Tema 6
Clustering

Index

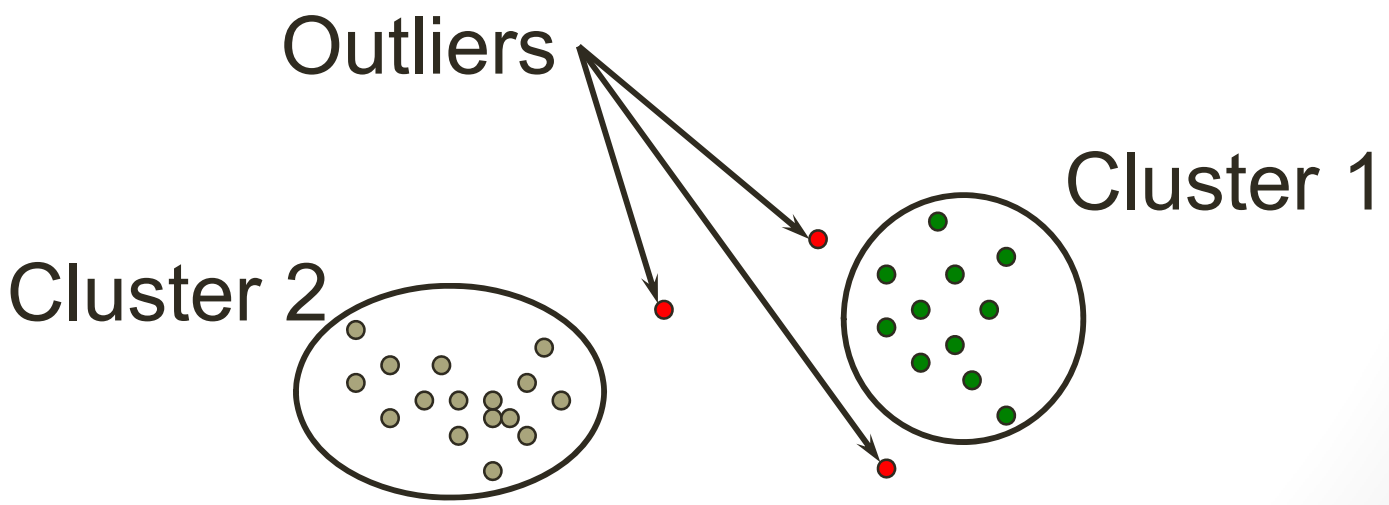
- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

Index

- **Clustering**
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

Clustering

- Group data into clusters
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
  - Unsupervised learning: no predefined classes



Why is Clustering useful?

- "Discovery" of new knowledge from data
  - Contrast with supervised classification (where labels are known)
  - Long history in the sciences of categories, taxonomies, etc
  - Can be very useful for summarizing large data sets
    - For large n and/or high dimensionality

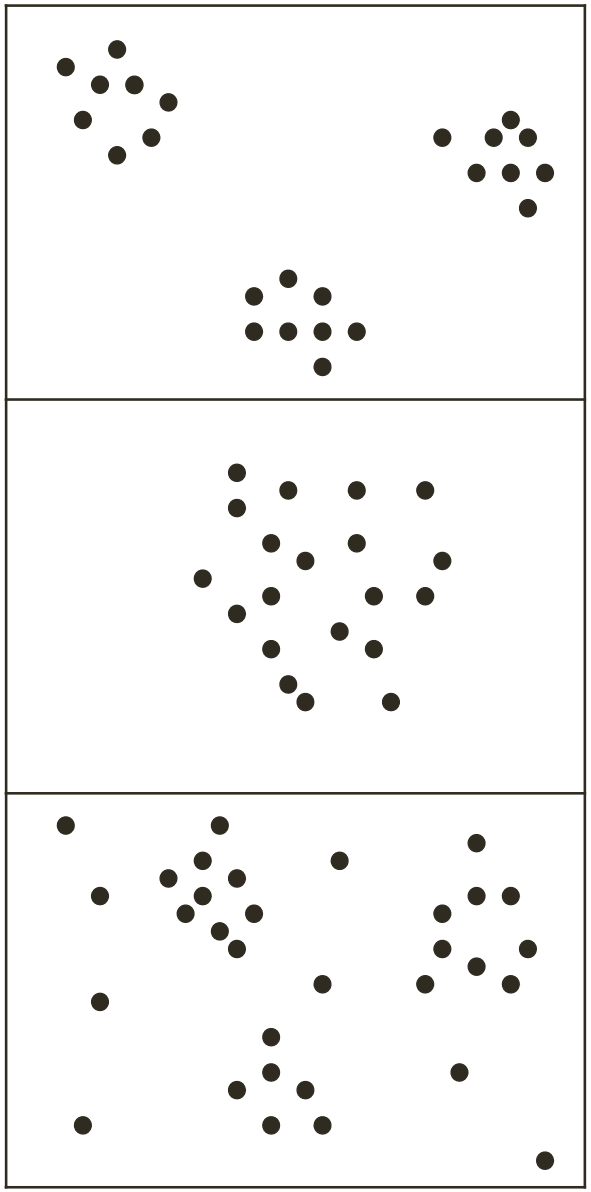
- Applications of clustering
  - Clustering of documents produced by a search engine
  - Segmentation of customers for an e-commerce store
  - Discovery of new types of galaxies in astronomical data
  - Clustering of genes with similar expression profiles
  - Cluster pixels in an image into regions of similar intensity

  - …. many more

Index

- Clustering
  - Definition
  - Areas of application
- **Distances and Measures**
- Algorithms:
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

Clustering



Sometimes easy

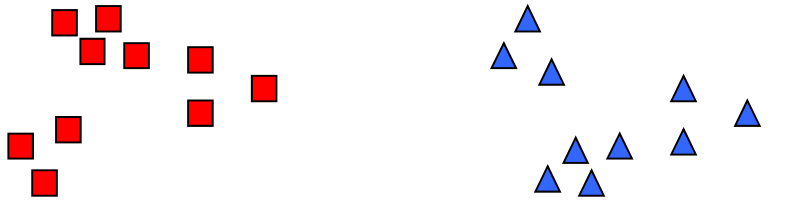
Sometimes impossible

and sometimes in between

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters



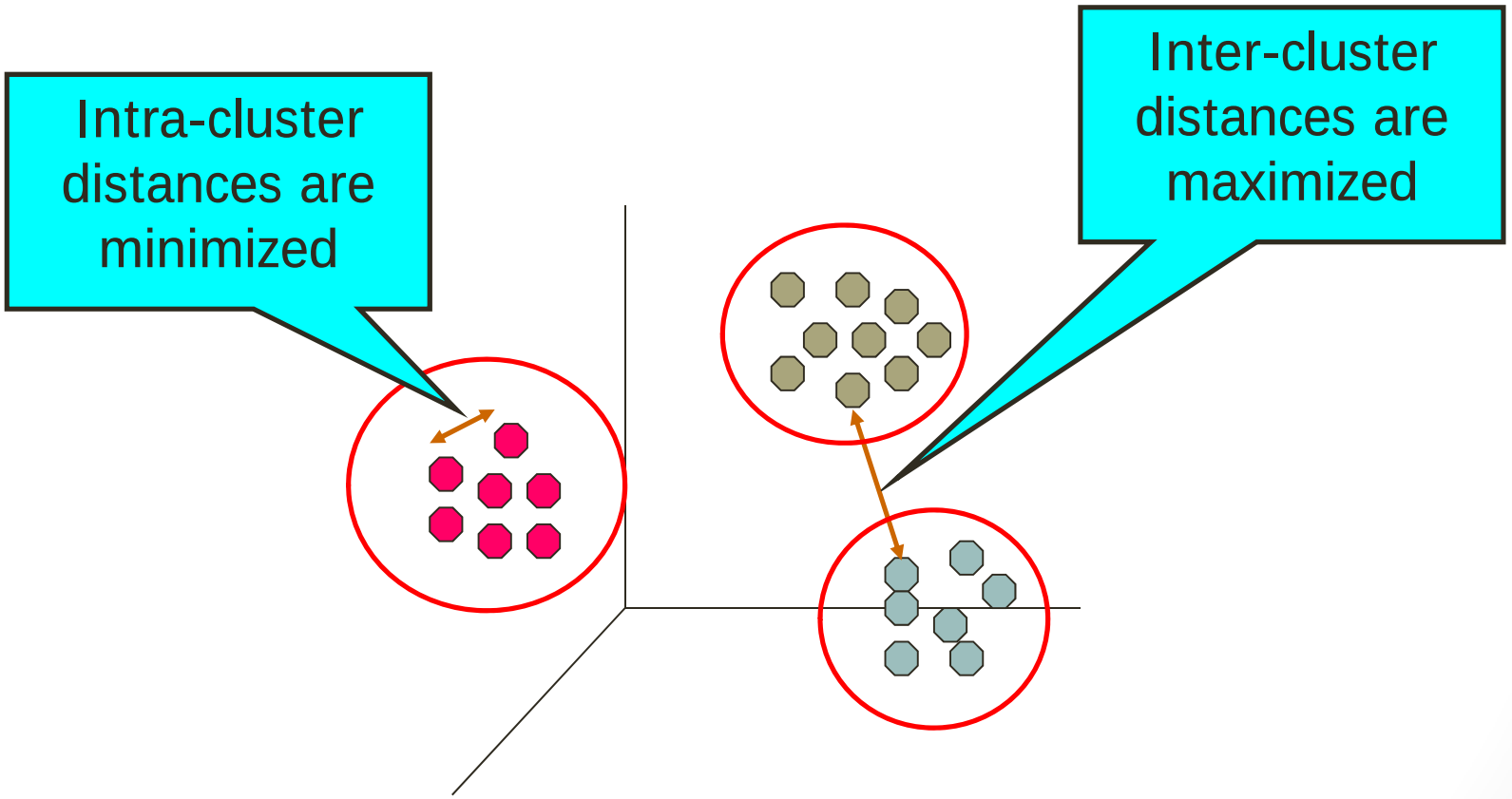
Two Clusters

Four Clusters

What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity

Clustering


Intra-cluster distances are minimized
Inter-cluster distances are maximized

What Is Good Clustering?

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Data Structures in Clustering

- Data matrix
  - (two modes)

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - (one mode)

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Measuring Similarity

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, which is typically metric: $d(i, j)$
- There is a separate "quality" function that measures the "goodness" of a cluster.
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal and ratio variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define "similar enough" or "good enough"
  - the answer is typically highly subjective.

What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features.

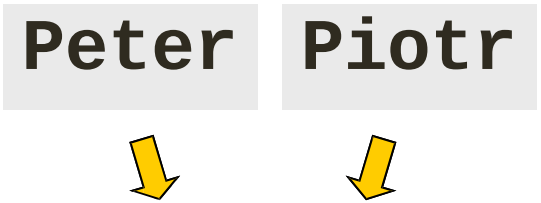
Webster's Dictionary

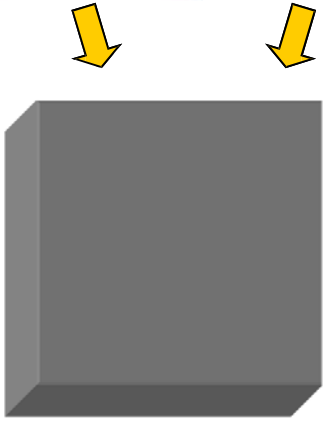
Similarity is hard to define, but… "*We know it when we see it*"

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

Defining 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 $D(O_1,O_2)$





0.23

3

342.7

SIMILARITY MEASURE

- The word ***"similarity"*** in clustering means that the value of **s(x, x')** is large when x and x' are two similar samples; the value of s(x, x') is small when x and x' are not similar.
- Similarity measure *s* is symmetric:

$$s(x, x') = s(x', x), \qquad \forall x, x' \in X$$

- Similarity measure is normalized:

$$\forall x, x' \in X$$
$$0 <= s(x, x') <= 1,$$

Dissimilarity measure is denoted by **d(x, x')** , $\forall x, x' \in X$, and it is frequently called a **distance**:

- $d(x, x') >= 0, \qquad \forall x, x' \in X$
- $d(x, x') = d(x', x), \qquad \forall x, x' \in X$
- $d(x, x'') <= d(x, x') + d(x', x''), \qquad \forall x, x', x'' \in X$ (triangular inequality)

DISTANCE MEASURES

1. Euclidean distance in m-dimensional feature space:

$$d_2(x_i, x_j) = \left(\sum_{k=1}^{m} (x_{ik} - x_{jk})^2 \right)^{1/2}$$

2. L_1 metric or **city block distance**:

$$d_1(xi, xj) = \sum_{k=1}^{m} |x_{ik} - x_{jk}|$$

3. Minkowski metric (includes the Euclidean distance and city block distance as special cases):

$$d_p(x_i, x_j) = \left(\sum_{k=1}^{m} (x_{ik} - x_{jk})^p \right)^{1/p}$$

4. Cosine-correlation:

$$\text{scos}(x_i, x_j) = \frac{\sum_{k=1}^{m} (x_{ik}\ x_{jk})}{\left(\sum_{k=1}^{m} x_{ik}^2 \times \sum_{k=1}^{m} x_{jk}^2 \right)^{1/2}}$$

$$\text{scos}(x_i, x_j) = 1 \Leftrightarrow \forall i, j \text{ and } \lambda > 0 \text{ where } x_i = \lambda \cdot x_j$$

$$\text{scos}(x_i, x_j) = -1 \Leftrightarrow \forall i, j \text{ and } \lambda < 0 \text{ where } x_i = \lambda \cdot x_j$$

DISTANCE MEASURES (example)

For 4-dimensional vectors:

$\mathbf{x_1 = \{1, 0, 1, 0\}}$ and $\mathbf{x_2 = \{2, 1, -3, -1\}}$

1.Euclidean distance in m-dimensional feature space:

$$d_2(x_i, x_j) = (1 + 1 + 16 + 1)^{1/2} = \mathbf{4.36}$$

2.L_1 metric or **city block distance**:

$$d_1(xi, xj) = 1 + 1 + 4 + 1 = \mathbf{7}$$

3.Minkowski metric (for p=3):

$$d_3(x_i, x_j) = (1 + 1 + 64 + 1)^{1/3} = \mathbf{4.06}$$

4.Cosine-correlation:

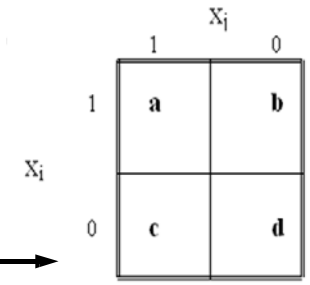
$$s_{cos}(x_i, x_j) = (2 + 0 - 3 + 0) / (2^{1/2}\ 15^{1/2}) = \mathbf{-0.18}$$

SIMILARITY / DISTANCE MEASURES

Distance measure for n-dimensional vectors with binary data:

A {0,0,1,0,0,1,0,1,0,0,1,0,1} and {1,1,1,0,0,0,1,1,1,1,0,1,0}

· Use the 2 · 2 contingency table for samples xi and xj :



a) **Simple Matching Coeficient** (SMC): $s_{smc}(x_i, x_j) = (a + d) / (a + b + c + d)$

b) **Jaccard Coefficient**: $s_{jc}(x_i, x_j) = a / (a + b + c)$

c) **Rao's Coefficient**: $s_{rc}(x_i, x_j) = a / (a + b + c + d)$

Example: For x_i = {0,0,1,1,0,1,0,1} and x_j = {0,1,1,0,0,1,0,0} parameters are

a = 2, b = 2, c = 1, and d = 3

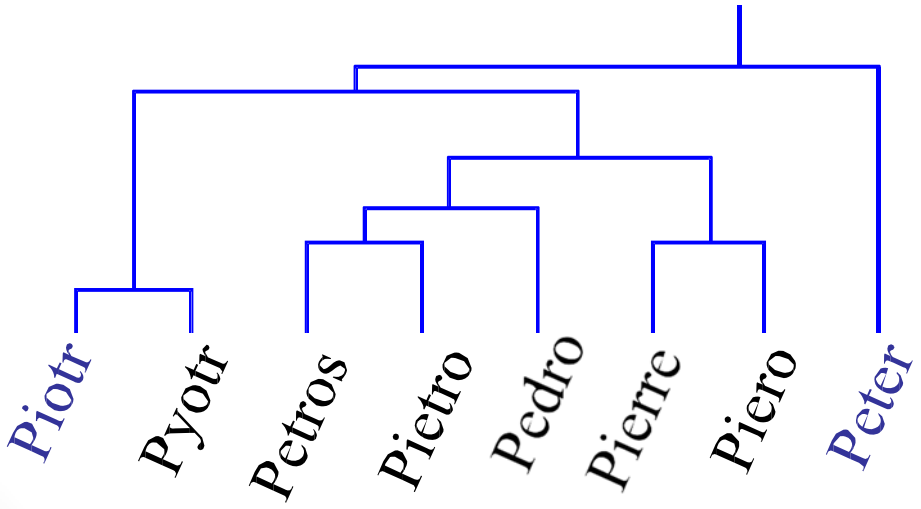
and similarity measures: $s_{smc}(x_i, x_j) = 5/8$, $s_{jc}(x_i, x_j) = 2/5$, and $s_{rc}(x_i, x_j) = 2/8$

Edit Distance Example

It is possible to transform any string *Q* into string *C*, using only *Substitution*, *Insertion* and *Deletion*.

Assume that each of these operators has a cost associated with it.

The similarity between two strings can be defined as the cost of the cheapest transformation from *Q* to *C*.



How similar are the names "Peter" and "Piotr"?

Assume the following cost function

Substitution	1 Unit
Insertion	1 Unit
Deletion	1 Unit

D(**`Peter,Piotr`**) is 3

`Peter`

Substitution (i for e)

`Piter`

Insertion (o)

`Pioter`

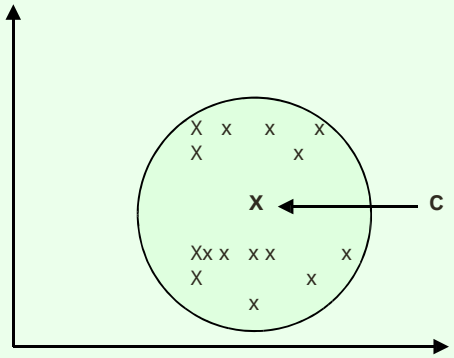
Deletion (e)

`Piotr`

Output: Cluster Representation

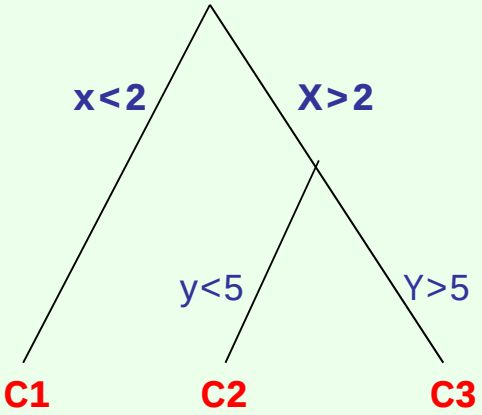
Schemas:

a)



Centroid

b)



Clustering Tree

c)

C1: X<2

C2: X>2 ∋ Y<5

C3: X>2 ∋ Y>5

Logical Expression

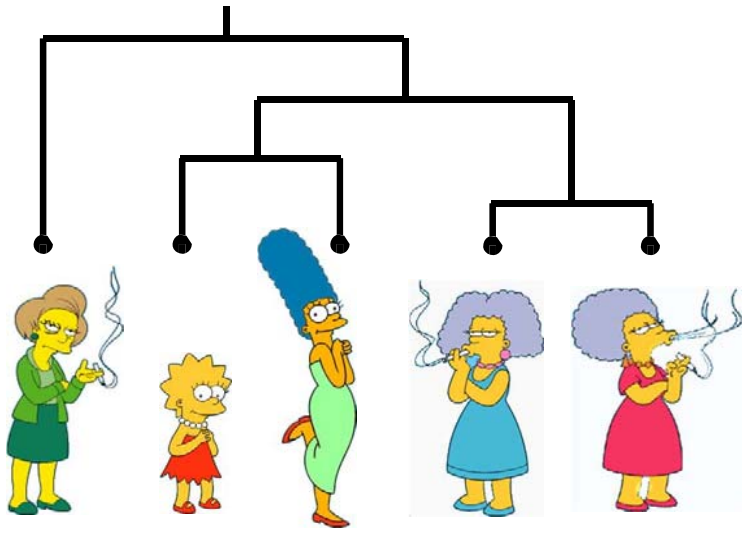
Index

- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- **Algorithms:**
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

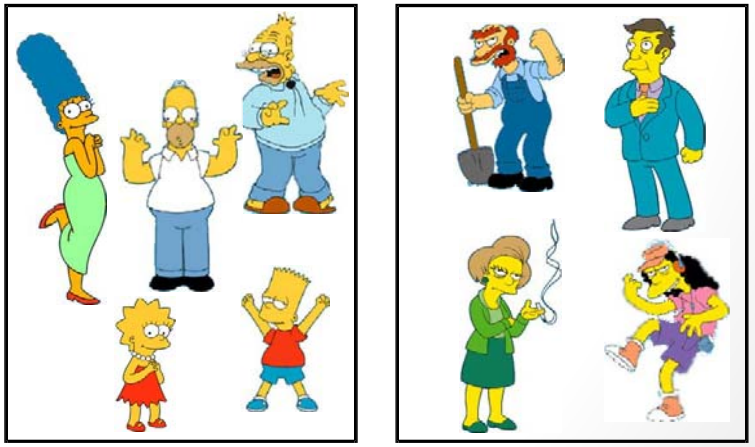
Two Main Types of Clustering

- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion
- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

Hierarchical



Partitional



Index

- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - **Partitioning algorithms: K-means**
  - Hierarchical clustering:
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

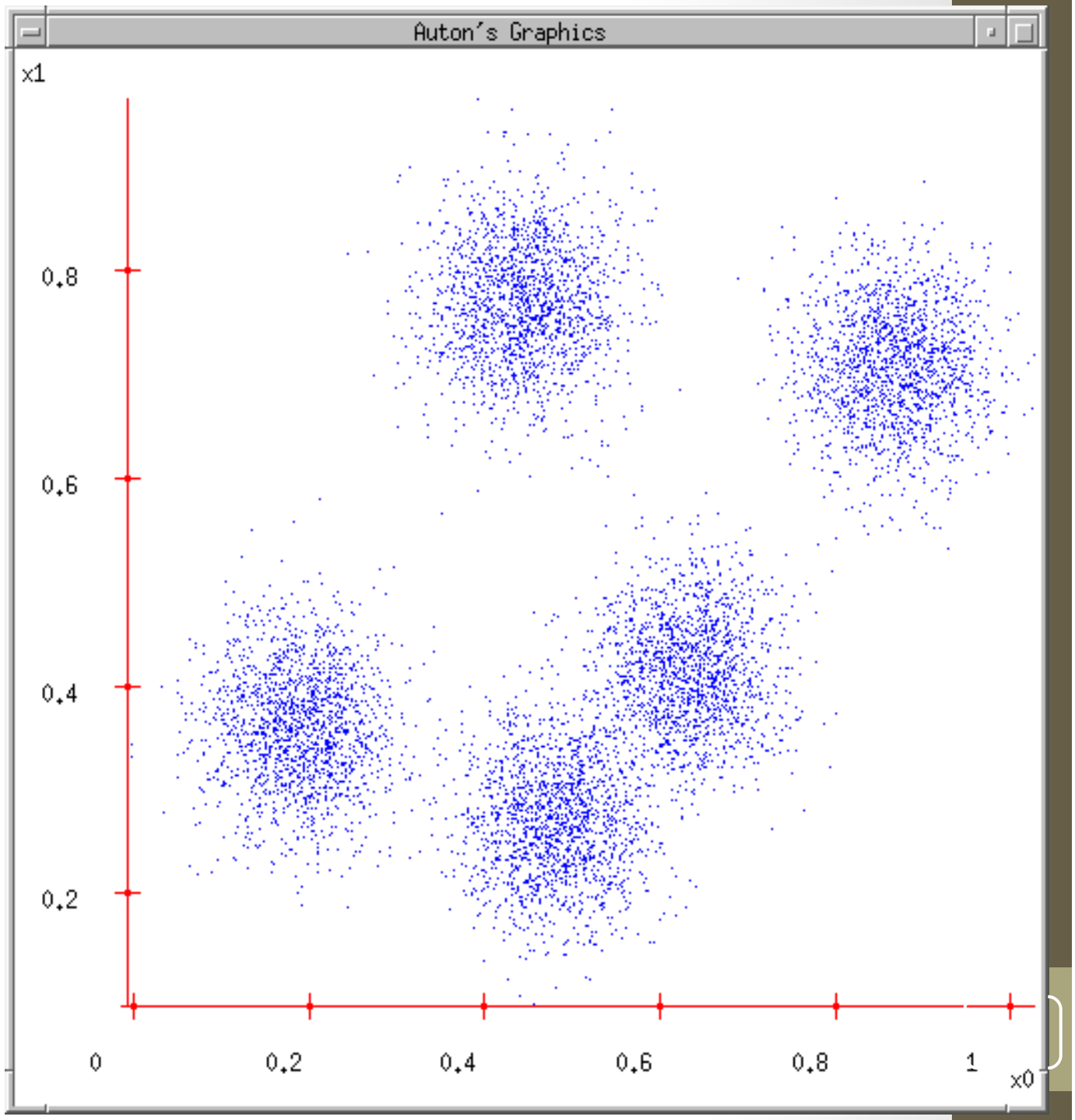
K-means Clustering

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

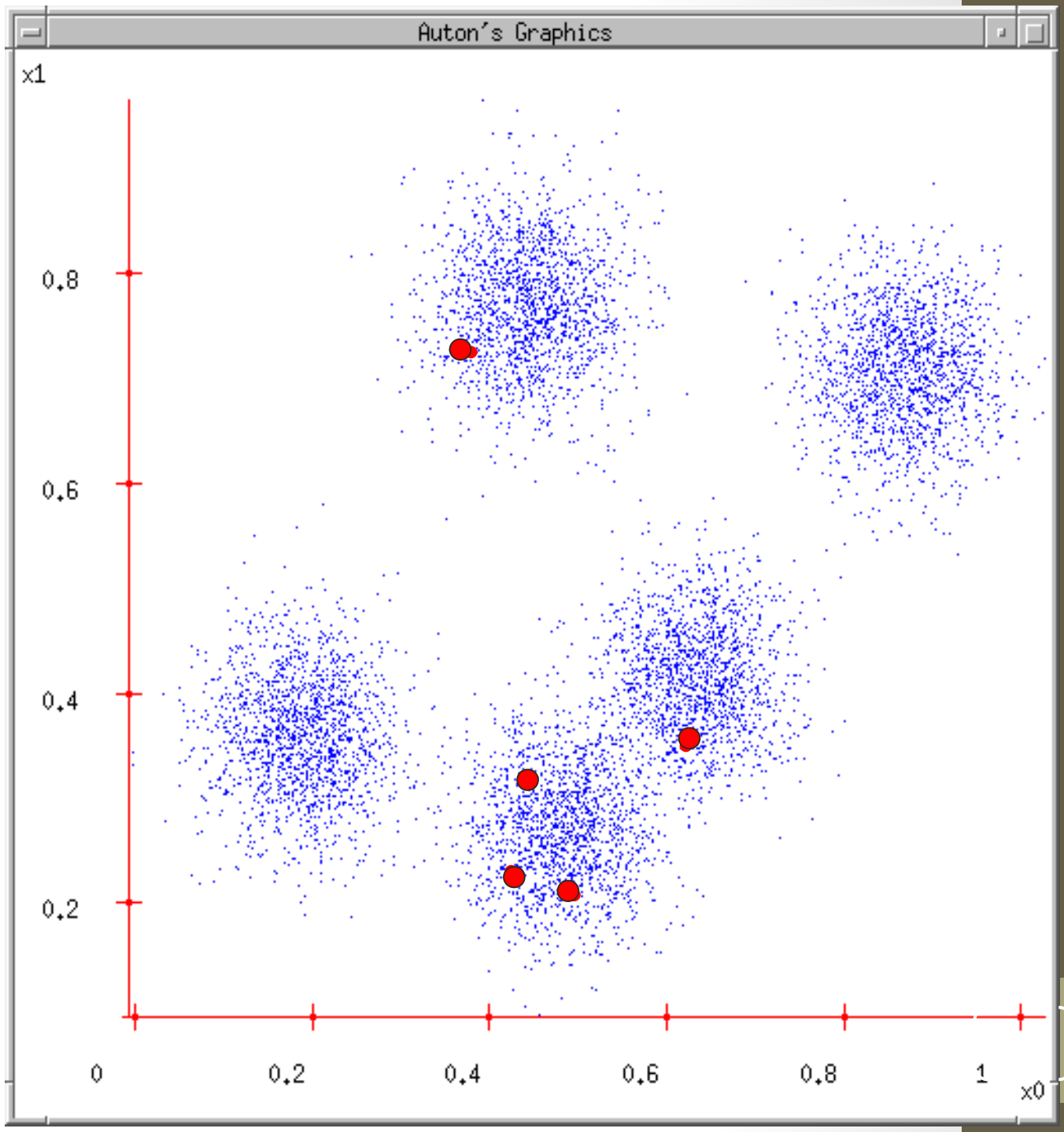
K-means

1. Ask user how many clusters they'd like. *(e.g. K=5)*



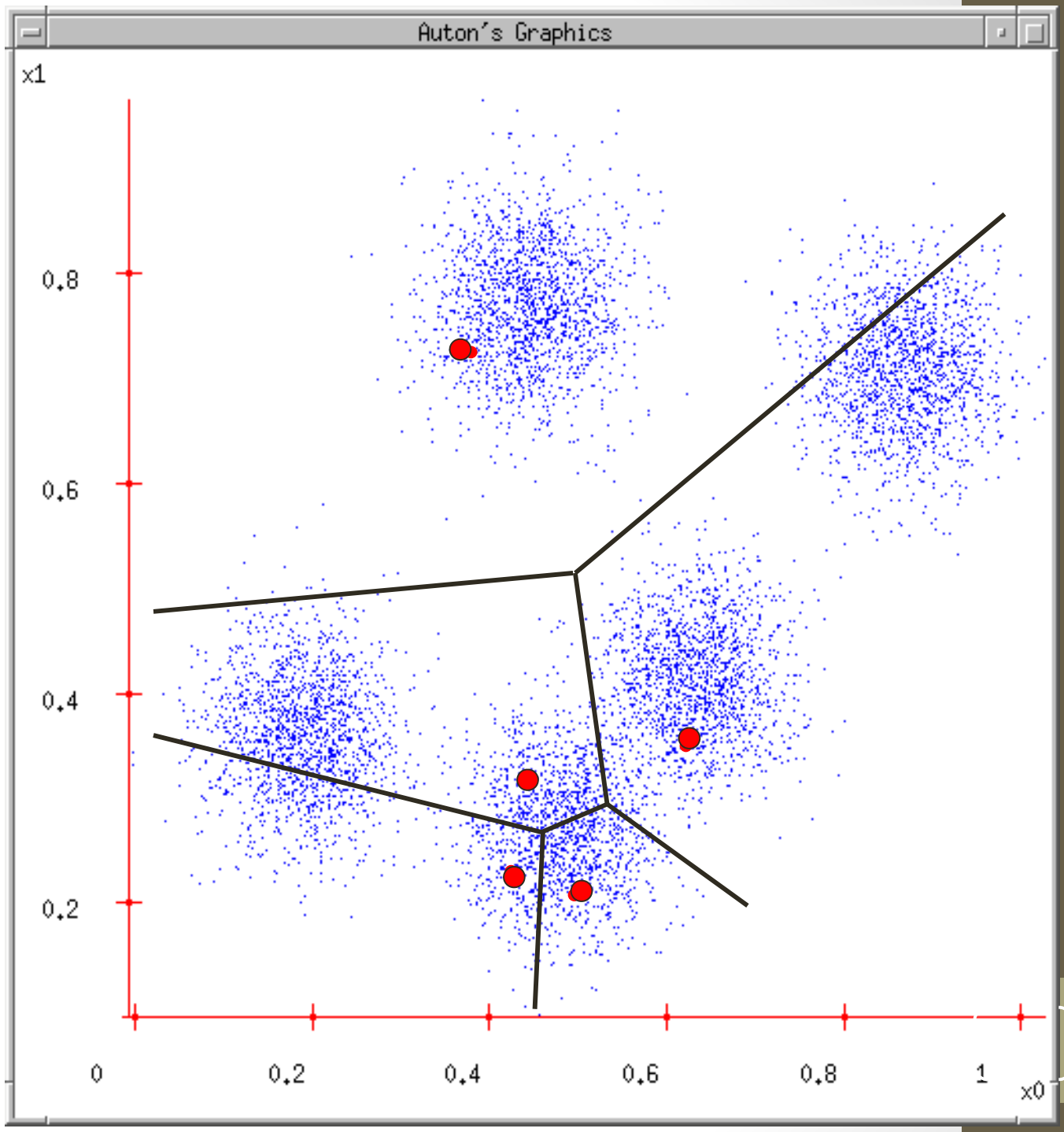
K-means

1. Ask user how many clusters they'd like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations



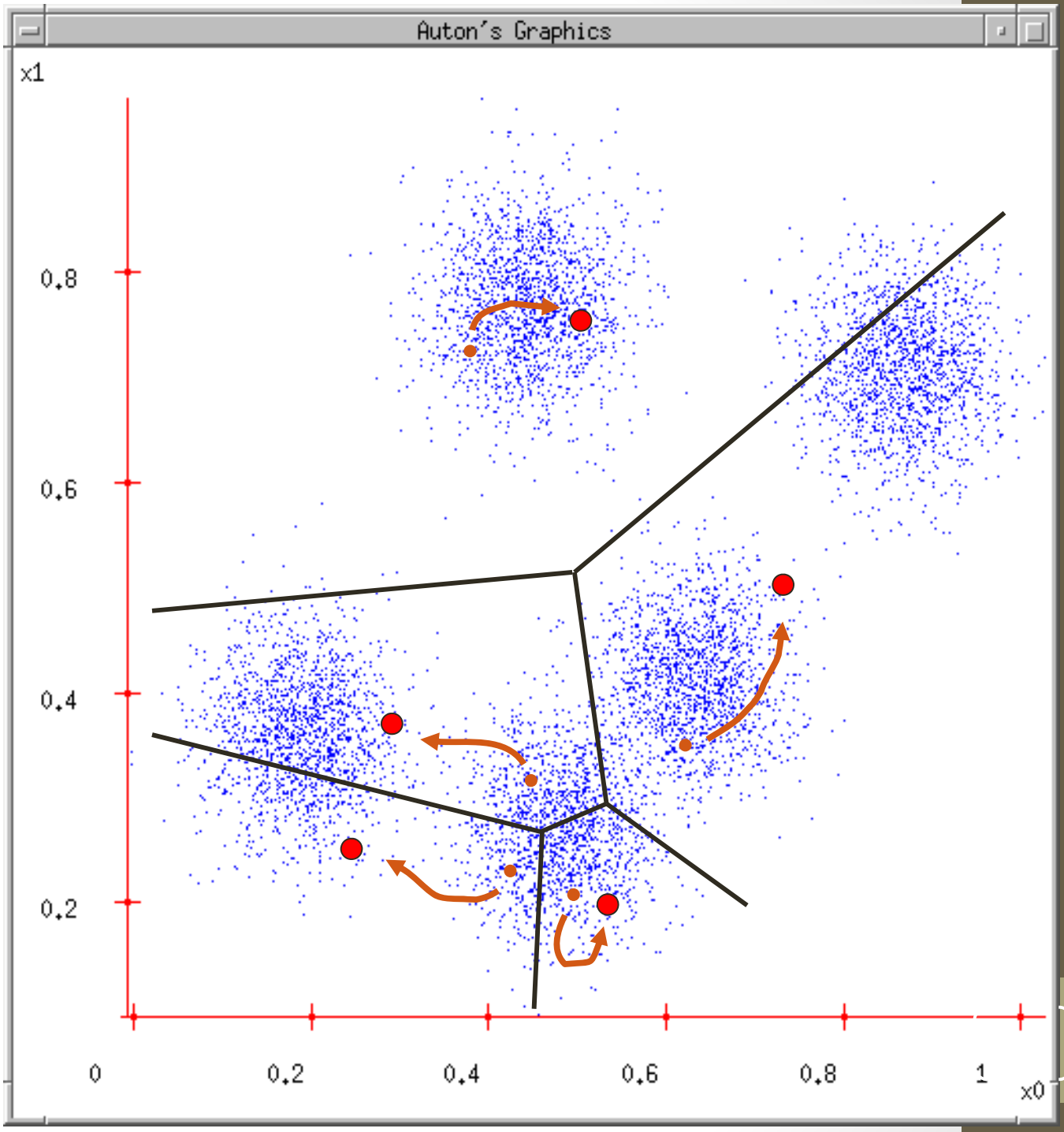
K-means

1. Ask user how many clusters they'd like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



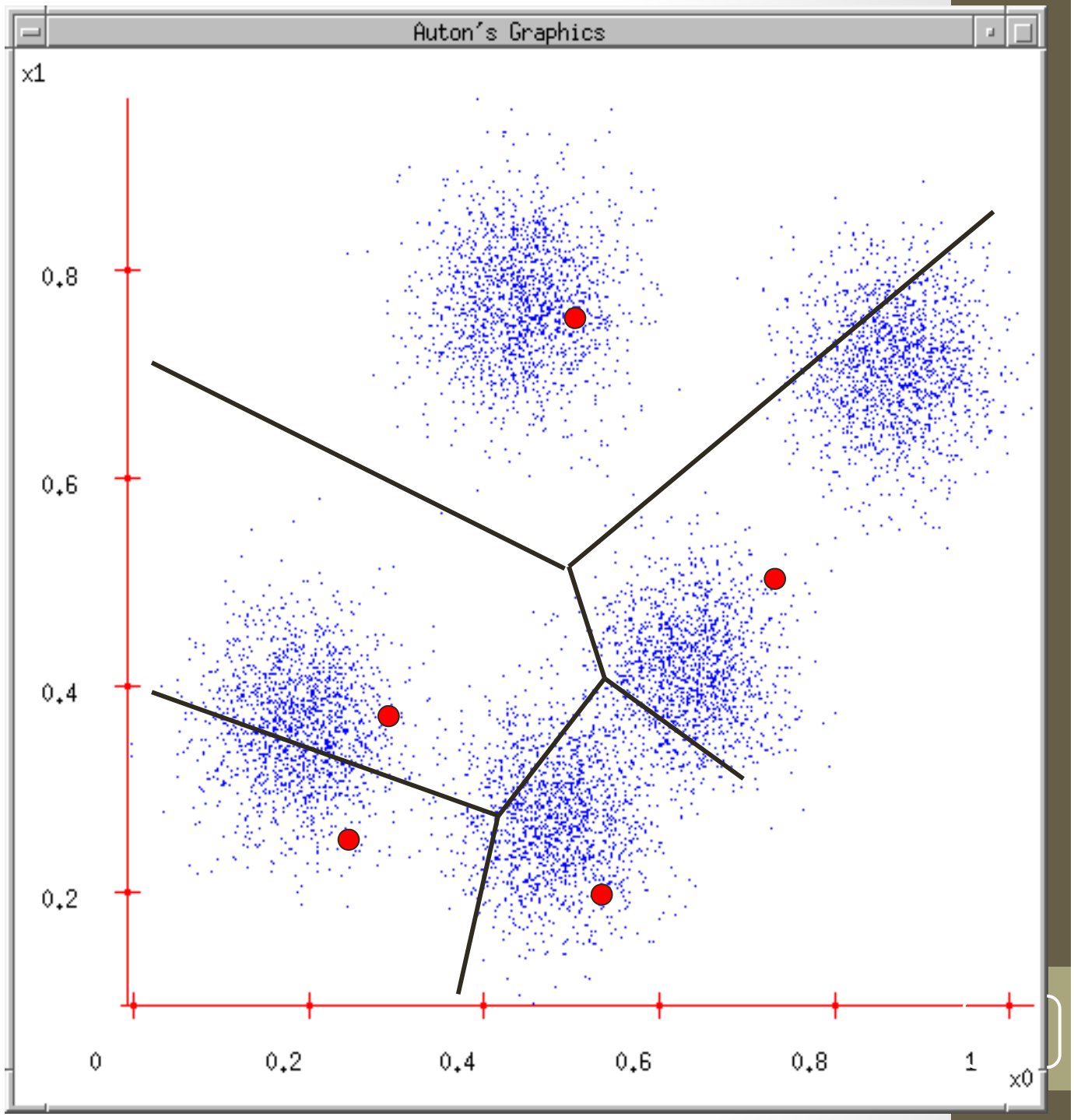
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means

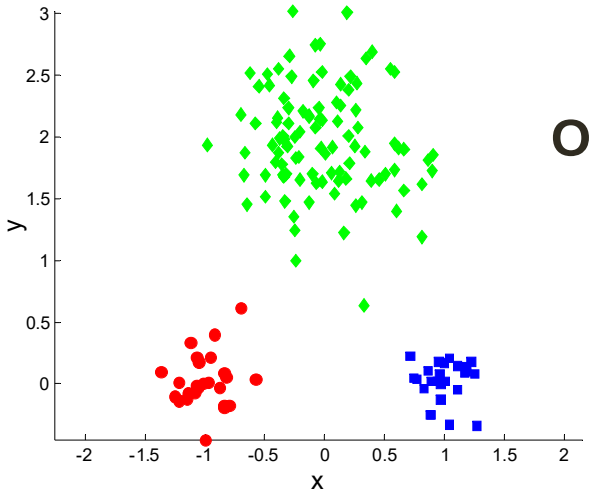
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns
5. New Centers => new boundaries
6. Repeat until no change



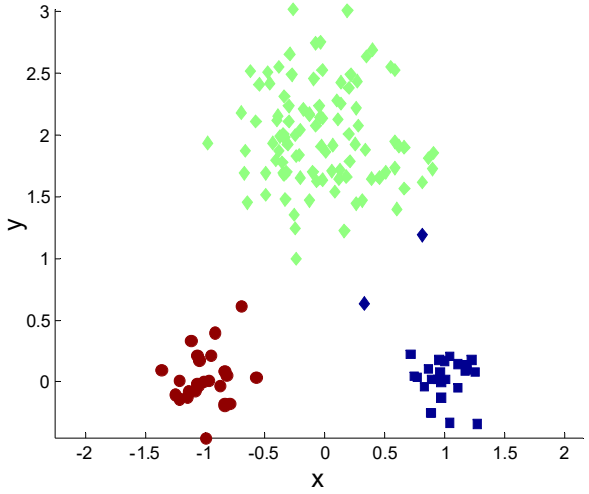
K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

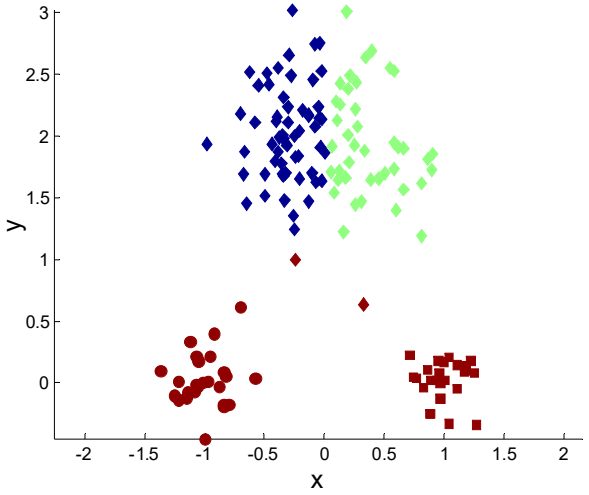
Two different K-means Clusterings



Original Points

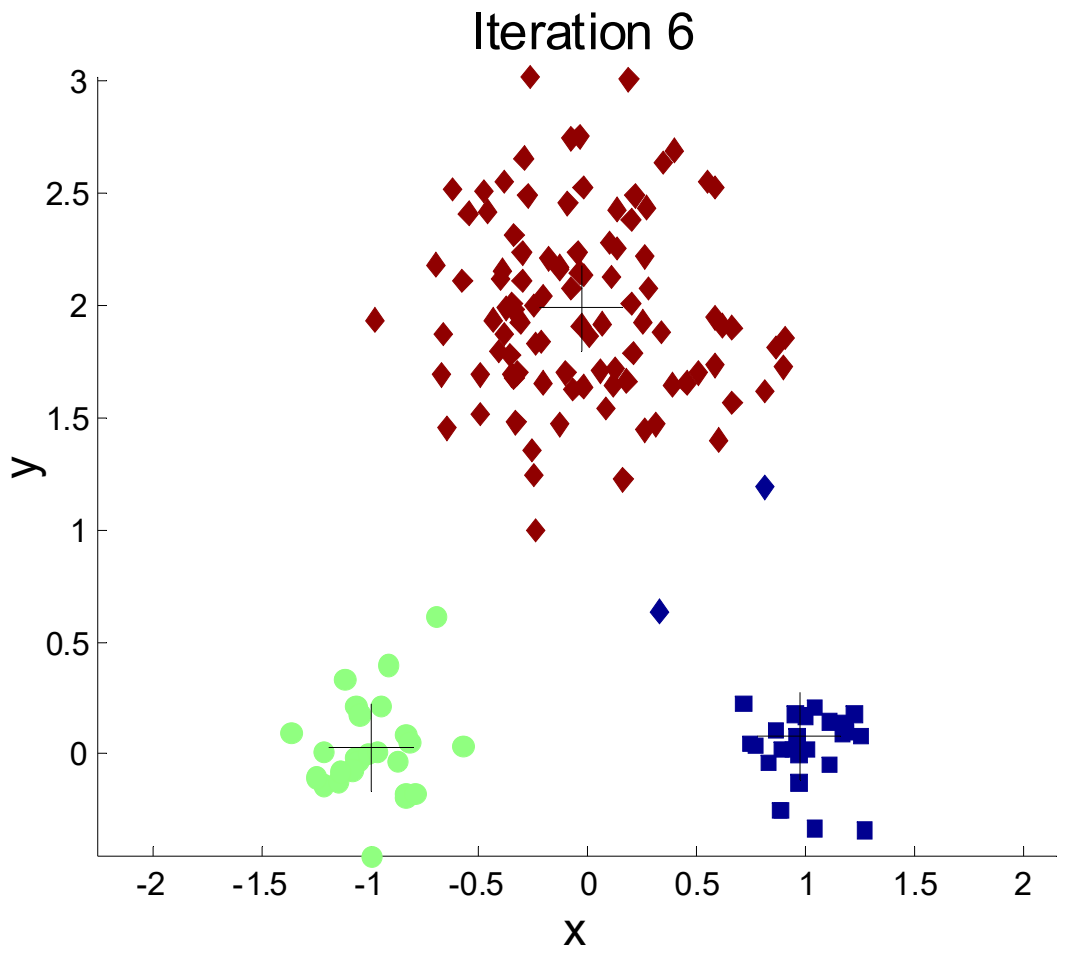


Optimal Clustering

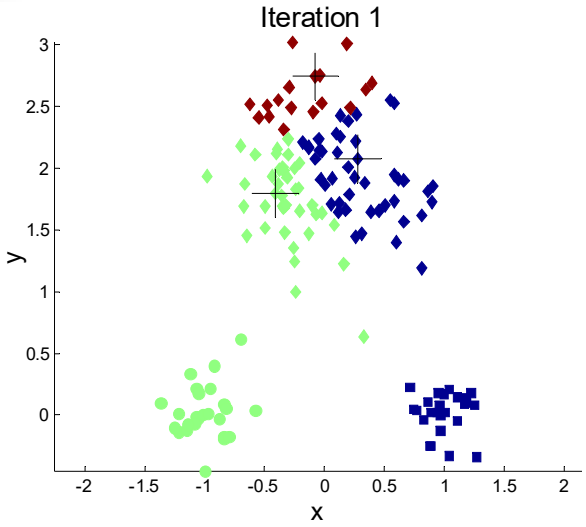


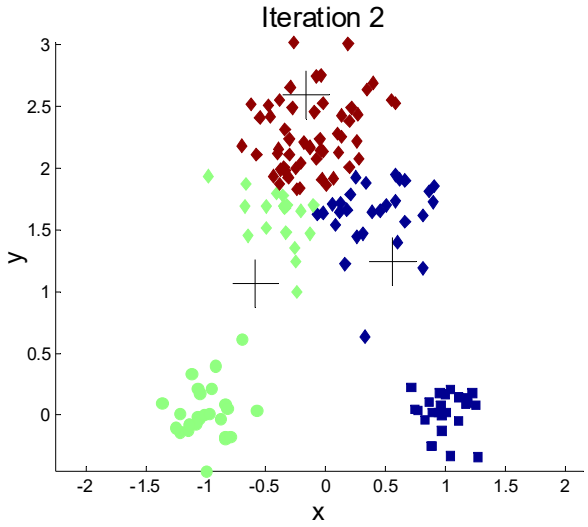
Sub-optimal Clustering

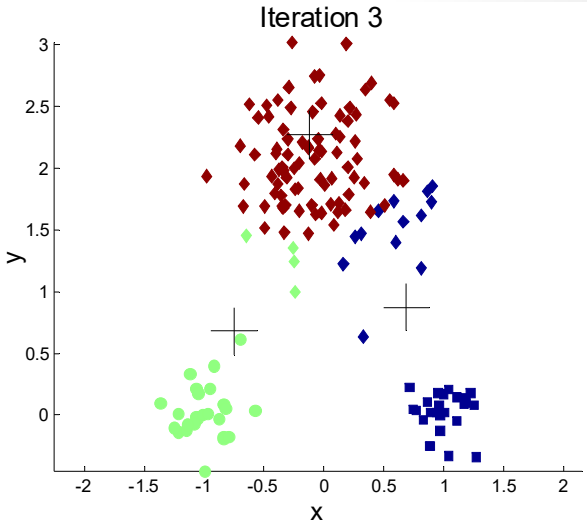
Importance of Choosing Initial Centroids

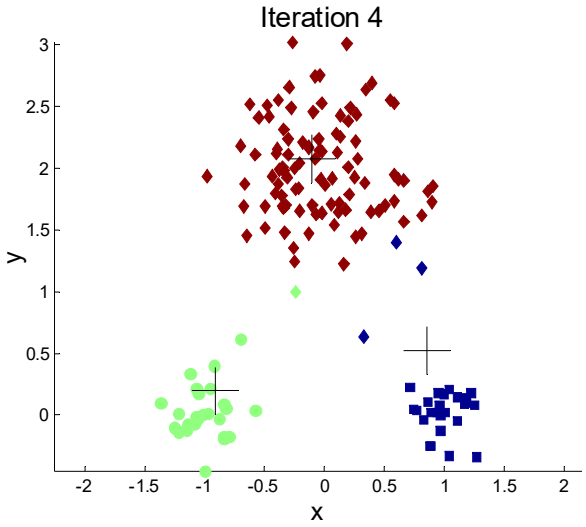


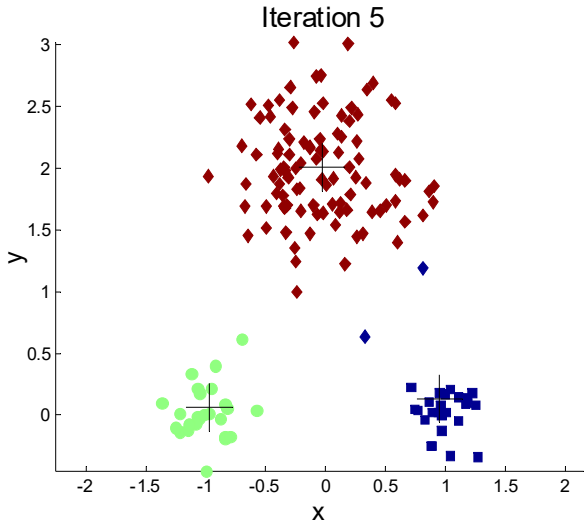
Importance of Choosing Initial Centroids

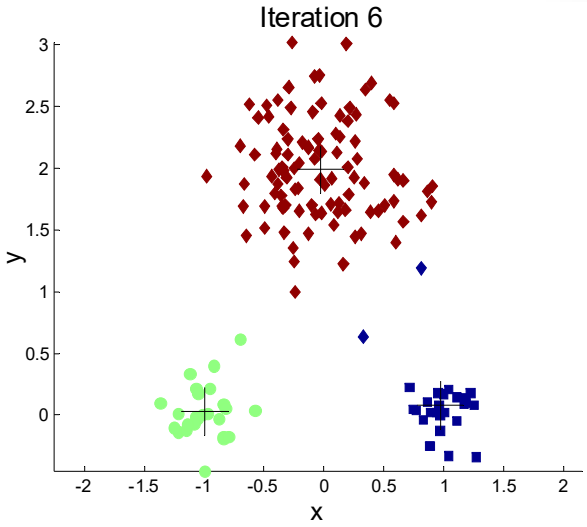




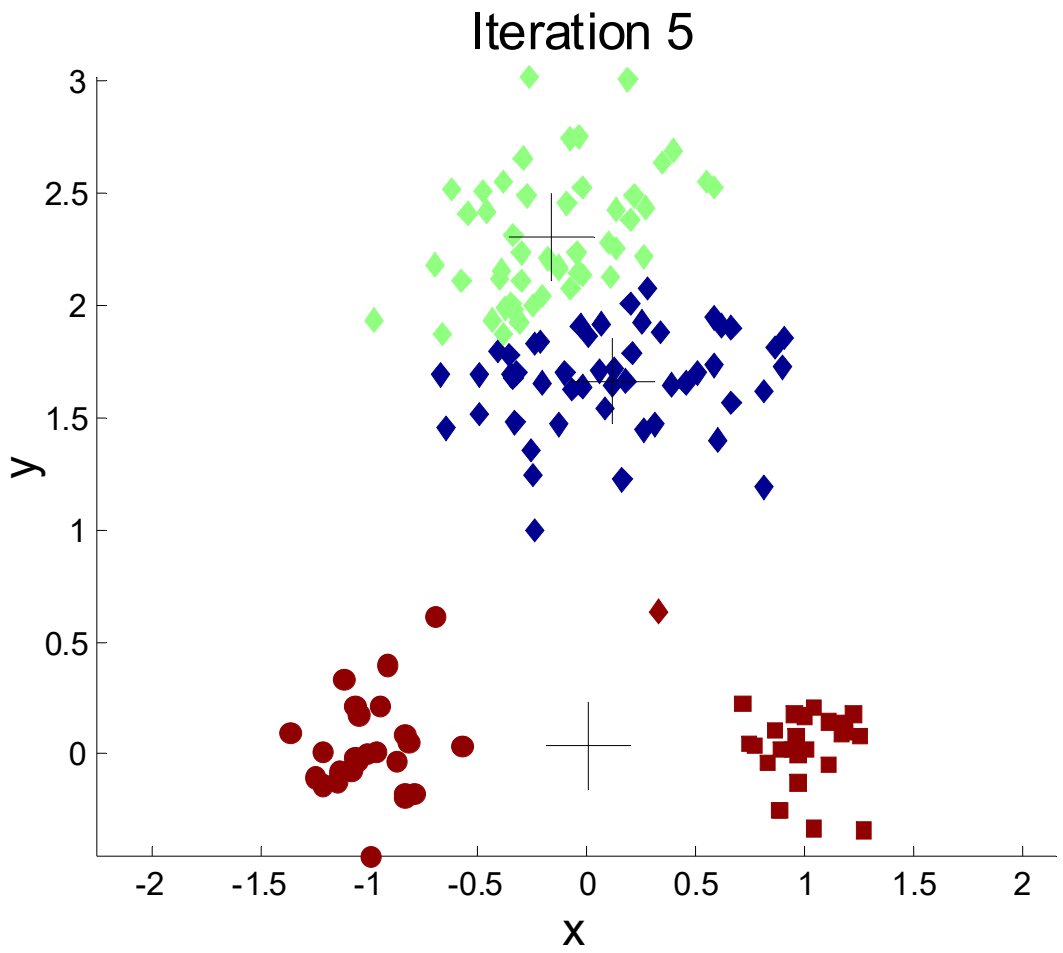




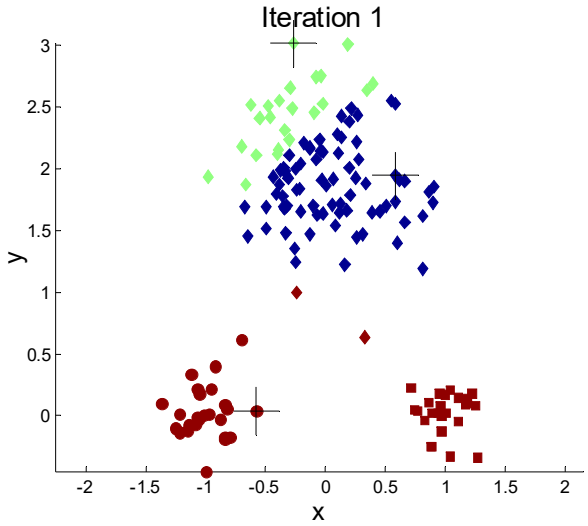


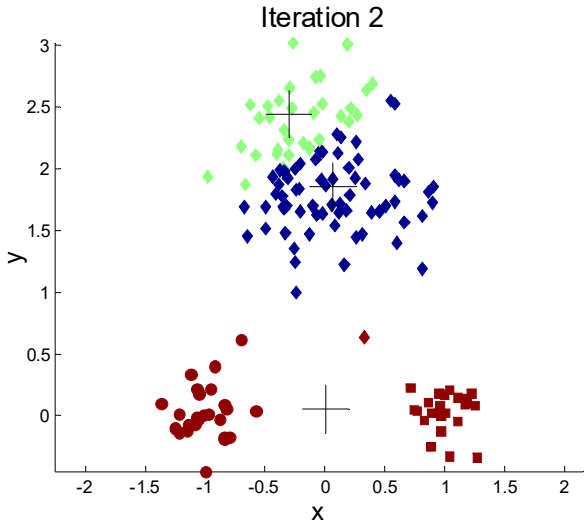


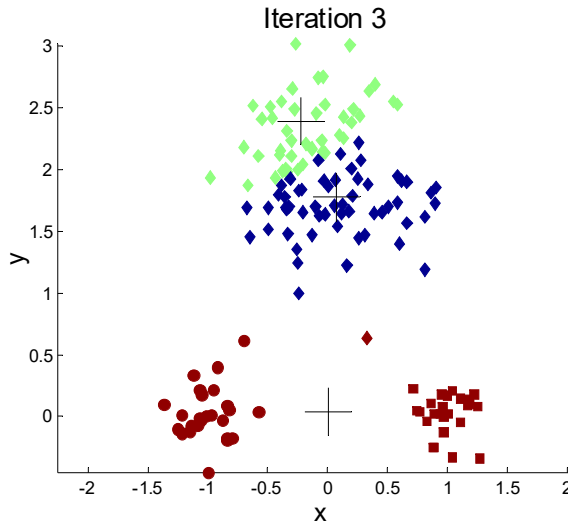
Importance of Choosing Initial Centroids …

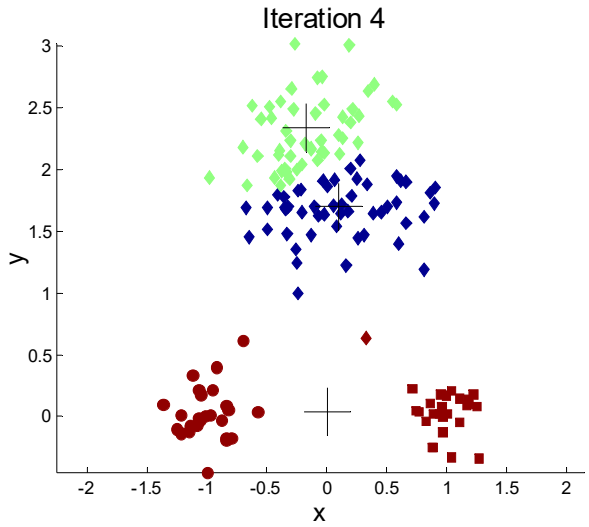


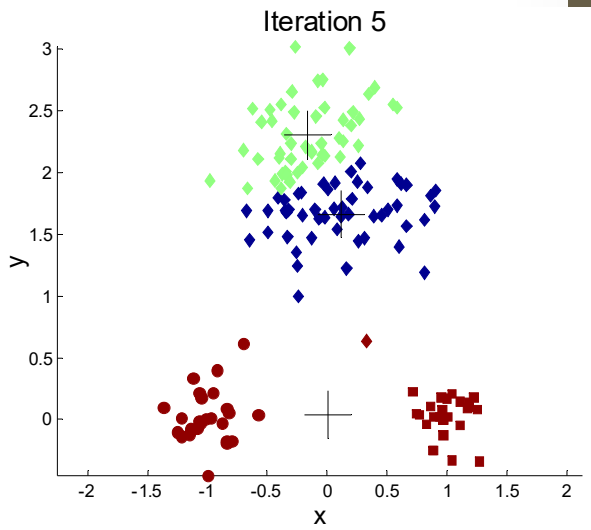
Importance of Choosing Initial Centroids …











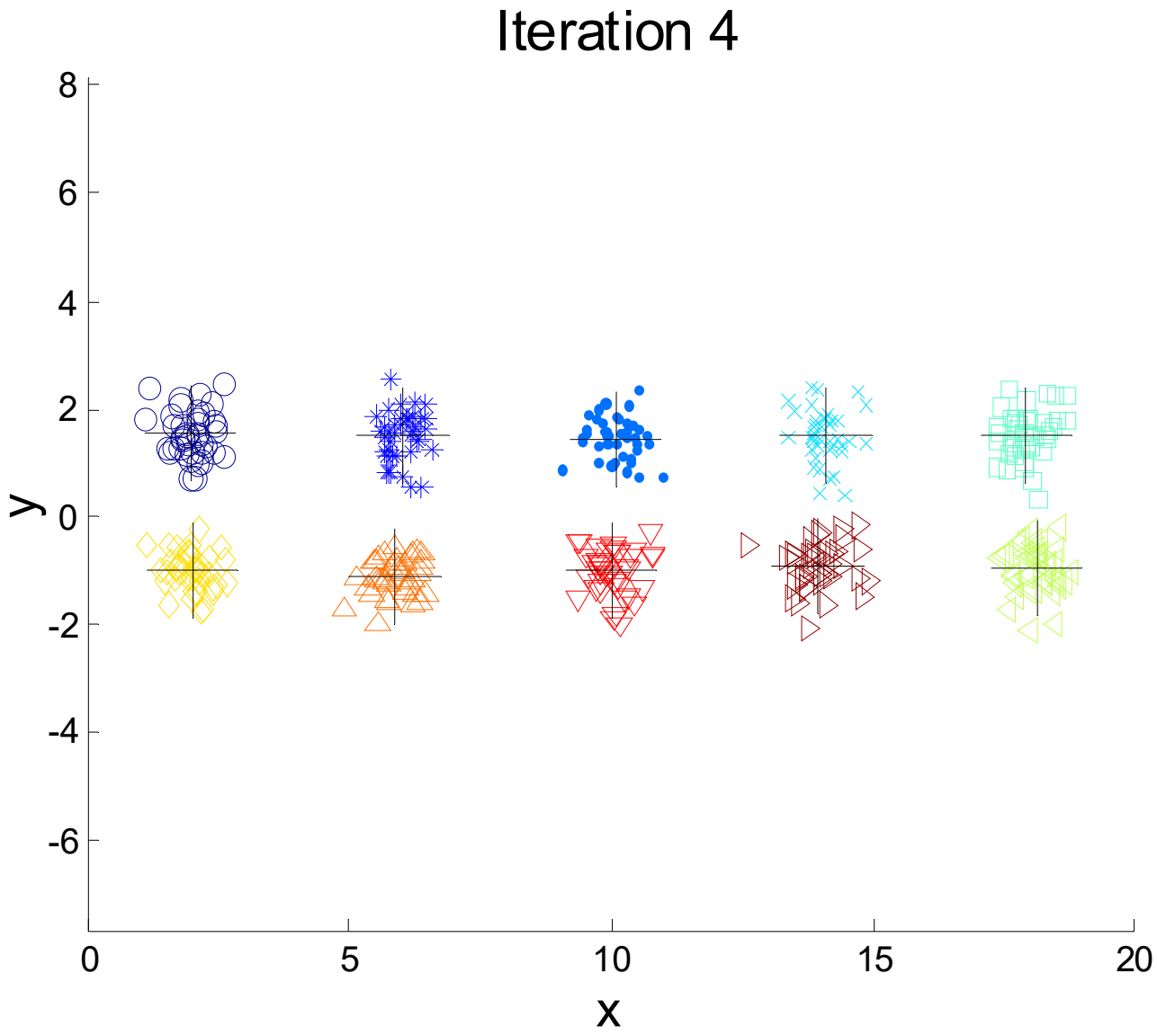
Problems with Selecting Initial Points

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

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

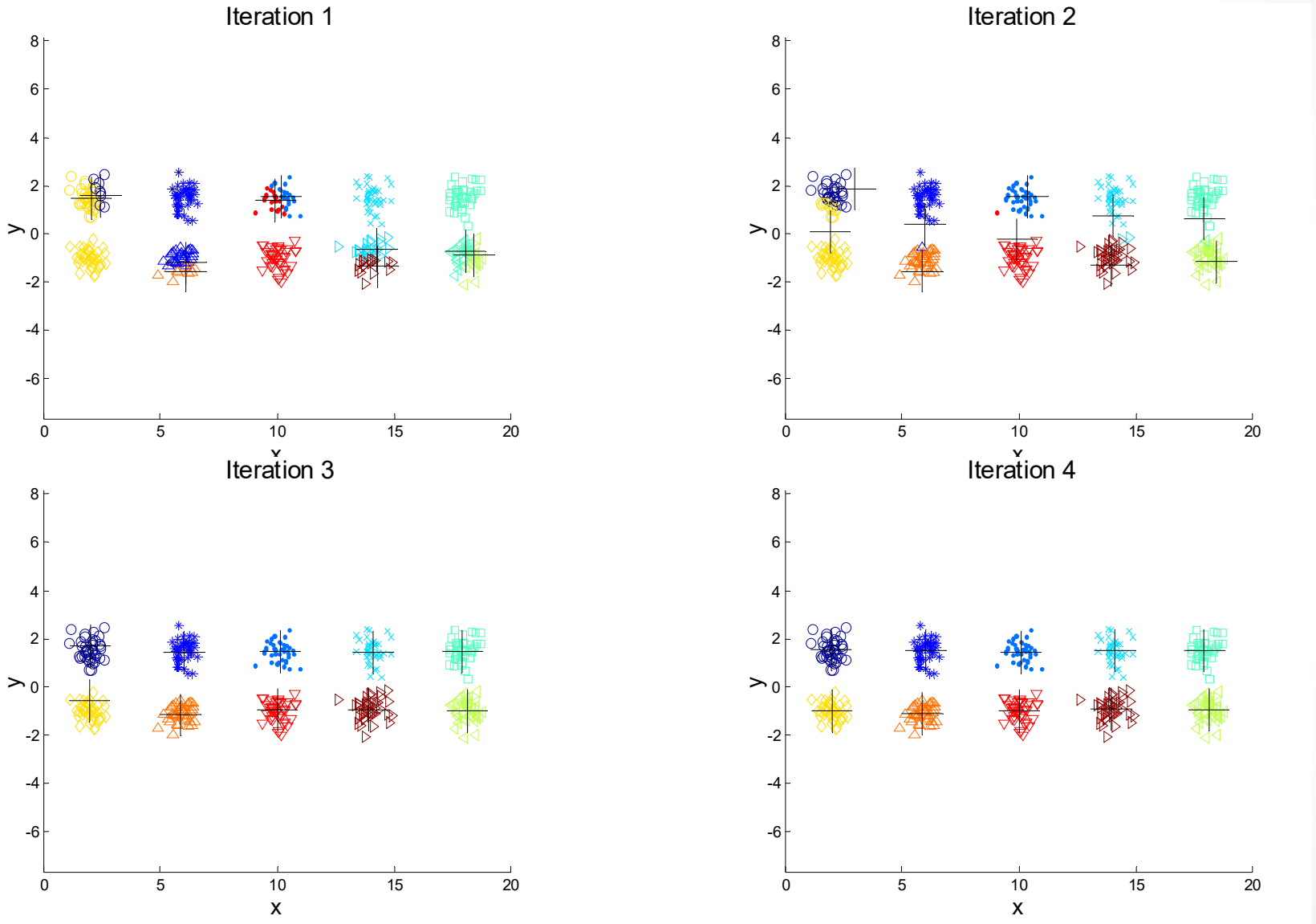
  - For example, if K = 10, then probability = $10!/10^{10} = 0.00036$
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

10 Clusters Example



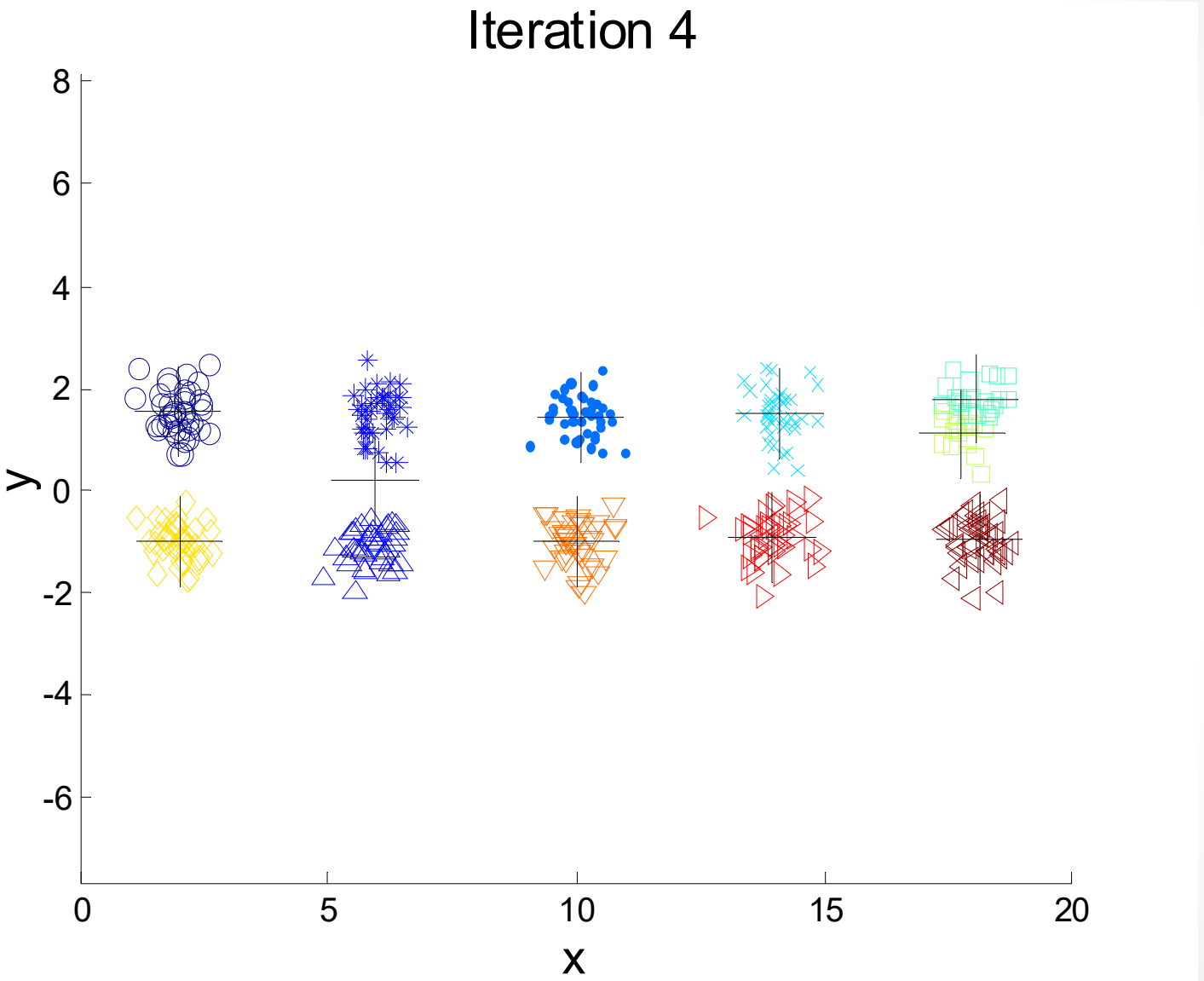
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



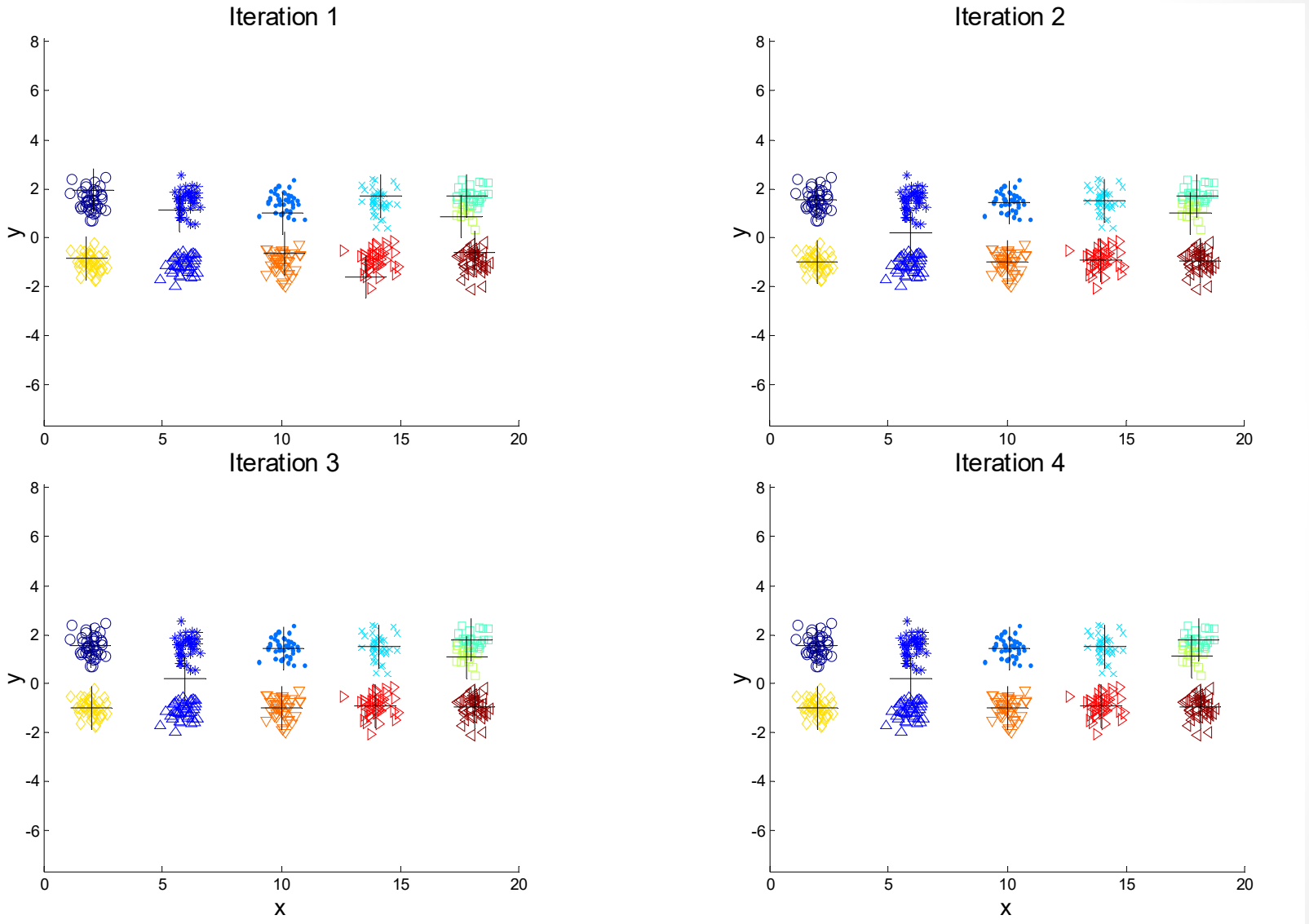
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



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

10 Clusters Example



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

Evaluating K-means Clusters

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

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

  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated

Pre-processing and Post-processing

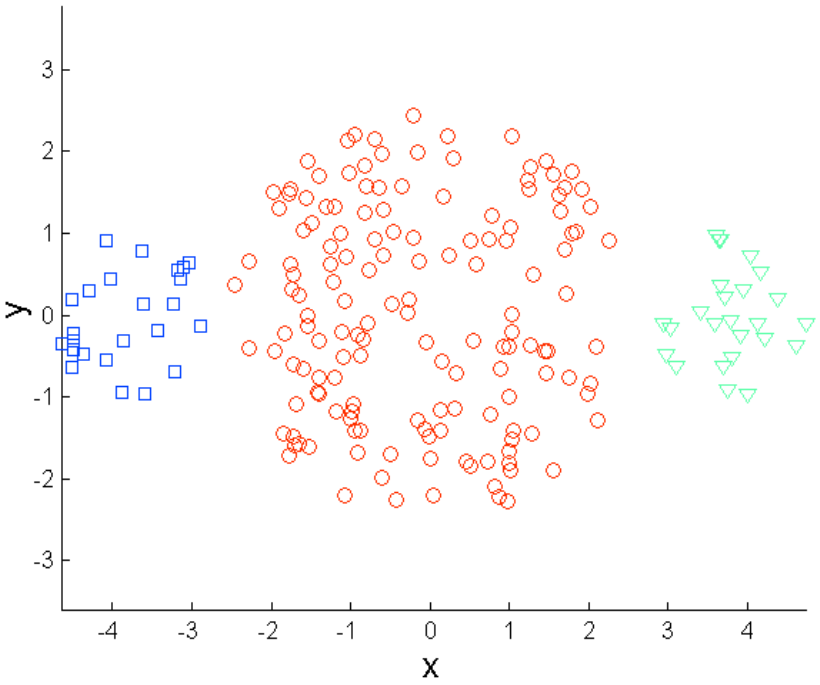
- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE
  - Can use these steps during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

Limitations of K-means

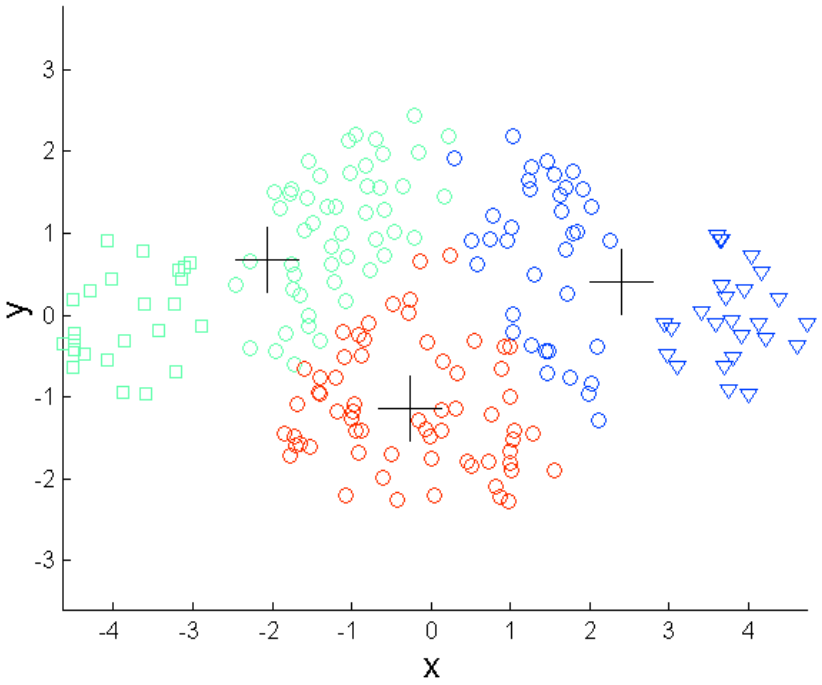
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

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

Limitations of K-means: Differing Sizes

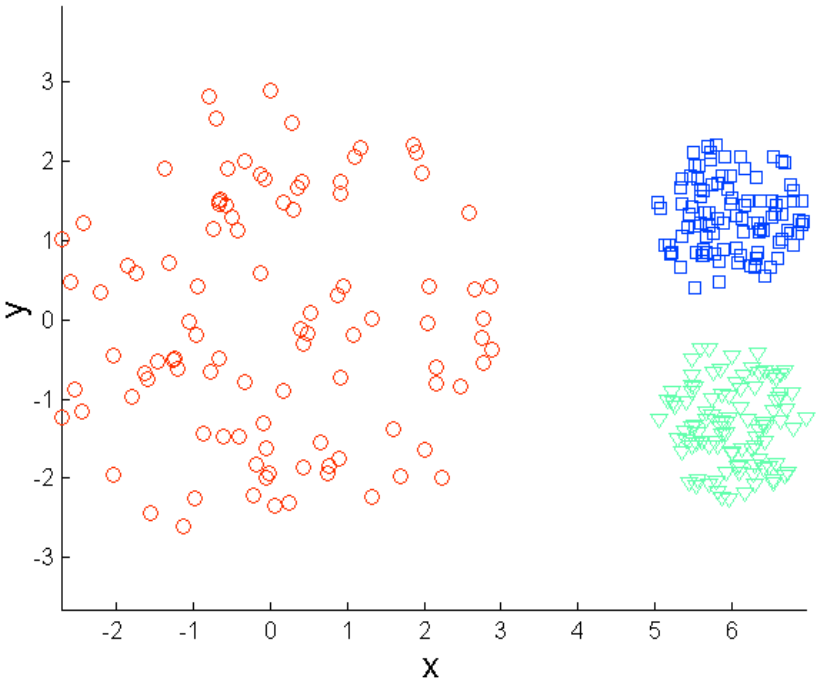


Original Points

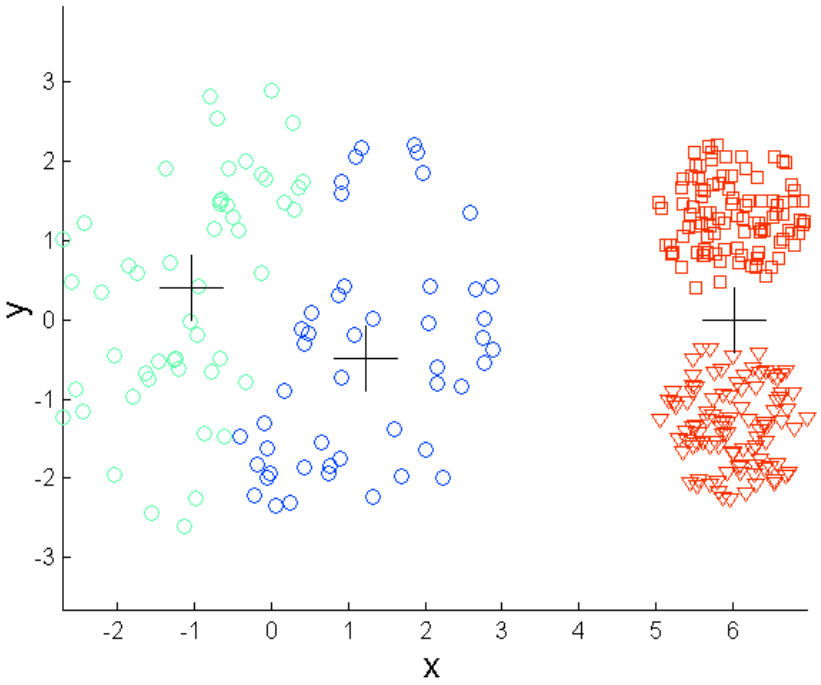


K-means (3 Clusters)

Limitations of K-means: Differing Density

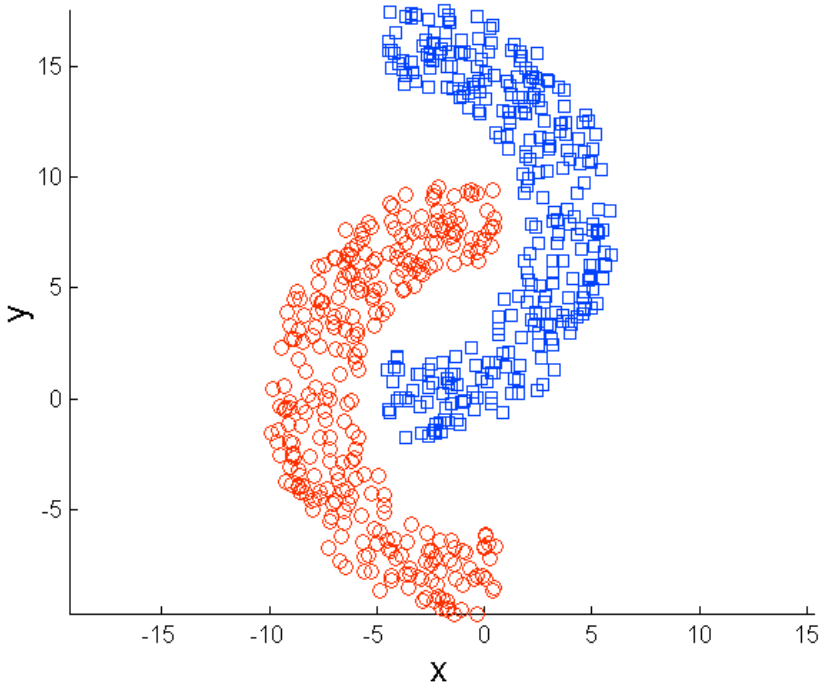


Original Points

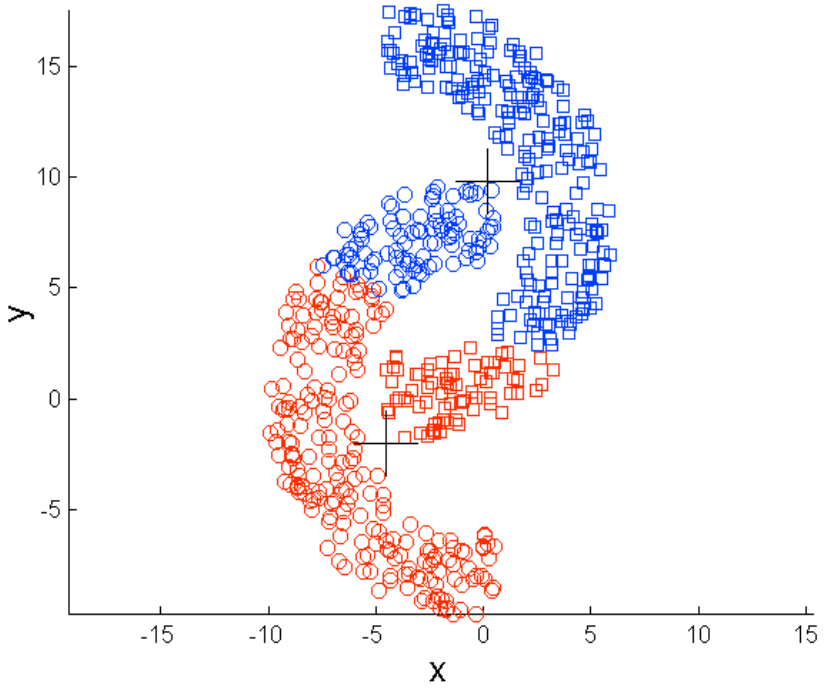


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes



Original Points

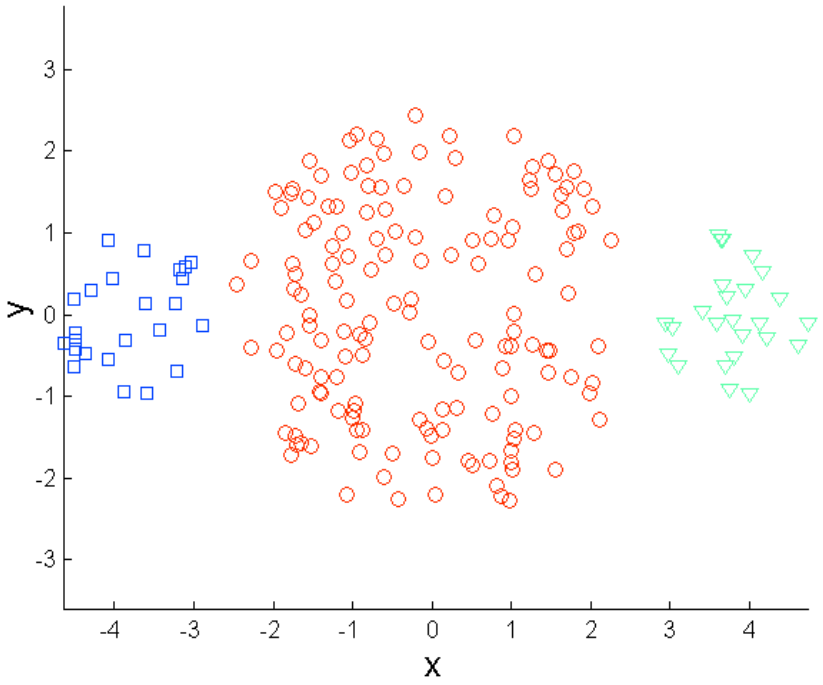


K-means (2 Clusters)

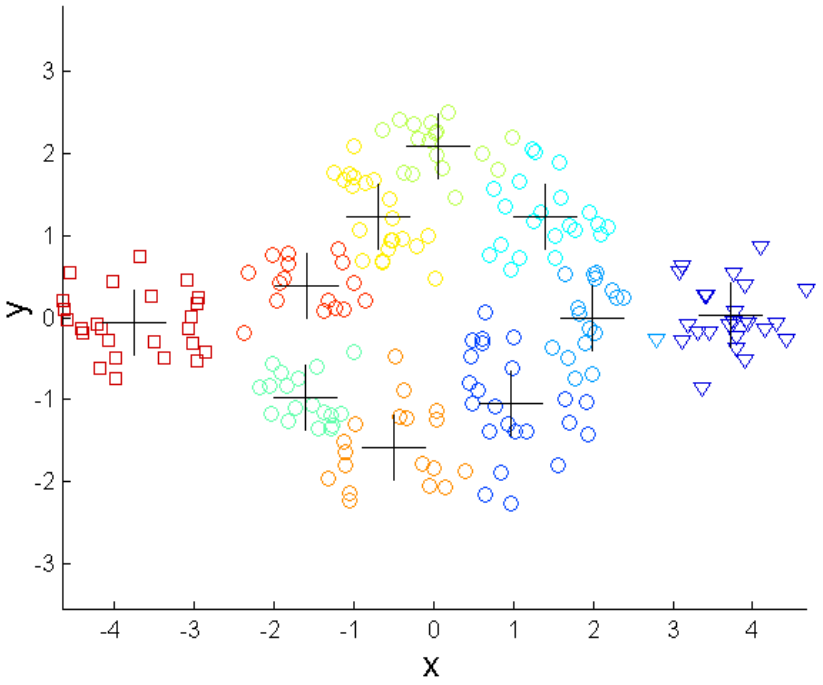
Overcoming K-means Limitations

One solution is to use many clusters.

Find parts of clusters, but need to put together (postprocessing).

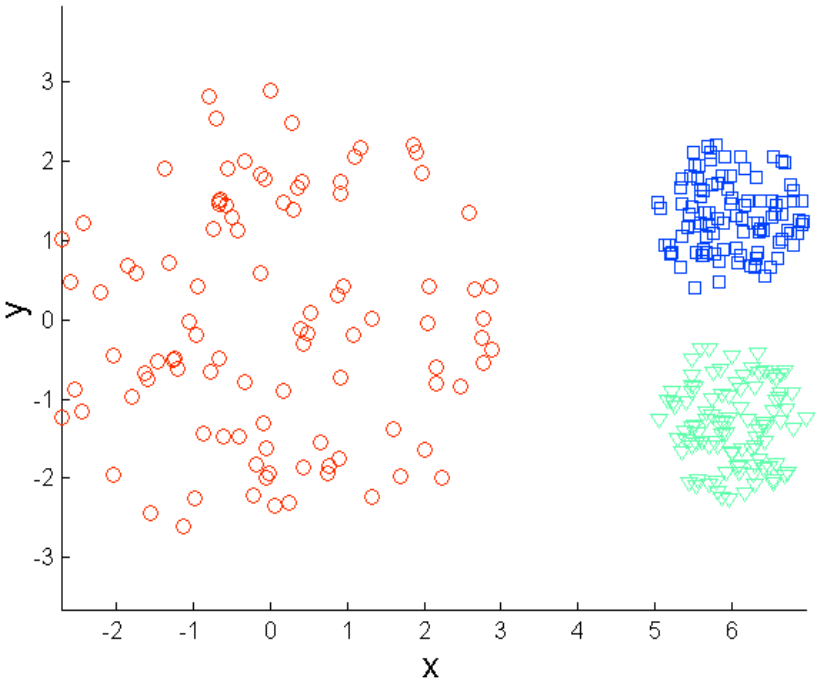


Original Points

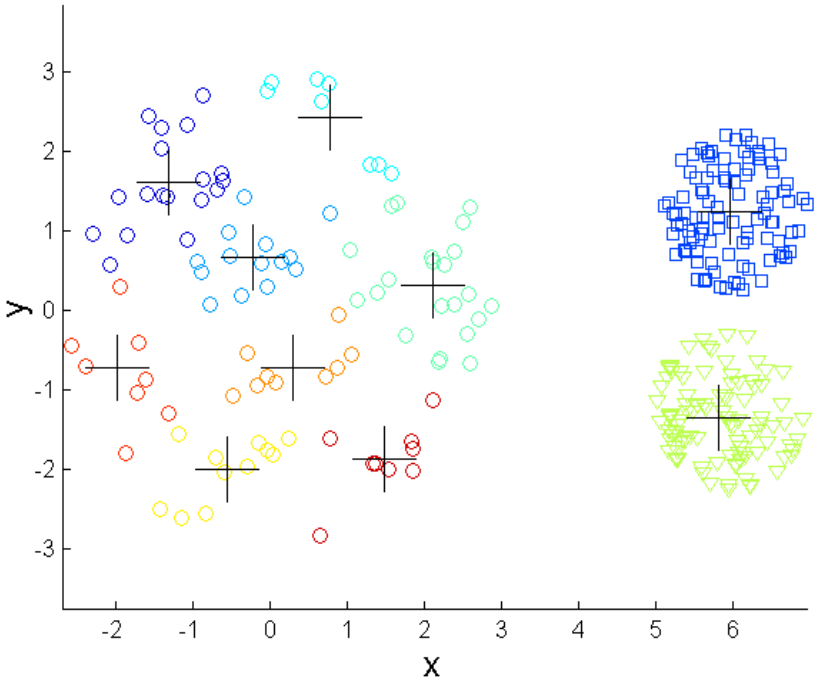


K-means Clusters

Overcoming K-means Limitations

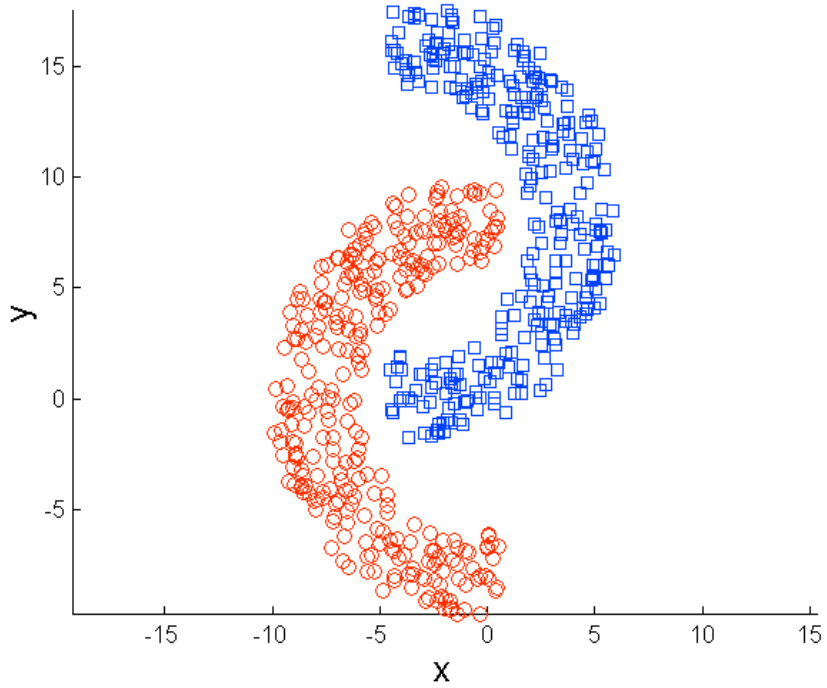


Original Points

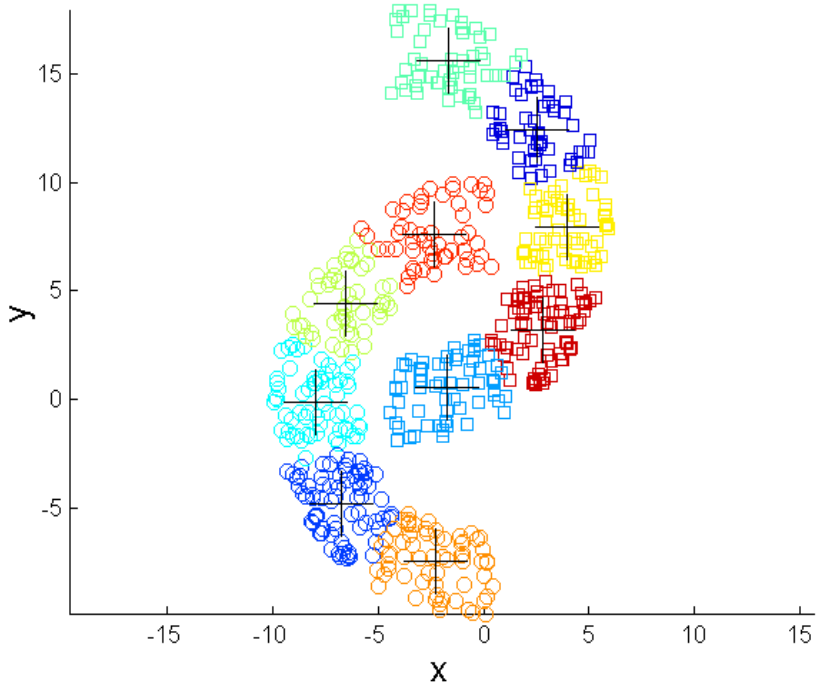


K-means Clusters

Overcoming K-means Limitations



Original Points

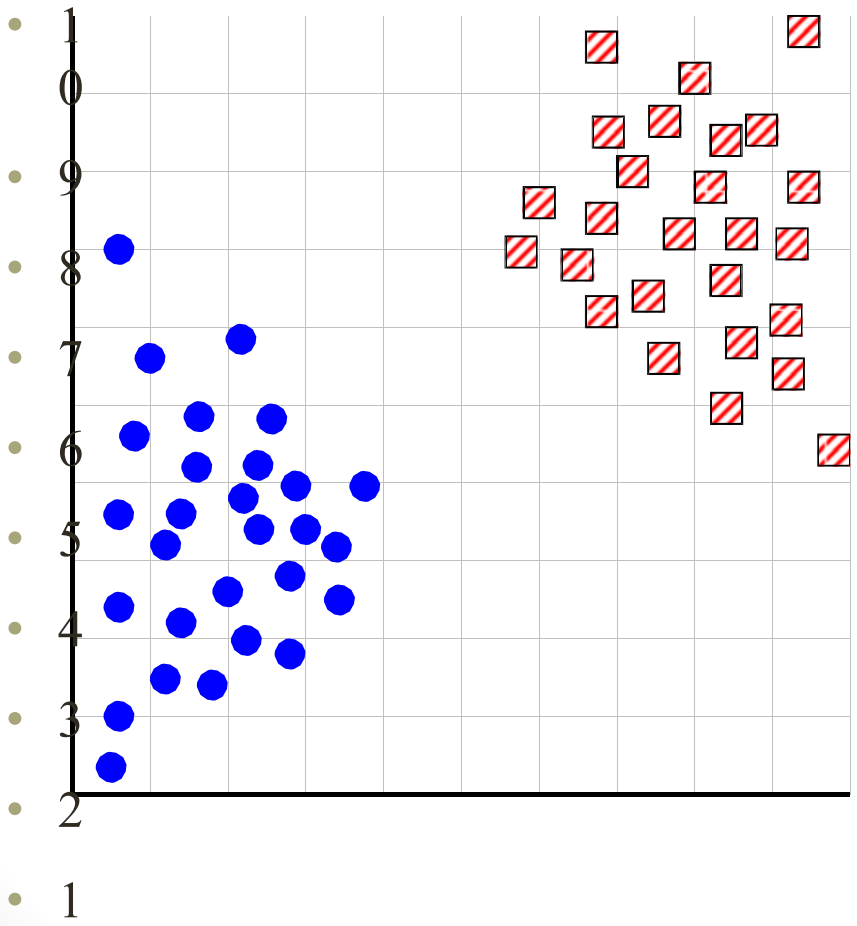


K-means Clusters

K-means: Determining the Number of Clusters

- In general, this is an unsolved problem.
- Mostly heuristic and domain dependent approaches.
- Plot the error for K=2, 3, … clusters, and find the knee in the curve.
- Use domain specific knowledge and inspect clusters for desired characteristics.

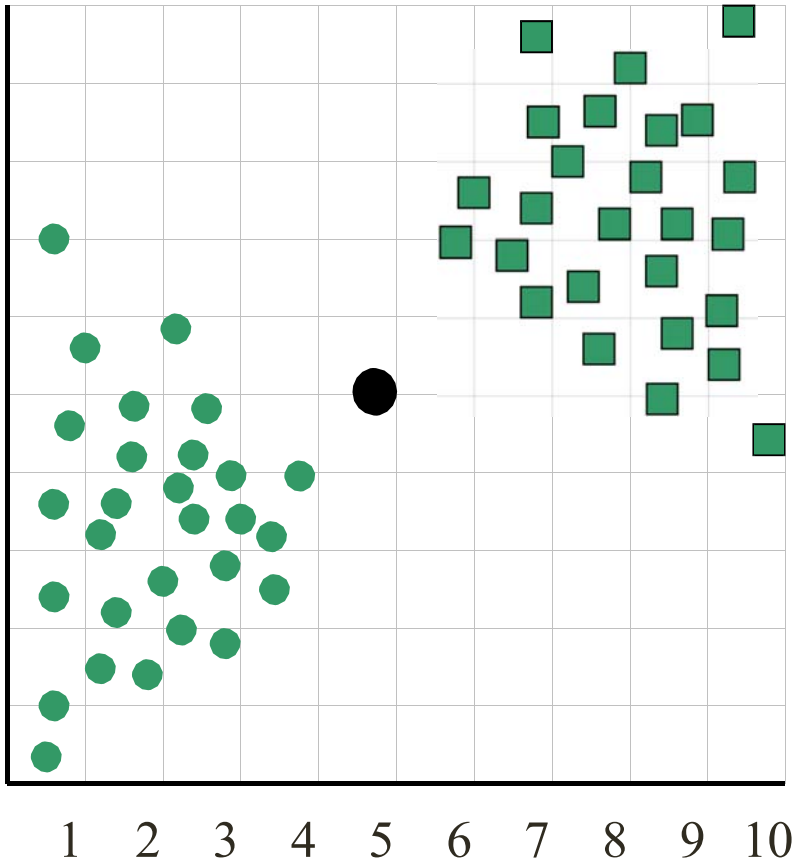
How can we tell the *right* number of clusters?



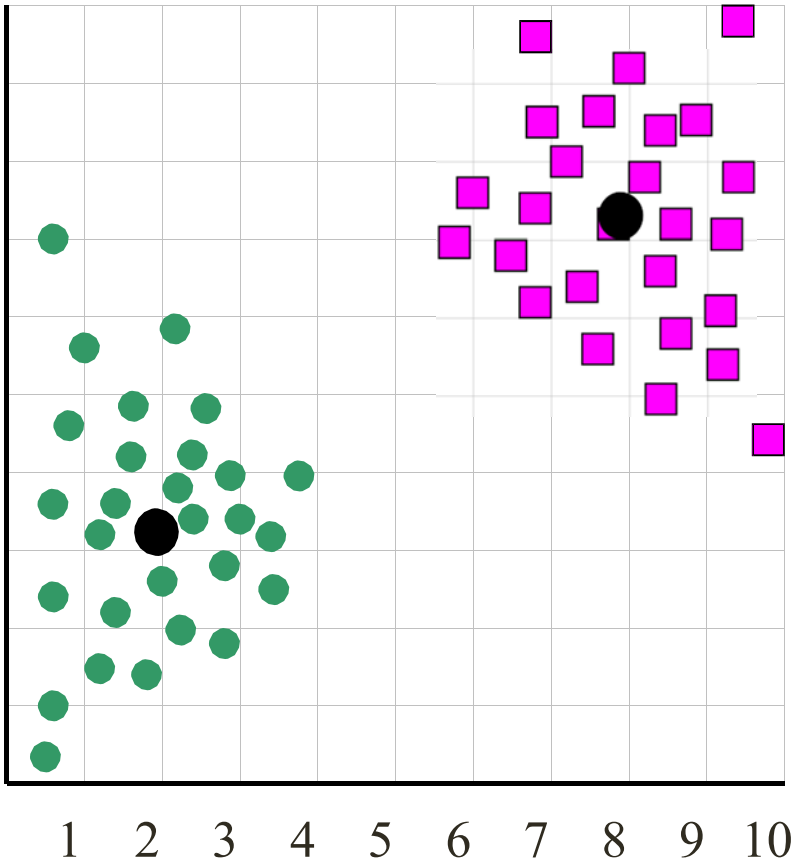
For our example, we will use the familiar katydid/grasshopper dataset.

We are imagining that we do NOT know the class labels. We are only clustering on the X and Y axis values.

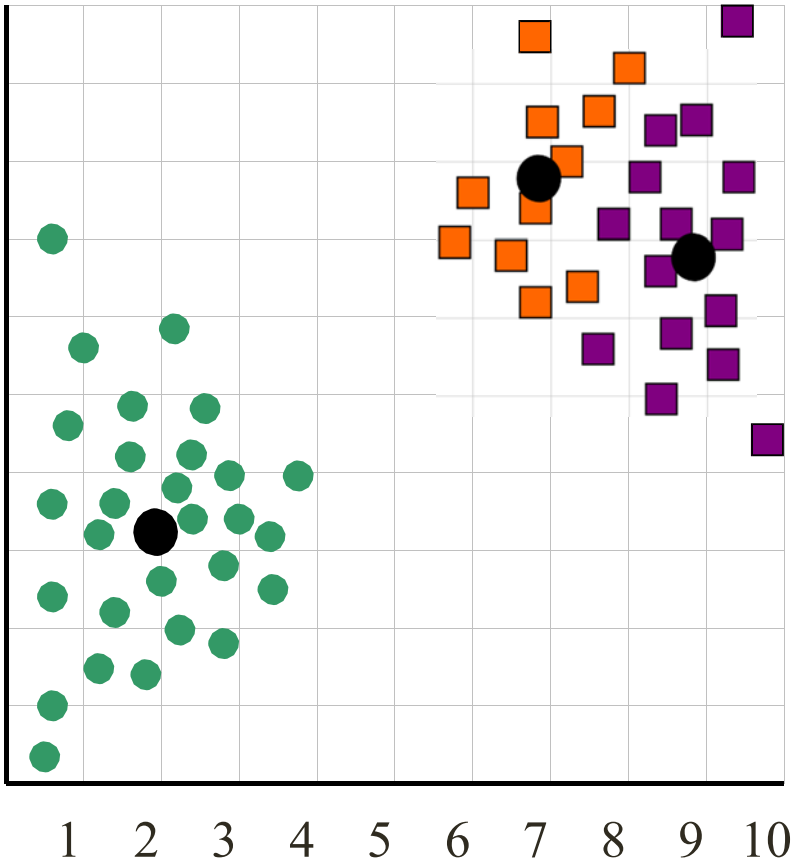
When k = 1, the objective function is 873.0 (MSE)



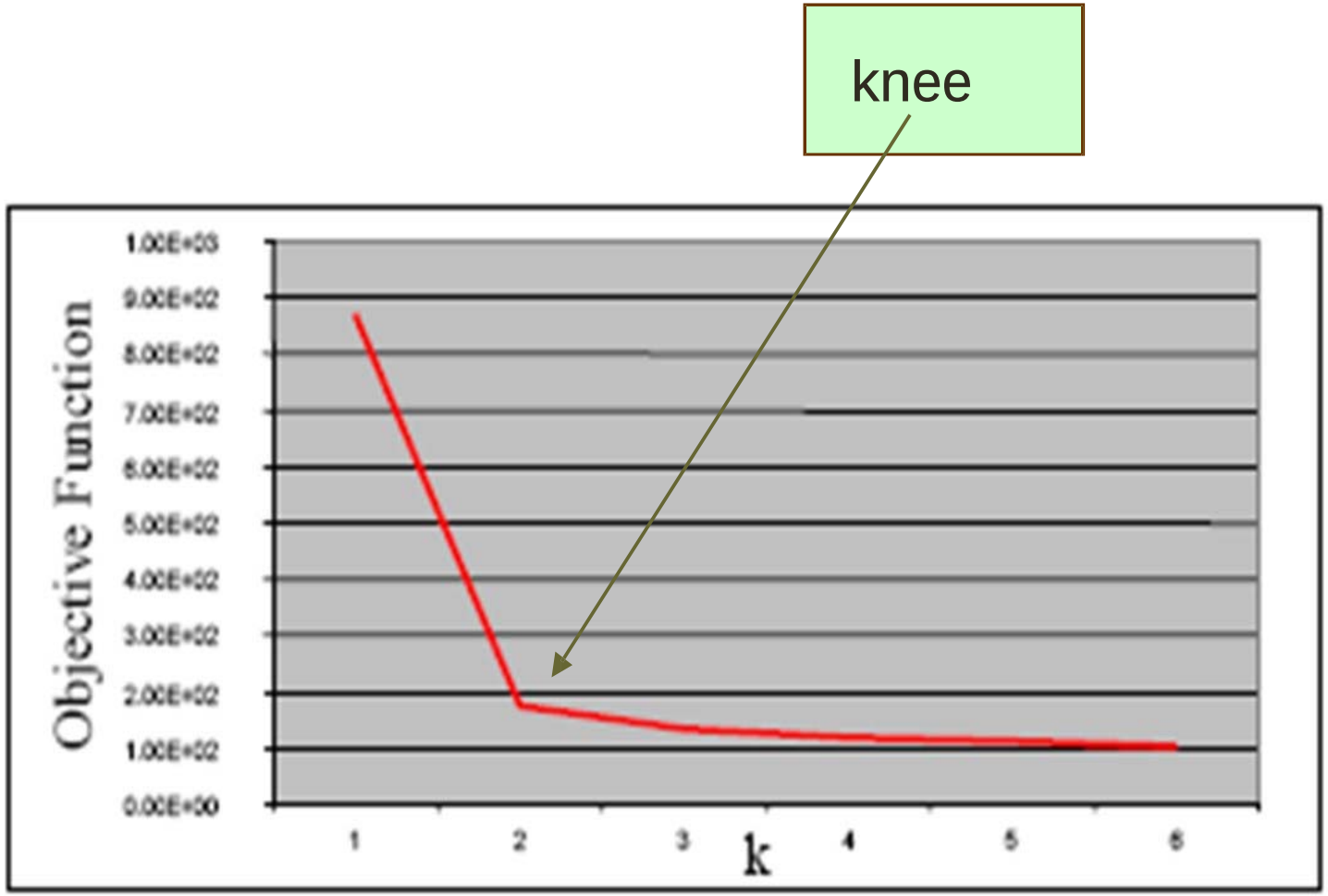
When k = 2, the objective function is 173.1 (MSE)



When k = 3, the objective function is 133.6 (MSE)



When k = 2, the objective function is 173.1

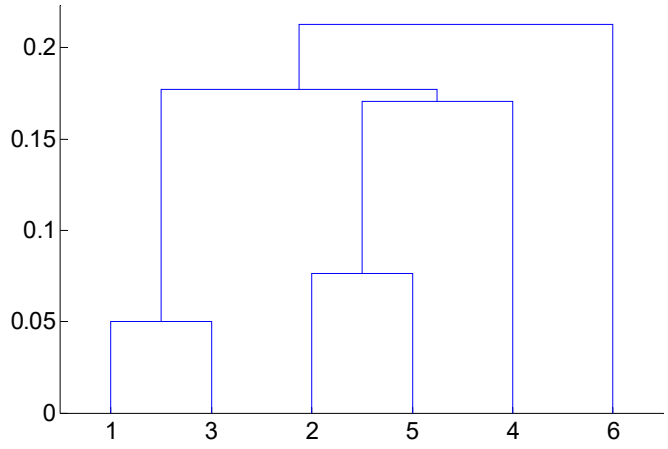


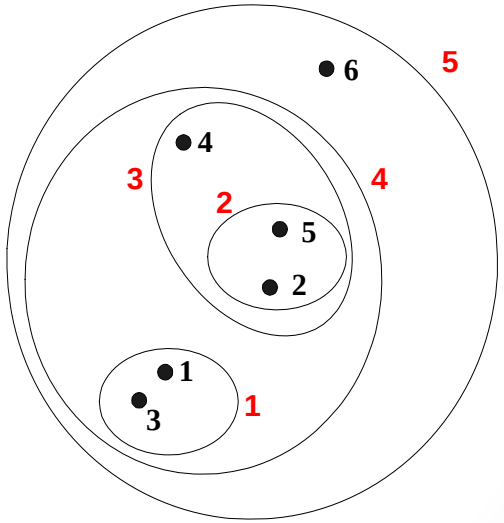
Index

- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - Partitioning algorithms: K-means
  - **Hierarchical clustering:**
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

Hierarchical Clustering

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





Strengths of Hierarchical Clustering

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

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

Hierarchical Clustering

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

Index

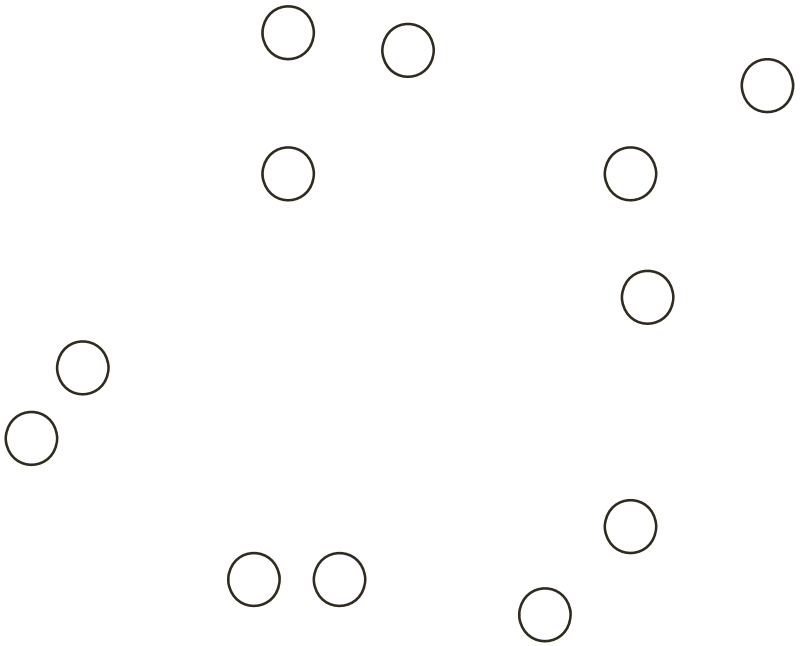
- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - **Aglomerative**
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

- Start with clusters of individual points and a proximity matrix

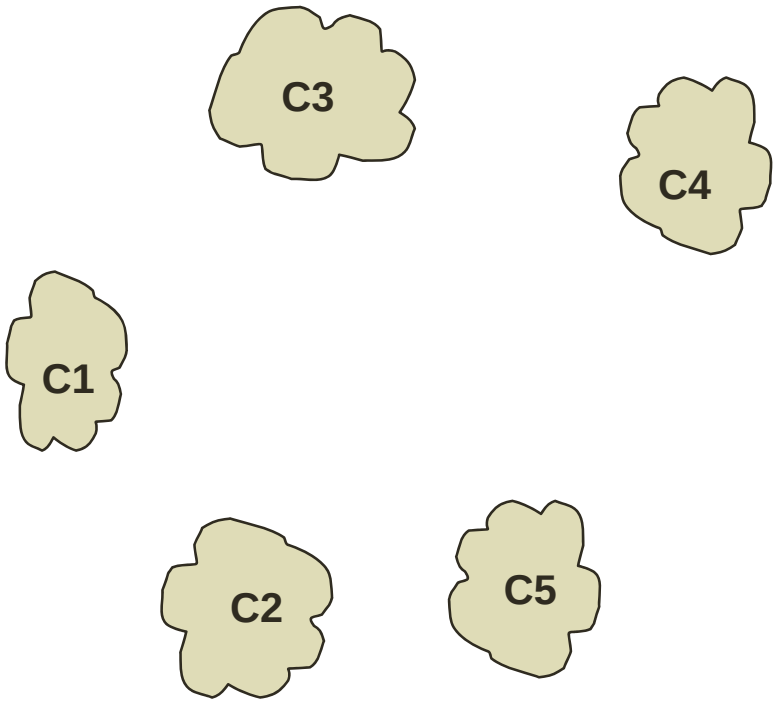


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

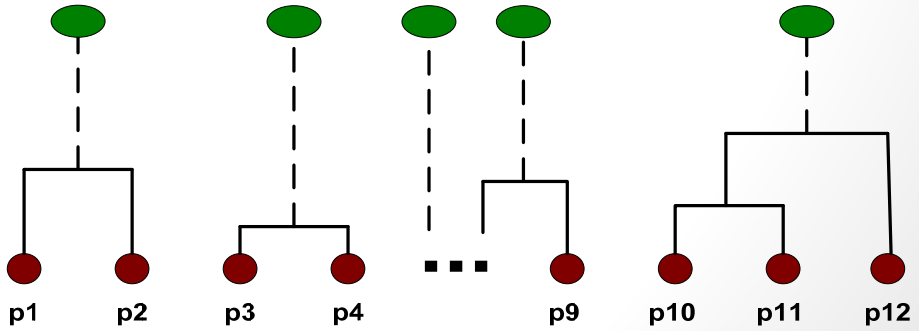
Intermediate Situation

- After some merging steps, we have some clusters



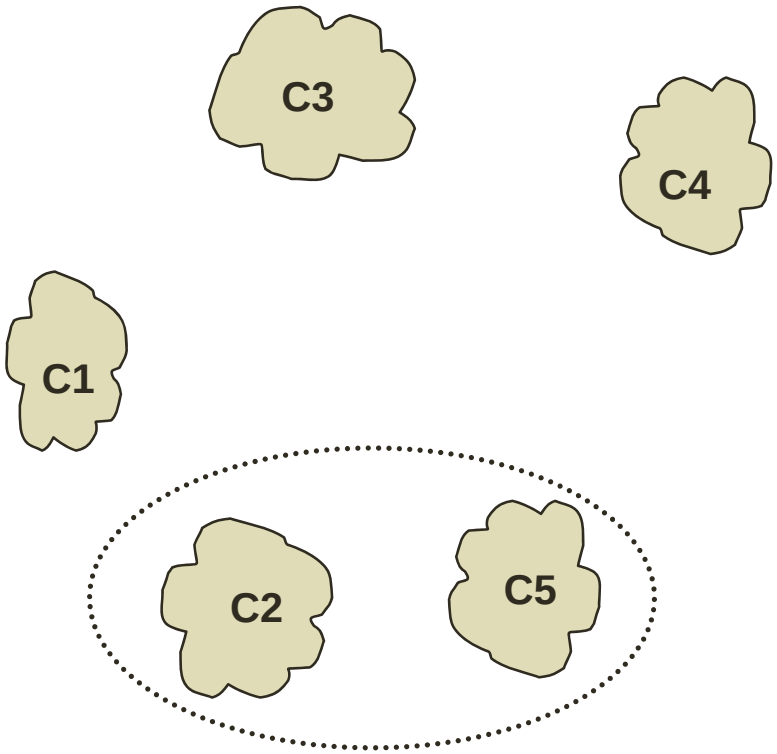
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



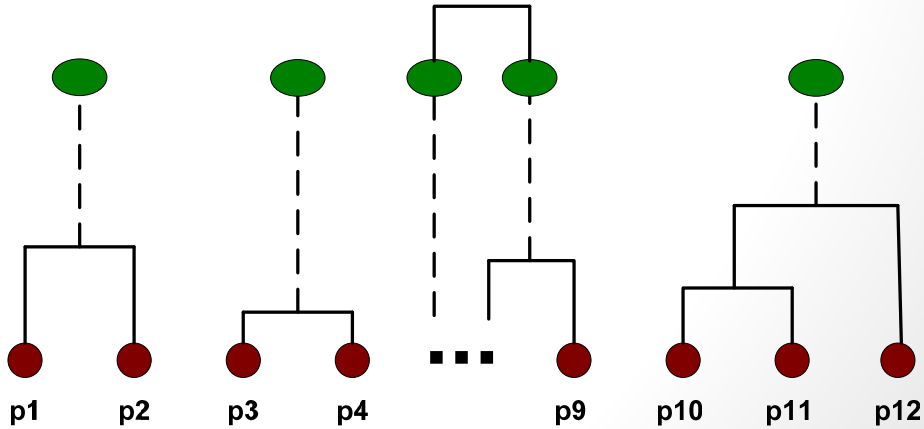
Intermediate Situation

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



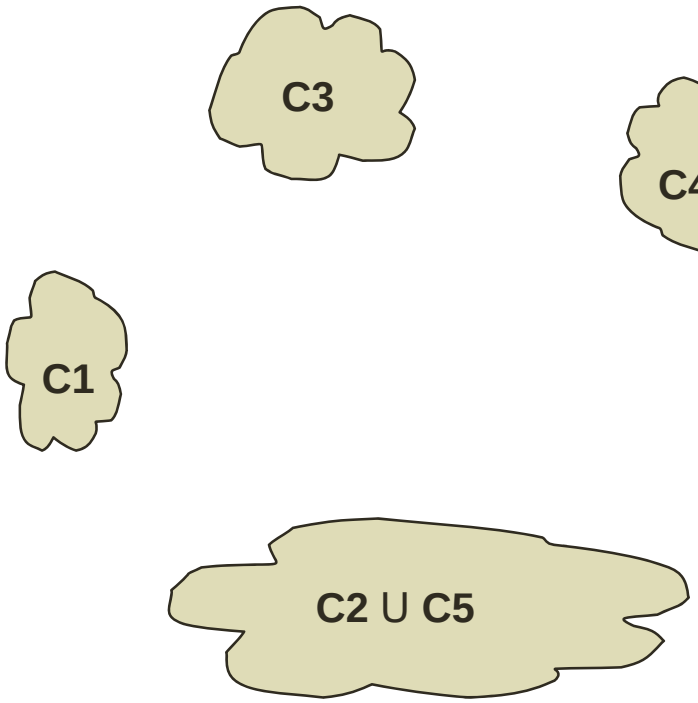
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



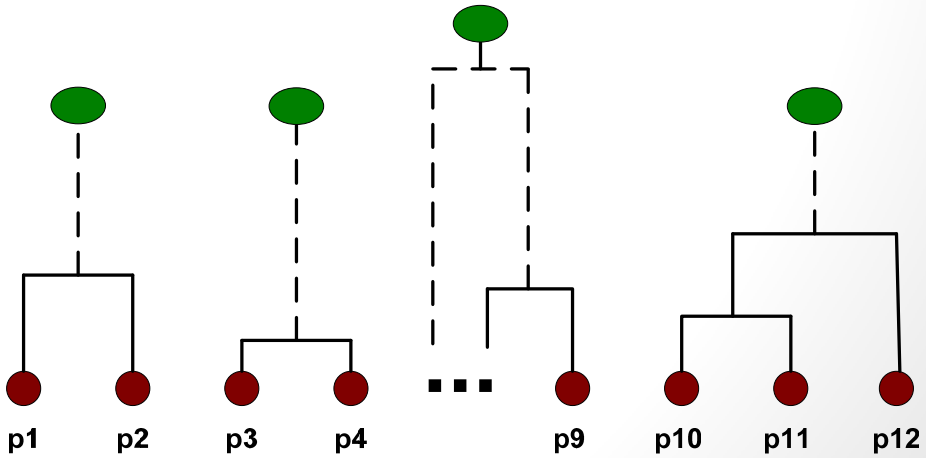
After Merging

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

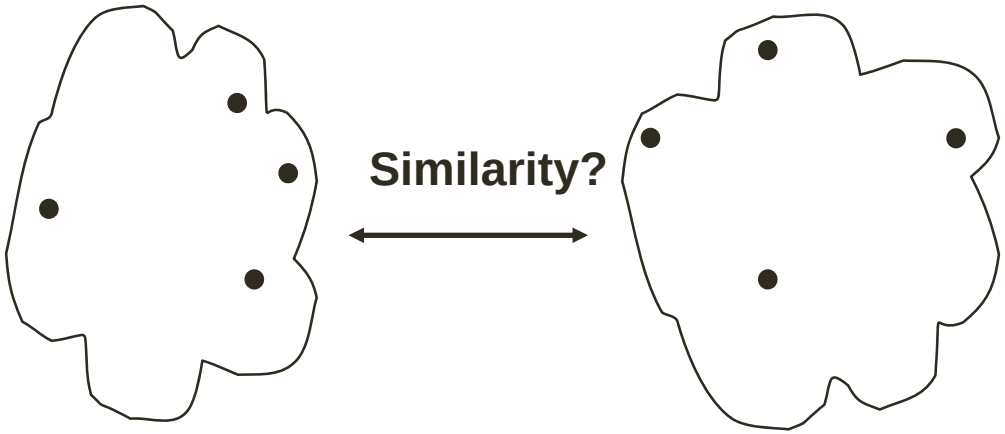


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

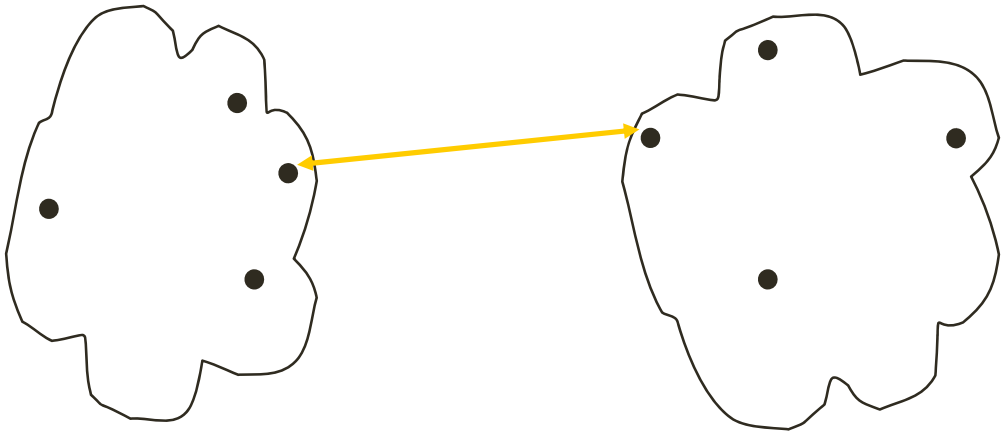


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

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

How to Define Inter-Cluster Similarity

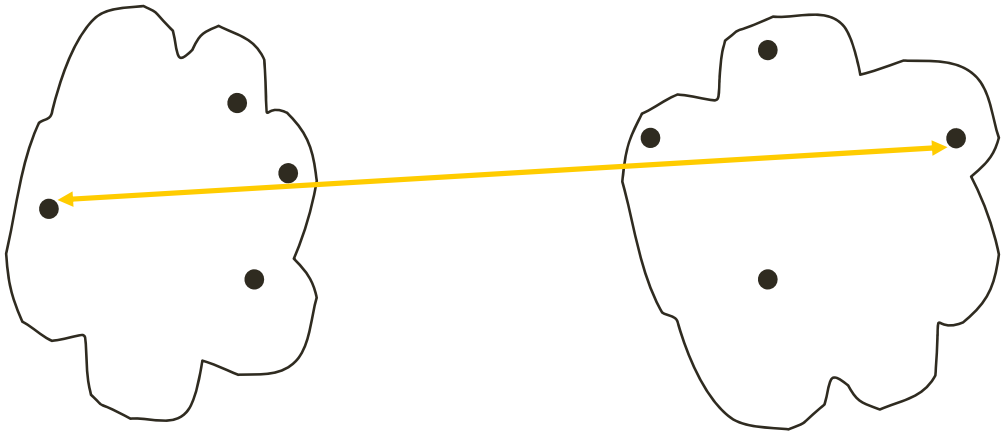


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

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

How to Define Inter-Cluster Similarity

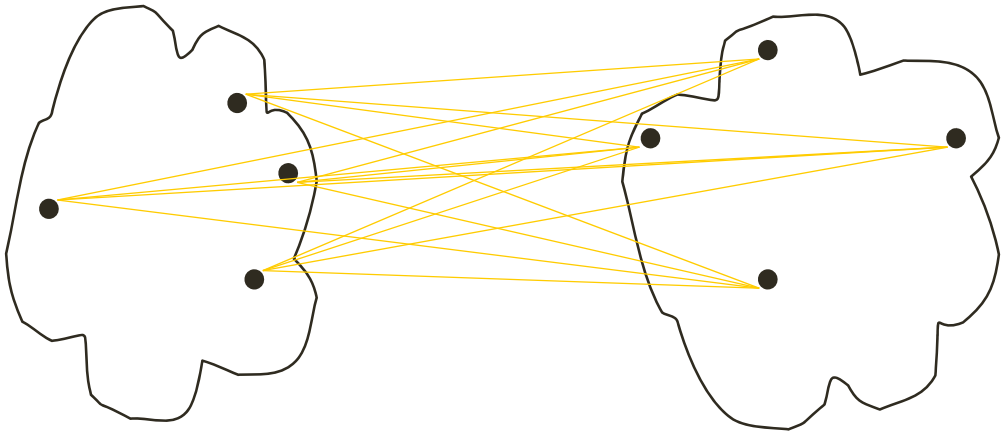


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

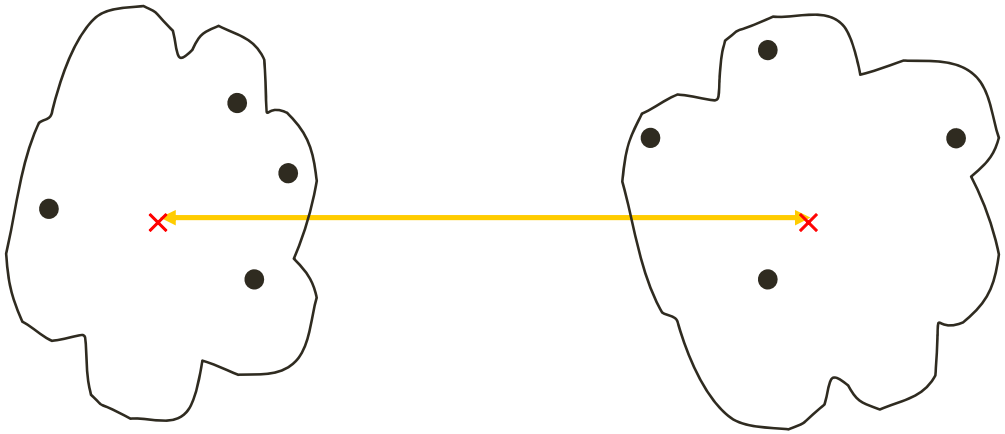


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

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

How to Define Inter-Cluster Similarity



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

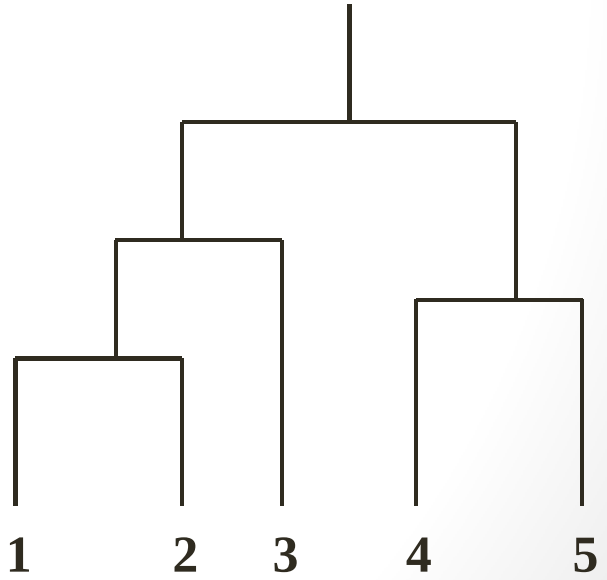
Proximity Matrix

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

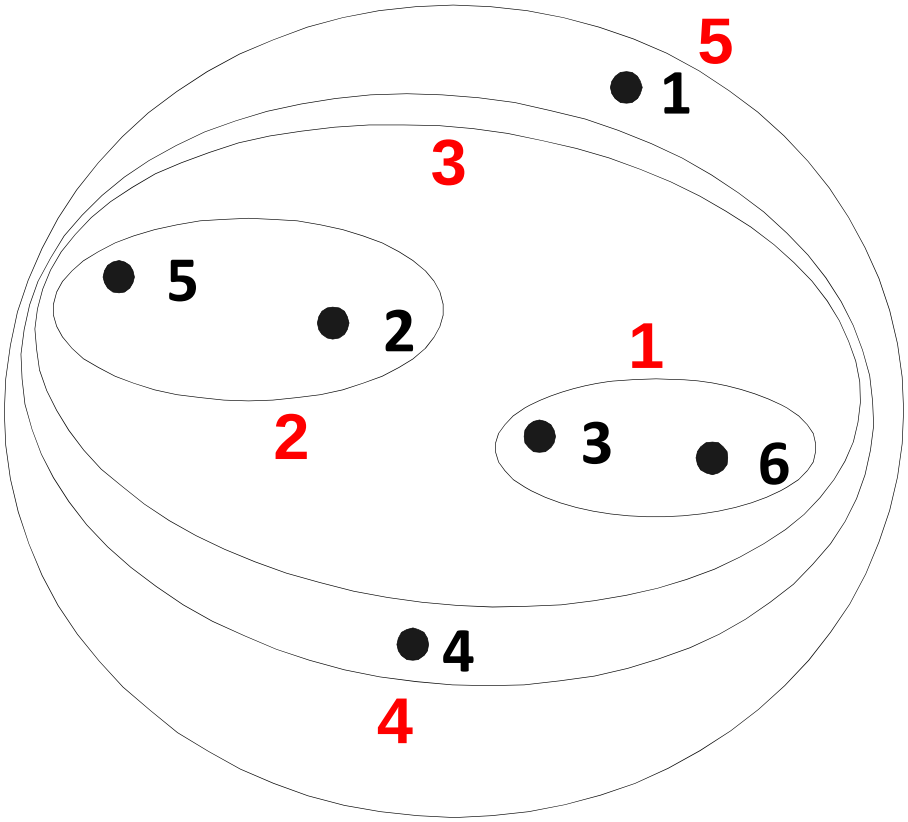
Cluster Similarity: MIN or Single Link

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

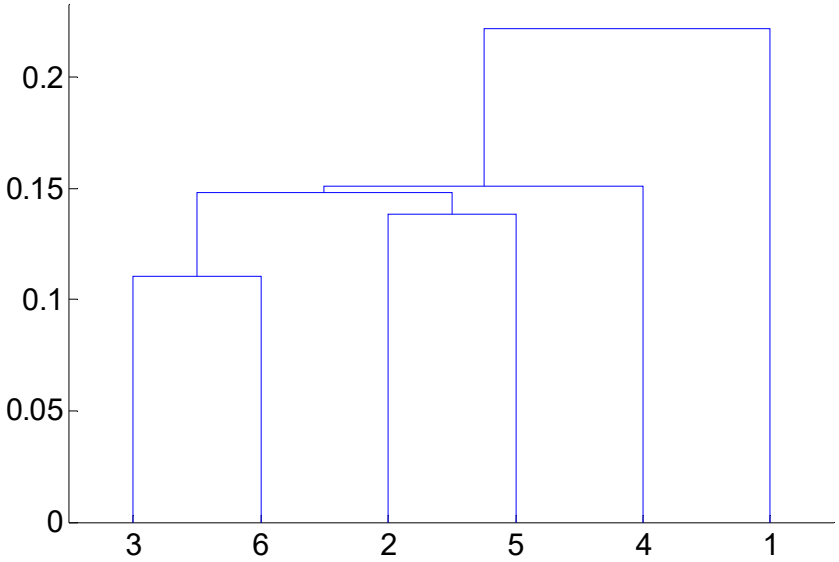
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN

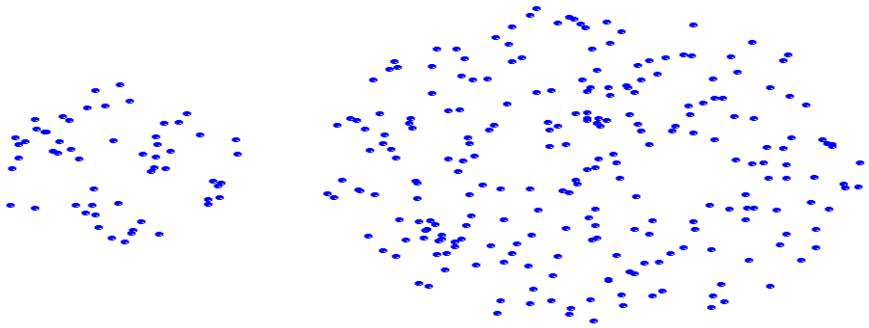


Nested Clusters

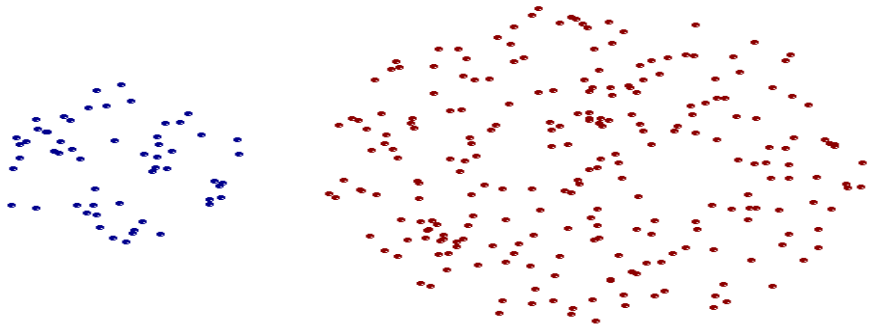


Dendrogram

Strength of MIN



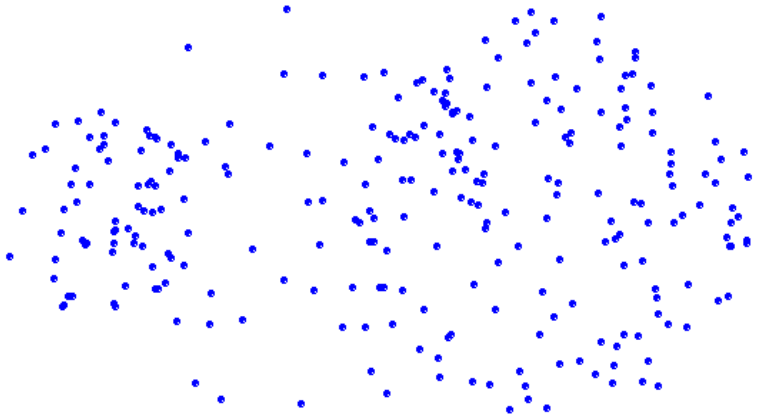
Original Points



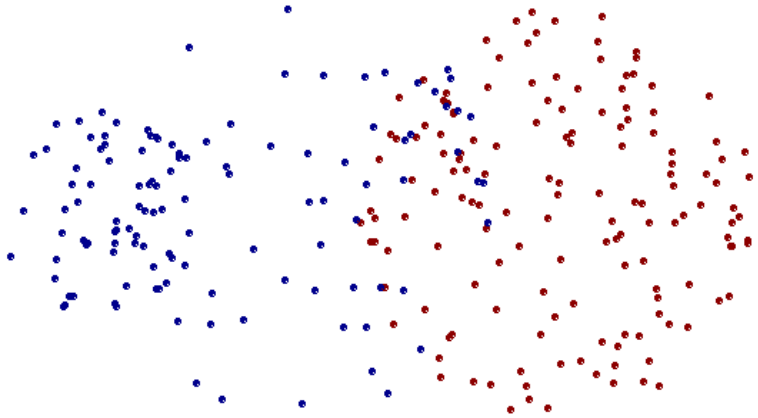
Two Clusters

- **Can handle non-elliptical shapes**

Limitations of MIN



Original Points



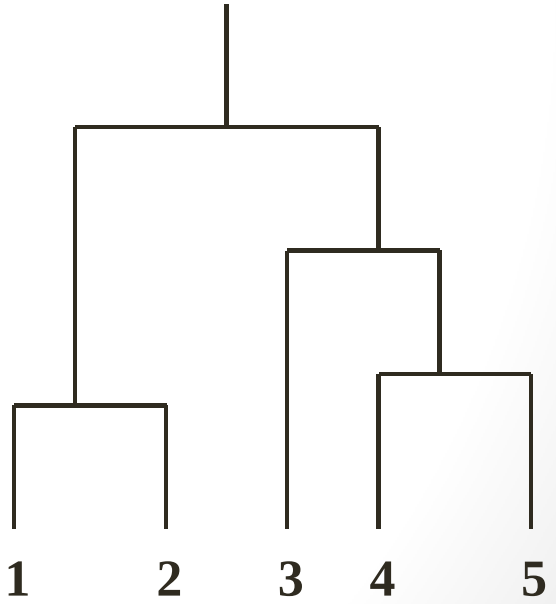
Two Clusters

- **Sensitive to noise and outliers**

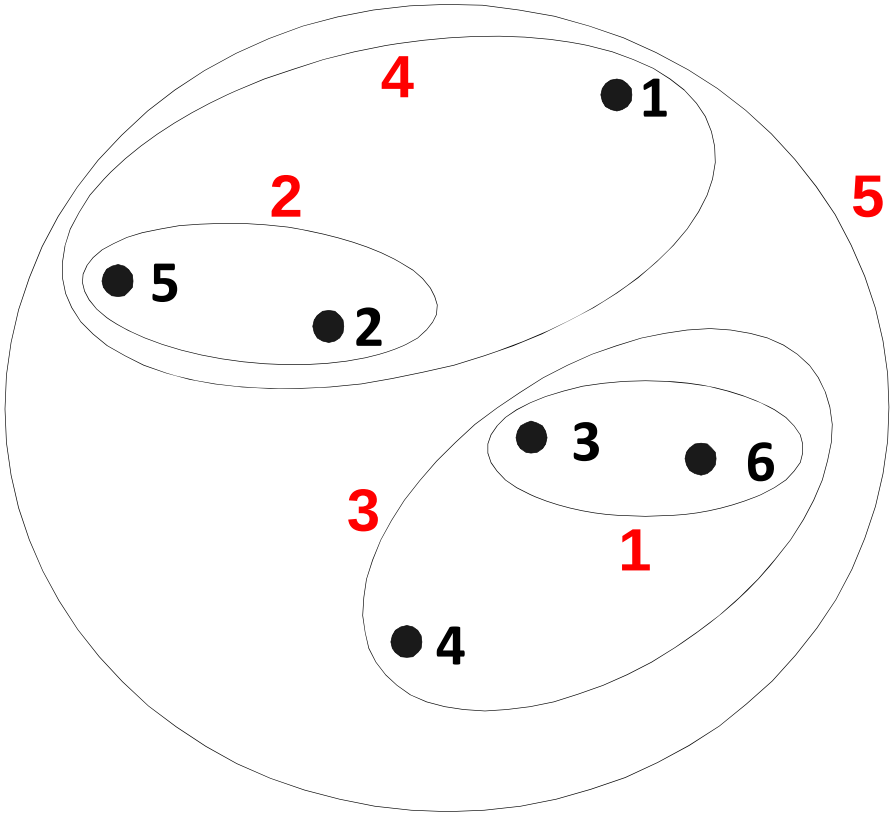
Cluster Similarity: MAX or Complete Linkage

- 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

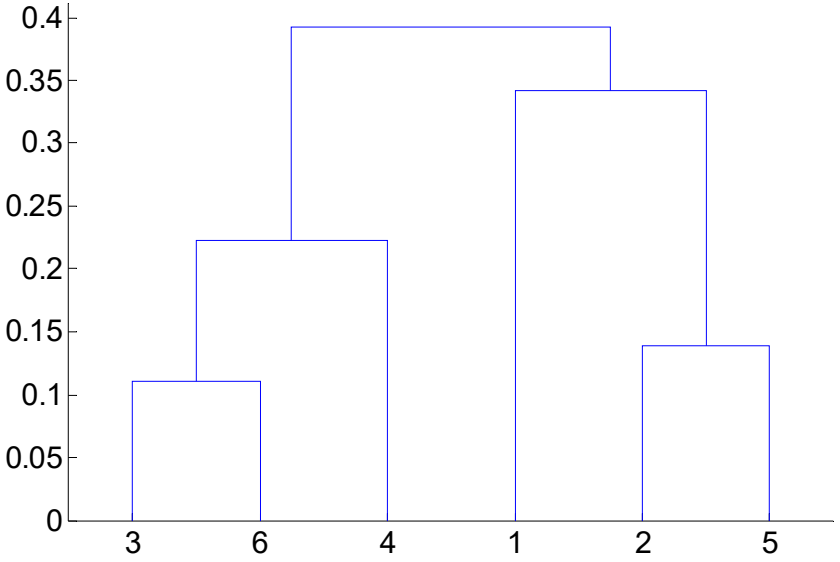
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

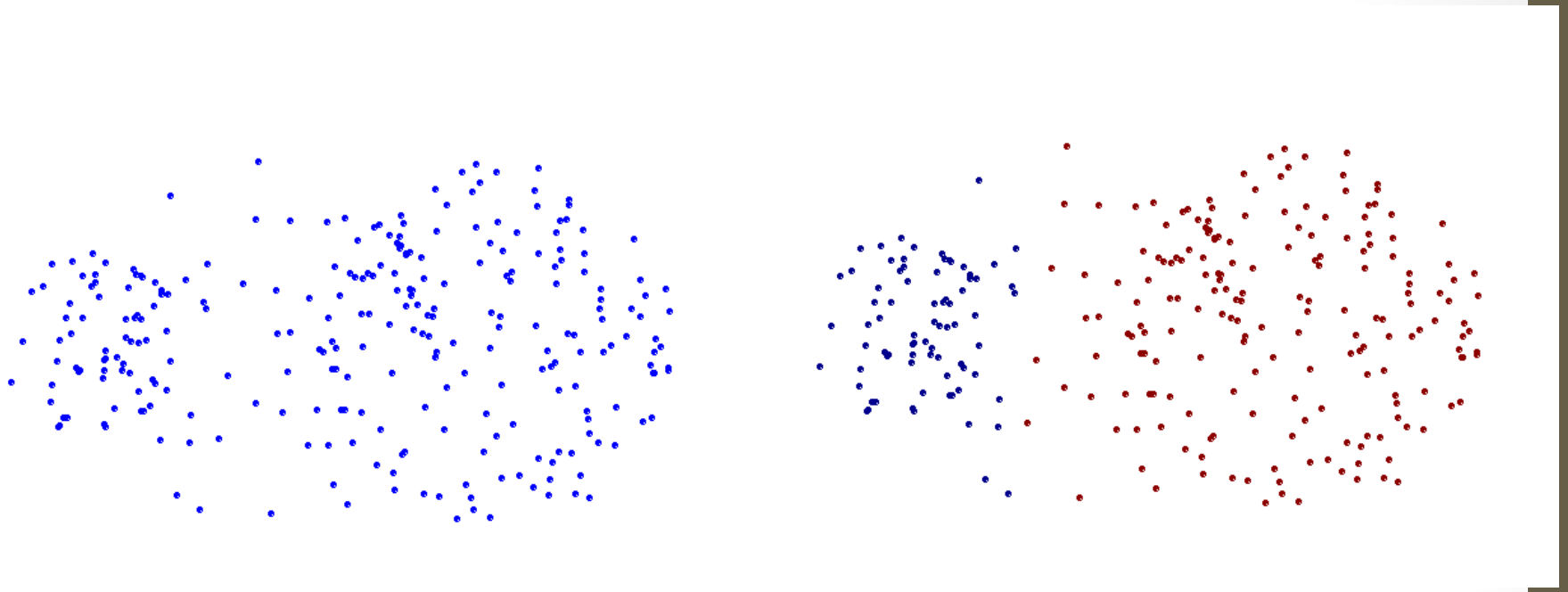


Nested Clusters



Dendrogram

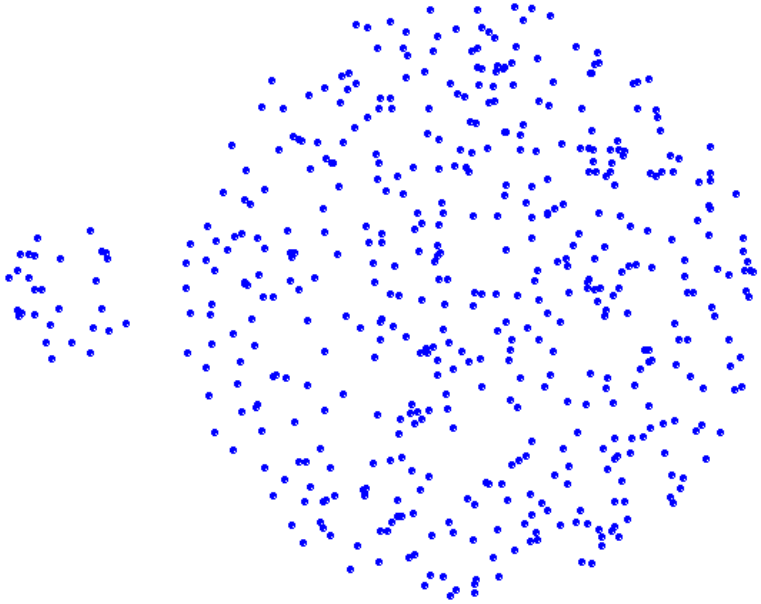
Strength of MAX



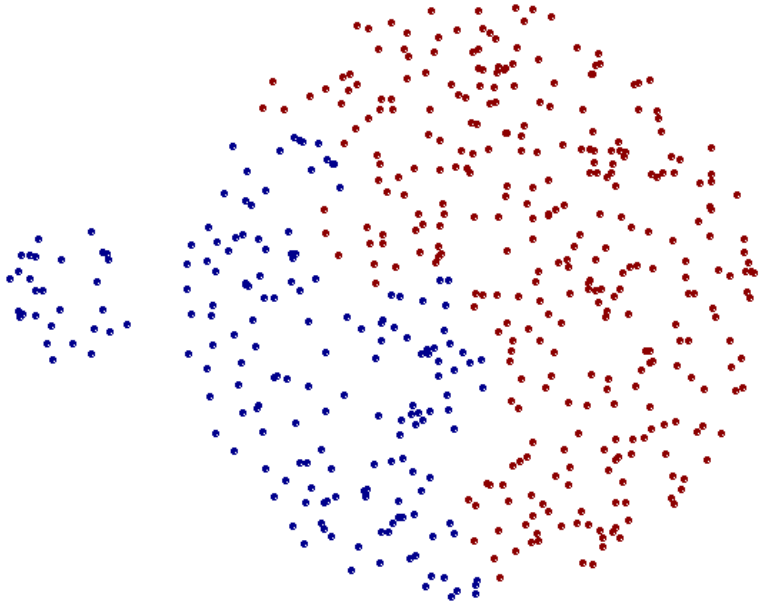
Original Points **Two Clusters**

- **Less susceptible to noise and outliers**

Limitations of MAX



Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

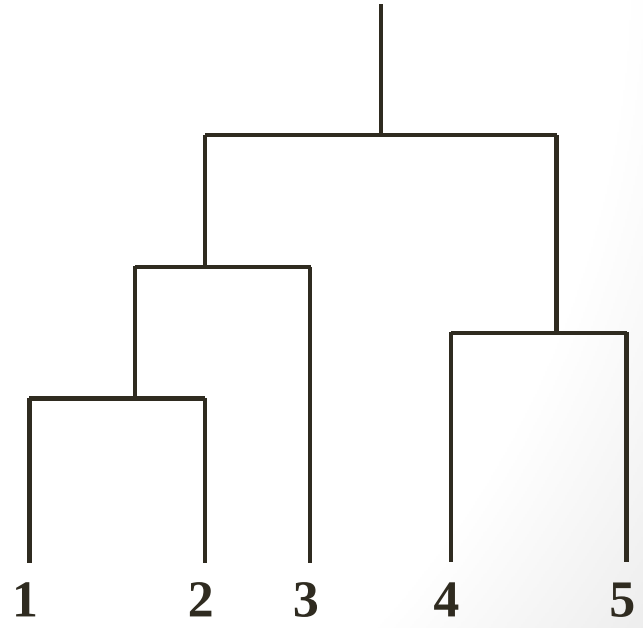
Cluster Similarity: Group Average

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

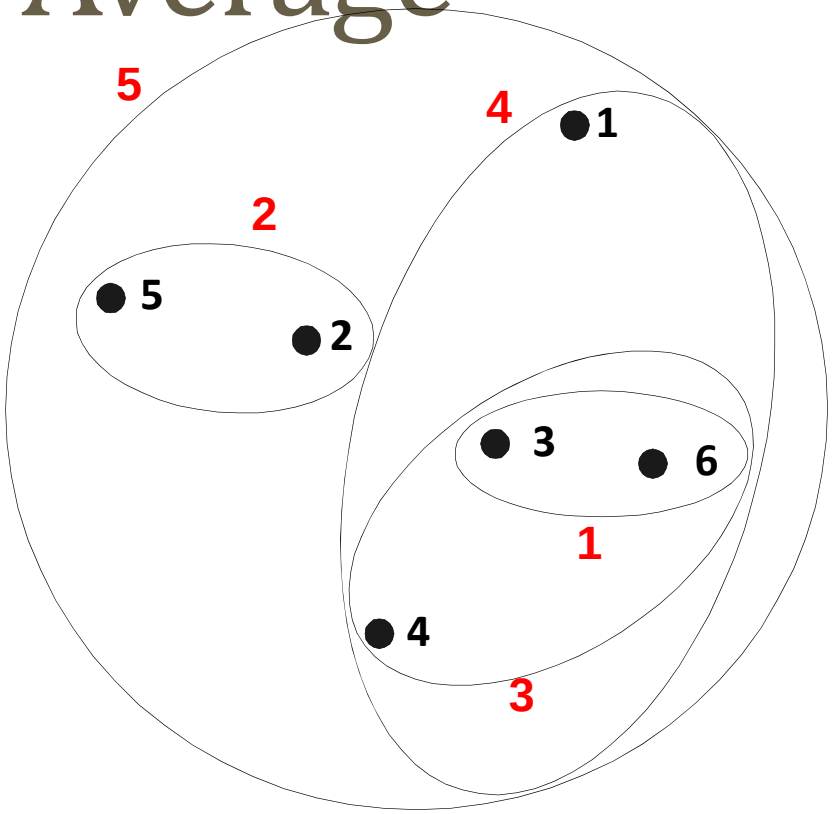
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

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

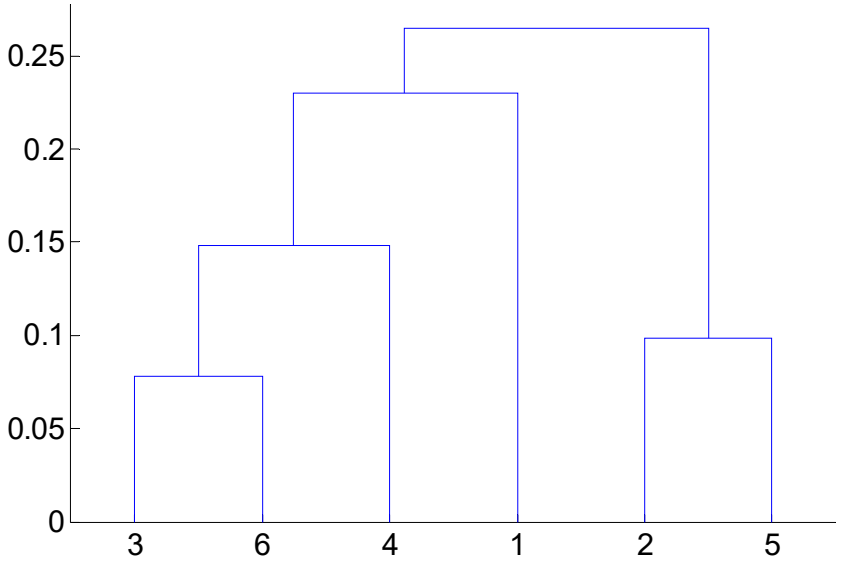
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

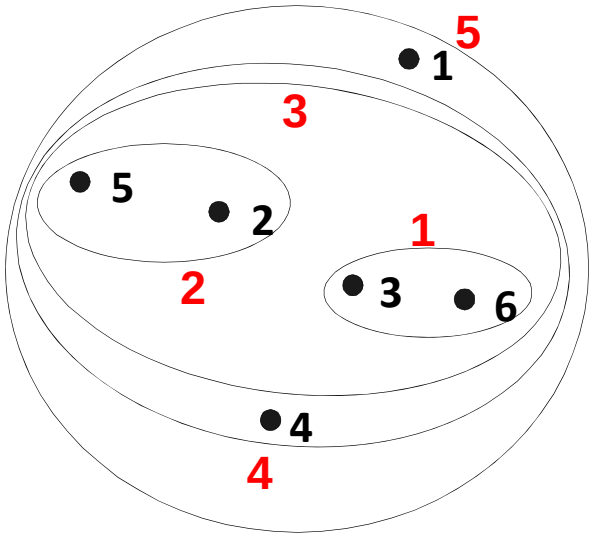
Hierarchical Clustering: Group Average

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

Cluster Similarity: Ward's Method

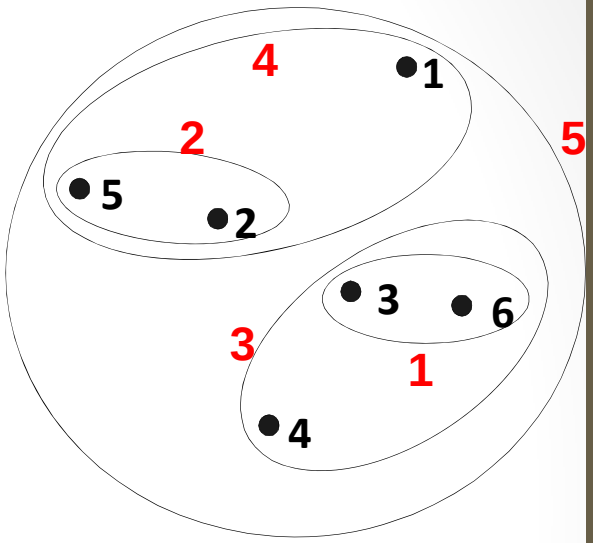
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used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

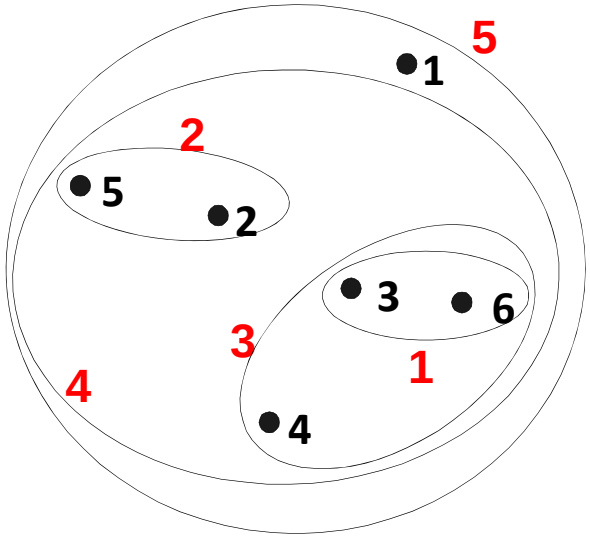
Hierarchical Clustering: Comparison



MIN

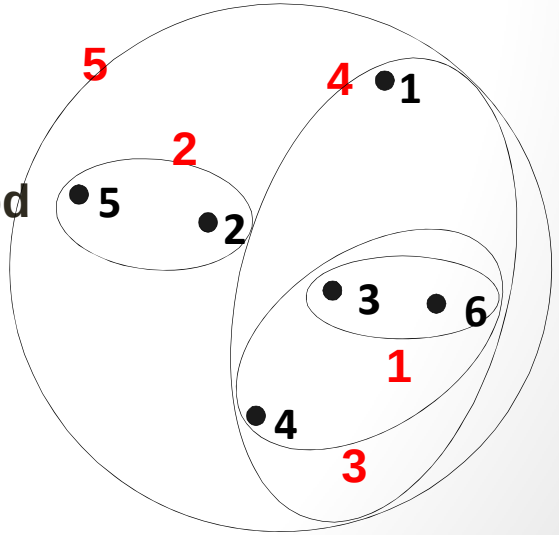
MAX





Group Average

Ward's Method



Hierarchical Clustering: Time and Space requirements

- $O(N^2)$ space since it uses the proximity matrix.
  - N is the number of points.

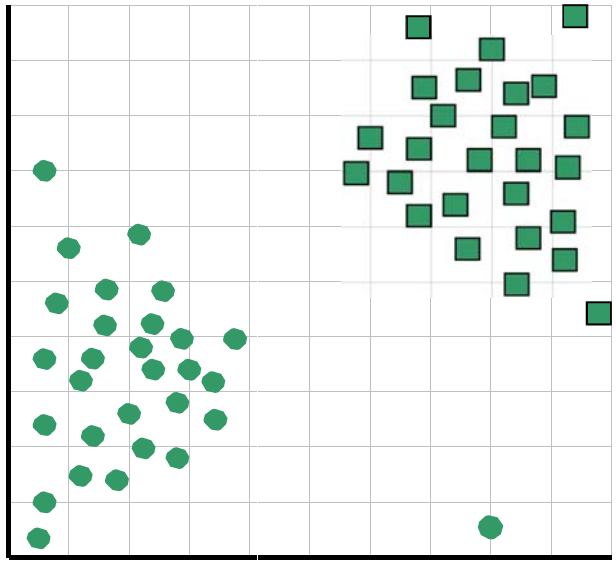
- $O(N^3)$ time in many cases
  - There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Hierarchical Clustering: Problems and Limitations

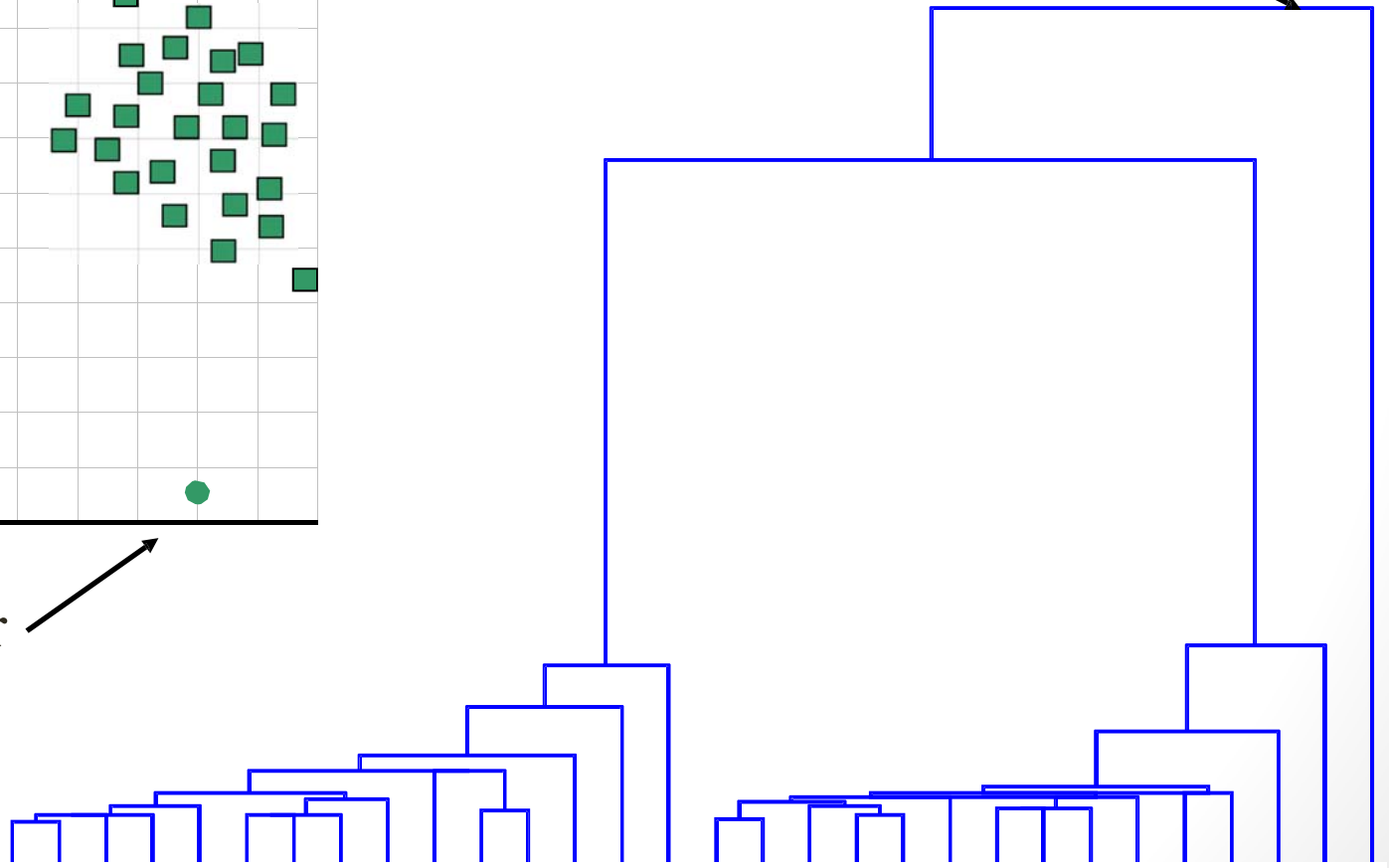
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking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

One potential use of a dendrogram is to detect outliers

The single isolated branch is suggestive of a data point that is very different to all others



Outlier

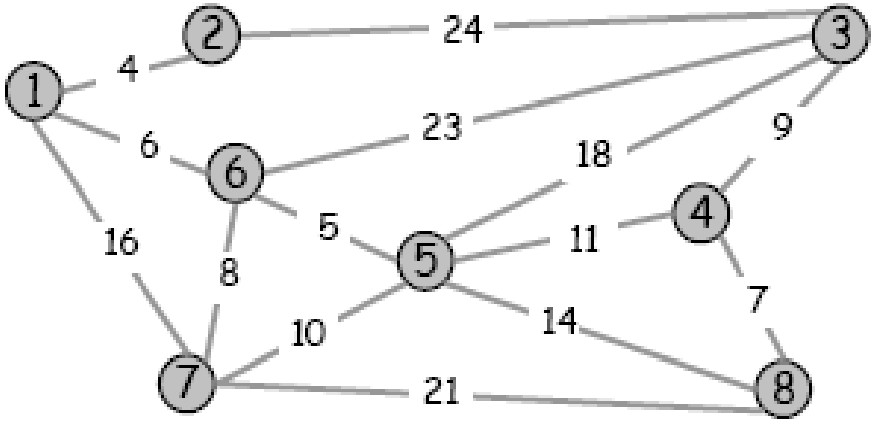


Index

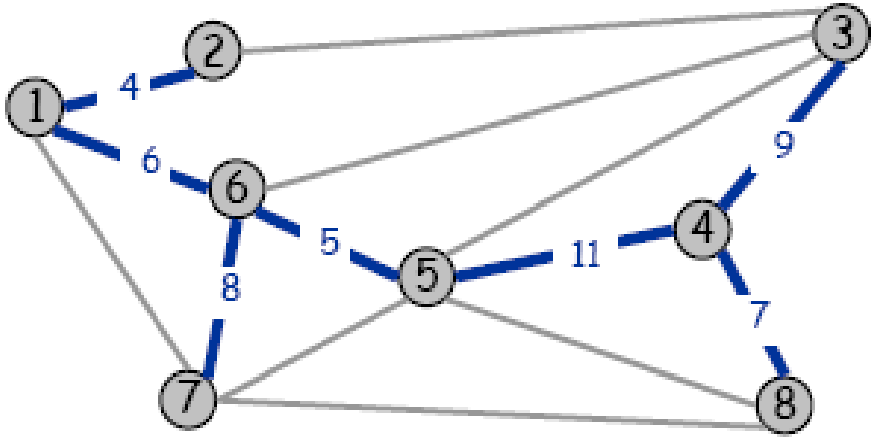
- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - Aglomerative
    - **Divisive graph based approaches (MST and flow)**
  - [Other]Density estimation algorithms: DBSCAN
- Evaluation of clusters

Minimum Spanning Tree

MST. Given connected graph G with positive edge weights, find a min weight set of edges that connects all of the vertices.



G



T

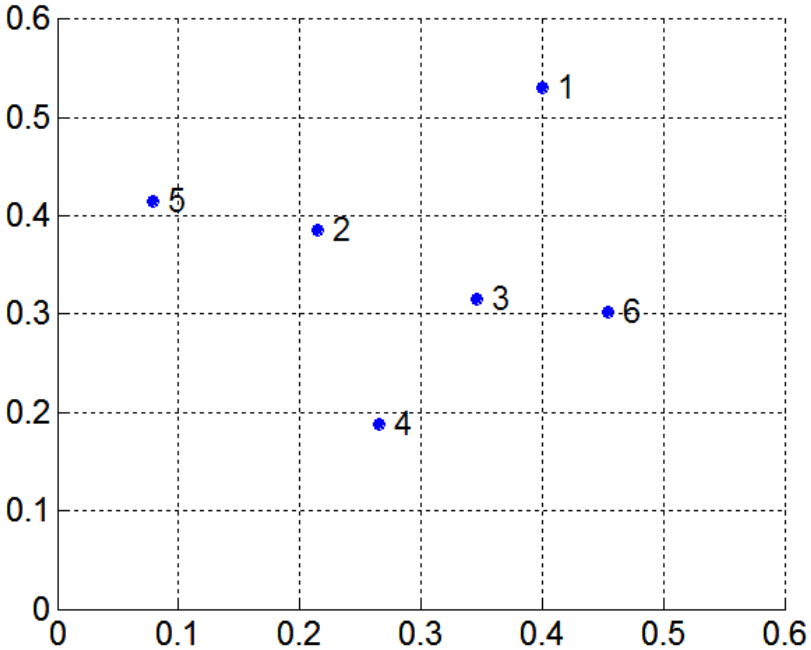
w(T) = 50

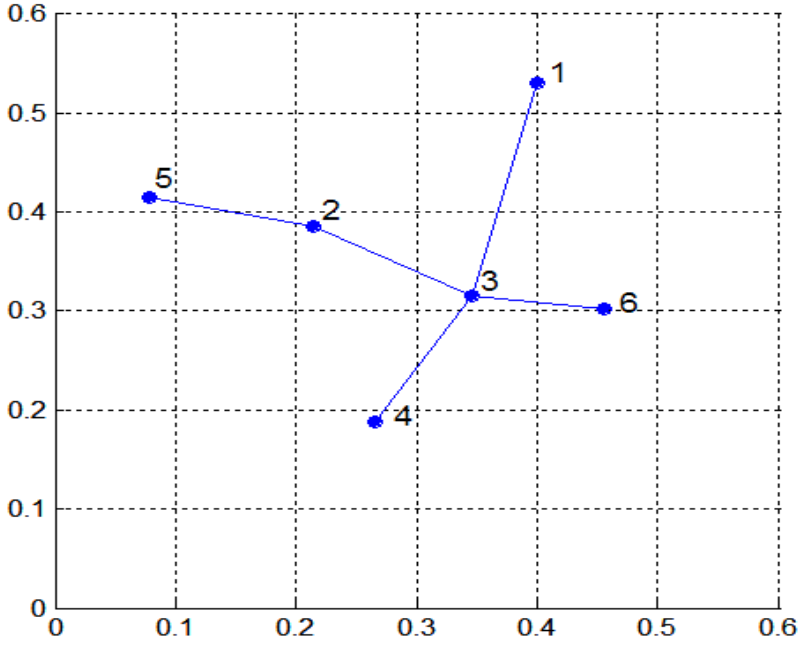
Cayley's Theorem (1889). There are V^{V-2} spanning trees on the complete graph on V vertices.

- Can't solve MST by brute force.

MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
  - Add q to the tree and put an edge between p and q





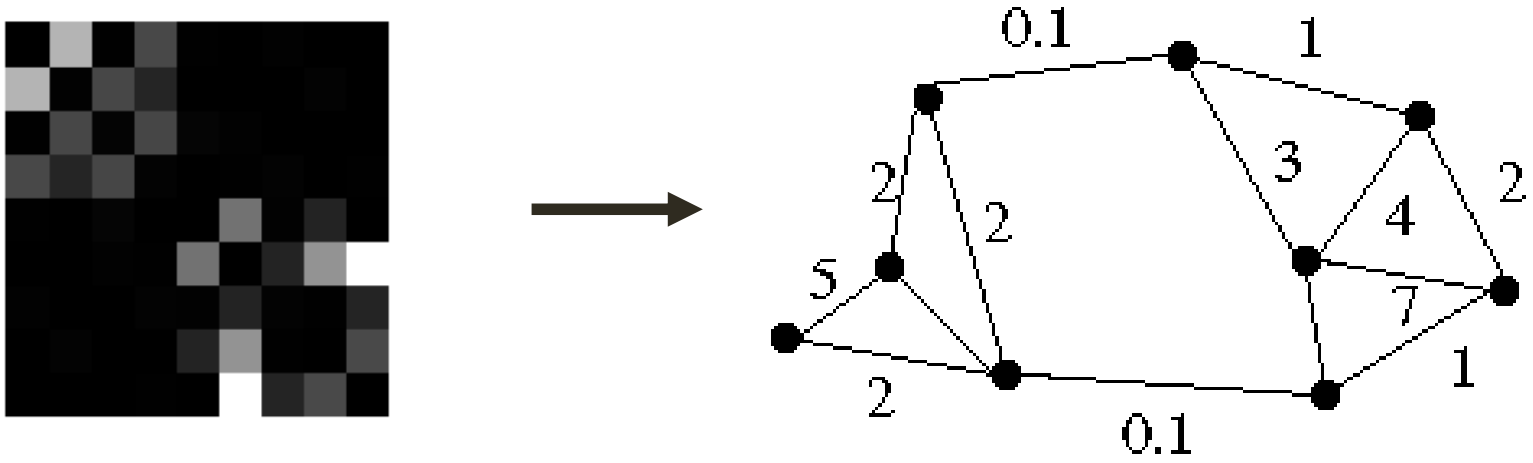
MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

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

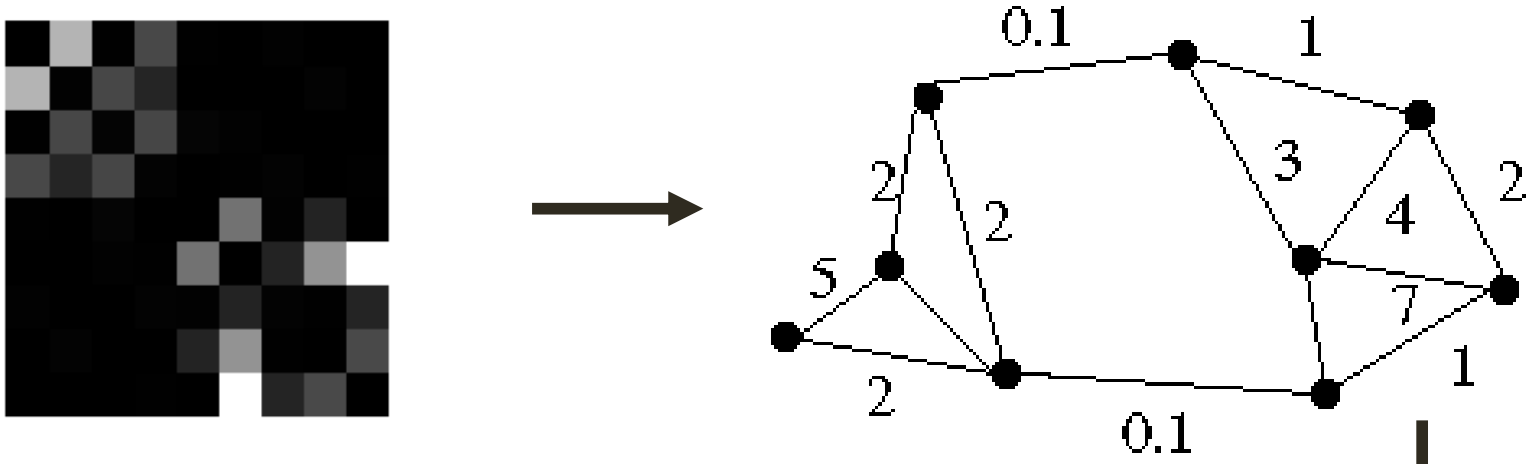
Spectral/Graph-based Clustering

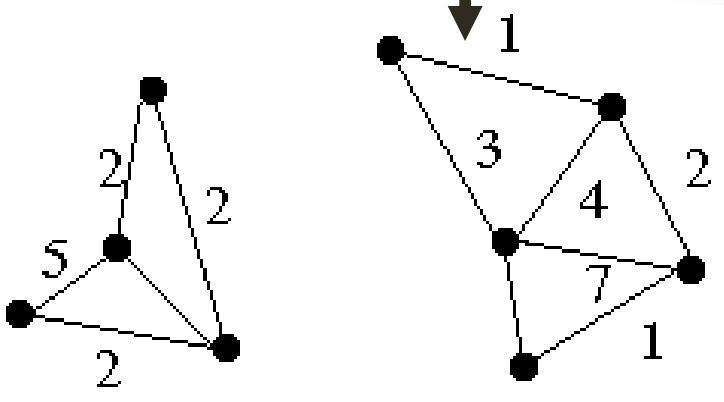


Idea:

- think of distance matrix as a weighted graph where
  - objects are nodes
  - edges exist for objects with high similarity

Spectral/Graph-based Clustering





Idea:

- think of distance matrix as a weighted graph where
  - objects are nodes
  - edges exist for objects with high similarity

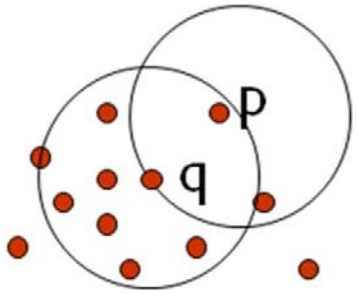
- finding a good grouping of the objects is somewhat equivalent to finding low-weight subgraphs
  - can be reduced to “min-cut” algorithms
  - related to eigenstructure of distance matrix
  - e.g., Shi and Malik, 1997, for image segmentation

Index

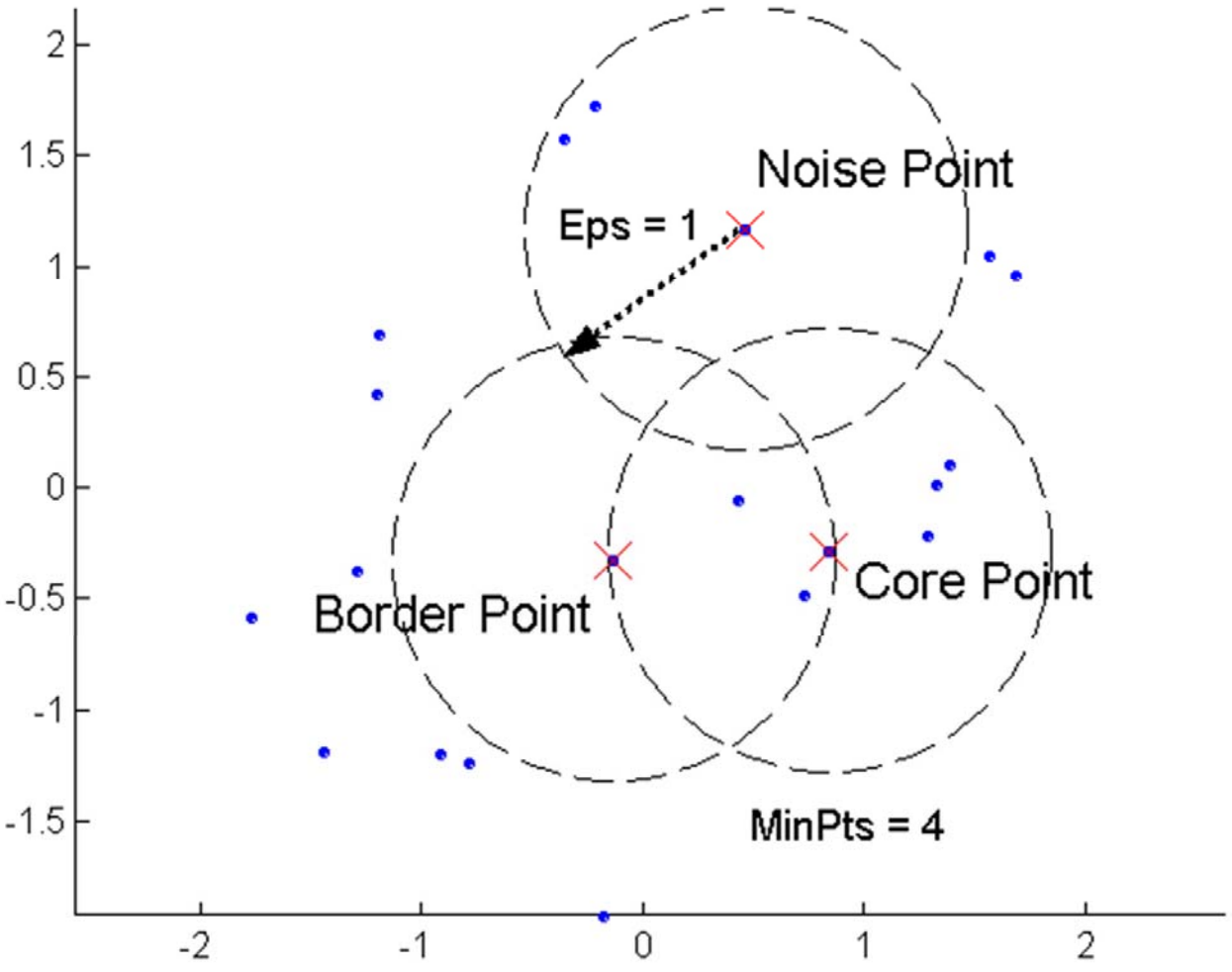
- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - **[Other]Density estimation algorithms: DBSCAN**
- Evaluation of clusters

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



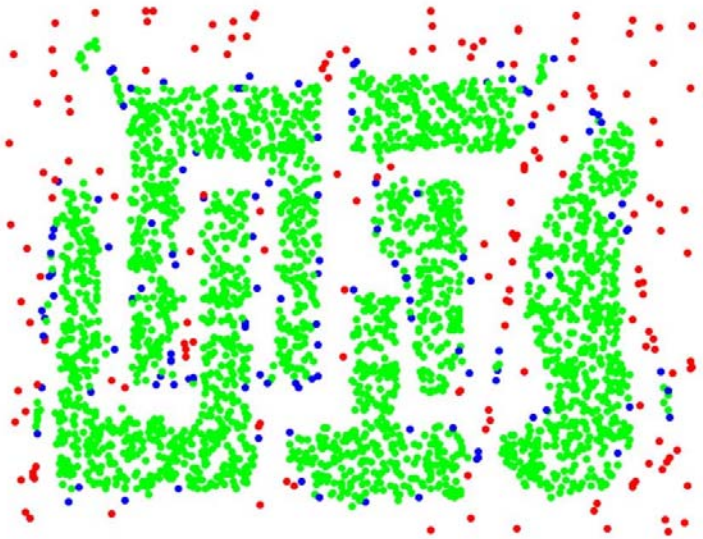
DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

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

DBSCAN: Core, Border and Noise Points

Original Points

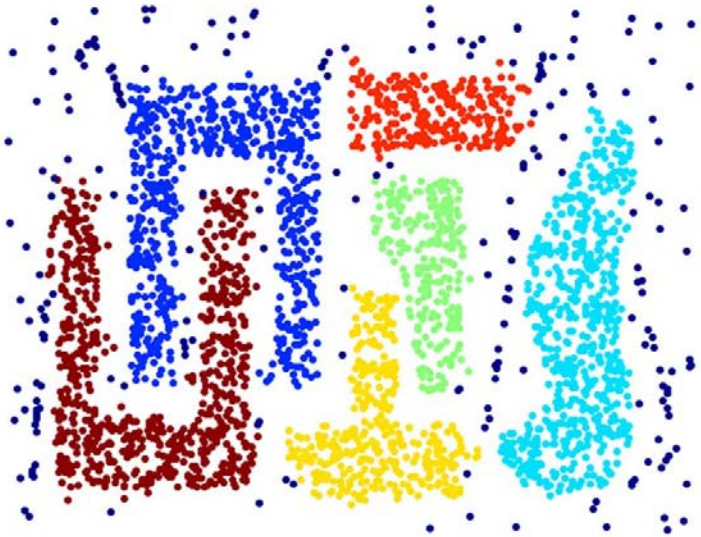


Point types: core, border and noise

Eps = 10, MinPts = 4

When DBSCAN Works Well?

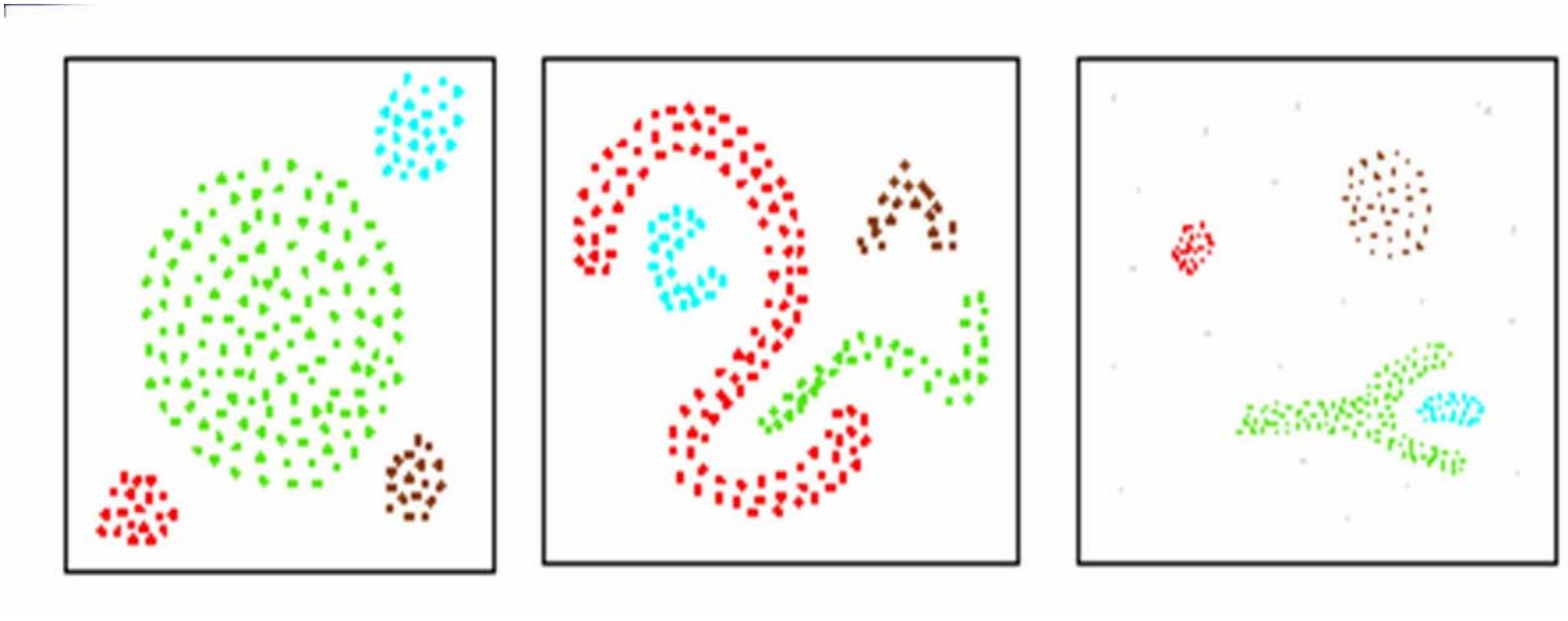
Original Points



Clusters

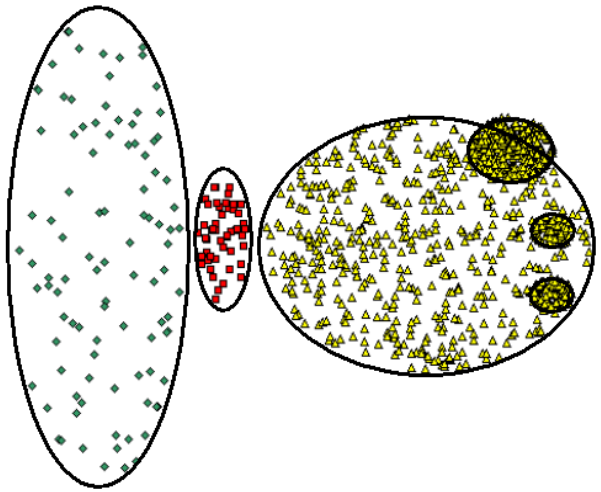
- Resistant to Noise and eliminate outliers from clusters.
- Can handle clusters of different shapes and sizes

When DBSCAN Works Well?

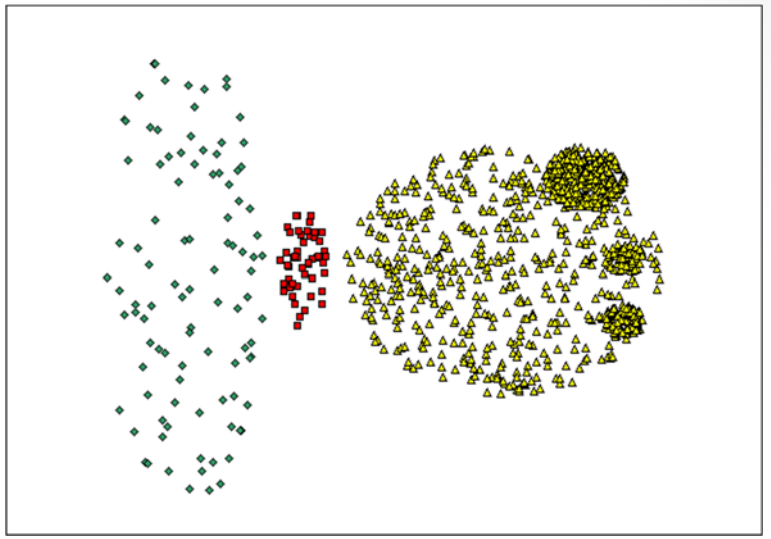


- DBSCAN does not require the number of clusters a priori
- DBSCAN requires just two parameters, and is mostly insensitive to the ordering of the points in the database

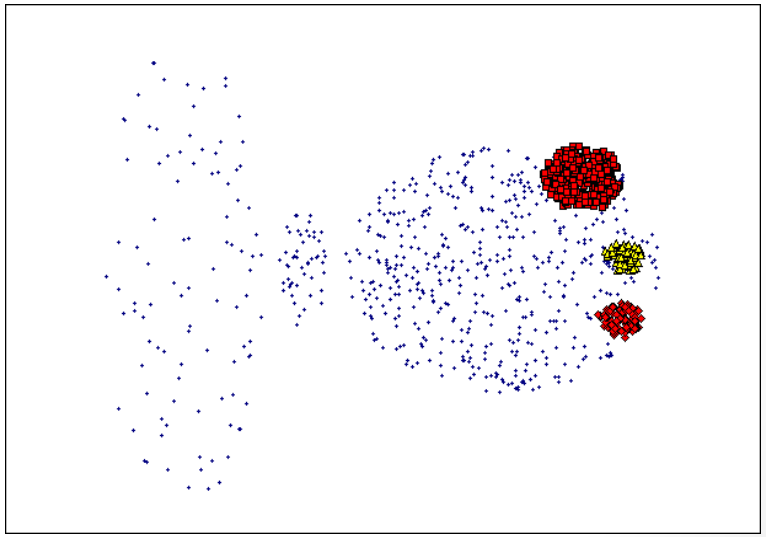
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).

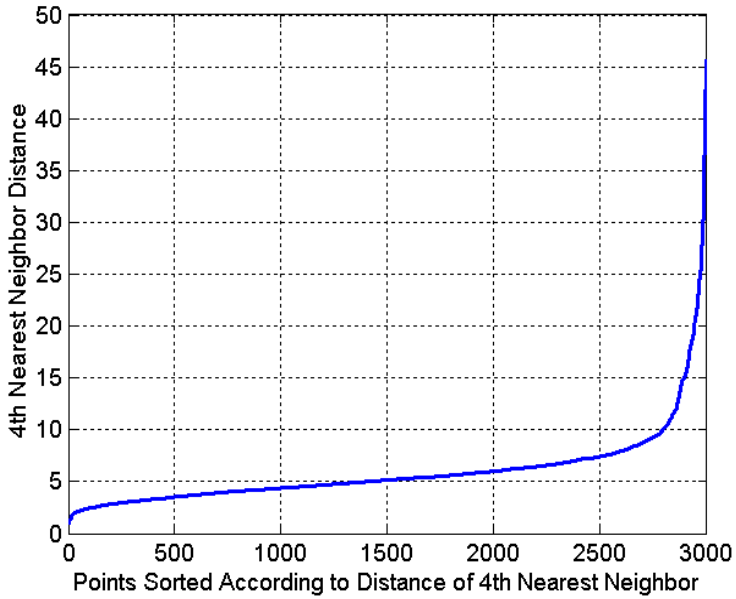


(MinPts=4, Eps=9.92)

- **Varying densities**
- **High-dimensional data**

DBSCAN: Determining EPS and MinPts

- Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance
- Noise points have the k^{th} nearest neighbor at farther distance
- So, plot sorted distance of every point to its k^{th} nearest neighbor



Index

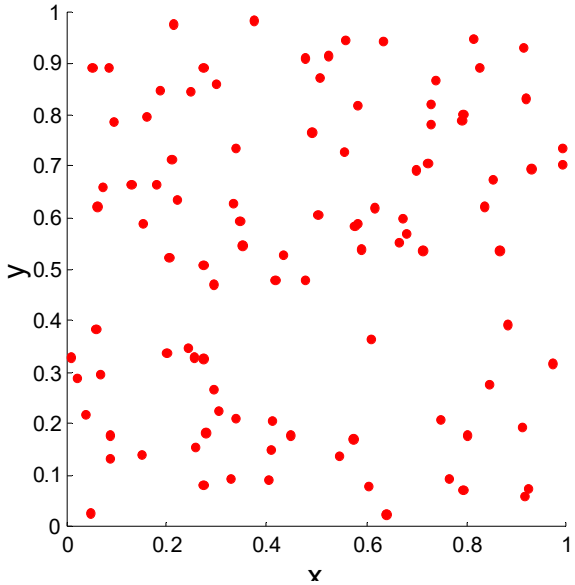
- Clustering
  - Definition
  - Areas of application
- Distances and Measures
- Algorithms:
  - Partitioning algorithms: K-means
  - Hierarchical clustering:
    - Aglomerative
    - Divisive graph based approaches (MST and flow)
  - [Other]Density estimation algorithms: DBSCAN
- **Evaluation of clusters**

Cluster Validity

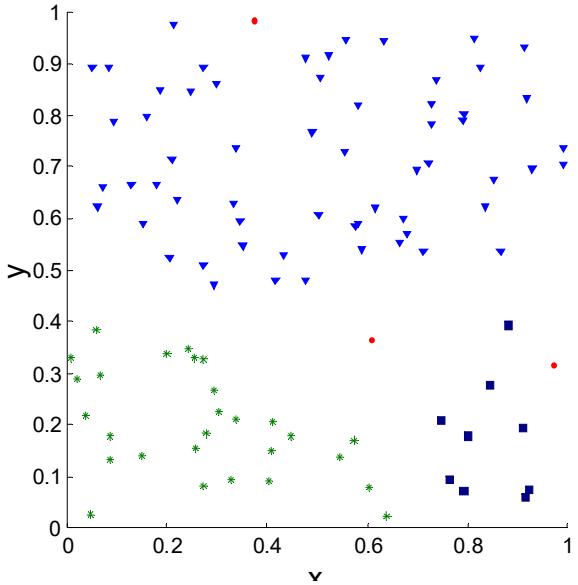
- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?
- But "clusters are in the eye of the beholder"!
- Then why do we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

Clusters found in Random Data

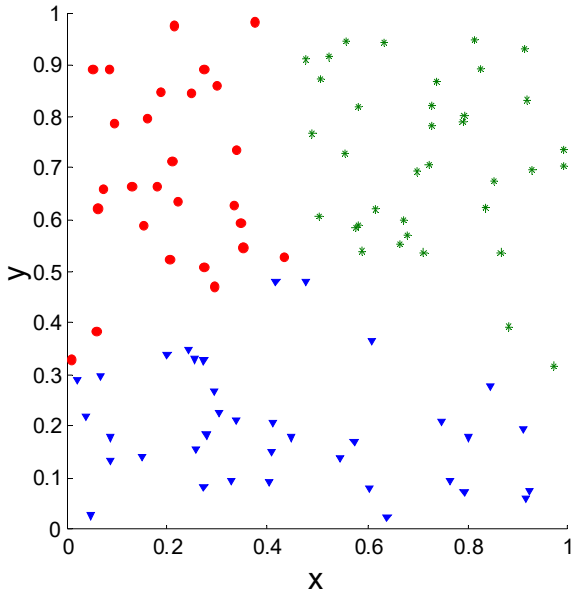
Random Points



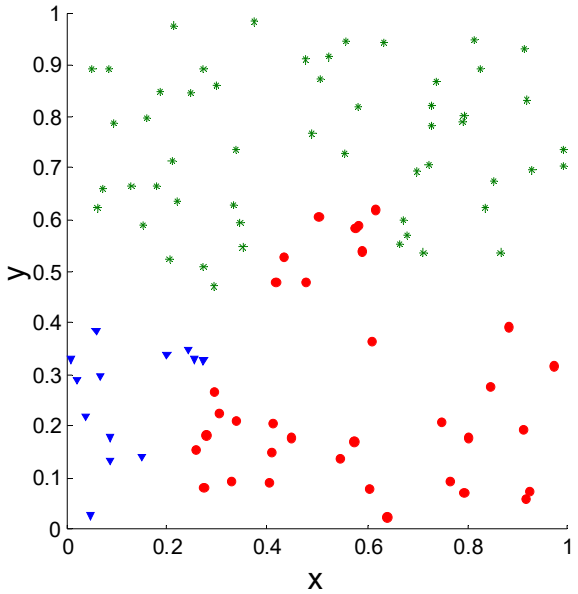
DBSCAN



K-means



Complete Link



Measures of Cluster Validity

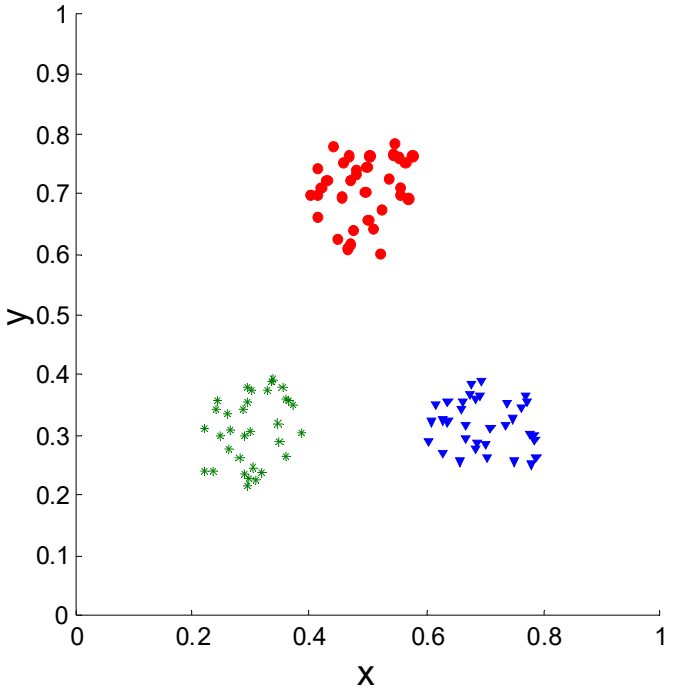
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness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

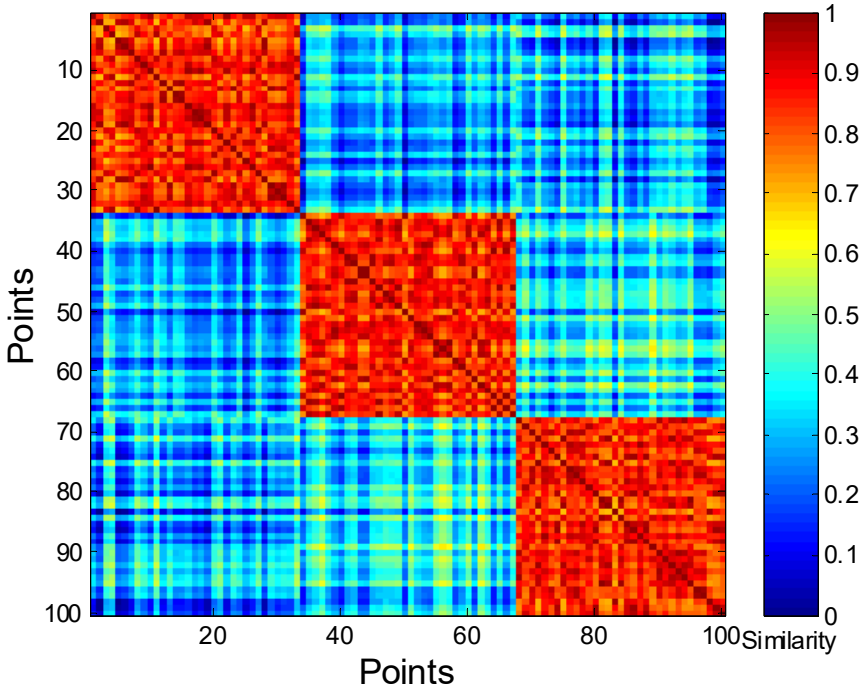
Measuring Cluster Validity Via Correlation

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

Using Similarity Matrix for Cluster Validation

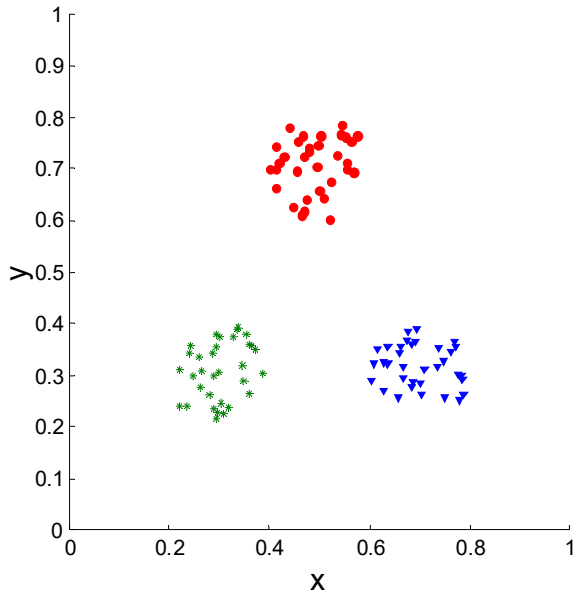
- Order the similarity matrix with respect to cluster labels and inspect visually.



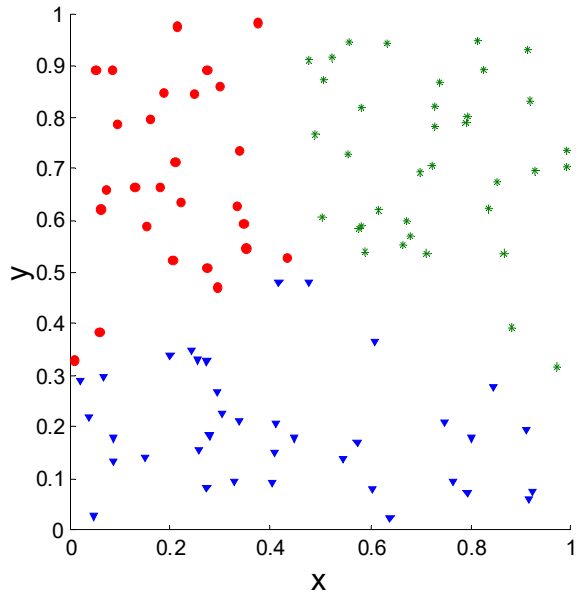


Measuring Cluster Validity Via Correlation

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



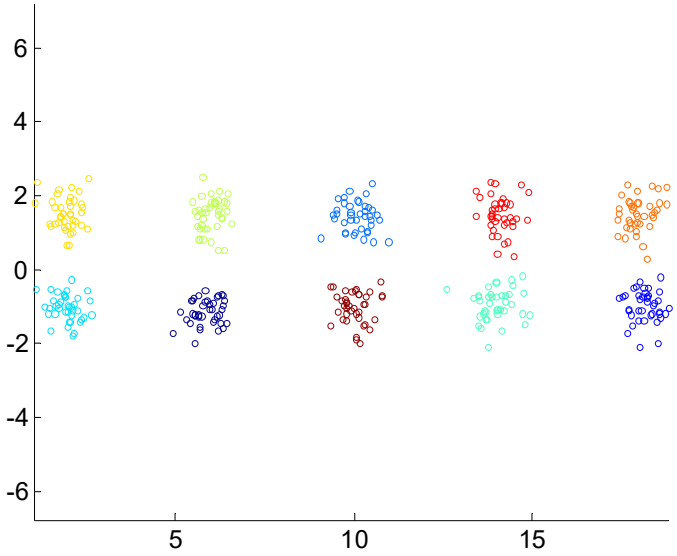
Corr = -0.9235

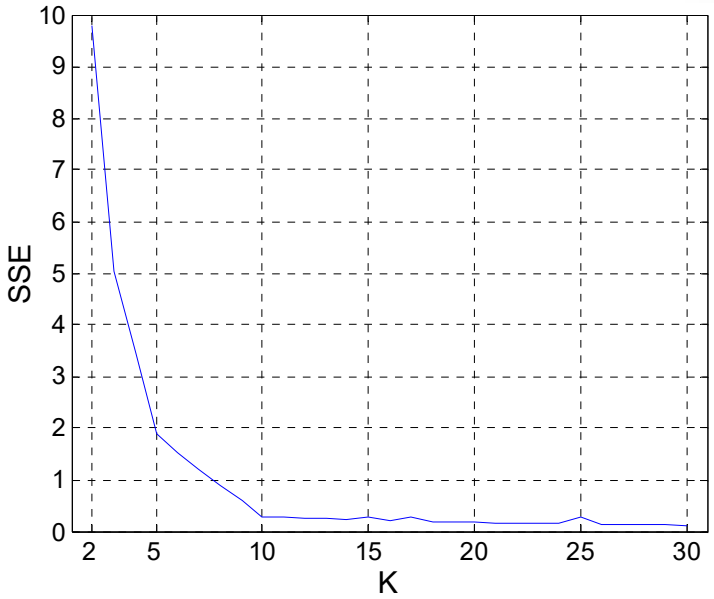


Corr = -0.5810

Internal Measures: SSE

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





Internal Measures: Cohesion and Separation

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

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

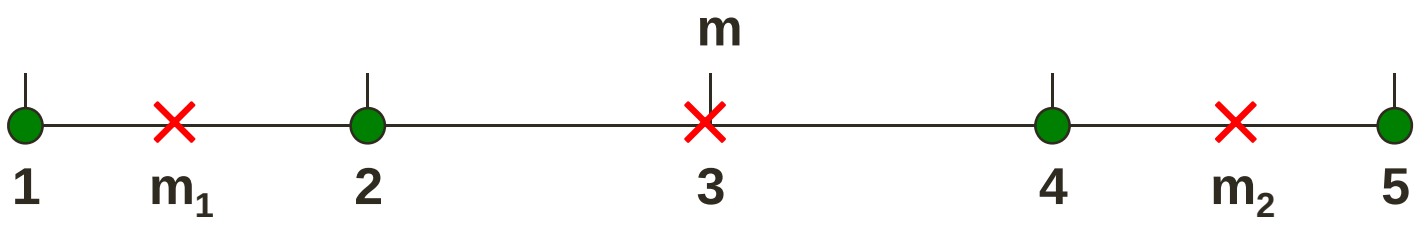
  - Separation is measured by the between cluster sum of squares

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

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

Internal Measures: Cohesion and Separation

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

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

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

$$Total = 10 + 0 = 10$$

K=2 clusters:

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

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

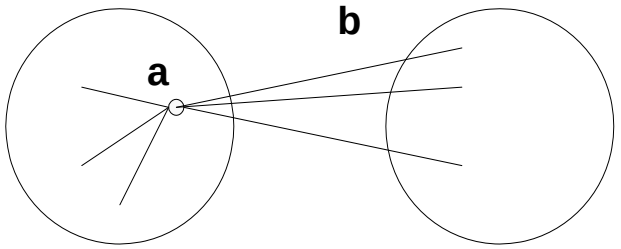
$$Total = 1 + 9 = 10$$

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)

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



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

Don’t forget....

- Every clustering algorithm will find clusters in a given data set whether they exist or not
- There is no best clustering algorithm for all problems
- The outputs of clustering algorithms only suggest or sometimes confirm hypotheses, but never prove any hypothesis about natural organization of data.