DSA5101 Introduction to Big Data for Industry

Lecture 6 Cluster Analysis: Basic Concepts and Algorithms

Li Xiaoli

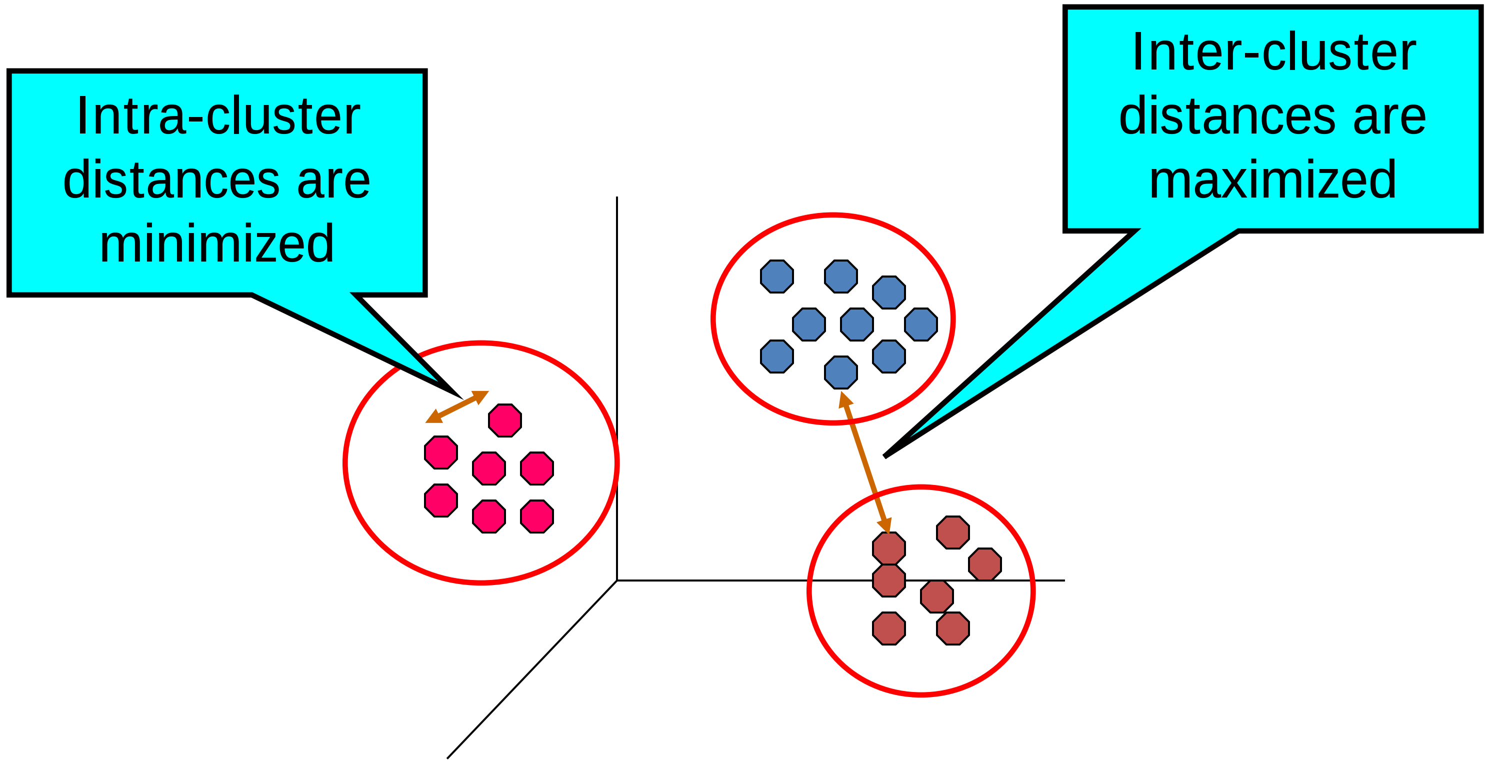
National University of Singapore

Outline

- Definition
- K-means Clustering
- Hierarchical Clustering
- DBSCAN
- Clustering Evaluation

Definition:
What is Cluster Analysis?

- 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.


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

Supervised learning vs Unsupervised Clustering

- Breast Cancer, supervised learning
  https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

id number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class: (2 for benign, 4 for malignant)
ID1	5	1	1	1	2	1	3	1	1	2
ID2	5	4	4	5	7	10	3	2	1	2
ID3	3	1	1	1	2	2	3	1	1	2
ID4	8	10	10	8	7	10	9	7	1	4

IDx	7	9	9	7	6	9	9	6	2	?

Supervised learning vs Unsupervised Clustering

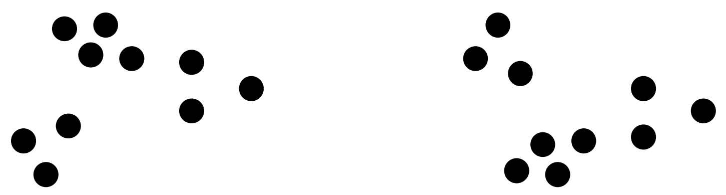
If we do not have class label, then it becomes unsupervised learning/clustering.

id number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses
ID1	5	1	1	1	2	1	3	1	1
ID2	5	4	4	5	7	10	3	2	1
ID3	3	1	1	1	2	2	3	1	1
ID4	8	10	10	8	7	10	9	7	1

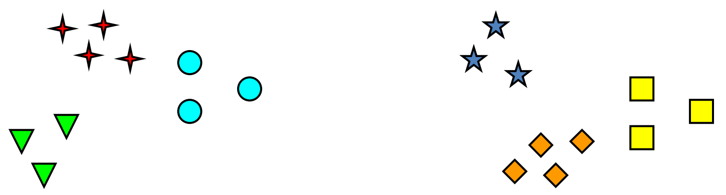
Which records (IDs) are similar and should be put them into same groups/clusters?

However, if we use another feature (e.g. Clump Thickness) as class label to do prediction, then it could become classification again. In general, if objects with the target information (i.e., have training data), then we can do supervised learning; Otherwise, we can learn the structure/organization from data by clustering (one type of unsupervised learning, as we do not have ground truth).

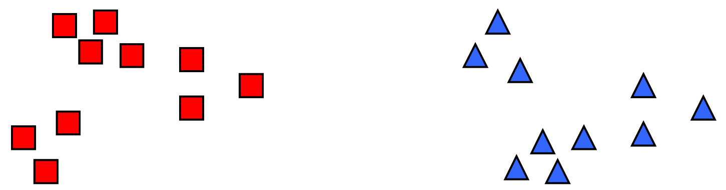
Notion of a cluster can be ambiguous or subjective



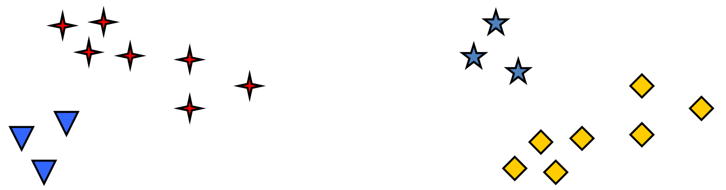
How many clusters?



Six Clusters



Two Clusters



Four Clusters

We use colors to represent the clustering results/groups

Outline

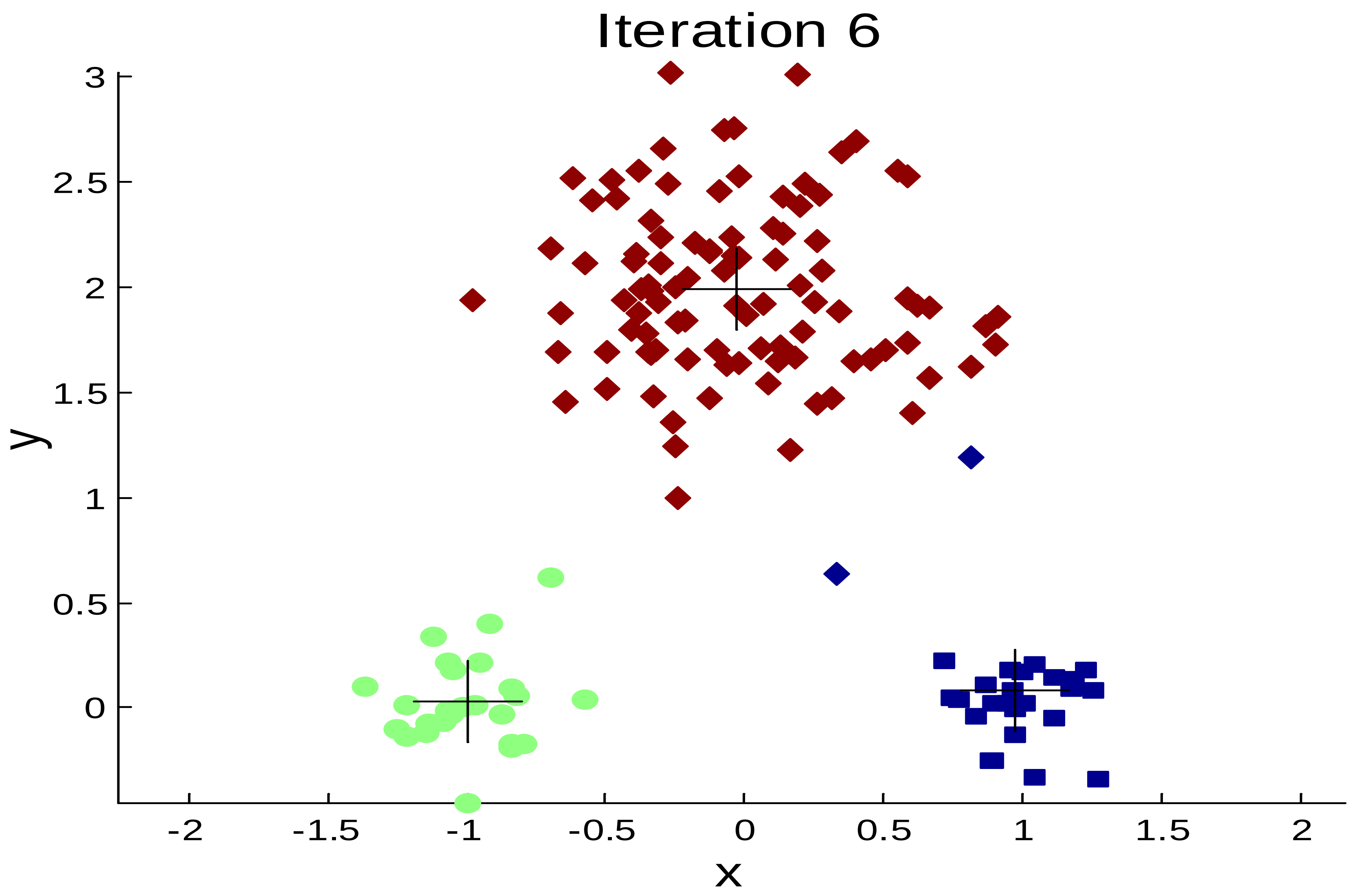
- Definition
- K-means Clustering
- Hierarchical Clustering
- DBSCAN
- Clustering Evaluation

K-means Clustering

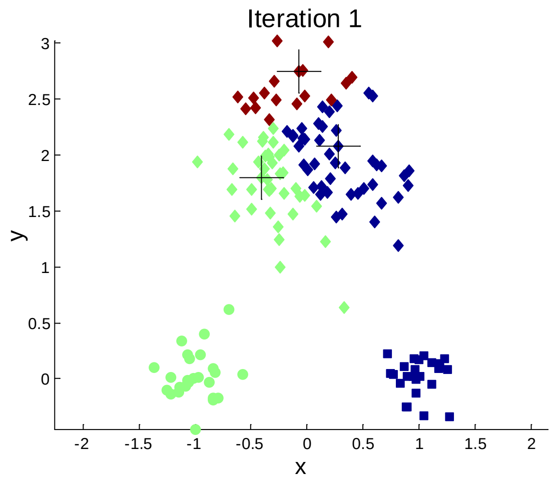
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, *K*, must be specified
- The basic algorithm is very simple

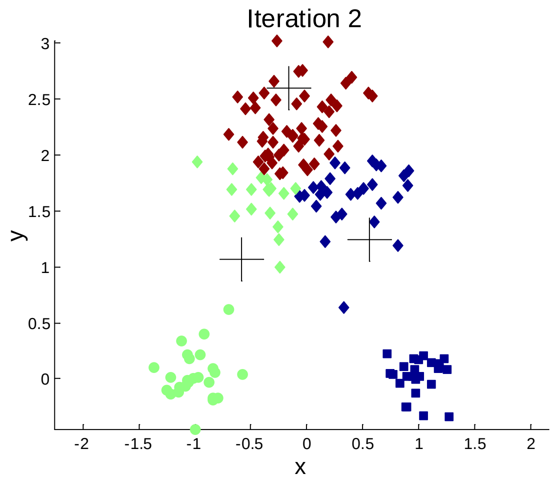
```
1: Select $K$ points as the initial centroids.
2: repeat                                                        Assignment
3:     Form $K$ clusters by assigning all points to the closest centroid.
4:     Recompute the centroid of each cluster.                   Update
5: until The centroids don't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

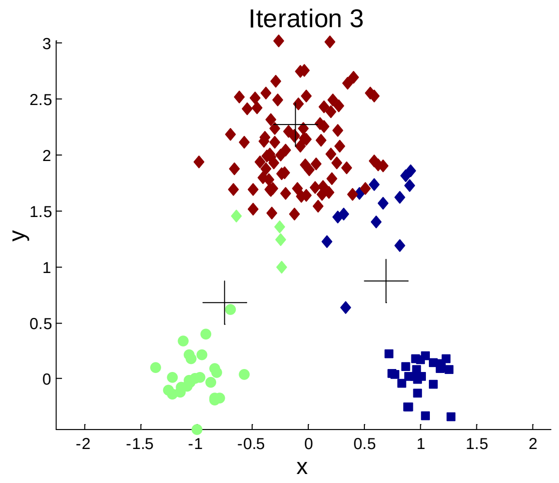
K-means Clustering Illustration

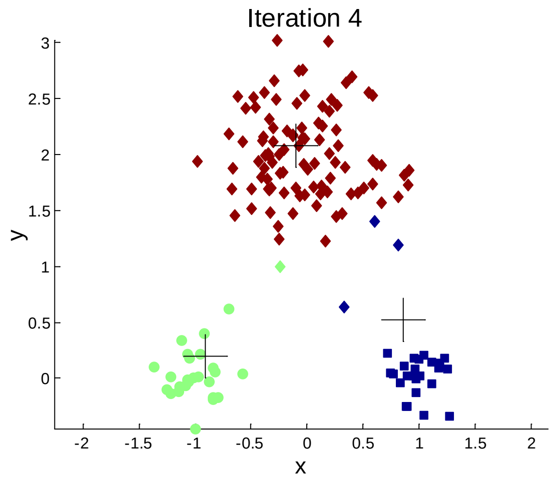


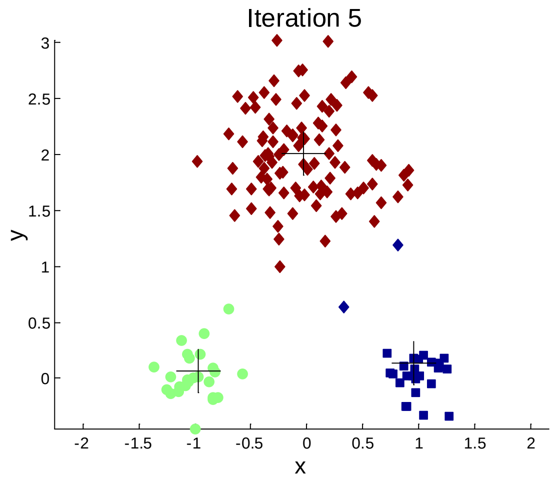
K-means Clustering Illustration

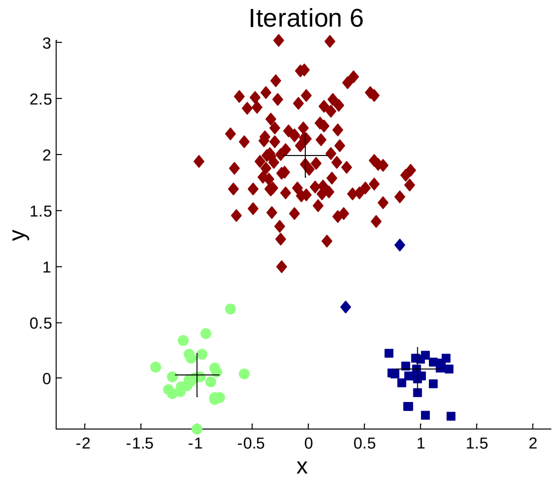












K-means Clustering – Details

- **Input of K-means**
  - Data

    $D_1=(V_{11}, V_{12}, \ldots, V_{1d})$

    $D_2=(V_{21}, V_{22}, \ldots, V_{2d})$

    ……

    $D_n=(V_{n1}, V_{n2}, \ldots, V_{nd})$
  - *K*: how many clusters do you want to group? Typically given by users

- Complexity is $O(I * n * K * d)$
  - I = number of iterations, n = number of points
  - K = number of clusters, d = number of attributes

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 (then normalization to better measure closeness)

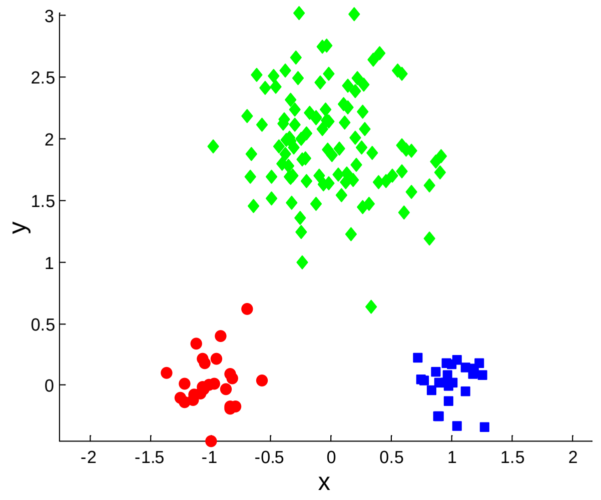
‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.

K-means will converge for common similarity measures mentioned above.

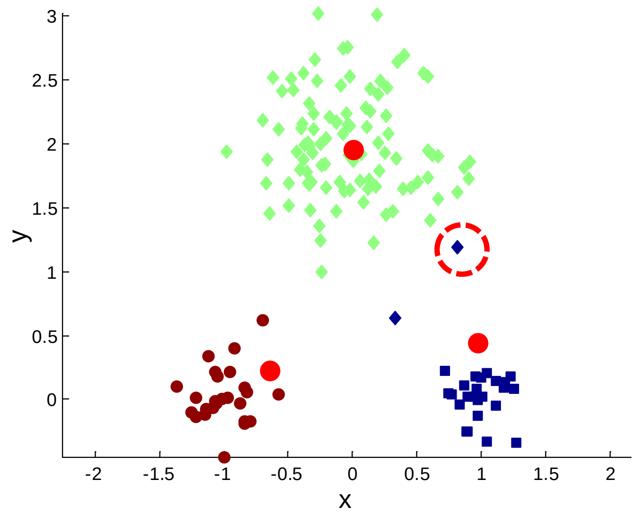
Most of the convergence happens in the first few iterations.

- Often the stopping condition is changed to ‘*Until relatively few points change clusters*’

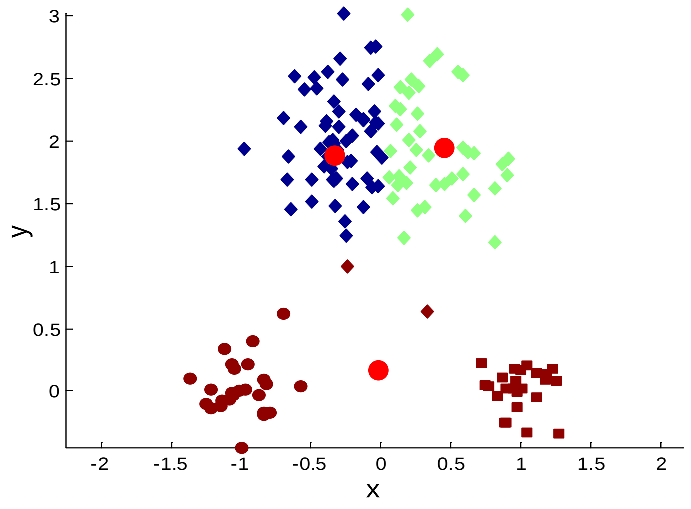
Possible Scenario:
Two different K-means Clusterings



Original Points



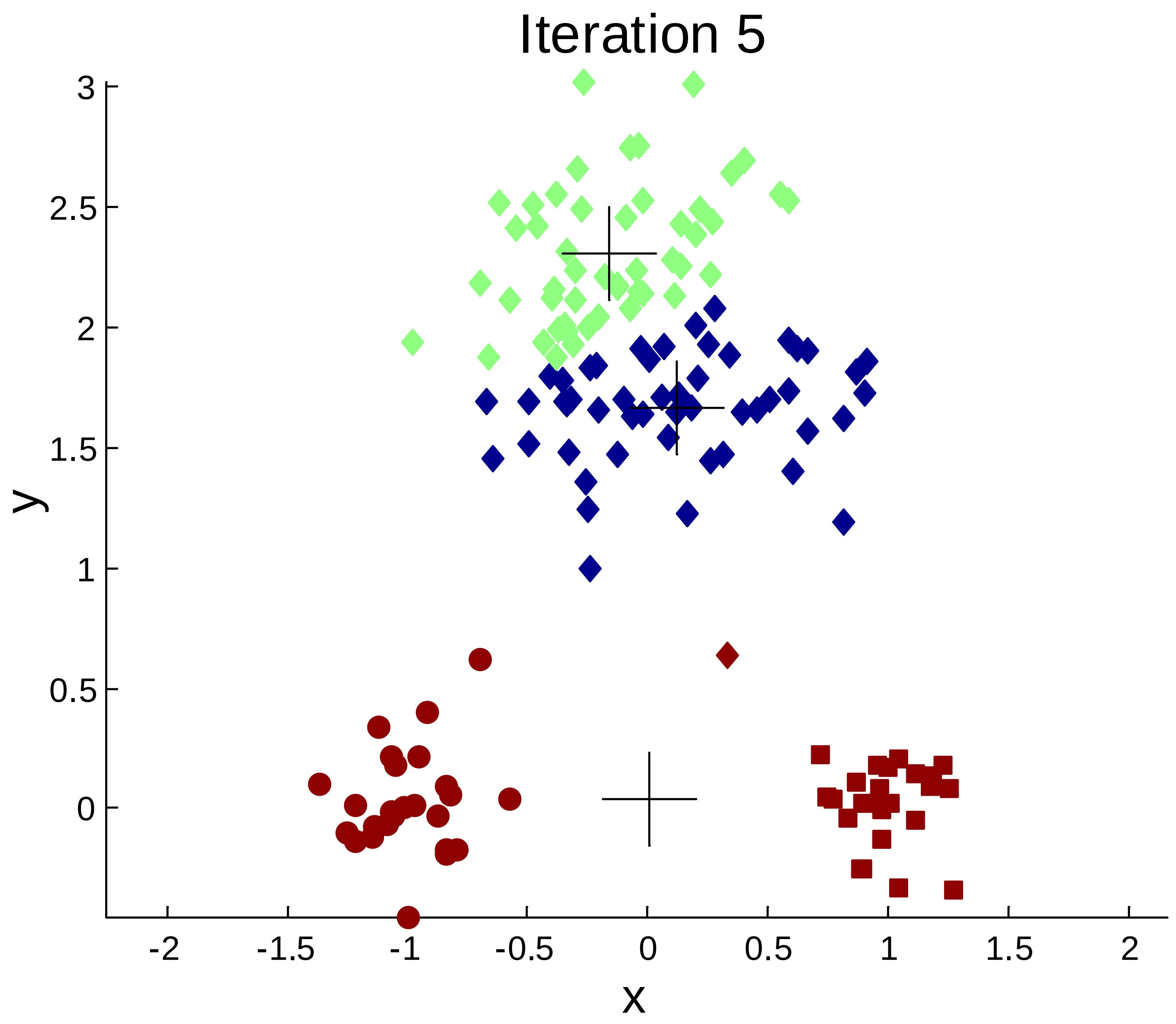
Optimal Clustering



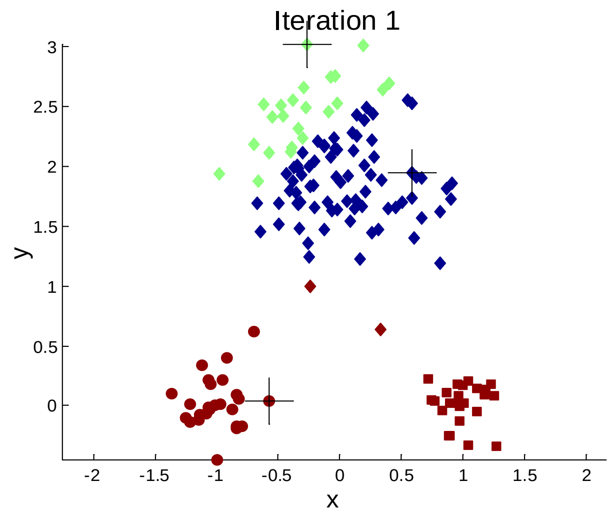
Sub-optimal Clustering

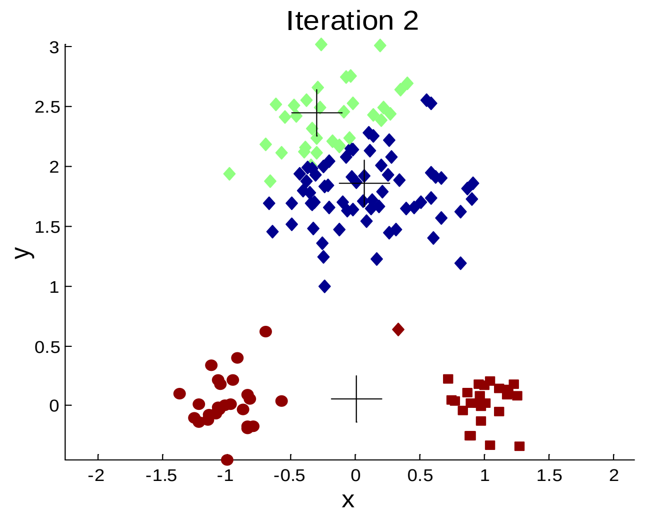
SSE of brown cluster will be very big

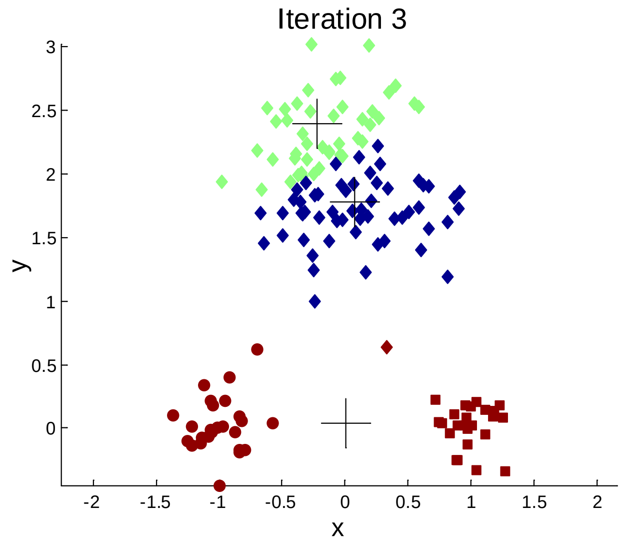
Importance of Choosing Initial Centroids …

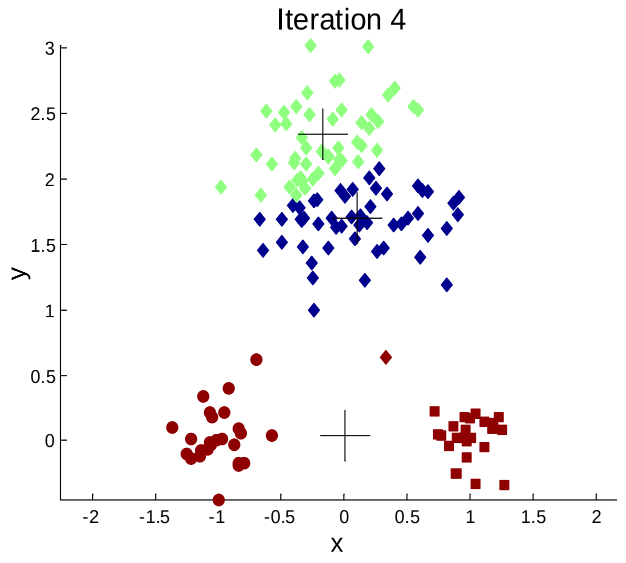


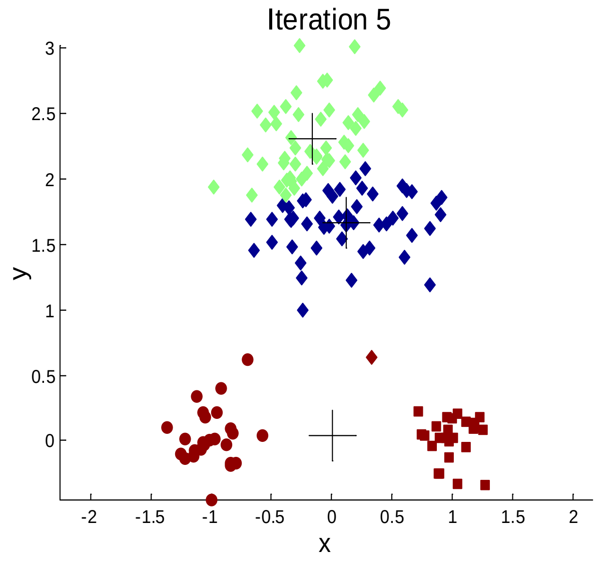
Importance of Choosing Initial Centroids …



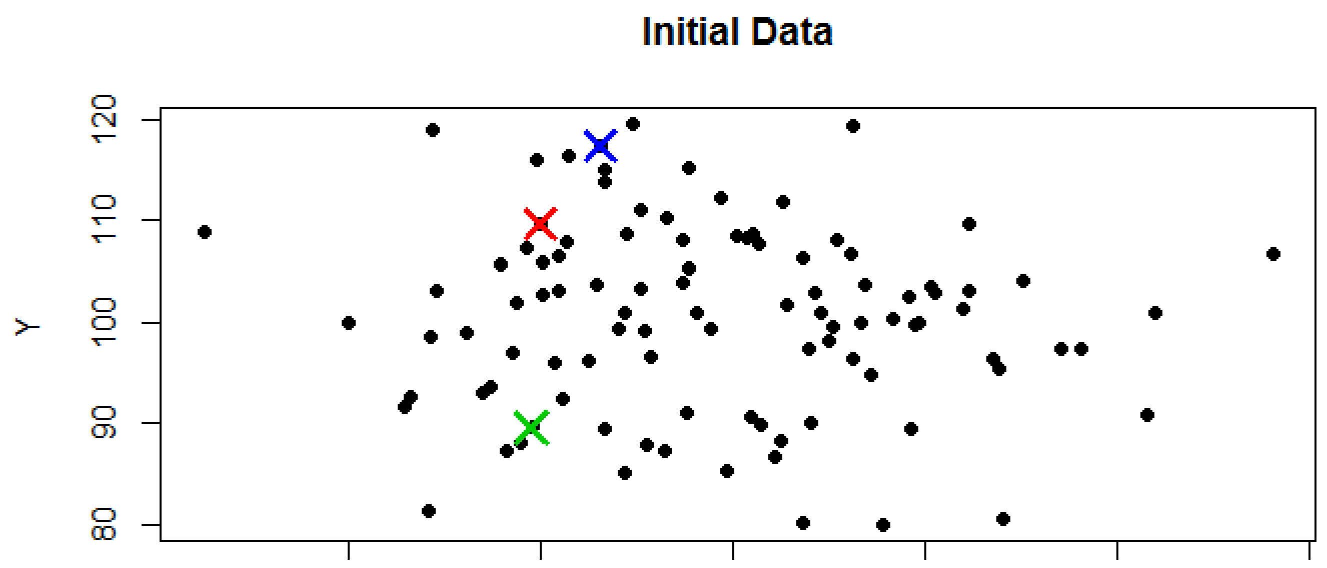




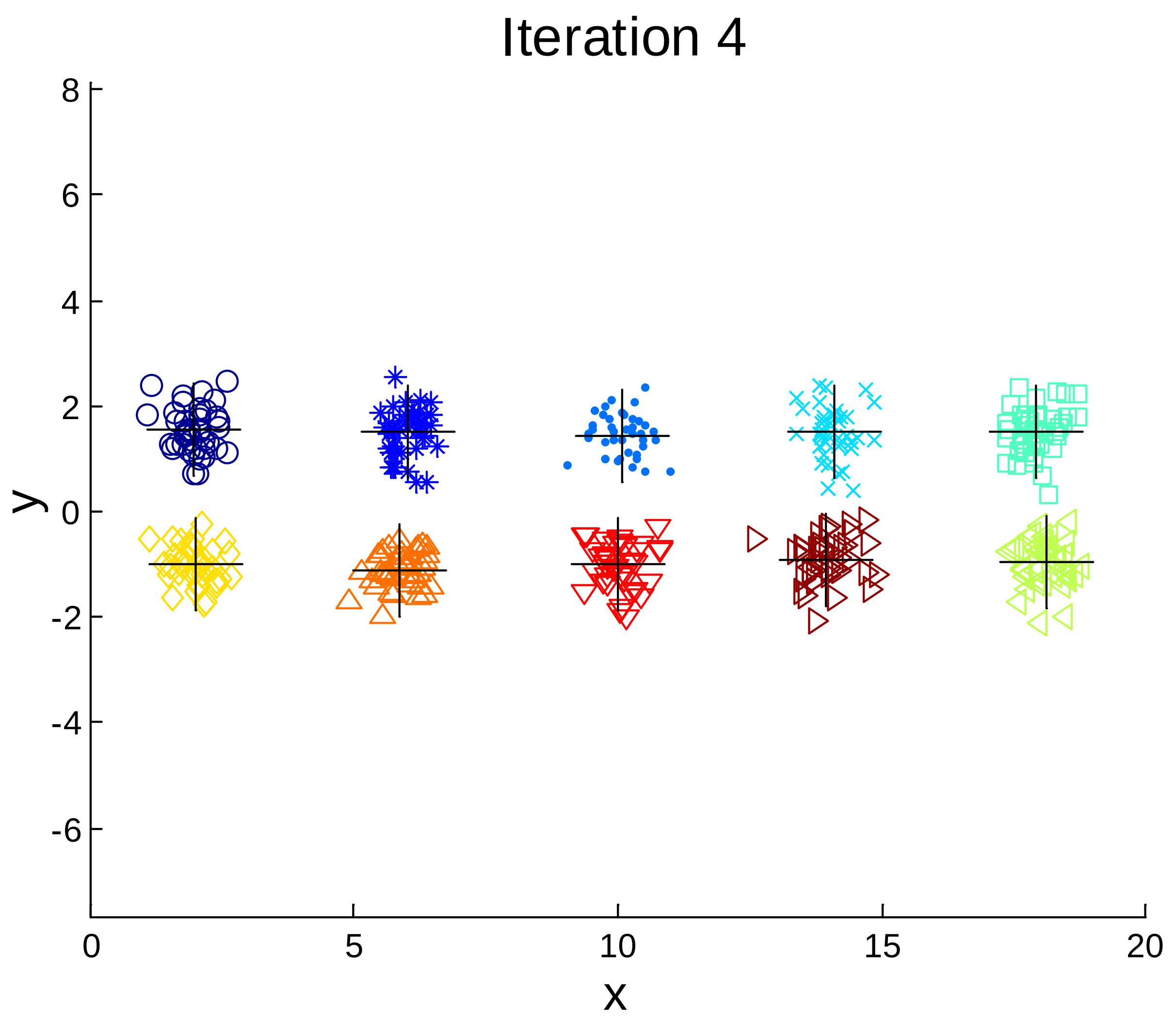




How can we choose good initial centroids?



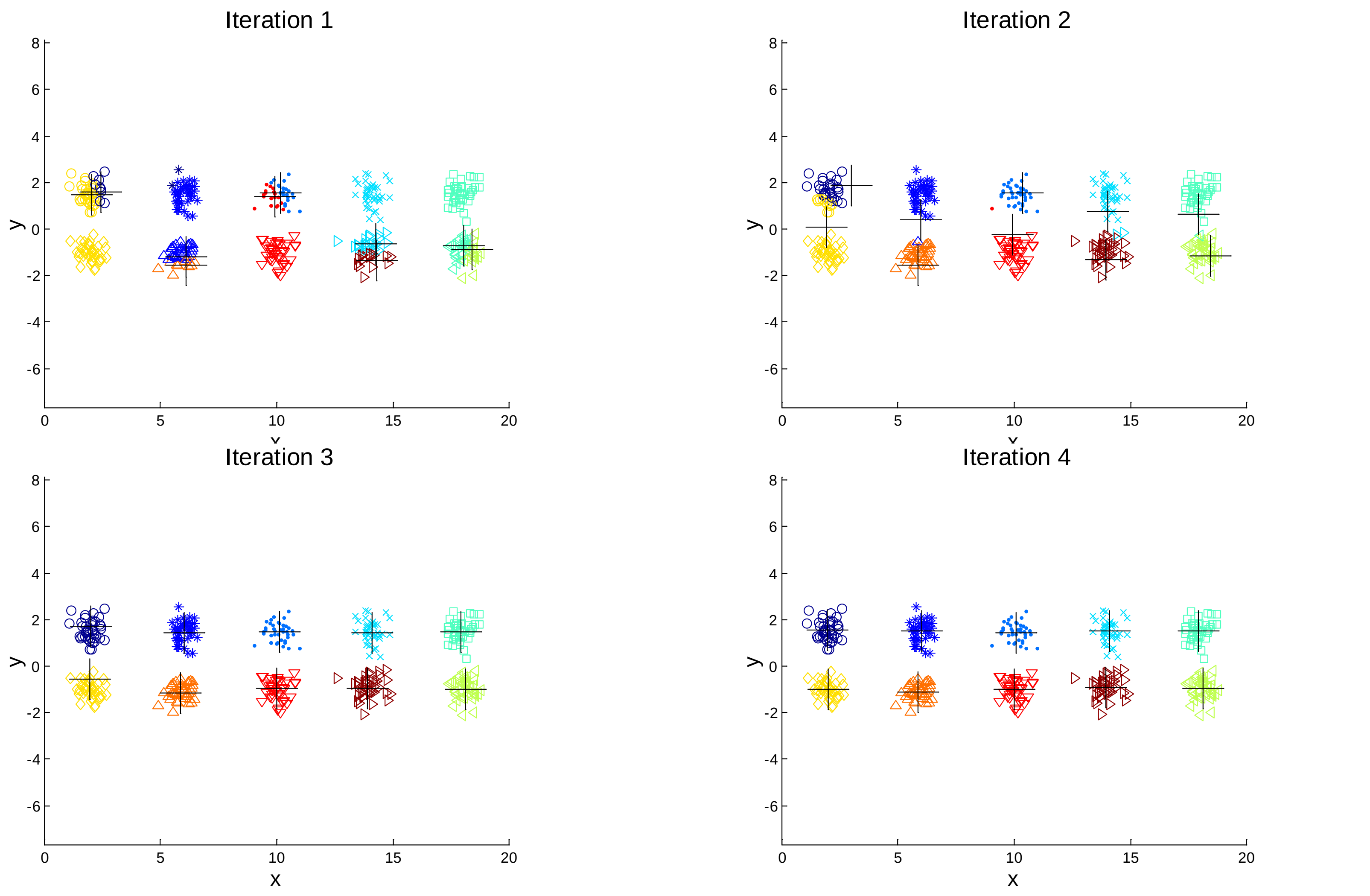
10 Clusters Example



Good adjustment:
Many points in the other cluster iteratively drag one centroid to correct location

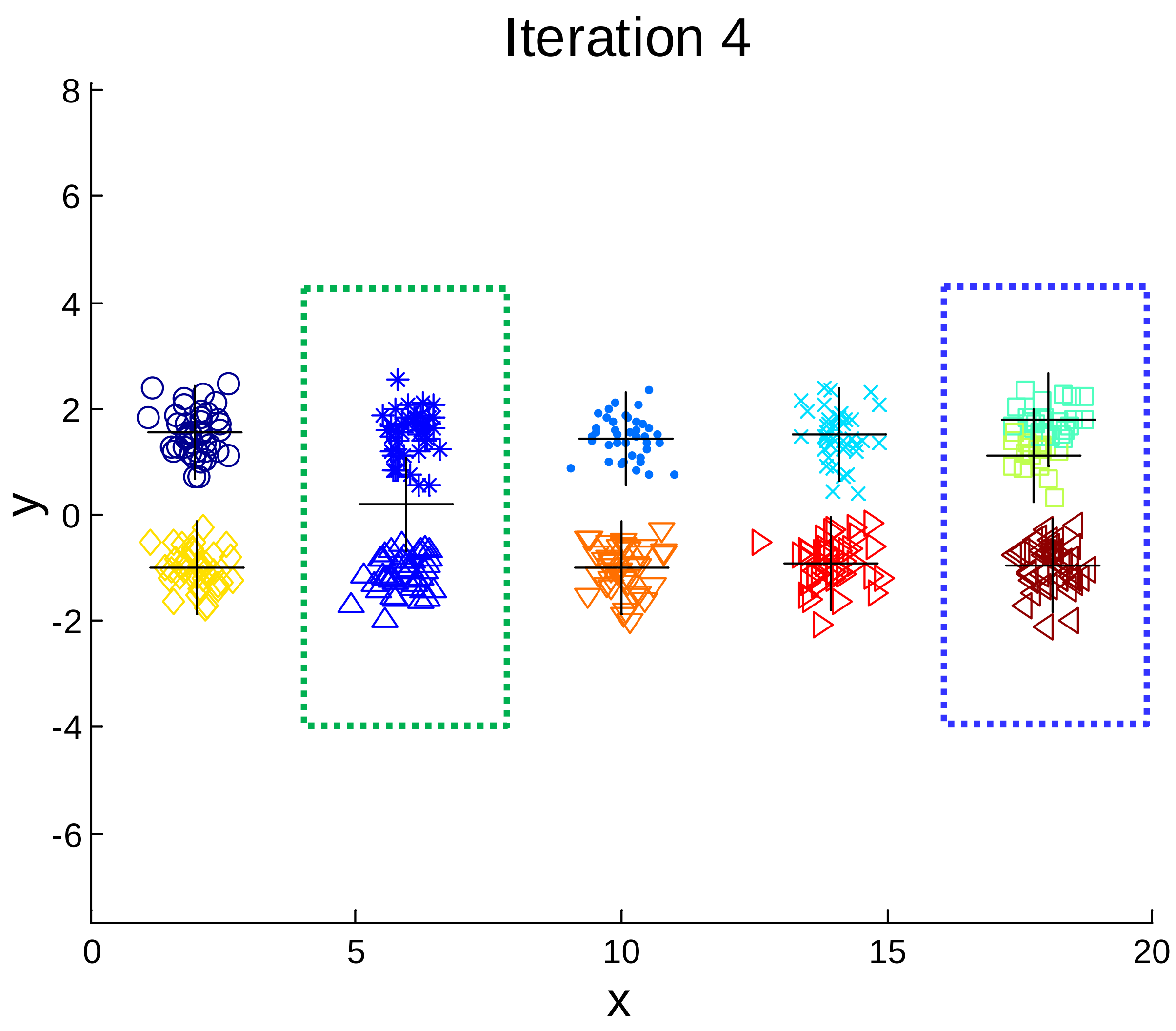
Starting with two initial centroids in one cluster of each pair of clusters (vertical pair)

10 Clusters Example



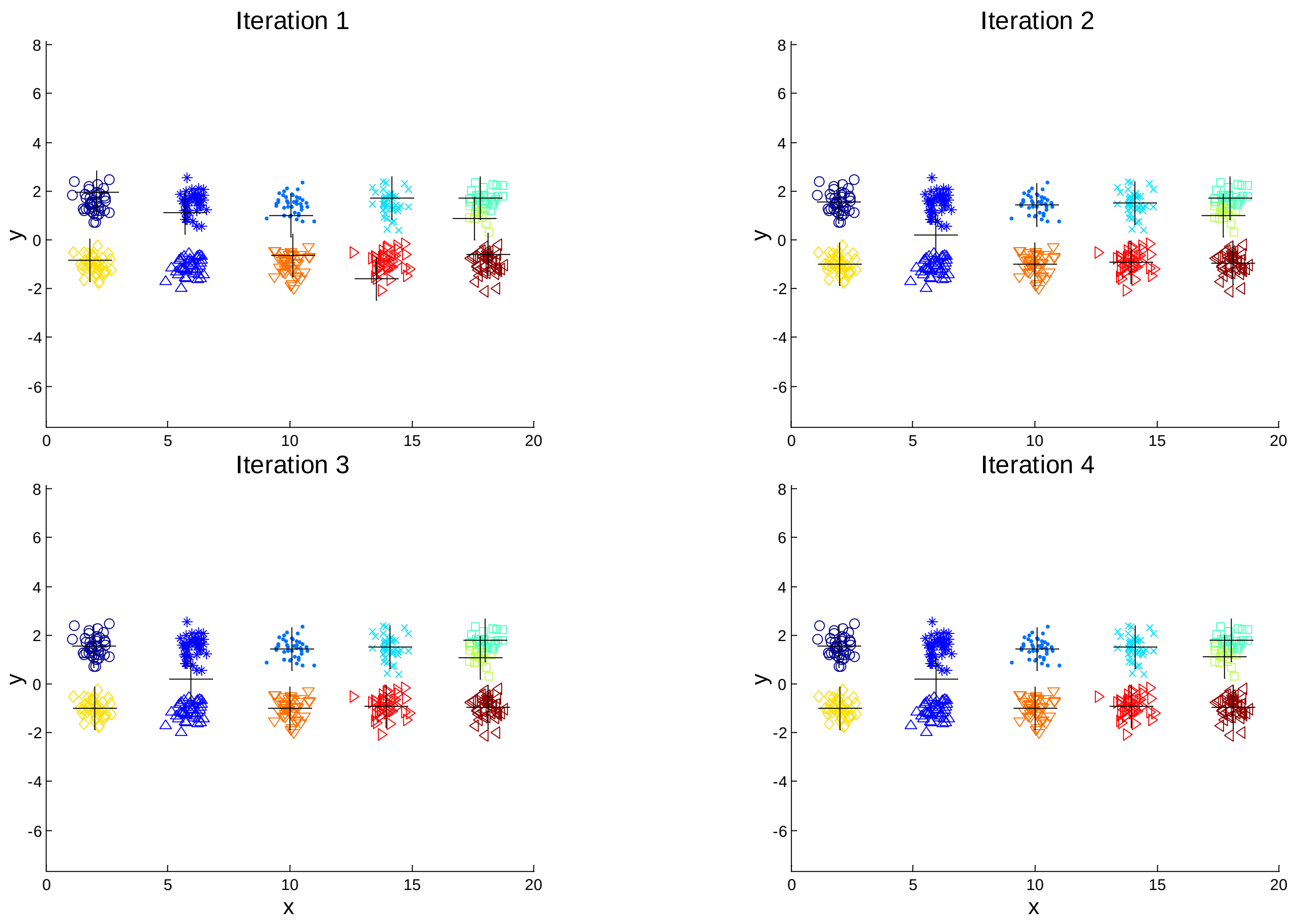
Starting with two initial centroids in one cluster of each pair of clusters (vertical pair)

10 Clusters Example



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

10 Clusters Example

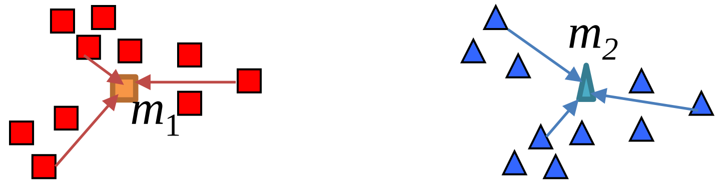


Starting with some pairs of clusters having three initial centroids, while other have only one.

Evaluating K-means (and other) Clustering methods

- 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.

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

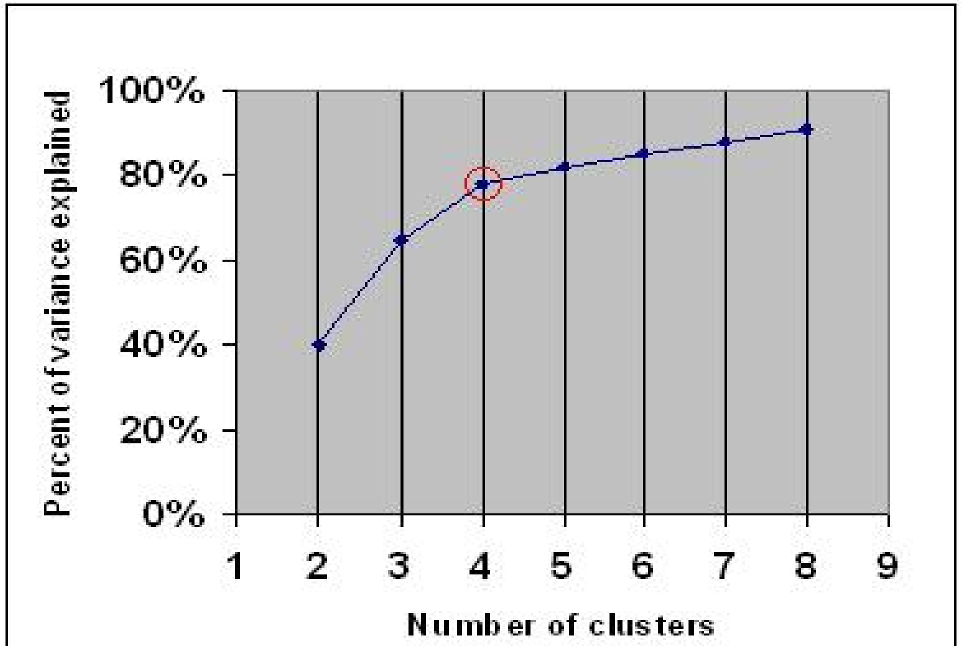


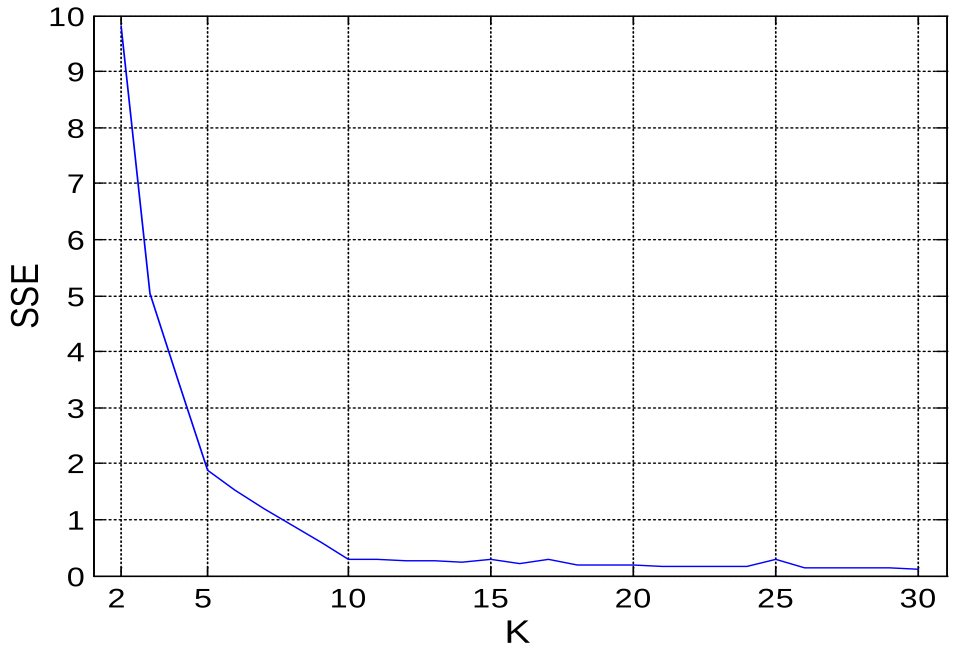
  - *x* is a data point in cluster C_i and m_i is the representative point for cluster C_i (i.e., centroid of the cluster)
  - Given two clustering results, we can choose the one with the smallest SSE error (overall/individual compact or loose)
  - One easy way to reduce SSE is to increase *K*, the number of clusters, why? Think about *k=n.*
    - A good clustering with smaller *K* can have a lower SSE than a poor clustering with higher *K*

Elbow method

From Wikipedia

- A heuristic used in determining the number of clusters in a data set. The method consists of plotting the **explained variation** (variance) as a function of the number of clusters and picking the elbow of the curve as the number of clusters to use.
- Using the "elbow" or "knee of a curve" as a cutoff point is a common heuristic in mathematical optimization to choose a point where diminishing returns are no longer worth the additional cost. In clustering, this means one should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data.





Solutions to Initial Centroids Problem

Multiple runs – you may need to do many times

- Helps, but probability is not on your side (you have less chance to choose initial centroids from all the ground-truth clusters)

Use **hierarchical clustering** to determine initial centroids

Select **more than *K*** initial centroids and then select among these initial centroids

- Select most widely separated (remove some centroids that near to other centroids)

Postprocessing (do clustering for more than *K*, finding clusters with low quality to separate, or merge)

Bisecting K-means

- Not as susceptible to initialization issues

Bisecting K-means

- Bisecting K-means algorithm
  - Variant of K-means that can produce a partitional or a hierarchical clustering

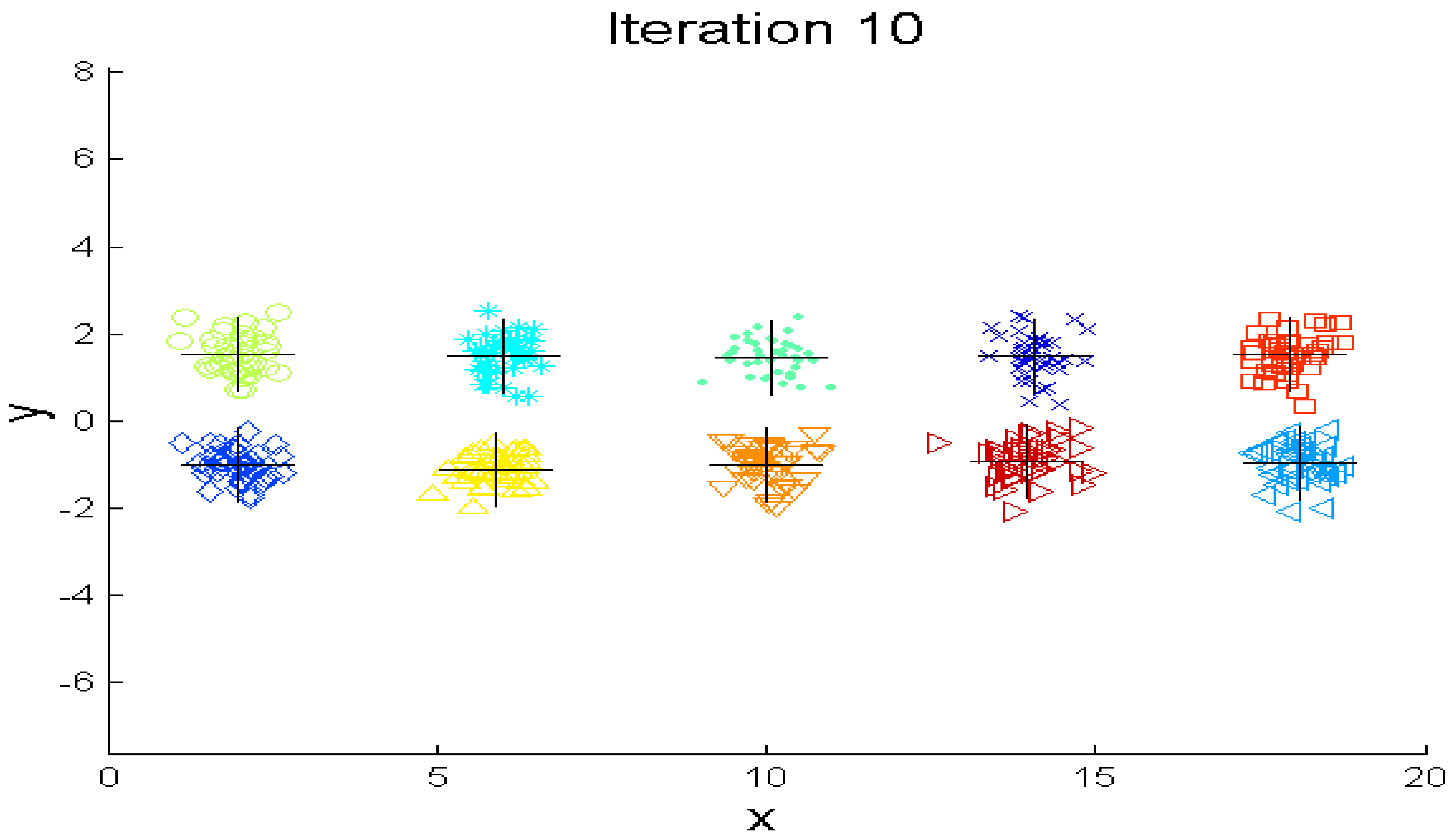
Initialization, 1 cluster or 2 clusters

```
1: Initialize the list of clusters to contain the cluster containing all points.
2: repeat
3:   Select a cluster from the list of clusters
4:   for $i = 1$ to $number\_of\_iterations$ do
5:     Bisect the selected cluster using basic K-means
6:   end for
7:   Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: until    the list of clusters contains $K$ clusters
```

Choose the worse quality one

Perform multiple times to generate multiple bisections. Take the best bisect (i.e. lowest SSE)

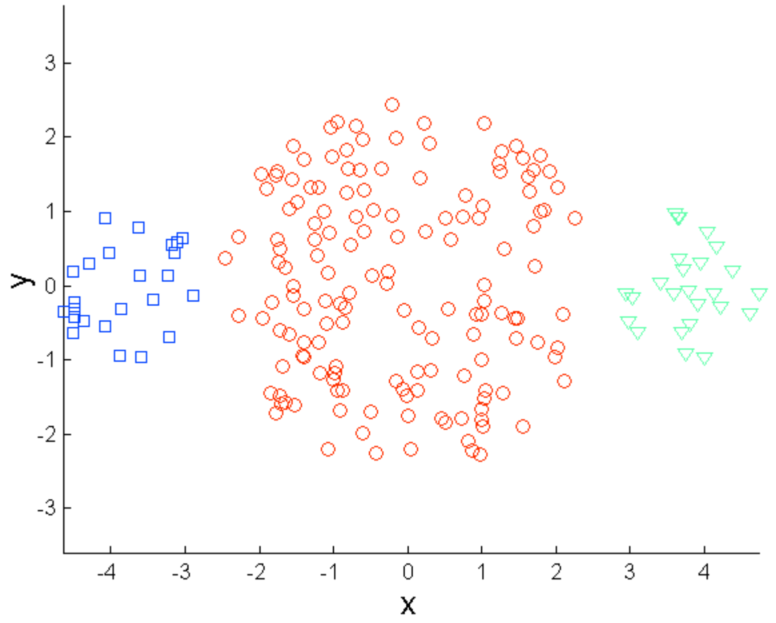
Bisecting K-means Example



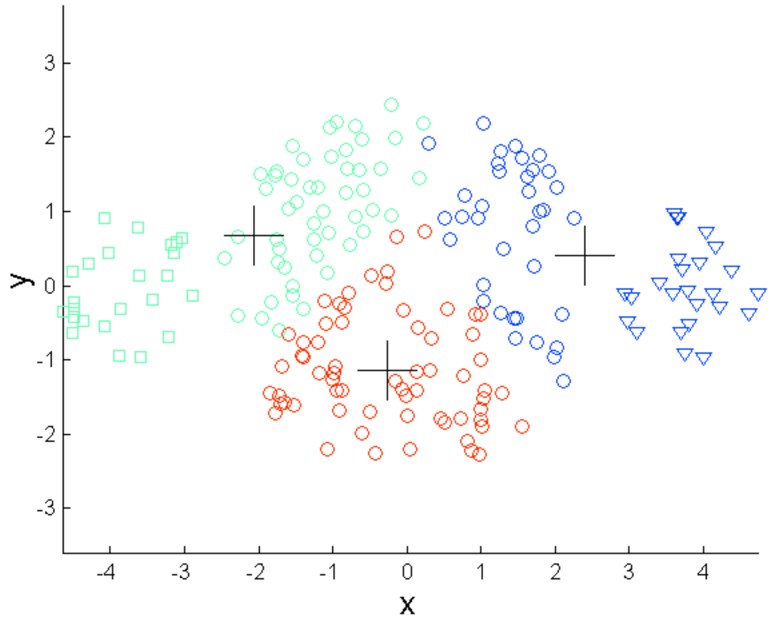
Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular (non-sphere) shapes
- K-means has problems when the data contains outliers.

Limitations of K-means: Differing **Sizes**



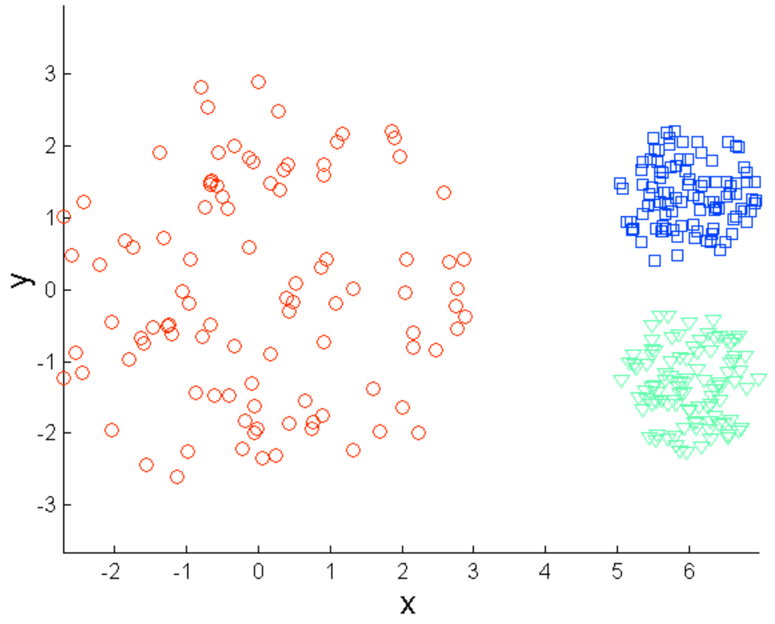
Original Points



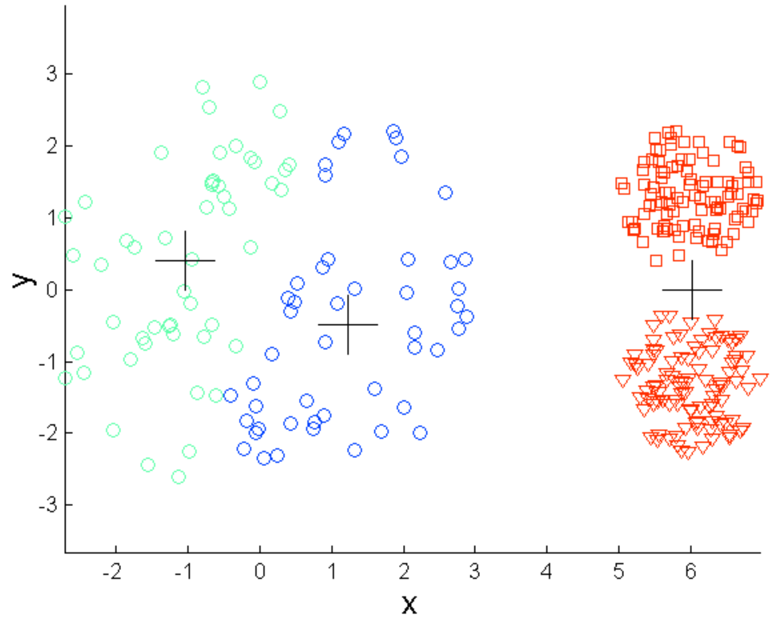
K-means (3 Clusters)

Big size clusters could be partitioned or eaten up by small clusters

Limitations of K-means: Differing **Density**

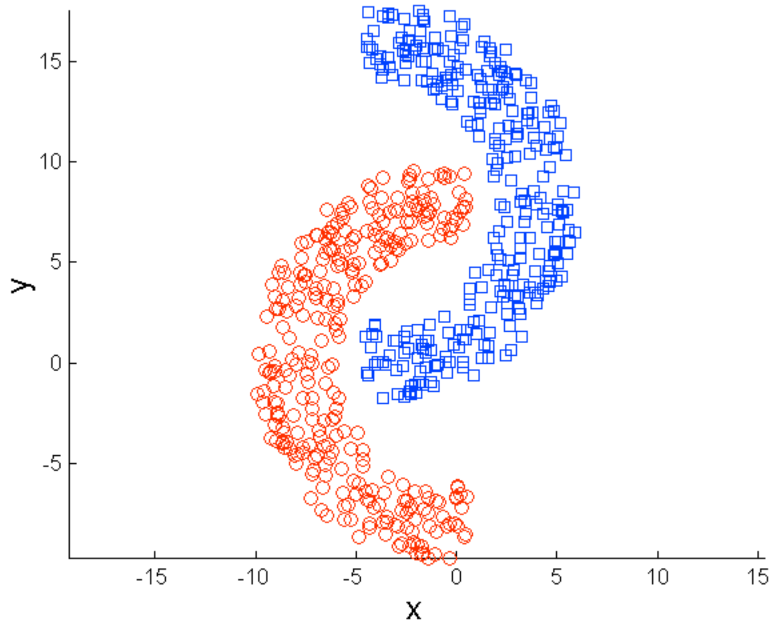


Original Points

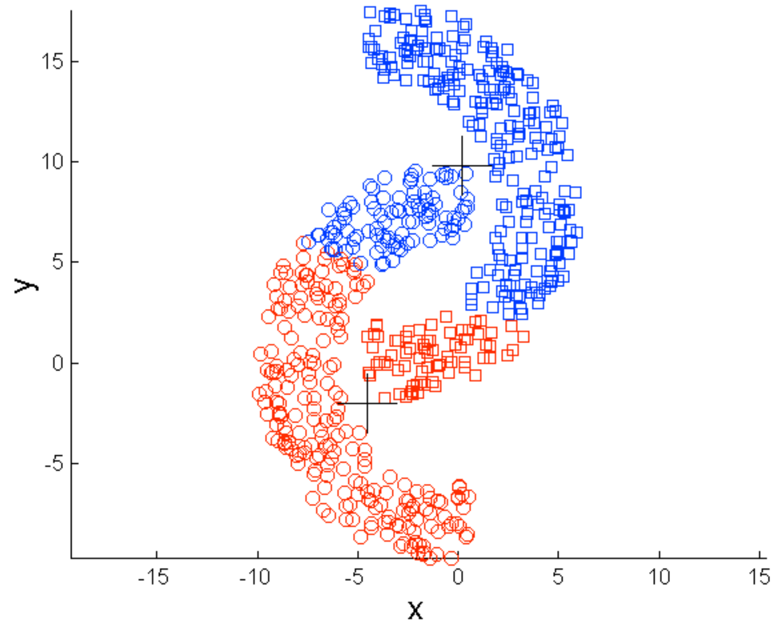


K-means (3 Clusters)

Limitations of K-means: **Non-globular Shapes**

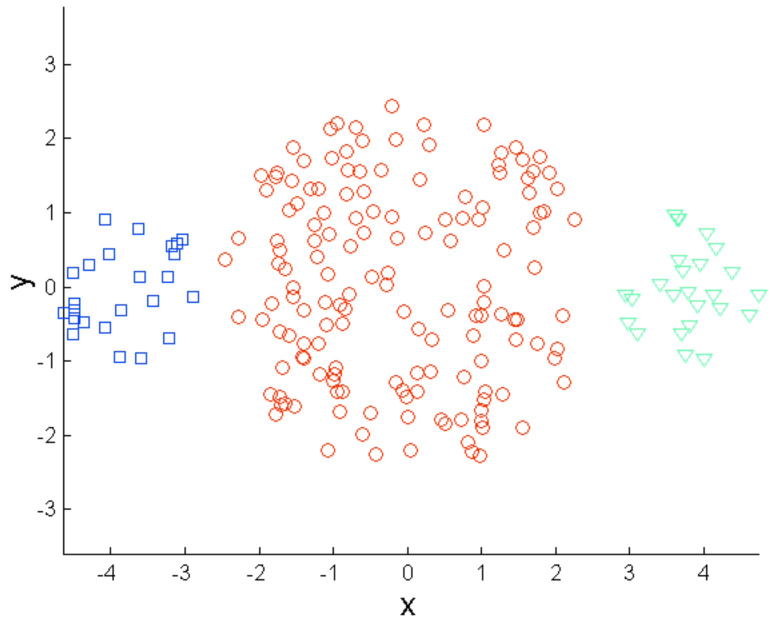


Original Points

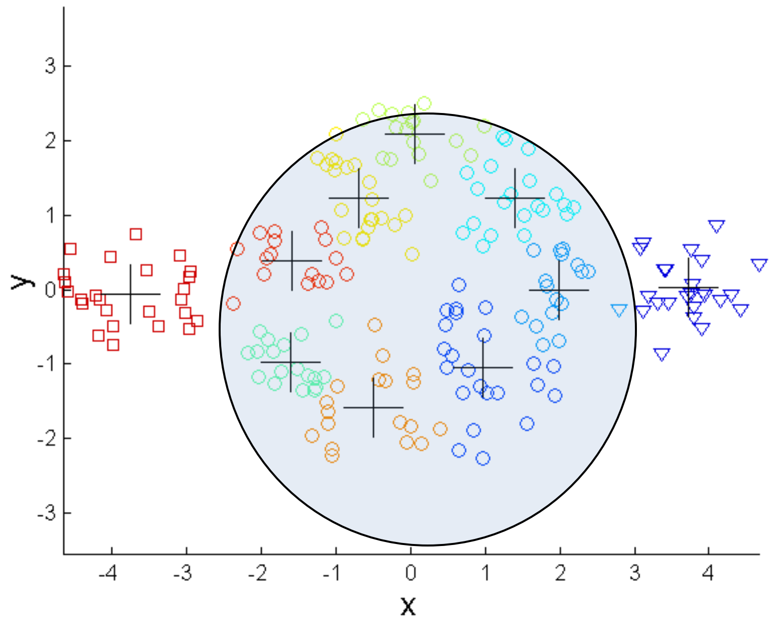


K-means (2 Clusters)

Overcoming K-means Limitations: Differing Sizes



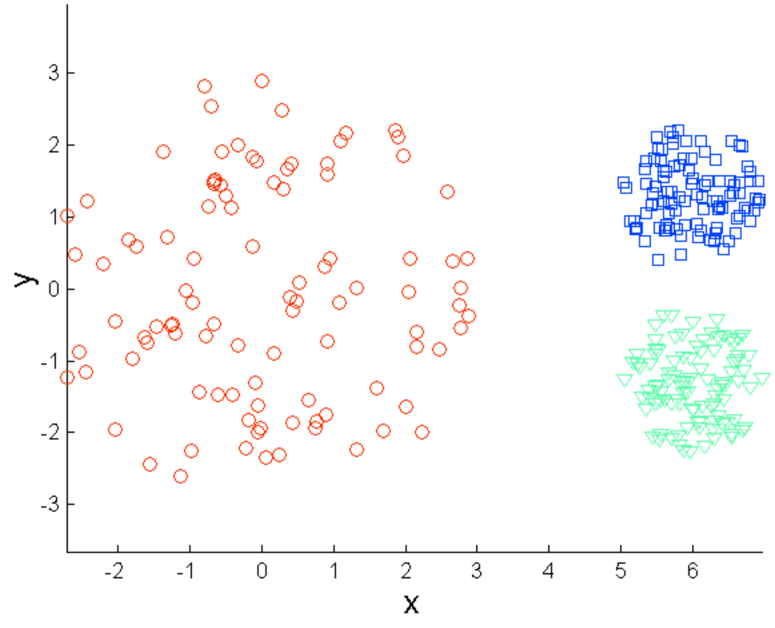
Original Points



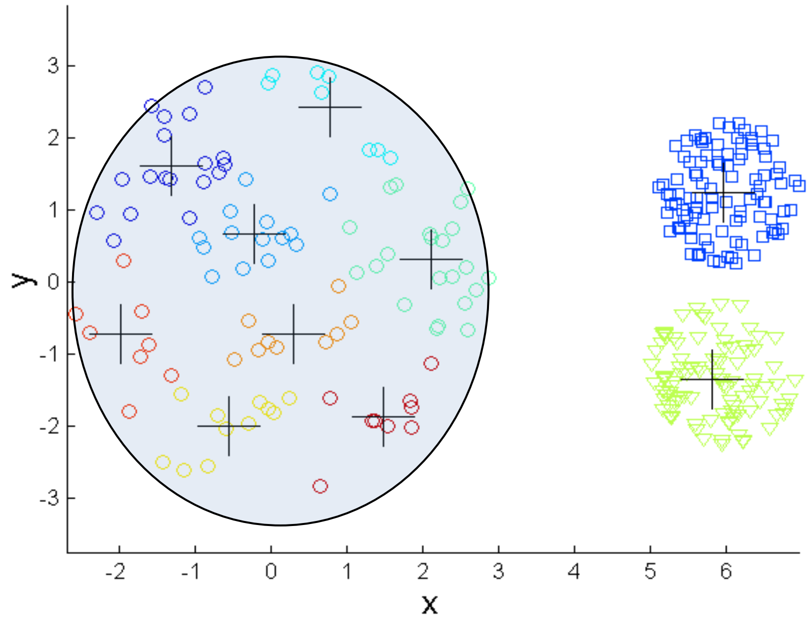
K-means Clusters

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

Overcoming K-means Limitations: Differing Density



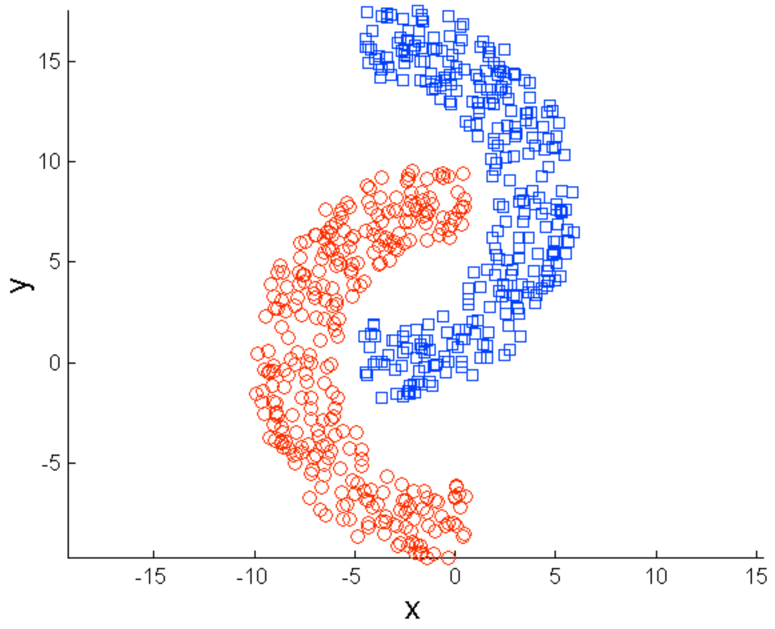
Original Points



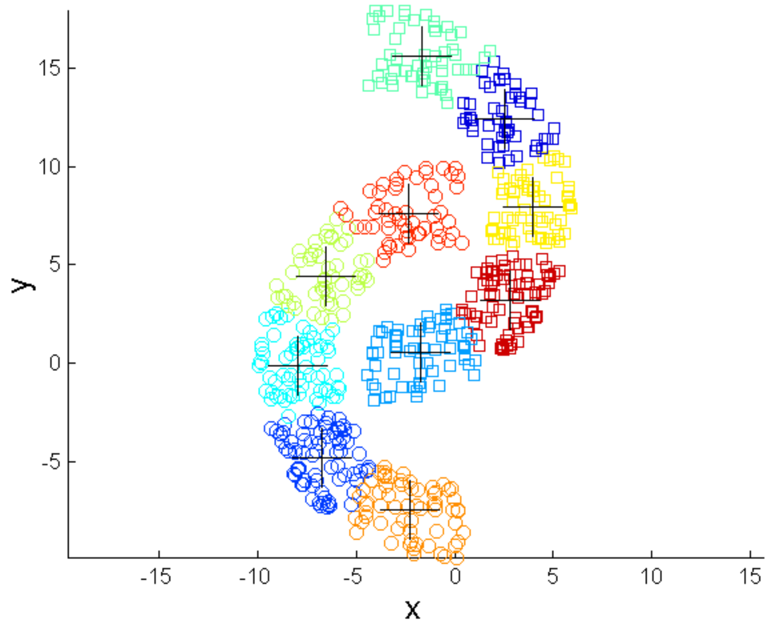
K-means Clusters

Merge the clusters with similar density, and not far away from each others

Overcoming K-means Limitations: Non-globular Shapes



Original Points



K-means Clusters

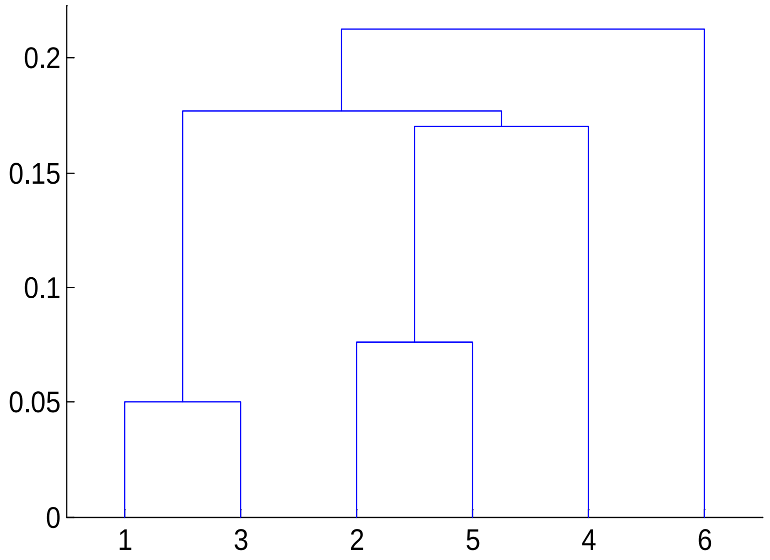
merge the clusters based on contiguity

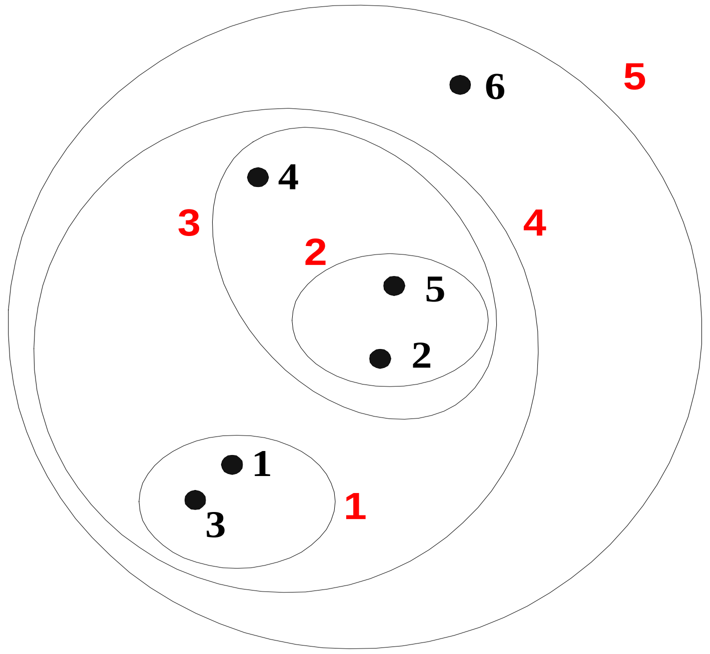
Outline

- Definition
- K-means Clustering
- Hierarchical Clustering
- DBSCAN
- Clustering Evaluation

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

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

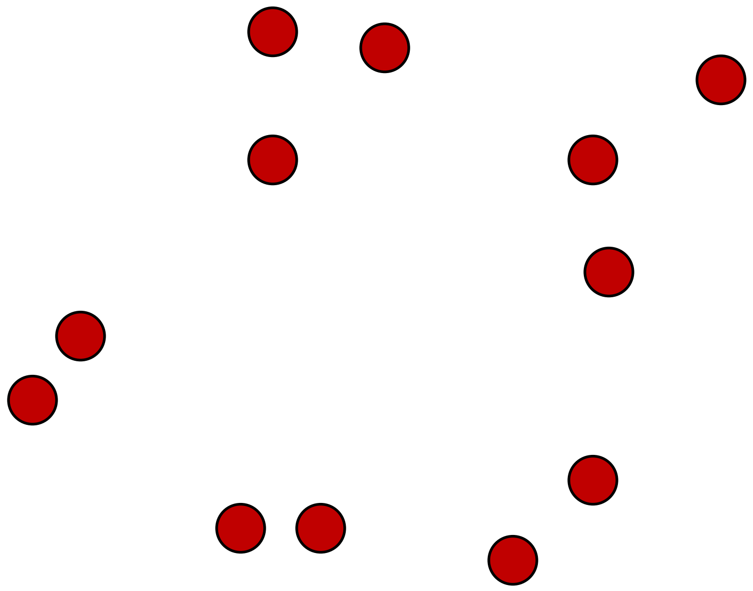
Merge: always merge closest ones

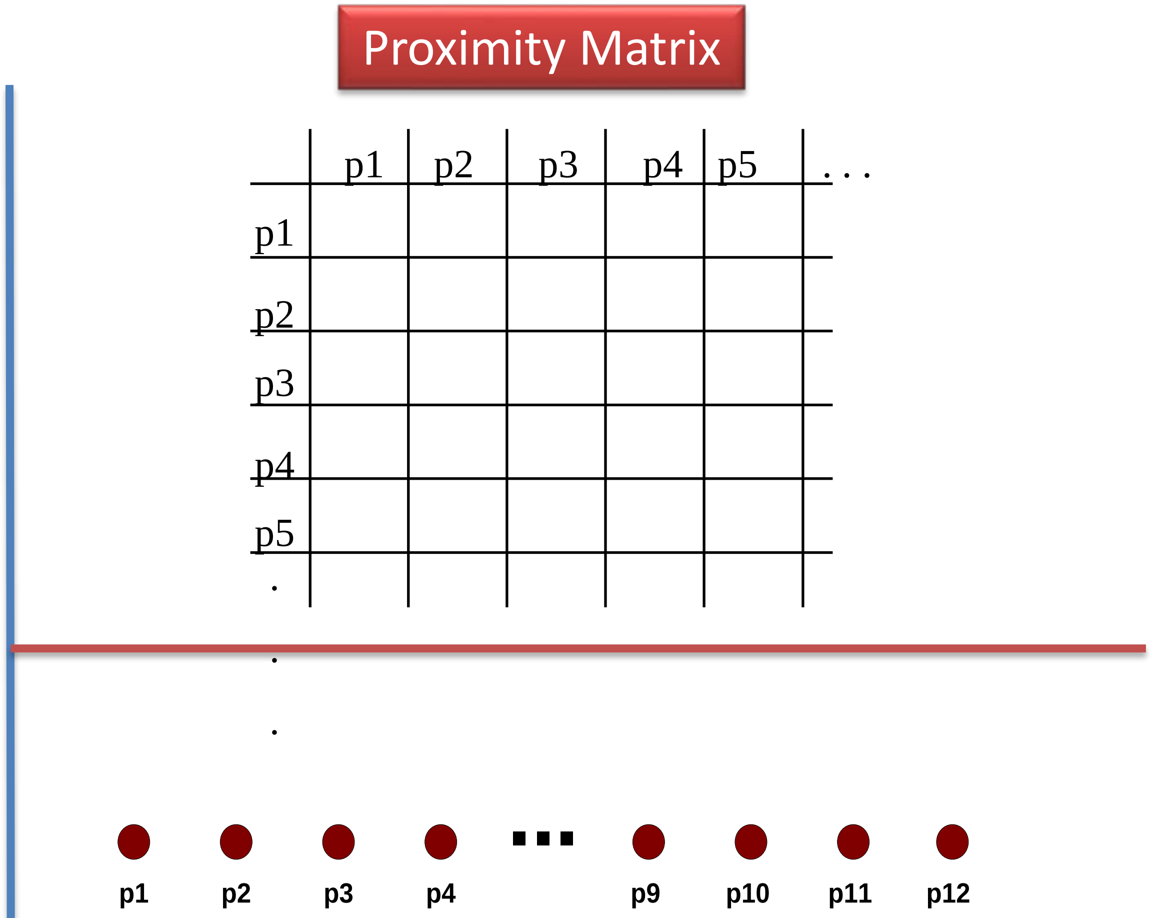
Update: there are different definitions on how to compute. Could confuse ppl when use *max* distance

- 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

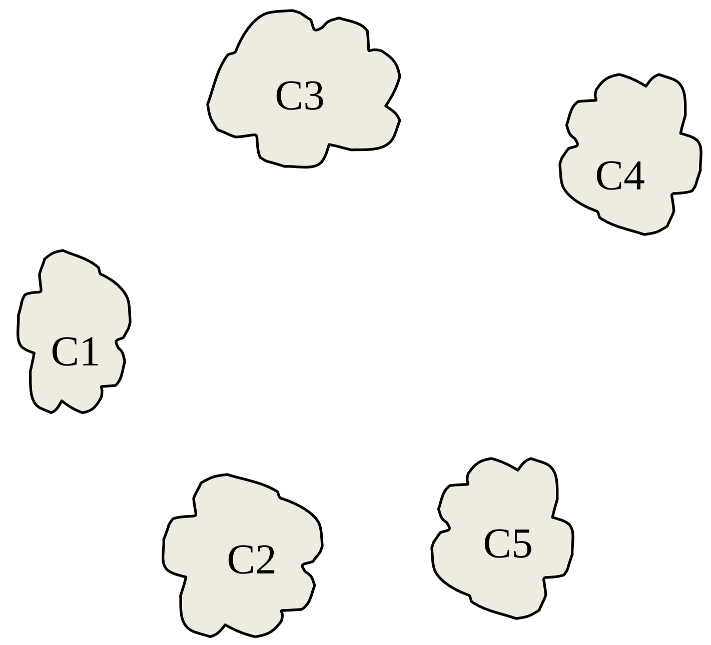
- Start with clusters of individual points and a proximity matrix (similarity or distance)





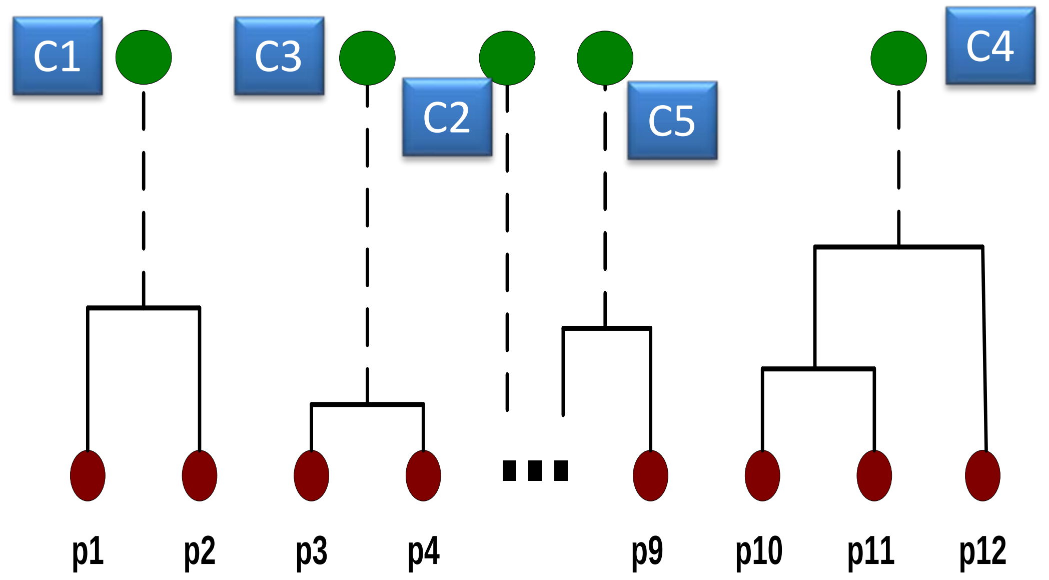
Intermediate Situation

- After some merging steps, we have some clusters



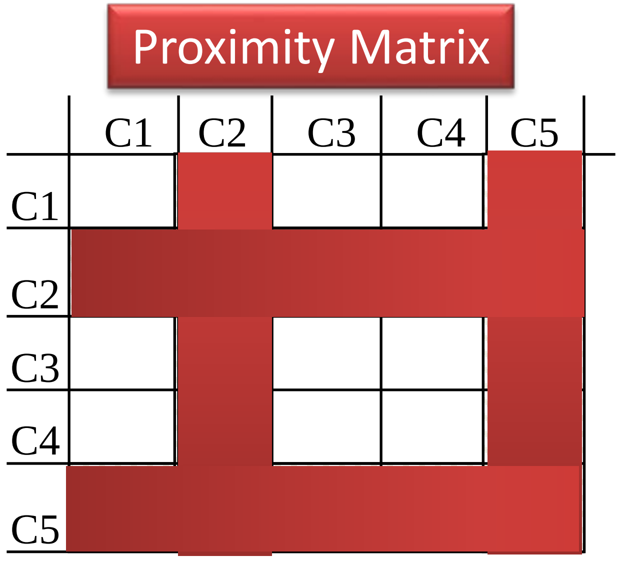
Proximity Matrix

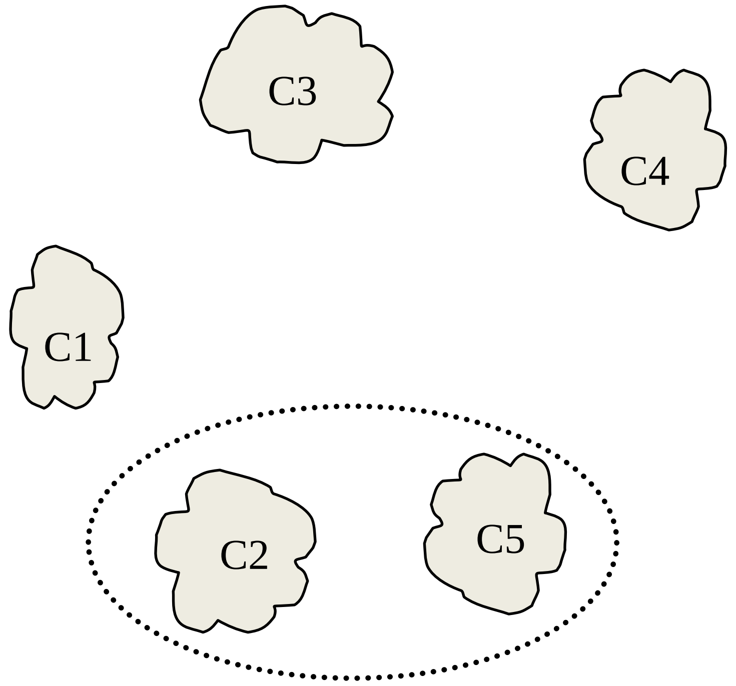
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

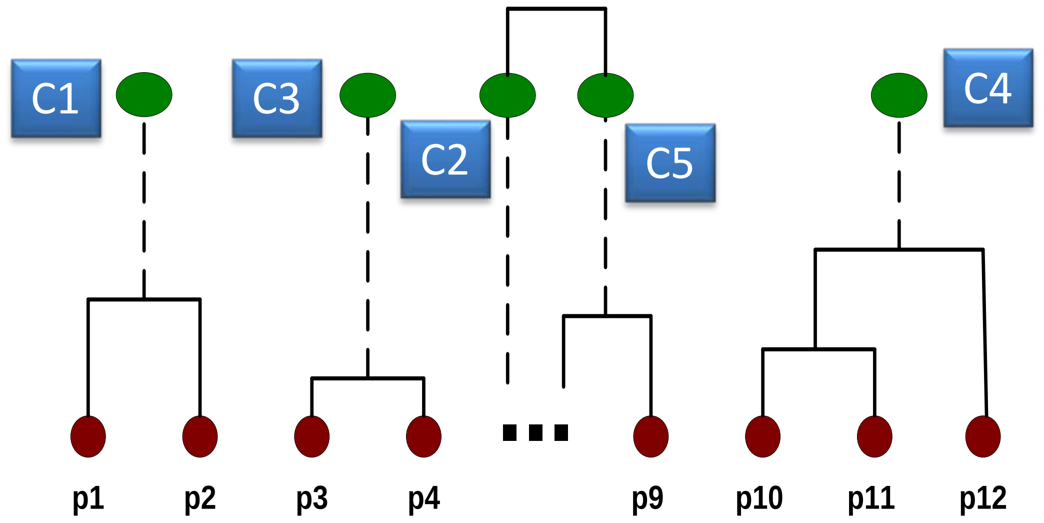


Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) into one (say C2 U C5) and update the proximity matrix .
- What are the similarities/distances between the newly formed cluster C2 U C5 to other clusters (C1, C3, C4)?

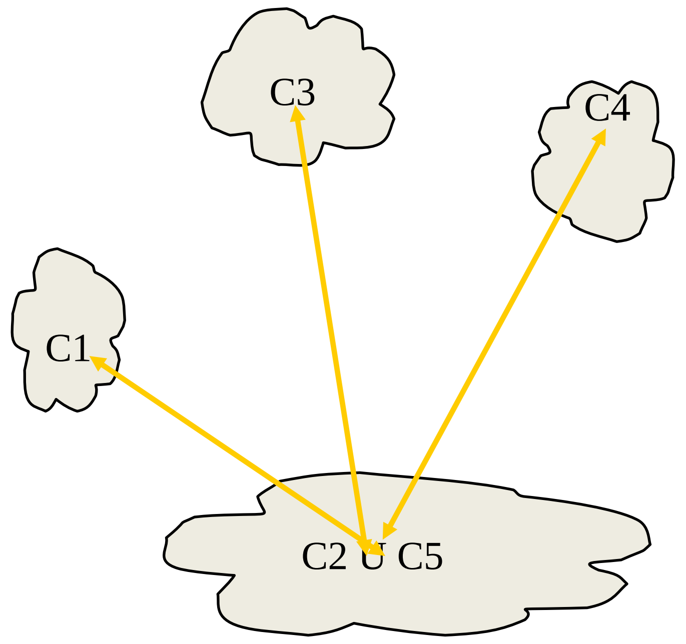






After Merging

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

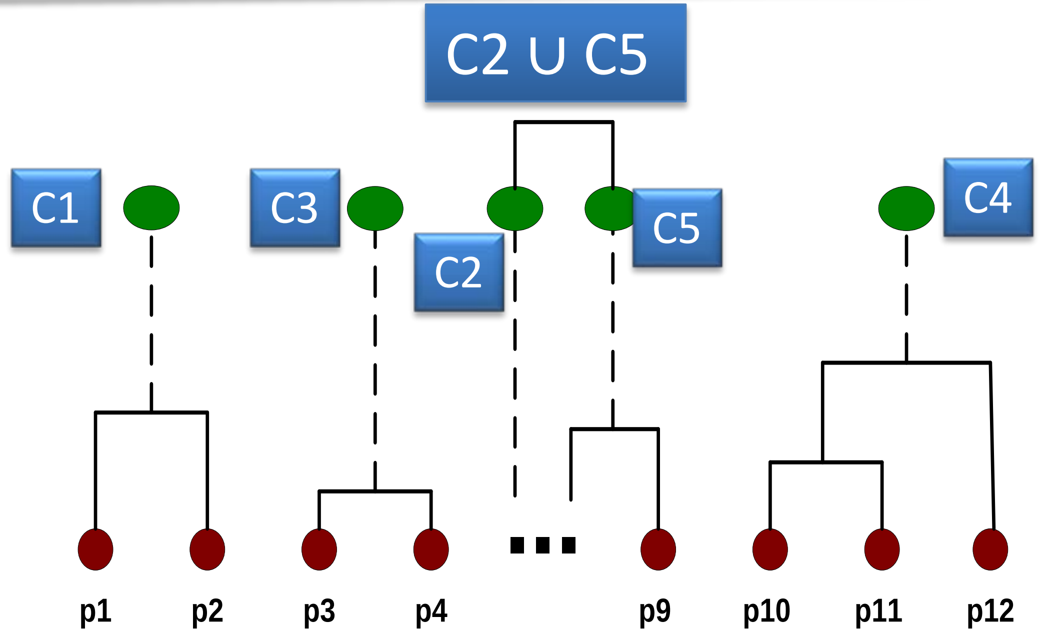


Proximity Matrix

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

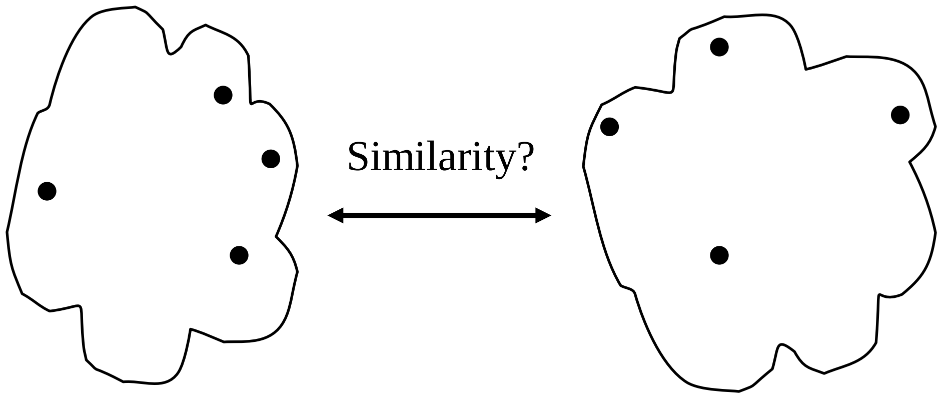
What is the proximity between C2 U C5 and C1?

Similarly between C2 UC5 to C3 and C4?



Key issue: How to Define Inter-Cluster Similarity

Each cluster has 1 or more points



- MIN (single)
- MAX (complete)
- Group Average (average)
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

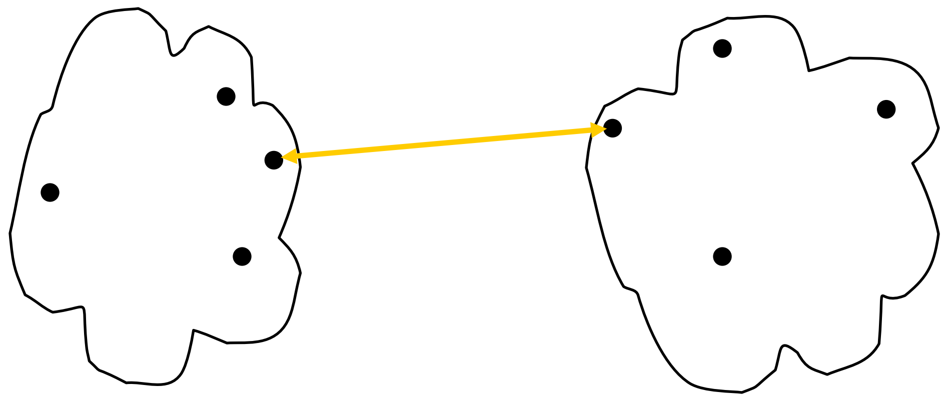
Proximity Matrix

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

.

.

How to Define Inter-Cluster Similarity

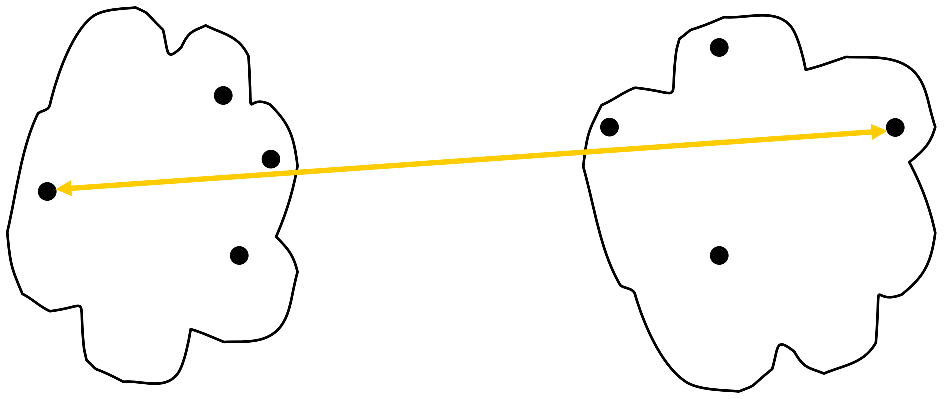


- MIN (among all cross-cluster pairs)
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Proximity Matrix

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

How to Define Inter-Cluster Similarity

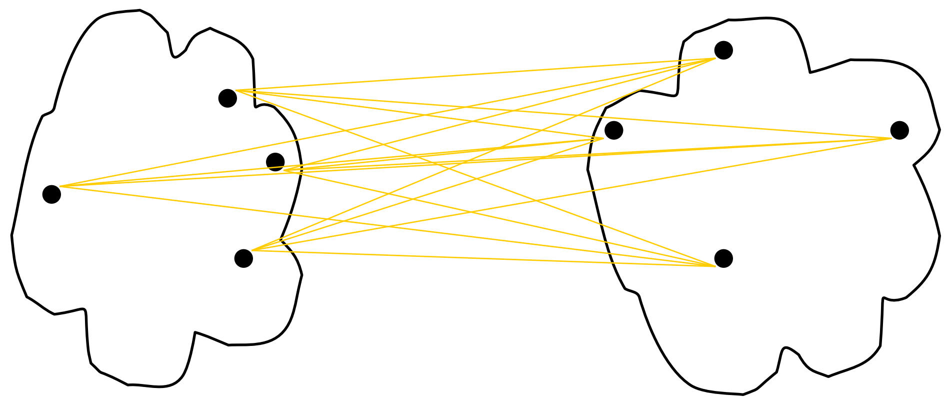


Proximity Matrix

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

- MIN
- MAX (among all cross-cluster pairs)
- 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 (among all cross-cluster pairs)
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

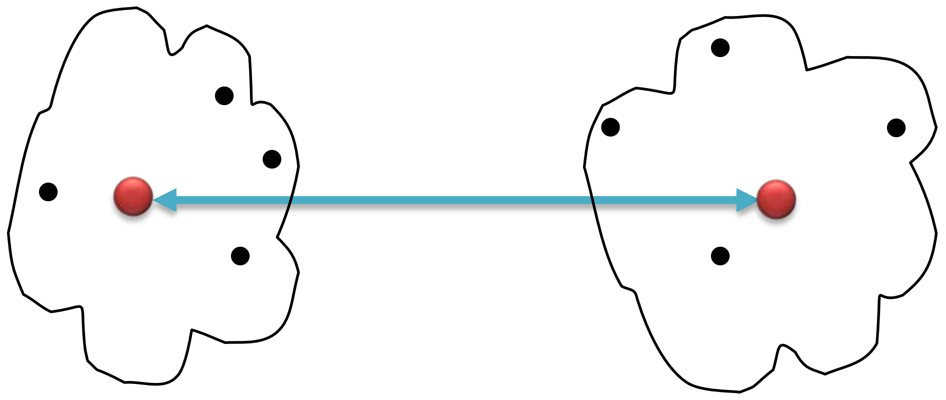
Proximity Matrix

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

.

.

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

Proximity Matrix

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

Ward's method

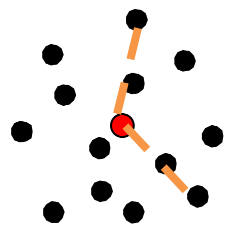
Ward's method is a criterion applied in hierarchical cluster analysis.

Ward suggested a criterion for choosing the pair of clusters to merge at each step based on *the optimal value of an objective function* which could be "*any function that reflects the investigator's purpose*".

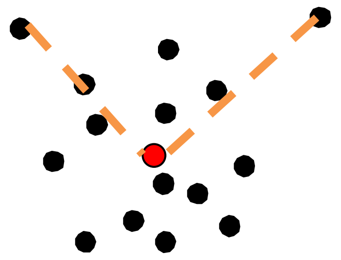
For illustration, Ward used the example where an objective function SSE *sum of squared errors* (*error sum of squares, residual sum of squares)* that minimizes the total within-cluster variance. It is called *minimum variance method*.

https://en.wikipedia.org/wiki/Ward%27s_method

Ward's method (Cont.)



A Merged cluster with Small within-cluster variance



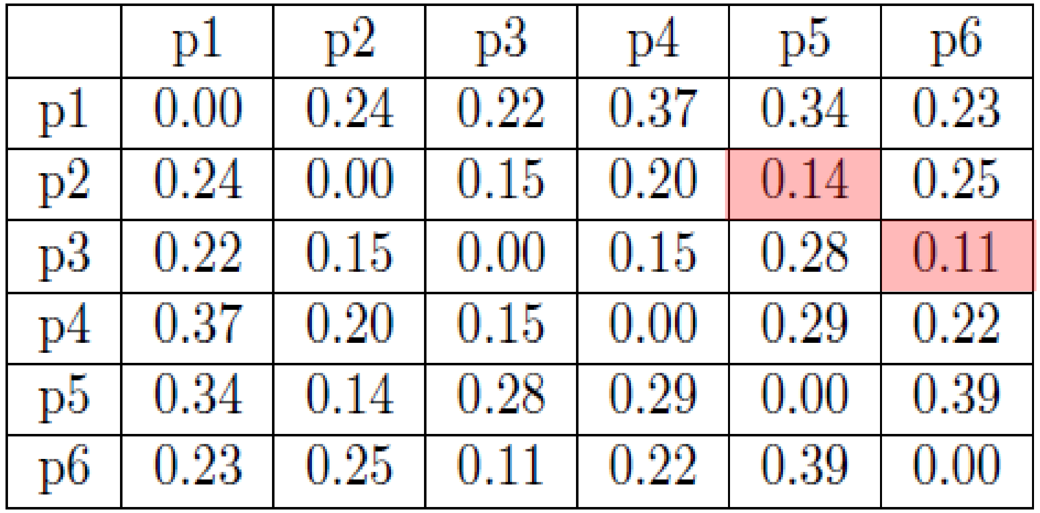
A Merged cluster with Big within-cluster variance

- Variance is defined as the average of the squared differences from the mean. For our case, mean is cluster center;
- We will merge cluster pairs and punish a potentially merged cluster with points having bigger distances to its cluster center, thus having bigger variance).
- Particularly, at each merging step, it finds the pair of clusters that leads to *minimum increase in total within-cluster variance after merging* (more compact cluster).

https://en.wikipedia.org/wiki/Ward%27s_method

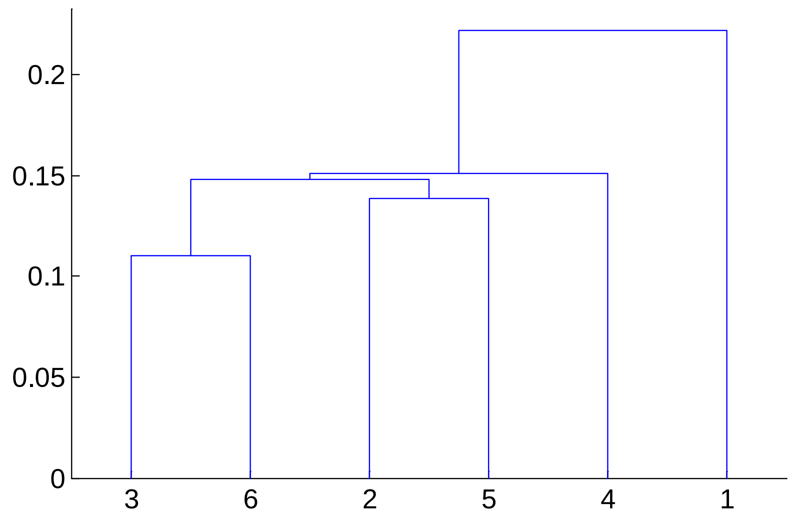
Cluster Similarity: MIN or Single Link

- 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.

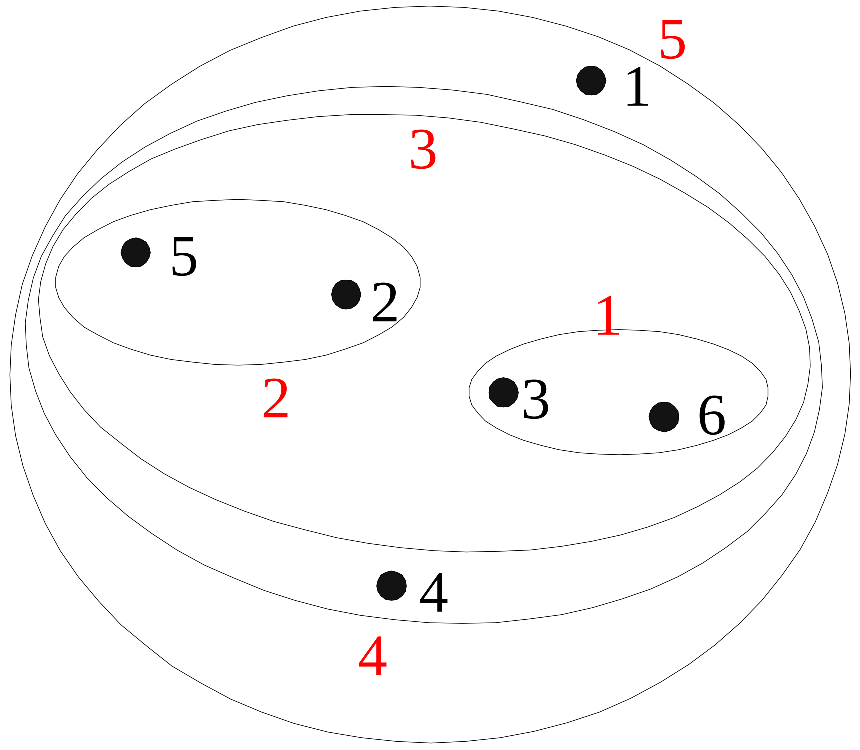


	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Table 8.4. Euclidean distance matrix for 6 points.



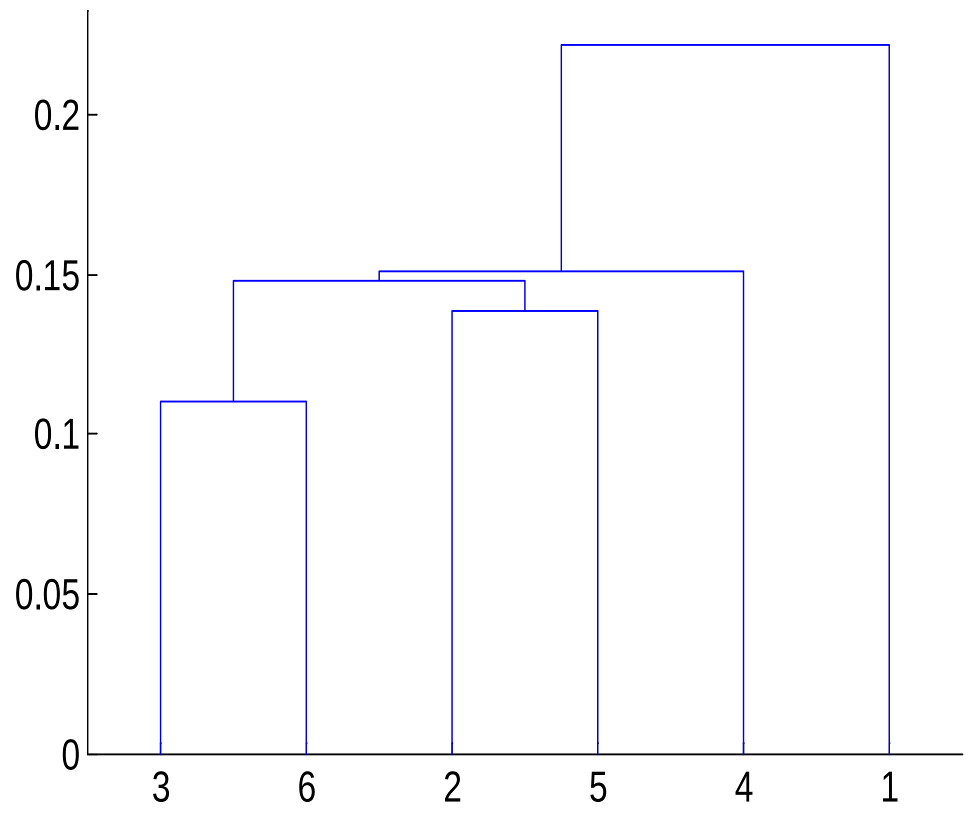
Hierarchical Clustering: MIN



	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 8.4. Euclidean distance matrix for 6 points.

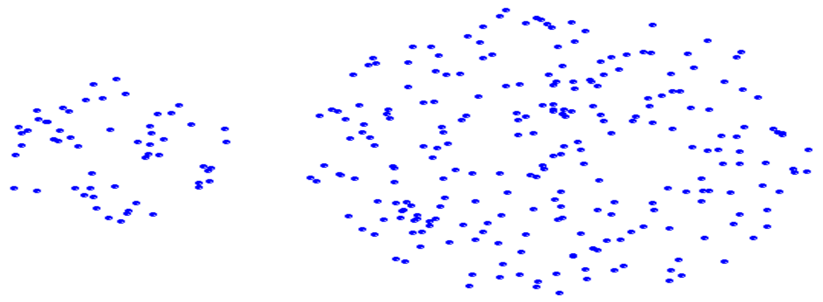
Single Link Clustering

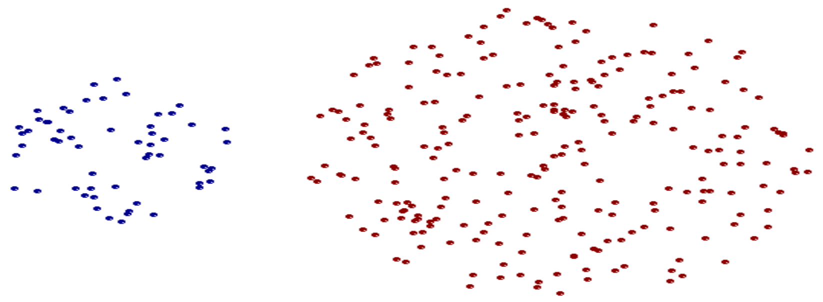


Single Link Dendrogram

Strength of MIN

- Can handle non-elliptical shapes

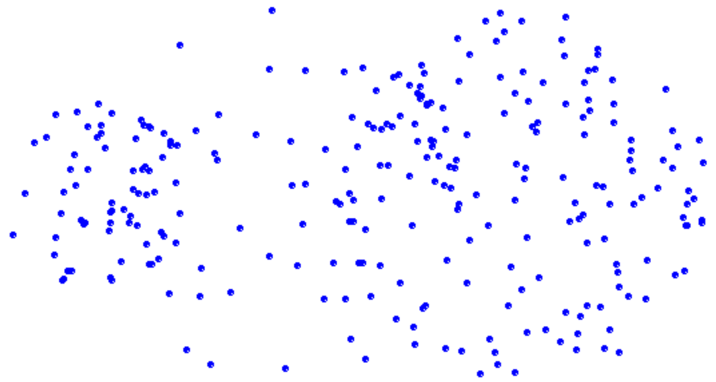

Original Points

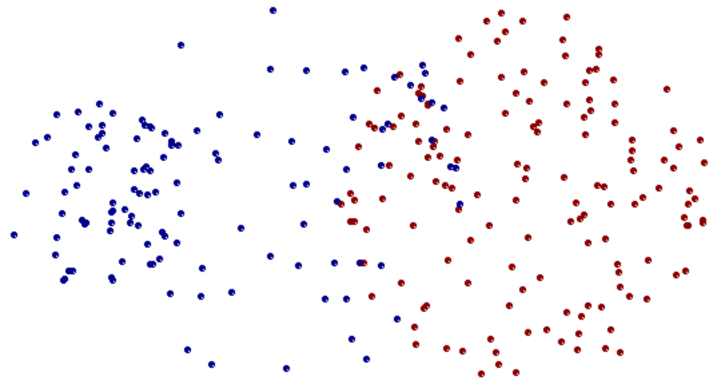

Two Clusters

The algo likely to merge the points within same clusters if they are clearly separated

Limitations of MIN

- Sensitive to noise and outliers: distance just based on one pair of nearest points


Original Points

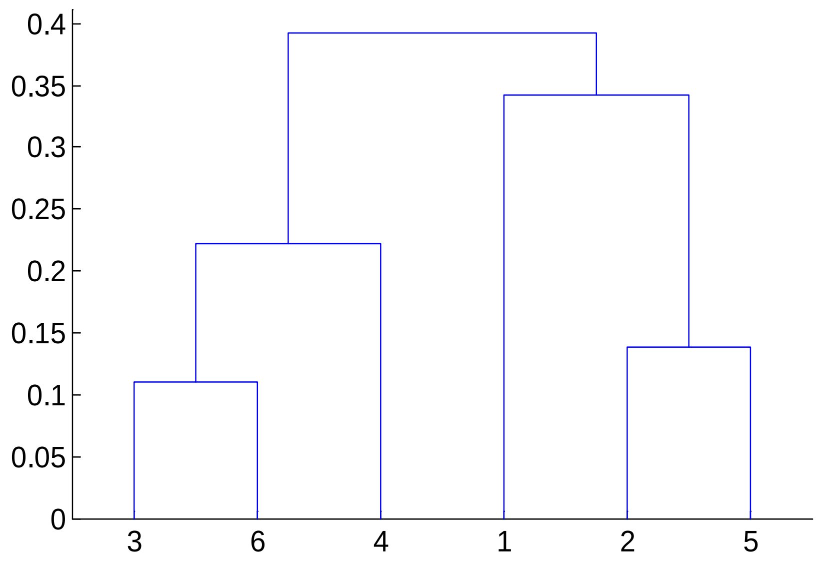

Two Clusters

Cluster Similarity: MAX or Complete Linkage

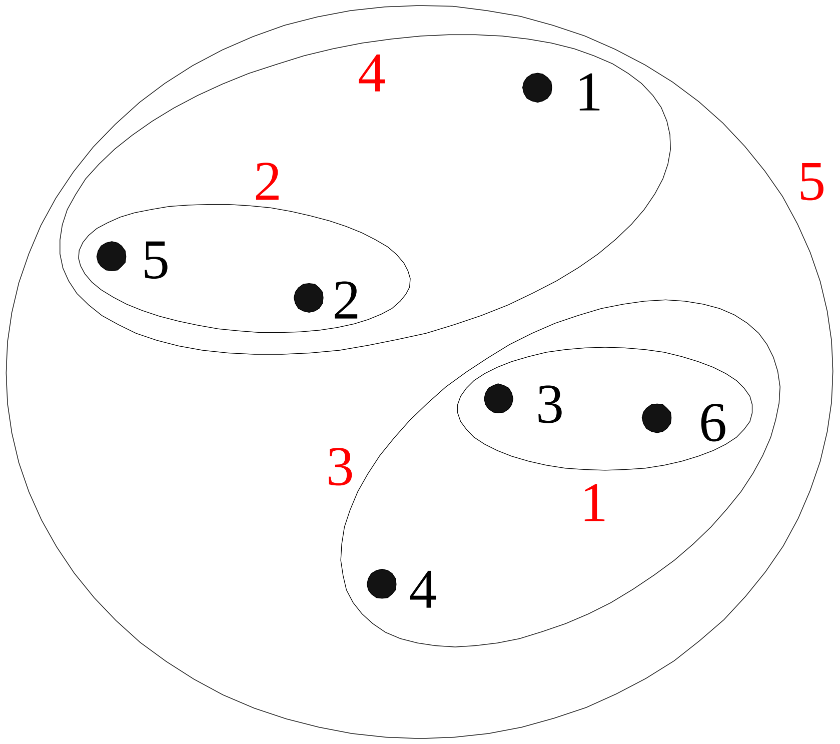
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by all pairs of points in the two clusters

	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 8.4. Euclidean distance matrix for 6 points.



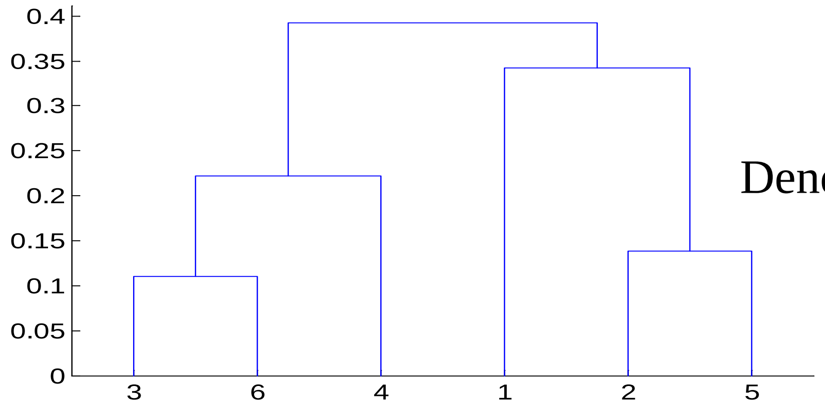
Hierarchical Clustering: MAX



Nested Clusters

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

Table 8.4. Euclidean distance matrix for 6 points.

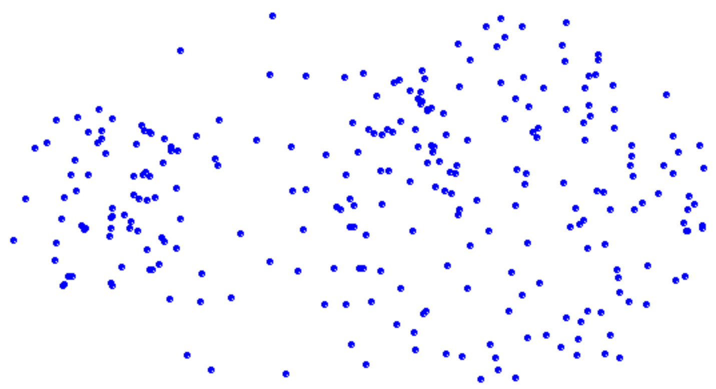


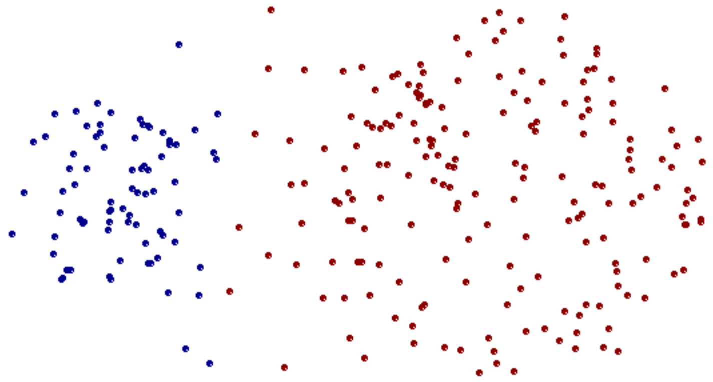
Dendrogram

Distance btw cluster C2 ∪C5 and C3 ∪C6 is biggest according to MAX, i.e. distance btw 5 and 6.

Note we **still** merge 2 most similar clusters each time during clustering. However, we define the distance between clusters based on MAX.

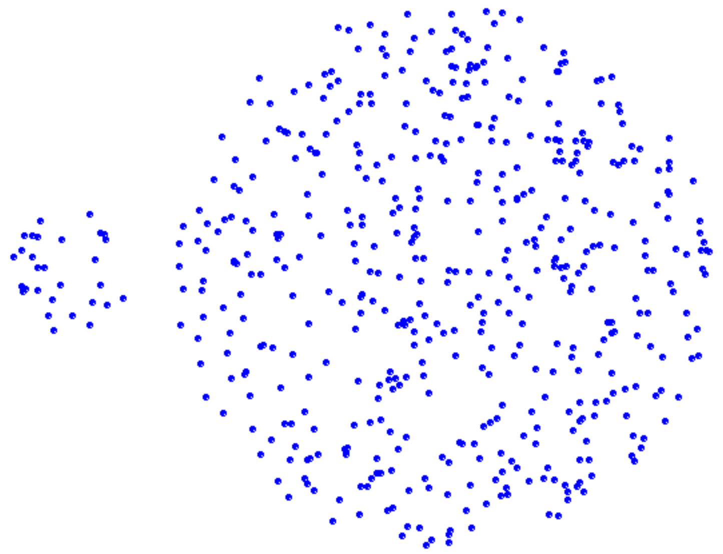
Strength of MAX

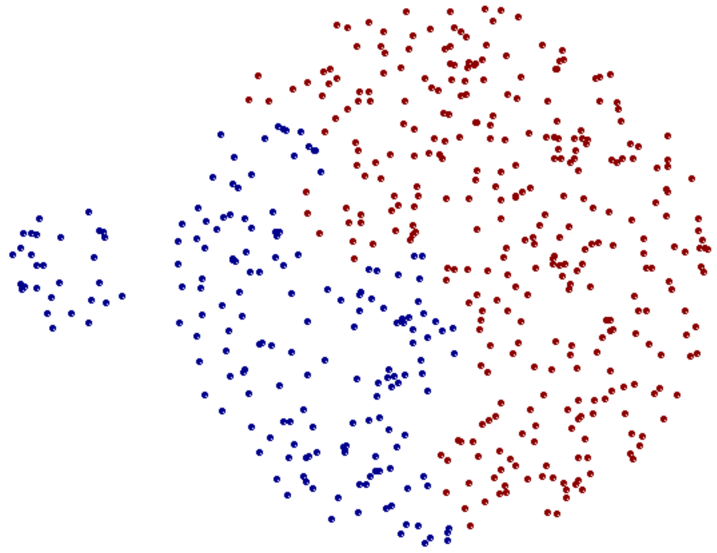

Original Points


Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX


Original Points

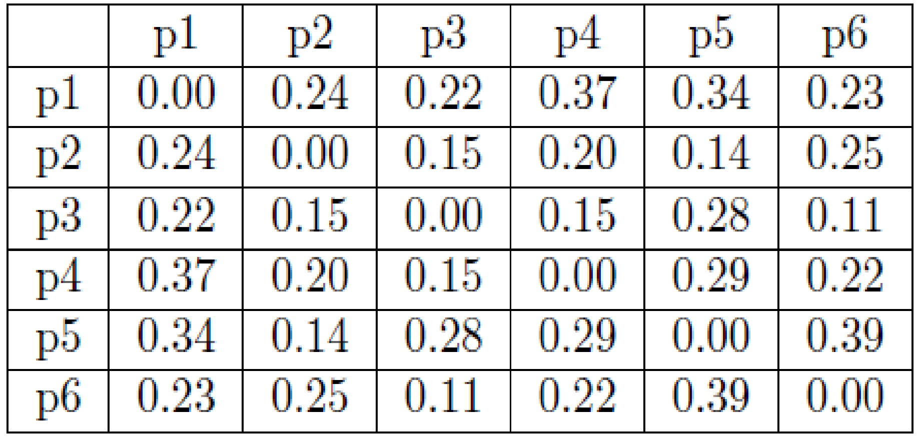

Two Clusters

- Tends to break large clusters (too big=> far away=> break)
- Biased towards globular clusters

Cluster Similarity: Group Average

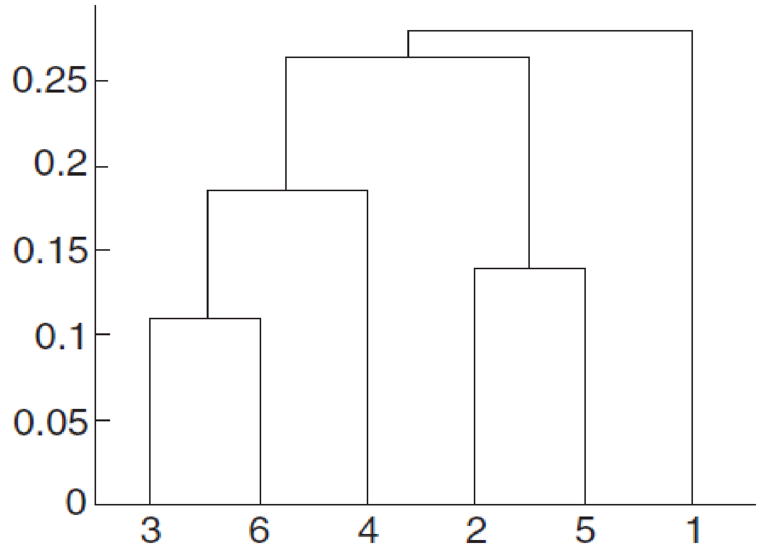
- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

$$proximity(C_i, C_j) = \frac{\sum_{p_{i \in C_i, P_j C_j}} proximity(p_i, p_j)}{|C_i| * |C_j|}$$

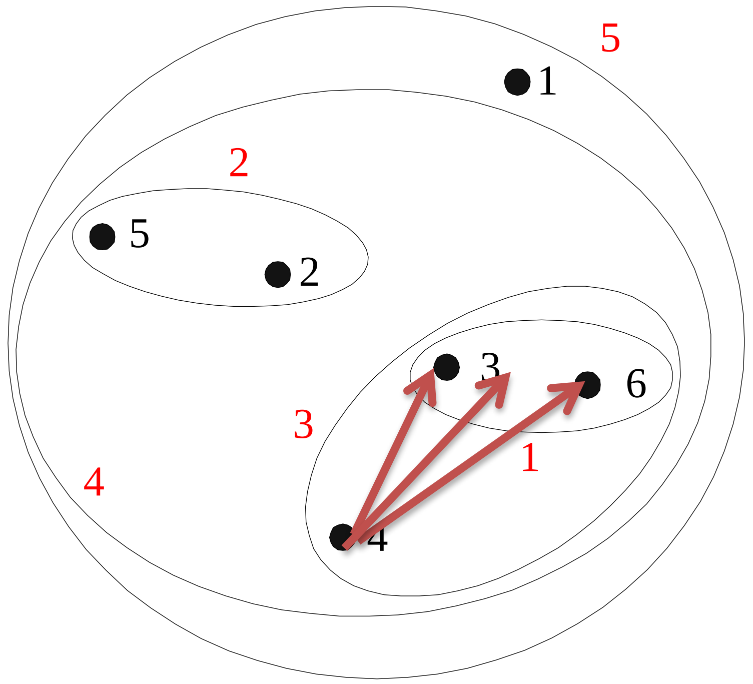


	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

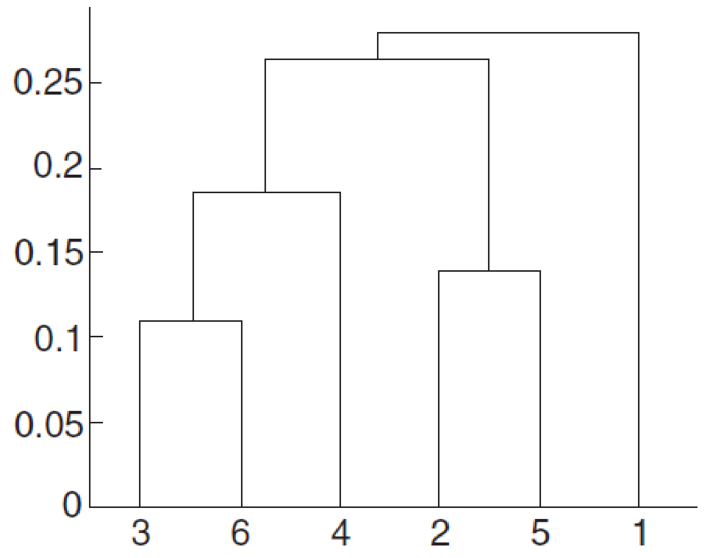
Table 8.4. Euclidean distance matrix for 6 points.



Hierarchical Clustering: Group Average



Group Average Clustering



Group Average Dendrogram

Hierarchical Clustering: Group Average

Compromise between Single and Complete Link

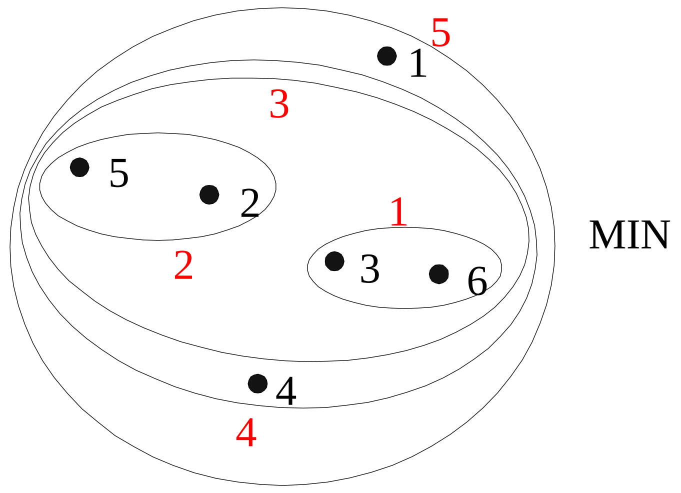
Strengths

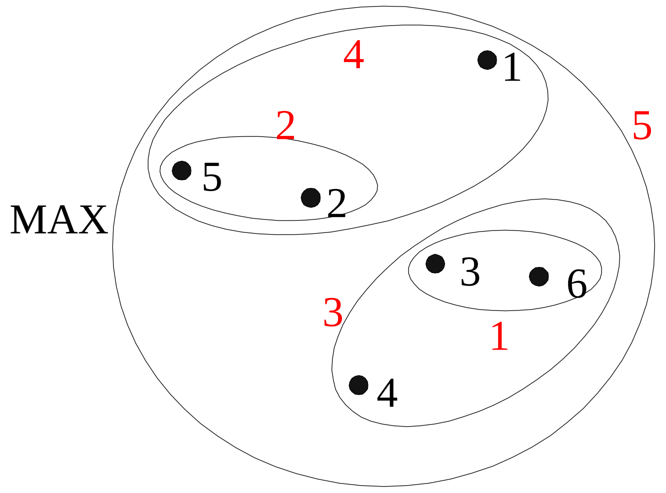
- Less susceptible to noise and outliers

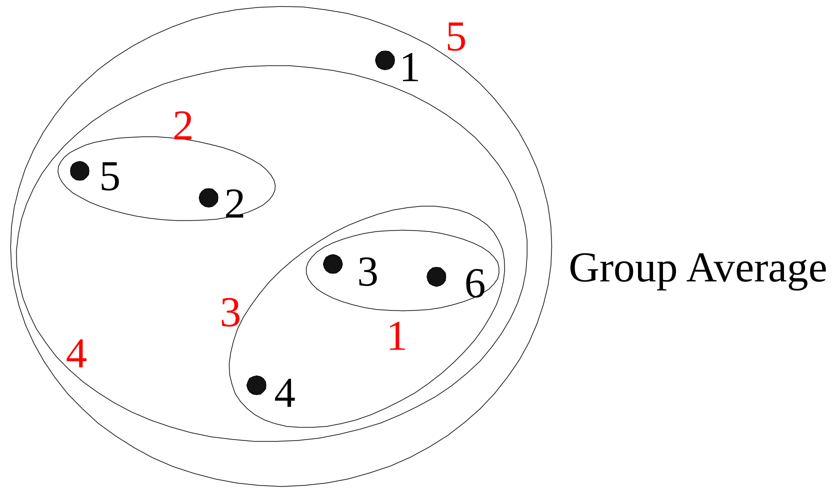
Limitations

- Biased towards globular clusters

Hierarchical Clustering: Comparison







Outline

- Definition
- K-means Clustering
- Hierarchical Clustering
- DBSCAN
- Clustering Evaluation

Hierarchical Clustering: Problems and Limitations

Once a decision is made to combine two clusters, it cannot be undone

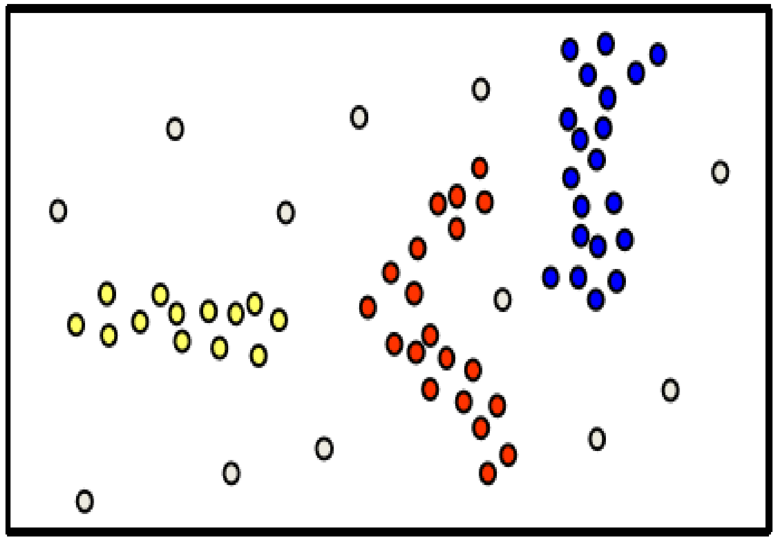
No objective function is directly minimized

Different schemes have problems with one or more of the following

- Sensitivity to noise and outliers
- Difficulty handling different sized clusters
- Breaking large clusters

Density-based Clustering

- Basic idea
  - Clusters are dense regions in the data space, separated by regions of lower object density
  - A cluster is defined as a *maximal set* of density connected points
  - Discovers clusters of *arbitrary shape*
- Method
  - DBSCAN: Density-based spatial clustering of applications with noise

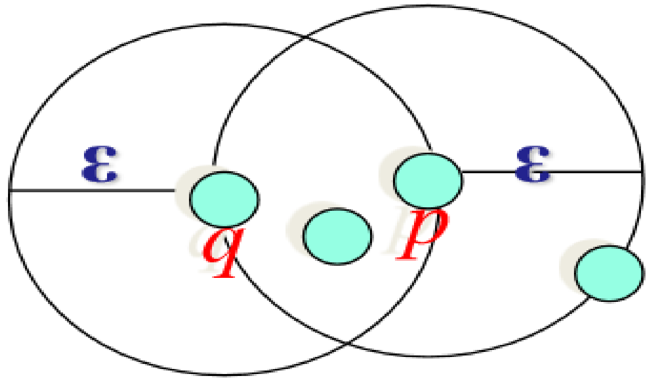


Density Definition

- ε-Neighborhood – Objects within a radius of ε from an object.

$$N_{\varepsilon}(p) : \{q \mid d(p,q) \le \varepsilon\}$$

- "High density" - ε-Neighborhood of an object contains at least MinPts of objects.



ε-Neighborhood of *p*
ε-Neighborhood of *q*

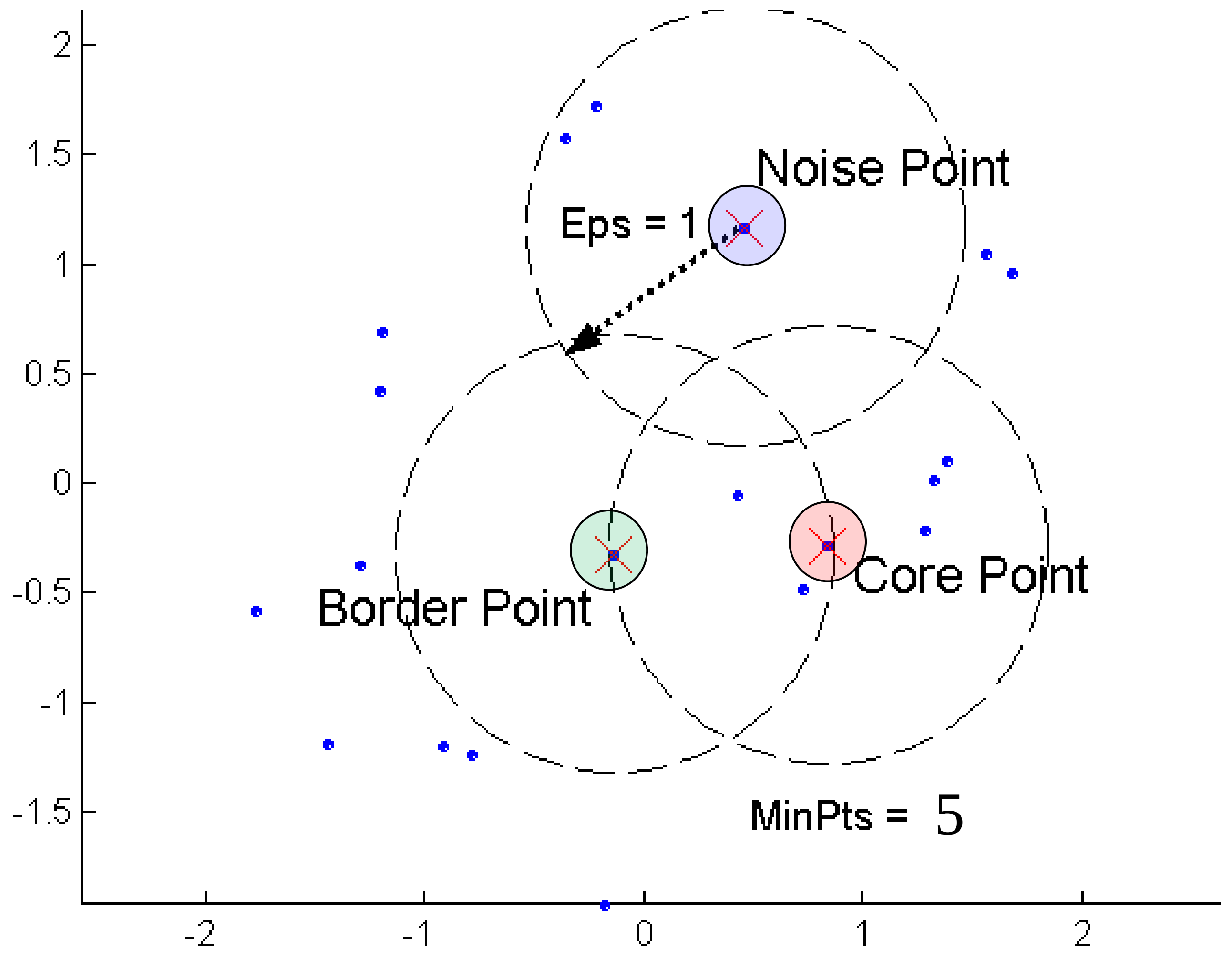
Density of p is "high" (MinPt $= 4$)

Density of q is "low" (MinPts $= 4$)

DBSCAN

- DBSCAN is a density-based algorithm.
- Given ε (Eps) and MinPts, categorize all the objects into three exclusive groups.
  - 1) A point is a core point, if it has more than a specified number of points (>=MinPts) within radius ε. These are points that are at the **interior** of a cluster. Why?
  - 2) A border point has fewer than MinPts within ε (< MinPts), but is in the *neighborhood of a core point.*
  - 3) A noise point is any point that is not a core point or a border point.

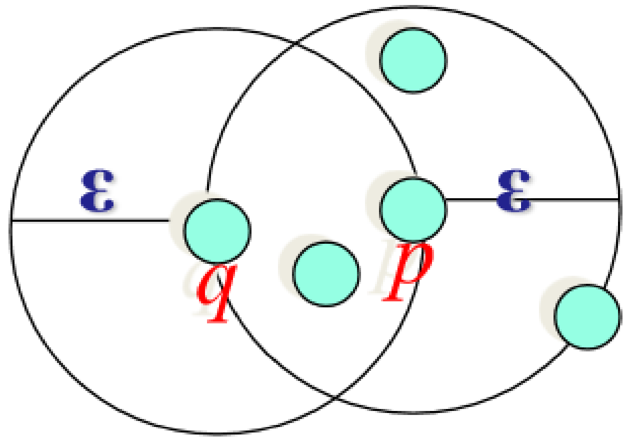
DBSCAN: Core, Border, and Noise Points



Density-reachability

- Directly density-reachable
  - An object *q* is directly density-reachable from object *p* if *p* is a core object and *q* is in p's ε-neighborhood.

p->q (core to other core or boarder points)



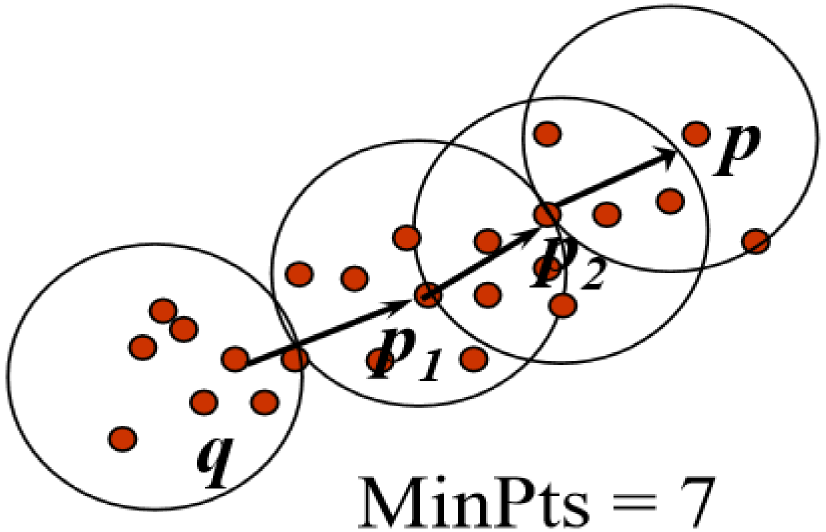
- *q* is directly density-reachable from *p*
- *p* is not directly density-reachable from *q* (from core point to its border points)
- Density-reachability is asymmetric

MinPts = 4

Density-reachability

- Density-Reachable (directly and indirectly):
  - A point p is directly density-reachable from p_2
  - p_2 is directly density-reachable from p_1
  - p_1 is directly density-reachable from q
  - $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$ form a chain

p is a border point,
q, p1, p2 are core points

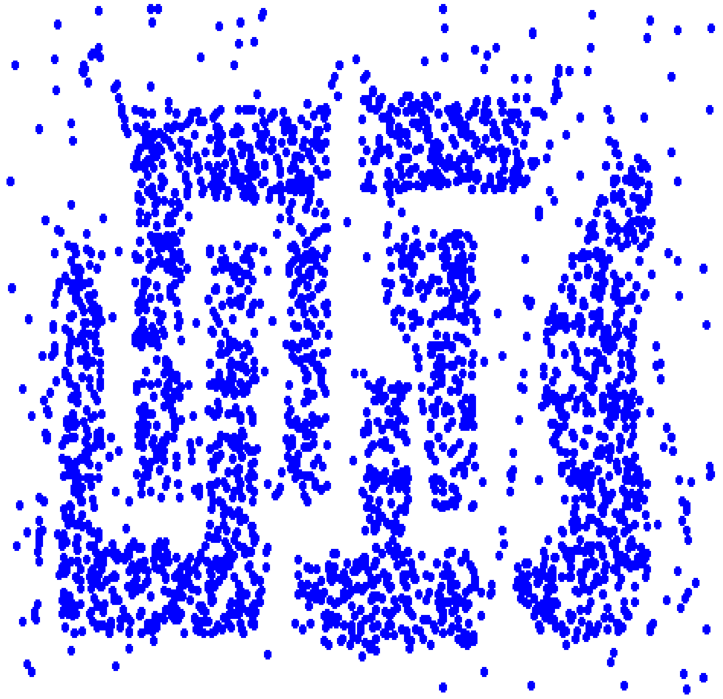


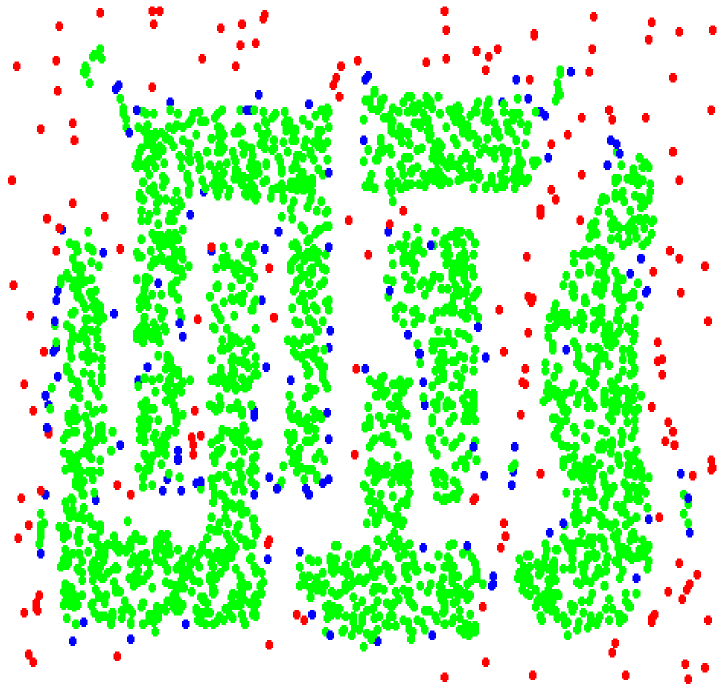
MinPts = 7

- p is (indirectly) density-reachable from q
- q is not density-reachable from p

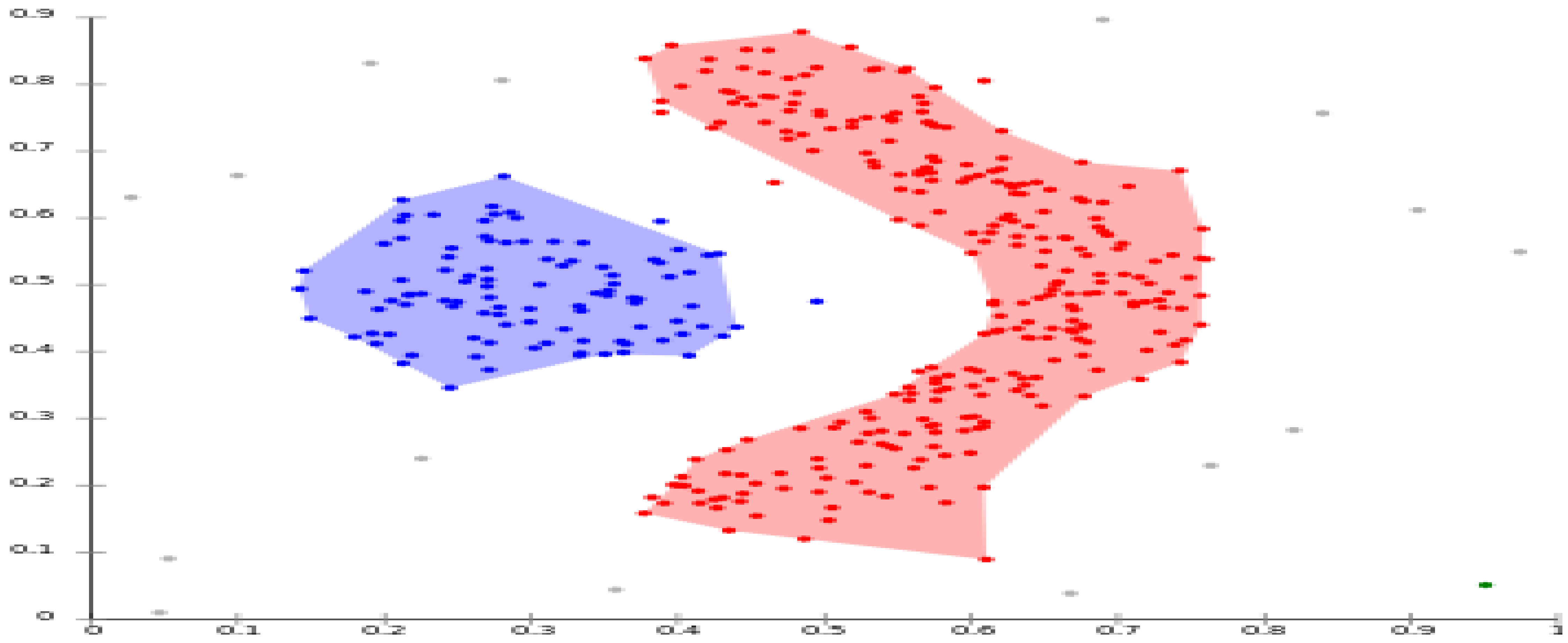
DBSCAN: Core, Border and Noise Points An Example of challenging cases

DBSCAN drops noise points. It expands from core-points to include other core or boarder points.


Original Points

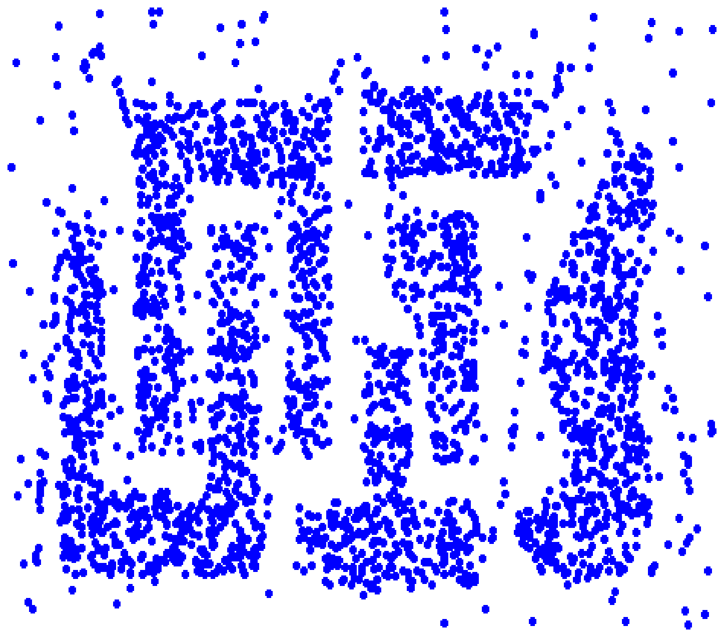

Point types: core, border and noise

Parameters ε (Eps) = 10, MinPts = 4

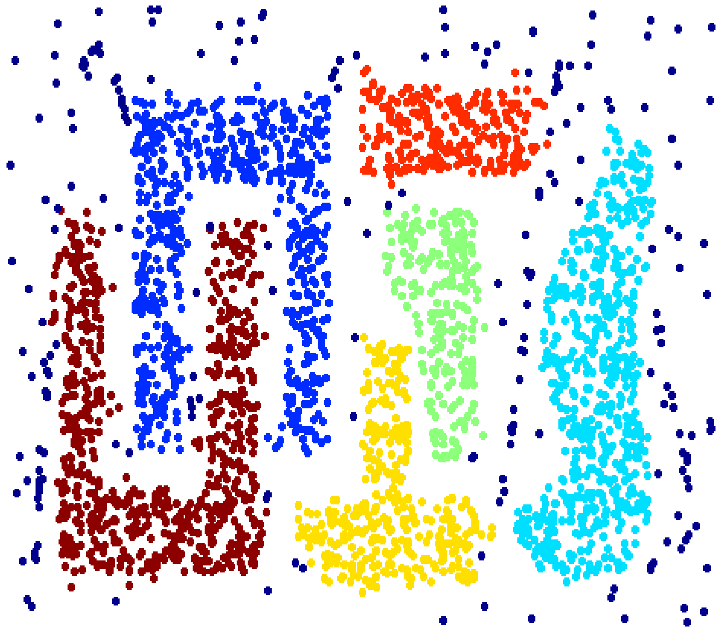


- DBSCAN can find non-linearly separable clusters. This dataset cannot be adequately clustered with k-means

When DBSCAN Works Well



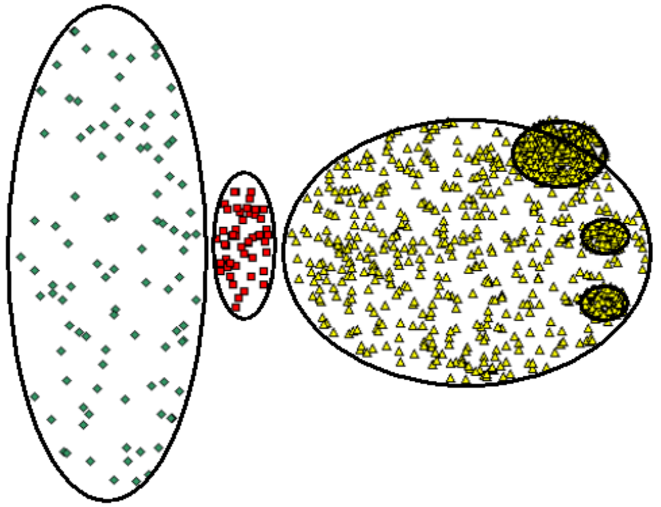
Original Points



Clusters

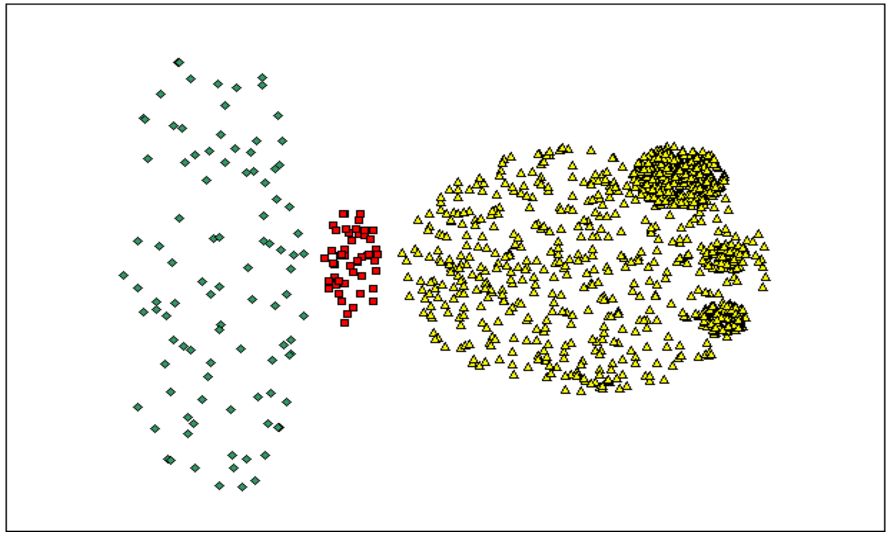
- Resistant to Noise.
- Can handle clusters of *different shapes and sizes*

When DBSCAN Does NOT Work Well

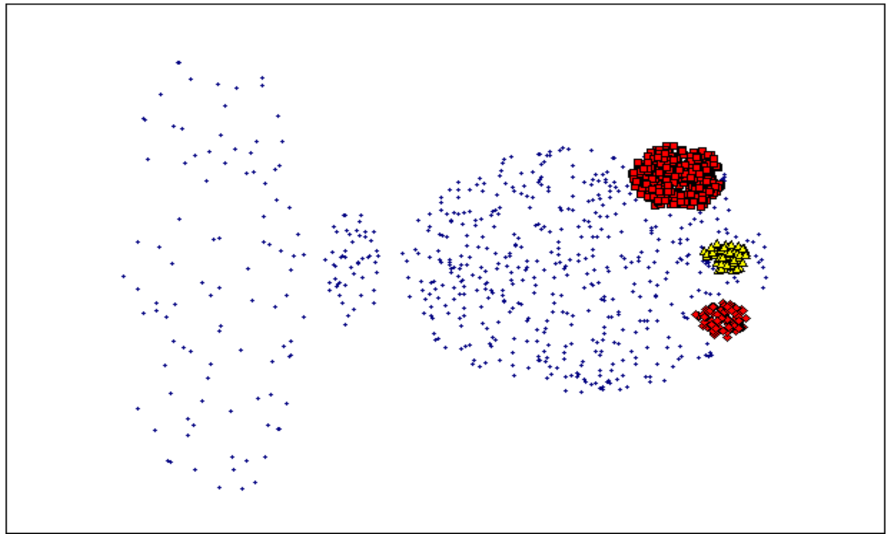


Original Points

Varying densities



Smaller density threshold (MinPts)



Bigger density threshold (MinPts)

OPTICS: Improved version of DBSCAN

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html; https://www.dbs.ifi.lmu.de/Publikationen/Papers/OPTICS.pdf

An important property of many real-data sets is that their intrinsic cluster structure cannot be characterized by *global* density parameters. Very different local densities may be needed to reveal clusters in different regions of the data space. For example, in the data set depicted in Figure 1, it is not possible to detect the clusters A, B, C_1, C_2, and C_3 simultaneously using one global density parameter. A global density-based decomposition would consist only of the clusters A, B, and C, or C_1, C_2, and C_3. In the second case, the objects from A and B are noise.

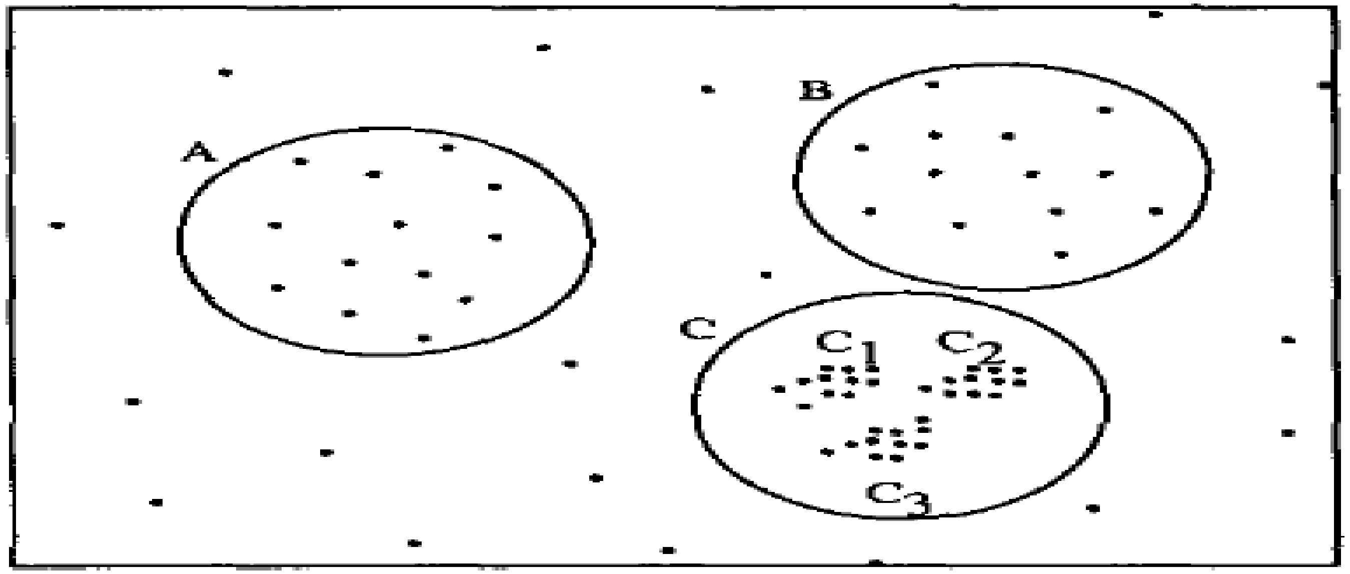


Figure 1. Clusters wrt. different density parameters

The first alternative to detect and analyze such clustering structures is to use a hierarchical clustering algorithm, for instance the single-link method. This alternative, however, has two drawbacks. First, in general it suffers considerably from the single-link effect, i.e. from the fact that clusters which are con-

Outline

- Definition
- K-means Clustering
- Hierarchical Clustering
- DBSCAN
- Clustering Evaluation

Evaluation Metrics

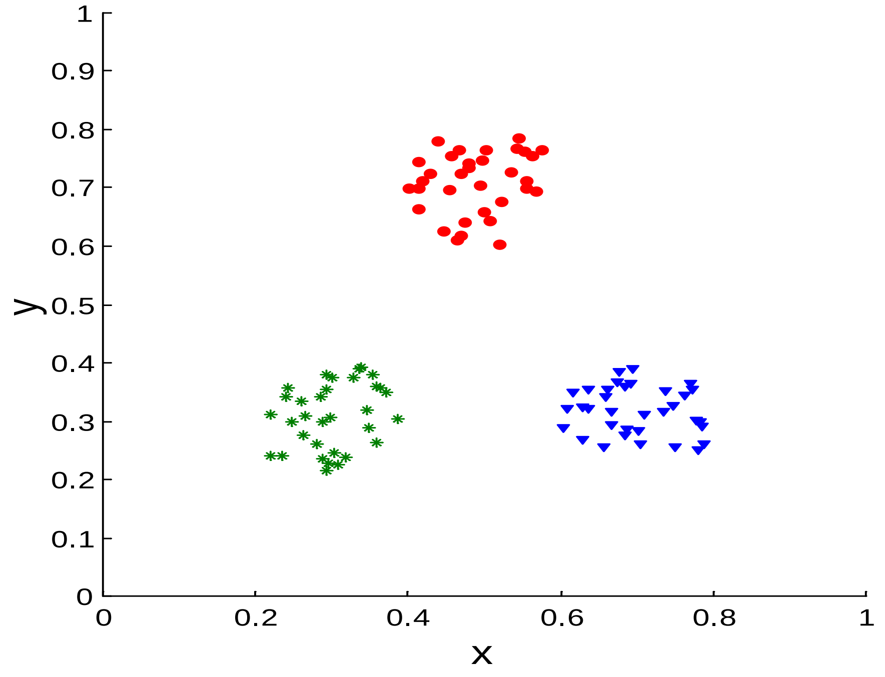
- Sum of Squared Error (SSE) (We already talked about it)
- Correlation
- Human perception
- Silhouette Coefficient
- Cohesion and Separation (in Appendix)
- ……

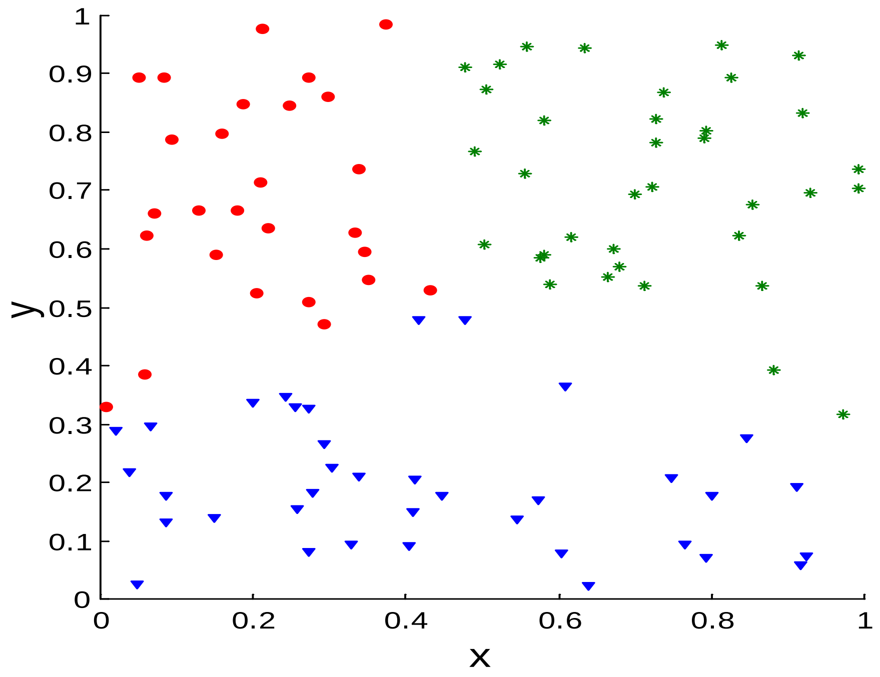
Correlation

- Two matrices
  - Proximity Matrix (data points are close or not)
  - "Incidence" Matrix (summary of clustering results)
    - 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 (transitive; not necessarily close to each other).

Correlation

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





Corr = 0.9235

Stronger correlation

Corr = 0.5810

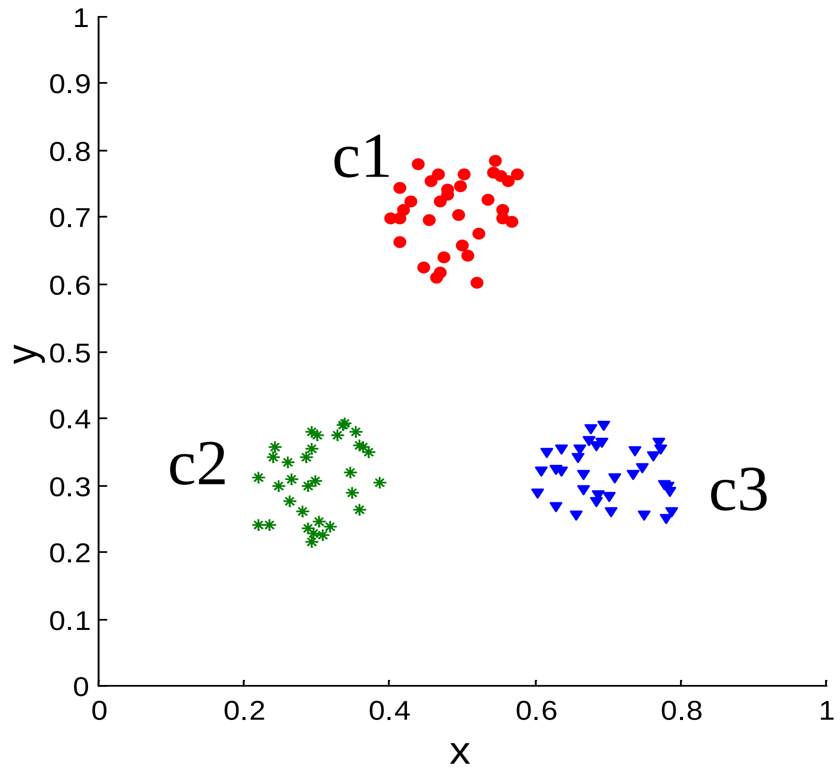
weaker correlation

Higher correlation values: Incidence Matrix and Proximity Matrix are kind of consistent

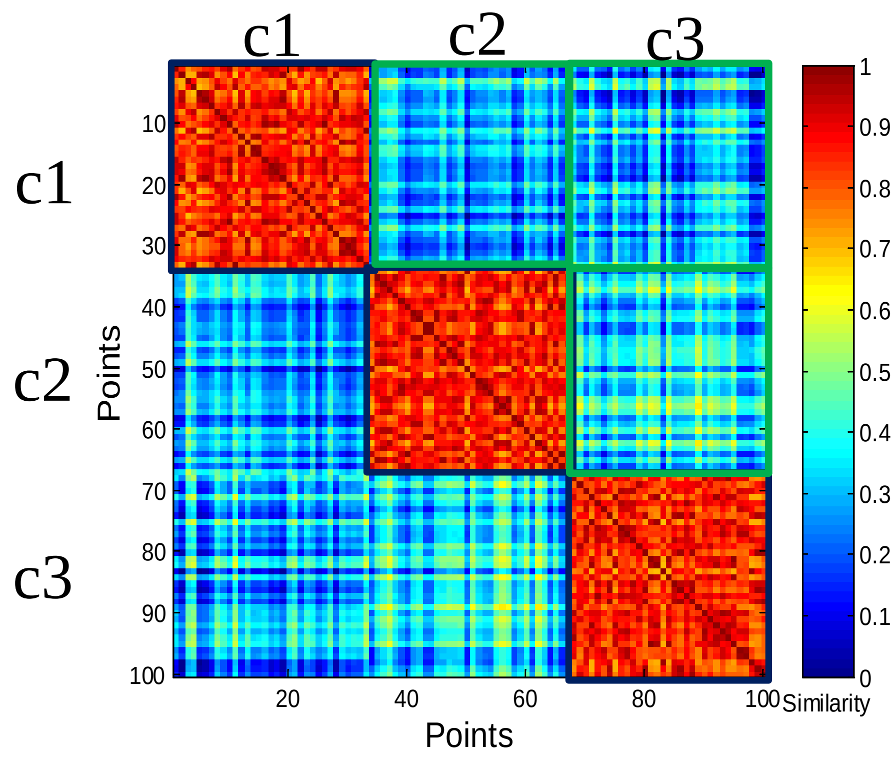
An entry (i,j) in Incidence Matrix is 1 (same cluster) ⬄ sim (i, j) is big in Proximity Matrix (near)

Human perception

- Order the similarity matrix with respect to cluster indexes (from cluster results) and inspect visually. First arrange data points belonging to same clusters together in matrix, and then show their similarity values.



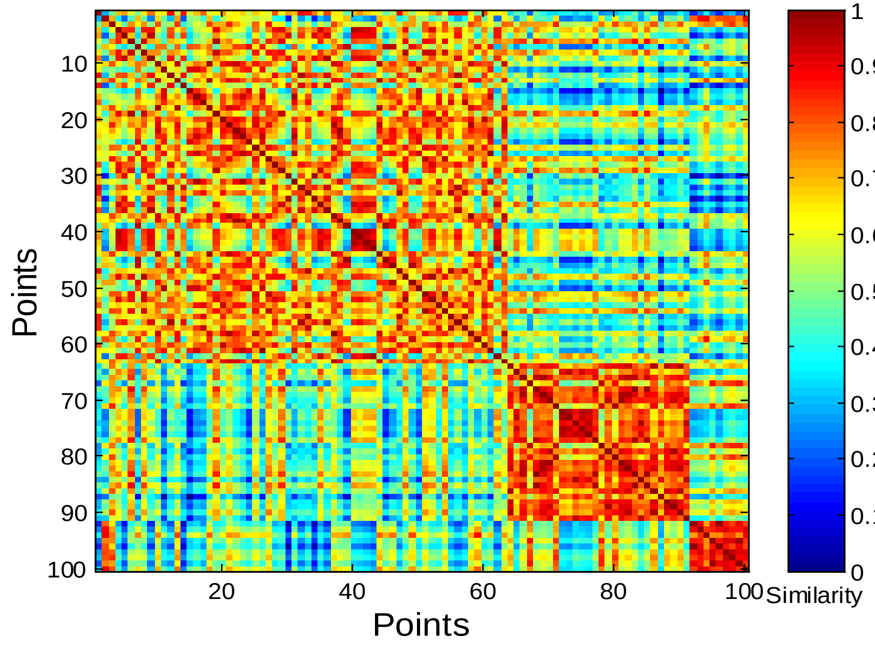
99 data points, 3 clusters



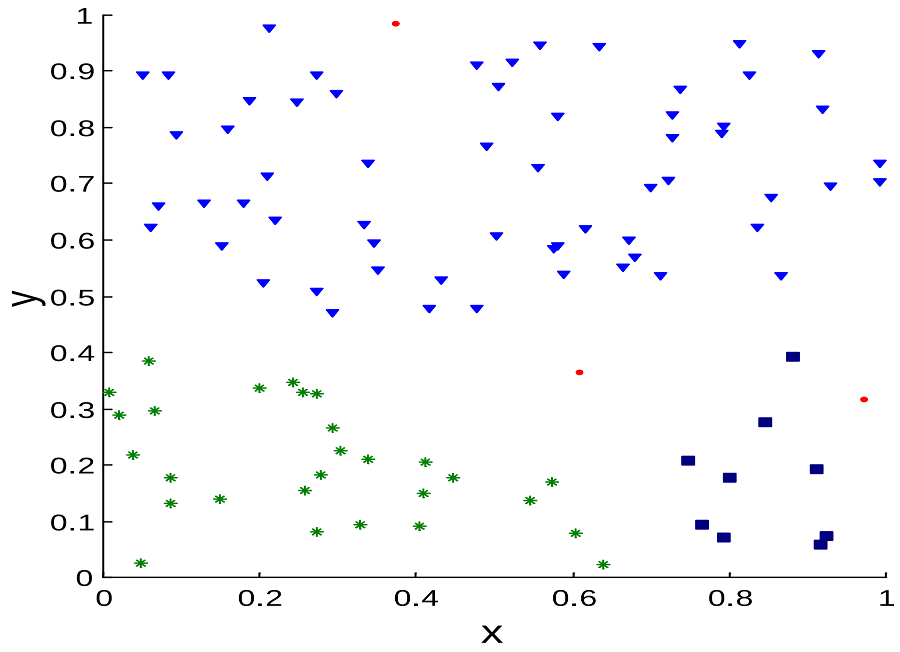
1-33 (from c1), 34-66 (from c2), 67-99 (from c3)

Human perception

- Clusters in **random** data are not so crisp
- it is difficult to find the high quality clusters in this case
- Note we can visualize data points which are of high-dimension, as we just need to use clustering results and similarity scores



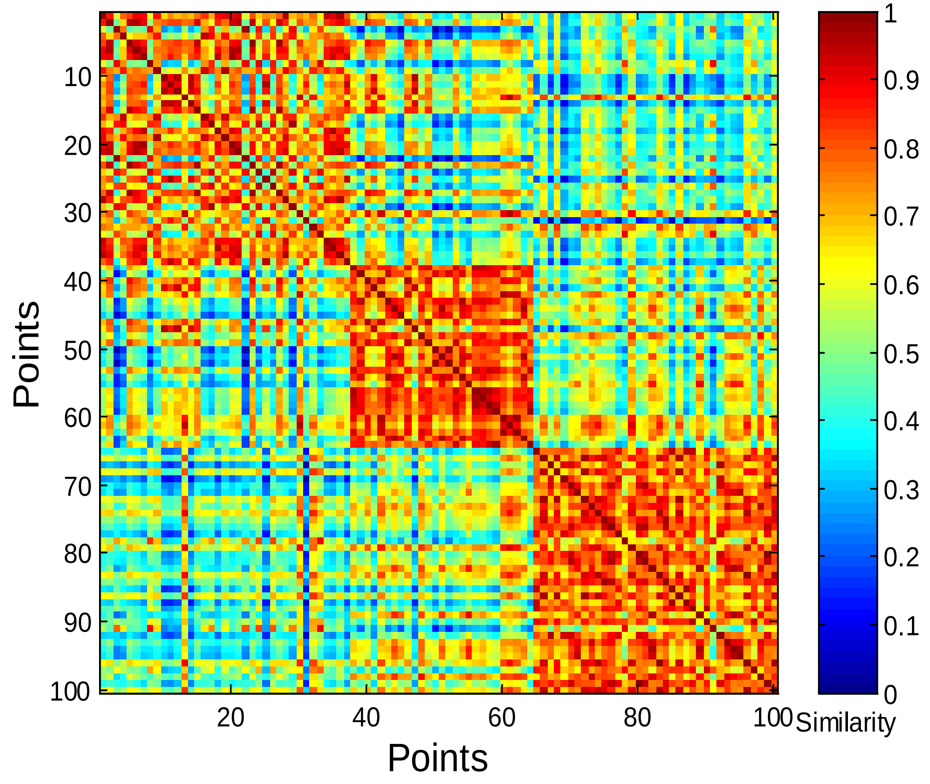
How to map and explain?

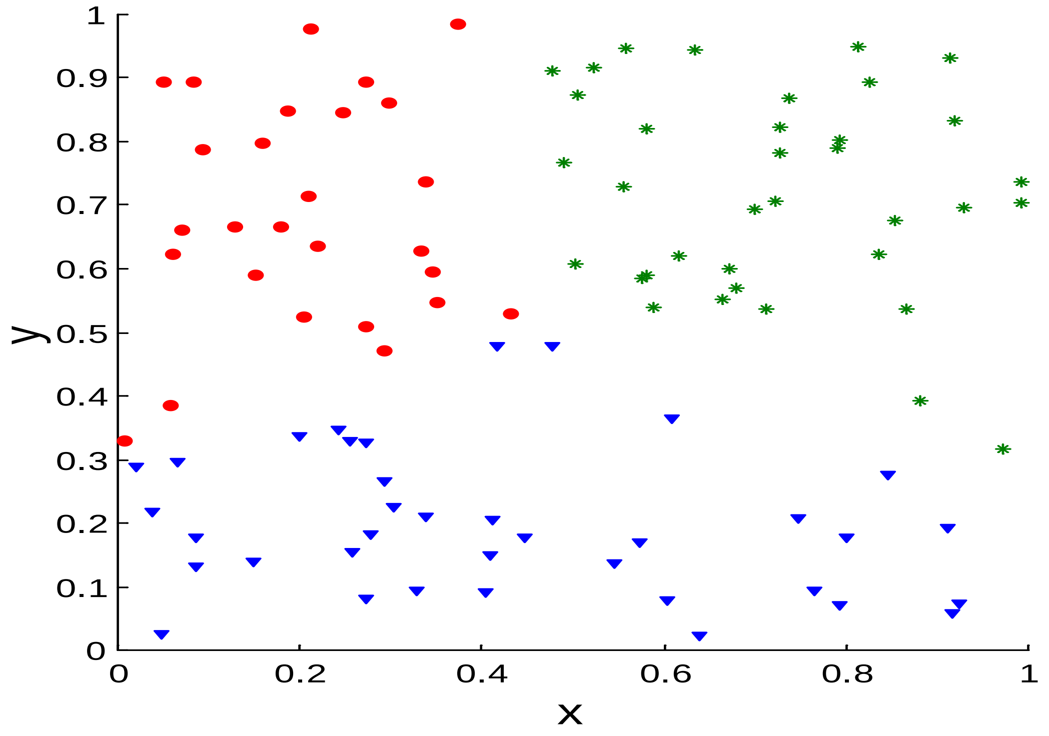


DBSCAN

Human perception

- Clusters in random data are not so crisp

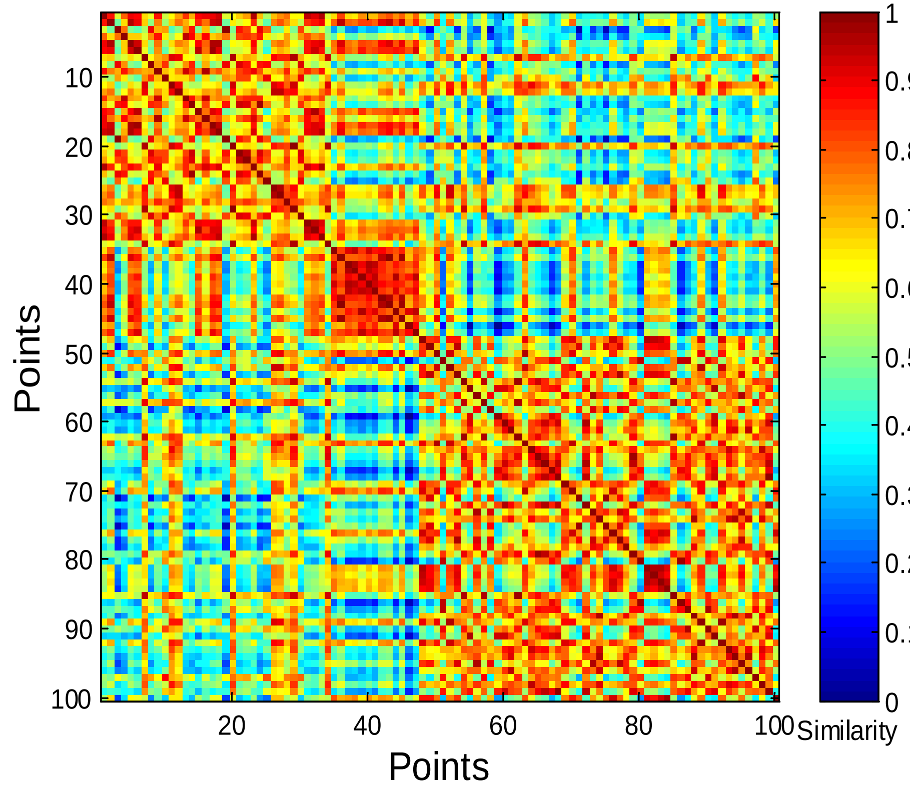


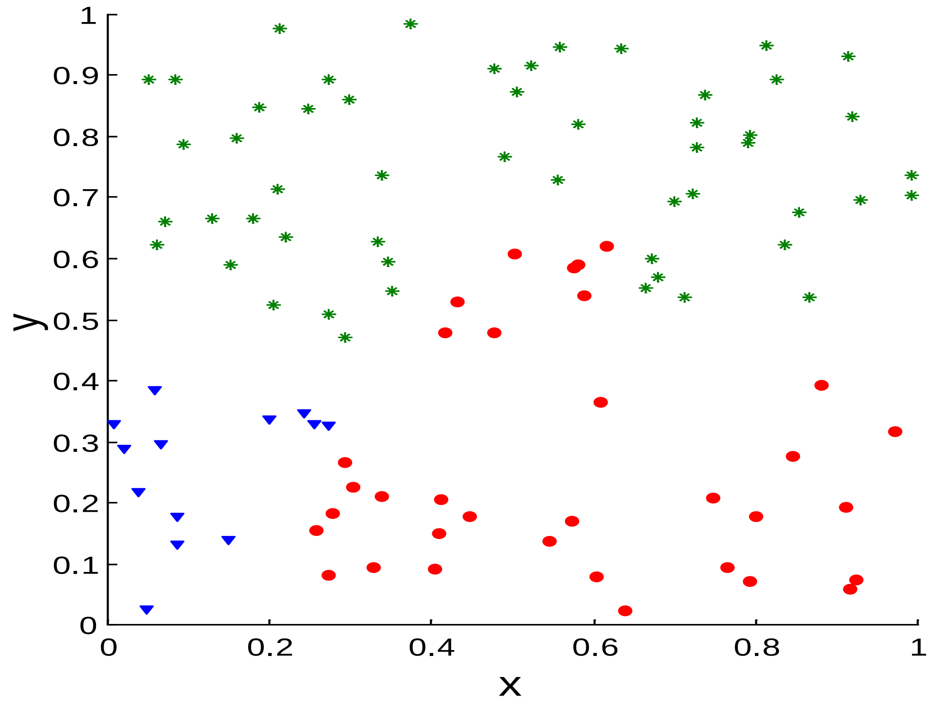


K-means

Human perception

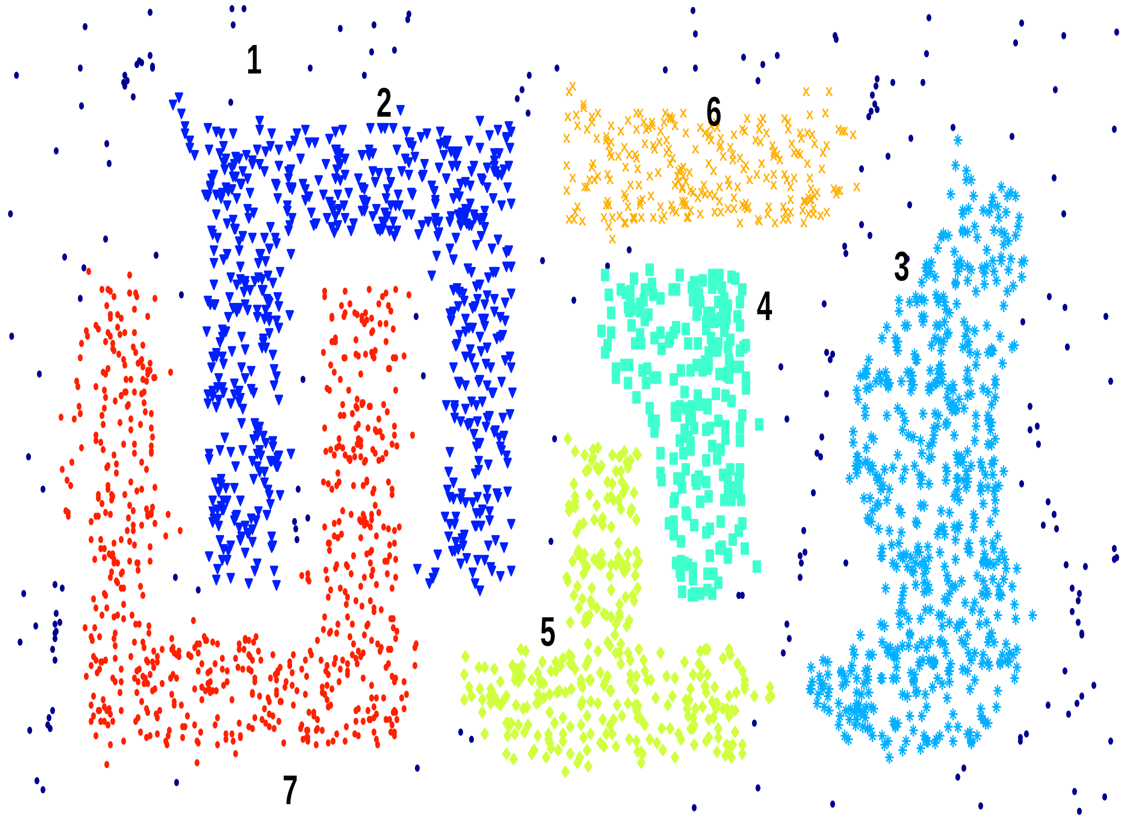
- Clusters in random data are not so crisp

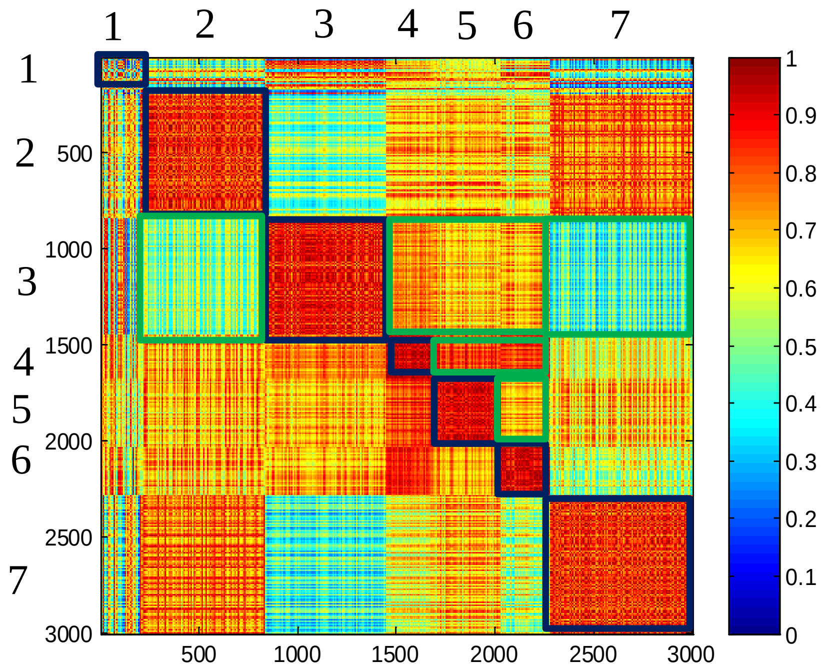




Complete Link

Human perception





DBSCAN

Diagonal is more consistent: similarity within same clusters. The remaining are the similarity between two different clusters, Cluster 2 and 7 could be somewhat similar; C4, 5; C4, 6 are also similar

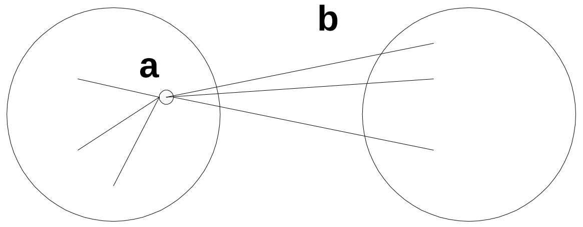
Take cluster 3 as an example, what can you find?

Silhouette Coefficient

- **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 $a < b$ (most cases)

    $s = b/a - 1$ if $a \geq b$ (not the usual cases)



  - Typically, between 0 and 1.
  - The closer to 1, the better (a<<b); negative values - bad quality
- Can calculate the Average Silhouette width for a cluster or a clustering

Thank You

Contact: xlli@i2r.a-star.edu.sg if you have questions

Apprendix
Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster, e.g. SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters, e.g. Squared Error
  - Cohesion is measured by the **w**ithin cluster **s**um of **s**quares (SSE)

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

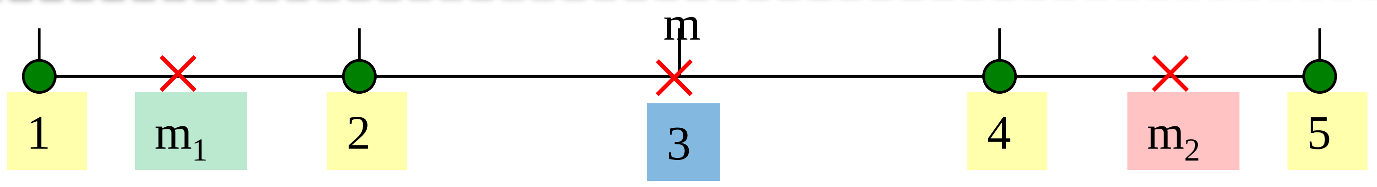
  - Separation is measured by the **b**etween cluster **s**um of **s**quares

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

    - Where $|C_i|$ is the size of cluster i
    - m is the center (mean) for all the data points
    - m_i is the representative point for cluster C_i

Cohesion and Separation

- Example: WSS and BSS (BSS + WSS = constant)



K=1 cluster:

C: point 1, 2, 4, 5

m=3

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

Not cohesive

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

We only have 1 cluster, no separation

$$m = m_i$$

$$Total = 10 + 0 = 10$$

k= 2 clusters:

C1: point 1 and 2
C2: point 4 and 5

m=3, m_1=1.5, m_2=4.5

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

cohesive

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

Bigger Separation

$$Total = 1 + 9 = 10$$

Which clustering is good?