Clustering

What is Cluster Analysis?

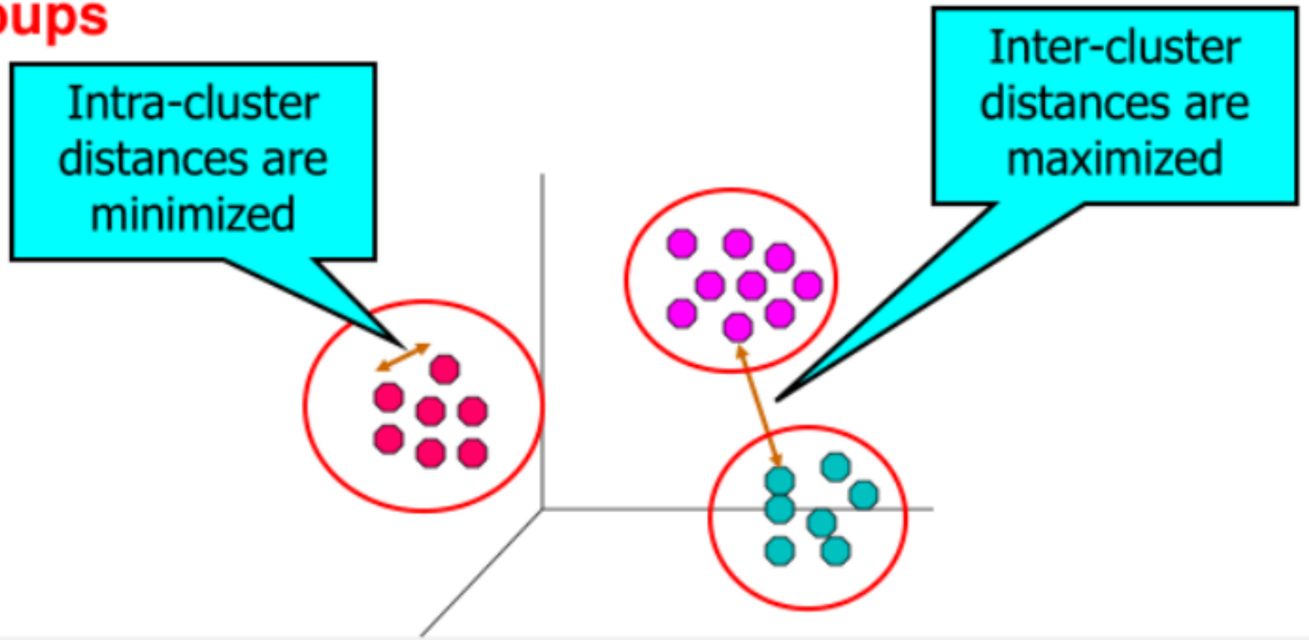
- Different than *prediction*…
- Dividing data into groups (clusters) in some meaningful or useful way

- Clustering should capture "natural structure of the data"

Unsupervised learning: no predefined classes

What is Cluster Analysis?

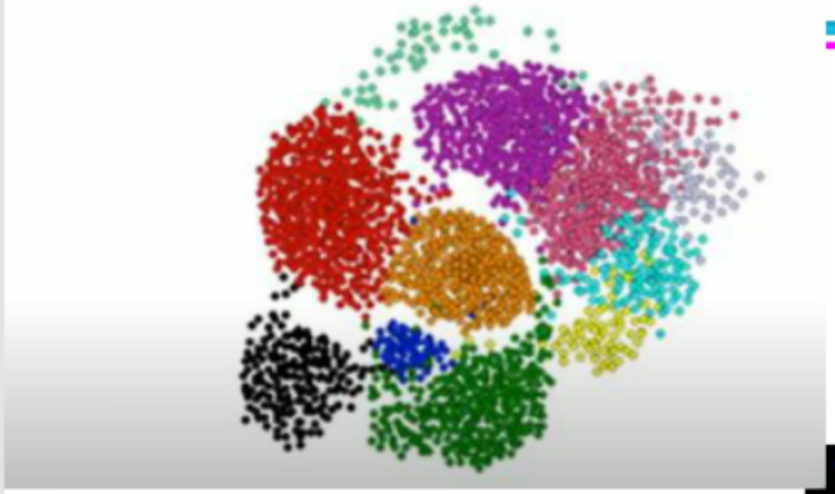
- Cluster Analysis: Automatically finding classes
- Finding groups of objects such that the **objects in a group will be similar (or related)** to one another and **different from (or unrelated to) the objects in other groups**



Goal:

1. **Increasing the similarity/homogeneity within a group**
2. **Increasing the difference/distance between groups.**

Clustering is a hard problem!



Why is it hard?

- Clustering in two dimensions looks easy
- Clustering small amounts of data looks easy
- And in most cases, looks are *not* deceiving

- Many applications involve not 2, but 10 or 10,000 dimensions
- **High-dimensional spaces look different:** Almost all pairs of points are at about the same distance

Applications of Cluster Analysis

- Group related **documents** for browsing, patients with similar **symptoms**, people with same **interests**/connections on social media, group **genes** and proteins that have similar functionality, or group stocks with **similar price** fluctuations

- Business
  - Businesses collect large amounts of information on current and potential customers.
  - Clustering to segment customers into a small number of groups, for additional analysis and marketing activities.

Every cluster is characterized in terms of cluster Prototype i.e. data object that is representative of other objects in the class.

Cluster Analysis

- **Applications: As a preprocessing step for other algorithms**
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing, Vector quantization
  - Finding K-nearest Neighbors: Localizing search to one or a small number of clusters
  - Outlier detection: Outliers are often viewed as those “far away” from any cluster

Cluster Analysis as Unsupervised Learning

- Supervised learning: Discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in unseen data instances.
  - The set of classes is known before.
  - Training data is often provided by human annotators.
- Unsupervised learning: The data has no target attribute.
  - We want to explore the data to find some intrinsic structures in it.
  - The set of classes/clusters is not known before.
  - No training data is used.
- Cluster Analysis is an unsupervised learning task.

Relation of Clustering to Classification

- Clustering can be regarded as a form of classification
  - Creating a labeling of objects with cluster (class) labels
  - But…these labels are derived exclusively from the data.
  - Cluster analysis is sometimes referred to as unsupervised classification
    - No model from training data with class labels

Iris Example

- With Decision Trees (supervised classification):
  - "Training set" has class labels:

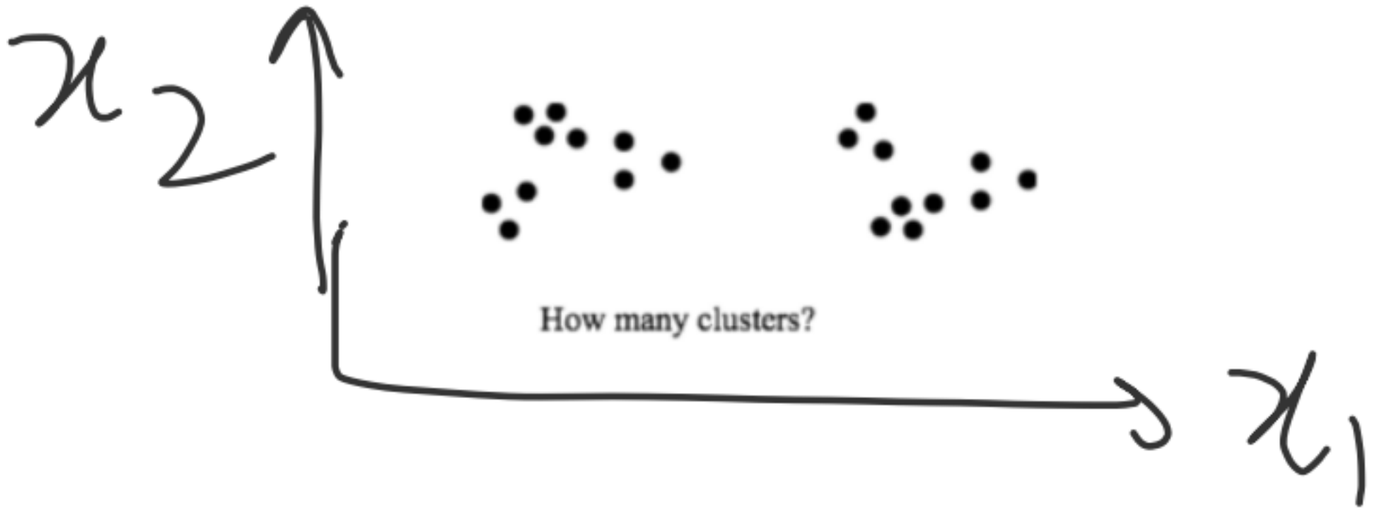
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4.9	3.0	1.4	0.2	setosa
4.6	3.1	1.5	0.2	setosa
6.7	3.1	4.4	1.4	versicolor
6.4	2.8	5.6	2.2	virginica

- With Clustering (unsupervised classification):
  - Only data

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
4.9	3.0	1.4	0.2
4.6	3.1	1.5	0.2
6.7	3.1	4.4	1.4
6.4	2.8	5.6	2.2

Notion of a Cluster can be Ambiguous

- The notion of a cluster may not be well defined.



Notion of a Cluster can be Ambiguous

How many clusters?

By the human visual system, it looks like two clusters.

Two Clusters

But it really depends on the characteristics of the data.
These clusterings may not be unreasonable:

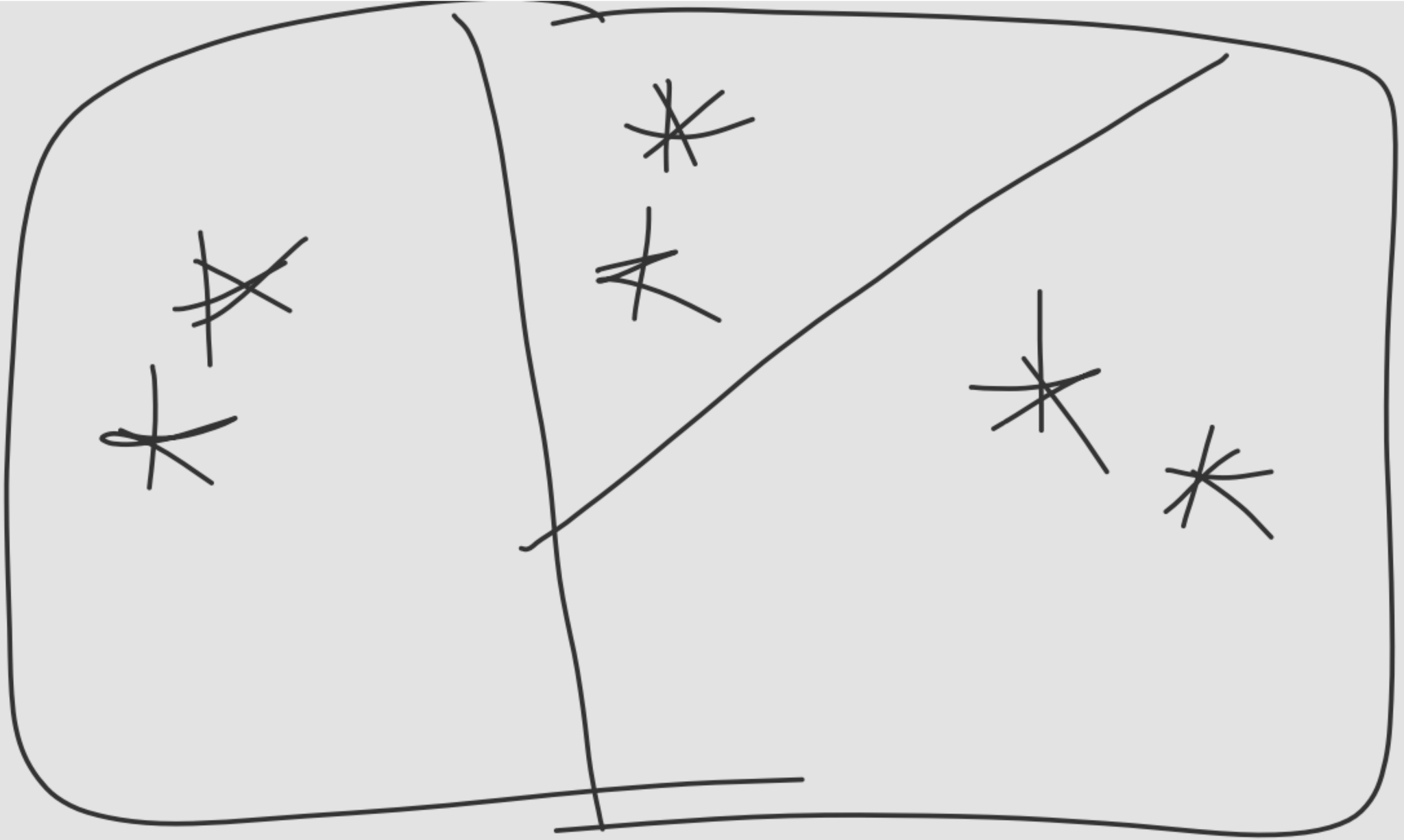
Six Clusters

Four Clusters

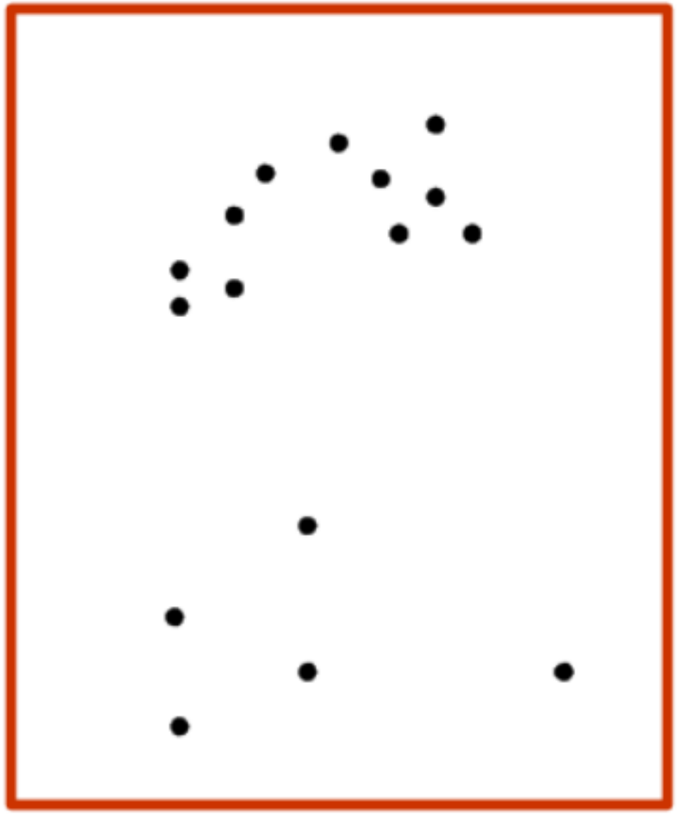
Types of Clusterings

- A clustering is a set of clusters

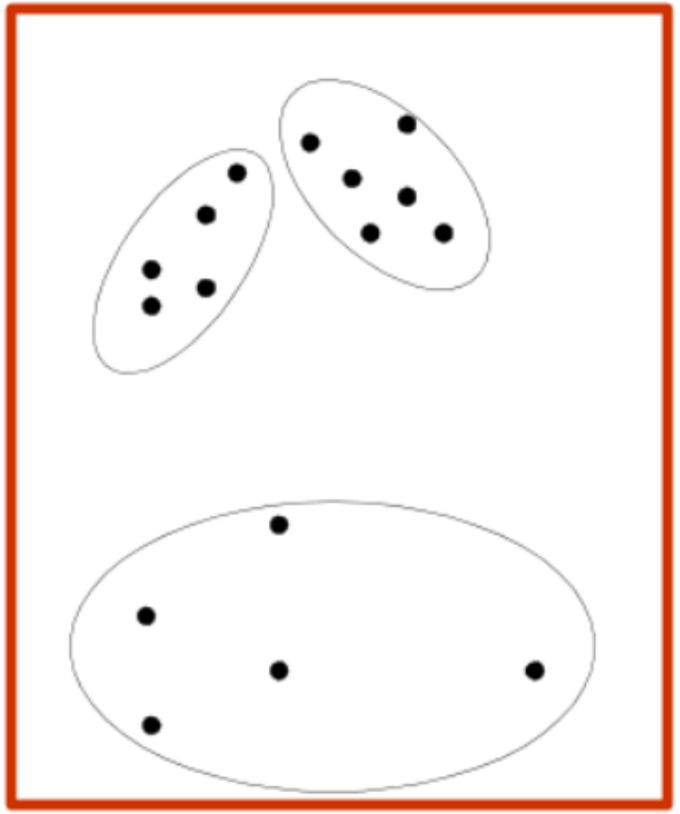
- Partitional vs. Hierarchical
  - Partitional Clustering: A division of data into non-overlapping clusters, such that each data object is in exactly one subset
  - Hierarchical Clustering: A set of nested clusters organized as a hierarchical tree
    - Each node (cluster) is union of its children (subclusters)
    - Root of tree: cluster containing all data objects
    - Leaves of tree: singleton clusters



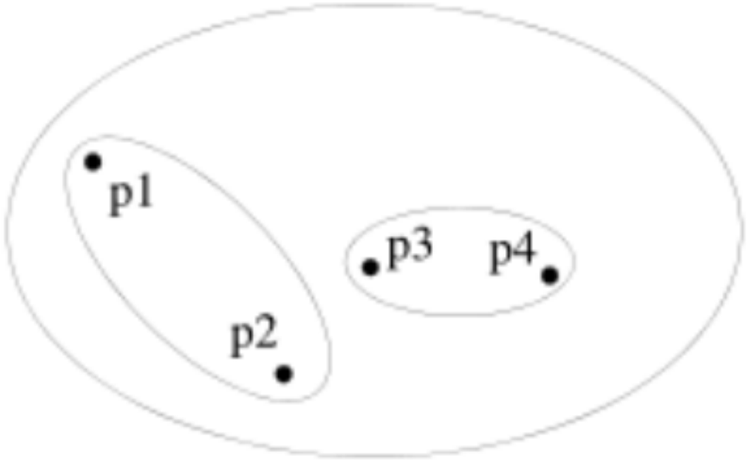
Partitional Clustering



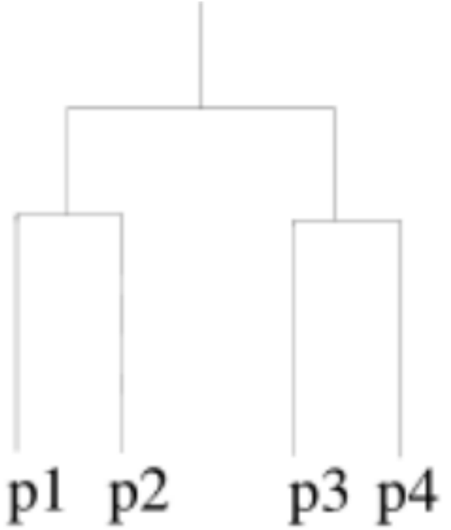
Original Points



A Partitional Clustering



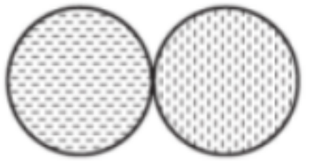
Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Types of Clusters: Prototype-Based

- **Prototype-based/ Center based cluster**
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the prototype or "center" of a cluster, than to the center of any other cluster
  - **Tend to be globular.**
    - an object in a cluster is closer to the *center* of a cluster than to the *center* of any other cluster
    - Center of a cluster ("the most central point"):
      1. Centroid: the mean of all the points in the cluster *(usually for continuous attributes)*
      2. Medoid: the most "representative" point of a cluster *(usually for categorical attributes)*



(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

2 center-based clusters

Clustering Algorithms

1. K-means and its variants

2. Hierarchical clustering

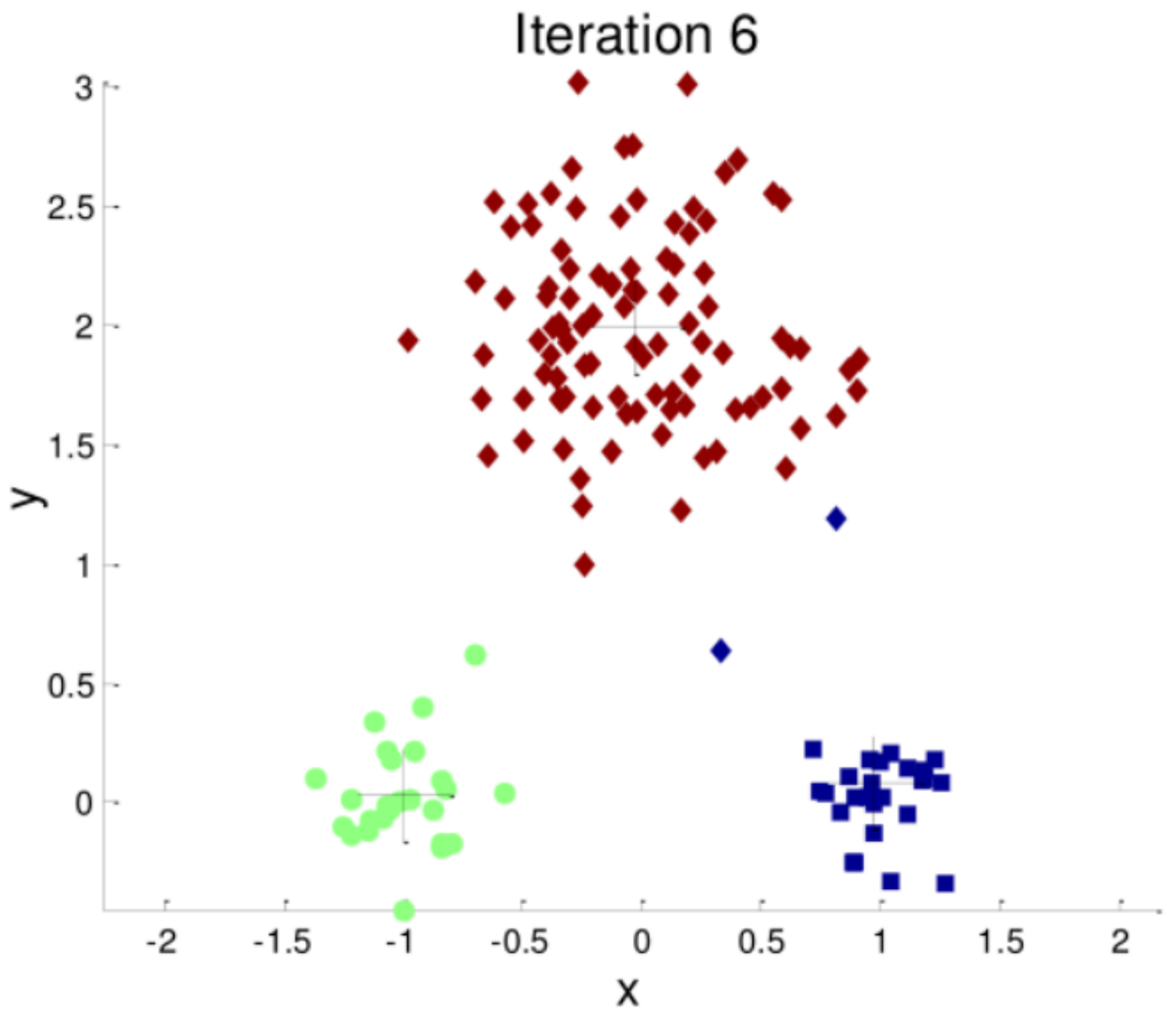
K Means Clustering

K-means Clustering

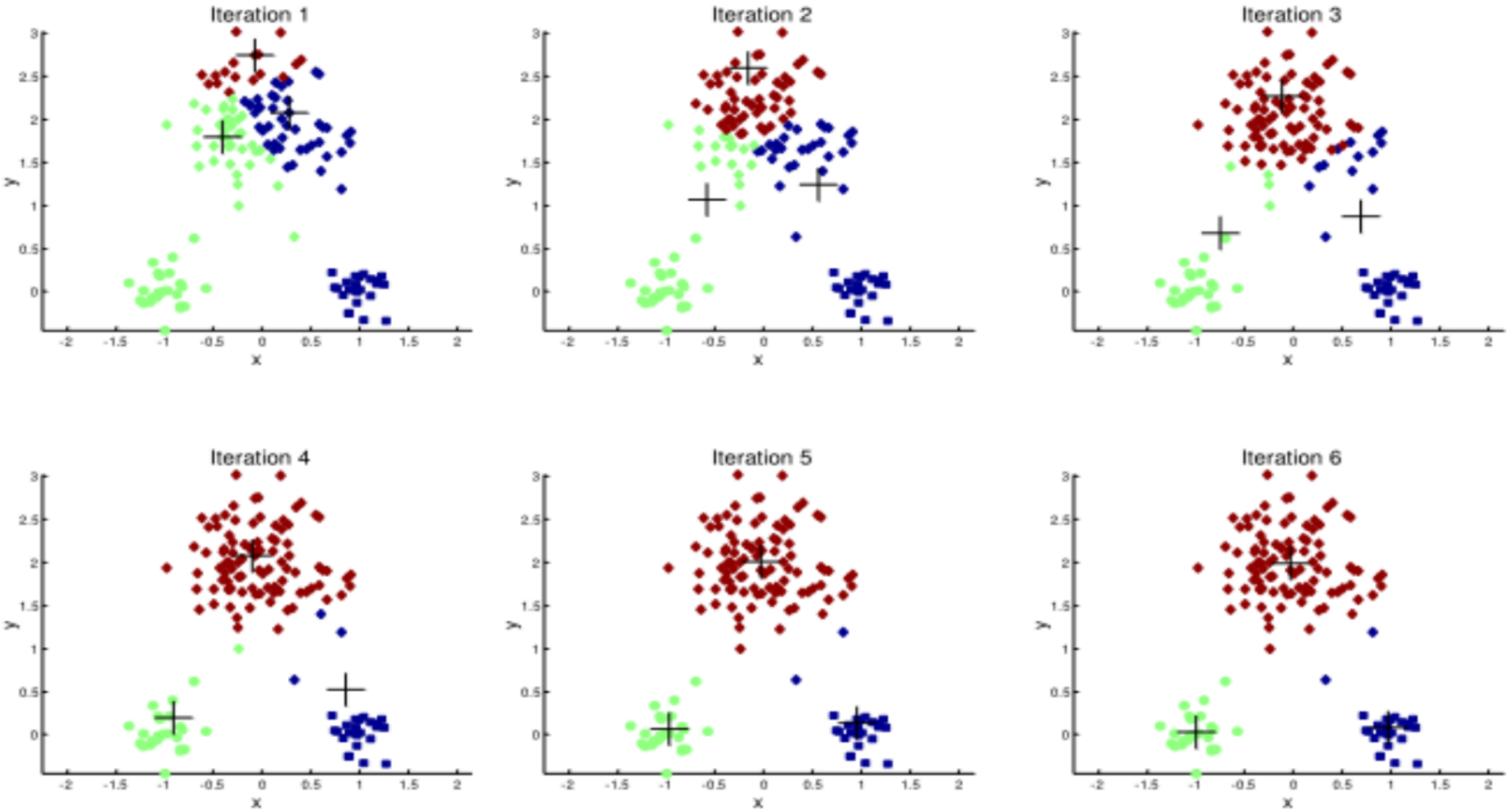
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

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

Example of K-means Clustering

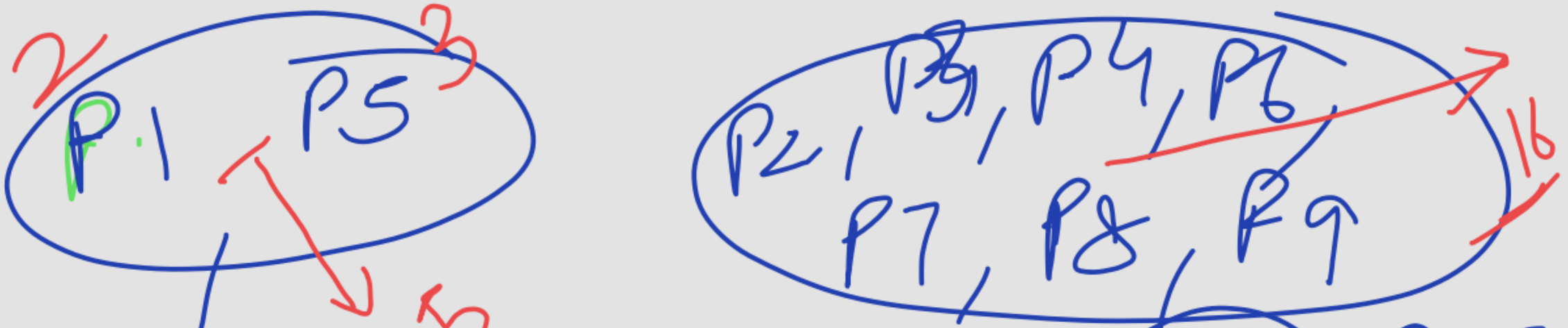


Example of K-means Clustering



K=2 [2, 4, 10, 12, 3, 20, 30, 11, 25]

		m1=2	m2=4	Centroid/ Clusters
P1	2	$\sqrt{(2-2)^2}=0$	$\sqrt{(2-4)^2}=2$	m1
P2	4	2	0	m2
P3	10	8	6	m2
P4	12	10	8	m2
P5	3	1	1	m1
P6	20	18	16	m2
P7	30	28	26	m2
P8	11	9	7	m2
P9	25	23	21	m2



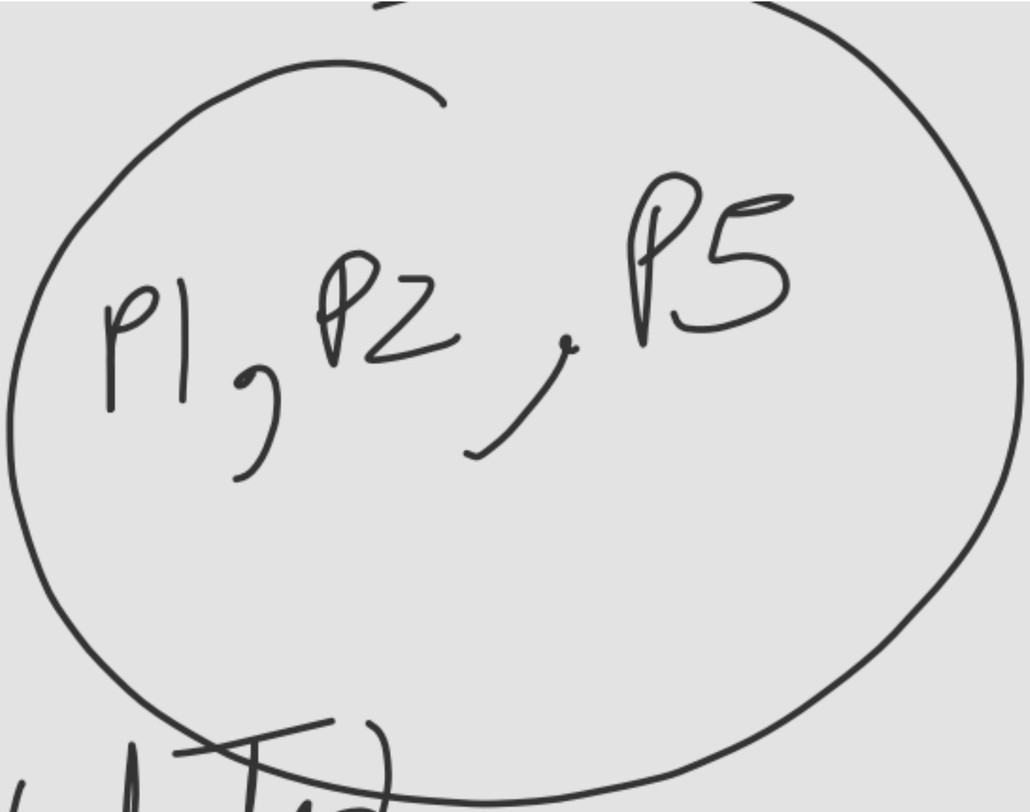
Updated Centroid $m_1 = \frac{2+3}{2} = 2.5$

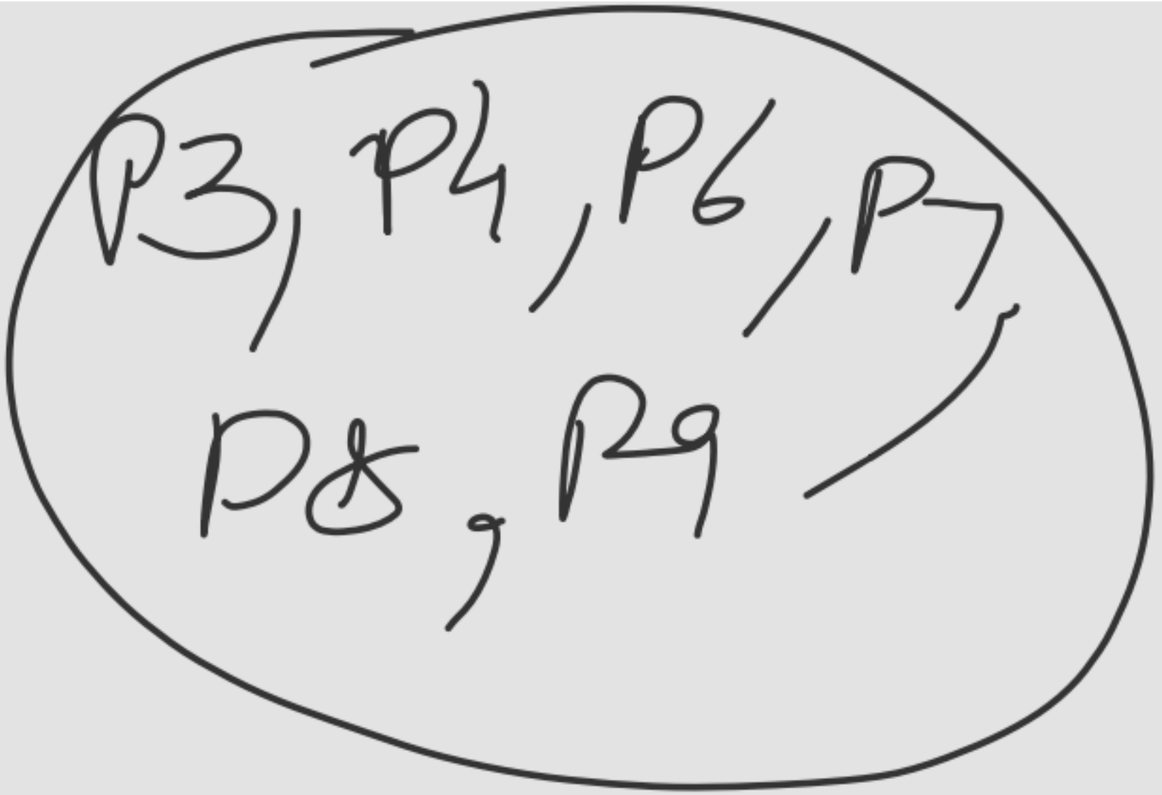
Updated Centroid $m_2 = \frac{4+10+12+20+30+11+25}{7}$

$(.5)^2 + (.5)^2 = 0.5$

$= 16$ //

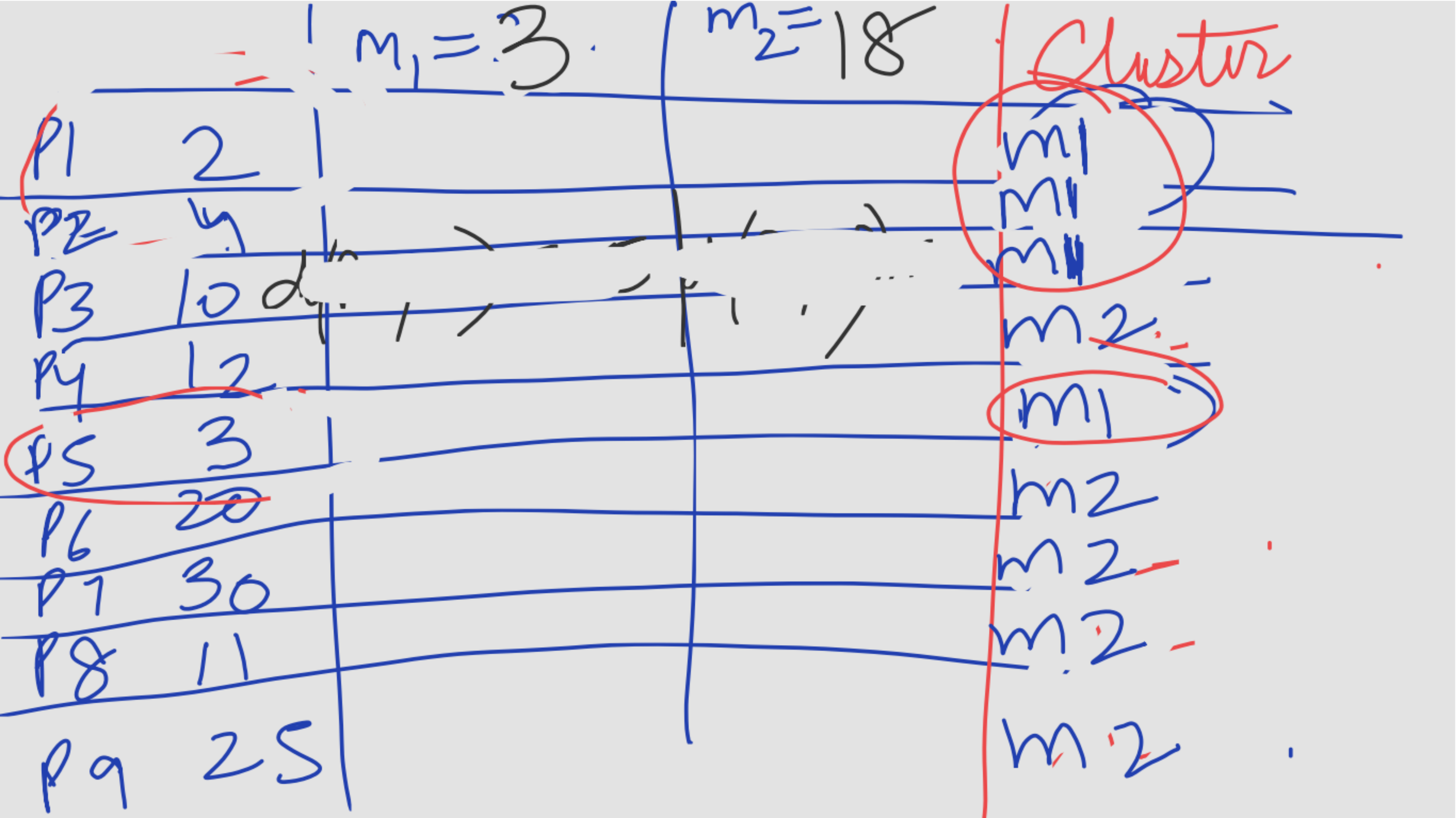
		$m_1 = 2.5$	$m_2 = 16$	Cluster
P1	2			m_1
P2	4			m_1
P3	10	$d(P3, m1)$ 7.5	$d(P3, m2)$ 6	m_2
P4	12			m_2
P5	3			m_1
P6	20			m_2
P7	30			m_2
P8	11			m_2
P9	25			m_2





Updated centroid $m_1 = 3$

$m_2 = 18$


m1=3
m2=18
Clustr
P1 2
P2 4
P3 10
P4 12
P5 3
P6 20
P7 30
P8 11
P9 25
m1
m1
m1
m2
m1
m2
m2
m2
m2

K-means Clustering – Details

- Simple iterative algorithm.
  - Choose initial centroids; repeat {assign each point to a nearest centroid; re-compute cluster centroids} until centroids stop changing.
- Initial centroids are often chosen randomly.
  - Clusters produced can vary from one run to another
- The centroid is (typically) the mean of the points in the cluster, but other definitions are possible (see Table 7.2).
- K-means will converge for common proximity measures with appropriately defined centroid (see Table 7.2)
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Repeat until 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

K-means Objective Function

- A common objective function (used with Euclidean distance measure) is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster center
  - To get SSE, we square these errors and sum them.

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

  - x is a data point in cluster C_i and c_i is the centroid (mean) for cluster C_i
  - SSE improves in each iteration of K-means until it reaches a local or global minima.

in Table 8.1, the centroid (mean) of the i^{th} cluster is defined by Equation 8.2.

$$\mathbf{c}_i = \frac{1}{m_i} \sum_{\mathbf{x} \in C_i} \mathbf{x} \qquad (8.2)$$

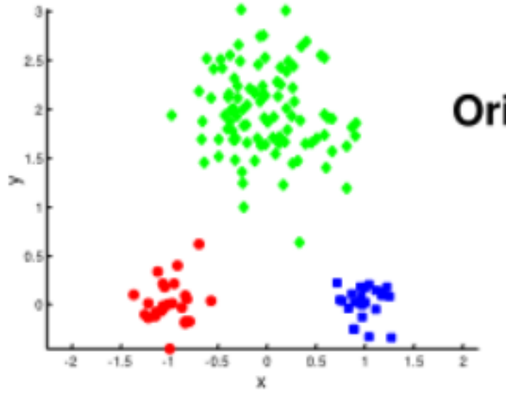
Table 8.2. K-means: Common choices for proximity, centroids, and objective functions.

Proximity Function	Centroid	Objective Function
Manhattan (L_1)	median	Minimize sum of the L_1 distance of an object to its cluster centroid
Squared Euclidean (L_2^2)	mean	Minimize sum of the squared L_2 distance of an object to its cluster centroid

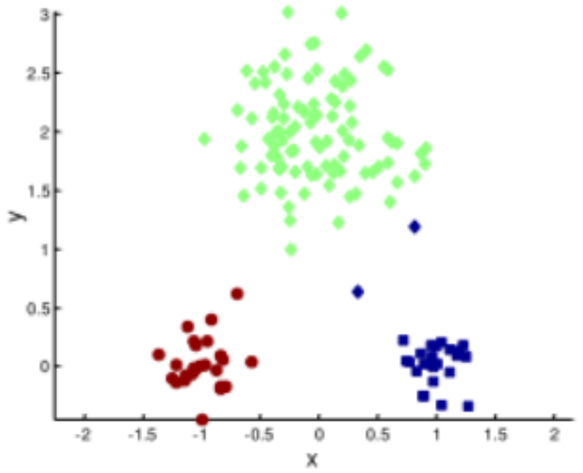
Given two different clusterings, produced by two different runs of k-means, prefer the clustering with the smaller SSE.

- Clustering is a better representation

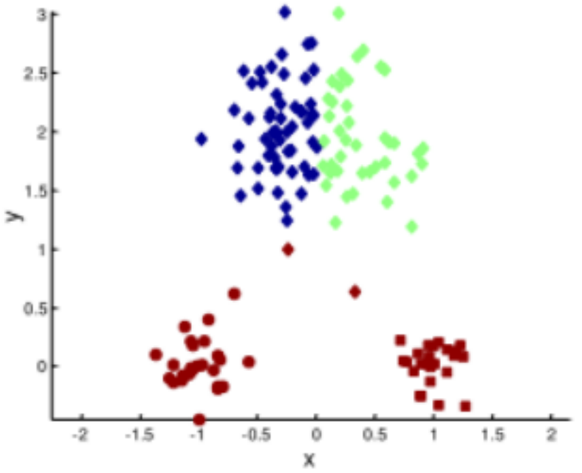
Two different K-means Clusterings



Original Points

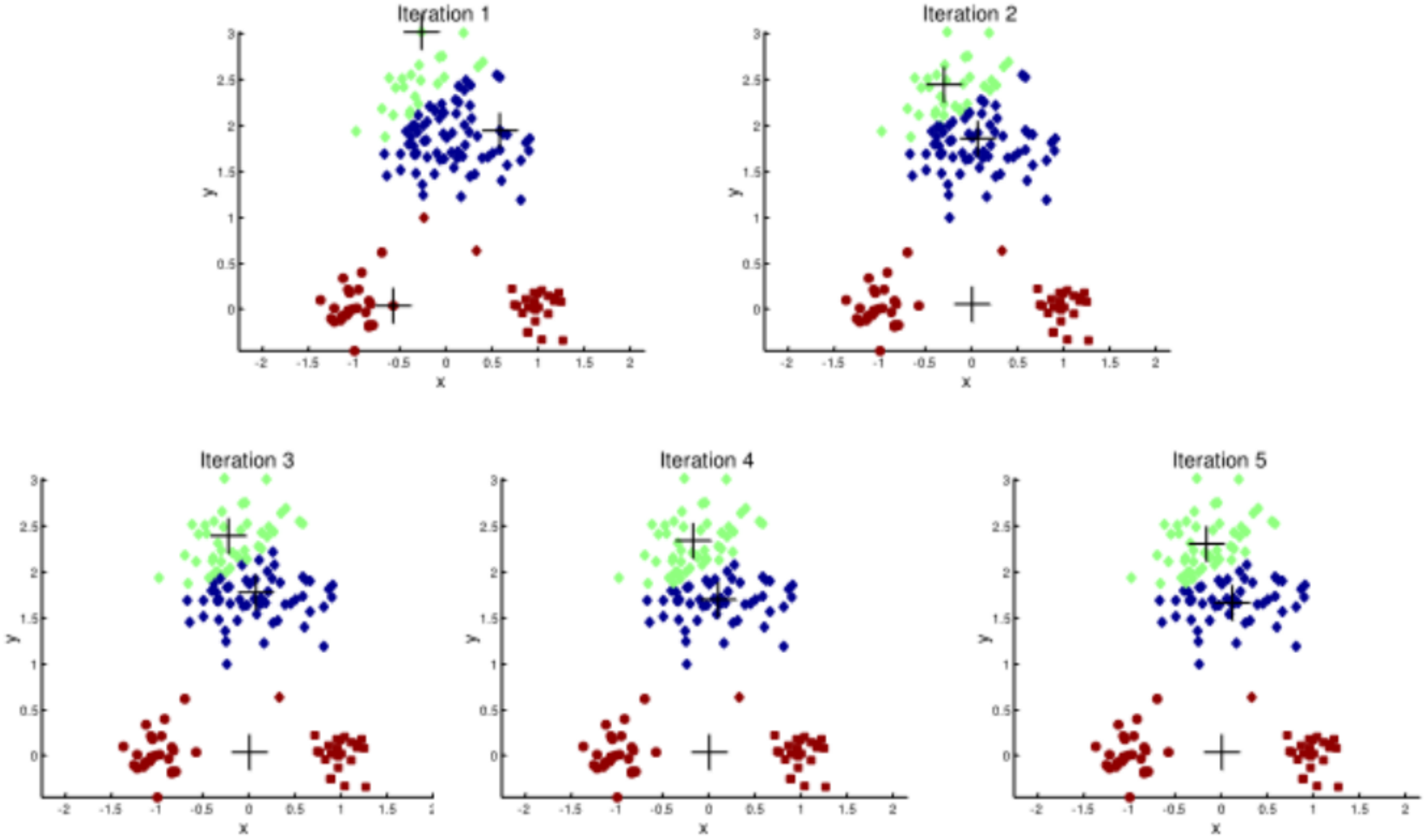


Optimal Clustering



Sub-optimal Clustering

Importance of Choosing Initial Centroids ...



Initial Centroids Problem

- **The algorithm does not guarantee convergence to the global optimum.**
- **The result may depend on the initial centroids.**

- **Solutions**
  - Multiple runs – common choice, but probability is not on your side
  - Sample and use hierarchical clustering to determine initial centroids

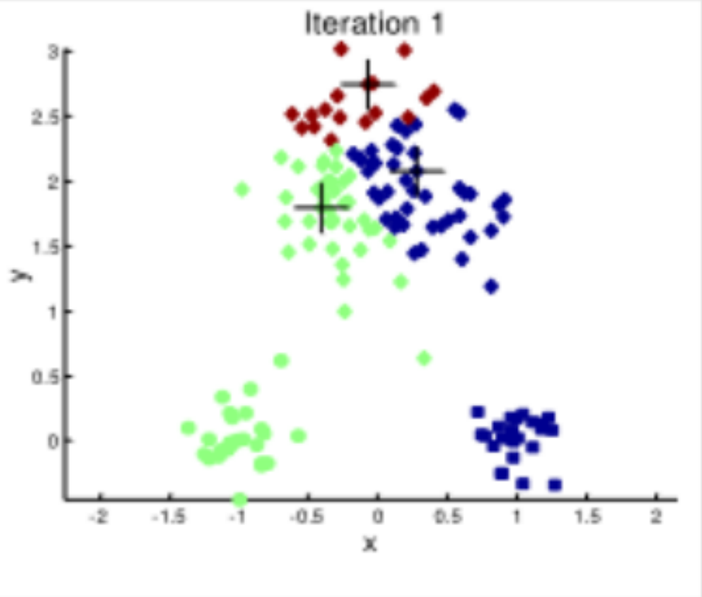
Solution: Choose centroids which are distant apart
1. Select initial centroid (could be mean of all data points)
2. Remaining centroids can be chosen using distance from the initial centroid.

Pre-processing and Post-processing

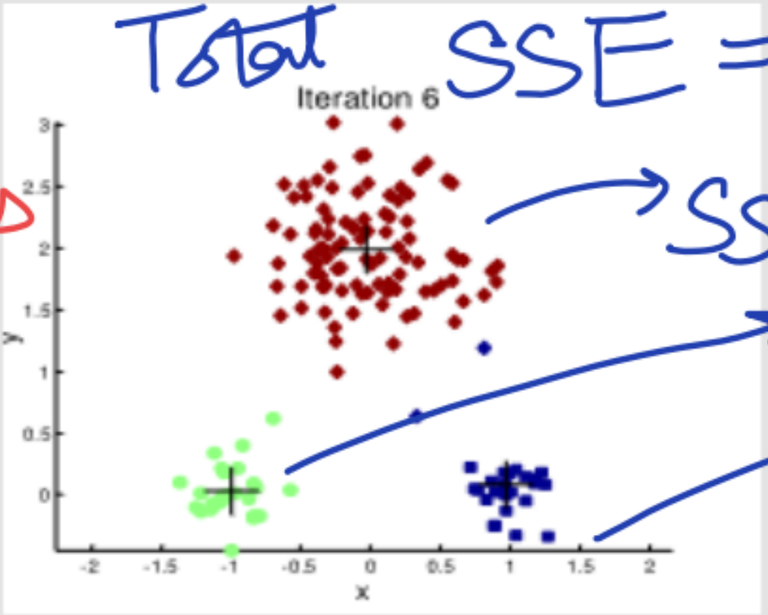
- Pre-processing
  - Normalize the data
  - Eliminate outliers

Reduce the SSE Using Post-processing

- **Split a cluster**: Split 'loose' clusters, i.e., clusters with relatively high SSE
- **Disperse a cluster**: Eliminate small clusters that may represent outliers
- **Merge two clusters**: Merge clusters that are 'close' and that have relatively low SSE

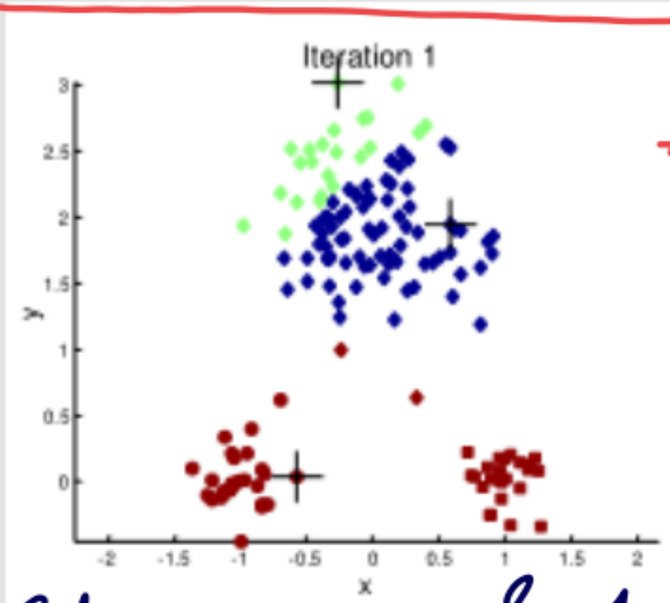


After 5 iterations →



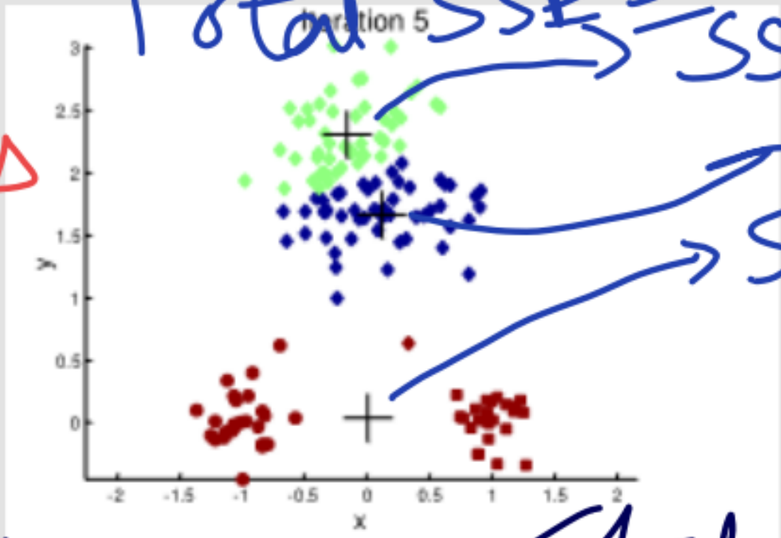
Total SSE = SSE1 + SSE2 + SSE3

Sd1 = 1



After 5 iterations →

K = 3



Total SSE = SSE1 + SSE2 + SSE2 ...

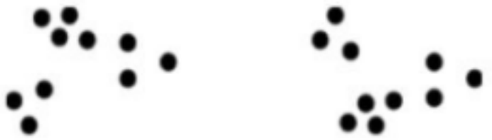
Sd2

Choose Solution with min Total SSE

Additional Slide

How to decide no. of clusters K?

Notion of a Cluster can be Ambiguous



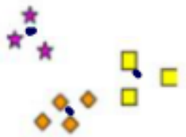
How many clusters?

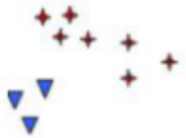
By the human visual system, it looks like two clusters.

Two Clusters

But it really depends on the characteristics of the data.
These clusterings may not be unreasonable:

Six Clusters

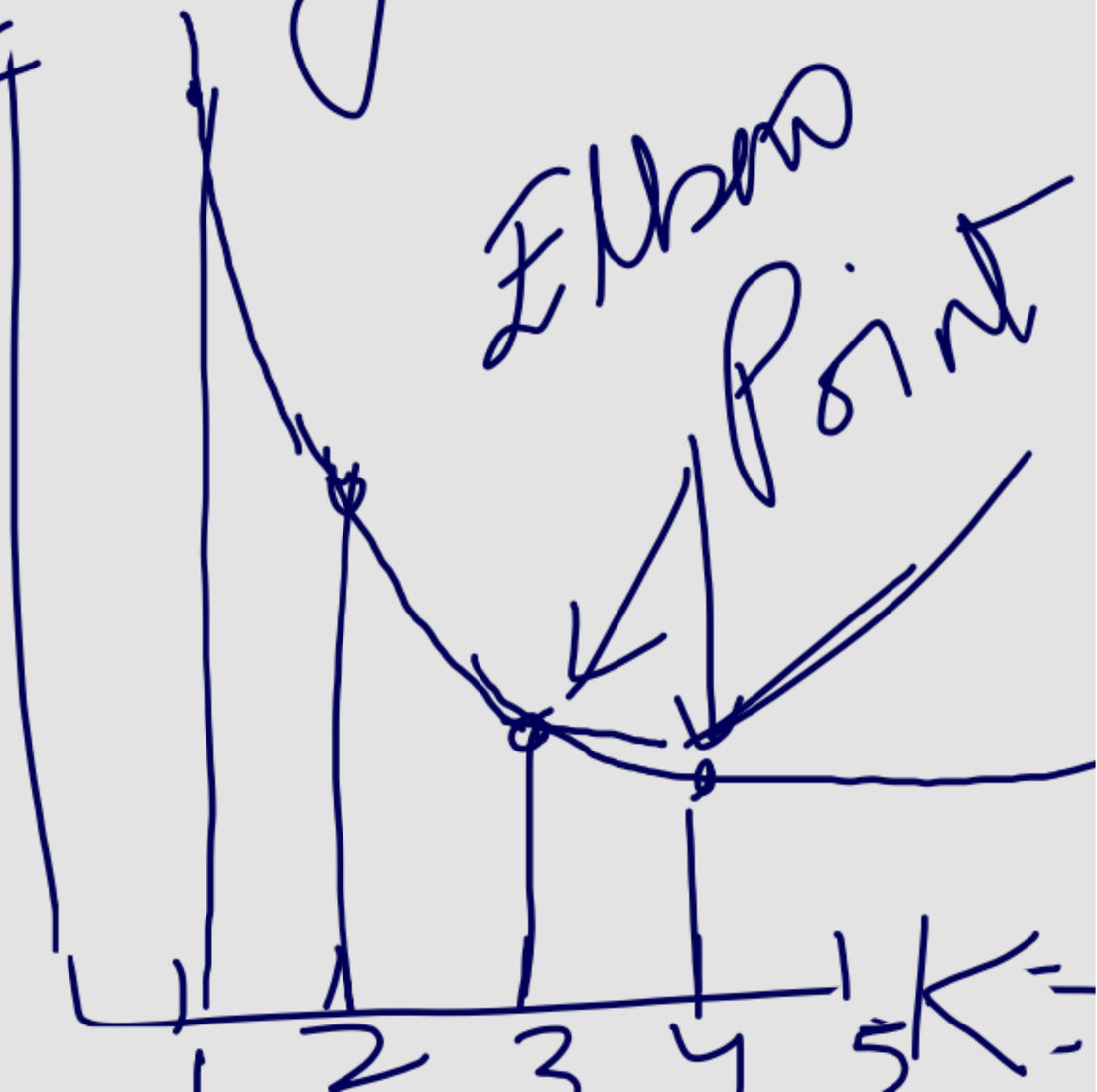




Four Clusters

SSE

Elbow Point



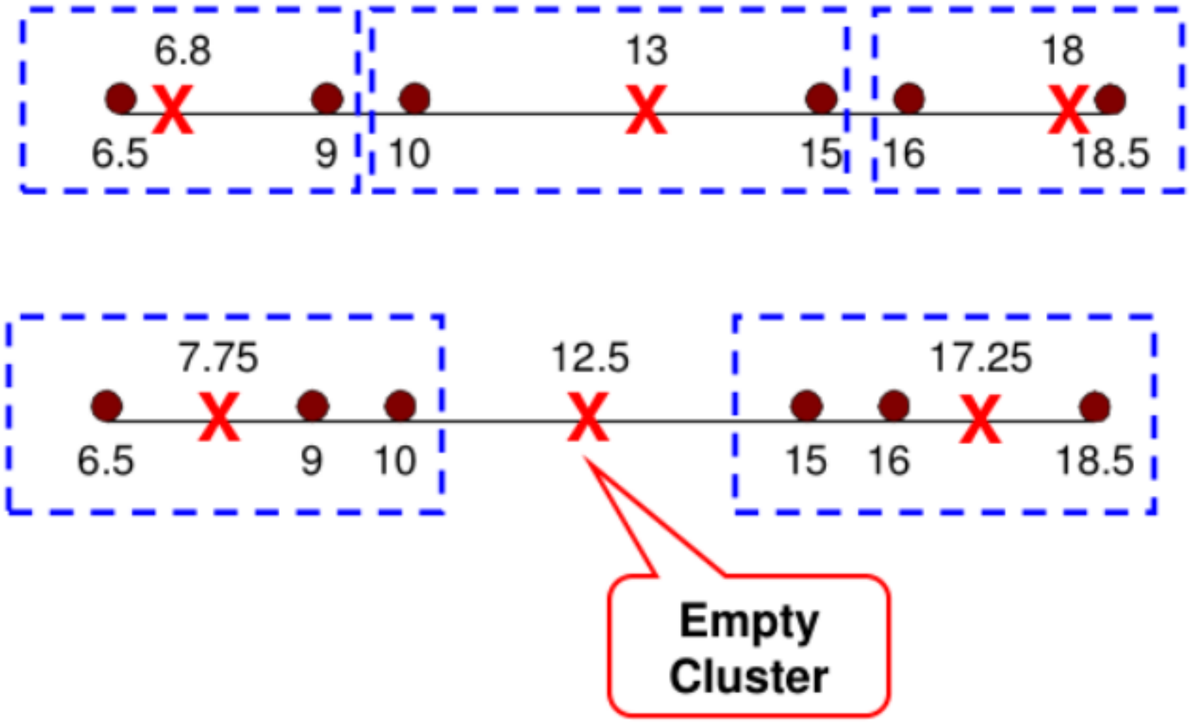
Discussion

- **Advantages:**
  - K-Means is relatively easy to understand and implement
  - It runs relatively quickly, and scales easily to large datasets

- **Disadvantages:**
  - It relies on a random initialization, which means the outcome of the algorithm depends on a random seed.
  - It requires to specify the number of clusters you are looking for (which might not be known in a real-world application).
  - It can yield empty clusters
  - The performance depends highly on scaling of the data.
  - It has problems when
    - clusters are of differing sizes, densities, or non-globular shapes
    - the data contains outliers.

Issues with K-Means: Empty Clusters

- K-means can yield empty clusters



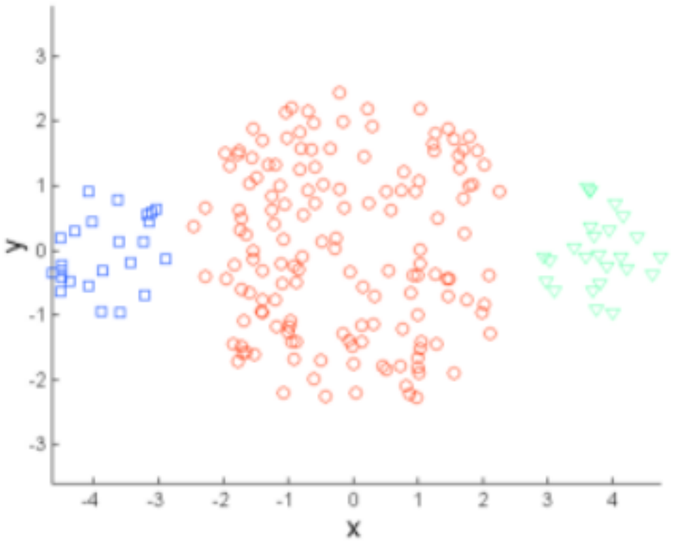
Break till 11:53

Limitations of K-means

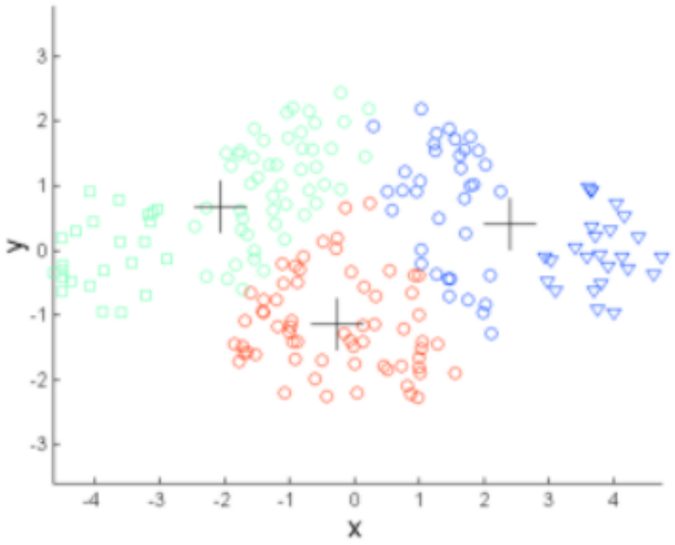
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

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

Limitations of K-means: Differing Sizes

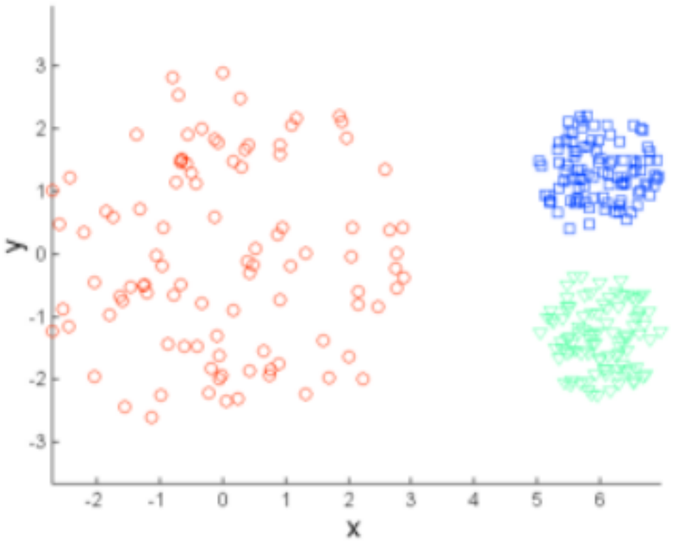


Original Points

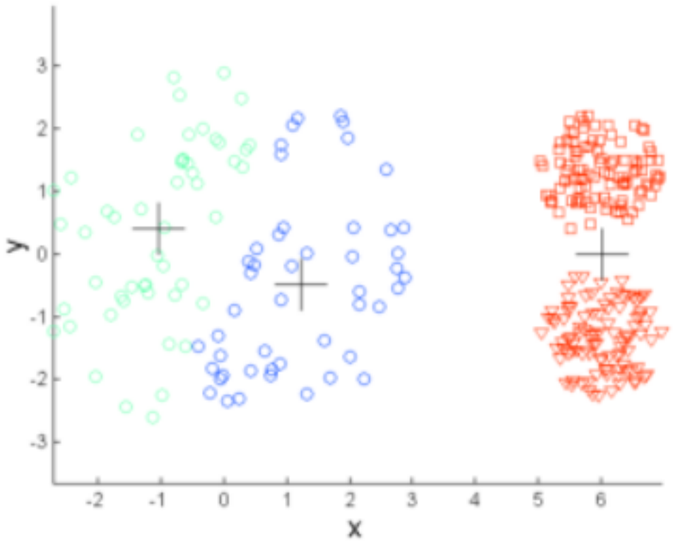


K-means (3 Clusters)

Limitations of K-means: Differing Density

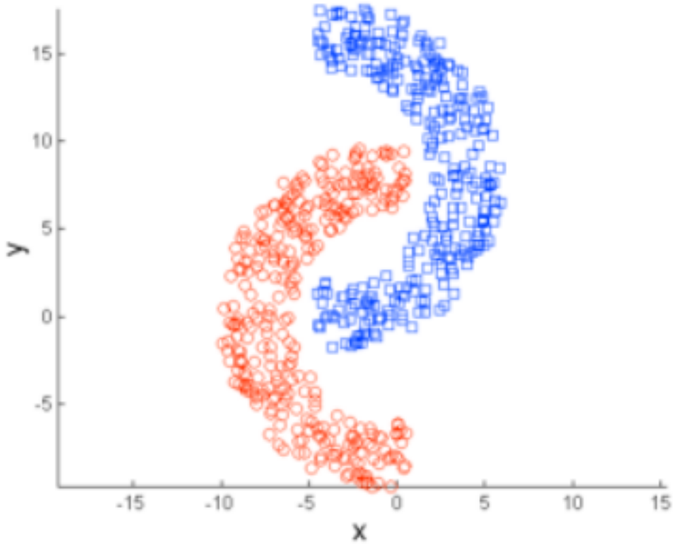


Original Points

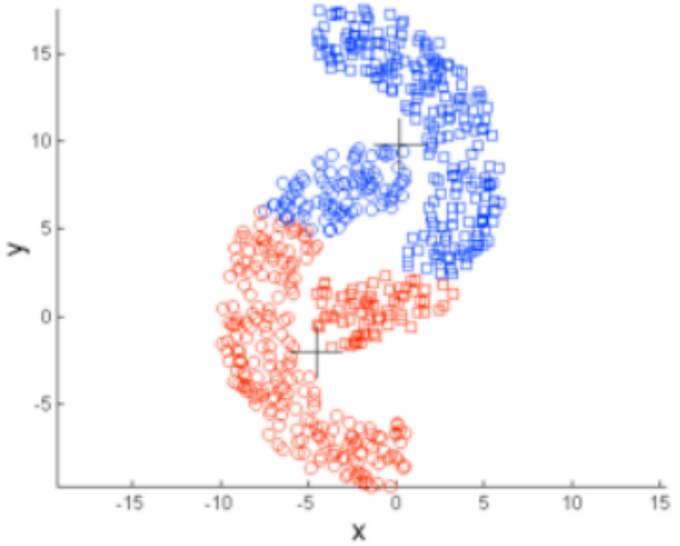


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

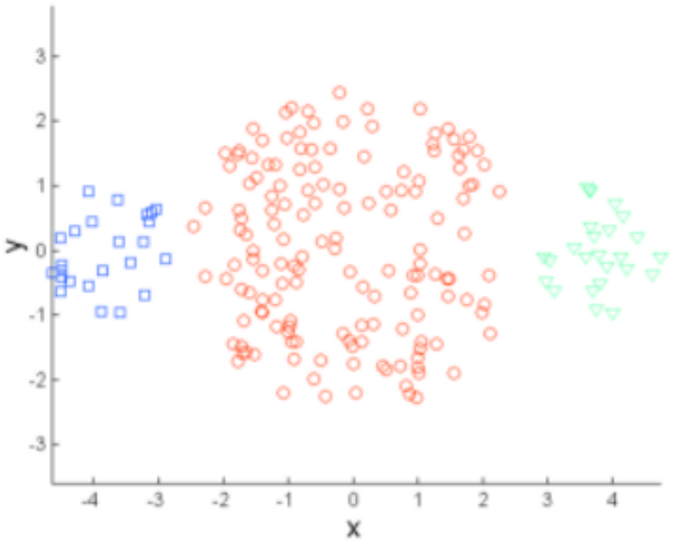


Original Points

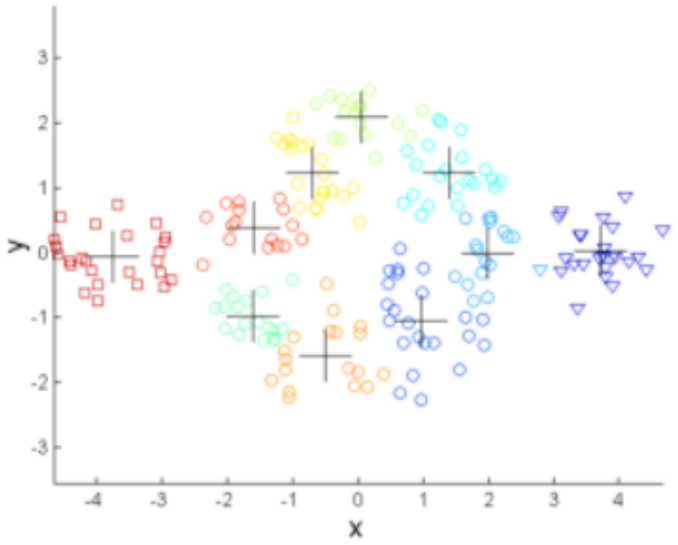


K-means (2 Clusters)

Overcoming K-means Limitations



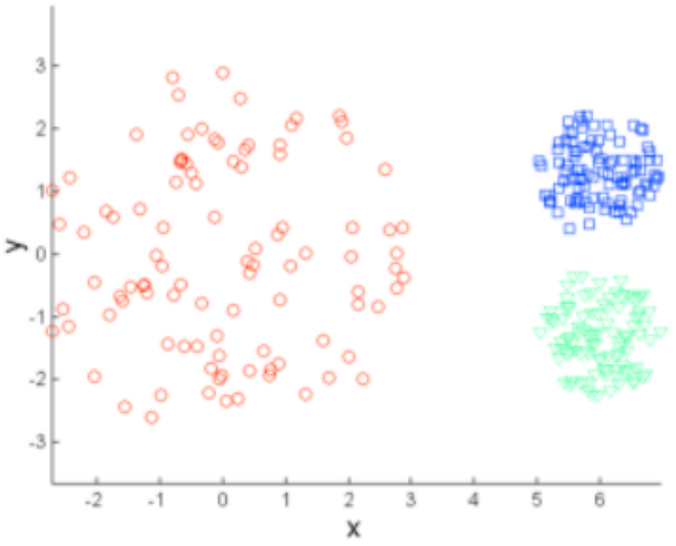
Original Points



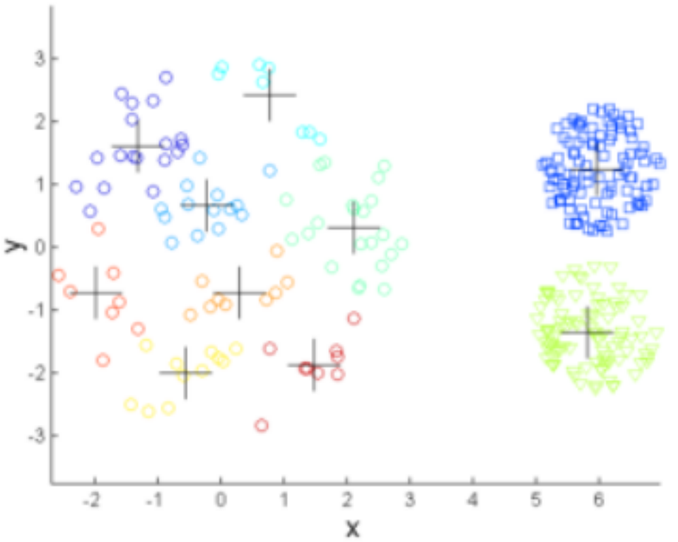
K-means Clusters

One solution is to use many clusters.
Find parts of clusters, but need to put together.

Overcoming K-means Limitations

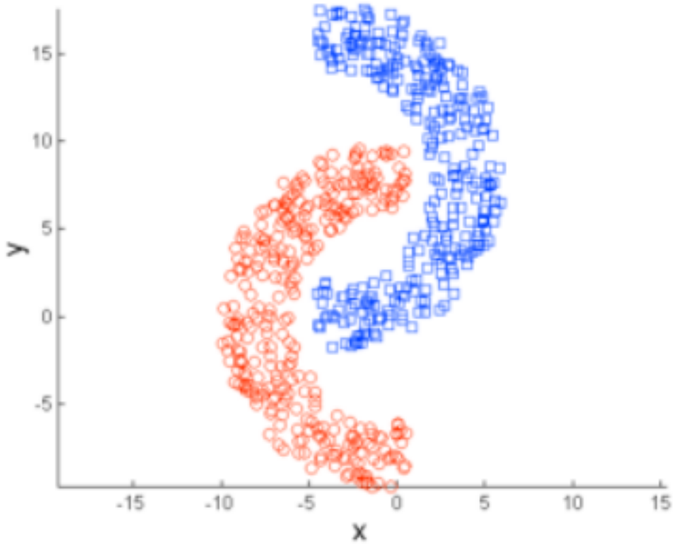


Original Points

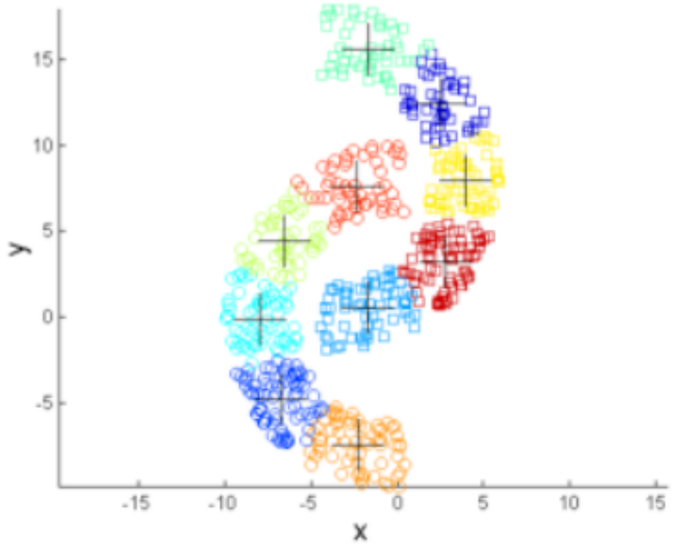


K-means Clusters

Overcoming K-means Limitations



Original Points

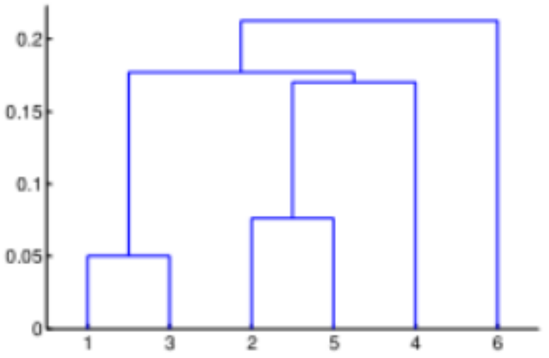


K-means Clusters

Hierarchical Clustering

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that represent cluster-subcluster relationships and records the sequences (order) of merges or splits



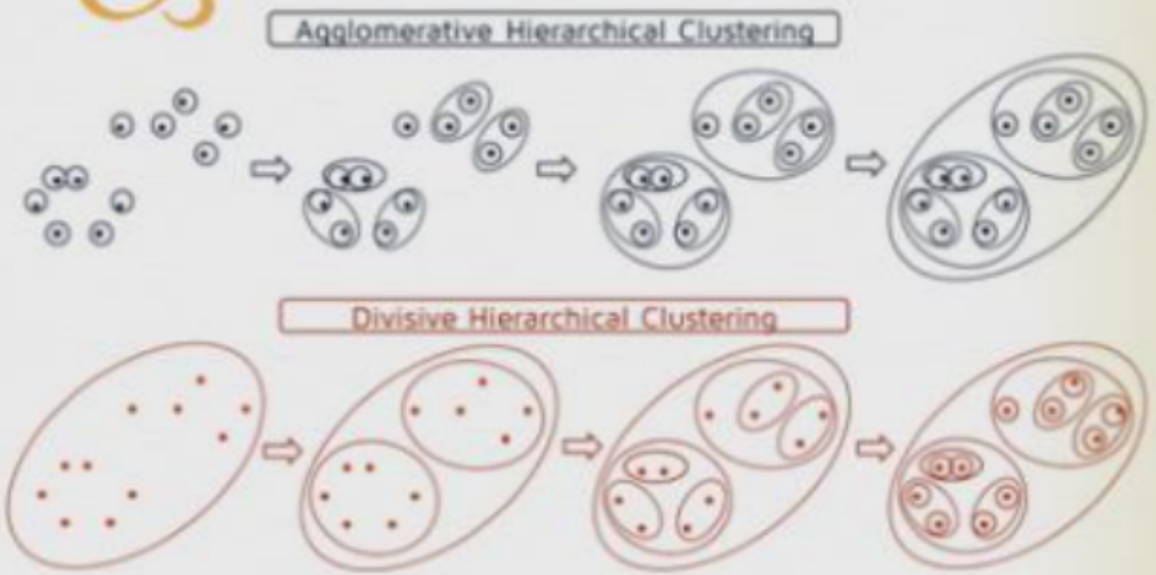
Strengths of Hierarchical Clustering

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

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

Hierarchical Clustering

- Clusters are created in levels actually creating sets of clusters at each level.
- *Agglomerative*
  - Initially each item in its own cluster
  - Iteratively clusters are merged together
  - Bottom Up
- *Divisive*
  - Initially all items in one cluster
  - Large clusters are successively divided
  - Top Down



Hierarchical Clustering Approaches

1. Agglomerative: start with data points as *individual clusters (bottom-up)*
   - at each step merge the closest pair of clusters
   - *Definition of "cluster proximity" needed.*
2. Divisive: start with one all-inclusive cluster *(top-down)*
   - at each step split a cluster until only singleton clusters remain
   - Need to decide which cluster to split and how to do splitting

- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

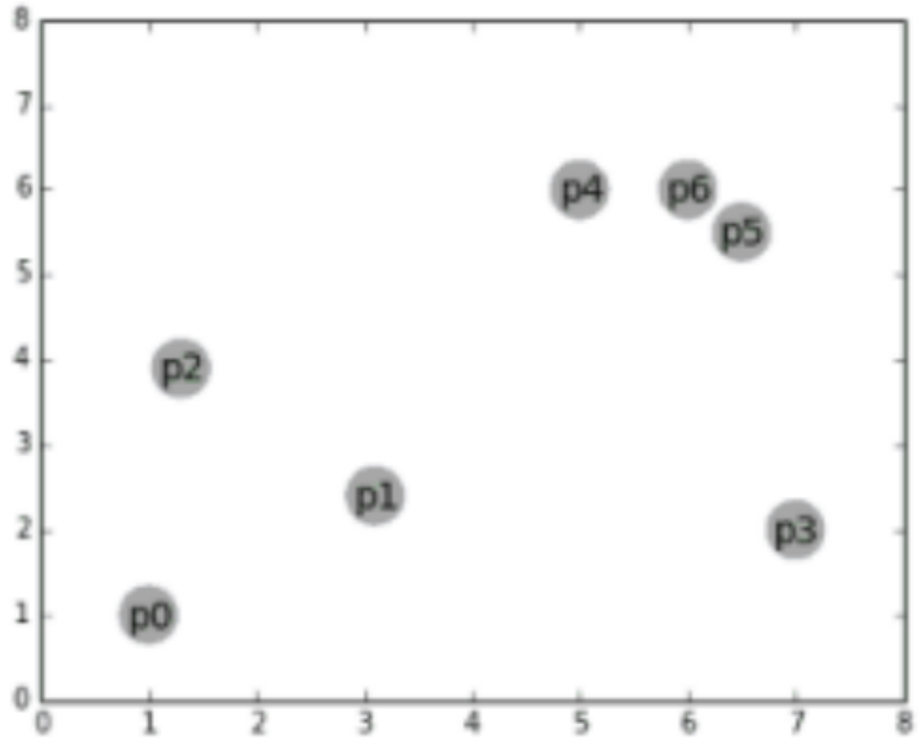
- Most popular hierarchical clustering technique
  - **Key Idea: Successively merge closest clusters**

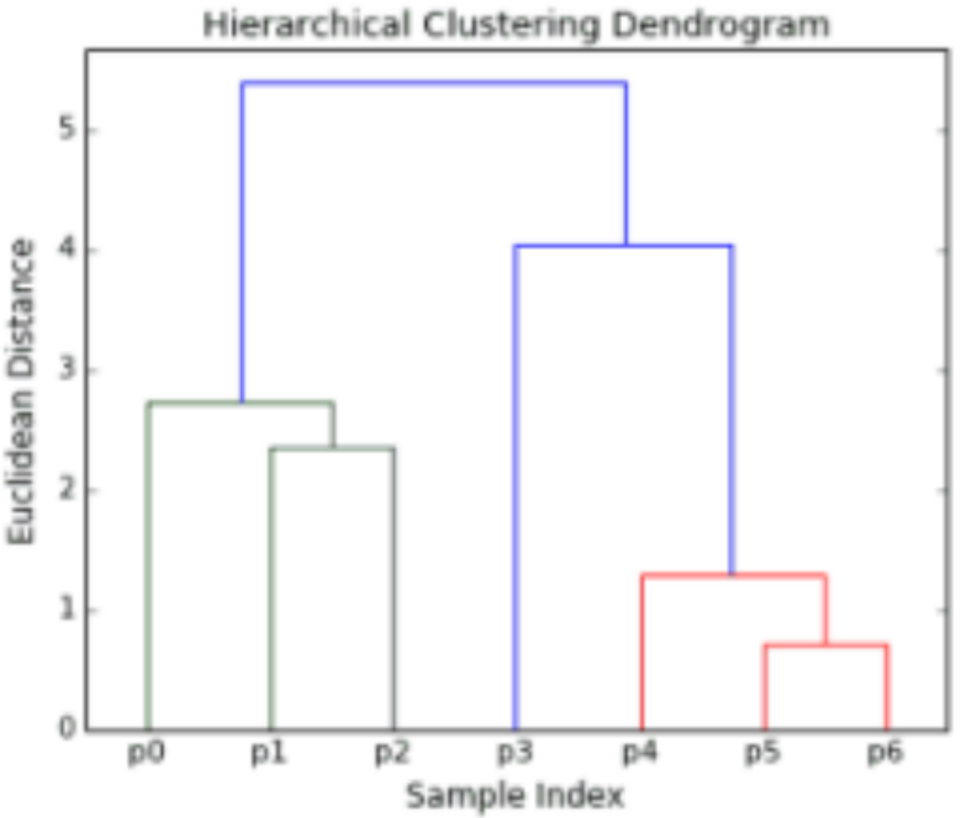
1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

Originally, the distance between two points

Update with distance between two clusters.
- How to define?

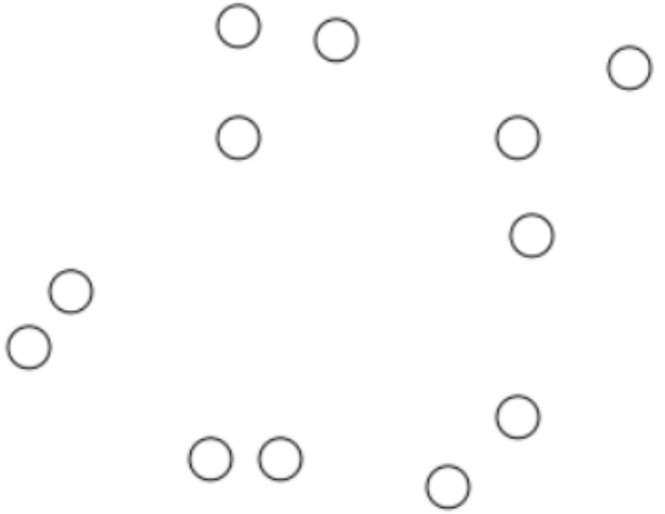
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms


p0
p1
p2
p3
p4
p5
p6


Hierarchical Clustering Dendrogram
Euclidean Distance
Sample Index
p0
p1
p2
p3
p4
p5
p6

Starting Situation

- Start with clusters of individual points and a proximity matrix



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

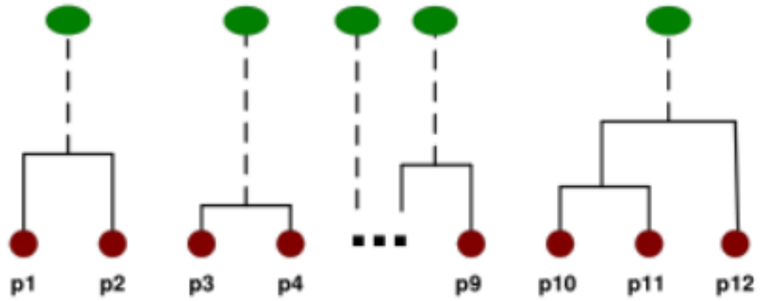
Proximity Matrix

Intermediate Situation

- After some merging steps, we have some clusters

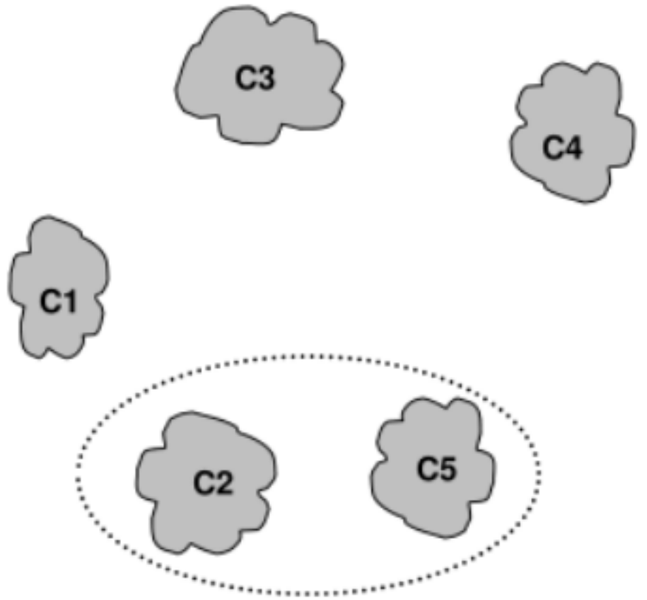
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



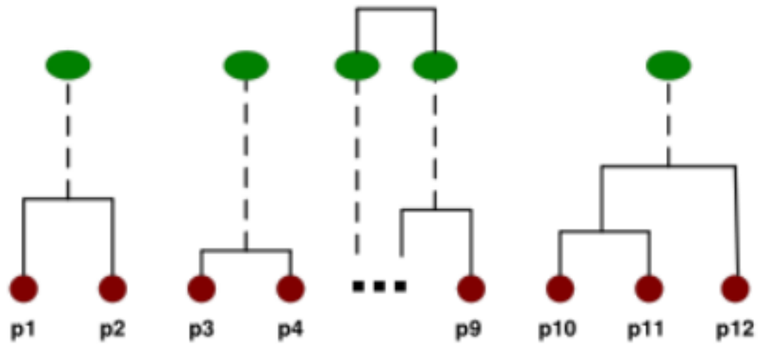
Intermediate Situation

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



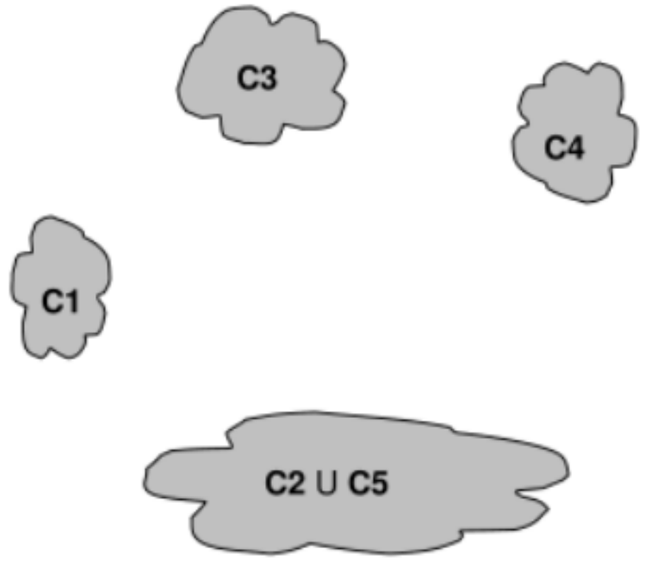
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



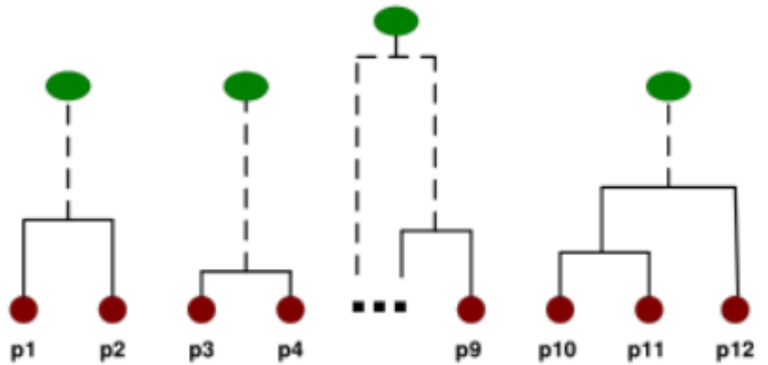
After Merging

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

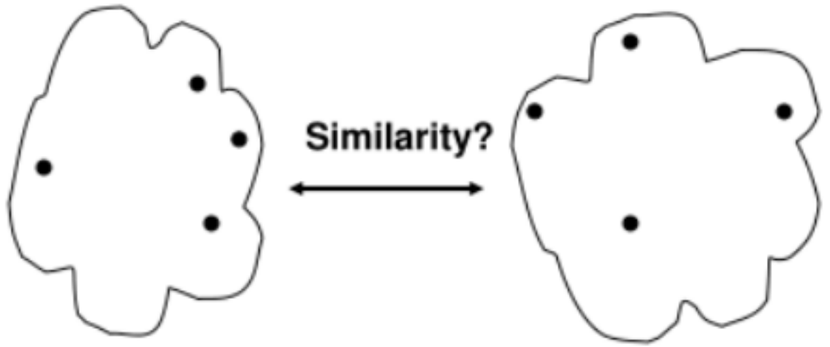


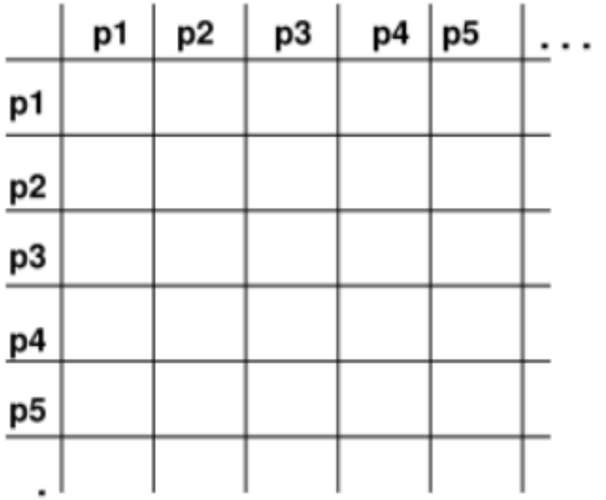
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Distance



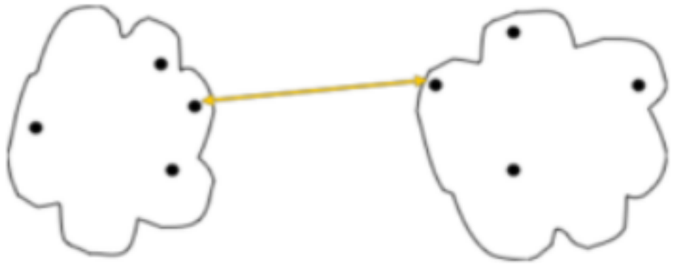


Proximity Matrix

- **MIN** (short/single-link)
- **MAX** (suggestive/complete-link)
- **Group Average**
- **Distance Between Centroids**
- Other methods driven by an objective function
  - Ward's Method uses squared error

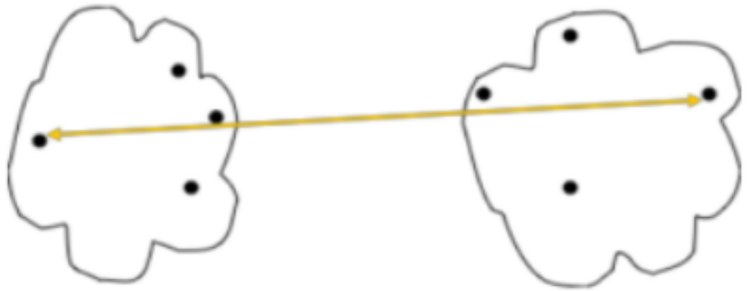
Defining Proximity between Clusters

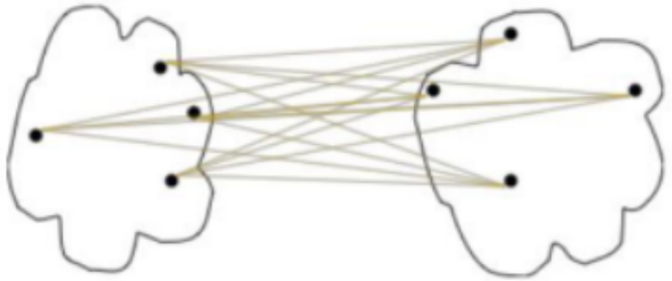
MIN (single-link)



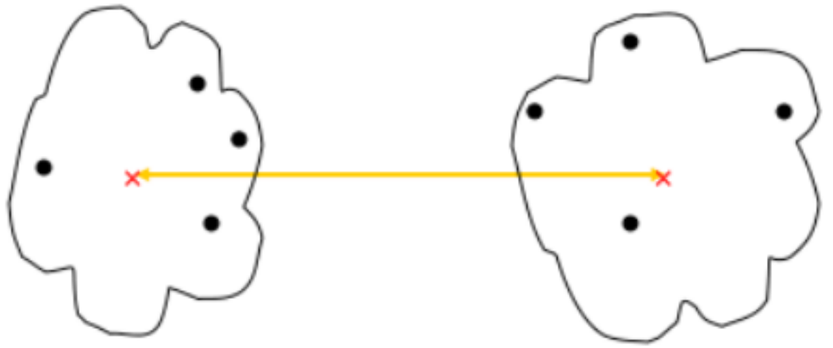
MAX (complete-link)

Group Average





How to Define Inter-Cluster Similarity



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

Proximity Matrix

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

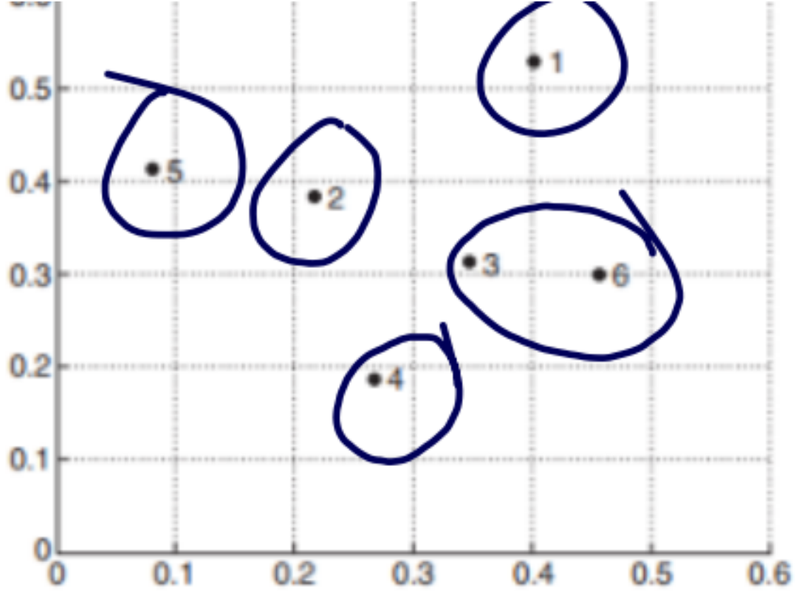


Figure 7.15. Set of six two-dimensional points.

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

Table 7.3. xy-coordinates of six points.

$$d(p1, p2) = \sqrt{|x_{p1} - x_{p1}|^2 + |y_{p1} - y_{p2}|^2}$$

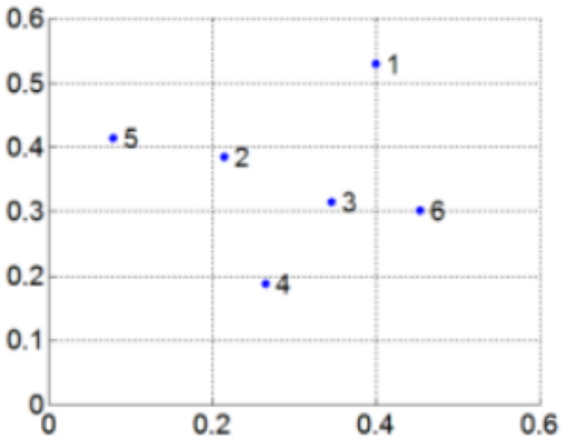
$$= \sqrt{|0.40 - 0.22|^2 + |0.53 - 0.38|^2}$$

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

Table 7.4. Euclidean distance matrix for six points.

MIN or Single Link

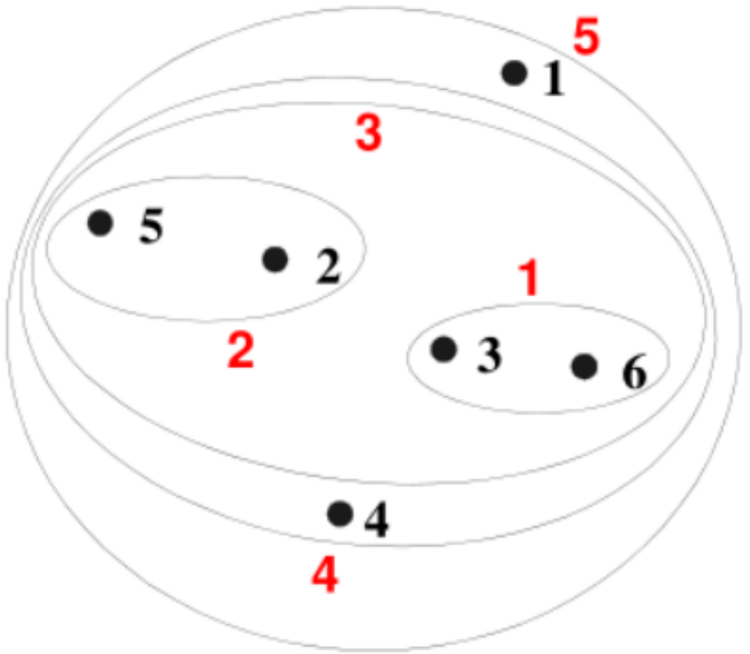
- Proximity of two clusters is based on the two closest points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph
- Example:



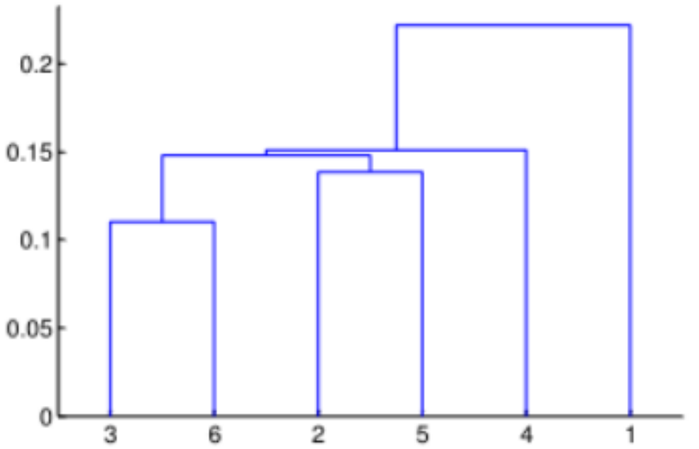
Distance Matrix:

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

Hierarchical Clustering: MIN



Nested Clusters



Dendrogram

Let's say we now have:

- Cluster {3,6}
- Cluster {2,5}
- Cluster {4}
- Cluster {1}

What's the next merge?

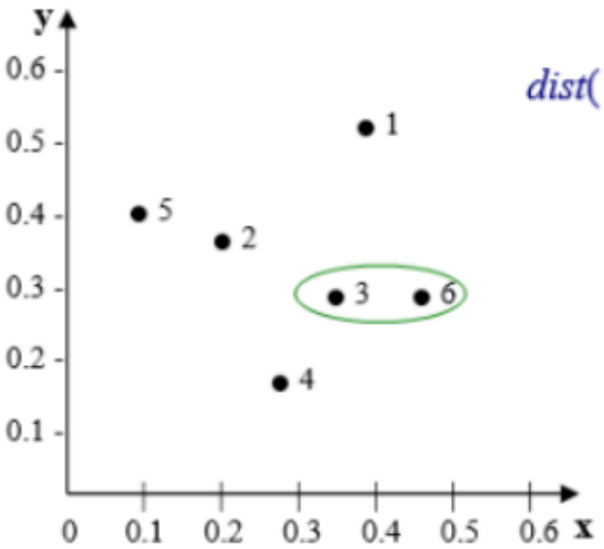
MIN({3,6},{2,5}) = 0.15
Is the next smallest value

Distance matrix

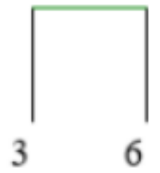
	p1	p2	p3	p4	p5	p6
p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0

space

dendogram



dist((p3, p6), p1) = MIN (*dist*(p3, p1) , *dist*(p6, p1))
= MIN (0.22 , 0.23)
= 0.22

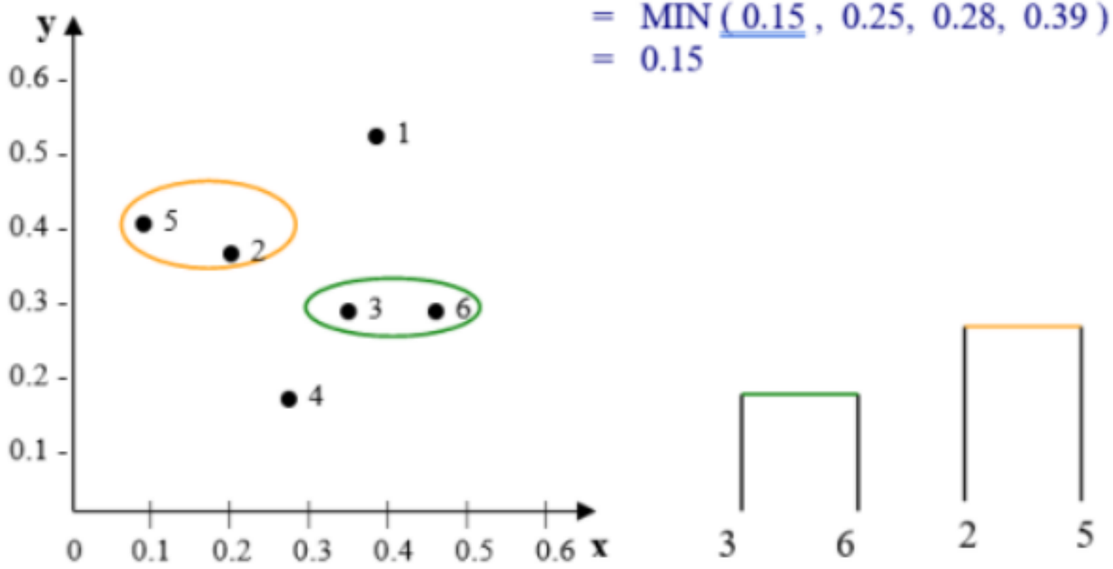


Distance matrix

p1	0				
p2	0.24	0			
(p3, p6)	**0.22**	**0.15**	0		
p4	0.37	0.20	**0.15**	0	
p5	0.34	0.14	**0.28**	0.29	0
	p1	p2	**(p3, p6)**	p4	p5

space

dist((p3, p6), (p2, p5)) = MIN (*dist*(p3, p2) , *dist*(p6, p2), *dist*(p3, p5), *dist*(p6, p5))
= MIN (0.15 , 0.25, 0.28, 0.39) //from original matrix
= 0.15

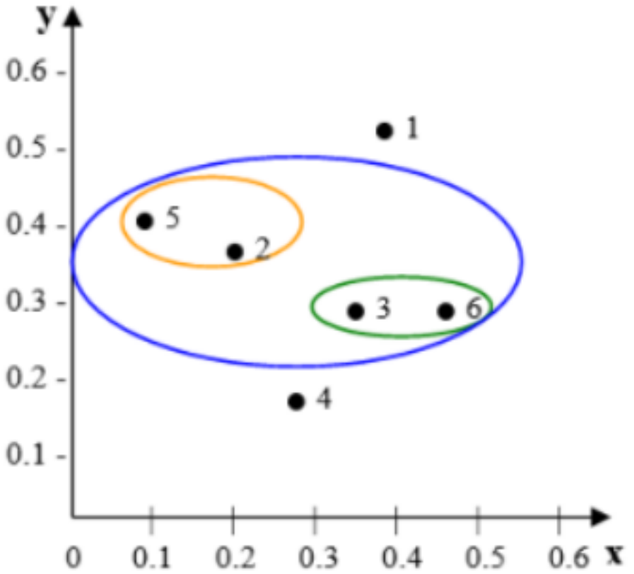


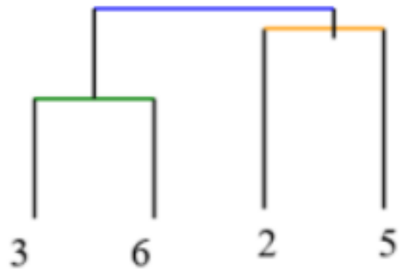
Distance matrix

p1	0			
(p2, p5)	**0.24**	0		
(p3, p6)	0.22	**0.15**	0	
p4	0.37	**0.20**	0.15	0
	p1	**(p2, p5)**	(p3, p6)	p4

space

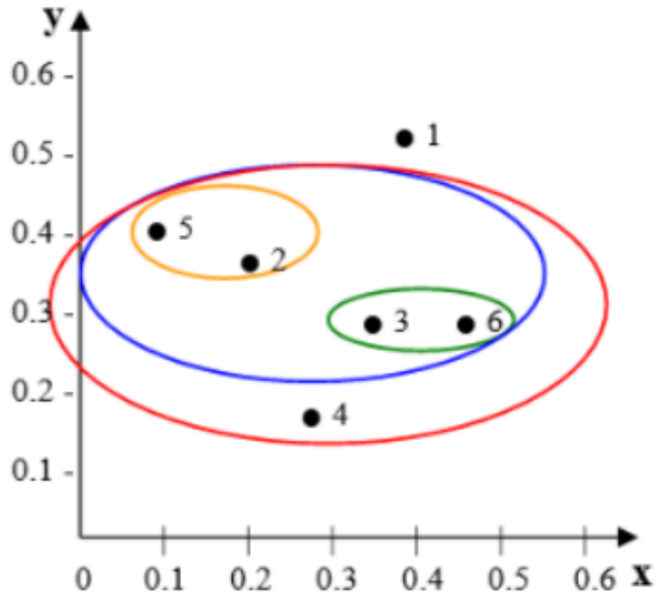
dendogram

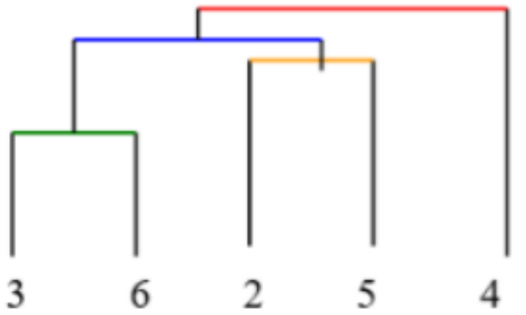




Distance matrix

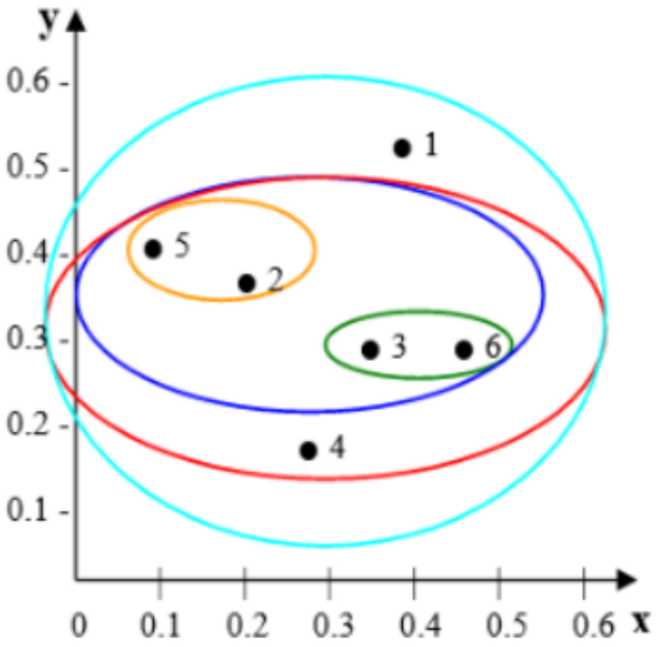
p1	0		
(p2, p5, p3, p6)	**0.22**	0	
p4	0.37	**0.15**	0
	p1	**(p2, p5, p3, p6)**	p4

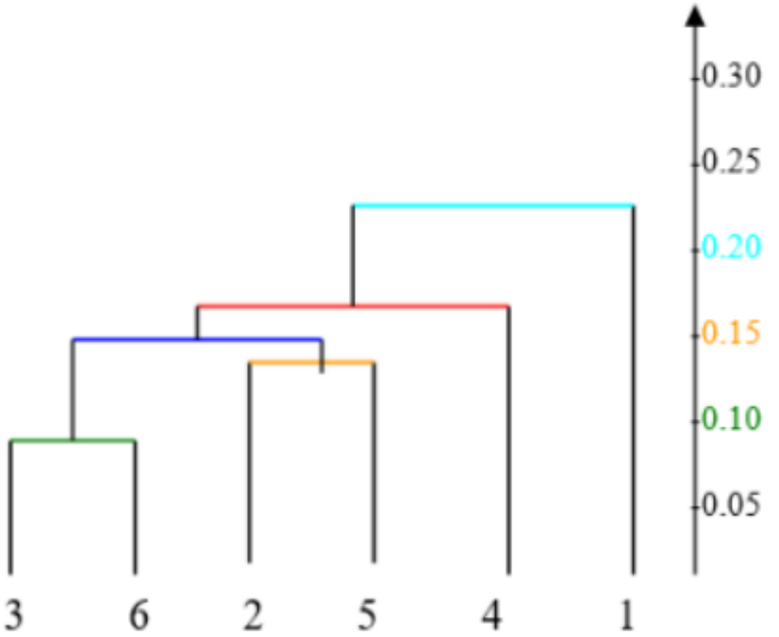




Distance matrix

p1	0	
(p2, p5, p3, p6, p4)	0.22	0
	p1	(p2, p5, p3, p6, p4)





Single Link Clustering - Example 7

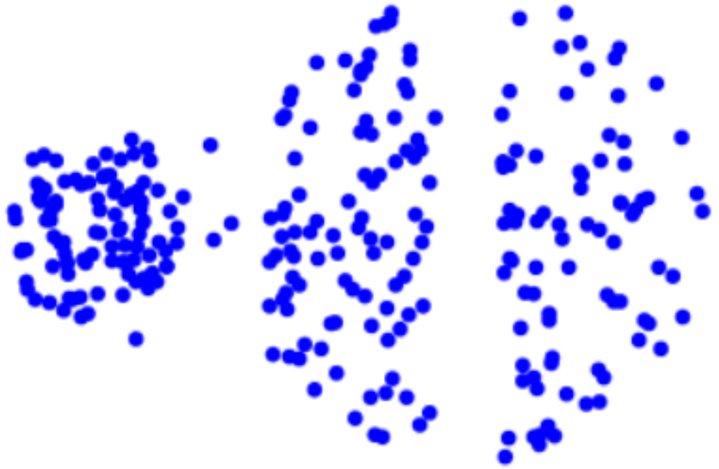
Strength of MIN

Original Points

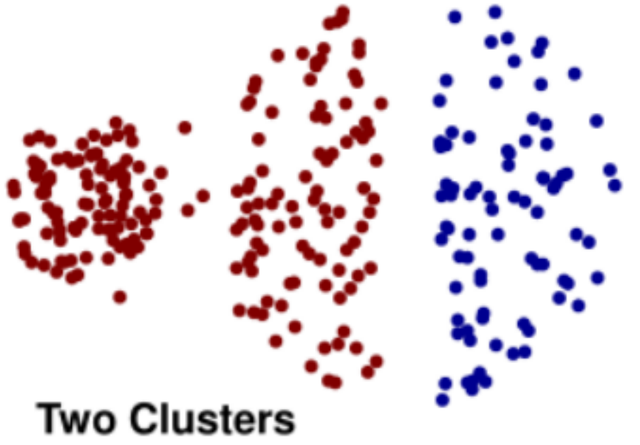
Six Clusters

- Can handle non-elliptical shapes

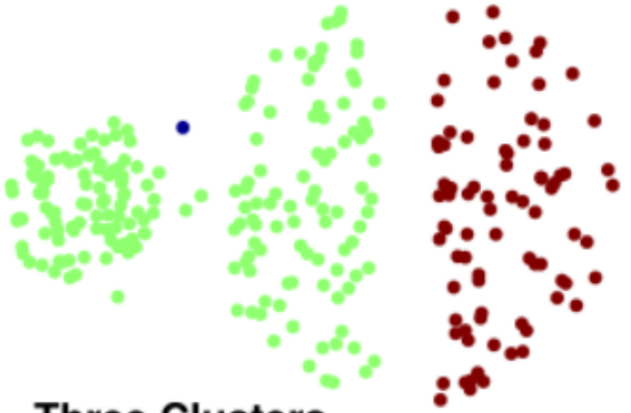
Limitations of MIN



Original Points



Two Clusters

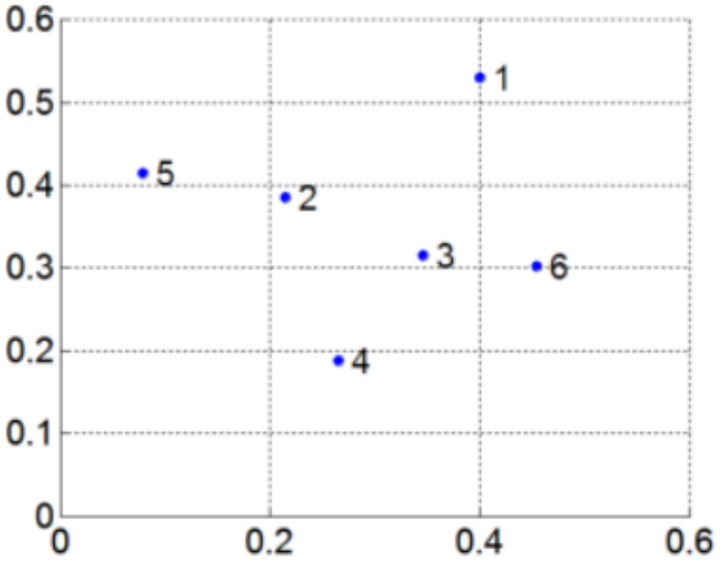


Three Clusters

- Sensitive to noise and outliers

MAX or Complete Linkage or CLIQUE

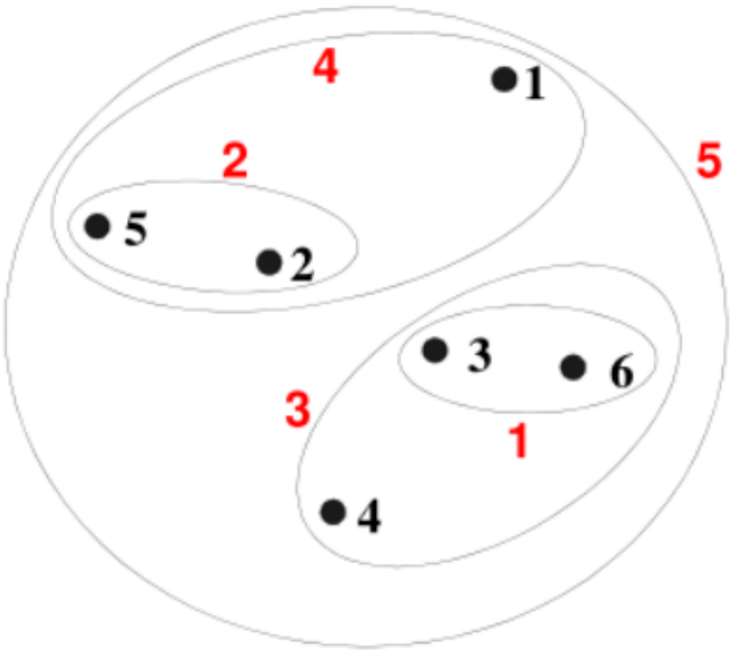
- Proximity of two clusters is based on the two most distant points in the different clusters
  - Determined by all pairs of points in the two clusters



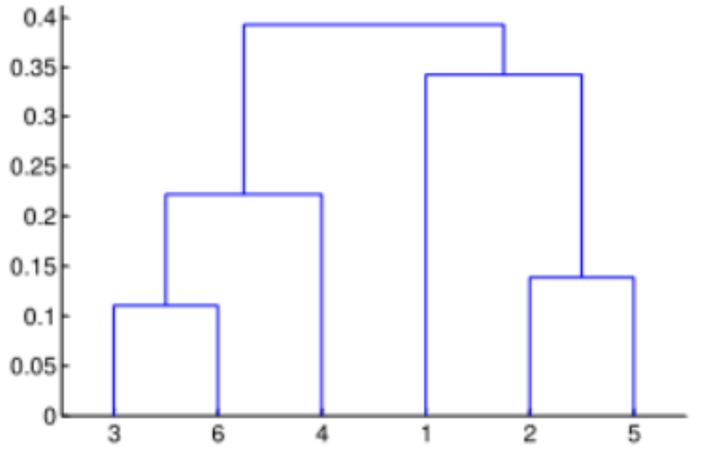
Distance Matrix:

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

Hierarchical Clustering: MAX



Nested Clusters



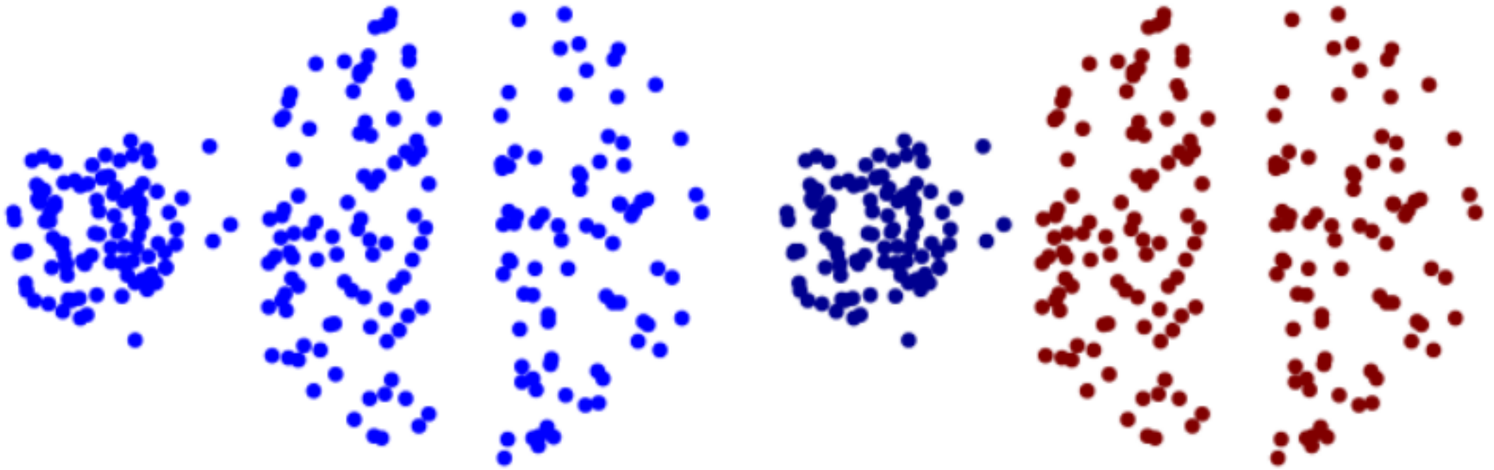
Distance Matrix:

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

As with single link, points 3 and 6
are merged first. However, $\{3,6\}$ is merged with $\{4\}$, instead of $\{2,5\}$ or $\{1\}$ because

$$
\begin{aligned}
dist(\{3,6\},\{4\}) &= \max(dist(3,4),dist(6,4)) \\
&= \max(0.15,0.22) \\
&= 0.22. \\
dist(\{3,6\},\{2,5\}) &= \max(dist(3,2),dist(6,2),dist(3,5),dist(6,5)) \\
&= \max(0.15,0.25,0.28,0.39) \\
&= 0.39. \\
dist(\{3,6\},\{1\}) &= \max(dist(3,1),dist(6,1)) \\
&= \max(0.22,0.23) \\
&= 0.23.
\end{aligned}
$$

Strength of MAX



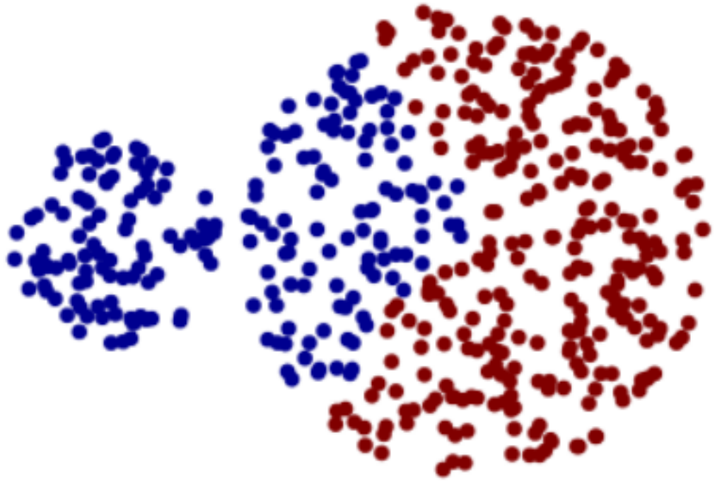
Original Points

Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX

Original Points



Two Clusters

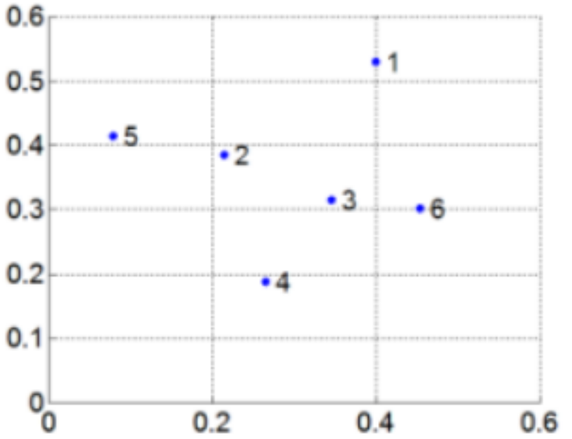
- Tends to break large clusters
- Biased towards globular clusters

Group Average

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

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

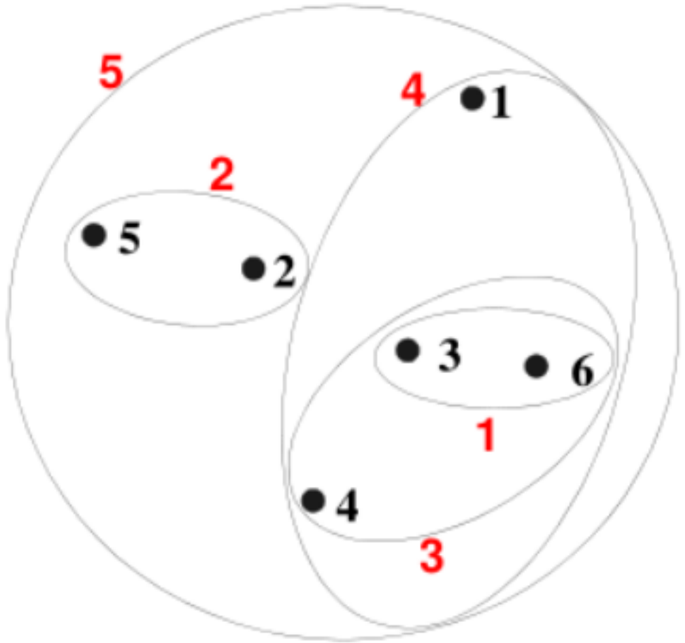
- Need to use average connectivity for scalability since total proximity favors large clusters



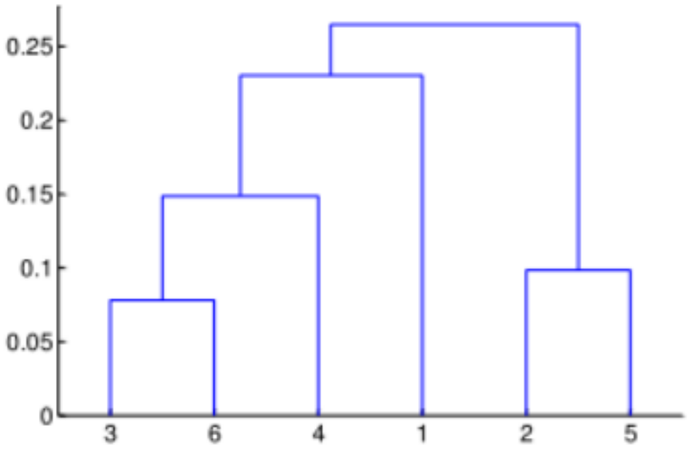
Distance Matrix:

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

Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

Example 8.6 (Group Average). Figure 8.18 shows the results of applying the group average approach to the sample data set of six points. To illustrate how group average works, we calculate the distance between some clusters.

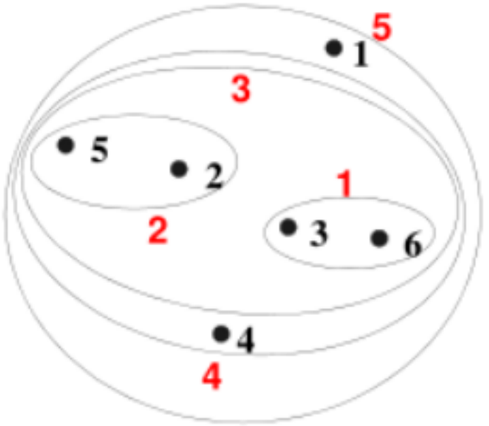
$$
\begin{aligned}
dist(\{3,6,4\},\{1\}) &= (0.22+0.37+0.23)/(3*1) \\
&= 0.28 \\
dist(\{2,5\},\{1\}) &= (0.2357+0.3421)/(2*1) \\
&= 0.2889 \\
dist(\{3,6,4\},\{2,5\}) &= (0.15+0.28+0.25+0.39+0.20+0.29)/(6*2) \\
&= 0.26
\end{aligned}
$$

Because $dist(\{3,6,4\},\{2,5\})$ is smaller than $dist(\{3,6,4\},\{1\})$ and $dist(\{2,5\},\{1$ clusters $\{3,6,4\}$ and $\{2,5\}$ are merged at the fourth stage. ■

Hierarchical Clustering: Group Average

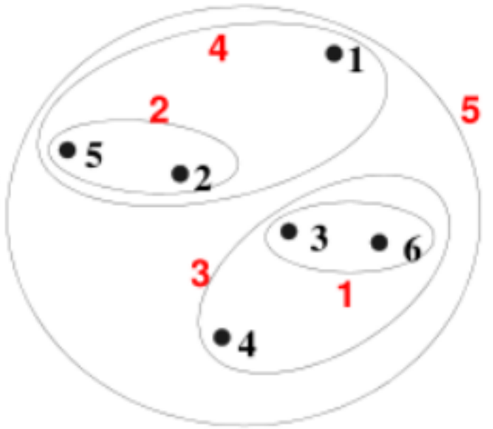
- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

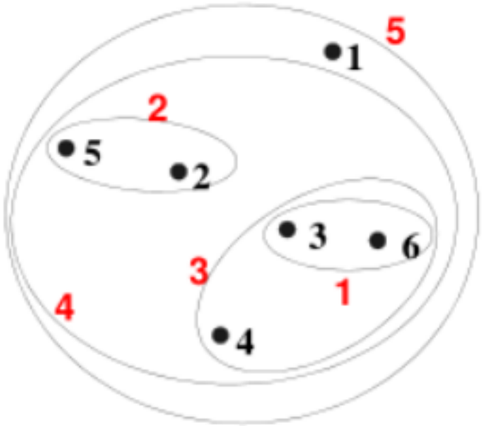
Hierarchical Clustering: Comparison



MIN

MAX





Group Average

Hierarchical Clustering: Time Complexity

- Space complexity: $O(n^2)$
- Time complexity:

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

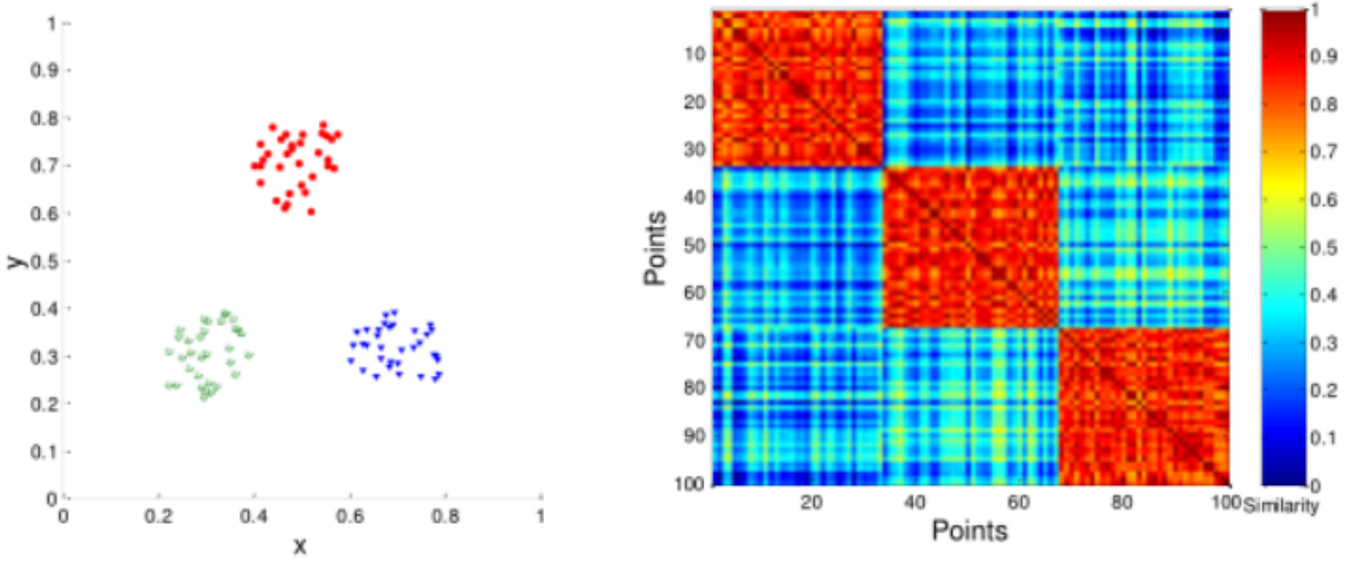
- $O(n^3)$
  - *n* steps (number of merges)
  - At each step: proximity matrix must be searched: n^2

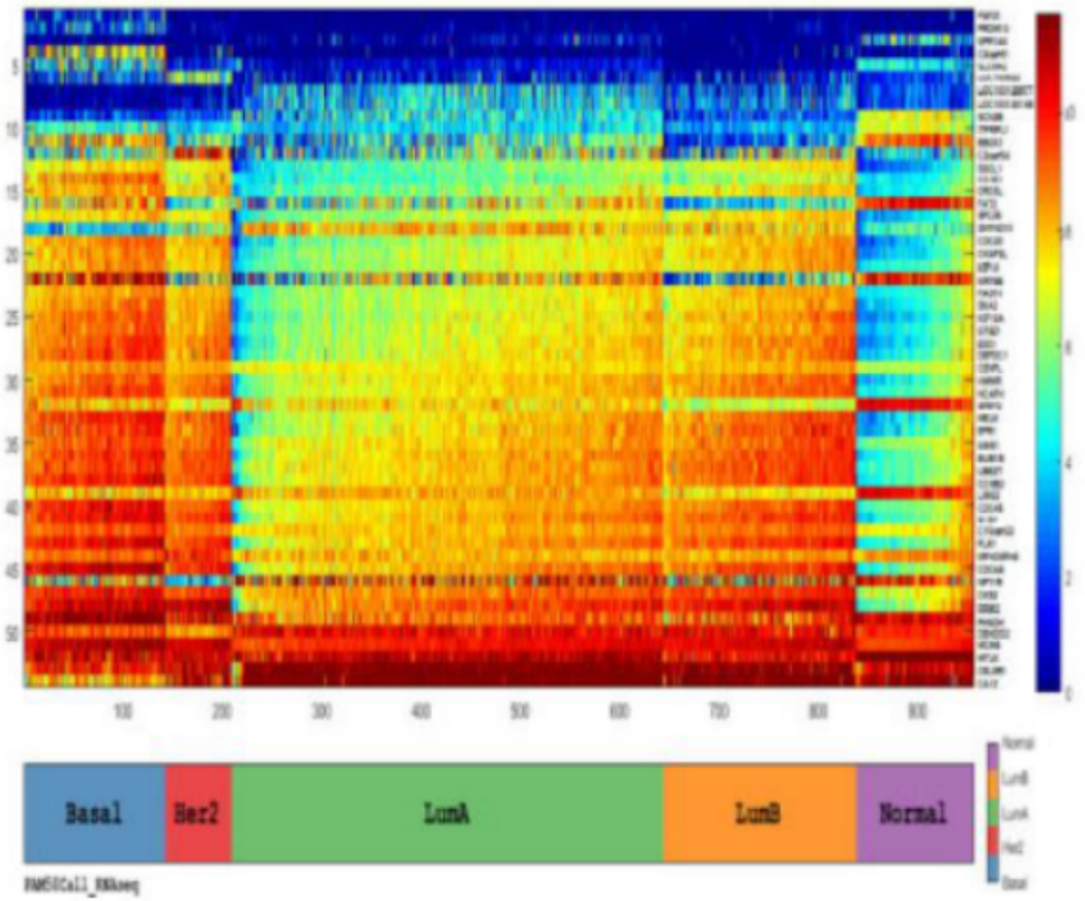
Hierarchical Clustering: Problems and Limitations

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

Using Similarity Matrix for Cluster Validation

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

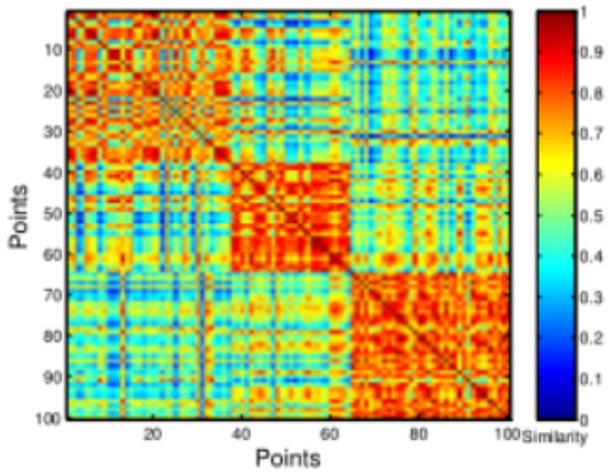


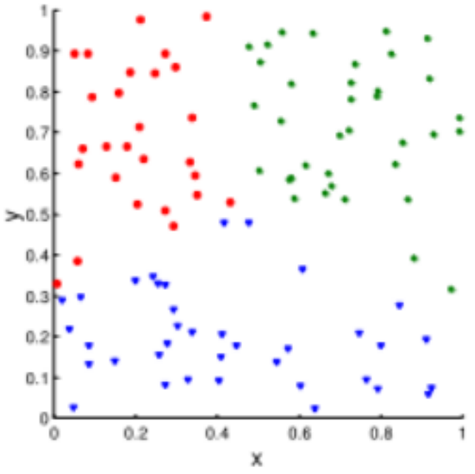


(a) Segregation of five breast cancer subtypes based on differential expression of identified 54 biomarker genes for 956 patients.

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp





K-means

Internal Measures: SSE

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