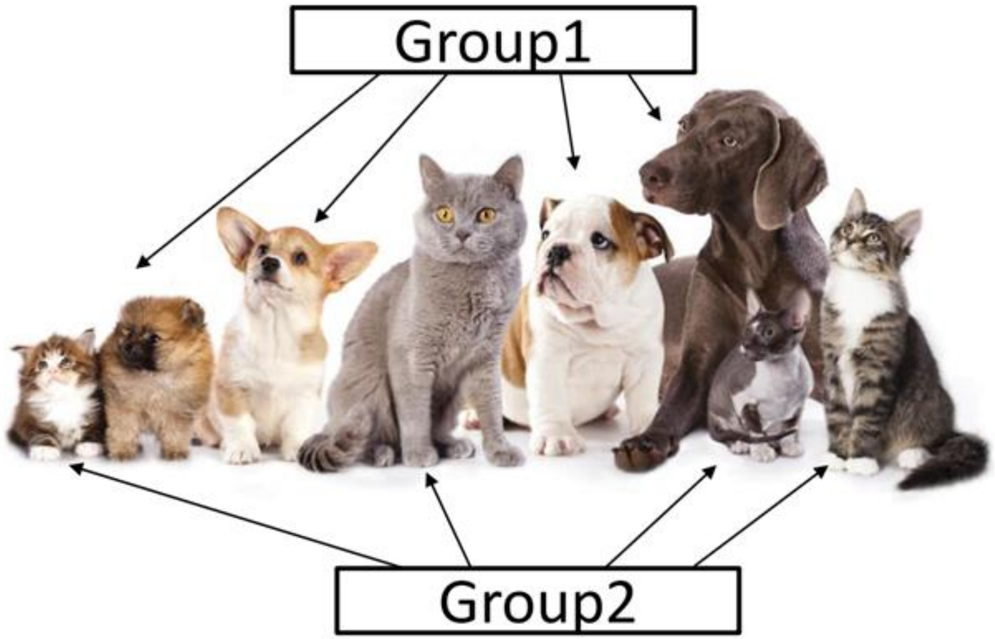
Introduction to clustering

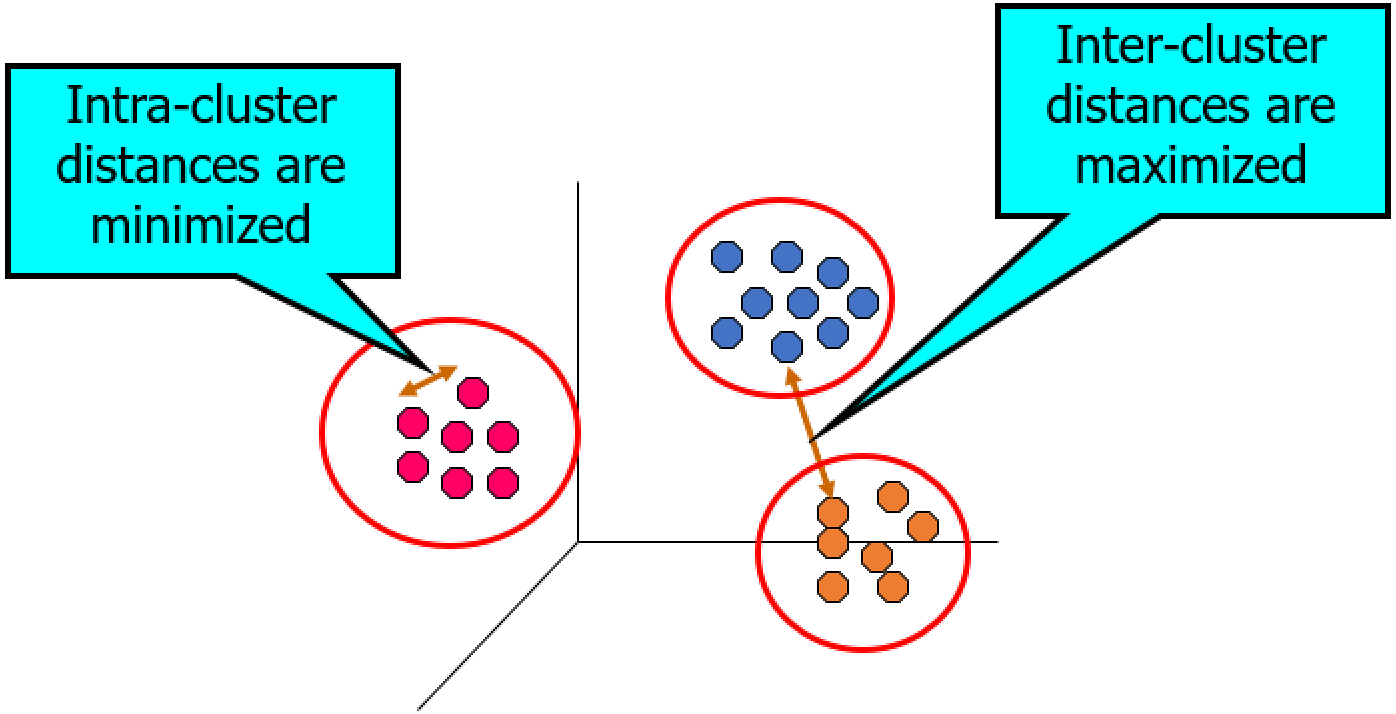
Unsupervised Learning- Clustering

Clustering refers to a very broad set of techniques for finding *subgroups* in a data set.



Group1

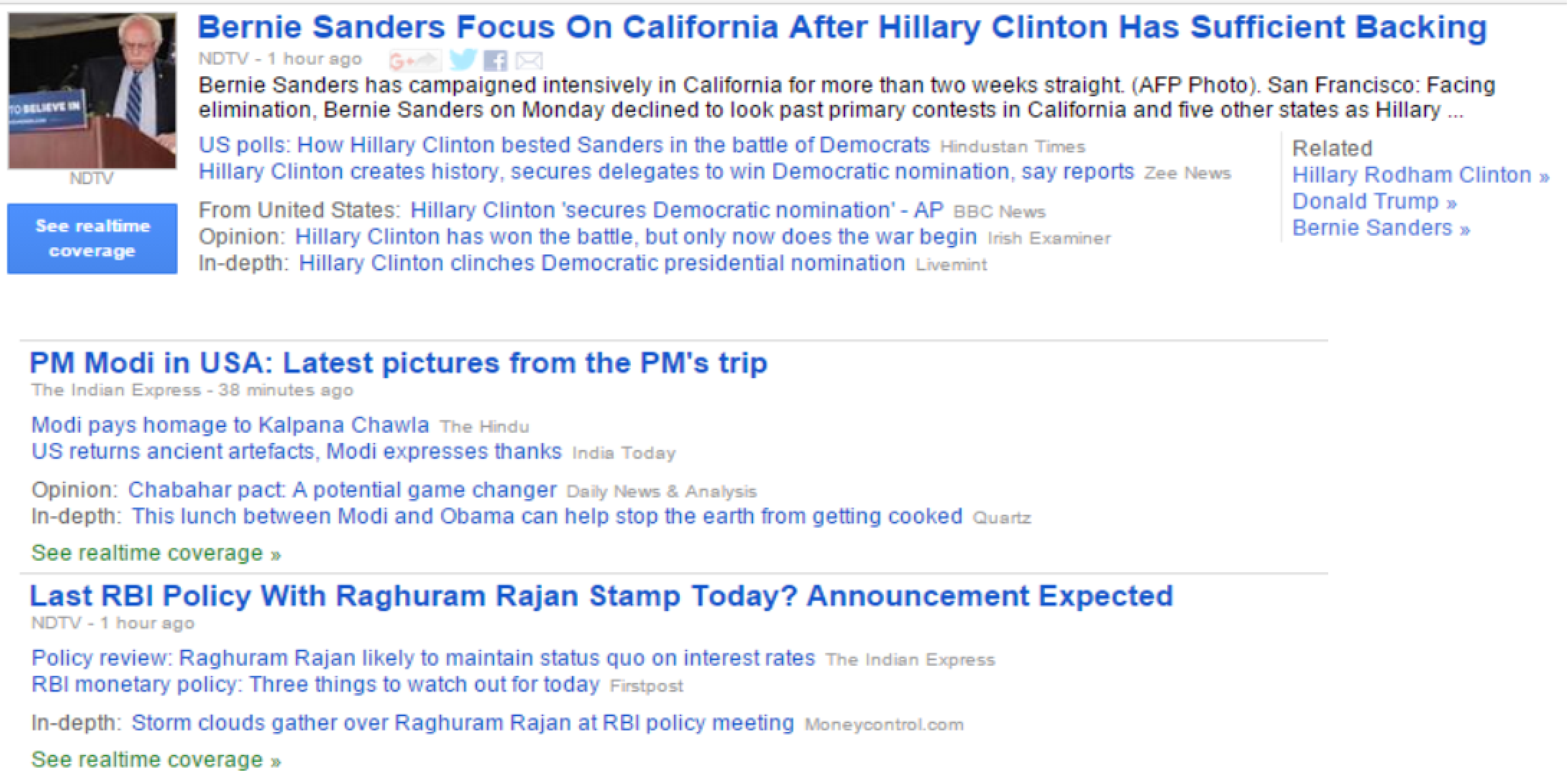
Group2



Intra-cluster distances are minimized

Inter-cluster distances are maximized

Applications of Clustering: Google News



NDTV

Bernie Sanders Focus On California After Hillary Clinton Has Sufficient Backing

NDTV - 1 hour ago

Bernie Sanders has campaigned intensively in California for more than two weeks straight. (AFP Photo). San Francisco: Facing elimination, Bernie Sanders on Monday declined to look past primary contests in California and five other states as Hillary ...

US polls: How Hillary Clinton bested Sanders in the battle of Democrats Hindustan Times
Hillary Clinton creates history, secures delegates to win Democratic nomination, say reports Zee News

From United States: Hillary Clinton 'secures Democratic nomination' - AP BBC News
Opinion: Hillary Clinton has won the battle, but only now does the war begin Irish Examiner
In-depth: Hillary Clinton clinches Democratic presidential nomination Livemint

Related
Hillary Rodham Clinton »
Donald Trump »
Bernie Sanders »

PM Modi in USA: Latest pictures from the PM's trip

The Indian Express - 38 minutes ago

Modi pays homage to Kalpana Chawla The Hindu
US returns ancient artefacts, Modi expresses thanks India Today

Opinion: Chabahar pact: A potential game changer Daily News & Analysis
In-depth: This lunch between Modi and Obama can help stop the earth from getting cooked Quartz

See realtime coverage »

Last RBI Policy With Raghuram Rajan Stamp Today? Announcement Expected

NDTV - 1 hour ago

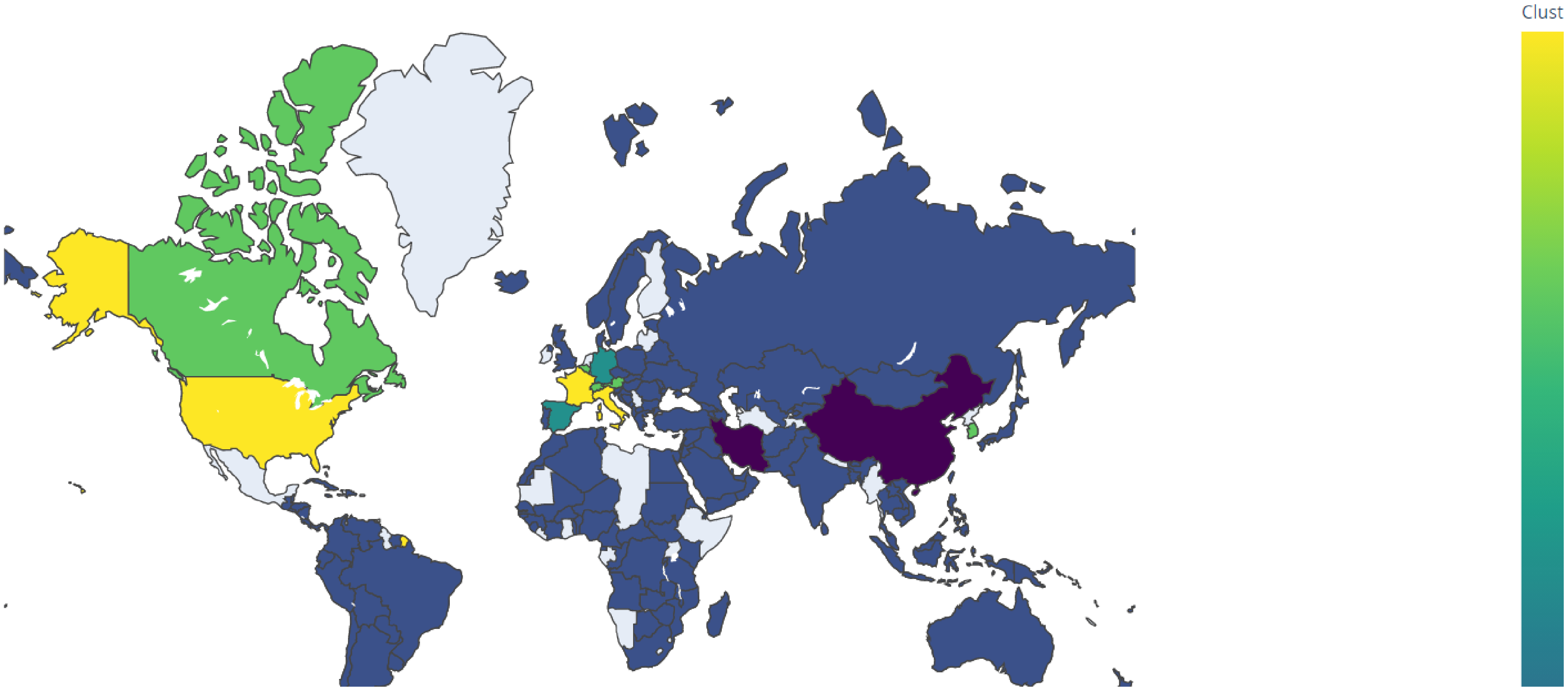
Policy review: Raghuram Rajan likely to maintain status quo on interest rates The Indian Express
RBI monetary policy: Three things to watch out for today Firstpost

In-depth: Storm clouds gather over Raghuram Rajan at RBI policy meeting Moneycontrol.com

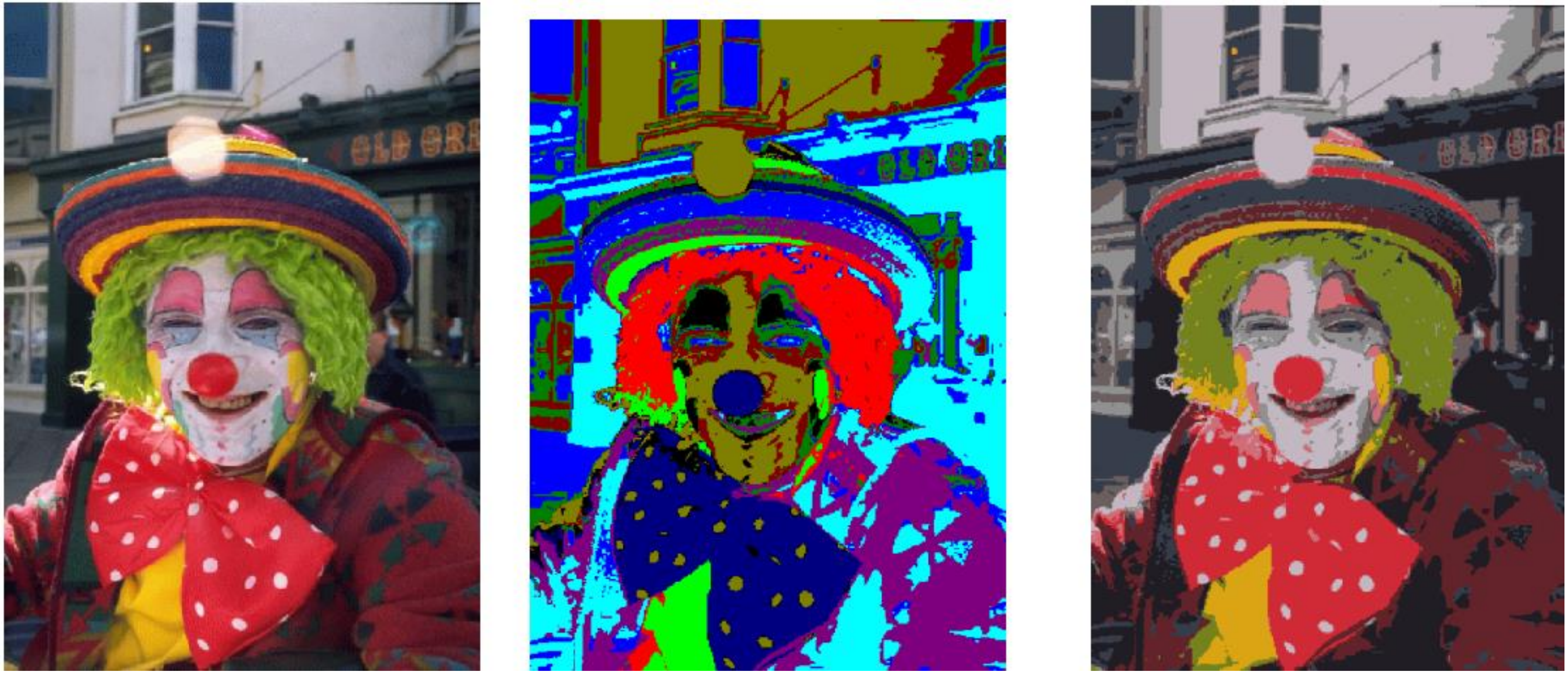
See realtime coverage »

Applications of Clustering: Visualization

Clustering World Countries affected by Coronavirus

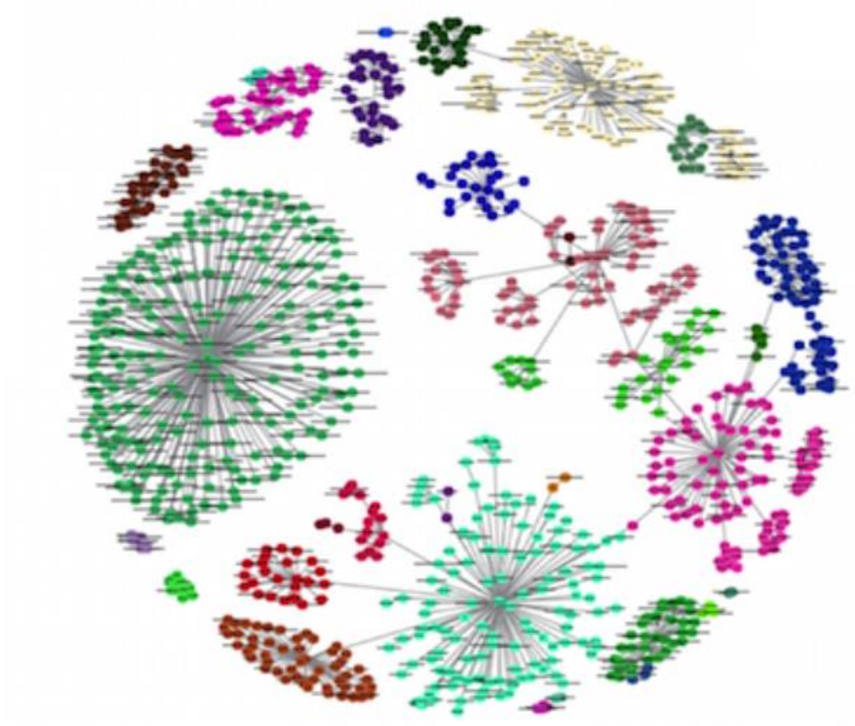


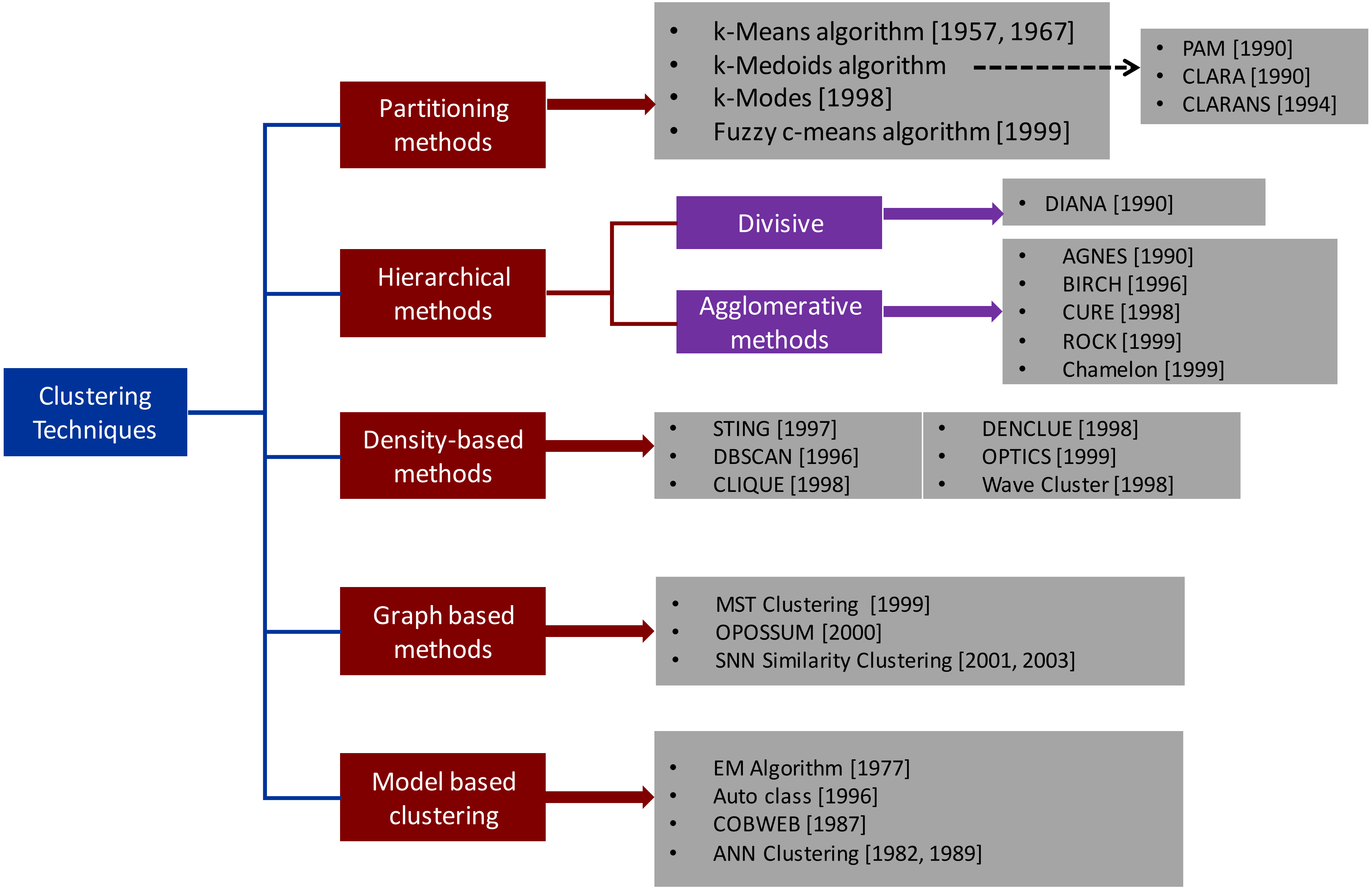
Applications of Clustering: Image Segmentation



Applications of Clustering: Social network analysis

how to extract communities
Interaction between groups





Clustering Techniques

Partitioning methods

- k-Means algorithm [1957, 1967]
- k-Medoids algorithm
- k-Modes [1998]
- Fuzzy c-means algorithm [1999]

- PAM [1990]
- CLARA [1990]
- CLARANS [1994]

Hierarchical methods

Divisive

- DIANA [1990]

Agglomerative methods

- AGNES [1990]
- BIRCH [1996]
- CURE [1998]
- ROCK [1999]
- Chamelon [1999]

Density-based methods

- STING [1997]
- DBSCAN [1996]
- CLIQUE [1998]

- DENCLUE [1998]
- OPTICS [1999]
- Wave Cluster [1998]

Graph based methods

- MST Clustering [1999]
- OPOSSUM [2000]
- SNN Similarity Clustering [2001, 2003]

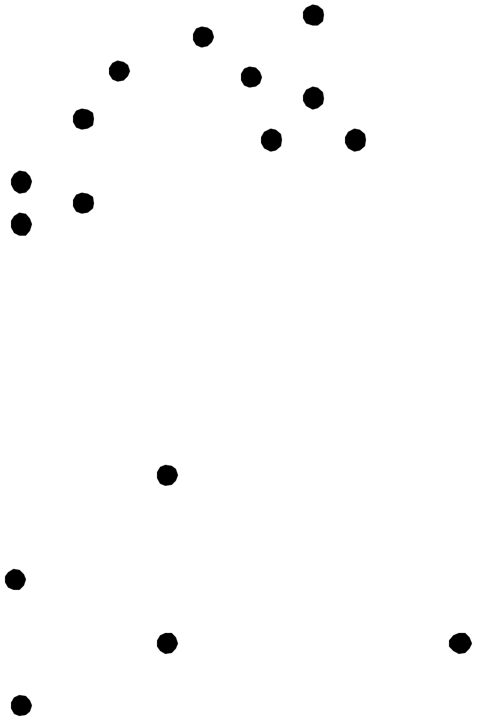
Model based clustering

- EM Algorithm [1977]
- Auto class [1996]
- COBWEB [1987]
- ANN Clustering [1982, 1989]

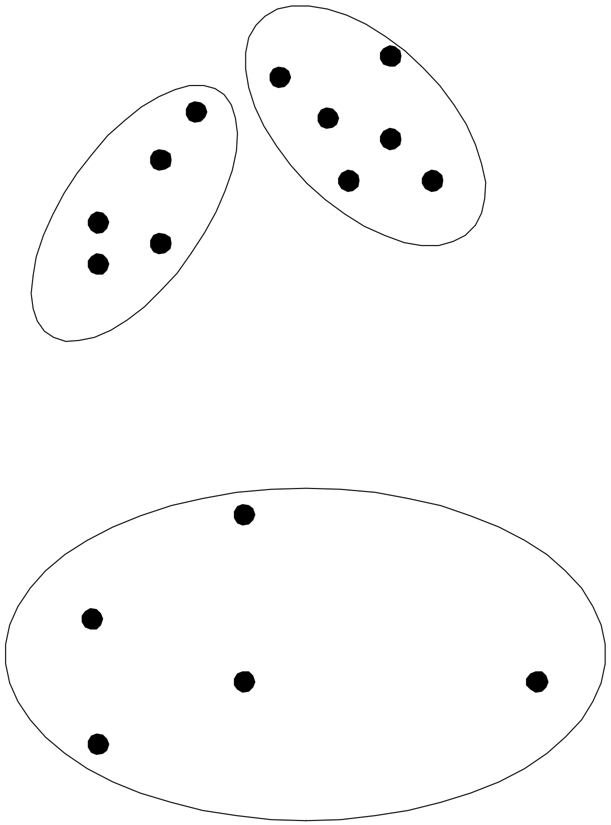
Types of Clustering: partitional vs hierarchical

Partitional Clustering

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

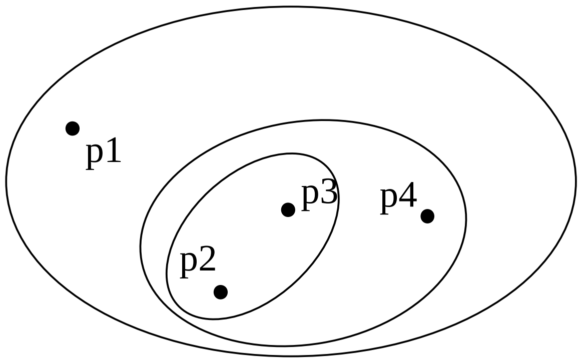


Original Points

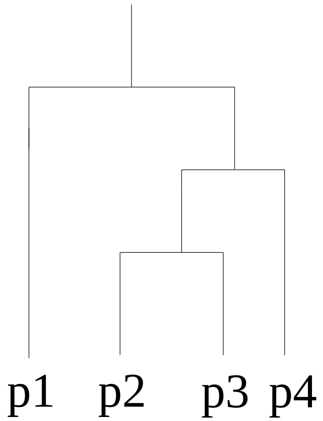


A Partitional Clustering

Types of Clustering: partitional vs hierarchical



Traditional Hierarchical Clustering



Traditional Dendrogram

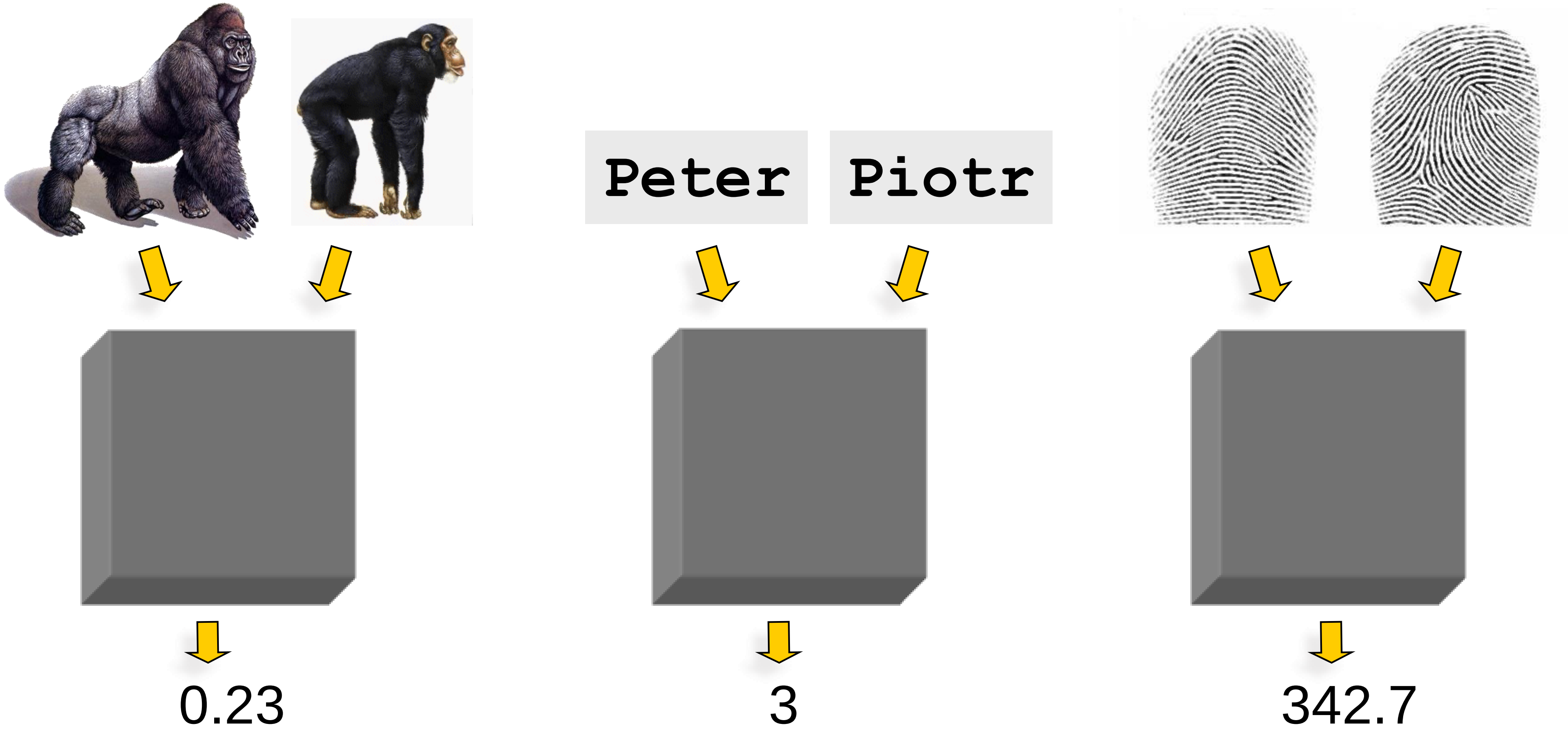
Hierarchical Clustering: A set of nested clusters organized as a hierarchical tree

Types of Clustering

- 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
- Partial versus complete
  - In some cases, we only want to cluster some of the data

Similarity Measures

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



Peter

Piotr

0.23

3

342.7

Properties of distance measure

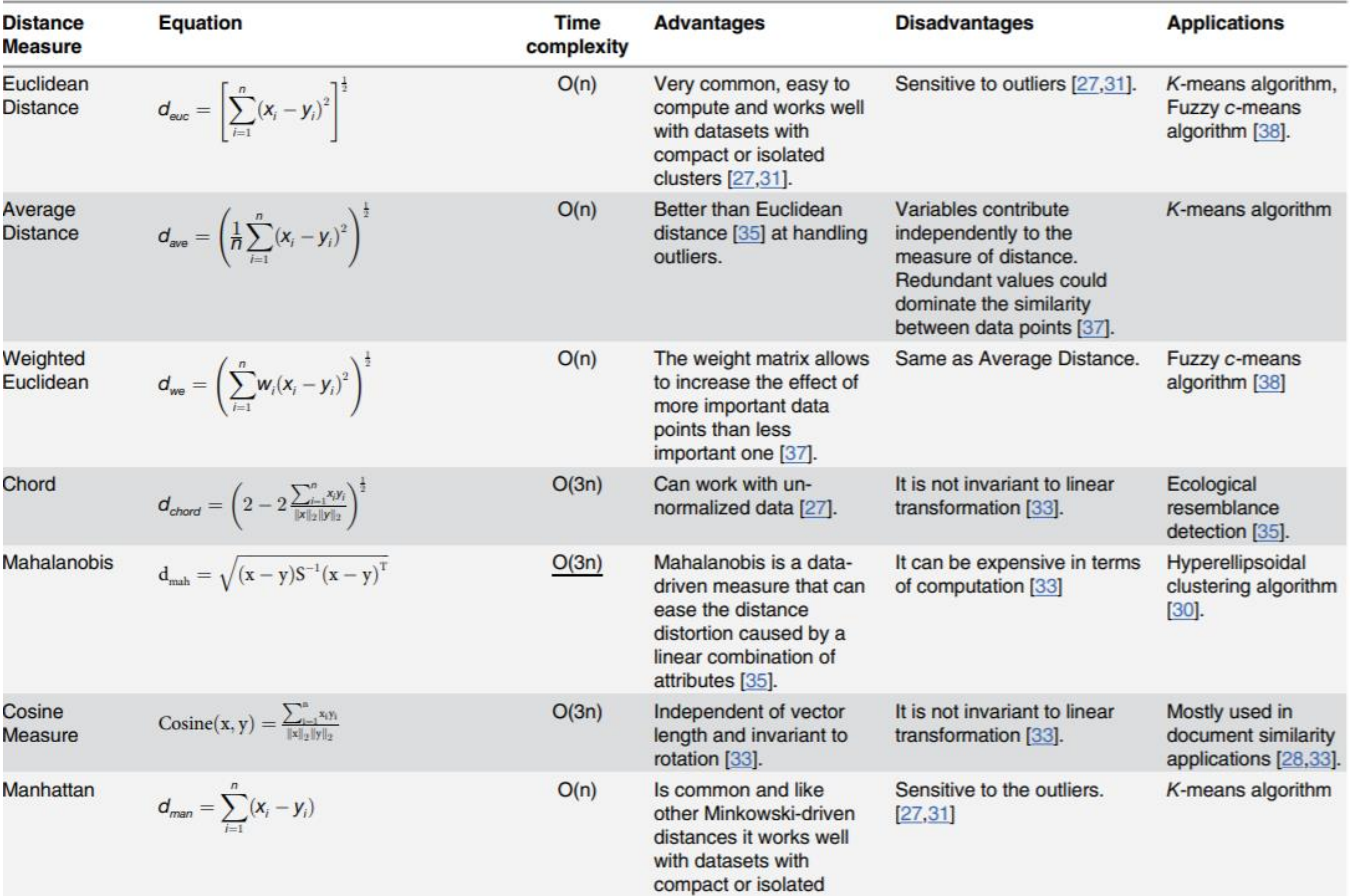
• $D(A,B) = D(B,A)$	*Symmetry*
• $D(A,A) = 0$	*Constancy of Self-Similarity*
• $D(A,B) = 0$ iif $A= B$	*Positivity (Separation)*
• $D(A,B) \leq D(A,C) + D(B,C)$	*Triangular Inequality*

$D(A,B) = D(B,A)$

Otherwise "Alex looks like Bob, but Bob looks nothing like Alex."

$D(A,B) \leq D(A,C) + D(B,C)$

Otherwise "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl."



Distance Measure	Equation	Time complexity	Advantages	Disadvantages	Applications
Euclidean Distance	$d_{euc} = \left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{\frac{1}{2}}$	O(n)	Very common, easy to compute and works well with datasets with compact or isolated clusters [27,31].	Sensitive to outliers [27,31].	*K*-means algorithm, Fuzzy *c*-means algorithm [38].
Average Distance	$d_{ave} = \left(\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2\right)^{\frac{1}{2}}$	O(n)	Better than Euclidean distance [35] at handling outliers.	Variables contribute independently to the measure of distance. Redundant values could dominate the similarity between data points [37].	*K*-means algorithm
Weighted Euclidean	$d_{we} = \left(\sum_{i=1}^{n} w_i(x_i - y_i)^2\right)^{\frac{1}{2}}$	O(n)	The weight matrix allows to increase the effect of more important data points than less important one [37].	Same as Average Distance.	Fuzzy *c*-means algorithm [38]
Chord	$d_{chord} = \left(2 - 2\frac{\sum_{i=1}^{n} x_i y_i}{\|x\|_2 \|y\|_2}\right)^{\frac{1}{2}}$	O(3n)	Can work with un-normalized data [27].	It is not invariant to linear transformation [33].	Ecological resemblance detection [35].
Mahalanobis	$\mathrm{d_{mah}} = \sqrt{(\mathrm{x} - \mathrm{y})\mathrm{S}^{-1}(\mathrm{x} - \mathrm{y})^{\mathrm{T}}}$	O(3n)	Mahalanobis is a data-driven measure that can ease the distance distortion caused by a linear combination of attributes [35].	It can be expensive in terms of computation [33]	Hyperellipsoidal clustering algorithm [30].
Cosine Measure	$\mathrm{Cosine}(\mathrm{x}, \mathrm{y}) = \frac{\sum_{i=1}^{n} \mathrm{x_i y_i}}{\|\mathrm{x}\|_2 \|\mathrm{y}\|_2}$	O(3n)	Independent of vector length and invariant to rotation [33].	It is not invariant to linear transformation [33].	Mostly used in document similarity applications [28,33].
Manhattan	$d_{man} = \sum_{i=1}^{n}(x_i - y_i)$	O(n)	Is common and like other Minkowski-driven distances it works well with datasets with compact or isolated	Sensitive to the outliers. [27,31]	*K*-means algorithm

Clustering techniques

- Partitioning
  - k-Means algorithm
  - PAM (k-Medoids algorithm)

- Hierarchical
  - divisive algorithm
  - Agglomerative algorithm

- Density – Based
  - DBSCAN

K-means clustering

k-Means Clustering

- Clustering used to group similar observations and form different groups-
  - Property 1: All the data points in a cluster should be similar to each other.
  - Property 2: The data points from different clusters should be as different as possible
- k-Means clustering algorithm proposed by J. Hartigan and M. A. Wong [1979].
- Partitional Clustering
- centroid-based algorithm(distance-based algorithm)
- Objective : minimize the sum of distances between the points and their respective cluster centroid
- Given a set of *n* distinct objects, the k-Means clustering algorithm partitions the objects into *k* number of clusters such that intracluster similarity is high but the intercluster similarity is low.

k-Means Algorithm

Input: D is a dataset containing *n* objects, *k* is the number of cluster

Output: A set of *k* clusters

Steps:

1. Randomly choose *k* objects from D as the initial cluster centroids.

2. **For** each of the objects in D **do**
   - Compute distance between the current objects and *k* cluster centroids
   - Assign the current object to that cluster to which it is closest.

3. Compute the “cluster centers” of each cluster. These become the new cluster centroids.

4. Repeat step 2-3 until the convergence criterion is satisfied

5. Stop

k-Means Algorithm

1) Objects are defined in terms of set of attributes. $A = \{A_1, A_2, \ldots\ldots, A_m\}$ where each A_i is continuous data type.

2) Distance computation: Any distance such as L_1, L_2 or cosine similarity.

3) Minimum distance is the measure of closeness between an object and centroid.

4) Mean Calculation: It is the mean value of each attribute values of all objects.

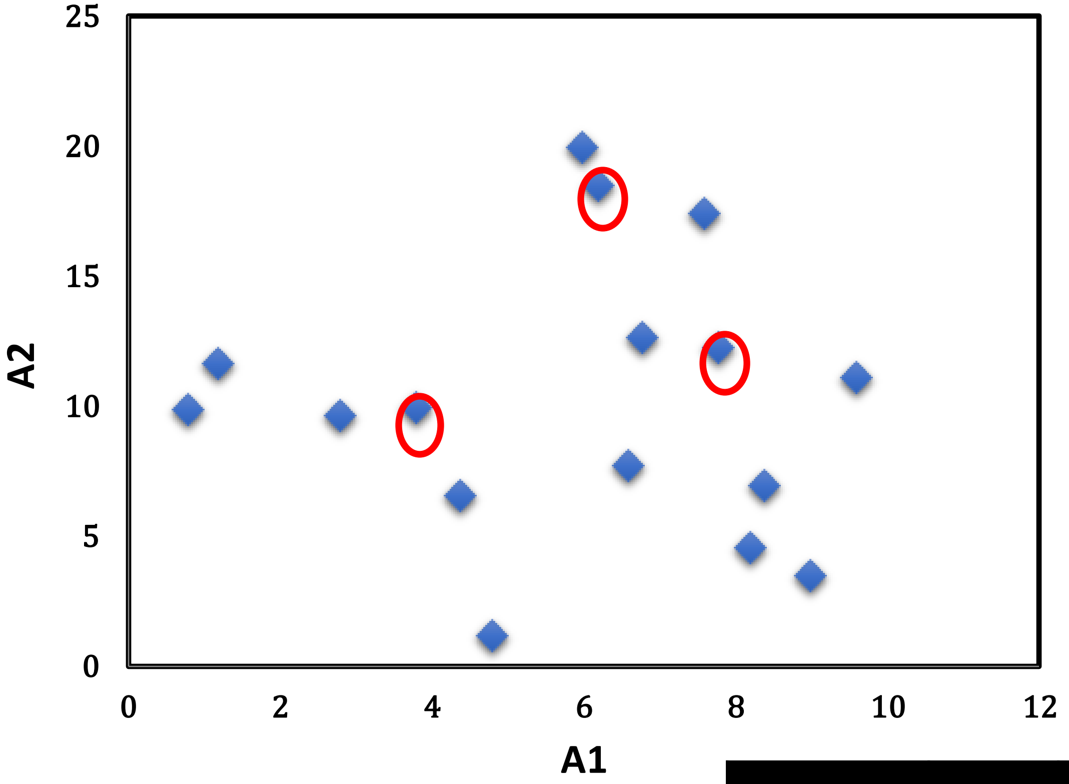
5) Convergence criteria: Any one of the following are termination condition of the algorithm.
   - Number of maximum iteration permissible.
   - No change of centroid values in any cluster.
   - Zero (or no significant) movement(s) of object from one cluster to another.
   - Cluster quality reaches to a certain level of acceptance.

Illustration of k-Means clustering algorithms

objects with two attributes A_1 and A_2.

A_1	A_2
6.8	12.6
0.8	9.8
1.2	11.6
2.8	9.6
3.8	9.9
4.4	6.5
4.8	1.1
6.0	19.9
6.2	18.5
7.6	17.4
7.8	12.2
6.6	7.7
8.2	4.5
8.4	6.9
9.0	3.4
9.6	11.1

Plotting data of Table 16.1



Centroid	Objects	
	A1	A2
c_1	3.8	9.9
c_2	7.8	12.2
c_3	6.2	18.5

Illustration of k-Means clustering algorithms

- Suppose, k=3. Three objects are chosen at random shown as circled (see Fig 16.1). These three centroids are shown below.

Initial Centroids chosen randomly

Centroid	Objects	
	A1	A2
c_1	3.8	9.9
c_2	7.8	12.2
c_3	6.2	18.5

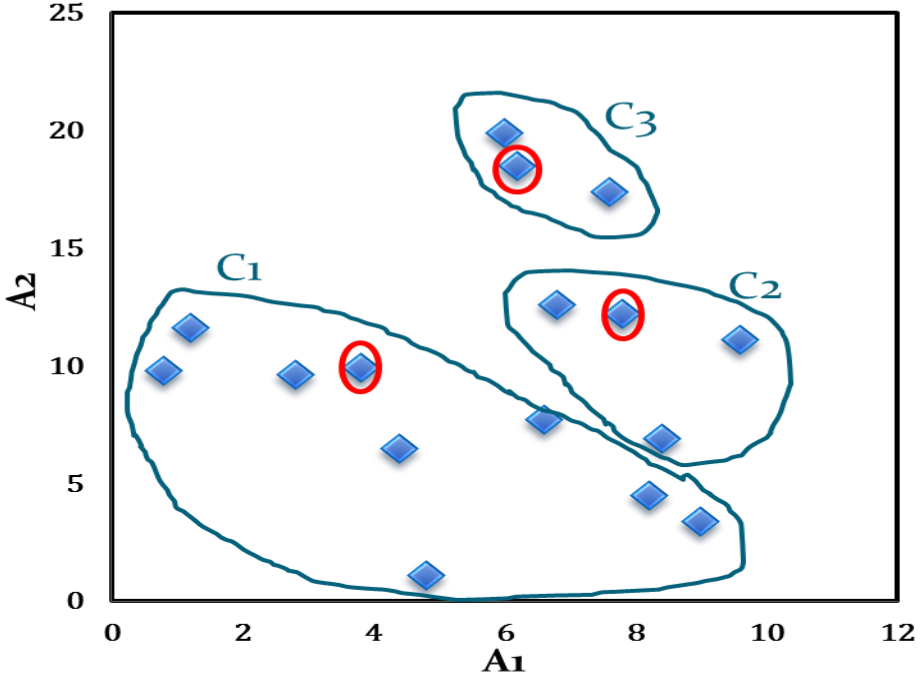
- Let us consider the Euclidean distance measure (L_2 Norm) as the distance measurement in our illustration.
- Let d_1, d_2 and d_3 denote the distance from an object to c_1, c_2 and c_3 respectively. The distance calculations are shown in Table 16.2.
- Assignment of each object to the respective centroid is shown in the right-most column and the clustering so obtained is shown in Fig 16.2.

Illustration of k-Means clustering algorithms

Table 16.2: Distance calculation

A_1	A_2	d_1	d_2	d_3	cluster
6.8	12.6	4.0	1.1	5.9	2
0.8	9.8	3.0	7.4	10.2	1
1.2	11.6	3.1	6.6	8.5	1
2.8	9.6	1.0	5.6	9.5	1
3.8	9.9	0.0	4.6	8.9	1
4.4	6.5	3.5	6.6	12.1	1
4.8	1.1	8.9	11.5	17.5	1
6.0	19.9	10.2	7.9	1.4	3
6.2	18.5	8.9	6.5	0.0	3
7.6	17.4	8.4	5.2	1.8	3
7.8	12.2	4.6	0.0	6.5	2
6.6	7.7	3.6	4.7	10.8	1
8.2	4.5	7.0	7.7	14.1	1
8.4	6.9	5.5	5.3	11.8	2
9.0	3.4	8.3	8.9	15.4	1
9.6	11.1	5.9	2.1	8.1	2

Fig 16.2: Initial cluster with respect to Table 16.2



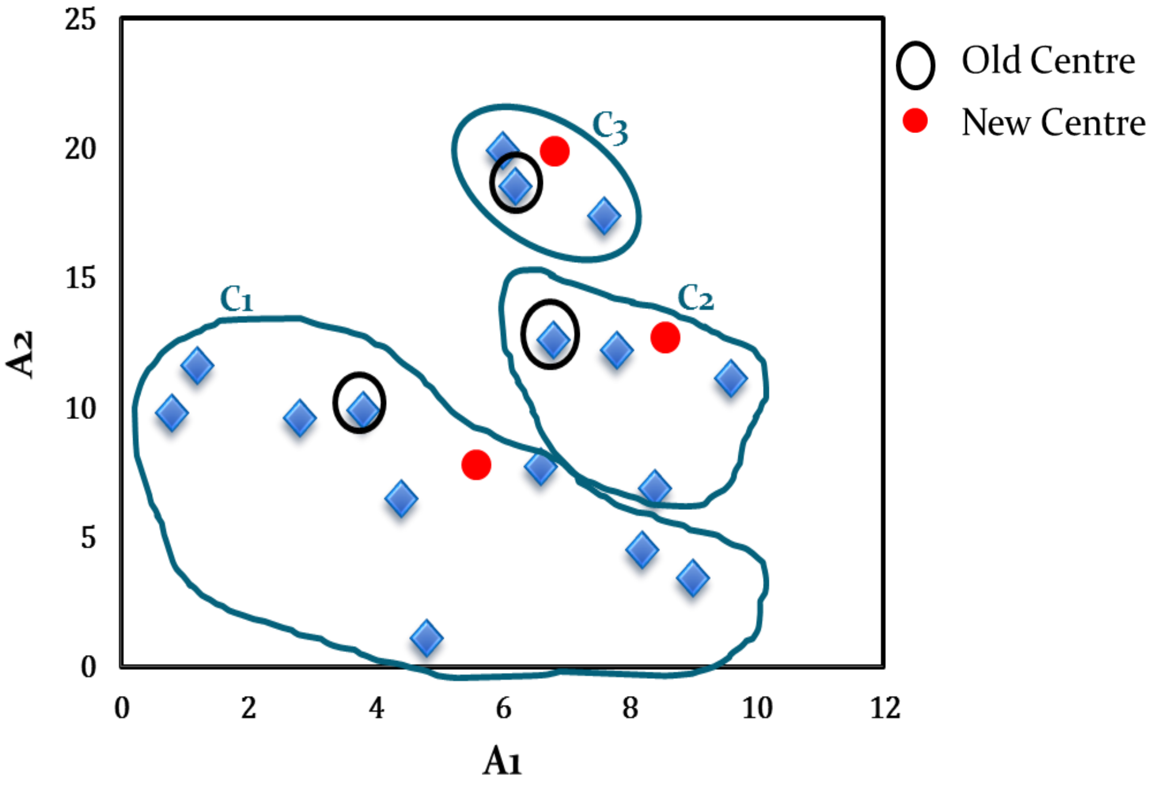
Centroid	Objects	
	A1	A2
c_1	3.8	9.9
c_2	7.8	12.2
c_3	6.2	18.5

Illustration of k-Means clustering algorithms

The calculation new centroids of the three cluster using the mean of attribute values of A_1 and A_2 is shown in the Table below. The cluster with new centroids are shown in Fig 16.3.

Calculation of new centroids

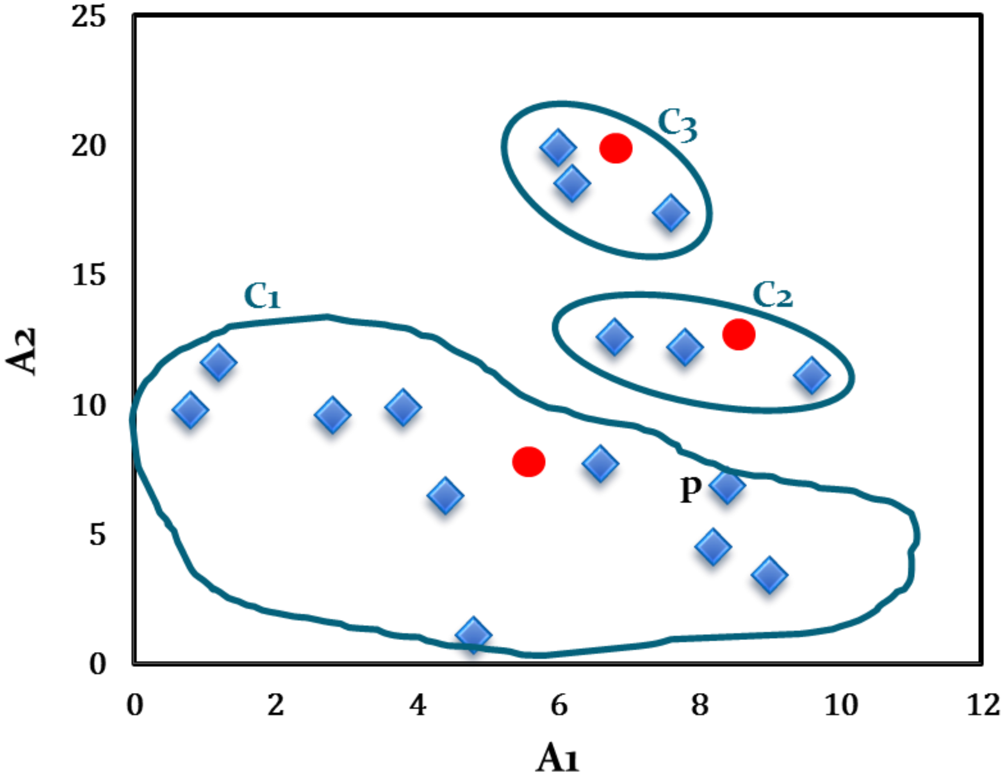
New Centroid	Objects	
	A1	A2
c_1	4.6	7.1
c_2	8.2	10.7
c_3	6.6	18.6



Initial cluster with new centroids

Illustration of k-Means clustering algorithms

Note that point p moves from cluster C_2 to cluster C_1.



Cluster after first iteration

Comments on k-Means algorithm

1. **Value of k:**
   - The k-means algorithm produces only one set of clusters, for which, user must specify the desired number, *k* of clusters.
2. **Choosing initial centroids:**
   - initial choice influences the cluster quality- local optima
   - choose initial centroids in multiple runs, each with a different set of randomly chosen initial centroids, and then select the best cluster
3. **Distance Measurement:**
   - **SSE**
4. **Type of objects under clustering:**
   - The k-Means algorithm can be applied only when the mean of the cluster is defined (hence it named k-Means).

$$c_i = \frac{1}{n_i} \sum_{x \in \boldsymbol{C}_i} x$$

   - mean calculation assumed that each object is defined with numerical attribute

Comments on k-Means algorithm

5. Complexity analysis of k-Means algorithm

Time complexity:

The time complexity of the k-Means algorithm can be expressed as

$$T(n) = O(n \times m \times k \times l)$$

where n = number of objects

m = number of attributes in the object definition

k = number of clusters

l = number of iterations.

Thus, time requirement is a linear order of number of objects and the algorithm runs in a modest time if $k \ll n$ and $l \ll n$ (the iteration can be moderately controlled to check the value of l).

Comments on k-Means algorithm

5. Complexity analysis of k-Means algorithm

Space complexity: The storage complexity can be expressed as follows.

It requires $n \times m$ space to store the objects and $n \times k$ space to store the proximity measure from n objects to the centroids of k clusters.

Thus the total storage complexity is

$$S(n) = O(n \times (m + k))$$

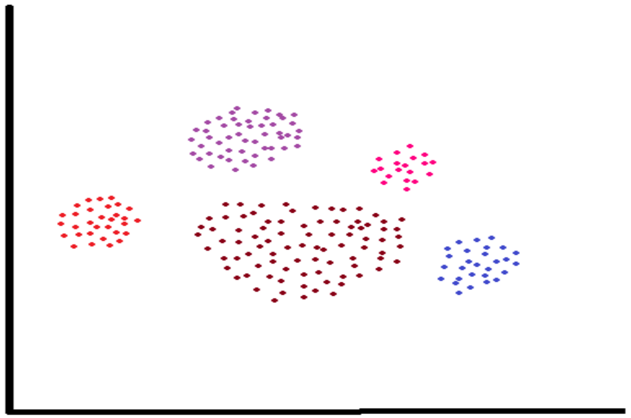
That is, space requirement is in the linear order of n if $k \ll n$.

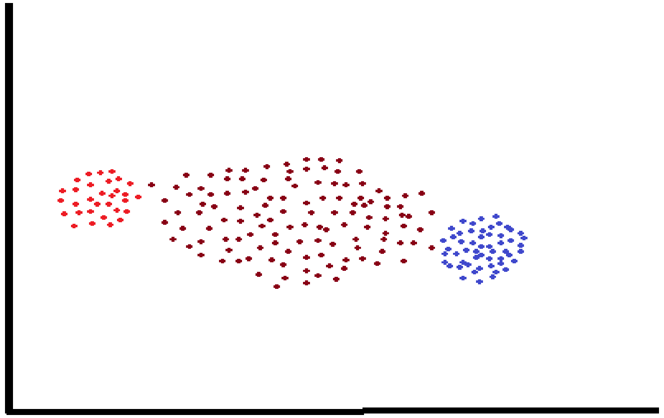
Comments on k-Means algorithm

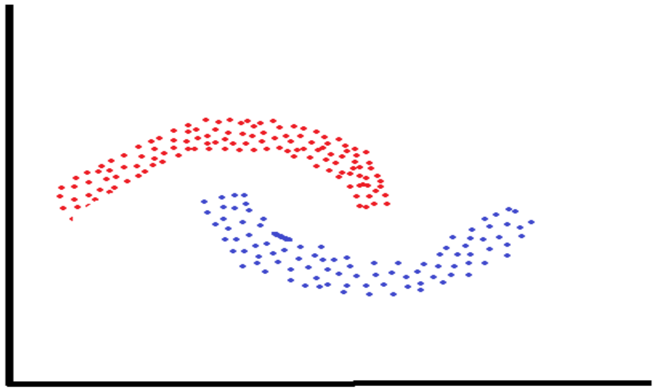
Limitations :

- k-means has trouble clustering data that contains outliers.
- k-Means algorithm cannot handle, clusters of different sizes and densities
- k-Means algorithm cannot handle non-globular clusters

- scalability issue (and not so practical for large databases).







Different variants of k-means algorithm

M. Steinbach, G. Karypis and V. Kumar "A comparison of document clustering techniques", *Proceedings of KDD workshop on Text mining*, 2000.

- B. zhan "Generalised k-Harmonic means – Dynamic weighting of data in unsupervised learning", *Technical report, HP Labs*, 2000.
- A. D. Chaturvedi, P. E. Green, J. D. Carroll, "k-Modes clustering", *Journal of classification*, Vol. 18, PP. 35-36, 2001.
- D. Pelleg, A. Moore, "x-Means: Extending k-Means with efficient estimation of the number of clusters", *17th International conference on Machine Learning*, 2000. N. B. Karayiannis, M. M. Randolph, "Non-Euclidean c-Means clustering algorithm", *Intelligent data analysis journal*, Vol 7(5), PP 405-425, 2003.
- V. J. Olivera, W. Pedrycy, "Advances in Fuzzy clustering and its applications", Edited book. John Wiley [2007]. (Fuzzy c-Means algorithm).
- A. K. Jain and R. C. Bubes, "Algorithms for clustering Data", Prentice Hall, 1988.

Online book at http://www.cse.msu.edu/~jain/clustering_Jain_Dubes.pdf

- A. K. Jain, M. N. Munty and P. J. Flynn, "Data clustering: A Review", *ACM computing surveys*, 31(3), 264-323 [1999]. Also available online.

Summary

- *K*-means clustering is an unsupervised learning algorithm
- The goal of this algorithm is to find clusters in the data given number of clusters
- Iteratively assign each data point to one of *K* groups using a similarity measure
- The results of the *K*-means clustering algorithm are:
  - The centroids of the *K* clusters, which can be used to label new data
  - Labels for the training data
- Fast, robust and easier to understand.
- Relatively efficient

Slides reference: https://cse.iitkgp.ac.in/~dsamanta/courses/da/index.html

K- medoid algorithm

The k-Medoids algorithm

- k-Means algorithm is sensitive to outliers . k-Medoids algorithm aims to diminish the effect of outliers.
- select an object as a cluster center
- cluster representative is called cluster medoid
- Initially, it selects a random set of *k* objects as the set of medoids.
- In each step, all objects from the set of objects, which are not currently medoids are examined one by one to see if they should be medoids.
- The sum-of-absolute error (SAE) function is used as the objective function.

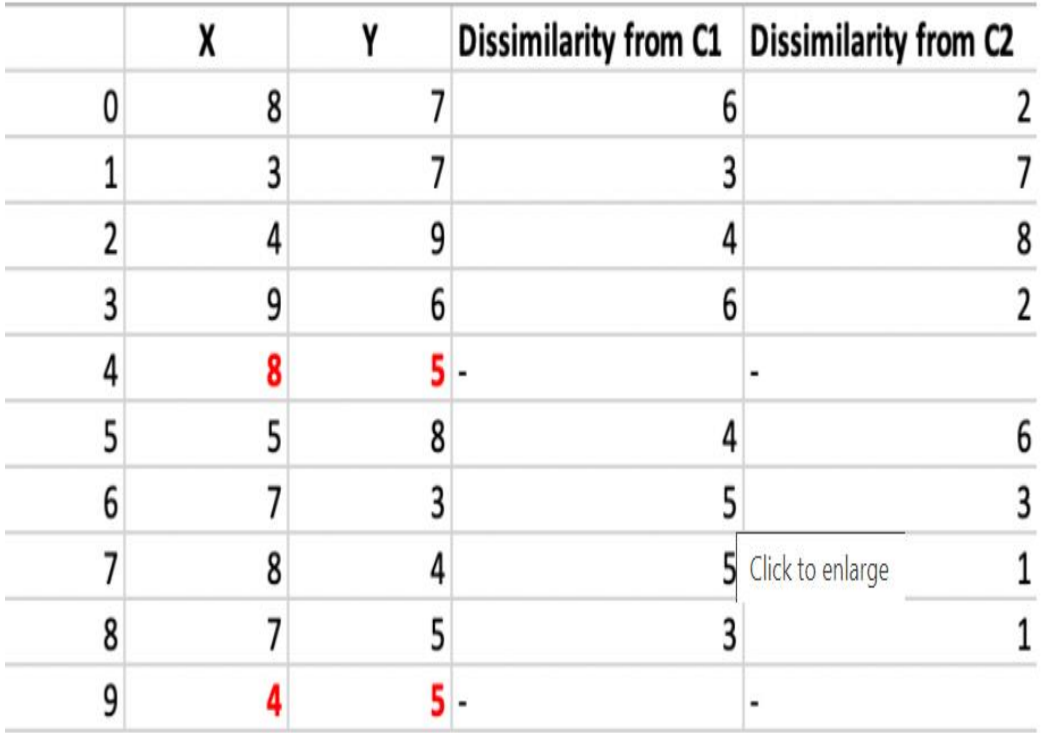
$$SAE = \sum_{i=1}^{k} \sum_{x \in \boldsymbol{C}_i\,, x \notin M \text{ and } c_m \in M} |x - c_m|$$

Where c_m denotes a medoid

M is the set of all medoids at any instant

x is an object belongs to set of non-medoid object, that is, x belongs to some cluster and is not a medoid. i.e. $x \in \boldsymbol{C}_i\,, x \notin M$

Numerical Example



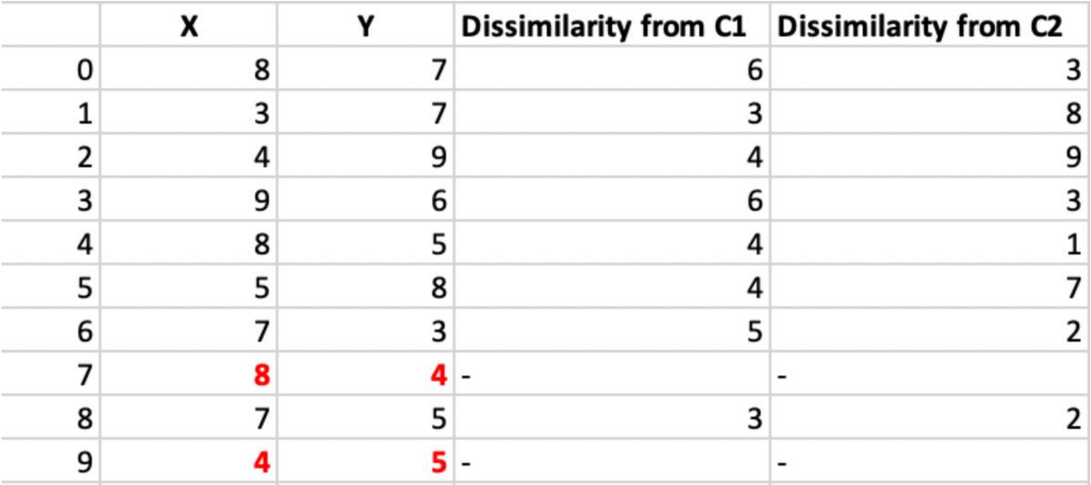
	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	8	5	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	4	5	-	-

Click to enlarge

The points 1, 2, 5 go to cluster c1 and 0, 3, 6, 7, 8 go to cluster c2

Cost = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2) = 20

randomly select one non-medoid point and recalculate the cost. (8, 4)



	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	8	4	-	-
8	7	5	3	2
9	4	5	-	-

New cost = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3) = 22
Swap Cost = New Cost – Previous Cost = 22 – 20 and **2 >0**
As the swap cost is not less than zero, we undo the swap

PAM (Partitioning around Medoids)

Algorithm PAM

Input: Database of objects D.

k, the number of desired clusters.

Output: Set of k clusters

Steps:

```
1. Arbitrarily select k medoids from D.
2. For each object $o_i$ not a medoid do
3.     For each medoid $o_j$ do
4. Let $M = \{o_1, o_2, \ldots, o_{i-1}, o_i, o_{i+1}, o_k\}$     //Set of current medoids
       $M' = \{o_1, o_2, \ldots, o_{j-1}, o_j, o_{j+1}, o_k\}$   //set of medoids but swap with non-medoids $o_j$
5. Calculate $cost(o_i, o_j) = SAE|_M - SAE_{M'}$
6.     End of 2 for loop
```

PAM (Partitioning around Medoids)

7. Find o_i, o_j for which the $\text{cost}(o_i, o_j)$ is the smallest.

8. Replace o_i with o_j and accordingly update the set M.

9. Repeat step 2 - step 8 until cost(o_i, o_i) ≤ 0.

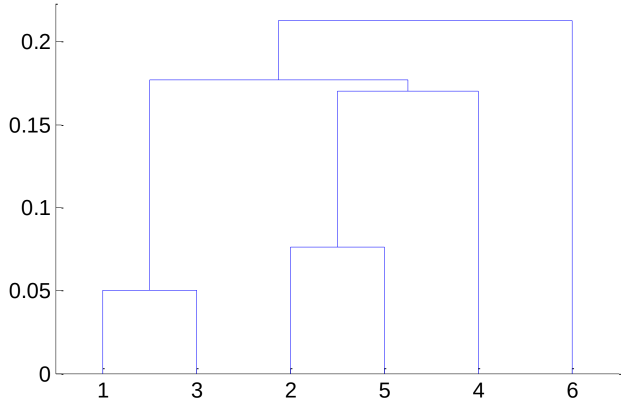
10. Return the cluster with M as the set of cluster centers.

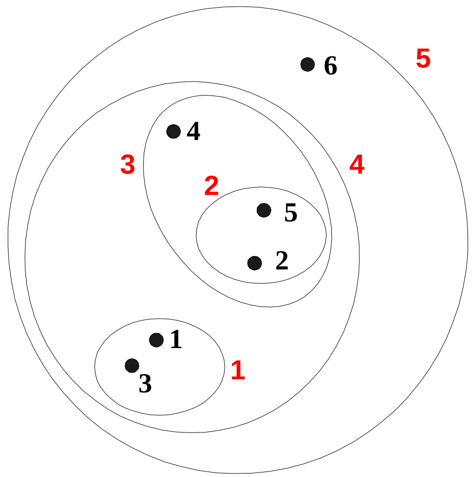
11. Stop

Hierarchical Clustering

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Do not have to assume any particular number of clusters
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits

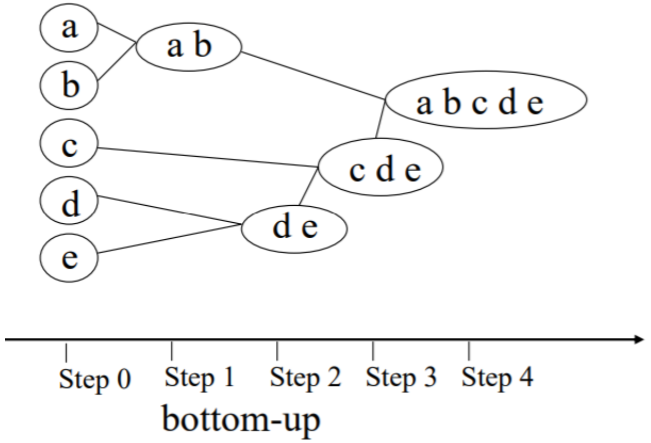




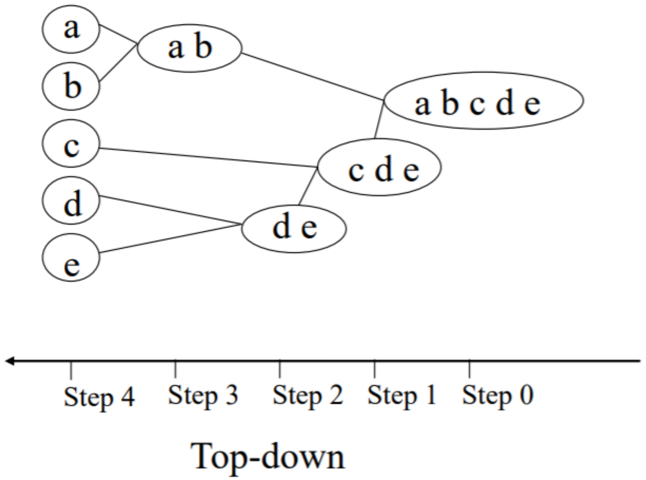
Hierarchical Clustering

- 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(proximity matrix)
  - Merge or split one cluster at a time

- **Agglomerative approach**

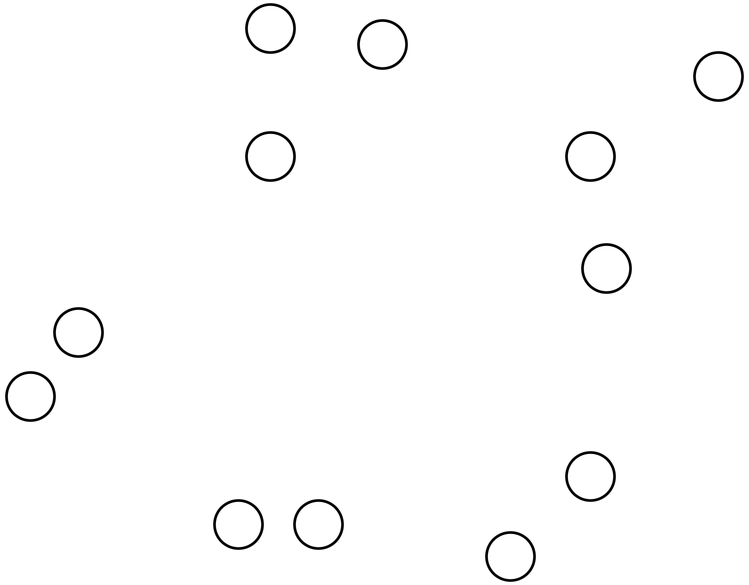


- **Divisive Approaches**



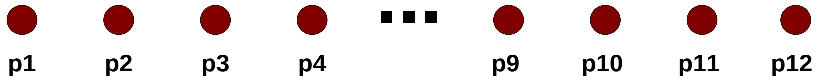
Agglomerative Clustering – Initial setup

- Start with clusters of individual points and a proximity matrix



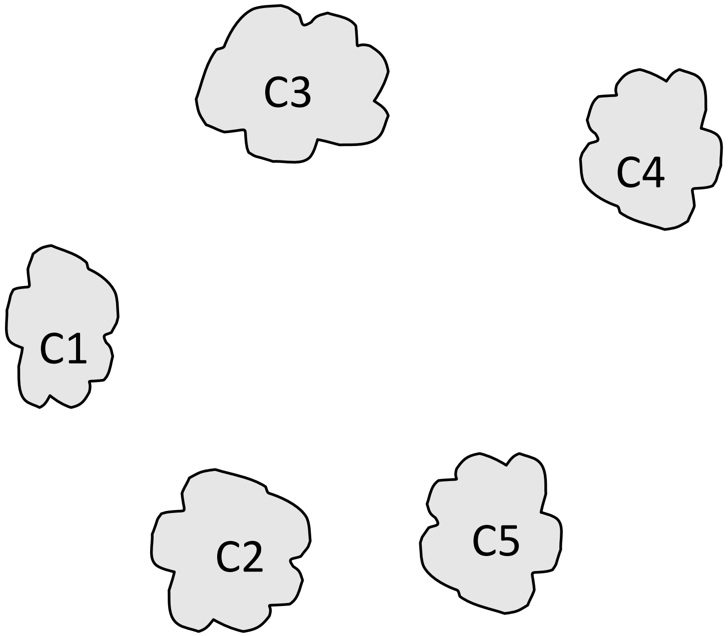
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix



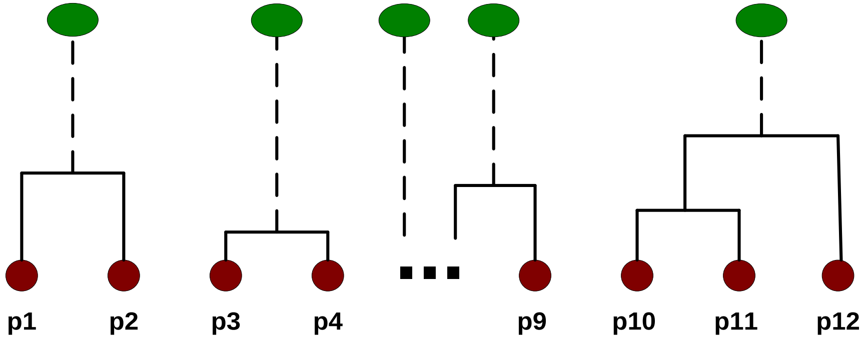
Intermediate Situation

- After some merging steps, we have some clusters



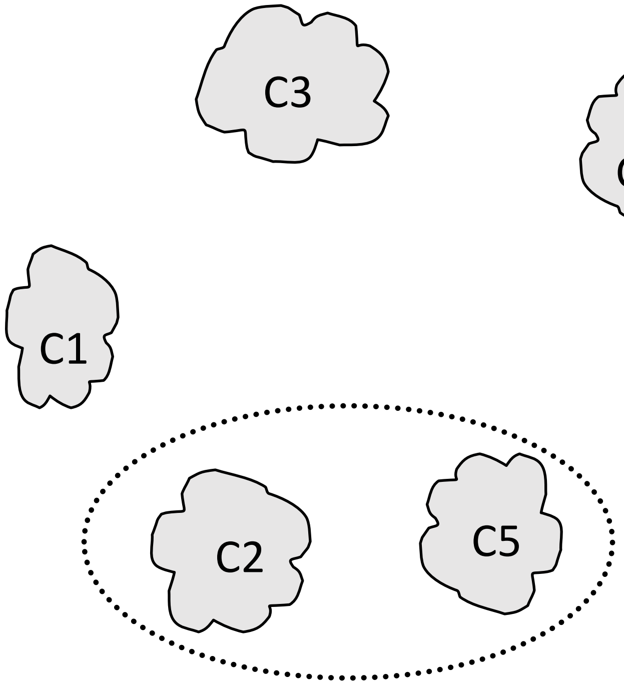
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

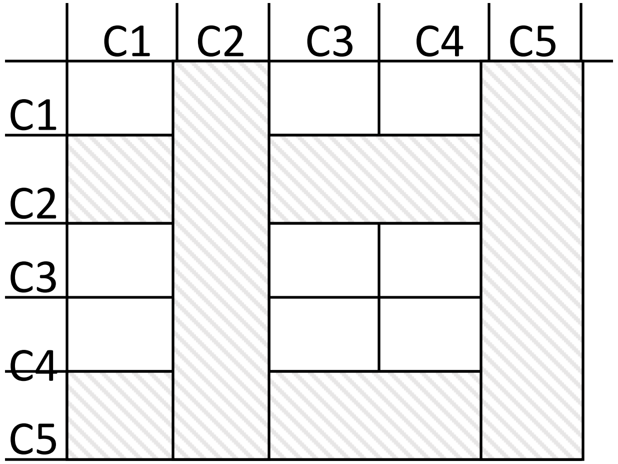
Proximity Matrix



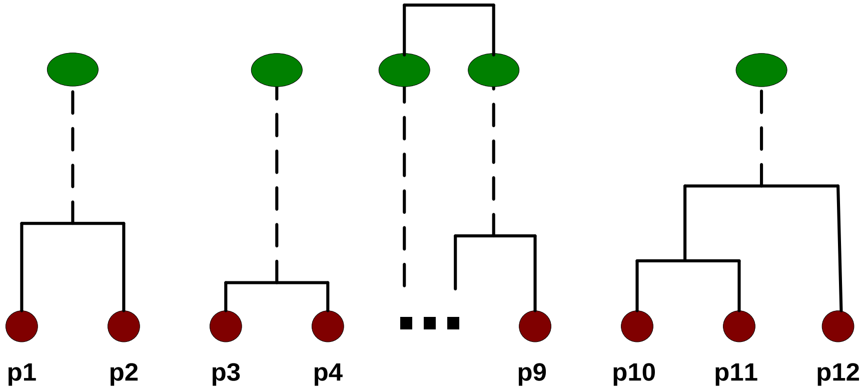
Intermediate Situation

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



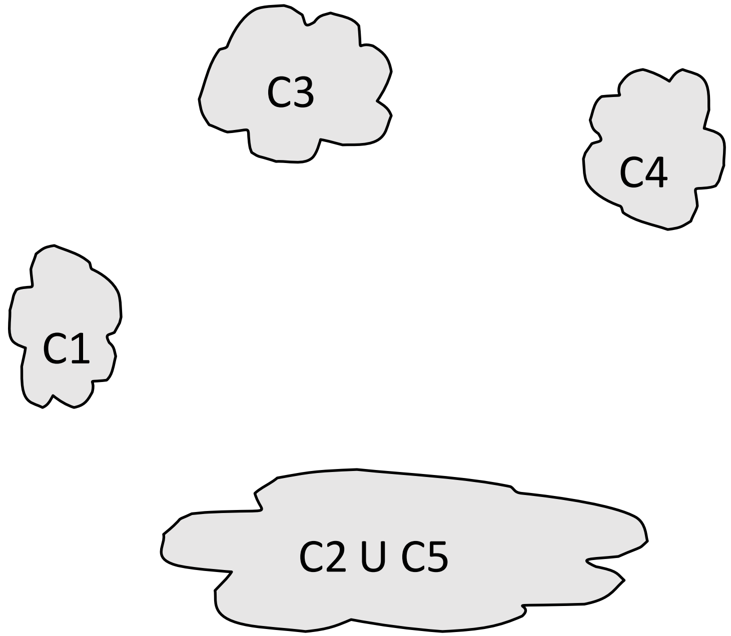


Proximity Matrix



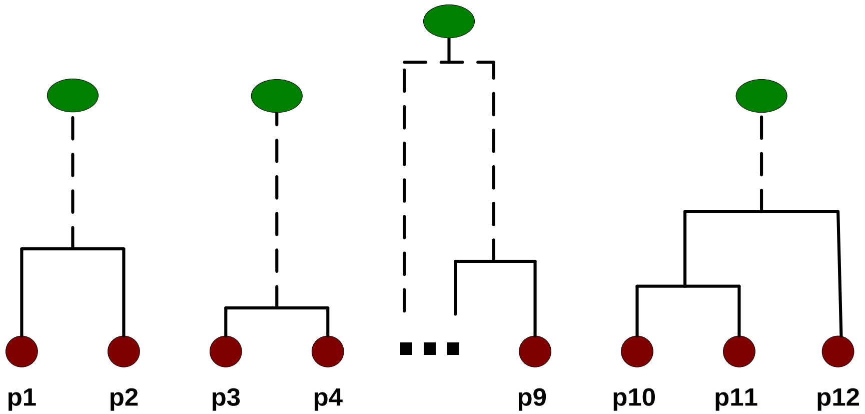
After Merging

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

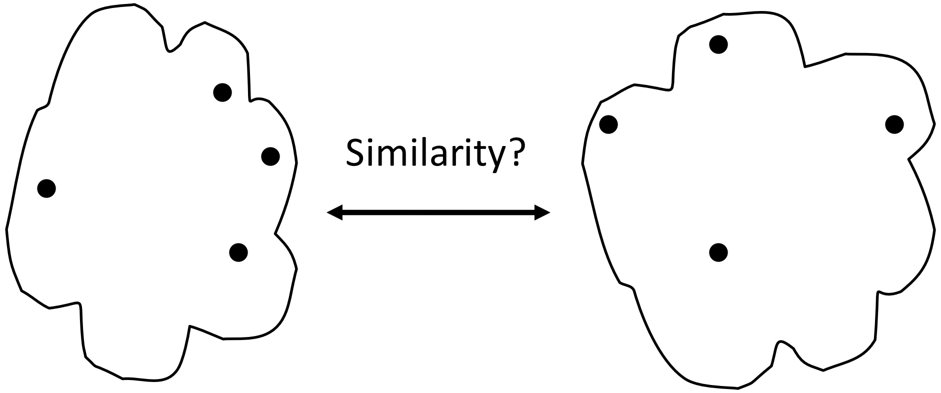


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

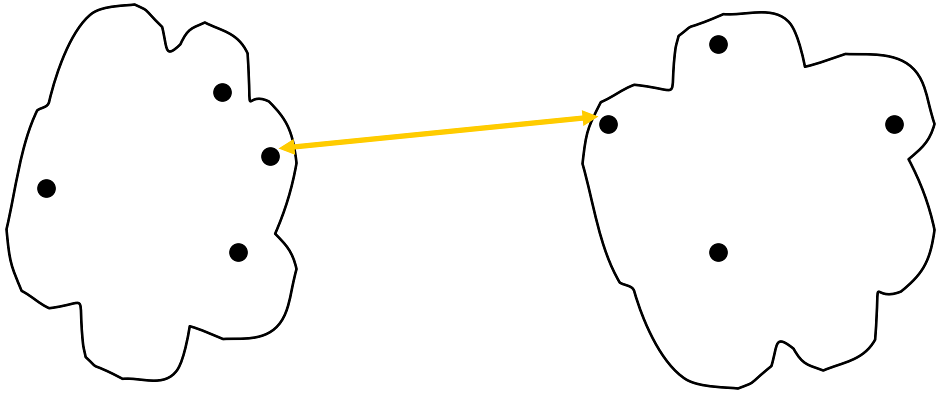


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN (Single Link)
- 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

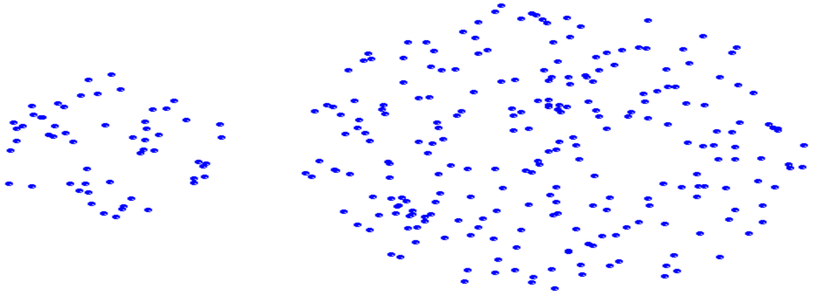


Proximity Matrix

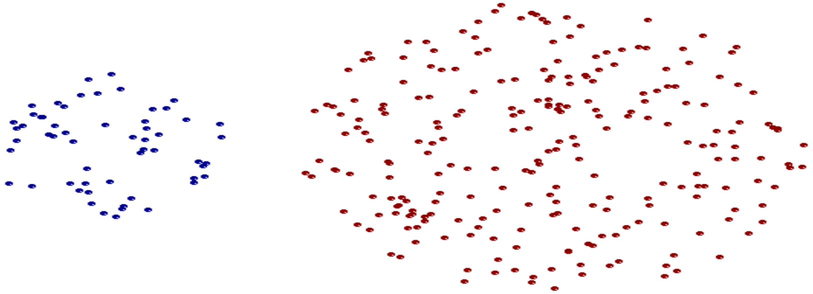
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						

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

Strength of MIN



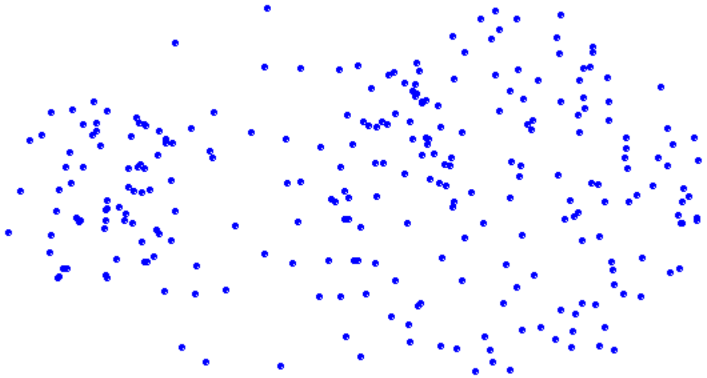
Original Points



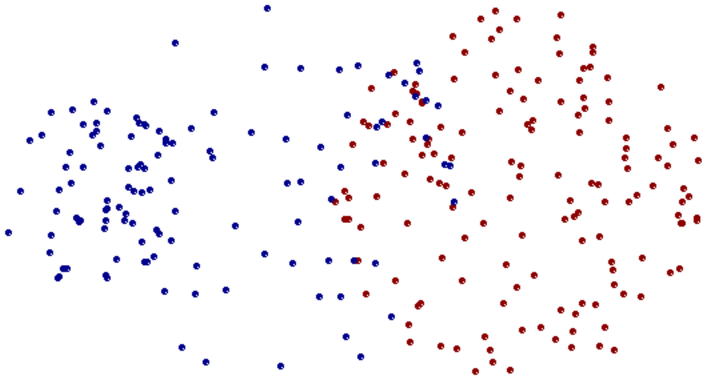
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



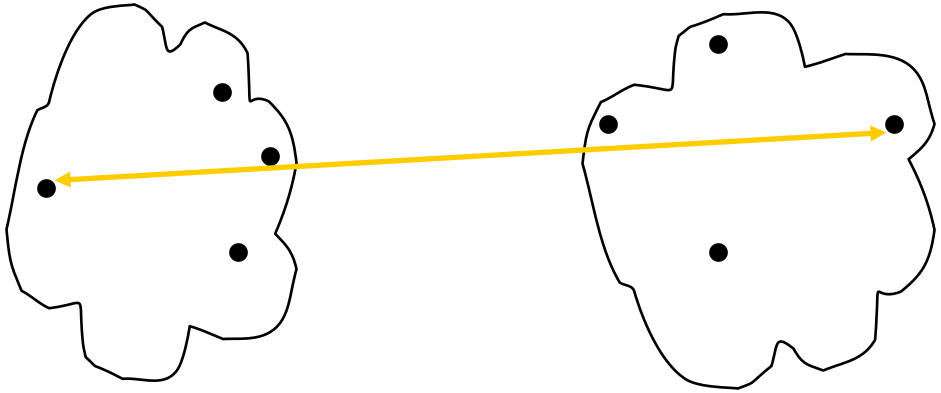
Original Points



Two Clusters

- Sensitive to noise and outliers

How to Define Inter-Cluster Similarity

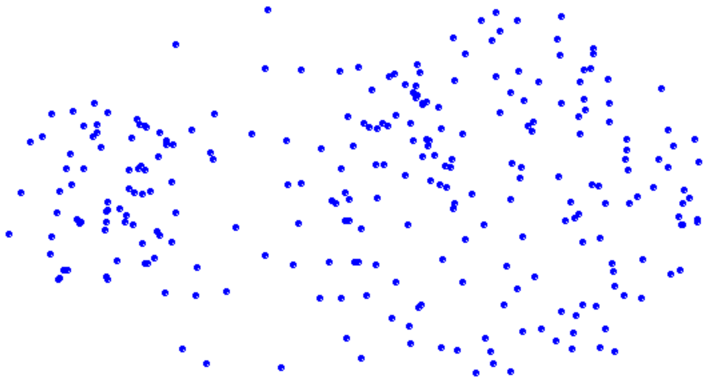


Proximity Matrix

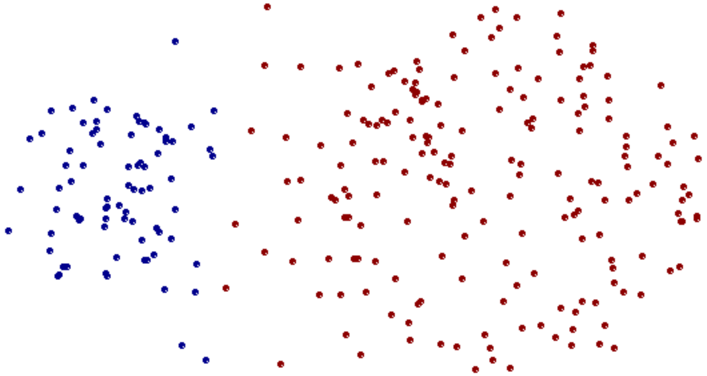
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						

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

Strength of MAX



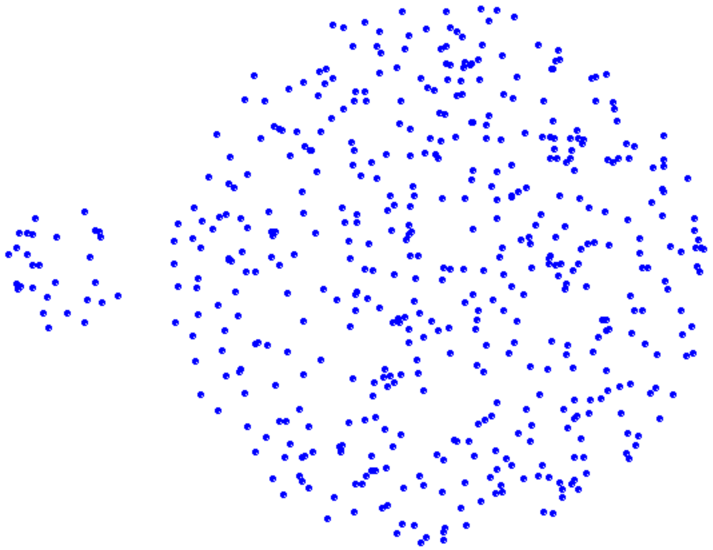
Original Points



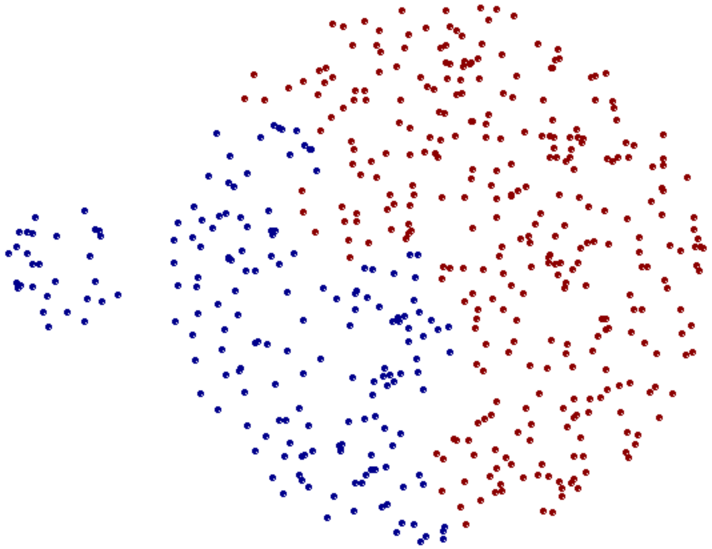
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



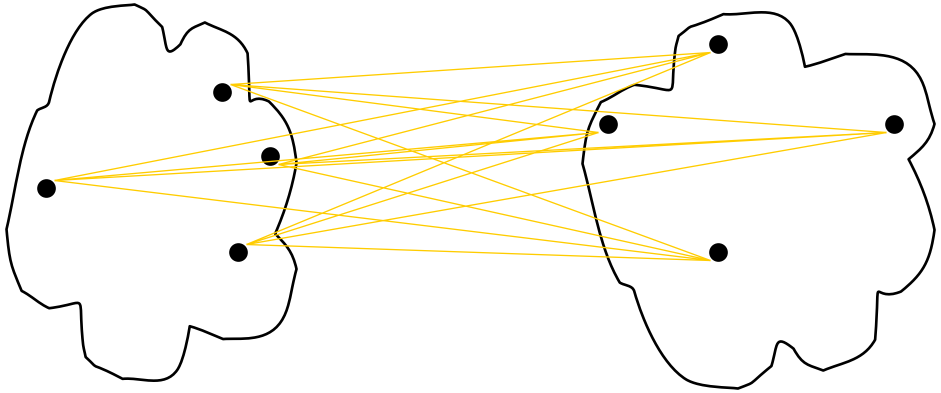
Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

How to Define Inter-Cluster Similarity



Proximity Matrix

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

- MIN
- MAX
- Group Average Proximity of two clusters is the average of pairwise proximity between points in the two clusters
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

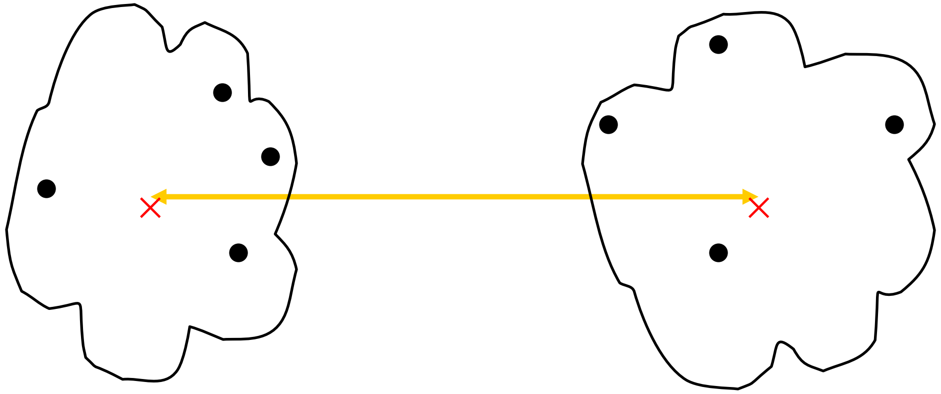
Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link

- Strengths
  - Less susceptible to noise and outliers

- Limitations
  - Biased towards globular clusters

How to Define Inter-Cluster Similarity



Proximity Matrix

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

.

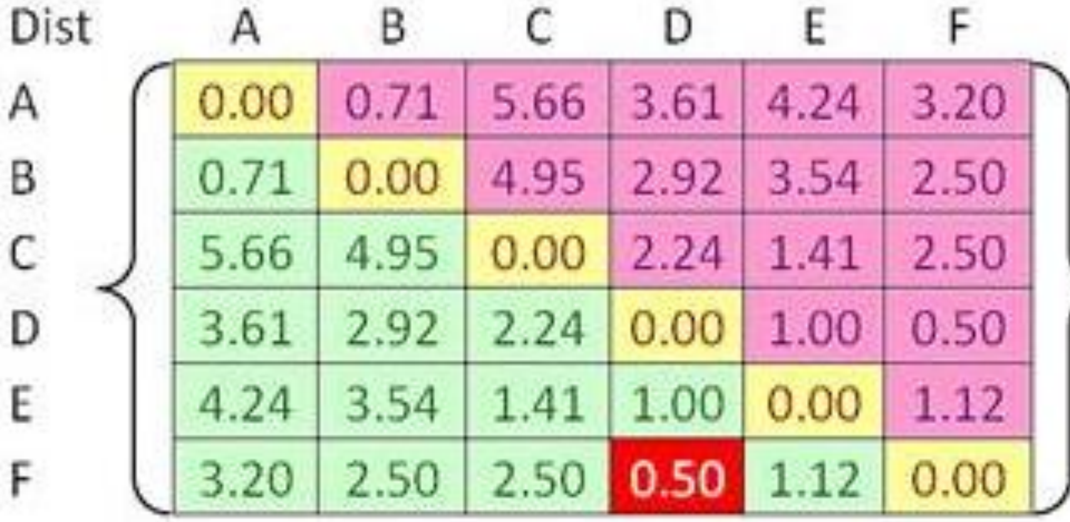
.

- MIN
- MAX
- Group Average
- Distance Between Centroids

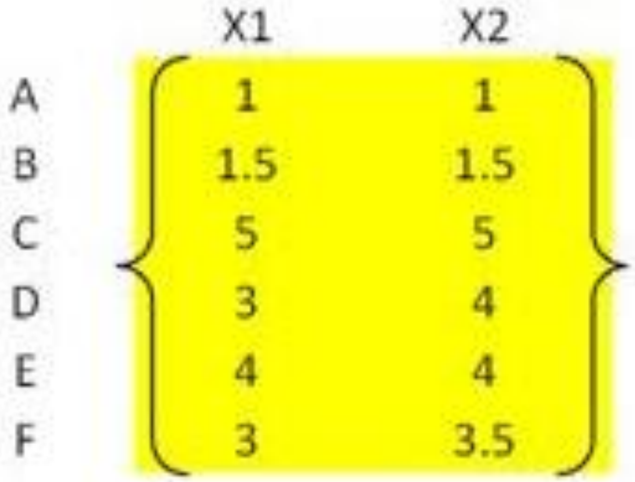
xample

- Given an input distance matrix based on object features
- We have 6 data points, we put each one into one cluster
- Our goal is to group those 6 clusters such that
  - at the end of the iterations, we will have only single cluster consists of the whole six original objects.

- The closest cluster is between cluster F and D with shortest distance of 0.5.
  - we group cluster D and F into cluster (D, F).
  - Then we update the distance matrix



Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

Example

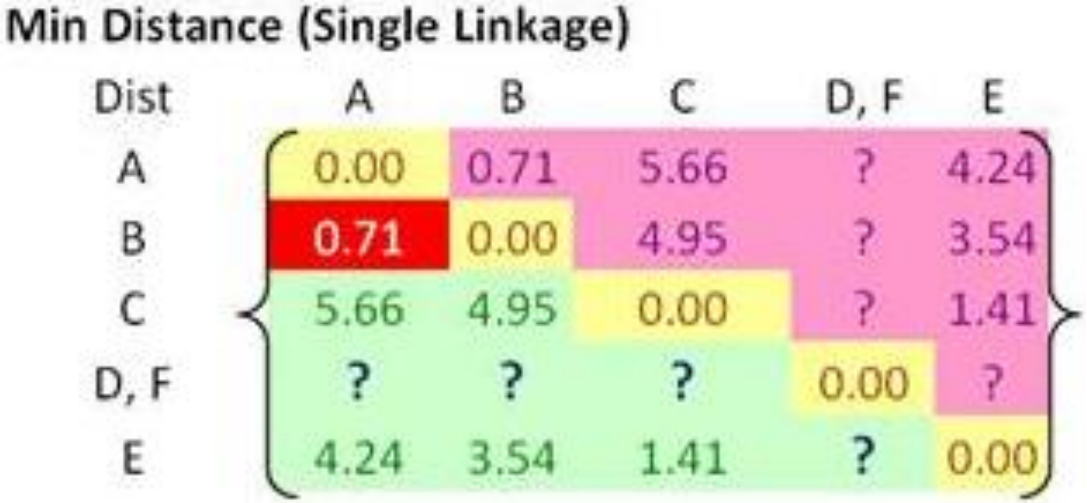
- Using single linkage, we specify minimum distance between original objects of the two clusters.

$$d_{(D,F)\mapsto A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\mapsto B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

$$d_{(D,F)\mapsto C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$



Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

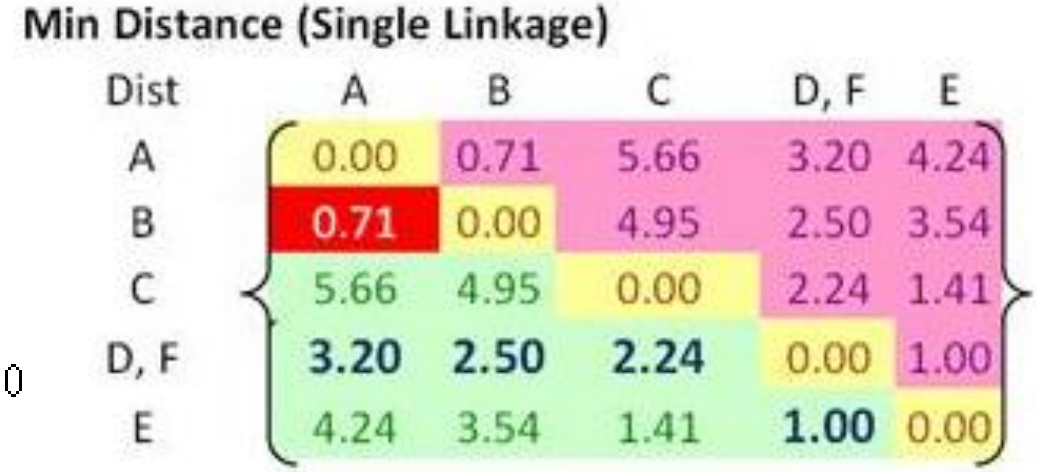
Example

- The closest distance between B and A is minimum.
- We group cluster A and cluster B into a sir

$$d_{C\rightarrow(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

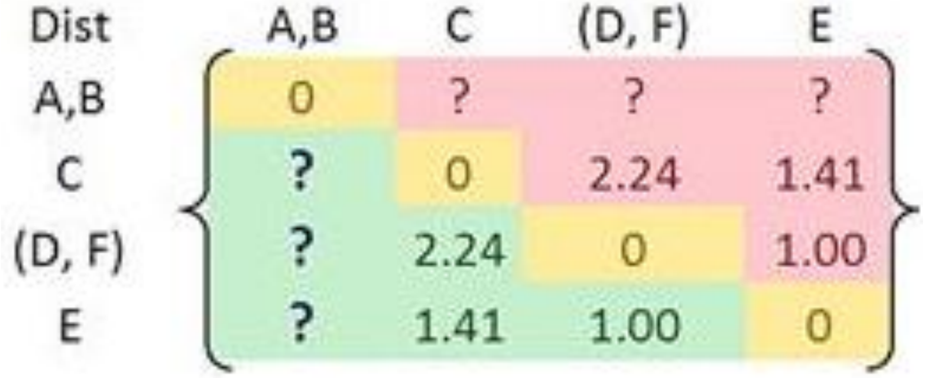
$$d_{(D,F)\rightarrow(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right) = \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

$$d_{E\rightarrow(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$



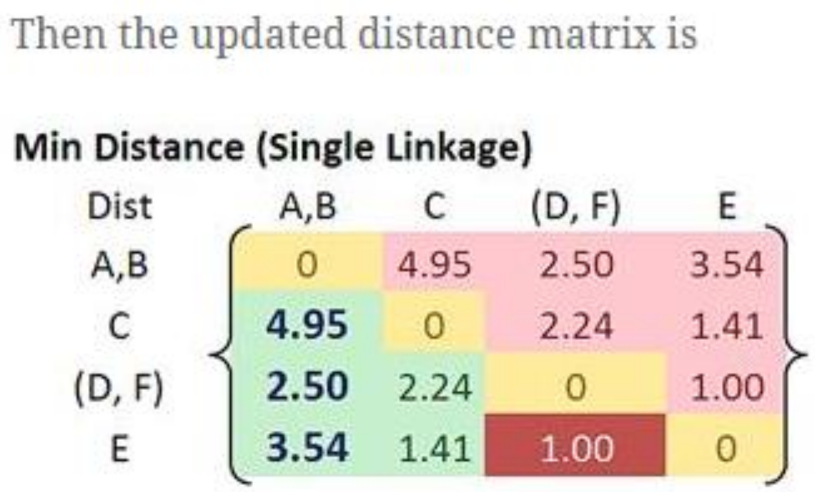
Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00



Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

Example

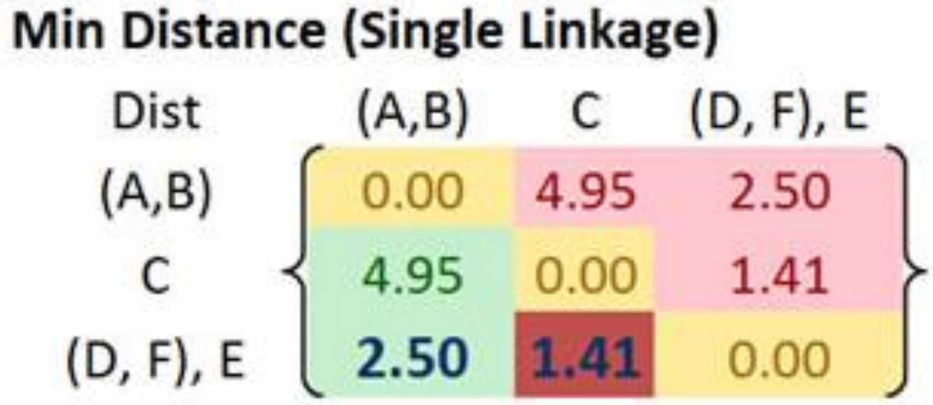


Then the updated distance matrix is

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

- We cluster them together into



Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

$$d_{((D,F),E)\mapsto(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}, d_{EA}, d_{EB}\right) = \min\left(3.61, 2.92, 3.20, 2.50, 4.24, 3.54\right) = 2.50$$

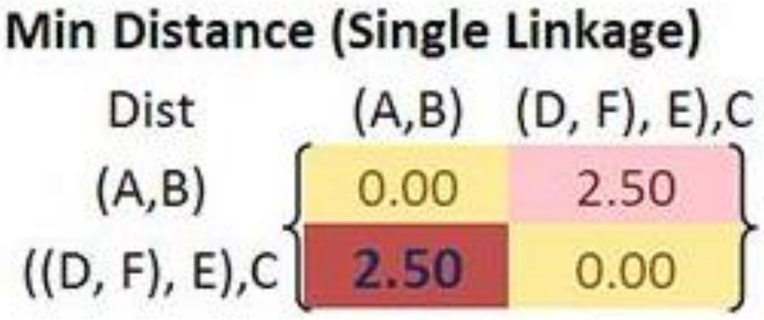
$$d_{((D,F),E)\mapsto C} = \min\left(d_{DC}, d_{FC}, d_{EC}\right) = \min\left(2.24, 2.50, 1.41\right) = 1.41$$

Example

- we merge cluster ((D, F), E) and cluster C into a new cluster name (((D, F), E), C).

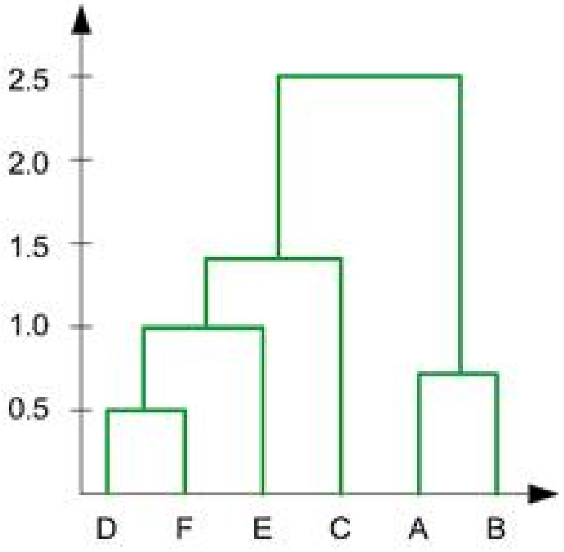
$$d_{(((D,F),E),C)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}, d_{EA}, d_{EB}, d_{CA}, d_{CB}\right)$$

$$d_{(((D,F),E),C)\to(A,B)} = \min\left(3.61, 2.92, 3.20, 2.50, 4.24, 3.54, 5.66, 4.95\right) = 2.50$$

- 

Min Distance (Single Linkage)

Dist	(A,B)	(D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	2.50	0.00



Hierarchical Clustering: Problems and Limitations

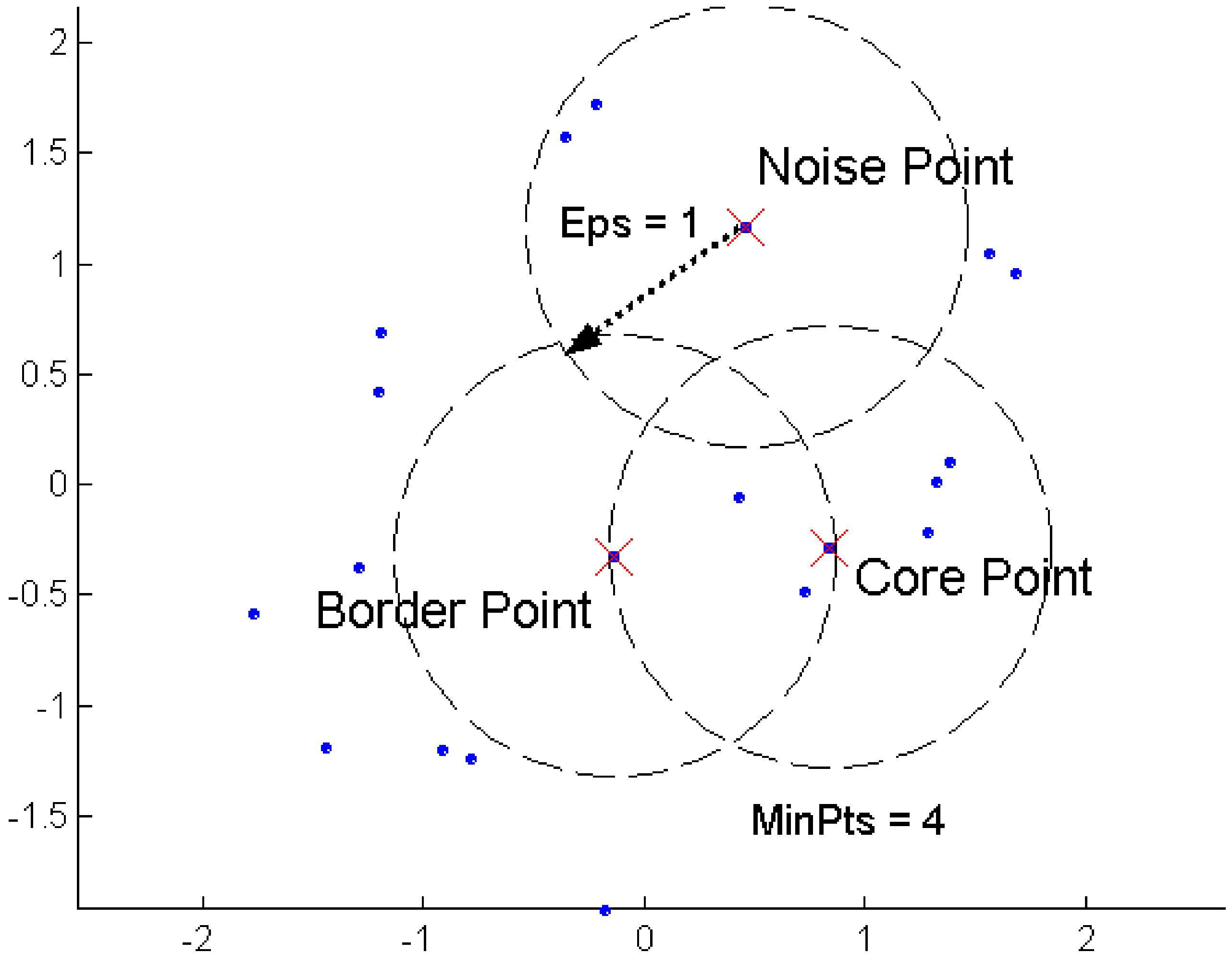
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking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

Density Based Clustering

DBSCAN

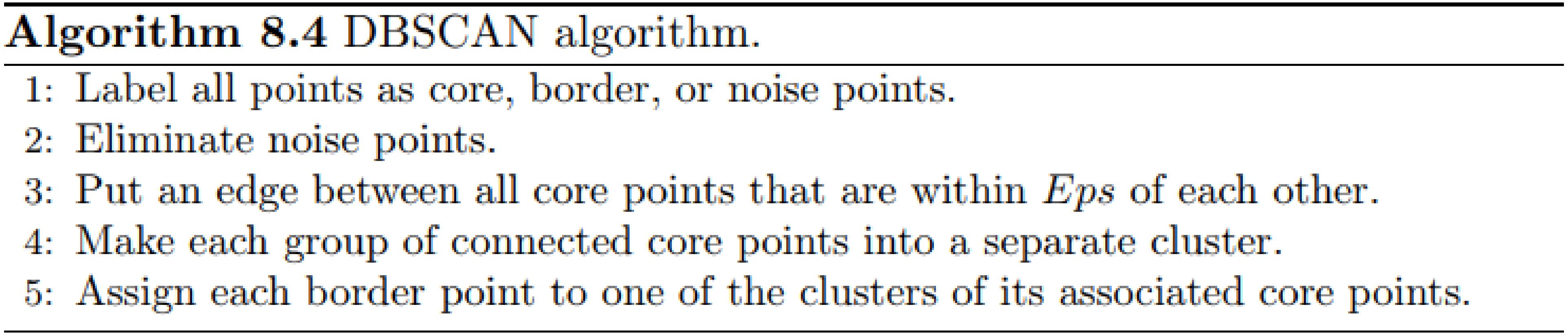
- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
  - A border point has fewer than MinPts 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 Algorithm

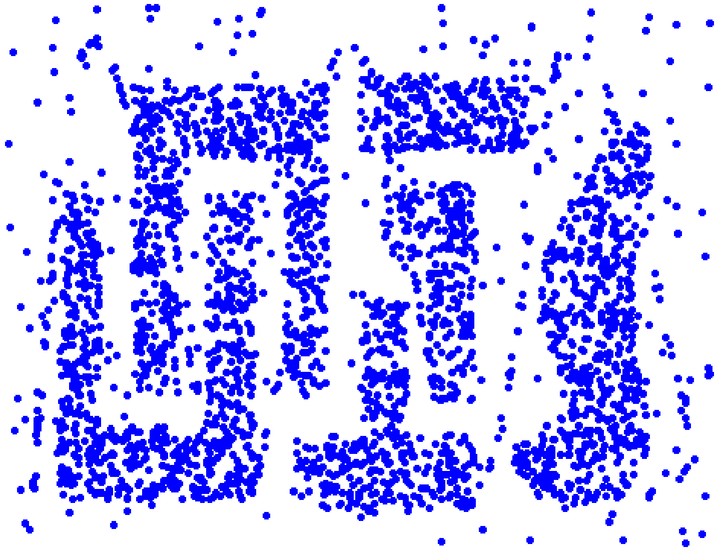
- Eliminate noise points
- Perform clustering on the remaining points



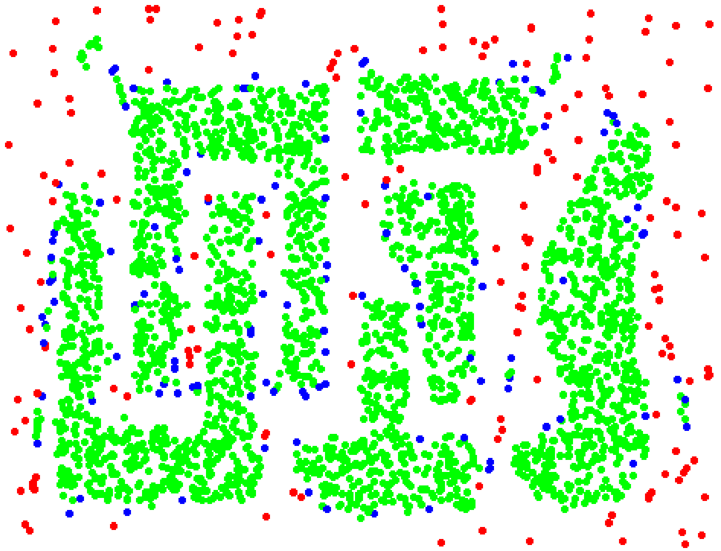
Algorithm 8.4 DBSCAN algorithm.

```
Label all points as core, border, or noise points.
Eliminate noise points.
Put an edge between all core points that are within $Eps$ of each other.
Make each group of connected core points into a separate cluster.
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 Does NOT Work Well

- Varying densities
- High-dimensional data