CSX4202/ITX4202: Data Mining Lecture 7-8

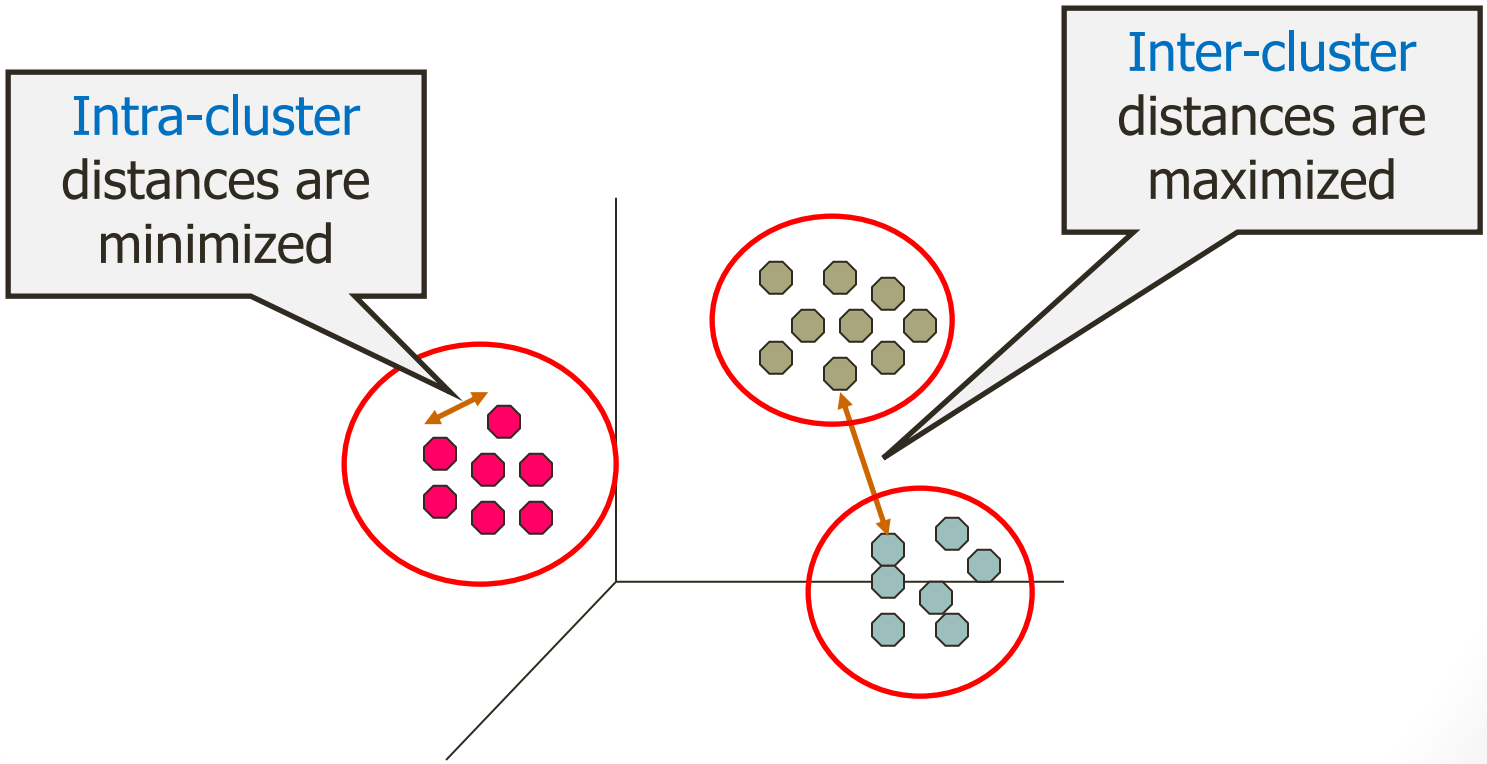
Asst. Prof. Dr. Rachsuda Setthawong
Computer Science Department
Assumption University

Outlines

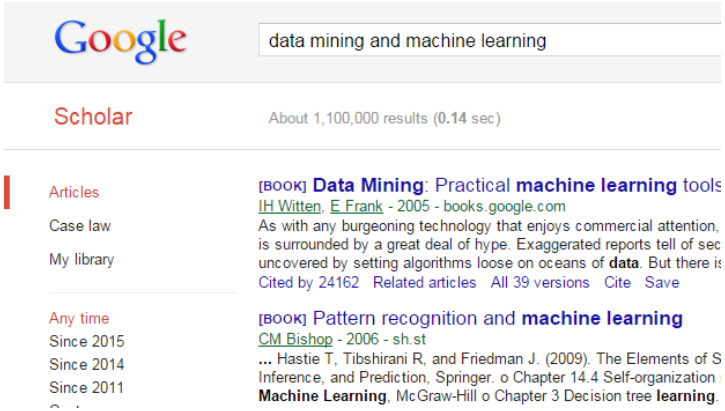
- Basic concepts about clustering
- Clustering algorithms
  - K-means
  - Hierarchical clustering
  - DBSCAN
- Cluster Validity
- Measures of Cluster Validity

What is Cluster Analysis?

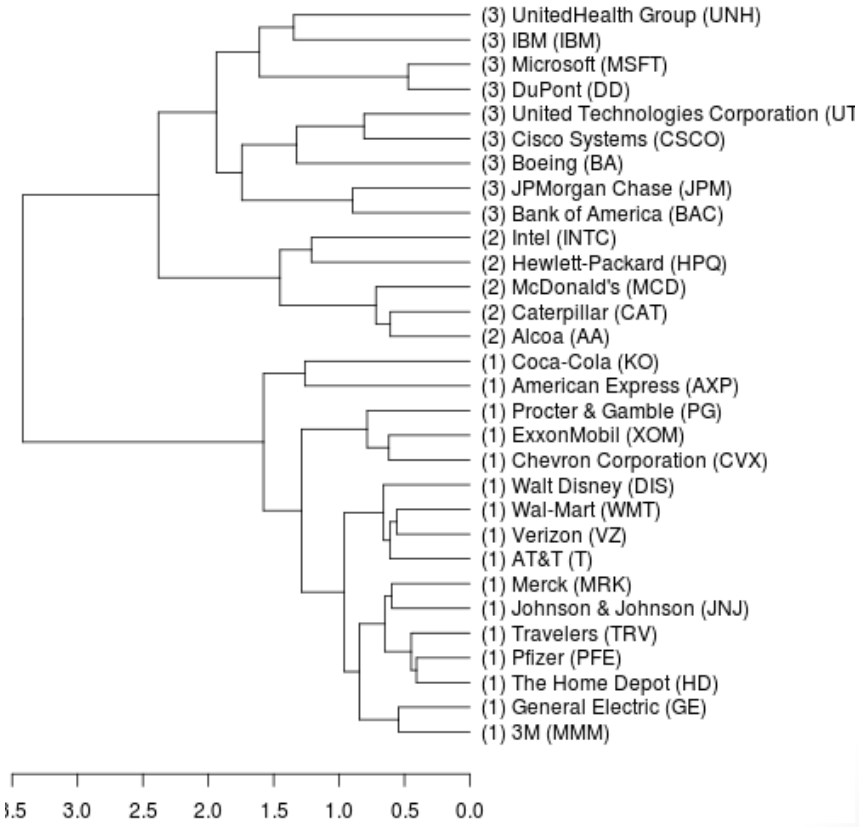
- Finding groups of objects
  - Similar objects are in the same group
  - Different objects are assigned in different groups



Applications of Cluster Analysis: **Understanding**

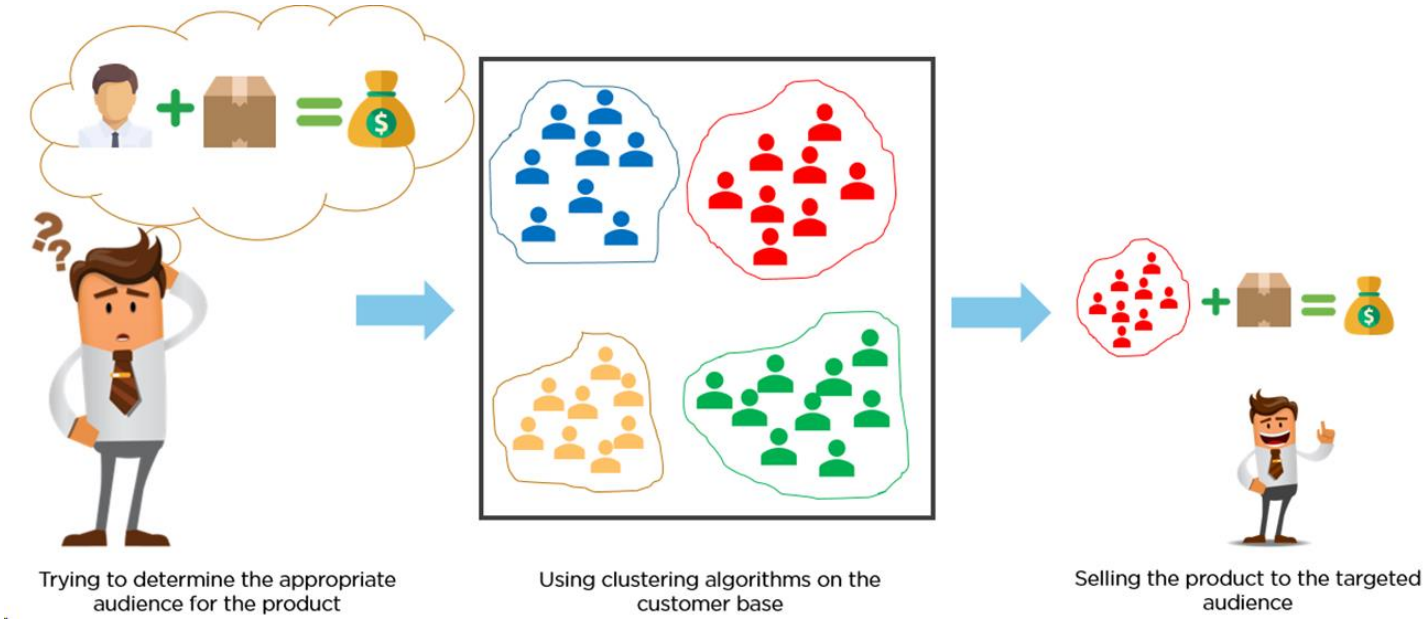


Group related documents



Group stocks

Applications of Cluster Analysis: **Understanding**



https://www.quora.com/What-is-clustering

Applications of Cluster Analysis: **Customer Segmentation – 1/3**

Applications of Cluster Analysis: **Customer Segmentation – 2/3**

Basic Data

- Occupation, age, gender, educational background, Marital status, Number of children

Lifestyle Data

- Hobby, sport, music, artist

Applications of Cluster Analysis: **Customer Segmentation – 3/3**

Basic Data

Lifestyle Data

Preprocessing

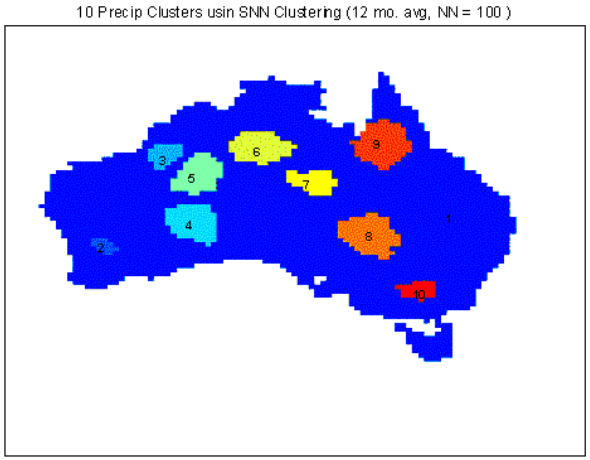
Select target customers and Check their Life Style's data

Use customer's data to select the presenter and create ads based on their life style

Clustering Algorithm

Clustering obtained

Applications of Cluster Analysis: **Summarization**



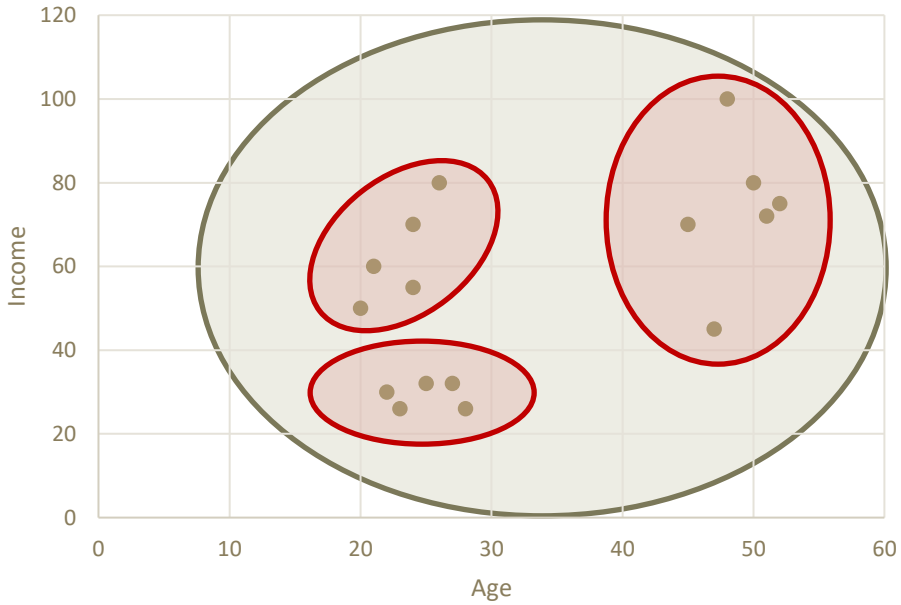
Clustering precipitation in Australia

What is not Cluster Analysis?

- Data query (Database)
- Simple segmentation
  - Dividing students into different groups alphabetically, by last name
- Supervised classification
  - Have class label information

Terminologies

- Cluster: a set of (similar) objects
- Clustering: a set of (dissimilar) clusters

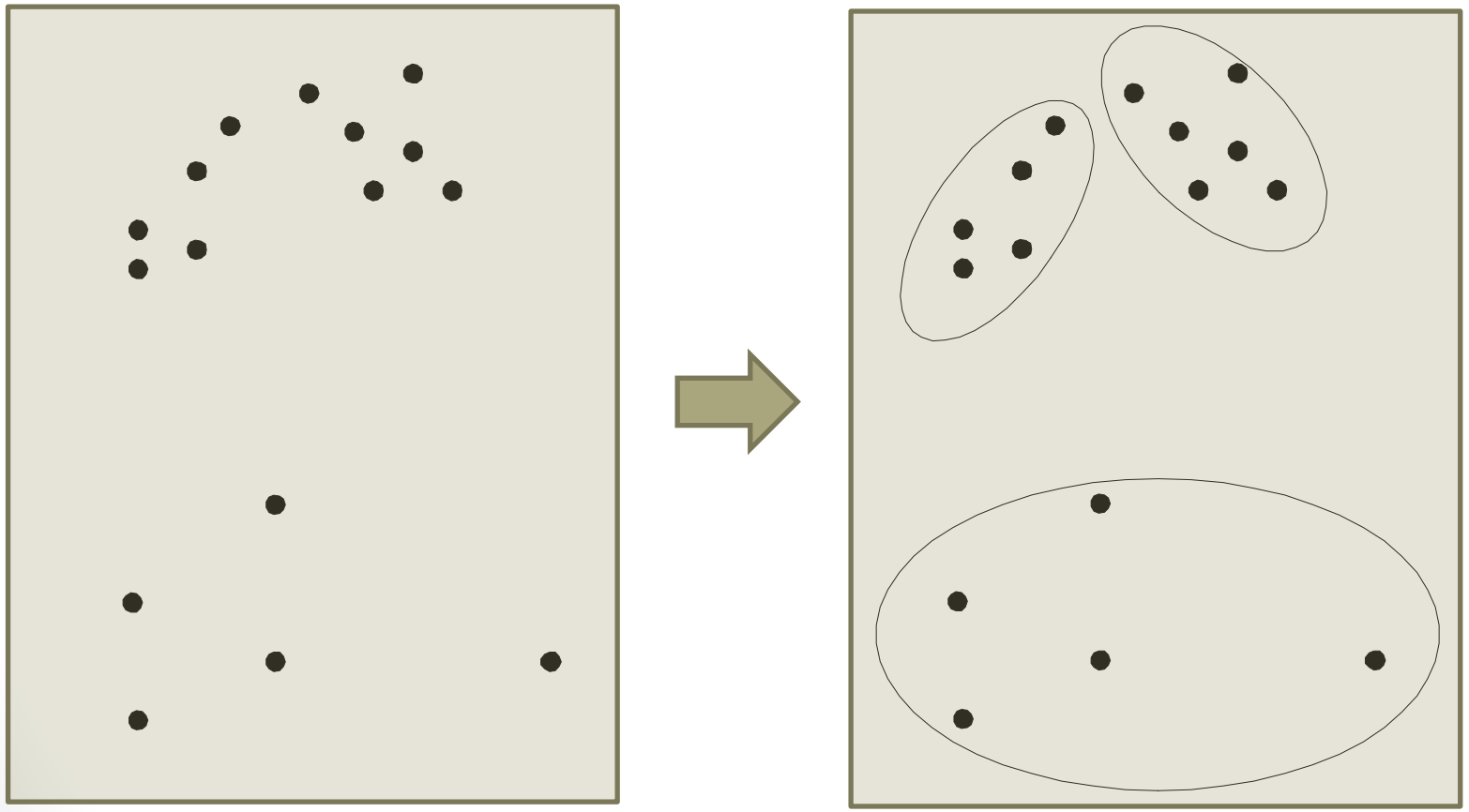


Types of Clusterings

- Partitional clustering
- Hierarchical clustering

Partitional Clustering

- A division data objects into **non-overlapping** subsets (clusters) such that each data object is in **exactly one** subset

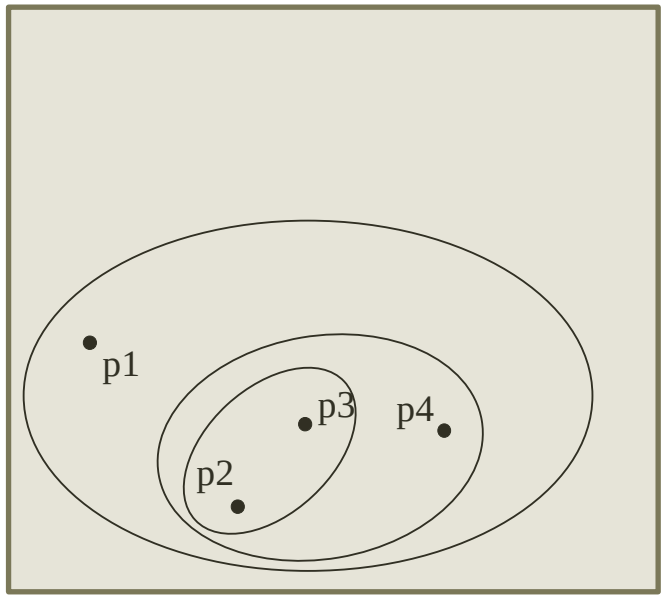


Original Points

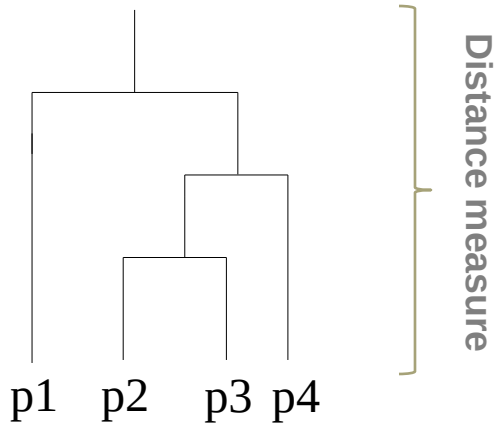
A Partitional Clustering

Hierarchical Clustering

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



Traditional Hierarchical Clustering



Traditional Dendrogram

Other Distinctions Between Sets of Clusters

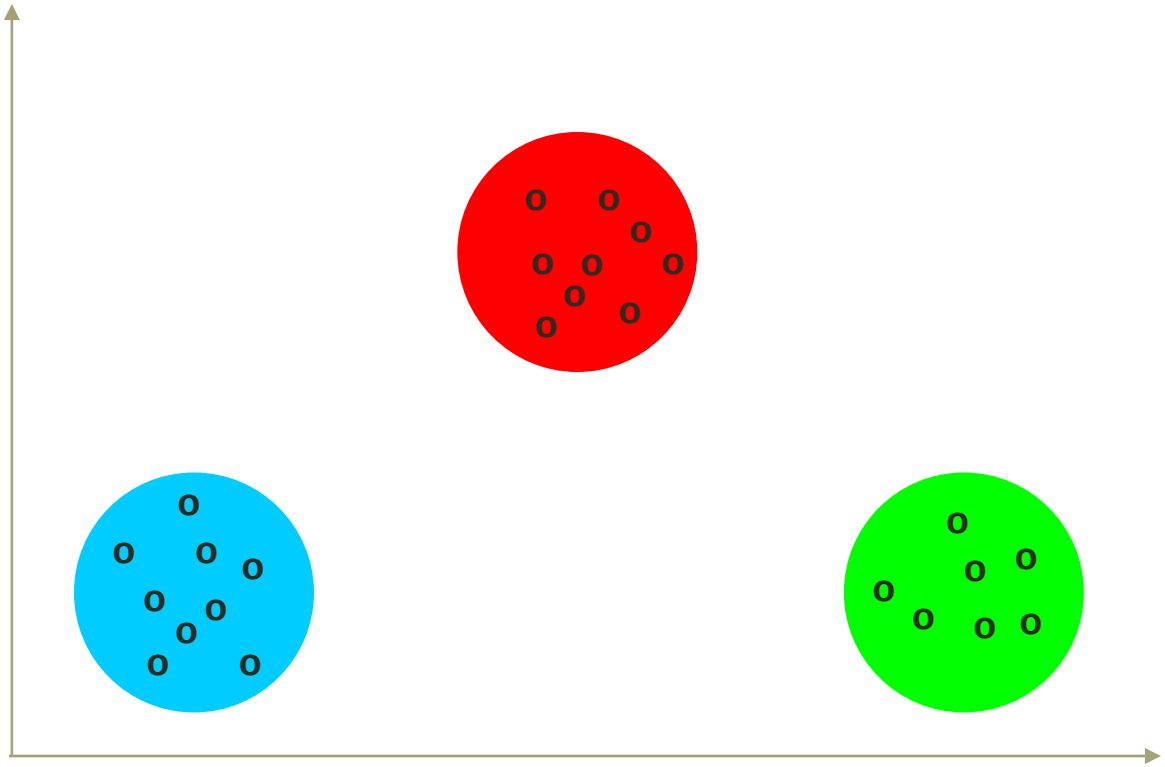
- Exclusive versus non-exclusive
  - Non-exclusive clusterings: points may belong to multiple clusters.
- Partial versus complete
  - Partial clustering: cluster some of the data
- Fuzzy versus non-fuzzy
  - Fuzzy clustering: a point belongs to every cluster with some weight between 0 and 1.
    - Weights must sum to 1.
- Heterogeneous versus homogeneous
  - Heterogeneous clustering: clusters of widely different sizes, shapes, and densities

Typical Types of Clusters (Clustering's Goals)

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters

Types of Clusters: Well-Separated

- Goal: any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.

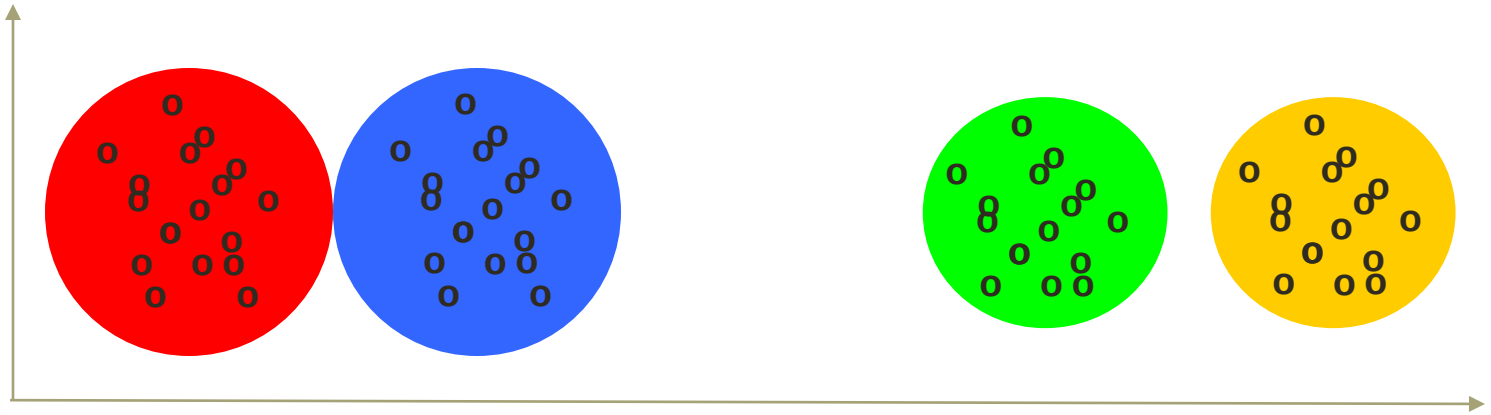


3 well-separated clusters

Types of Clusters: Center-Based

- Goal: any object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster

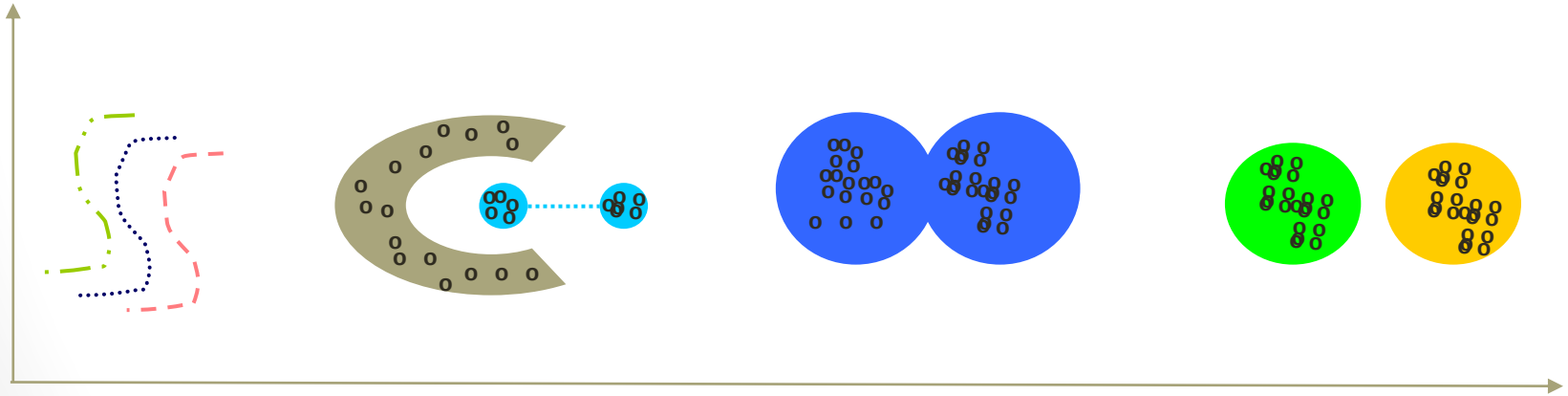
- The center of a cluster:
  - Centroid: the average of all the points in the cluster
  - Medoid: the most "representative" point of a cluster



4 center-based clusters

Types of Clusters: Contiguity-Based (Nearest neighbor or Transitive)

- Goal: any point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point in a different cluster.

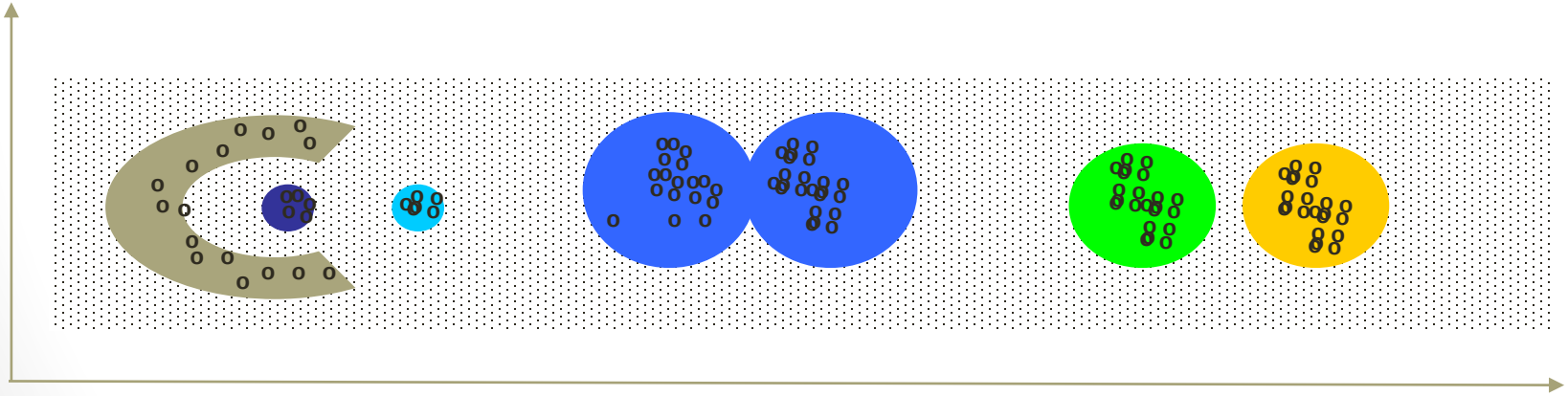


8 contiguous clusters

Types of Clusters: Density-Based

- Goal: a dense region of points that is surrounded by a region of low-density region.

- Used when
  - the clusters are irregular or intertwined
  - noise and outliers are present



6 density-based clusters

Clustering Algorithms

- K-means and its variants (e.g. K-medoid and bisect K-means)
- Hierarchical clustering
- Density-based clustering

K-means Clustering: Basic Concept

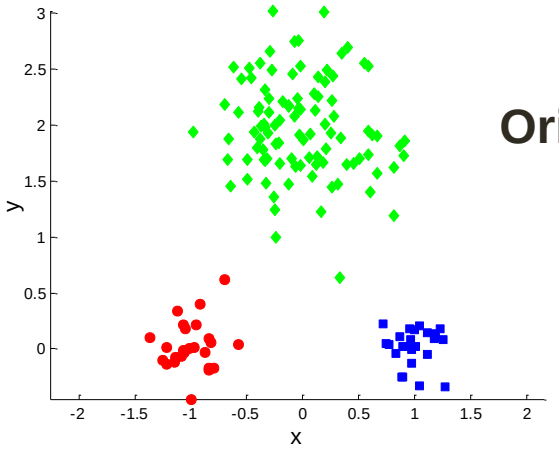
- Partitional clustering approach
- Each cluster is associated with a centroid (center point).
- Each point is assigned to the cluster with the closest centroid.
- Number of clusters (*K*) must be specified.

K-means Clustering: Algorithm

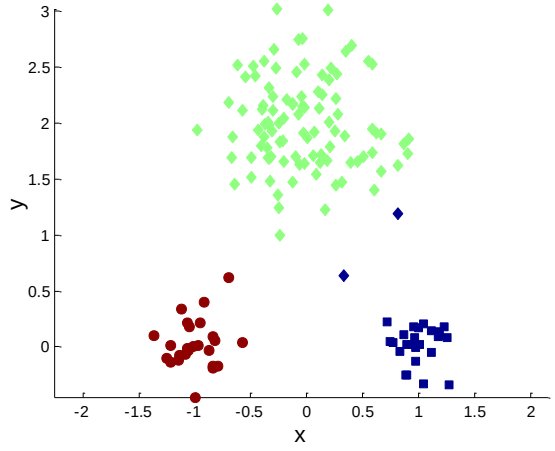
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

Euclidean Distance: $$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

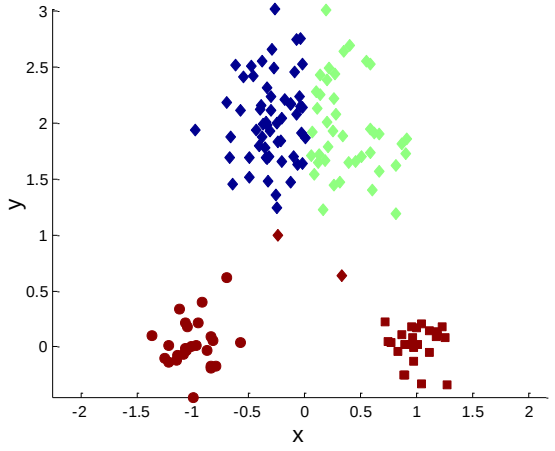
Two different K-means Clusterings



Original Points



Optimal Clustering



Sub-optimal Clustering

Evaluating K-means Clusters: **Sum of Squared Error (SSE)**

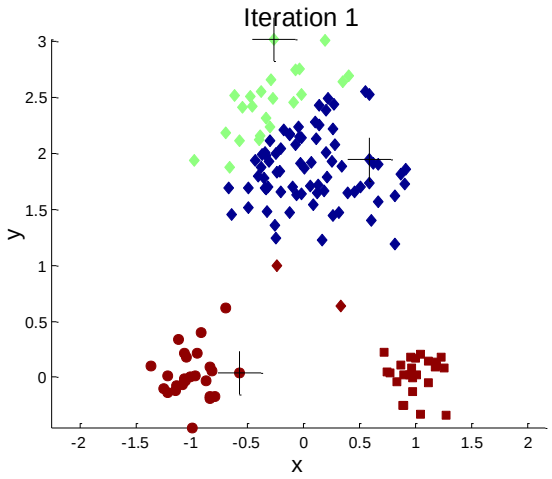
- For each point, the error is the distance to the nearest cluster (cluster's representative point).

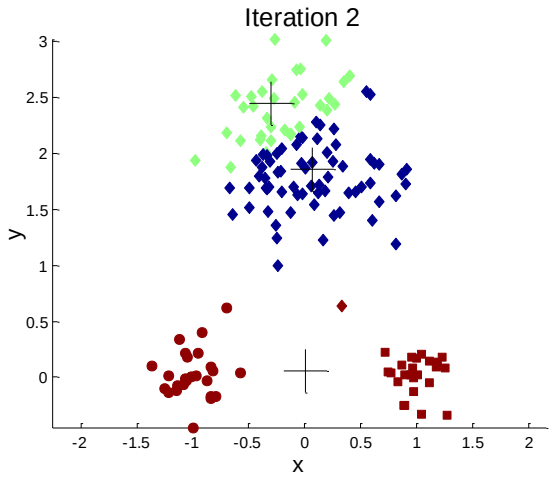
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(m_i, x)^2$$

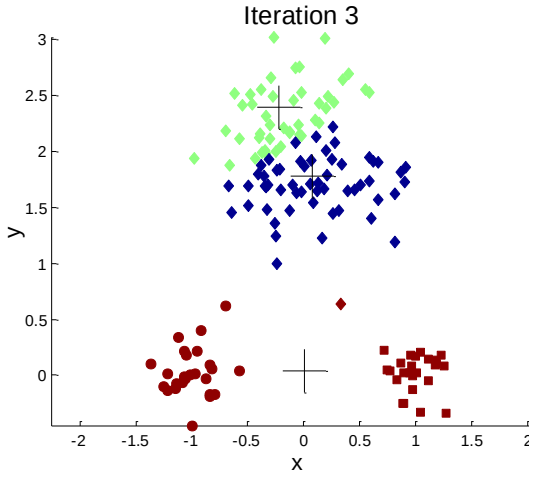
  - x is a data point in cluster C_i
  - m_i is the representative point for cluster C_i

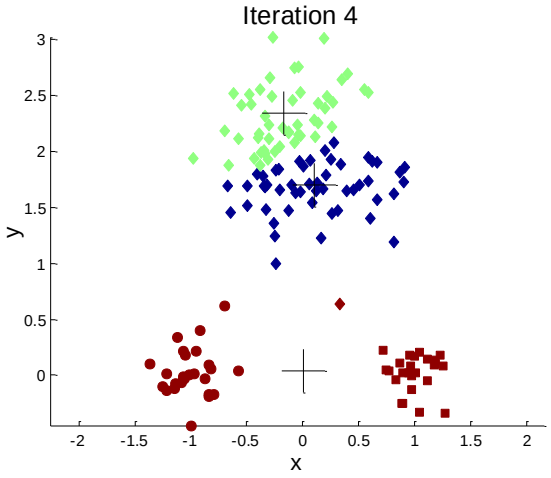
- If we have 2 or more clusterings generated, select the one with the **smallest SSE**.

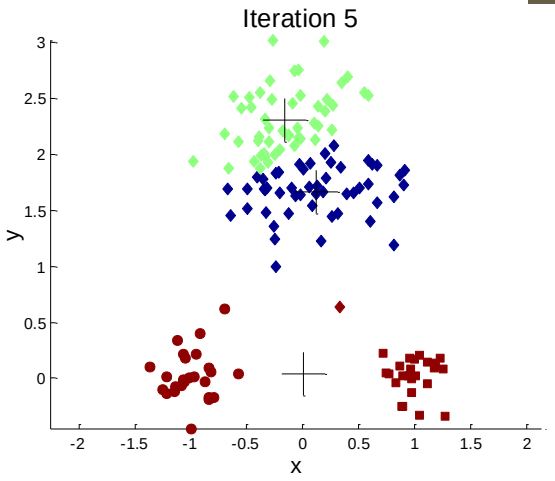
Importance of Choosing Initial Centroids











Problems with Selecting Initial Points

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - If clusters are the same size, n, then

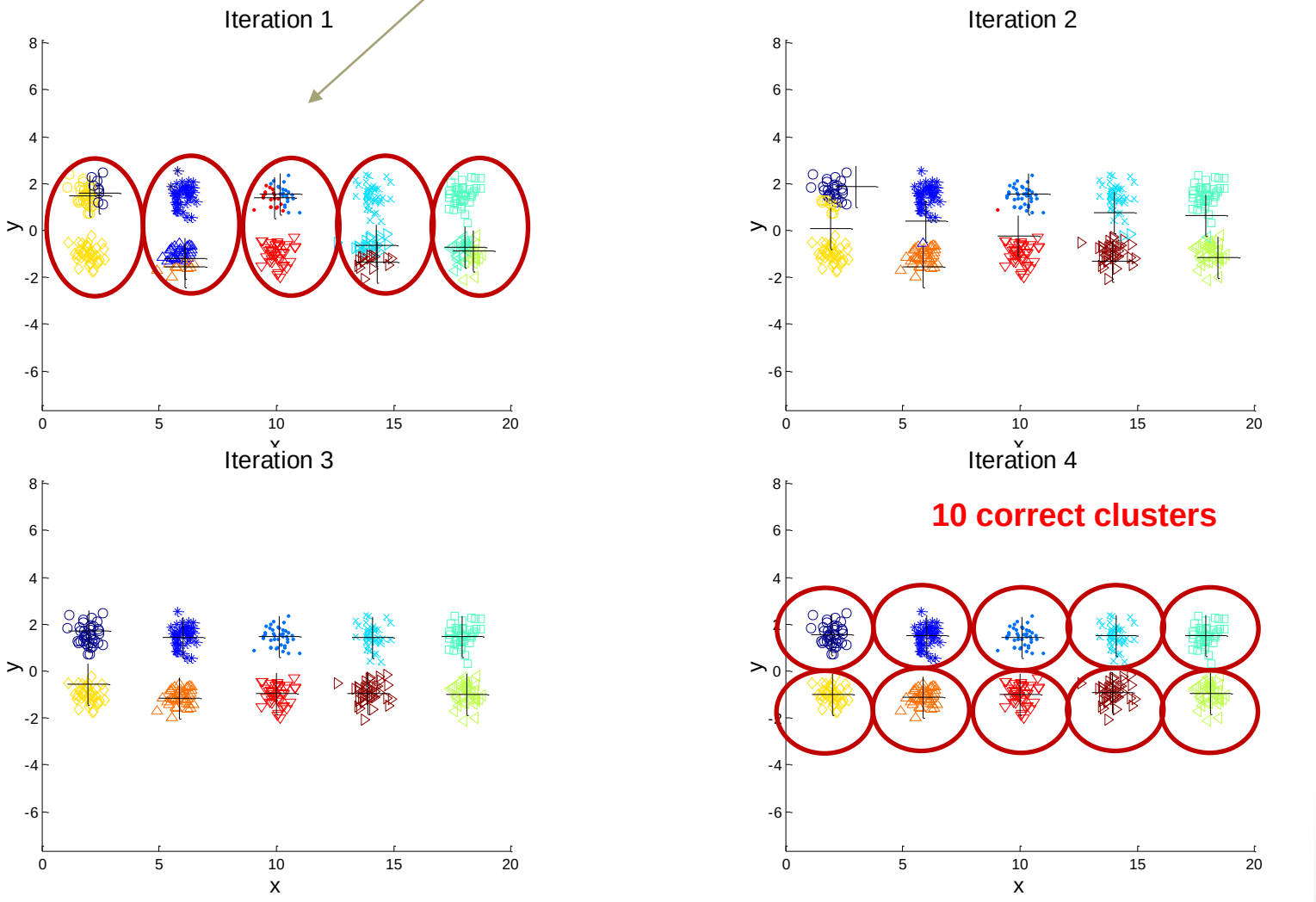
$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K! n^K}{(Kn)^K} = \frac{K!}{K^K}$$

  - For example,

    if K = 10, then probability = $10!/10^{10} = 0.00036$

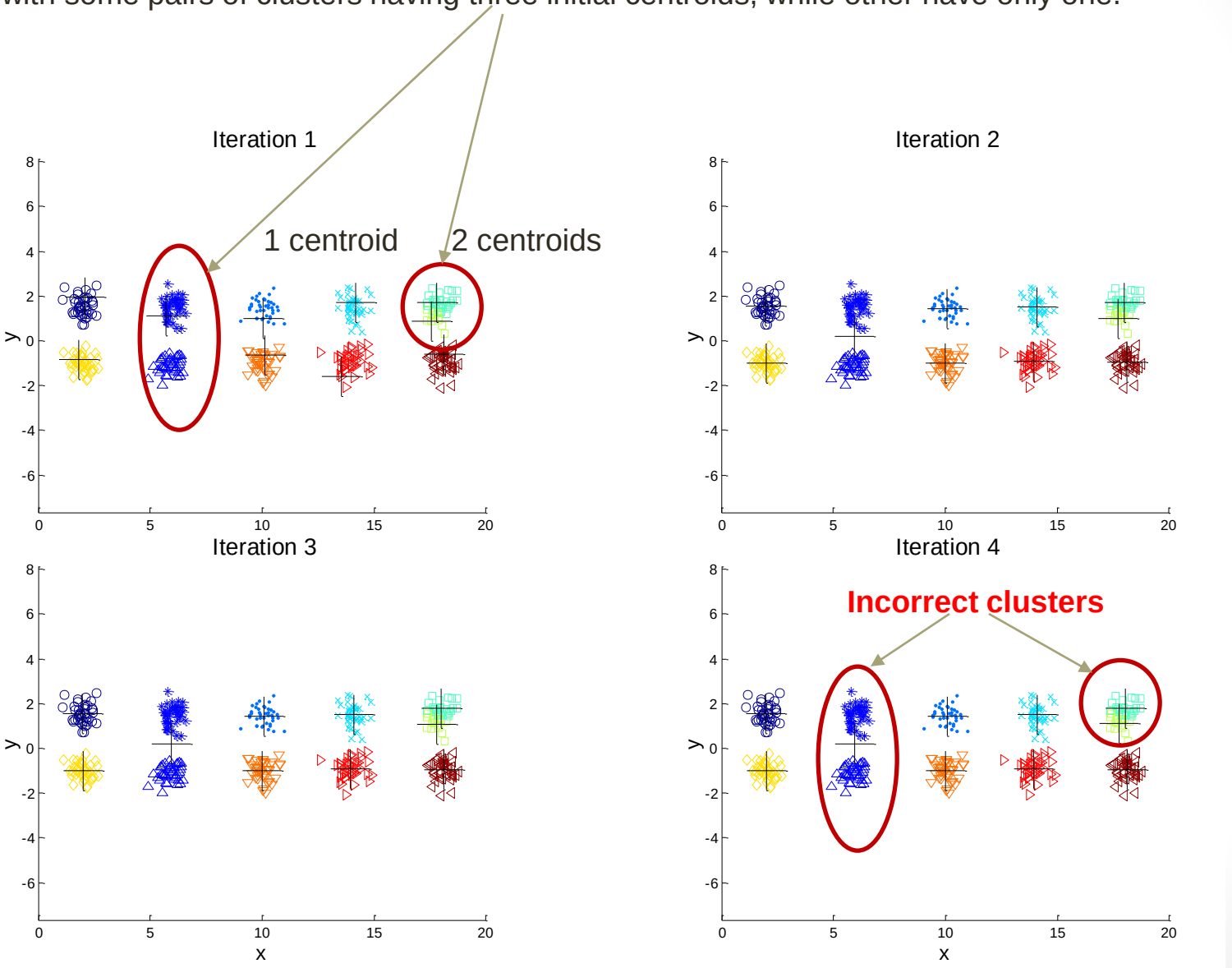
Example 1: Generating Ten Clusters

Starting with two initial centroids in one cluster of each pair of clusters



Example 2: Generating Ten Clusters

Starting with some pairs of clusters having three initial centroids, while other have only one.



Solutions to Initial Centroids Problem

- Soln. 1: Generate multiple results and select the best one.
- Soln. 2: Sample and use hierarchical clustering to determine initial centroids
- Soln. 3: Select more than k initial centroids and then select among these initial centroids
  - Select the farthest points from any of the initial centroids
- Soln. 4: Bisecting K-means
  - Not as susceptible to initialization issues
- And more

Another Problem of K-means: Empty Clusters

- Basic K-means algorithm can yield empty clusters
- Several strategies to handle empty clusters:
  - If there exists 1 or more empty cluster(s), then they will be replaced using the following strategies.
  - Strategy 1: Choose the point that is farthest away from any current centroid.
  - Strategy 1: Choose a point from the cluster with the highest SSE.
  - If there are several empty clusters, the above can be repeated several times.

Feasible Pre-processing and Post-processing Techniques Applied with K-means to Improve the Clustering Result

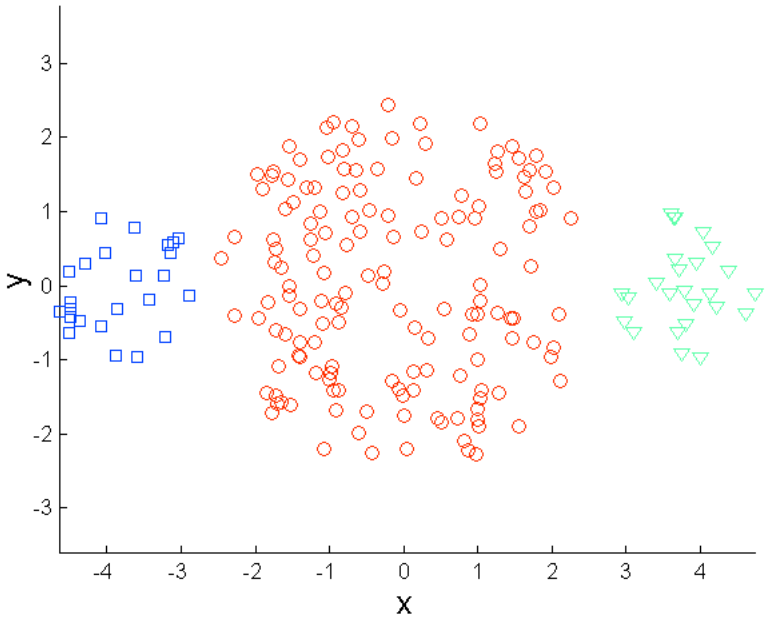
- Pre-processing
  - Normalize or standardize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE
  - Can use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

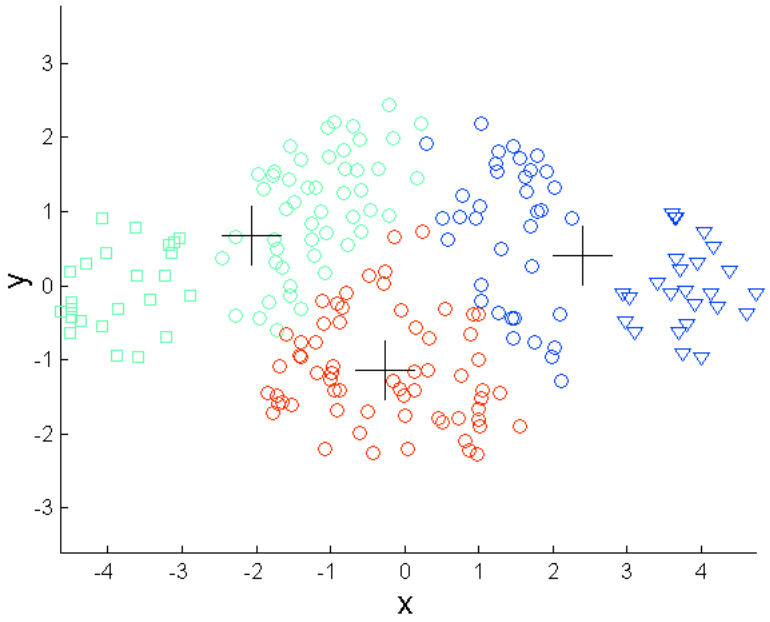
Limitations of K-means

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

- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes





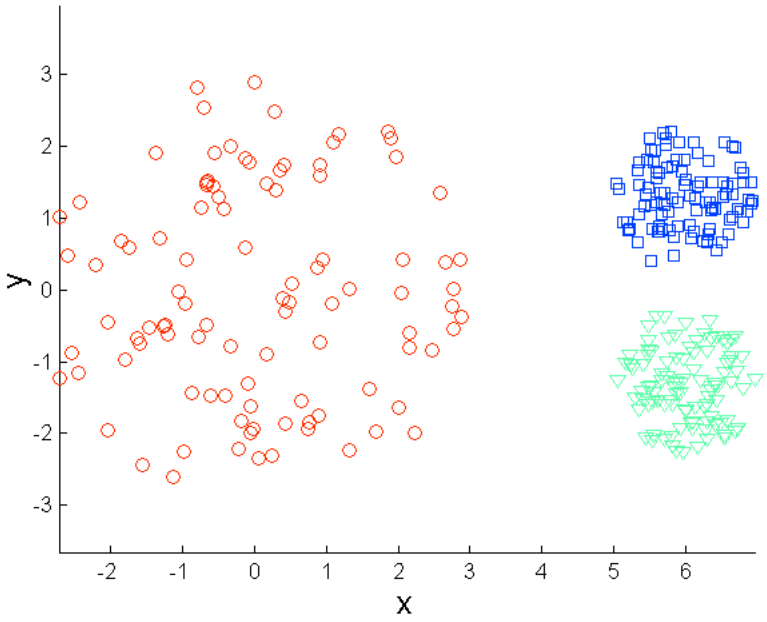
Original Points

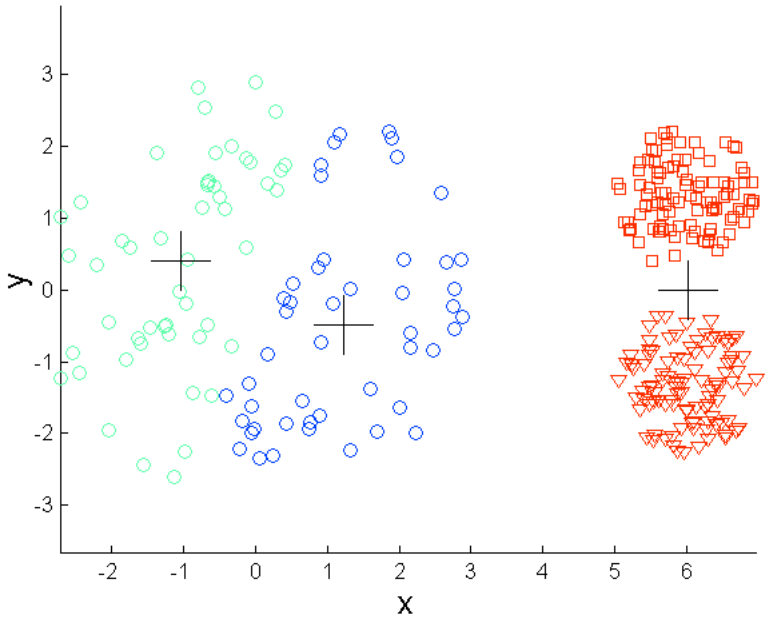
Expected no. of clusters: 3
Note: these groups of 2D-data points generated by 3 normal (or Gaussian) distributions – *data points' color illustrates different desirable clusters*

K-means (3 Clusters)

Result: generates 3 clusters but not correct as expected – *data points' color illustrates different desirable clusters*

Limitations of K-means: Differing Density





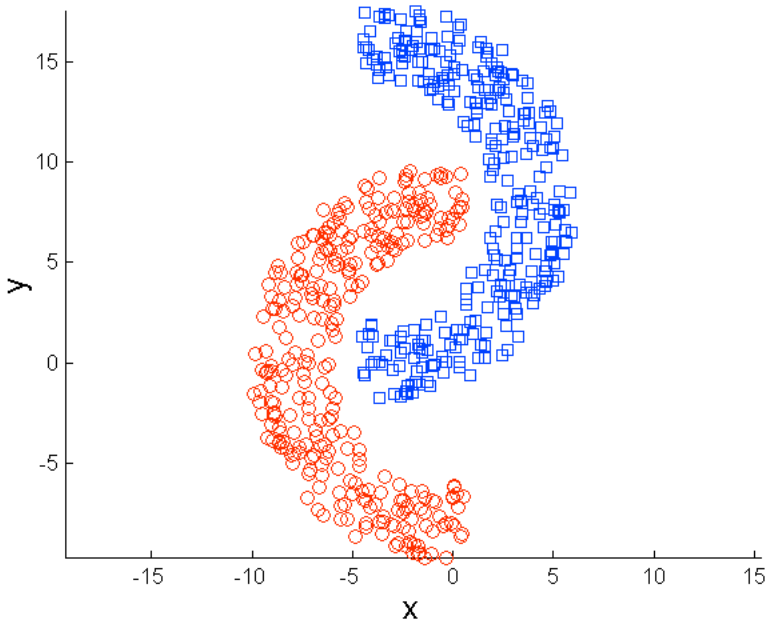
Original Points

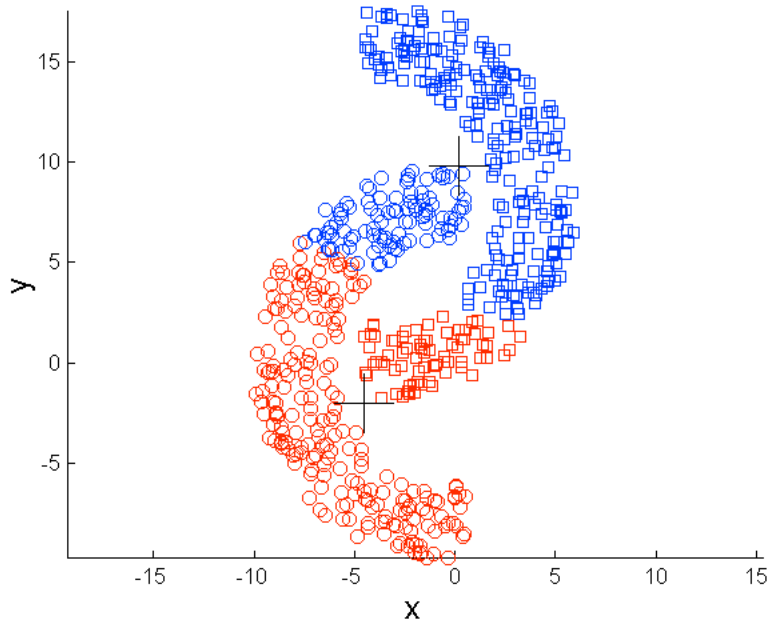
Expected no. of clusters: 3
Note: these groups of 2D-data points generated by 3 normal (or Gaussian) distributions – *data points' color illustrates different desirable clusters*

K-means (3 Clusters)

Result: generates 3 clusters but not correct as expected – *data points' color illustrates different desirable clusters*

Limitations of K-means: Non-globular Shapes





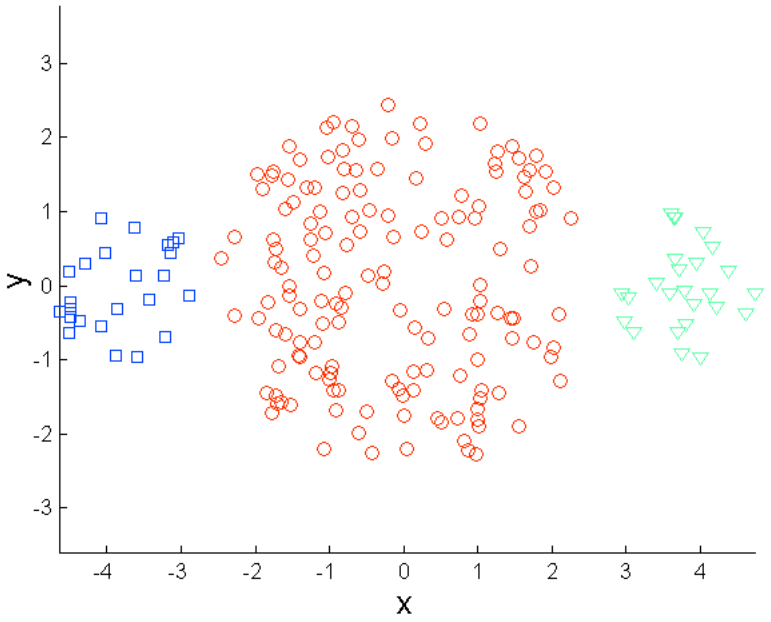
Original Points

Expected no. of clusters: 2
Note: the figure depicts 2 arbitrary-shaped clusters of 2D-data points – *data points' color illustrates different desirable clusters*

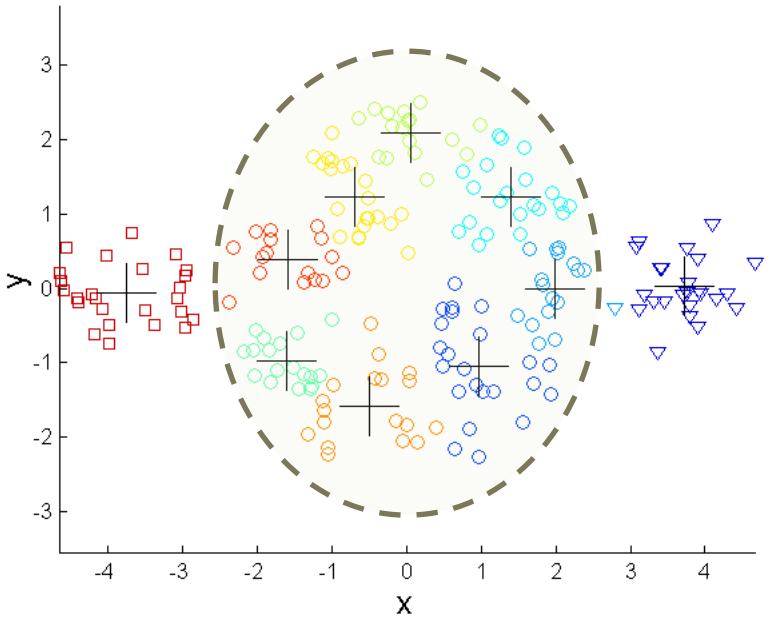
K-means (2 Clusters)

Result: generates 2 clusters but not correct as expected – *data points' color illustrates different desirable clusters*

Overcoming K-means Limitation on Differing Sizes



Original Points

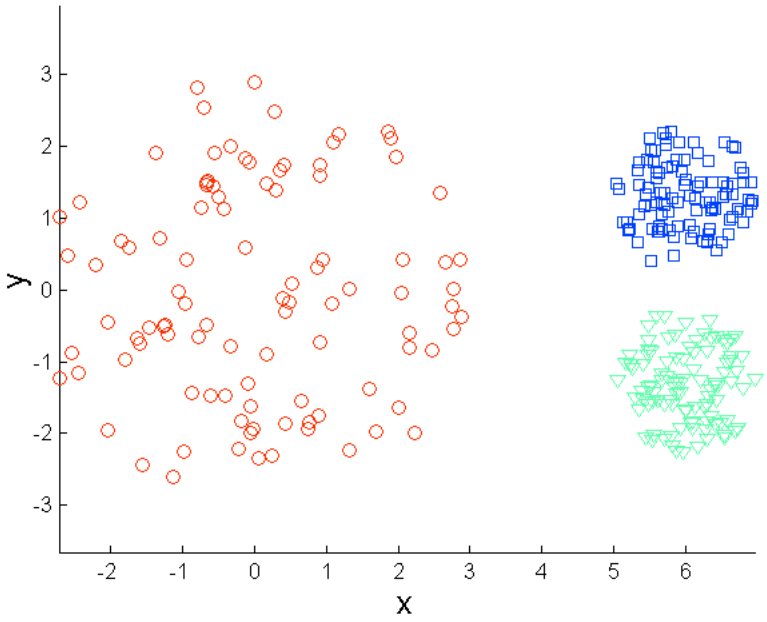


K-means Clusters

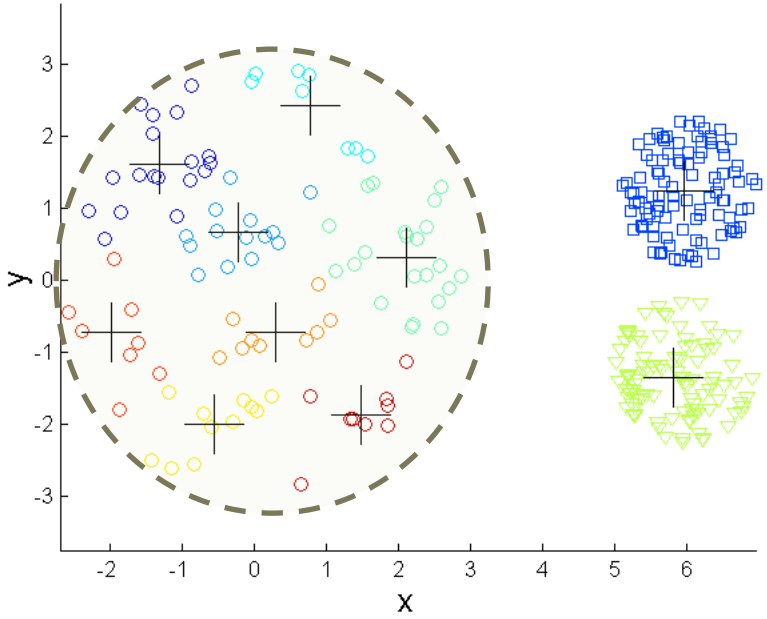
One solution is to generate many small clusters (> k clusters).

Then, agglomerate them.

Overcoming K-means Limitations on Differing Density



Original Points

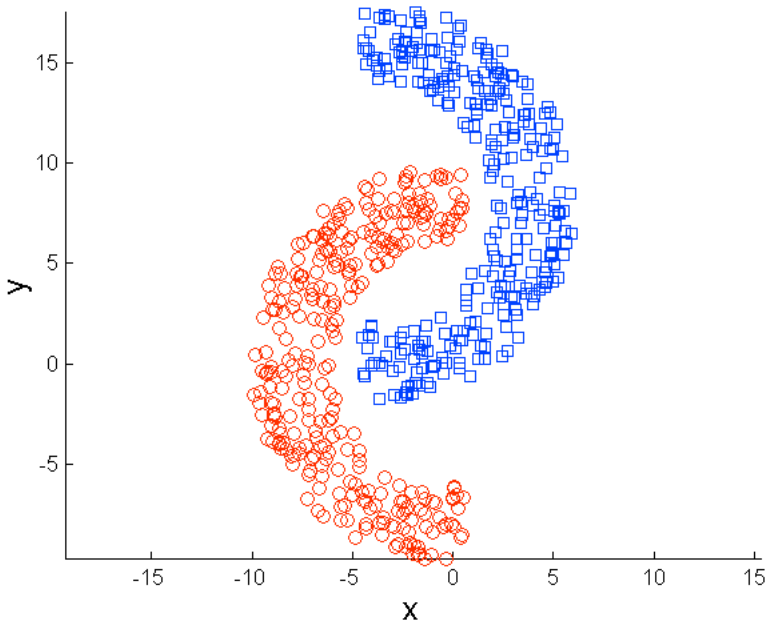


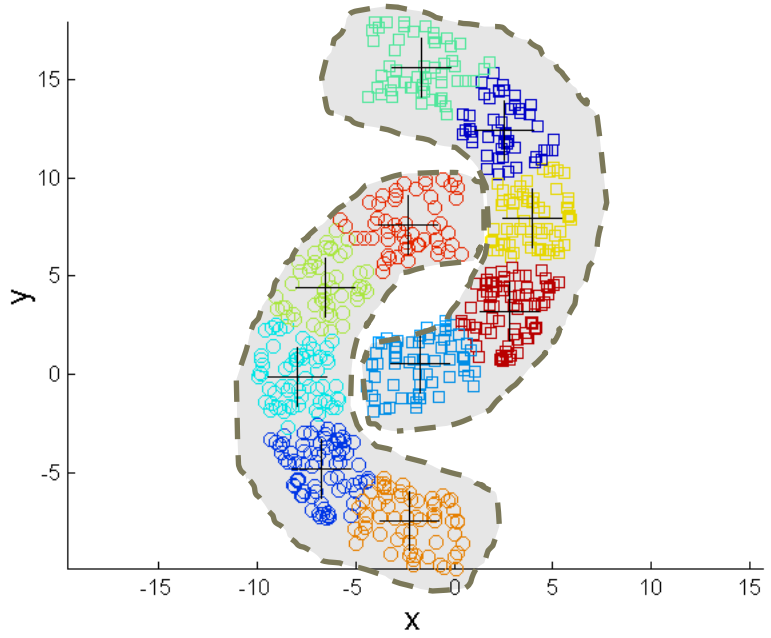
K-means Clusters

One solution is to generate many small clusters (> k clusters).

Then, agglomerate them.

Overcoming K-means Limitations on Non-globular Shapes





Original Points **K-means Clusters**

One solution is to generate many small clusters (> k clusters).

Then, agglomerate them.

Summary on K-means clustering - 1

- Initial centroids are often chosen randomly.
- Nondeterministic algorithm: clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- 'Closeness' measure is varied in its variation, e.g., Euclidean distance, cosine similarity, correlation, etc.

Summary on K-means clustering - 2

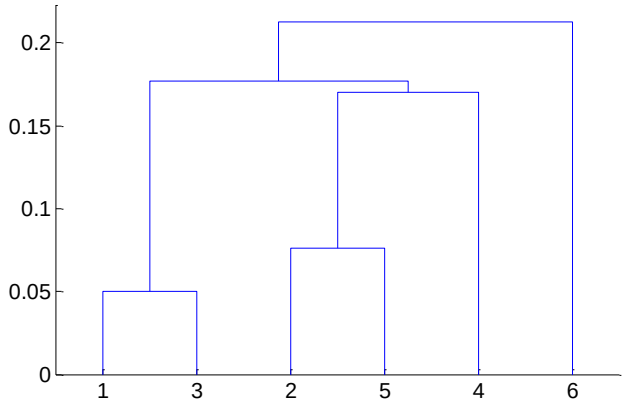
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to '**until** relatively few points change clusters'
- Complexity is O(n * K * I * d)
  - n = number of points
  - K = number of clusters
  - I = number of iterations
  - d = number of attributes

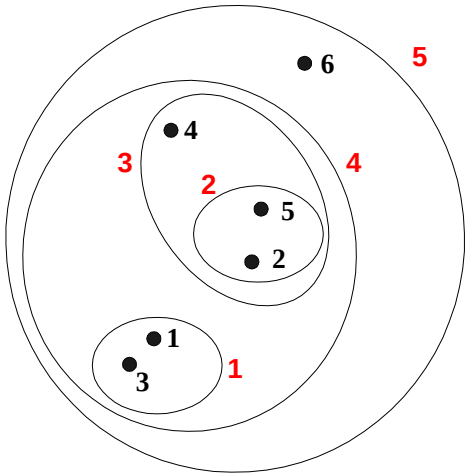
Outlines

- Basic concepts about clustering
- Clustering algorithms
  - K-means
  - Hierarchical clustering
  - DBSCAN
- Cluster Validity
- Measures of Cluster Validity

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

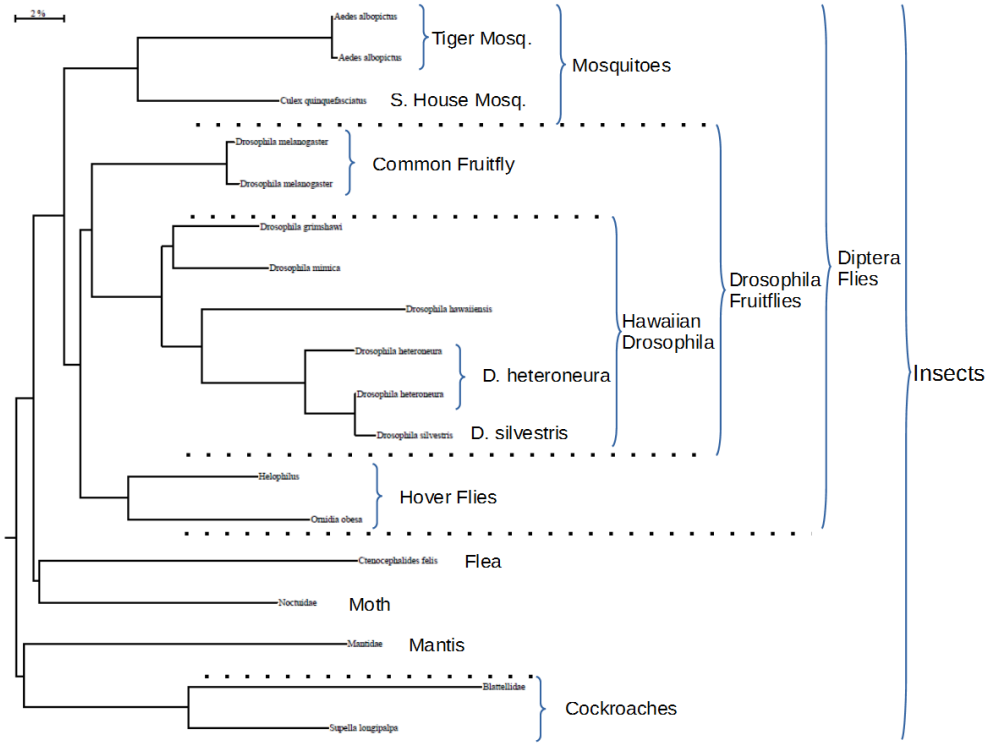




Black digits: 2D-data points
Red digits: IDs of cluster generated

Strengths of Hierarchical Clustering

- No pre-determined number of clusters required
- Result may correspond to meaningful taxonomies



Clustering insect species

Hierarchical Clustering

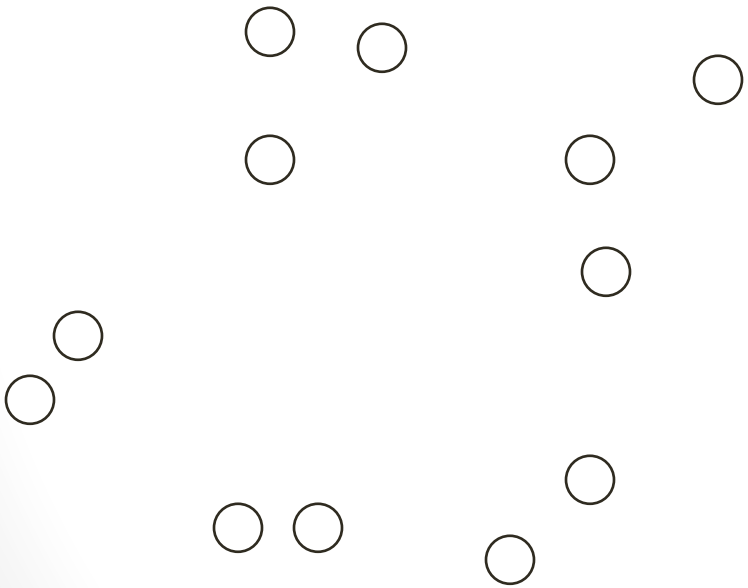
- Two main types of hierarchical clustering
  - Agglomerative
  - Divisive
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

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

Note: Different approaches define different proximity (distance or similarity) between clusters (distinguishing the different algorithms)

Starting Situation



	p1	p2	p3	p4	p5	. . .	p12
p1							
p2							
p3							
p4							
p5							
.							
.							
.							
p12							

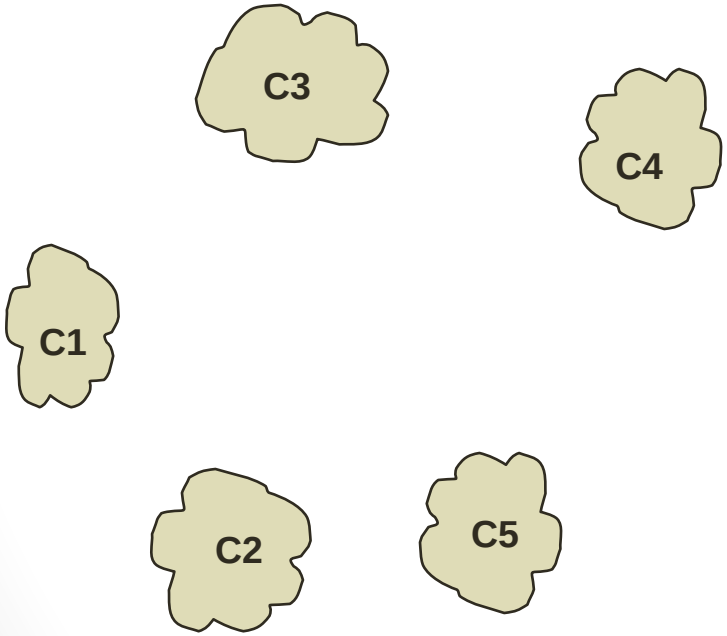
Proximity Matrix

(each cell contains similarity/distance between 2 data points)

Singleton cluster (One point per cluster)

Intermediate Situation

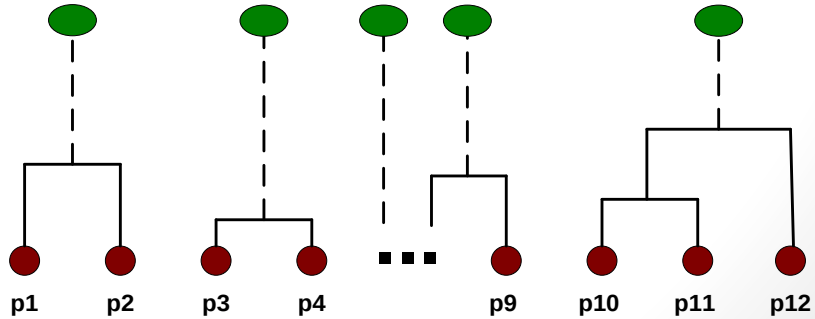
(after repeatedly merging until obtaining 5 clusters)



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix

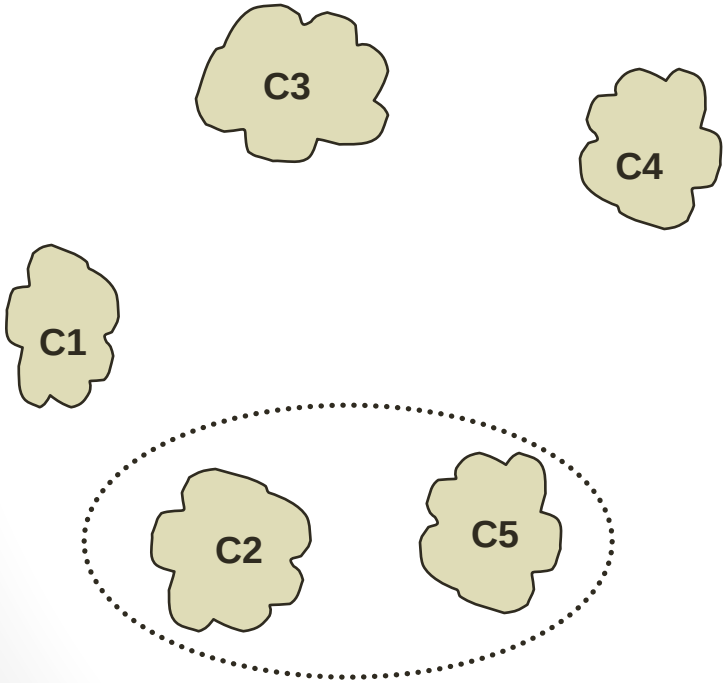
(each cell contains similarity/distance between 2 data points)



5 clusters (each cluster contains some data points.)

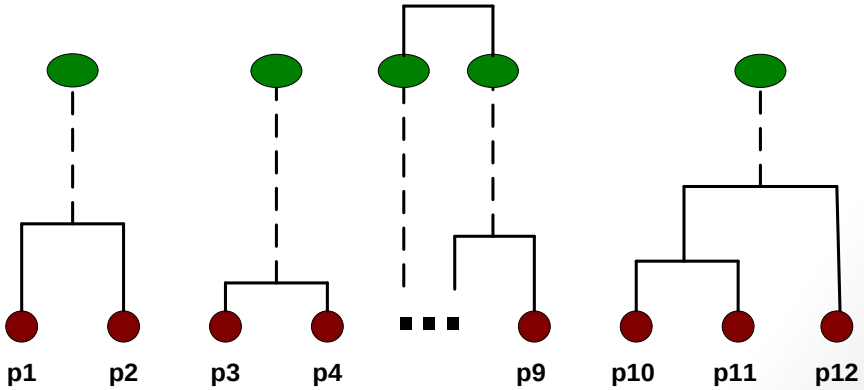
Intermediate Situation

(If merging c2 and c5, it will update the proximity matrix – *result in the next slide*)



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

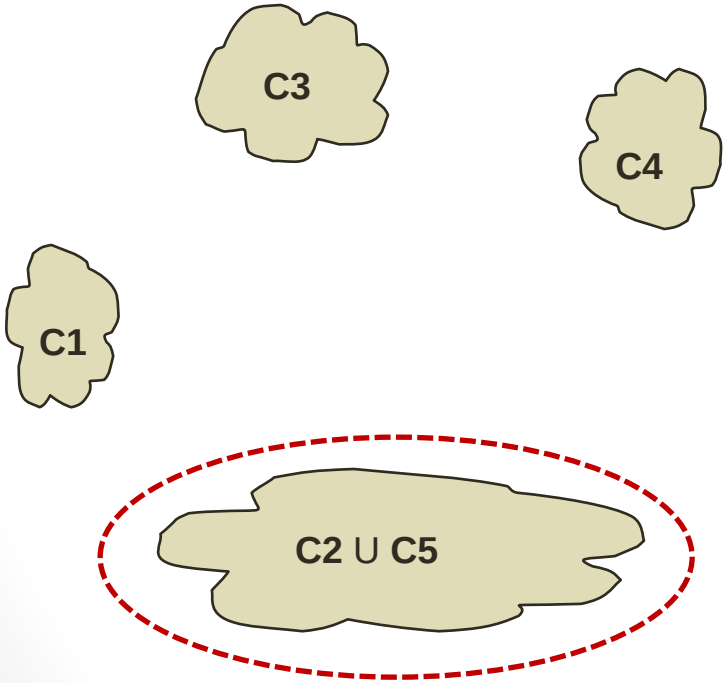
Proximity Matrix



After Merging

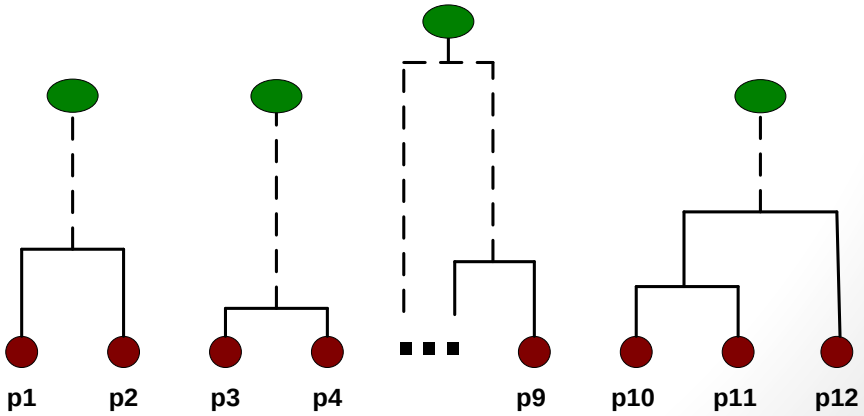
- Question: “How do we update the proximity matrix?”

(values in ‘?’ Cells)



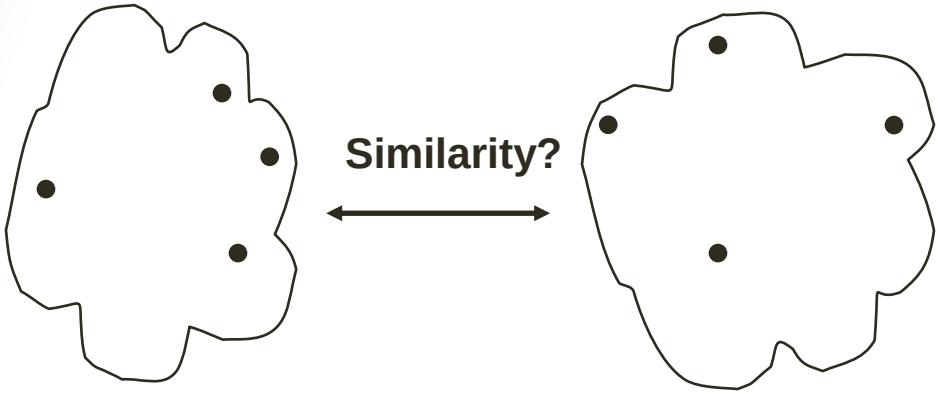
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

(To update values in '?' cells in the proximity matrix after merging 2 clusters)

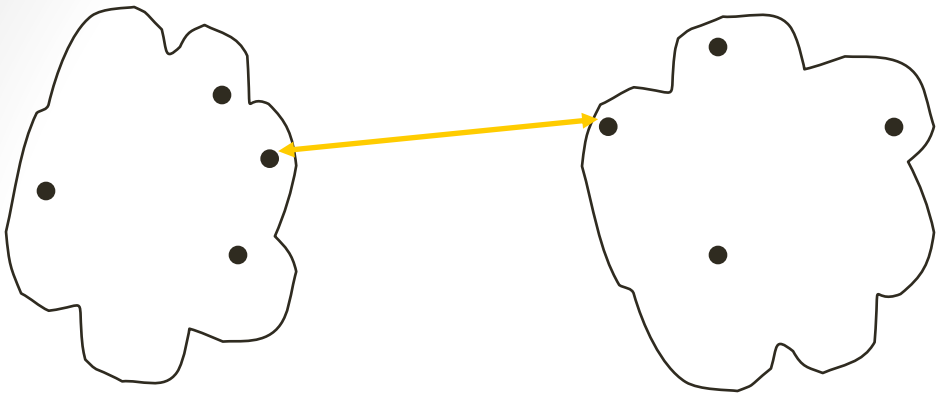


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- Frequently use techniques for updates
  - MIN
  - MAX
  - Group Average
  - Distance Between Centroids

How to Define Inter-Cluster Similarity

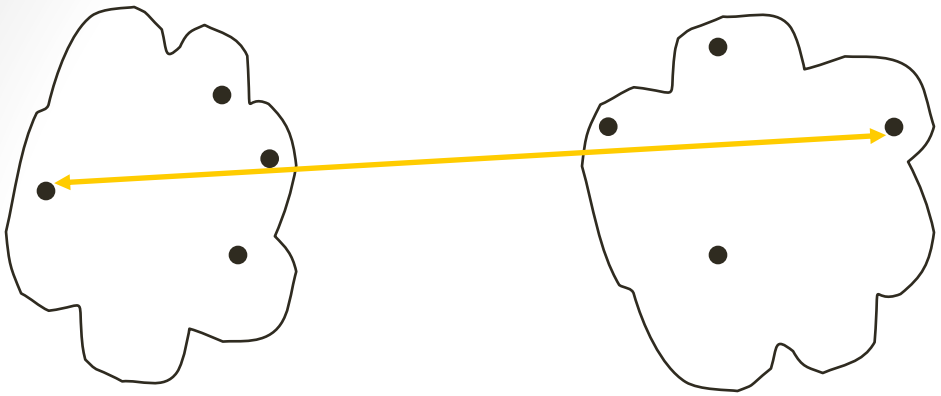


- MIN
- MAX
- Group Average
- Distance Between Centroids

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

Proximity Matrix

How to Define Inter-Cluster Similarity

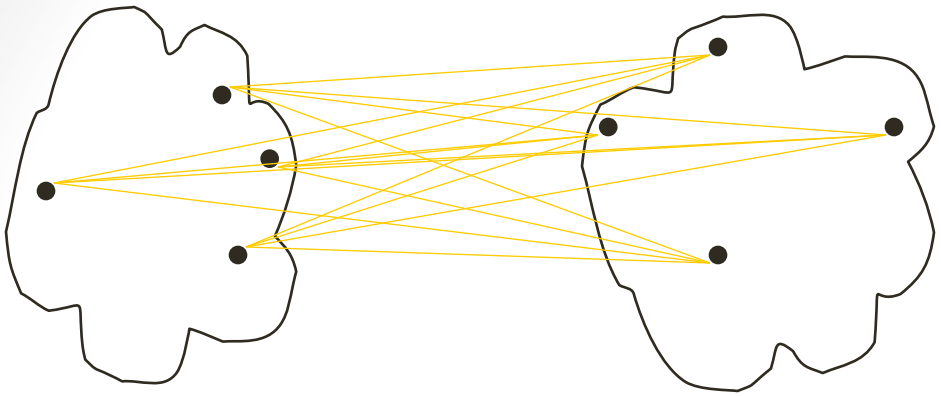


- MIN
- MAX
- Group Average
- Distance Between Centroids

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

Proximity Matrix

How to Define Inter-Cluster Similarity

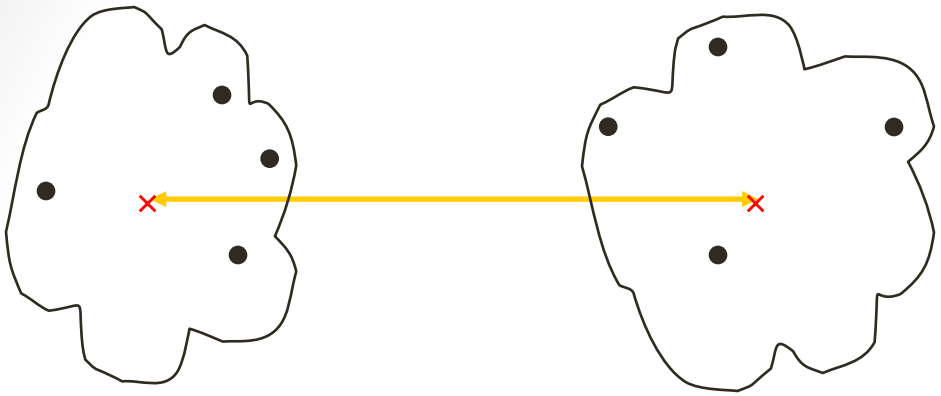


- MIN
- MAX
- Group Average
- Distance Between Centroids

	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

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

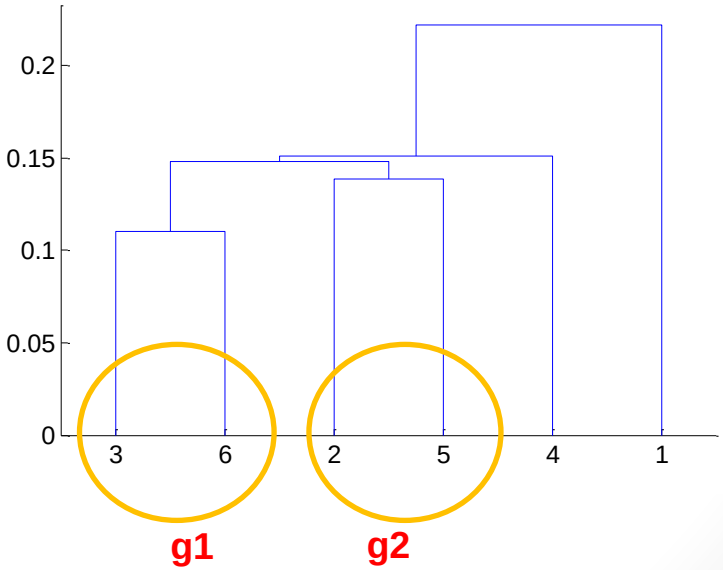
Proximity Matrix

Inter-Cluster Similarity: MIN or Single Link

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters.

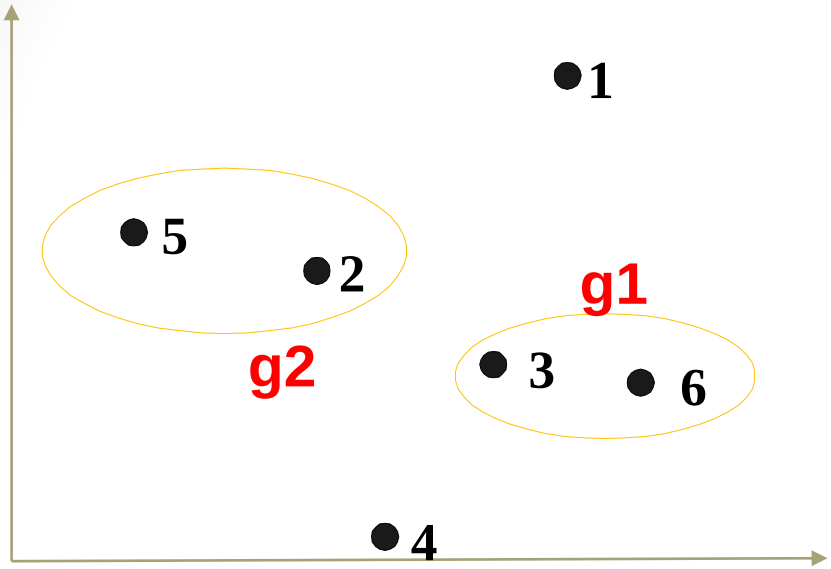
	I1	I2	I3	I4	I5	I6
I1	0.00	**0.24**	0.22	0.37	0.34	0.23
I2	0.24	0.00	0.15	0.20	0.14	0.25
I3	0.22	0.15	0.00	0.15	0.28	0.11
I4	0.37	0.20	0.15	0.00	0.29	0.22
I5	0.34	0.14	0.28	0.29	0.00	0.39
I6	0.23	0.25	0.11	0.22	0.39	0.00

Euclidean distance (dissimlarity) matrix for 6 points



Hierarchical Clustering: MIN

Example: Calculating distance of g1 and g2 (to be merged)



	I1	I2	I3	I4	I5	I6
I1	0.00	**0.24**	0.22	0.37	0.34	0.23
I2	0.24	0.00	0.15	0.20	0.14	0.25
I3	0.22	0.15	0.00	0.15	0.28	0.11
I4	0.37	0.20	0.15	0.00	0.29	0.22
I5	0.34	0.14	0.28	0.29	0.00	0.39
I6	0.23	0.25	0.11	0.22	0.39	0.00

dist(g1,g2) = dist({3,6}, {2,5})

= min(dist(3,2), dist(6,2), dist(3,5), dist(6,5))

= min(0.15, 0.25, 0.28, 0.39)

=0.15

dist(g1,4) = dist({3,6}, {4})

= min(dist(3,4), dist(6,4))

= min(0.15, 0.22)

=0.15

dist(g1, 1) = dist({3,6}, {1})

= min(dist(3,1), dist(6,1))

= min(0.22, 0.23)

=0.22

dist(g2, 4) = dist({5,2}, {4})

= min(dist(5,4), dist(2,4))

= min(0.29, 0.20)

=0.20

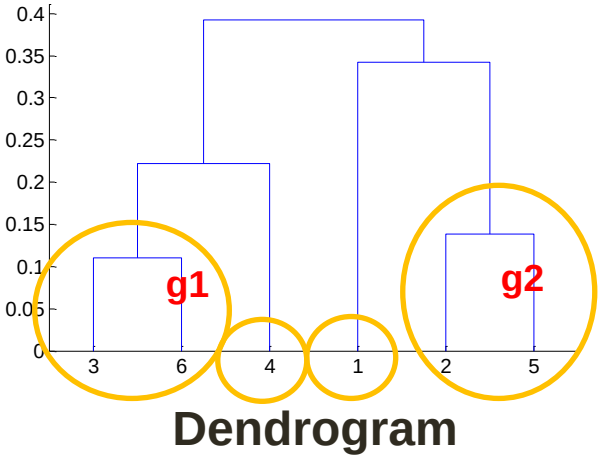
dist(g2, 1) = dist({5,2}, {1}) = 0.2

Inter-Cluster Similarity: MAX or Complete Linkage

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

	I1	I2	I3	I4	I5	I6
I1	0.00	**0.24**	0.22	0.37	0.34	0.23
I2	0.24	0.00	0.15	0.20	0.14	0.25
I3	0.22	0.15	0.00	0.15	0.28	0.11
I4	0.37	0.20	0.15	0.00	0.29	0.22
I5	0.34	0.14	0.28	0.29	0.00	0.39
I6	0.23	0.25	0.11	0.22	0.39	0.00

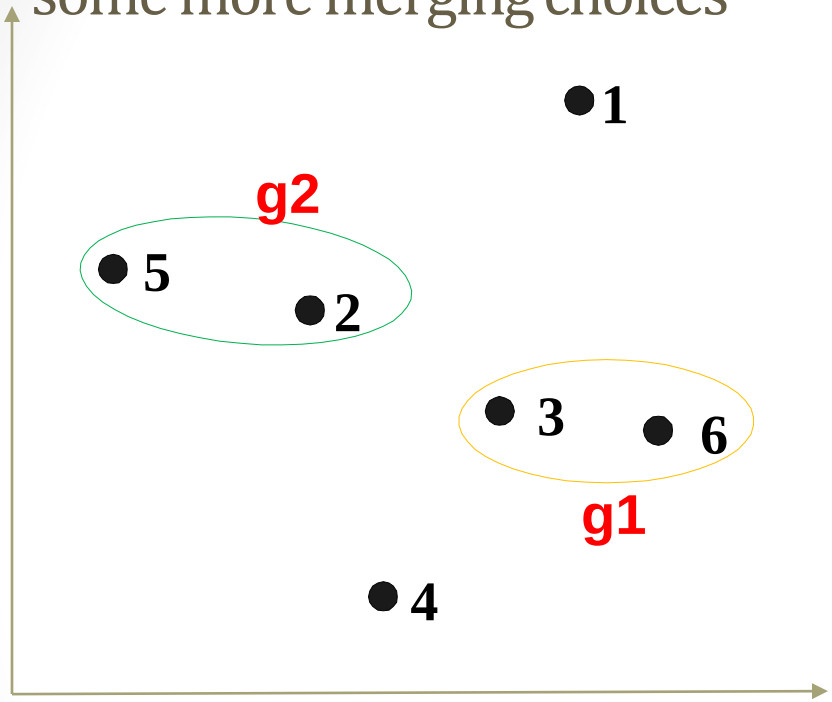
Euclidean distance (dissimlarity) matrix for 6 points



Dendrogram

Hierarchical Clustering: MAX

Example: Calculating distance of g1 and g2 (to be merged) and some more merging choices



dist(g1, g2) = dist({3,6}, {2,5})
= max(dist(3,2), dist(6,2),
dist(3,5), dist(6,5))
= max(0.15, 0.25, 0.28, 0.39)
=0.39

dist(g1, 4) = dist({3,6}, {4})
= max(dist(3,4), dist(6,4))
= max(0.15, 0.22)
= 0.22

dist(g1, 1) = dist({3,6},{1})
= max(dist(3,1), dist(6,1))
= max(0.22, 0.23)
= 0.23

dist(g2, 4) = dist({5,2}, {4})
= max(dist(5,4), dist(2,4))
= max(0.29, 0.2)
= 0.29

dist(g2, 1) = dist({5,2},{1})
= max(dist(5,1), dist(2,1))
= max(0.34, 0.02)
= 0.34

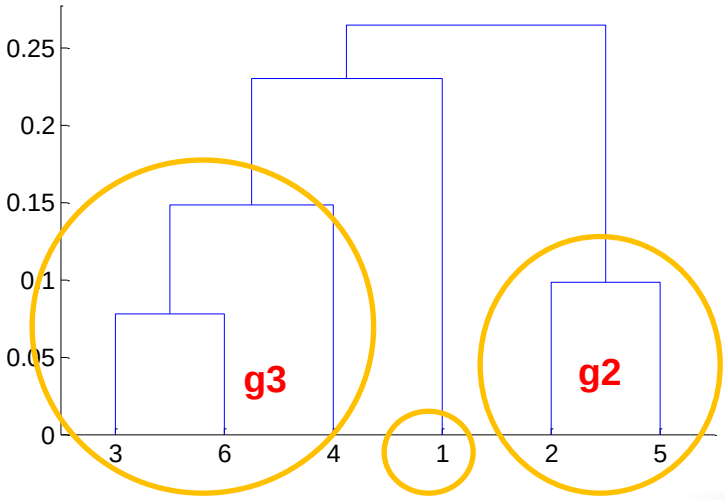
Inter-Cluster Similarity: Group Average

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

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

	I1	I2	I3	I4	I5	I6
I1	0.00	**0.24**	0.22	0.37	0.34	0.23
I2	0.24	0.00	0.15	0.20	0.14	0.25
I3	0.22	0.15	0.00	0.15	0.28	0.11
I4	0.37	0.20	0.15	0.00	0.29	0.22
I5	0.34	0.14	0.28	0.29	0.00	0.39
I6	0.23	0.25	0.11	0.22	0.39	0.00

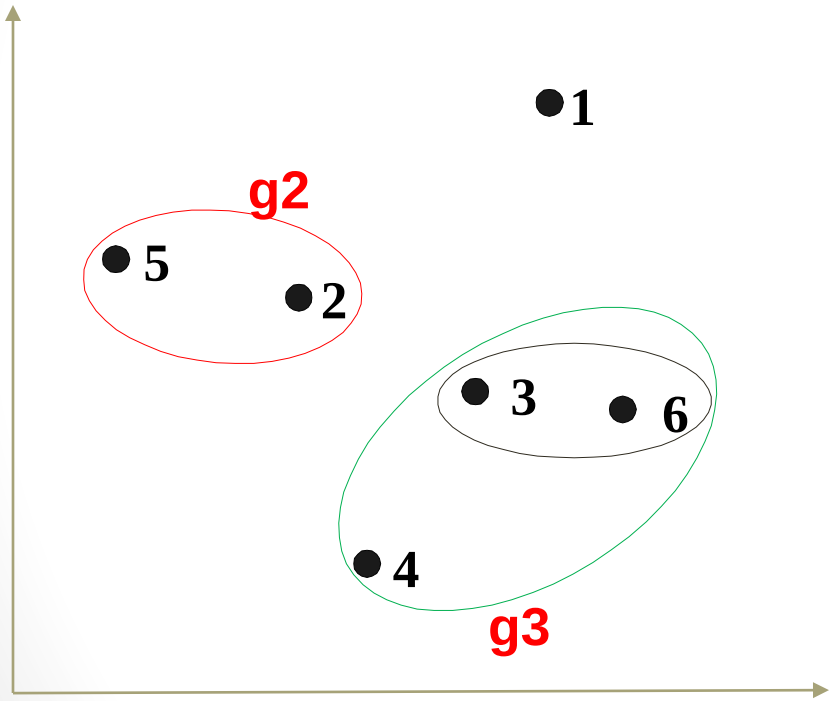
Euclidean distance (dissimlarity) matrix for 6 points



Dendrogram

Hierarchical Clustering: Group Average

Example: Calculating distance of g1 and g2 (to be merged) and some more merging choices



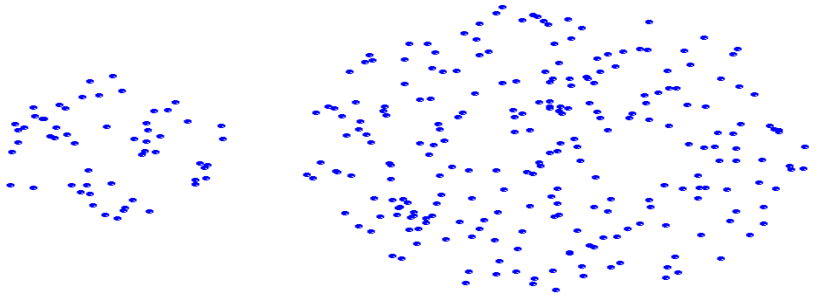
dist(g2, g3) = dist({3,6,4}, {2,5})
= (0.15+0.28+0.25+0.39+0.20+0.29) / (3*2)
= **0.26**

dist(g2, 1) = dist({2,5}, {1})
= (0.2357+0.3421) / (2*1)
= 0.2889

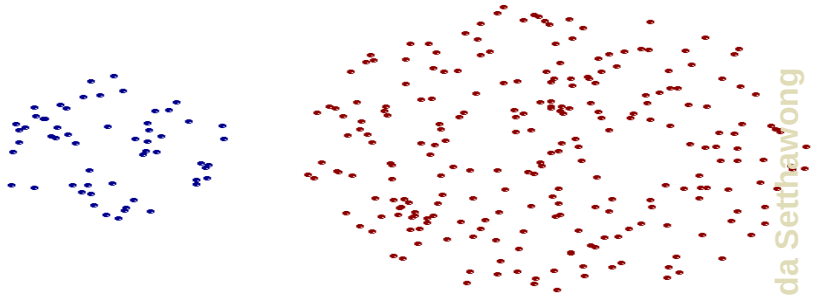
dist(g3, 1) = dist({3,6,4}, {1})
= (0.22+0.37+0.23) / (3*1)
= 0.28

	I1	I2	I3	I4	I5	I6
I1	0.00	**0.24**	0.22	0.37	0.34	0.23
I2	0.24	0.00	0.15	0.20	0.14	0.25
I3	0.22	0.15	0.00	0.15	0.28	0.11
I4	0.37	0.20	0.15	0.00	0.29	0.22
I5	0.34	0.14	0.28	0.29	0.00	0.39
I6	0.23	0.25	0.11	0.22	0.39	0.00

Strength of MIN



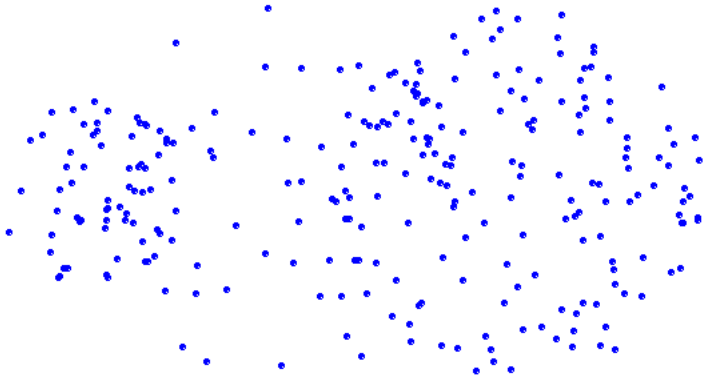
Original Points



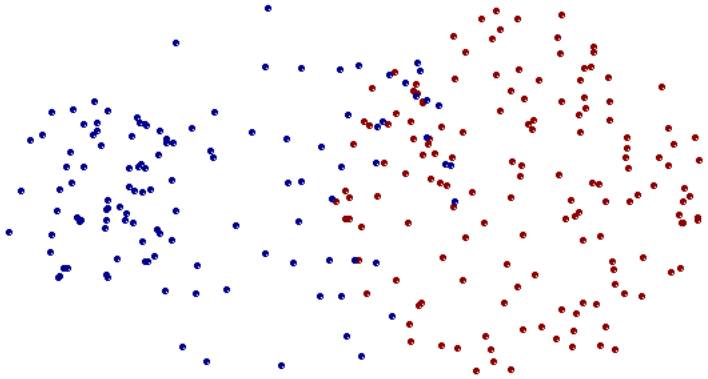
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



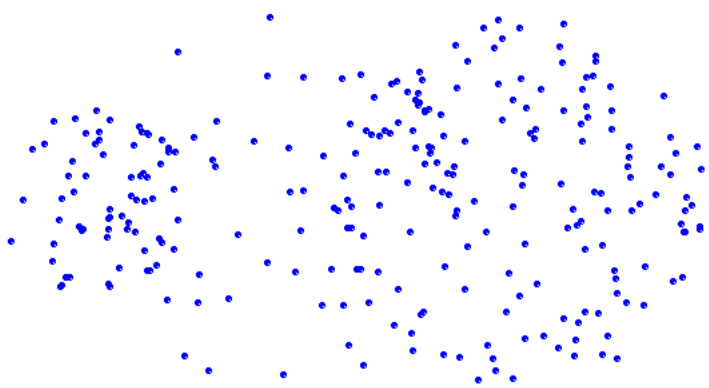
Original Points



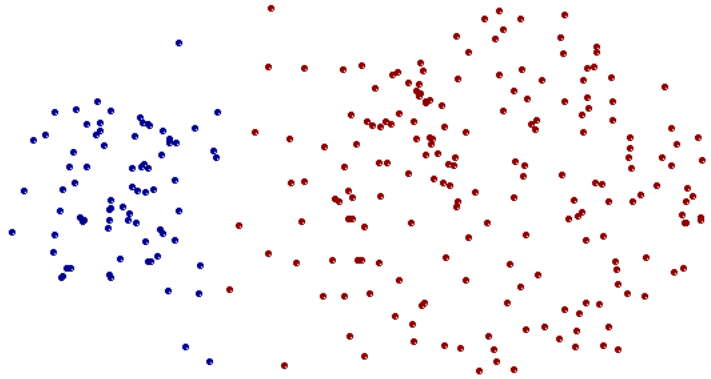
Two Clusters

- **Sensitive to noise and outliers**

Strength of MAX



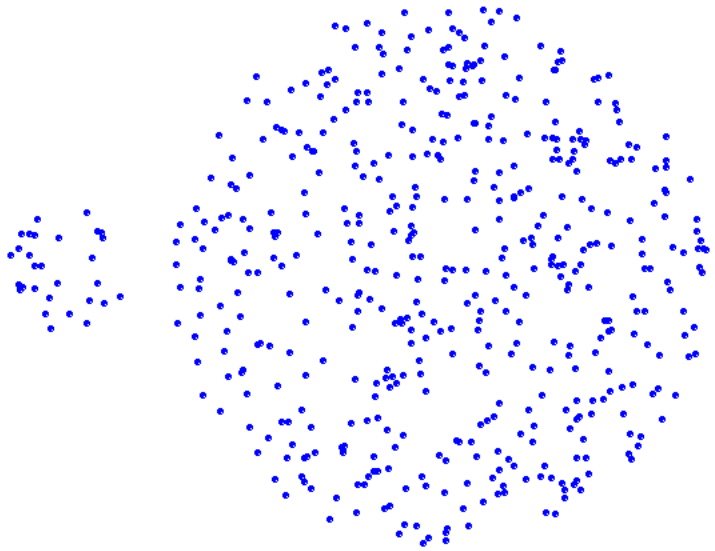
Original Points



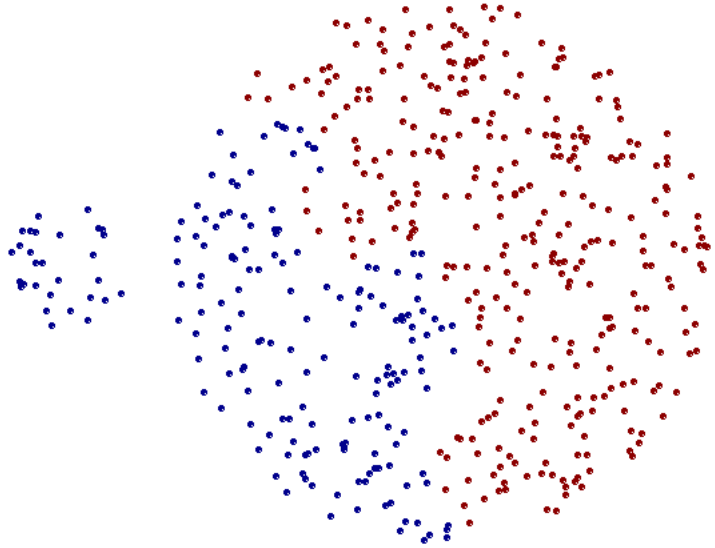
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



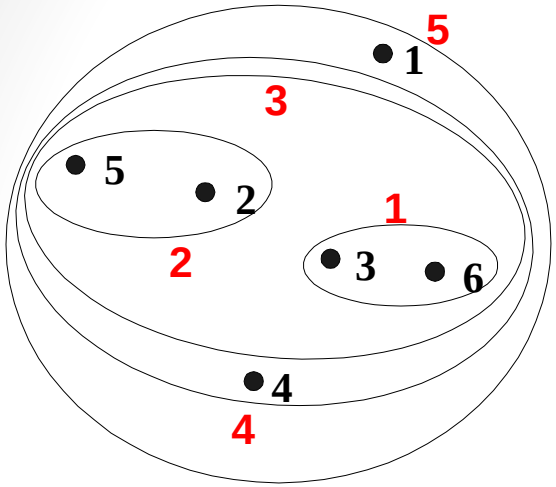
Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

Hierarchical Clustering: Group Average

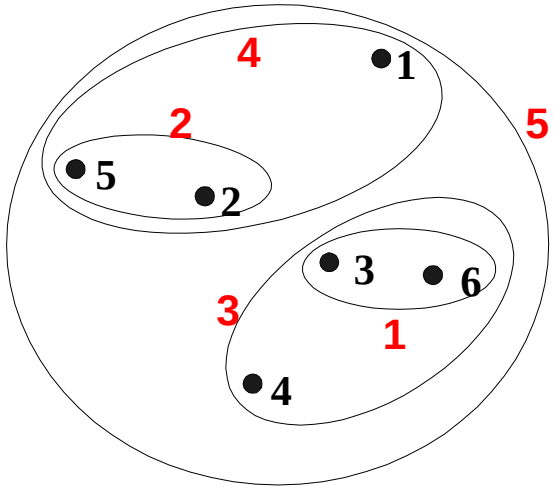
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

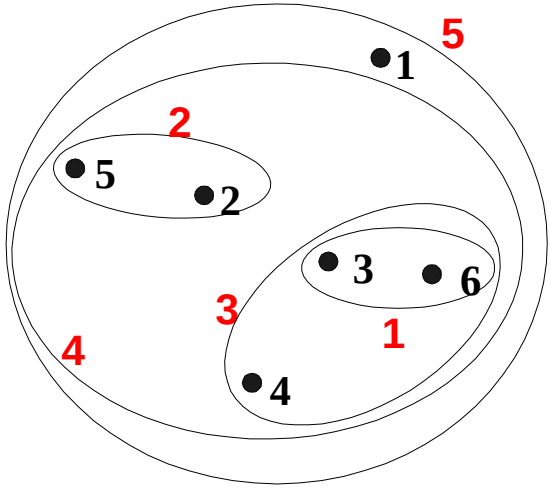
Hierarchical Clustering: Comparison



MIN

MAX





Group Average

Hierarchical Clustering: 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

Hierarchical Clustering: Problems and Limitations

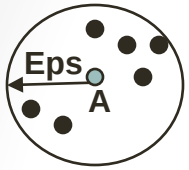
- Once a decision is made to combine two clusters, it cannot be undone.
- No objective function is directly minimized.
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

Outlines

- Basic concepts about clustering
- Clustering algorithms
  - K-means
  - Hierarchical clustering
  - DBSCAN
- Cluster Validity
- Measures of Cluster Validity

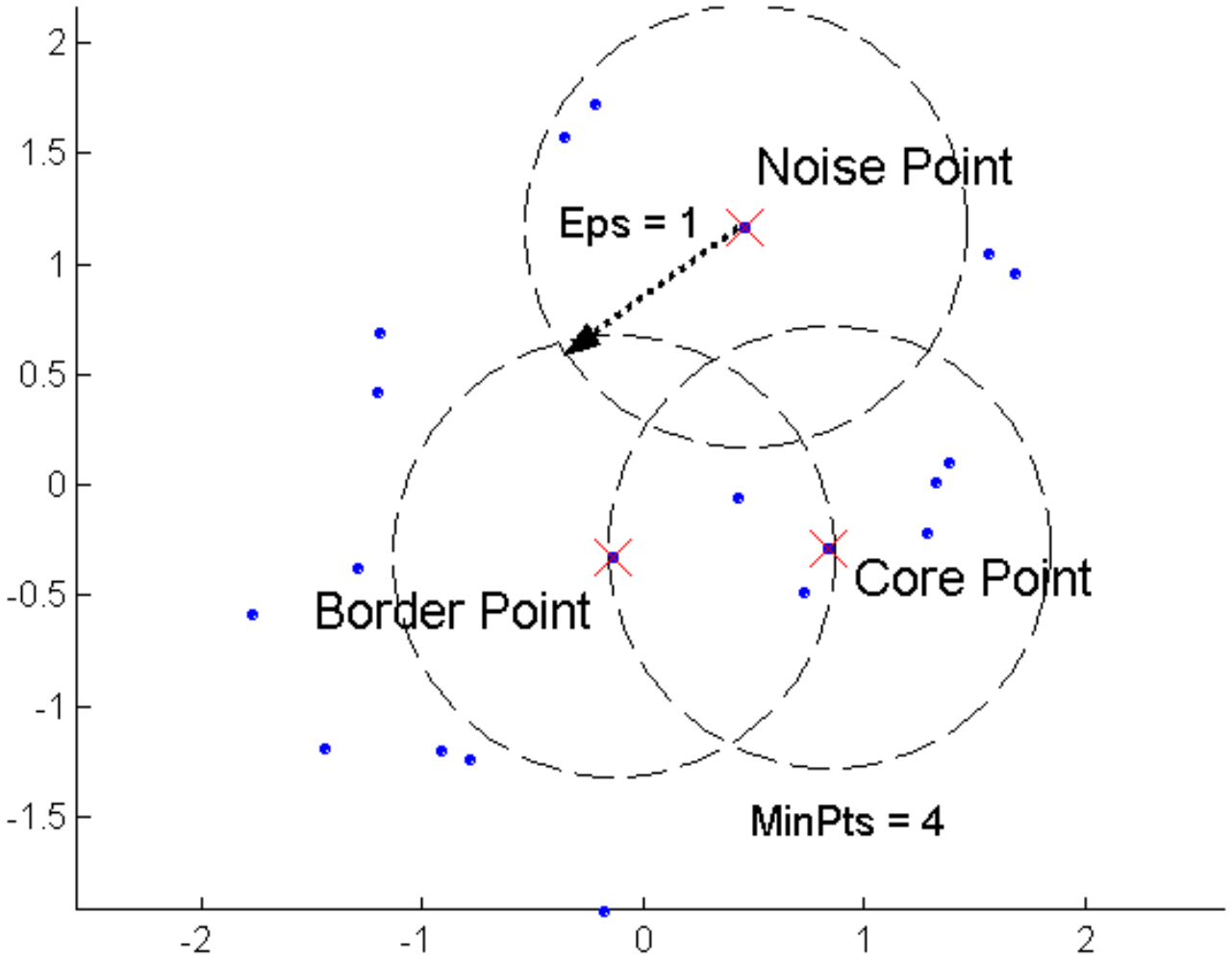
DBSCAN

- DBSCAN is a density-based algorithm.
  - 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



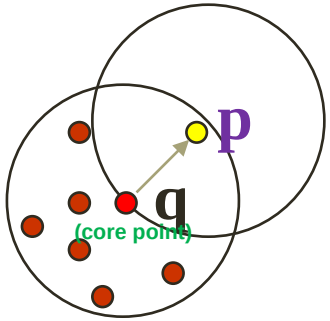
DBSCAN Parameters

- Eps and MinPts of each cluster (global values)

DBSCAN: Key Idea 1

- Two parameters*:*
  - ***Eps***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Eps-neighbourhood of that point
- Concept of Neighbor:
  - **N_{Eps}(q) = {p belongs to D | dist(p,q) <= Eps}**
    - Neighbor of a point q is any point whose distance is less than or equal to Eps.
- Directly density-reachable**:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** if
  - 1) ***p*** is neighbor of *q.*
  - 2) q is core point.
    - core point condition:

      $|N_{Eps}(q)| >= MinPts$

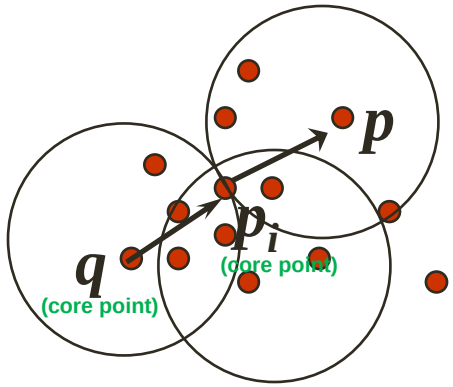


MinPts = 5

Eps = 1 cm

DBSCAN: Key Idea 2

- Density-reachable:
  - A point *p* is density-reachable from a point *q* wrt. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, *(in the figure:* $p_1 = q$, $p_n = p$*)* such that p_{i+1} is directly density-reachable from p_i

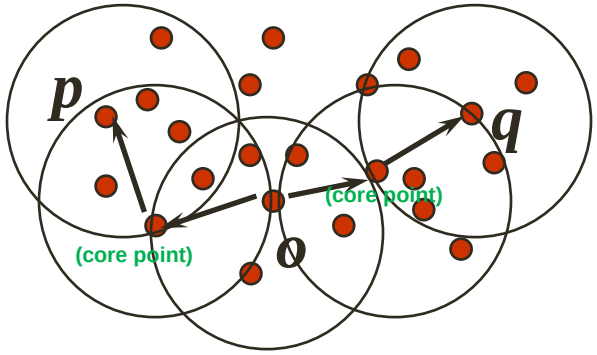


- However, **Reachability** is not a symmetric relation.
  - Only core points can reach non-core points, but not vice versa.

DBSCAN: Key Idea 3

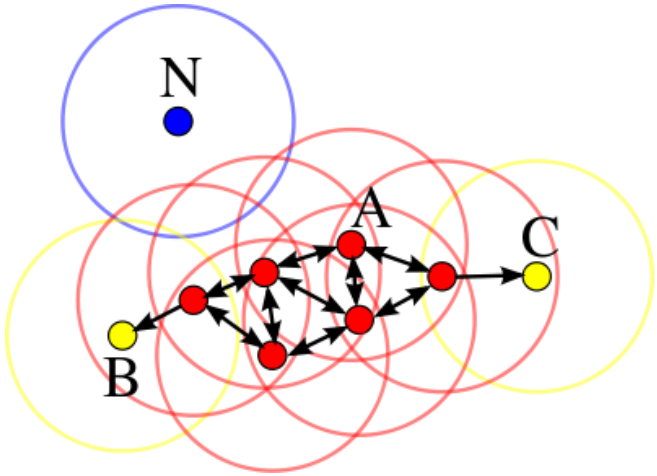
- To make the reachability symmetric, it requires another notion, **connectedness**.

- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



DBSCAN: Key Idea 4

- Relies on a *density-based* notion of cluster, a *cluster* is defined as a maximal set of density-connected points.



minPts = 4

- A cluster satisfies two properties:
  1. All points within the cluster are mutually density-connected.
  2. If a point is density-reachable from some point of the cluster, it is part of the cluster as well.

Ref: https://en.wikipedia.org/wiki/DBSCAN#Algorithm

DBSCAN: Abstract Algorithm

1. Find the points in the ε (eps) neighborhood of every point, and identify the core points with more than minPts neighbors.

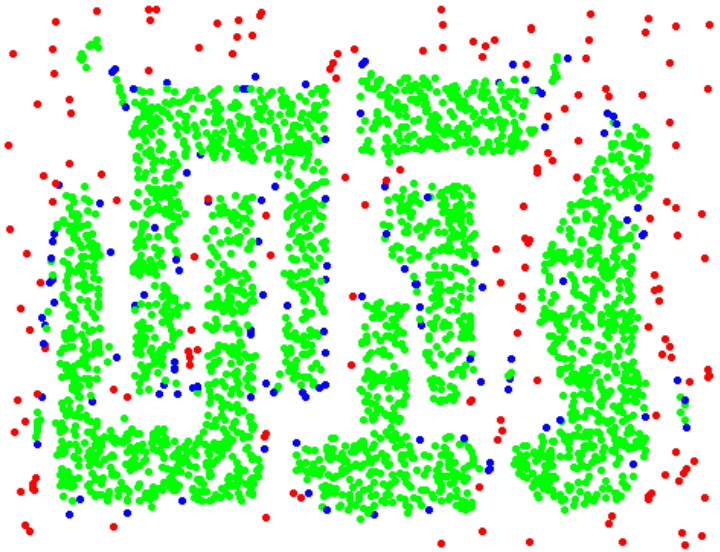
2. Find the connected components of *core* points on the neighbor graph, ignoring all non-core points.

3. Assign each non-core point to a nearby cluster if the cluster is an ε (eps) neighbor, otherwise assign it to noise.

Ref: https://en.wikipedia.org/wiki/DBSCAN#Algorithm

DBSCAN: Core, Border and Noise Points

Original Points

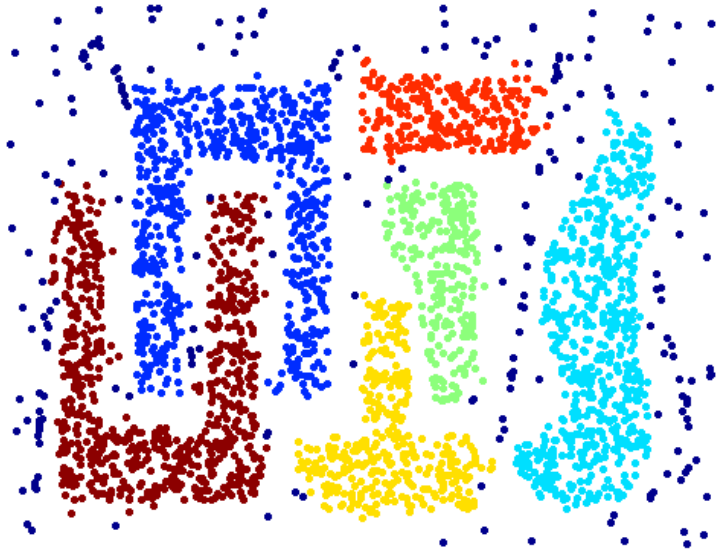


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well

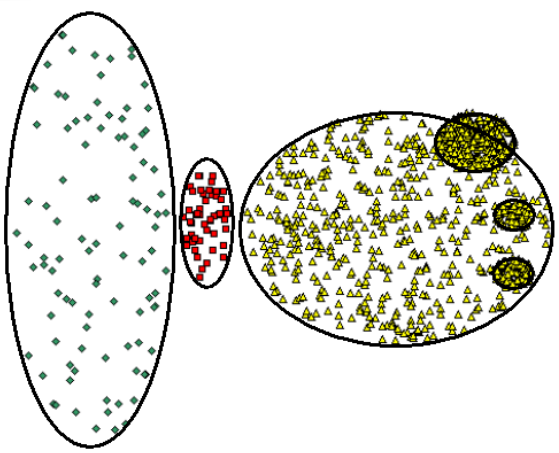
Original Points



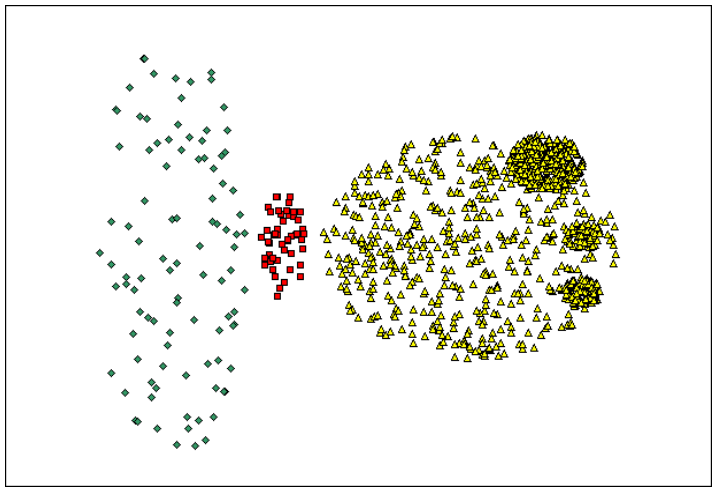
Clusters

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

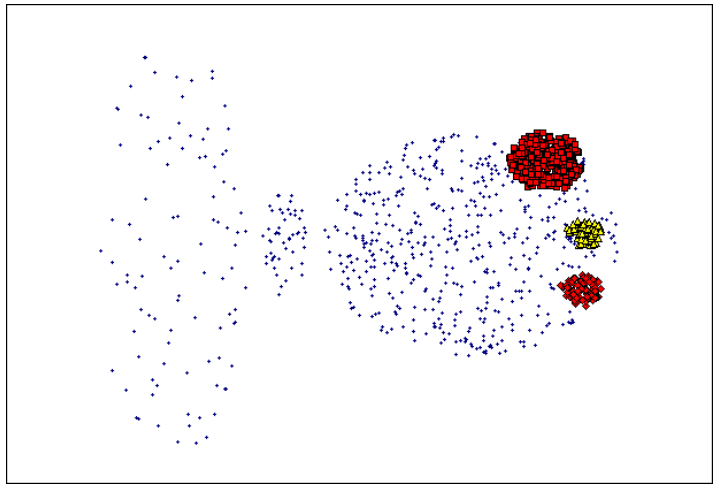
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data

Determining Parameter: MinPts

- Using a rule of thumb:
  - **Suggestion 1:** minPts ≥ D + 1
    - where, D denotes dimensions (number of attributes) in a data set
    - *Note: minPts must be chosen at least 3.*
  - **Suggestion 2:** minPts = 2 × D
    - For very large data, for noisy data or for data that contains many duplicates

Ref: https://en.wikipedia.org/wiki/DBSCAN#Algorithm

Determining Parameter: Eps

- Heuristic: the "thinnest" cluster in the dataset D
- How? – Compute and sort the k^{th} distance of all points
  - Find a threshold with the maximal k-dist value in the thinnest cluster of D.
  - The threshold point is the first point in the first "valley" of the sorted k-dist graph.

E.g., MinPts = k = 4

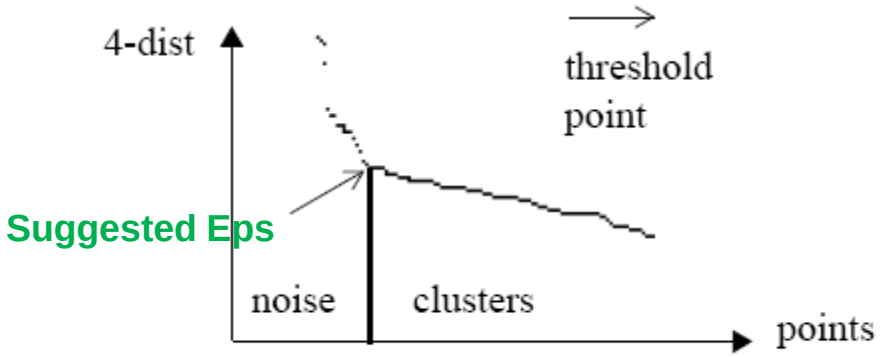


figure 4: sorted 4-dist graph for sample database 3

Outline

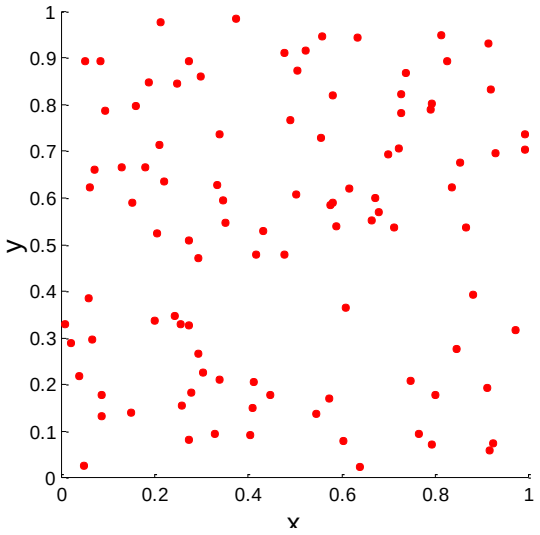
- Clustering Algorithm: DBSCAN
- Cluster Validity
- Measures of Cluster Validity
- Case Study: Clustering with RapidMiner

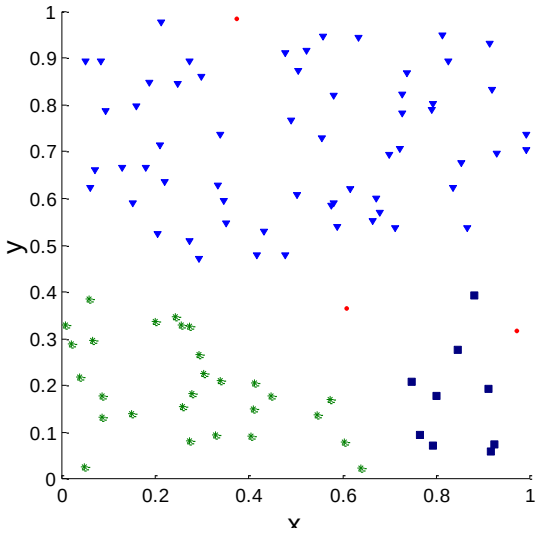
Cluster Validity

- 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?
  - "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 results:
    - Two clustering (two sets of clusters)
    - Two clusters

Clusters found in Random Data

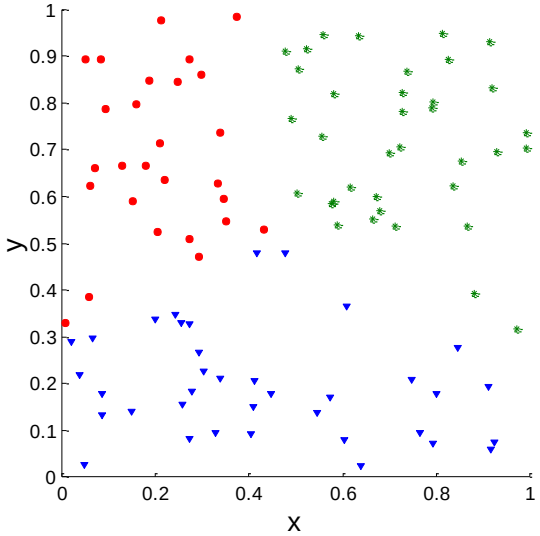
Random Points

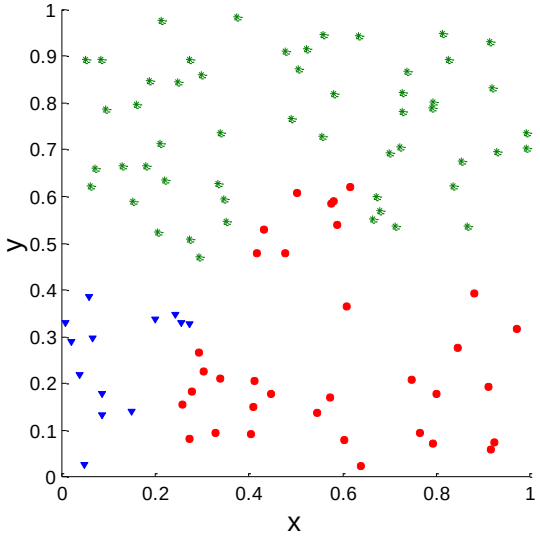




DBSCAN

K-means





Complete Link

Different Aspects of Cluster Validation

1. Determining whether non-random structure actually exists in the data.
2. [To evaluate algorithms' performance] 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.

Outlines

- Basic concepts about clustering
- Clustering algorithms
  - K-means
  - Hierarchical clustering
  - DBSCAN
- Cluster Validity
- Measures of Cluster Validity
- Case Study with Rapidminer

Measures of Cluster Validity

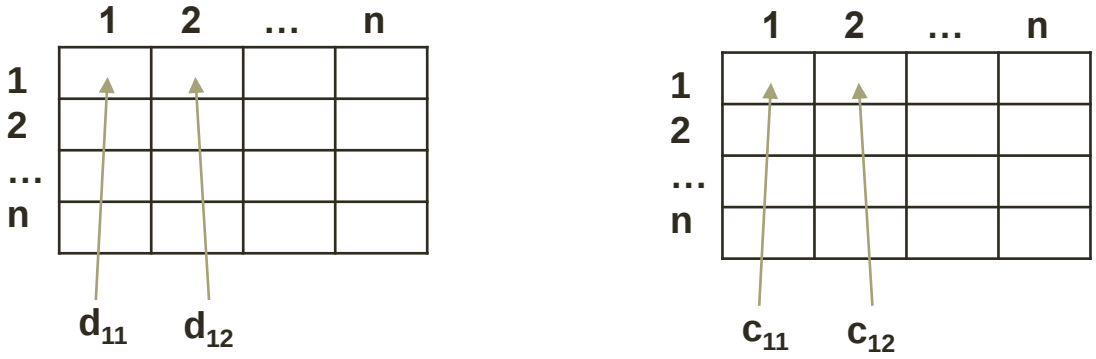
- External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
  - Purity, Entropy
- Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
  - Sum of Squared Error (SSE)

Measuring Cluster Validity Via Correlation – 1/3

- Use 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 – 2/3

- Given Distance Matrix D = {d_{11},d_{12}, …, d_{nn}} and Incidence Matrix C= { c_{11}, c_{12},…, c_{nn} },

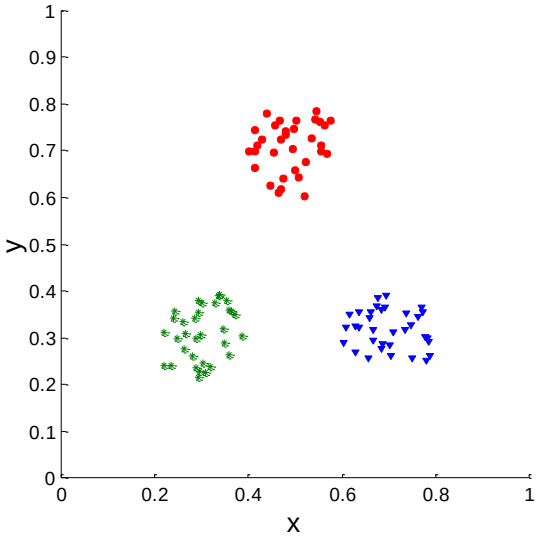


Correlation between D and C:

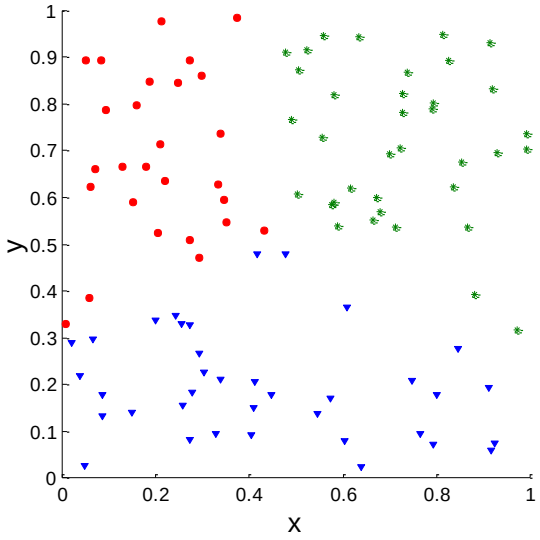
$$r = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij}-\bar{d})(c_{ij}-\bar{c})}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij}-\bar{d})^2}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij}-\bar{c})^2}}$$

Measuring Cluster Validity Via Correlation – 3/3

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



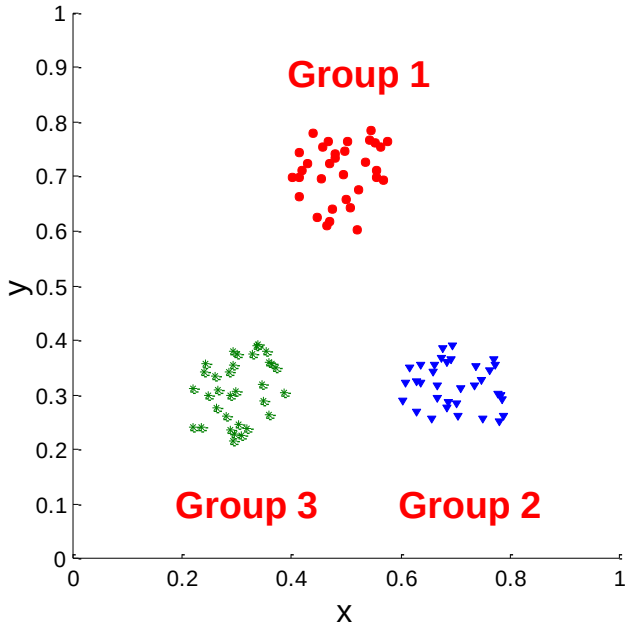
Corr = -0.9235

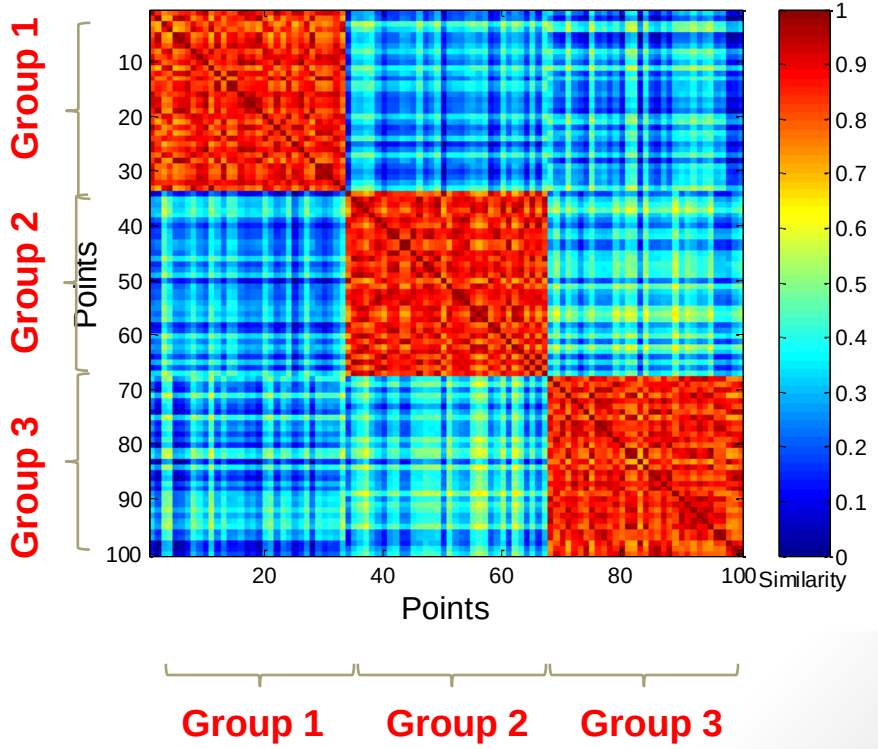


Corr = -0.5810

Using **Similarity Matrix** for Cluster Validation

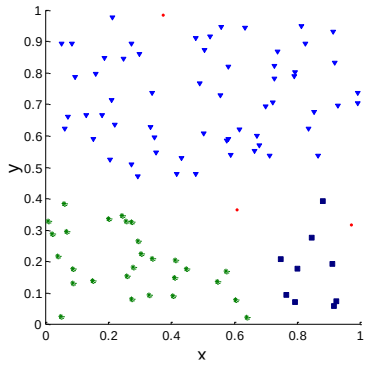
- For the similarity matrix,
  - Sort the points (objects) on both x and y axes using cluster labels
  - Color of each point represents similarity values (very blue = 0, very red = 1)
  - Then, inspect visually.

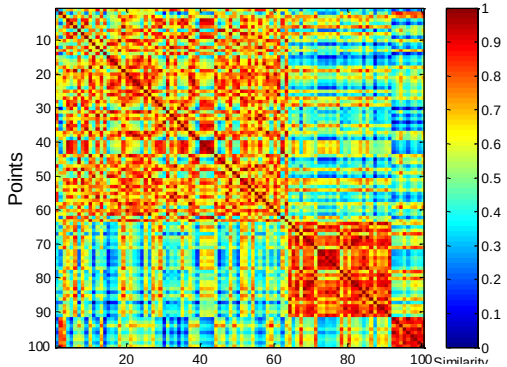




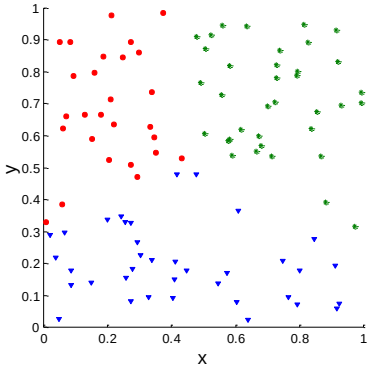
Using **Similarity Matrix** for Cluster Validation

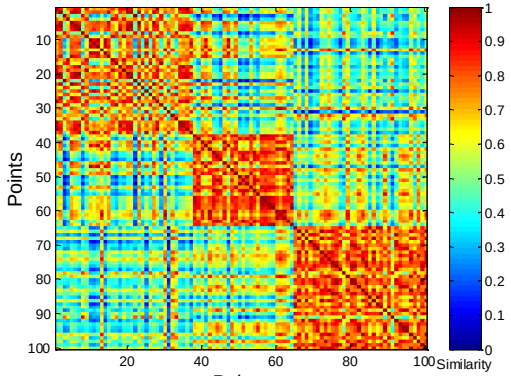
Clusters in random data are not so crisp



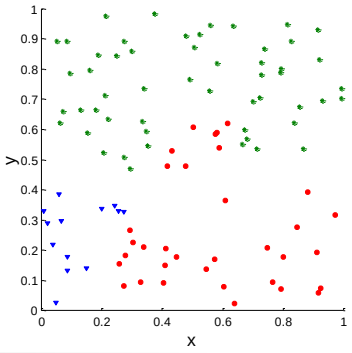


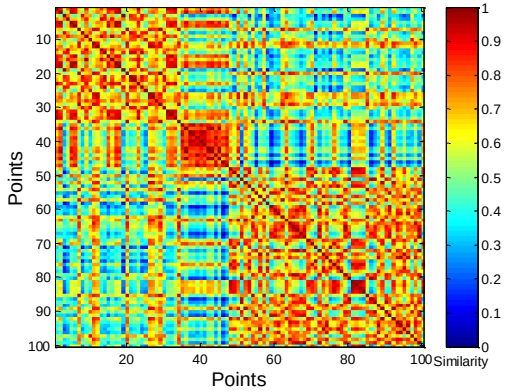
DBSCAN





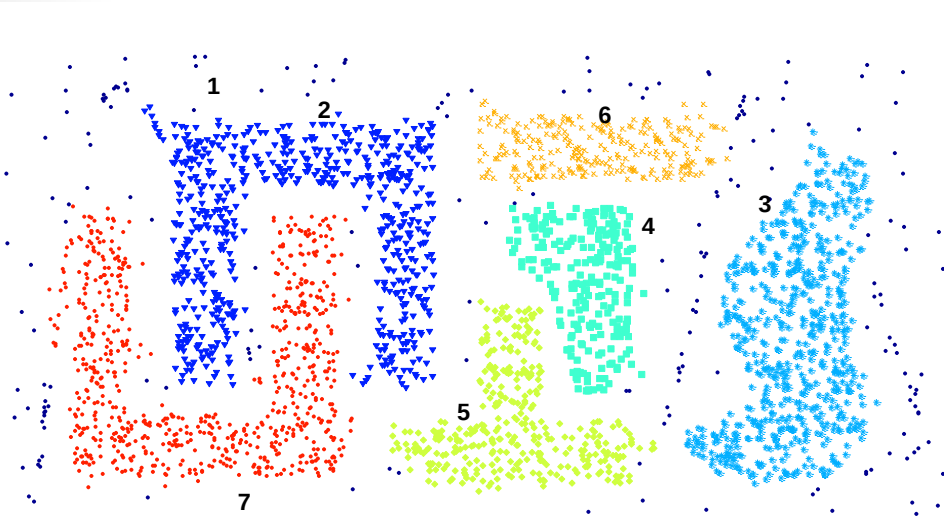
K-means

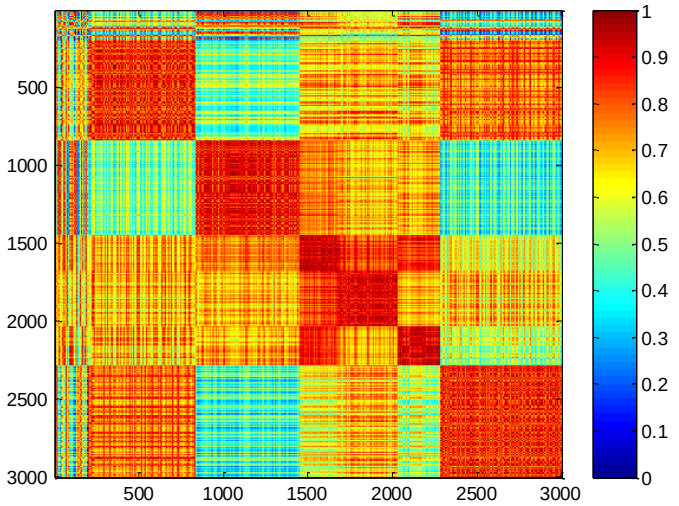




Complete Link

Using **Similarity Matrix** for Cluster Validation





DBSCAN

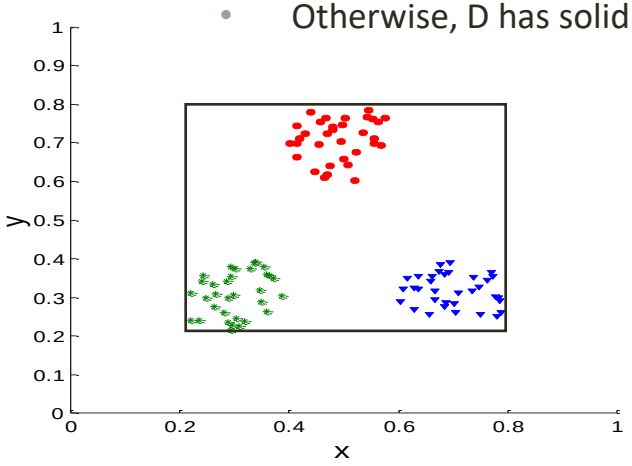
A Statistical Framework for Cluster Validity

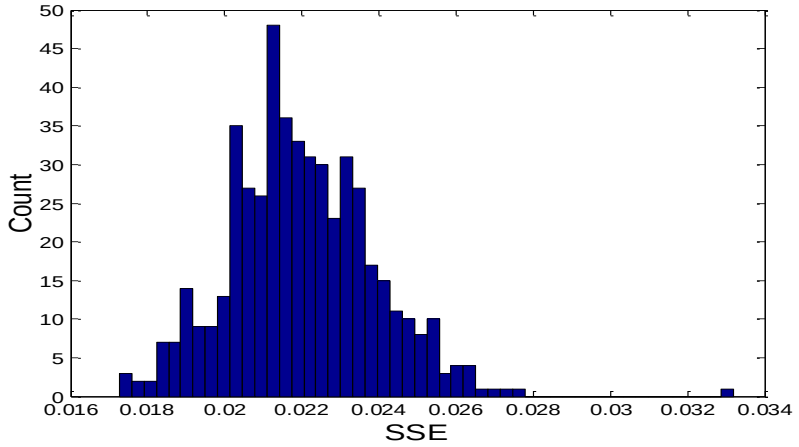
- Idea: the more "**atypical**" a clustering result is, the more likely it represents valid structure (pattern) in the data
- It can ***compare*** the calculated values *of an index that result* from random data or clusterings ***to*** those of a clustering result.
  - If the value of the index is unlikely, then the cluster results are valid

- Note: for comparing the results of two different sets of cluster analyses, a framework is less necessary.

Statistical Framework for Sum of Squared Error (SSE): To verify if a dataset has solid structure/patterns.

- Steps:
  1. Calculate **SSE** of clustering generated by K-means (=0.005) on a dataset D.
  2. Compare the **SSE** in step 1) against three clusters in random data (the detail is provided below)
     - Randomly generate 500 datasets and use K-means on each dataset to generate clusterings and calculate SSE for each result to plot histogram.
     - **Analysis:** If the SSE in step 1) is in the histogram in step 2) then dataset D has no solid pattern.
       - Otherwise, D has solid patterns.

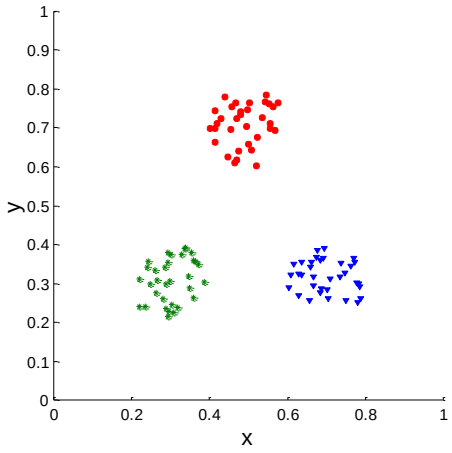




Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

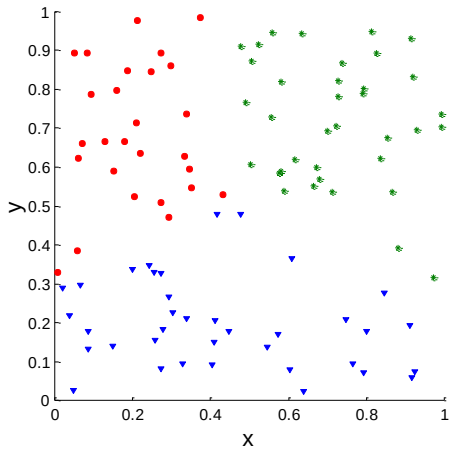
Statistical Framework for Correlation

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



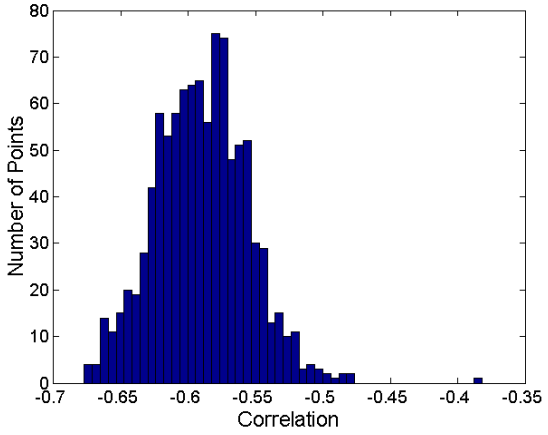
Dataset D

Corr = -0.9235



A Random Dataset

Corr = -0.5810



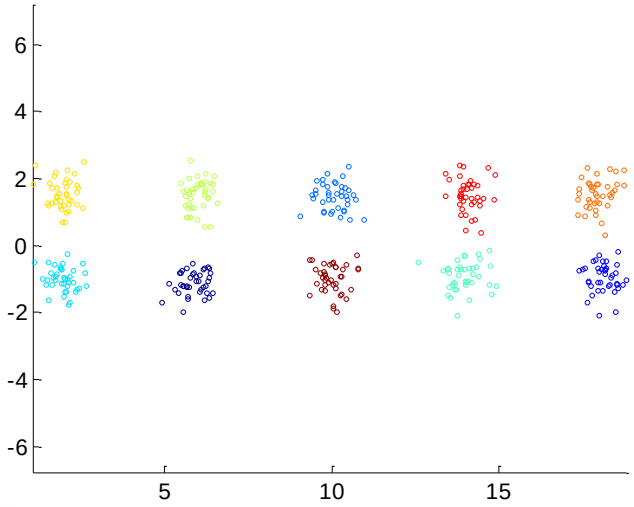
Outlines

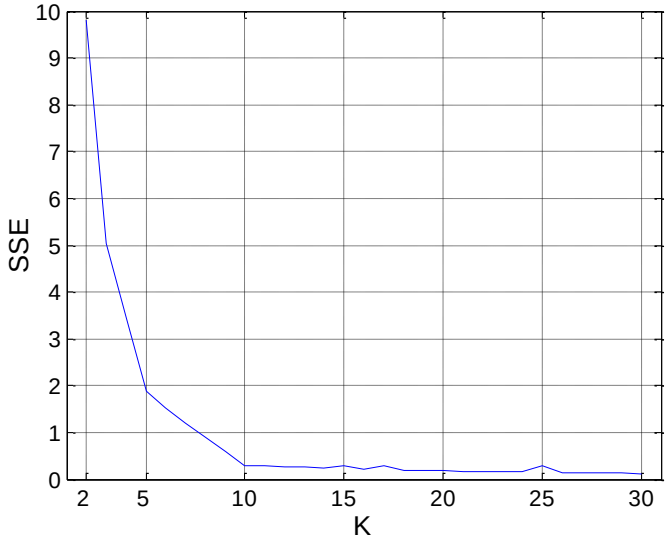
- Basic concepts about clustering
- Clustering algorithms
  - K-means
  - Hierarchical clustering
  - DBSCAN
- Cluster Validity
- Measures of Cluster Validity

Internal Measures: Sum of Squared Error (SSE)

- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
    - Good for comparing two clusterings or two clusters (average SSE).
    - Can also be used to estimate the number of clusters

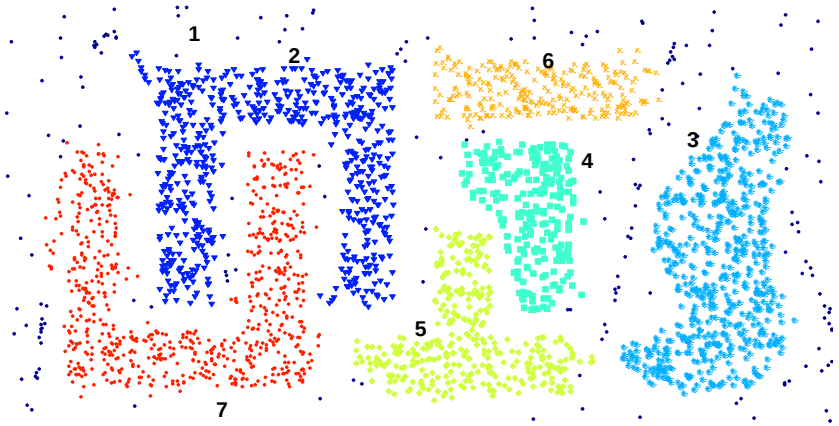
$$SSE = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$



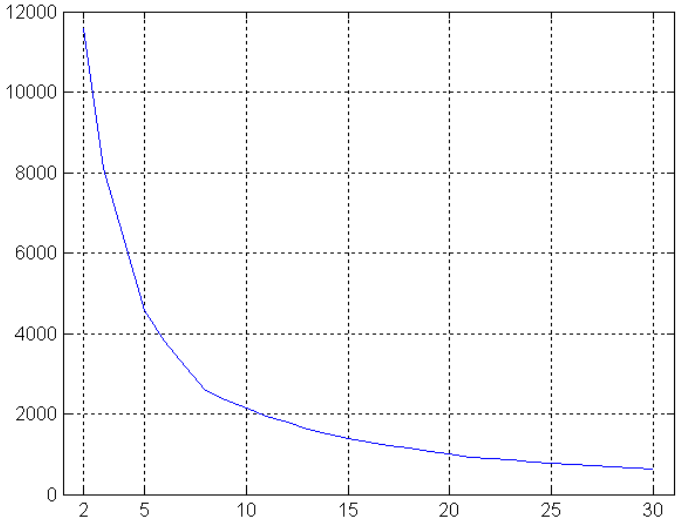


Internal Measures: SSE

- SSE curve is used to suggest no. of clusters for a more complicated data set



True clusters



SSE of clusters found using K-means

Internal Measures: Cohesion

- Cluster Cohesion: Measures how closely related are objects in a cluster (how close they are to their centroid)

$$\textbf{Total Cohesion} = \sum_{i=1}^{K} \sum_{x \epsilon C_i} \textbf{sim}(x, c_i),$$

where *sim()* denotes a proximity function

Proximity Function	Centroid (c_i)
Manhattan	Median
Euclidean	Mean
Cosine	Mean
Etc.	…

- A simple cohesion for 1 dimensional data is measured by the **W**ithin cluster **S**um of **S**quares:

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

Internal Measures: Cohesion and Separation

- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
  - A simple cohesion for 1 dimensional data is measured by the Between cluster Sum of Squares

$$BSS = \sum_i |C_i|(m - m_i)^2$$

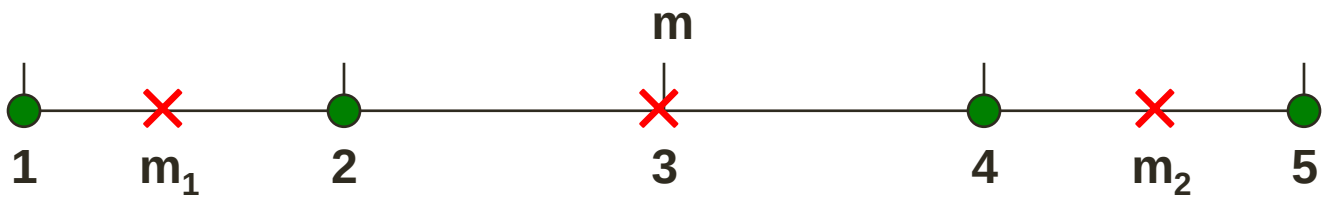
where

- $|C_i|$ is the size of cluster i
- m is a global centroid (a centroid of the dataset).
- m_i is a local centroid (a centroid of a cluster).

Internal Measures: Cohesion and Separation

- Example: SSE
  - BSS + WSS = constant

E.g., 4 objects (1,2,4,5)



K=2 clusters:

$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$

$$Total = 1 + 9 = 10$$

K=1 cluster:

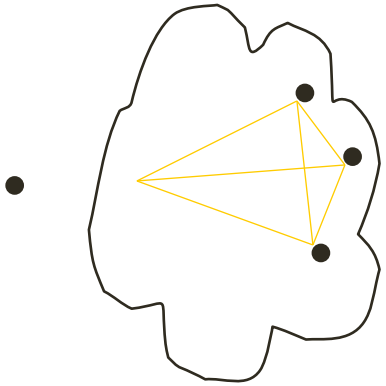
$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$BSS = 4 \times (3-3)^2 = 0$$

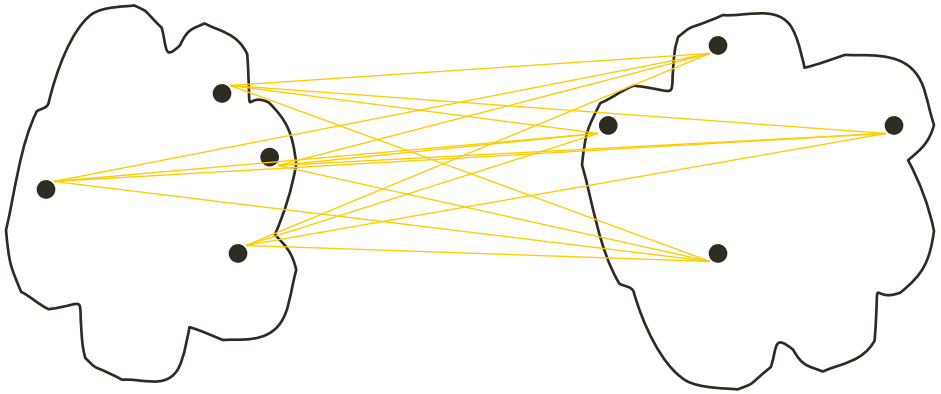
$$Total = 10 + 0 = 10$$

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



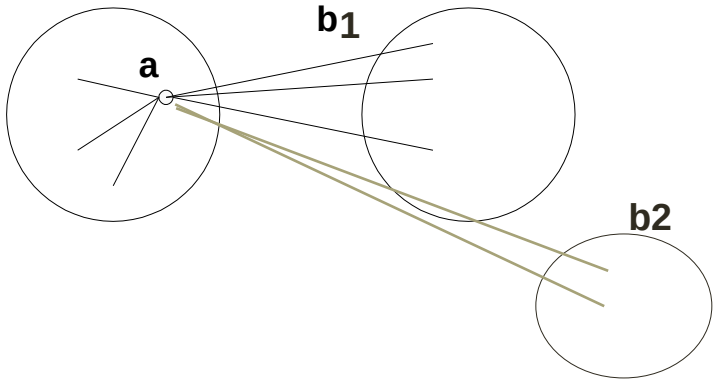
separation

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings

- For an individual point, *i*
  - Calculate cohesion ***a*** = average distance of *i* to the points in its cluster
  - Calculate separation ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

$$s = (b - a)/\max(a, b)$$



  - Typically between 0 and 1.
  - The closer to 1 the better.

- Can calculate the Average Silhouette width for a cluster or a clustering

External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

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

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

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