : Class similarity matrix.

$$\text{Rand index} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} = \frac{6}{10}$$

$$\text{Jaccard coefficient} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} = \frac{1}{3}$$

Classification-Oriented Measures

- As with classification, measure the degree to which predicted class (cluster labels) labels correspond to actual class labels
  - Entropy
  - Purity
  - Precision, Recall, F-measure

		Prediction outcome		
		+	−	total
Actual value	+	True Positive	False Negative	P
	−	False Positive	True Negative	N
	total	P′	N′	

Classification-Oriented Measures

- Entropy
  - The degree to which each cluster consists of objects of a single class

1. Calculate class distribution for each cluster
2. Compute entropy of each cluster
3. Total entropy for a set of clusters is a weighted sum of the cluster entropies

Classification-Oriented Measures

- Entropy

$p_{ij} = \frac{m_{ij}}{m_i}$ is the probability that a member of cluster i belongs to class j,

where m_i is the number of objects in cluster i and m_{ij} is the number of objects of class j in cluster i,

$e_i = -\sum_{j=1}^{L} p_{ij} \log_2 p_{ij}$, where L is the number of classes,

$e = \sum_{i=1}^{K} \frac{m_i}{m} e_i$, where K is the number of clusters and m is the number of objects.

Classification-Oriented Measures

- Purity:
  - Another measure of the extent to which a cluster contains objects of a single class

$$\text{purity}(i) = \max_j p_{ij}$$

$$\text{purity} = \sum_{i=1}^{K} \frac{m_i}{m} \text{purity}(i),$$

where K is the number of clusters, m_i is the number of objects in cluster i, and m is the total number of objects.

Example: Purity

K-means clustering results for the LA Times document dataset.

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Total	Entropy	Purity
1	3	5	40	506	96	27	677	1.2270	0.7474
2	4	7	280	29	39	2	361	1.1472	0.7756
3	1	1	1	7	4	671	685	0.1813	0.9796
4	10	162	3	119	73	2	369	1.7487	0.4390
5	331	22	5	70	13	23	454	1.3976	0.7134
6	5	358	12	212	48	13	648	1.5523	0.5525
Total	354	555	341	943	273	738	3204	1.1450	0.7203

$$\text{For Cluster 1, purity}(1) = \max_{j=1,\ldots,6} p_{1j} = \max_{j=1,\ldots,6} \left(\frac{3}{677}, \frac{5}{677}, \ldots, \frac{27}{677}\right) = \frac{506}{677} = 0.7474$$

$$\text{purity}_{total} = \sum_{i=1}^{6} \frac{m_i}{m}\text{purity}(i) = \frac{677}{3204}\text{purity}(1) + \frac{361}{3204}\text{purity}(2) + \cdots + \frac{648}{3204}\text{purity}(6) = 0.7203$$

Example: Precision/Recall

K-means clustering results for the LA Times document dataset.

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Total	Entropy	Purity
1	3	5	40	506	96	27	677	1.2270	0.7474
2	4	7	280	29	39	2	361	1.1472	0.7756
3	1	1	1	7	4	671	685	0.1813	0.9796
4	10	162	3	119	73	2	369	1.7487	0.4390
5	331	22	5	70	13	23	454	1.3976	0.7134
6	5	358	12	212	48	13	648	1.5523	0.5525
Total	354	555	341	943	273	738	3204	1.1450	0.7203

- Consider Cluster 1 and the Metro Section

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = 506/677 = 0.75$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 506/943 = 0.54$$

$$\text{F}_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.62$$

- Consider Cluster 3 and the Sports Section

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = 671/685 = 0.98$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 671/738 = 0.91$$

$$\text{F}_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.94$$

Internal Indices

- “Ground truth” may be unavailable
- Use on the data to measure cluster quality
  - Measure the cohesion and separation of clusters
  - Calculate the correlation between clustering results and similarity matrix
- SSE is good for comparing two clusterings or two clusters (average SSE)
- Can also be used to estimate the number of clusters

Internal Measures: Cohesion and Separation

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

Cohesion is measured by the within cluster sum of squares (WSS)

$$\mathrm{BSS} = \sum_i |C_i|(m - m_i)^2,$$

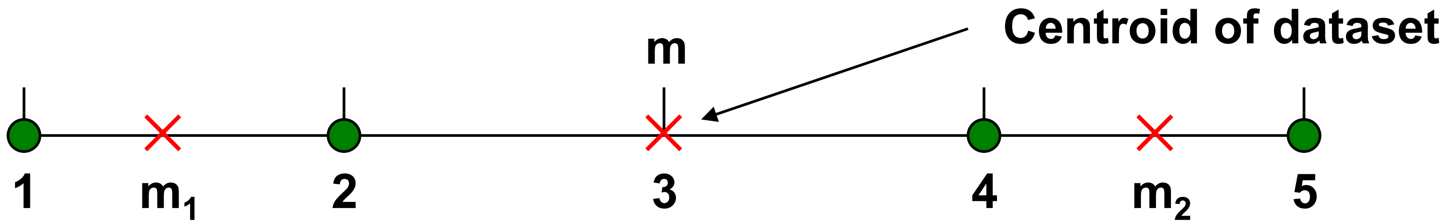
Separation is measured by the between cluster sum of squares (BSS)

where $|C_i|$ is the size of cluster i, and m is the centroid of the whole dataset.

$$\mathrm{WSS} + \mathrm{BSS} = \mathrm{constant}$$

- WSS (Cohesion) measure is called Sum of Squared Error (SSE) -- a commonly used measure
- A larger number of clusters tend to result in smaller SSE

Example: WSS and BSS



- $K = 1$ cluster:

$$\text{WSS} = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$
$$\text{BSS} = 4 \times (3-3)^2 = 0$$
$$\text{Total} = 10 + 0 = 10$$

- $K = 2$ clusters:

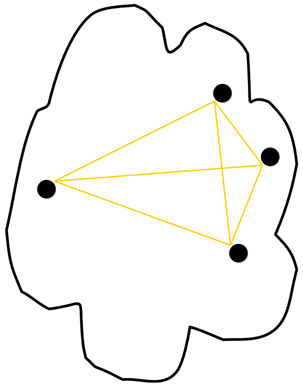
$$\text{WSS} = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$
$$\text{BSS} = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$
$$\text{Total} = 1 + 9 = 10$$

- $K = 4$ clusters:

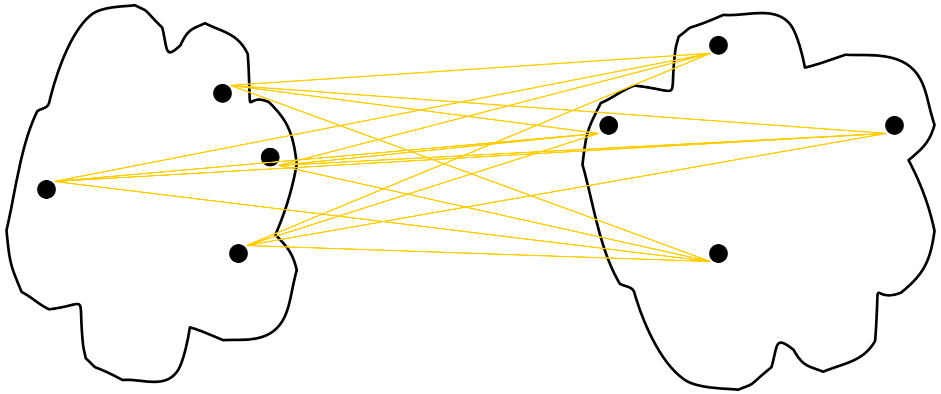
$$\text{WSS} = (1-1)^2 + (2-2)^2 + (4-4)^2 + (5-5)^2 = 0$$
$$\text{BSS} = 1 \times (3-1)^2 + 1 \times (3-2)^2 + 1 \times (3-4)^2 + 1 \times (3-5)^2 = 10$$
$$\text{Total} = 0 + 10 = 10$$

Internal Measures: Cohesion and Separation

- A proximity graph-based approach can also be used for cohesion and separation
  - Cluster cohesion is the sum of the weight of all links within a cluster
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster



cohesion



separation

Internal Measures: Silhouette Coefficient

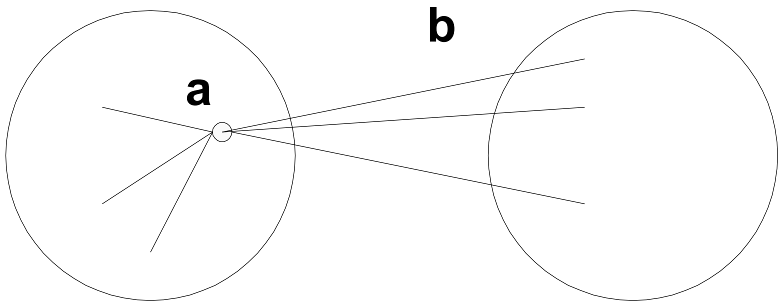
- *Silhouette Coefficient*: combines ideas of both **cohesion** and **separation**,
- For an individual object, i
  - Calculate a_i = average distance of i to all other objects in its cluster
  - Calculate b_i = min (average distance of i to objects in another cluster)
- The *silhouette coefficient* s_i for an object is then given by

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$

Internal Measures: Silhouette Coefficient

- The best value is 1 and the worst value is -1
- Values near 0 indicate overlapping clusters
- Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$



- Can calculate average silhouette width for a cluster or a clustering (measures **goodness of a clustering**)
- *Silhouette score* is reported as the mean Silhouette Coefficient for all samples

Internal Measures: Measuring Cluster Validity via Correlation

- Two matrices
  - Proximity Matrix

D_{ij} is the similarity between O_i and O_j.

  - Incidence Matrix

$N \times N$, one row and one per object,

$C_{ij} = 1$, if O_i and O_j belong to the same cluster in C; $C_{ij} = 0$ otherwise.

- Compute the correlation between the two matrices

$$\text{Symmetric} \rightarrow \text{only } \frac{n(n-1)}{2} \text{ entries need to be calculated.}$$

- **High correlation** indicates **good clustering**

Internal Measures: Measuring Cluster Validity via Correlation

Given Proximity Matrix, $D = \{d_{11}, d_{12}, \ldots d_{nn}\}$

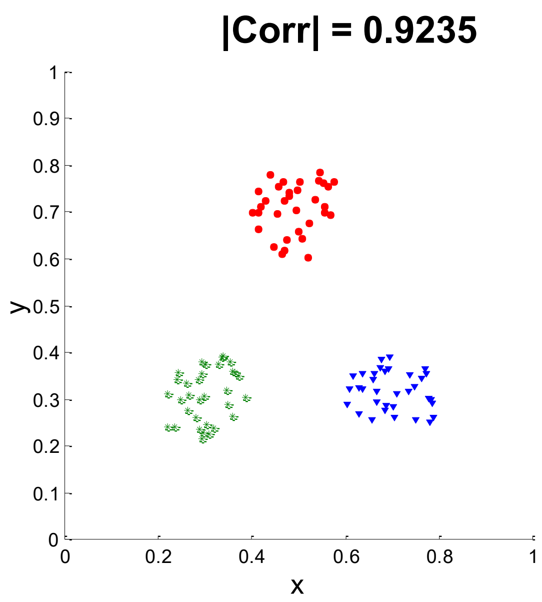
and Incidence Matrix, $C = \{c_{11}, c_{12}, \ldots, c_{nn}\}$,

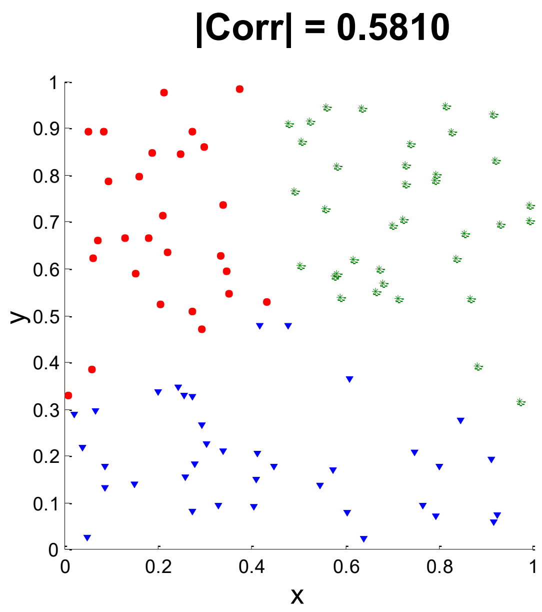
Correlation between D and C is given by:

$$r = \frac{\sum_{i=1,j=1}^{n} (d_{i,j} - \bar{d})(c_{i,j} - \bar{c})}{\sqrt{\sum_{i=1,j=1}^{n} (d_{i,j} - \bar{d})^2} \sqrt{\sum_{i=1,j=1}^{n} (c_{i,j} - \bar{c})^2}}$$

Internal Measures: Measuring Cluster Validity via Correlation

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

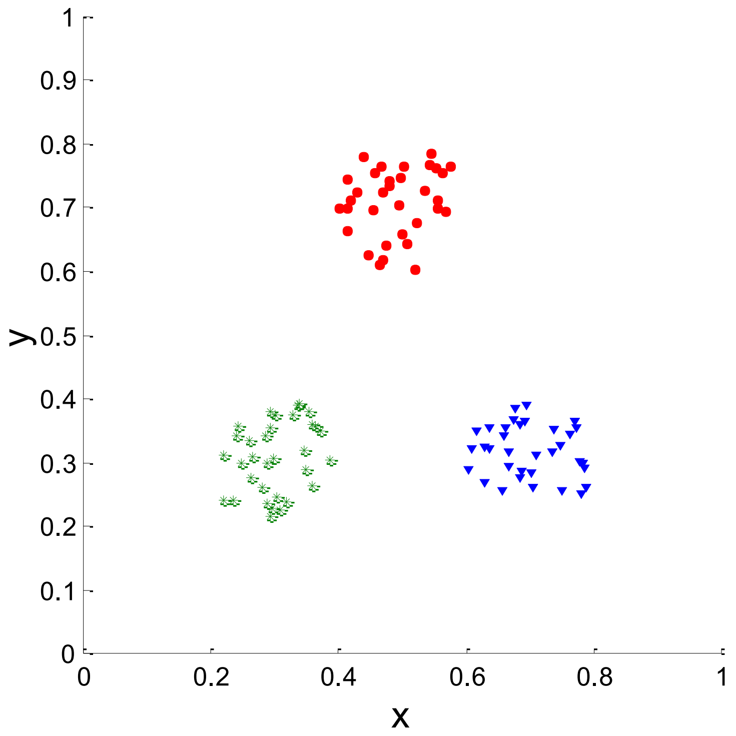


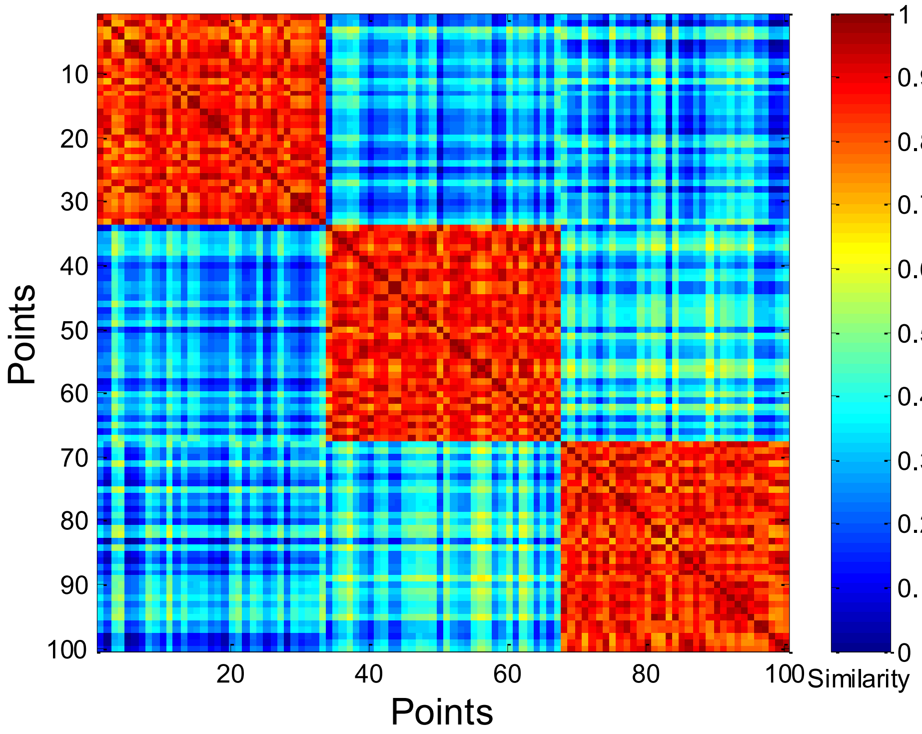


Clusters found by K-means in the random data are worse than clusters found by K-means in the well-separated clusters

Using Similarity Matrix for Cluster Validation

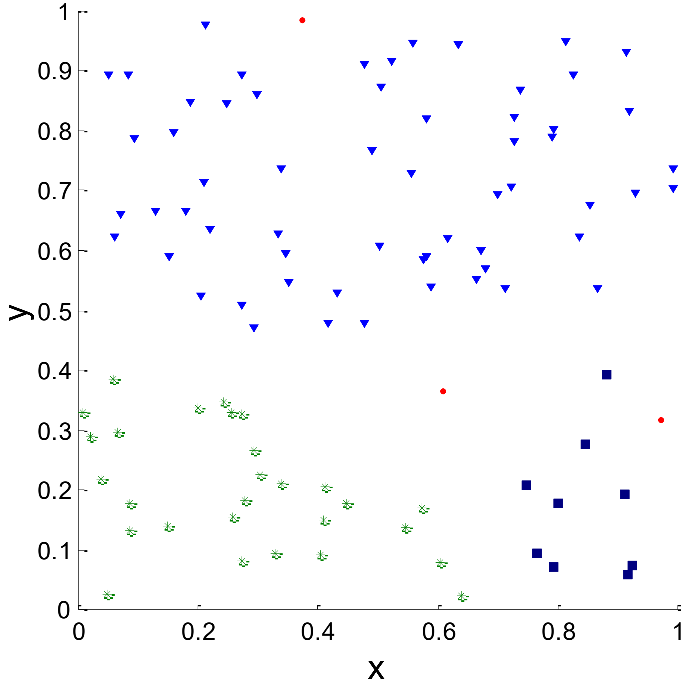
- Order the similarity matrix with respect to cluster labels and inspect visually

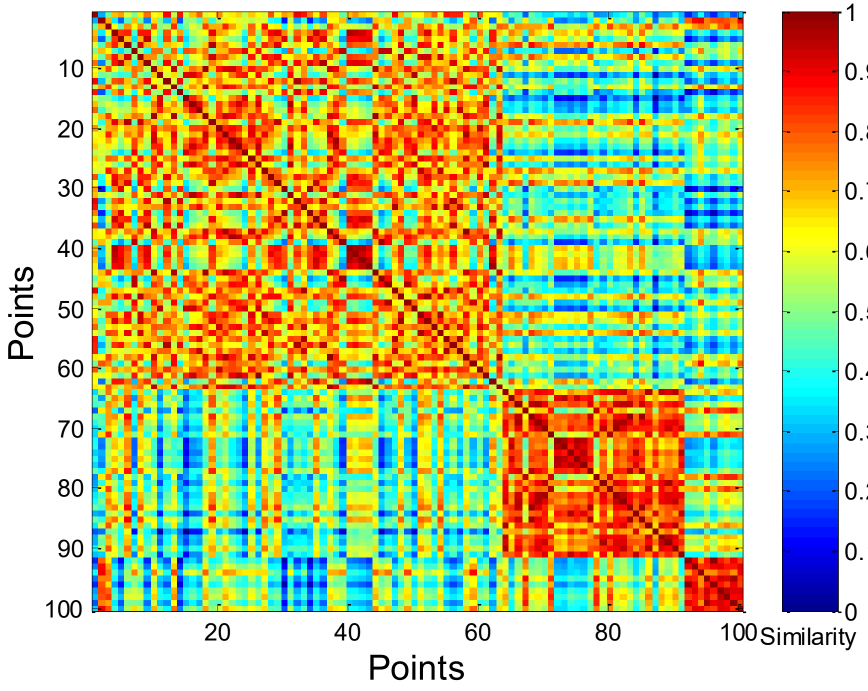




Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp





Framework for Cluster Validity

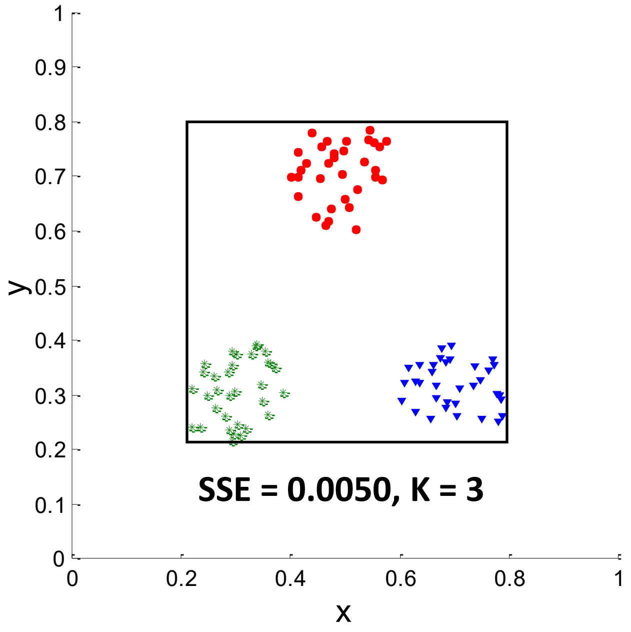
- Need a framework to interpret any measure
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- A statistical approach provides a framework for cluster validity
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result
    - If the value of the index is unlikely, then the cluster results are valid

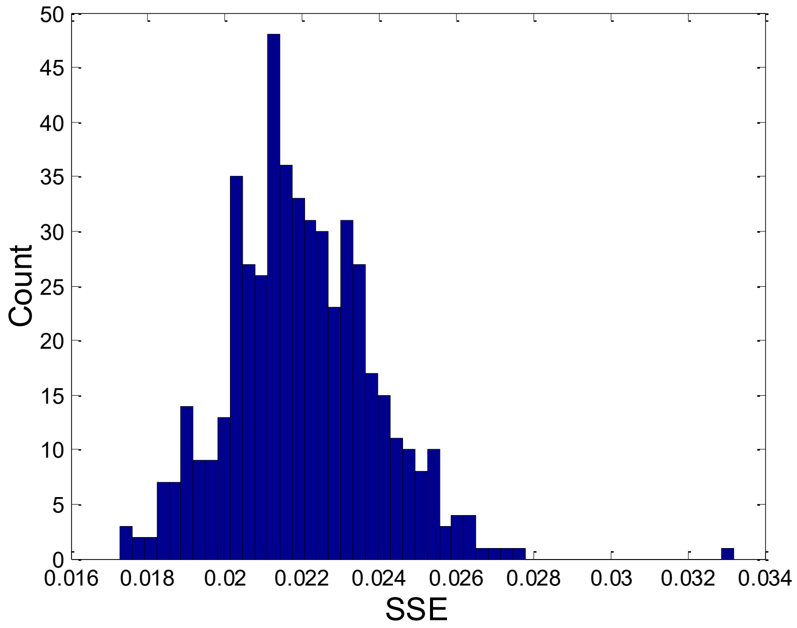
Statistical Framework for SSE

- Significance of SSE
  - Measure how good clustering is with respect to random data
  - Generate many random sets of 100 points having same range as points in the clustering case
  - Find equivalent number of clusters in each data set using same clustering algorithm
  - Accumulate distribution of SSE values for these clusterings
  - Using the distribution of SSE values → estimate P(SSE value)
  - Compare SSE of original case with random sets
  - Assess likelihood that clustering could occur by chance

Statistical Framework for SSE

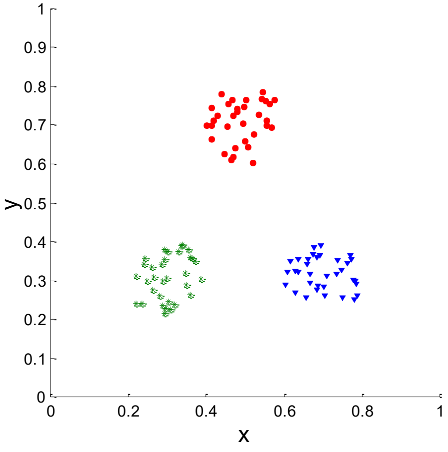
- Example
  - Compare SSE of 0.0050 against three clusters in random data
  - Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

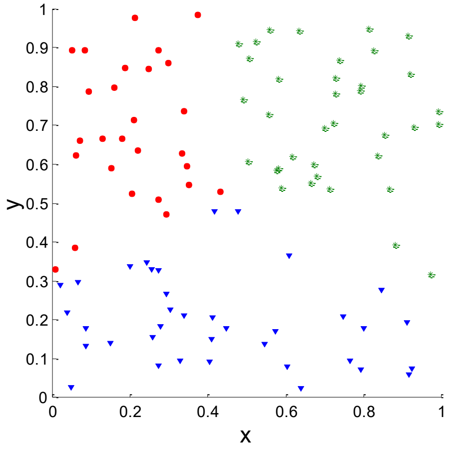


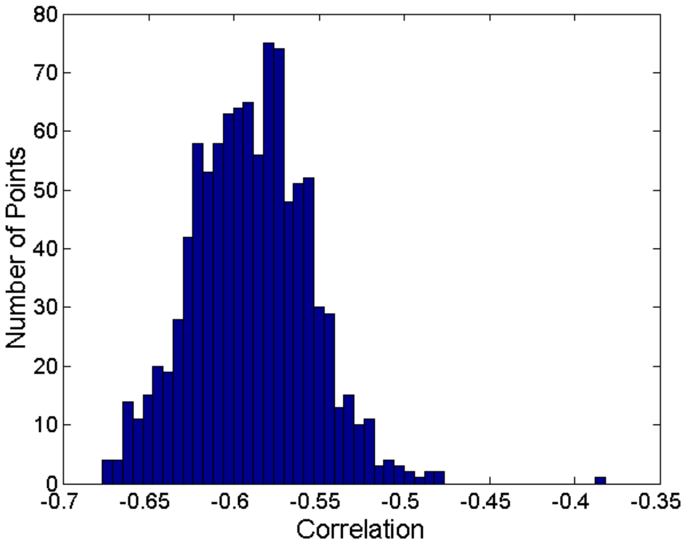


Statistical Framework for SSE

- Example
  - Correlation of incidence and proximity matrices for the K-means clusterings of the following two datasets







|Corr| = 0.9235

|Corr| = 0.5810

Final Comment on Cluster Validation

"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

Recap

- Introduction/Motivation
- Taxonomy of Cluster Analysis
- Clustering Algorithms
  - K-Means Clustering
  - Hierarchical Clustering
  - Density-based Clustering
- Cluster Validation