Clustering

May 14, 2017

Karsten Borgwardt, ETH-Department BSSE in Basel

<u>Content:</u>

- What is clustering?
- Why is clustering of essential use in systems biology?
- Which are the most popular and useful clustering algorithms?

Clustering - Definition

What is clustering?

Clustering is the search for “subgroups of similar objects” in a given dataset. Objects from one subgroup should be more similar to each other than objects from other groups.

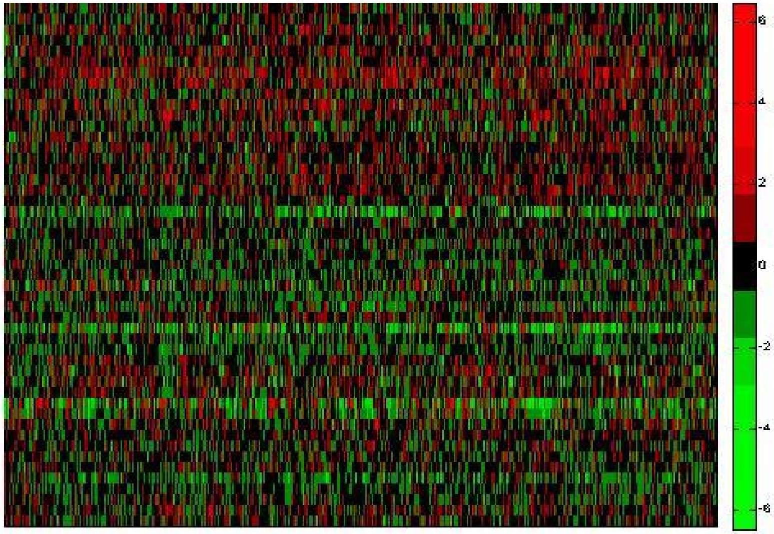
Synonyms (but rarely used): Data partitioning, Class discovery

Clustering - Example: Coregulated genes

Genes form clusters in microarray data:

The expression of several genes is often coupled, as transcription factors regulate more than one gene jointly.

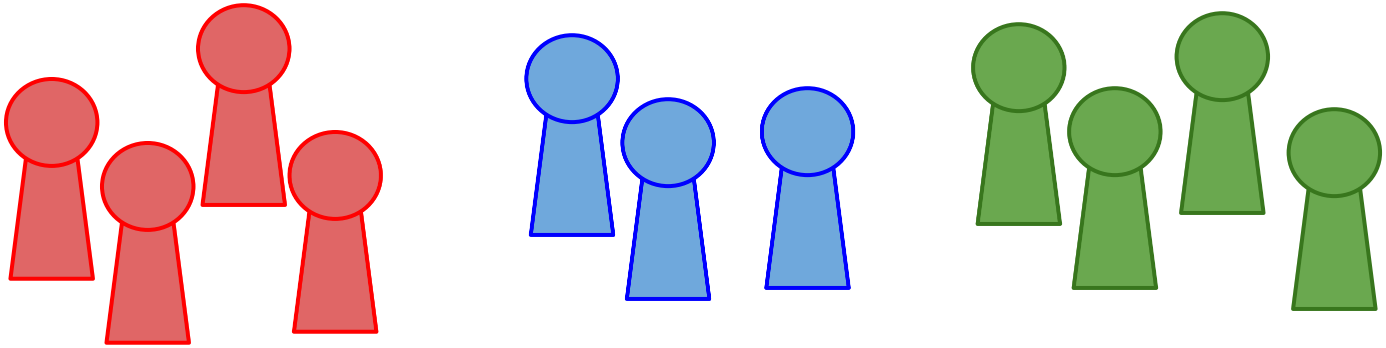
One task of clustering: Find these groups of co-regulated genes.



Clustering - Example: Subphenotype discovery

Subsets of patients may suffer from a particular variant of a disease, and this may be reflected in their gene expression levels.

Finding such subgroups of patients (subphenotype discovery) is one instance of clustering.



Popular reference: Golub et al., Science 1999 (Leukemia)

Clustering - k-means

Centroid-based clustering

Each cluster is represented by a representative vector (needs not be a point in the dataset itself).

Most popular instance (and probably the most popular clustering algorithm in general)**:**

k-means clustering **(Steinhaus, 1957)**

Clustering - k-means

k-means clustering

Input: A set of points $x_1,...,x_n$; an integer *k*

Output: A partitioning of the dataset into *k* disjoint clusters $C_1,...,C_k$ such that the following objective is minimized:

$$\sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} ||\mathbf{x}_i - \mu_j||_2^2$$

Here, μ_j is the center of cluster *j*.

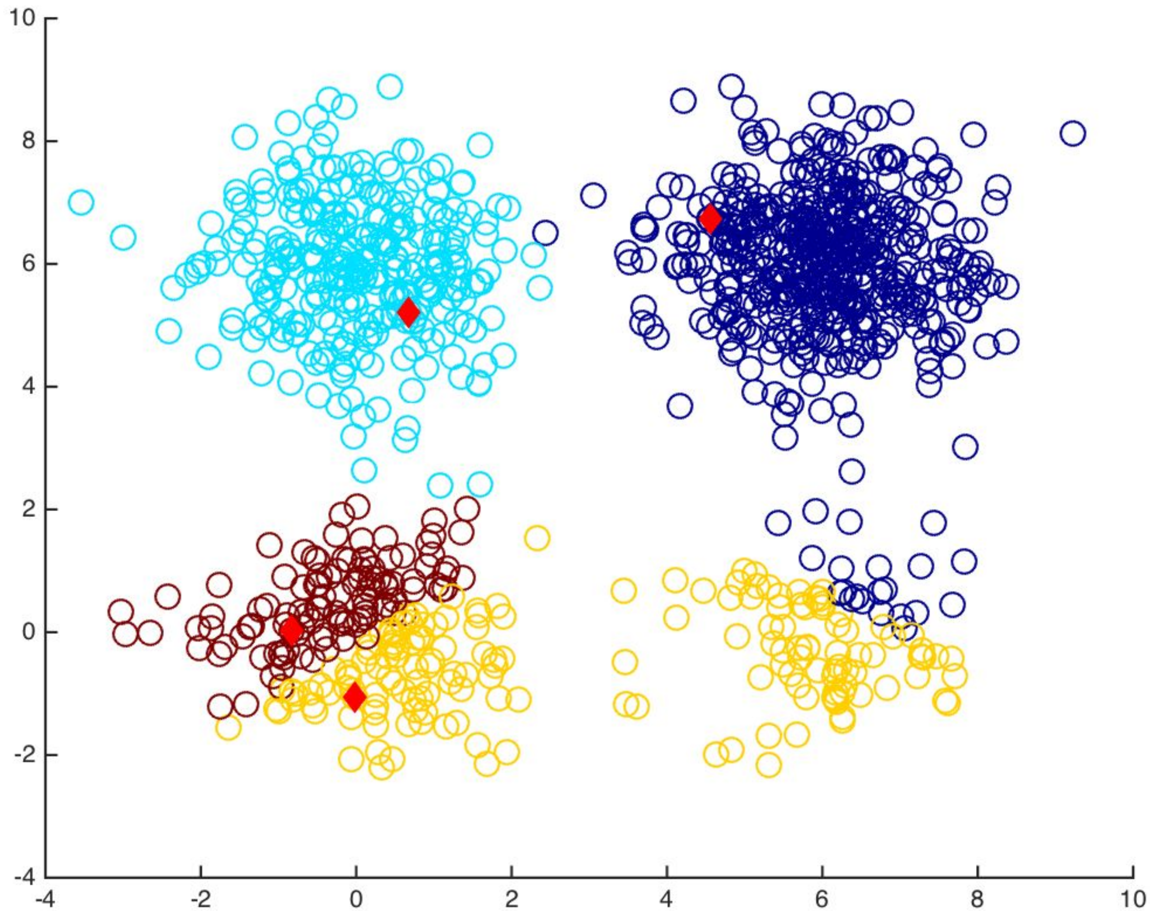
Clustering - k-means

In worst case, one has to consider all possible partitions to find the best k-means solution (NP-hard problem).

In practice, one uses typically the **Lloyds algorithm (Lloyds, 1957):**

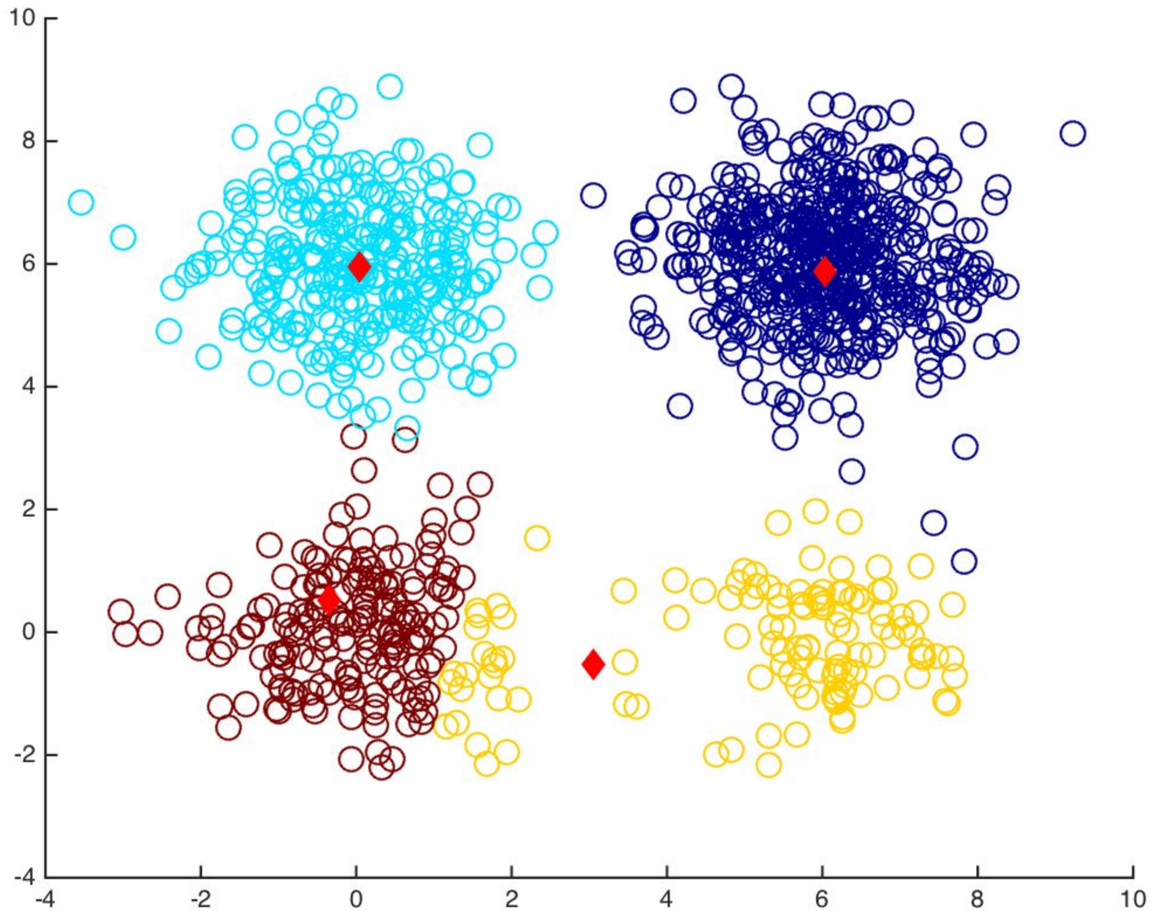
1. ~~Randomly pick k points as initial cluster means~~
2. ~~Assign each points to its nearest cluster mean~~
3. ~~Recompute the mean of each cluster~~
4. ~~Repeat steps 2 and 3 until cluster assignment does not change any more~~

Clustering - k-means - Example



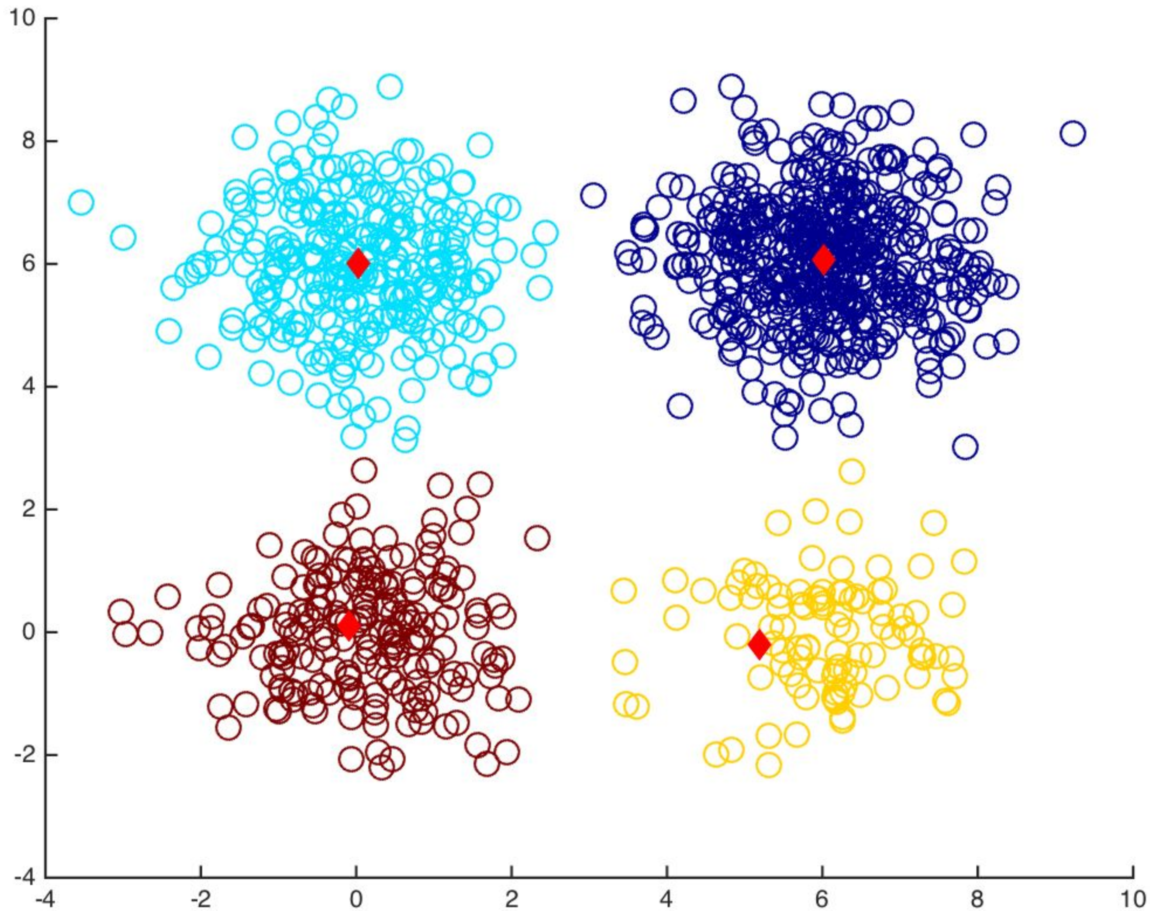
Iteration 1

Clustering - k-means - Example



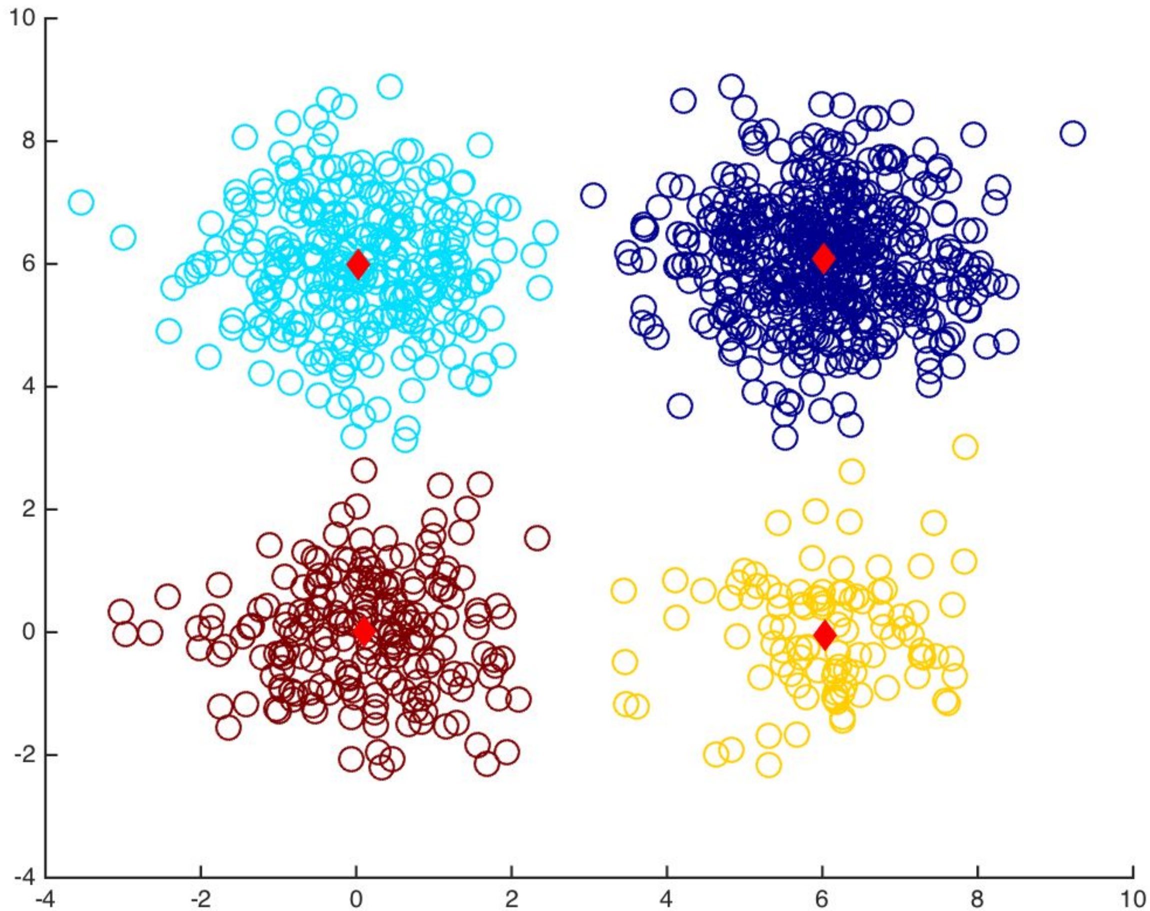
Iteration 2

Clustering - k-means - Example



Iteration 3

Clustering - k-means - Example



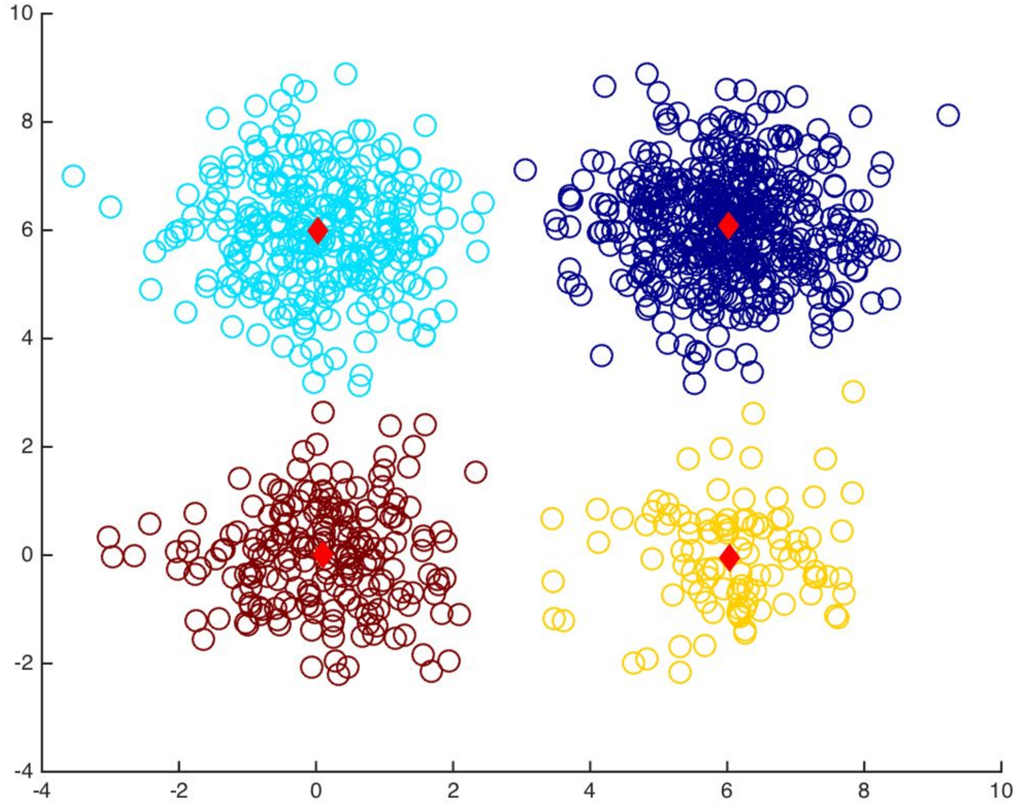
Iteration 4

Clustering - k-means - Limitations

Limitation 1:

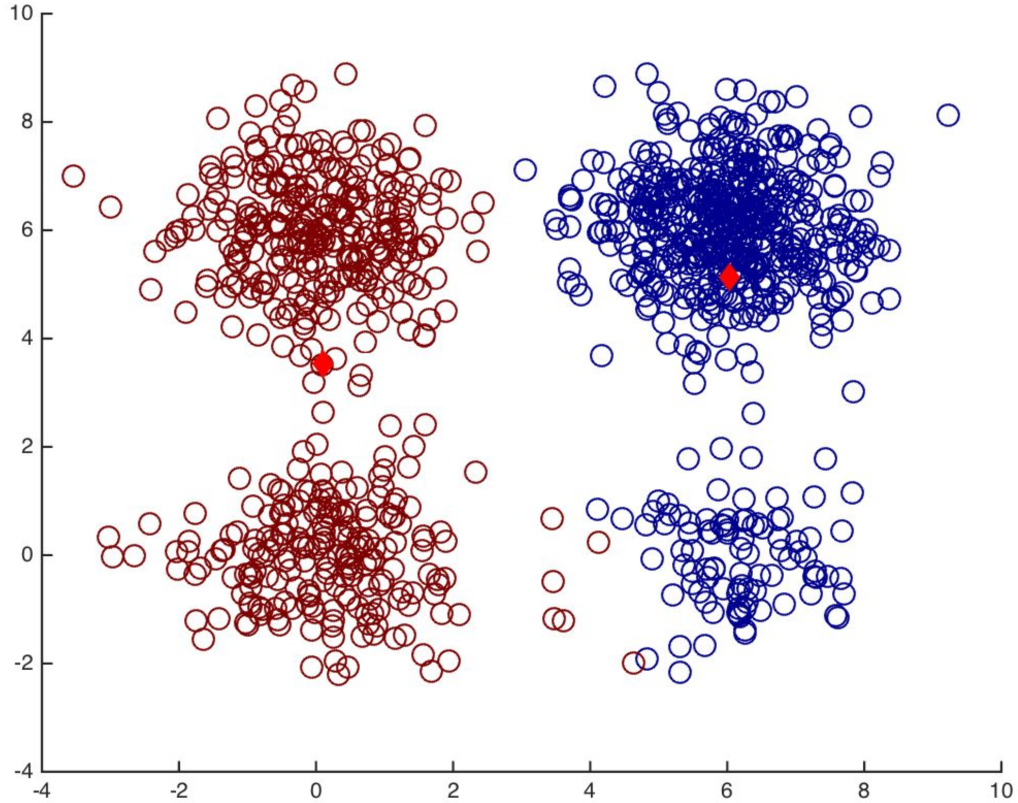
In k-means, the number of clusters k has to be specified by the user.

Clustering - k-means - Number of clusters



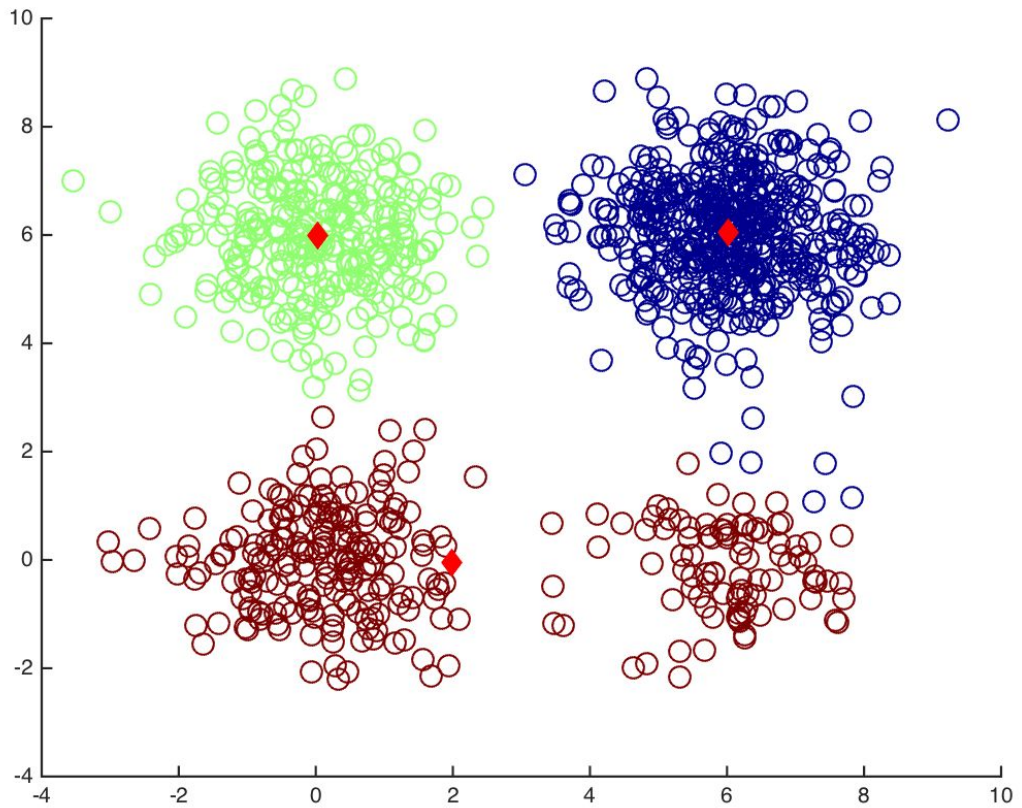
k=4

Clustering - k-means - Number of clusters



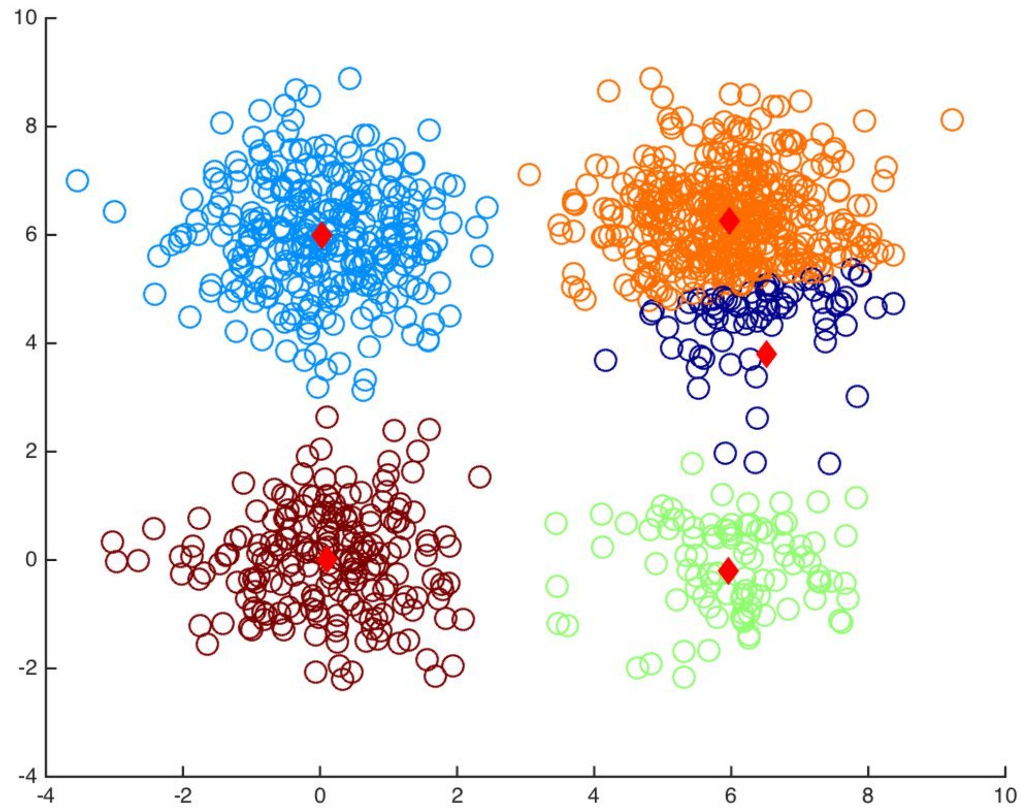
k=2

Clustering - k-means - Number of clusters



k=3

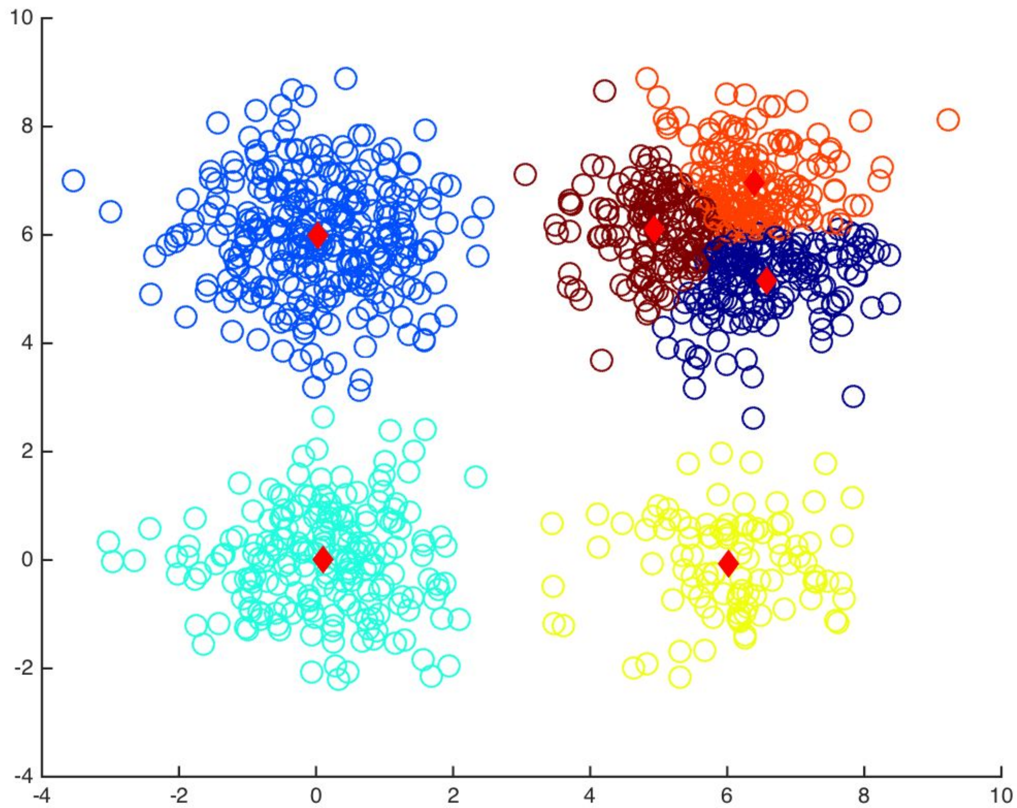
Clustering - k-means - Number of clusters



k=5

Clustering - k-means - Number of clusters

this is a spherical cluster which k means can adequately work with



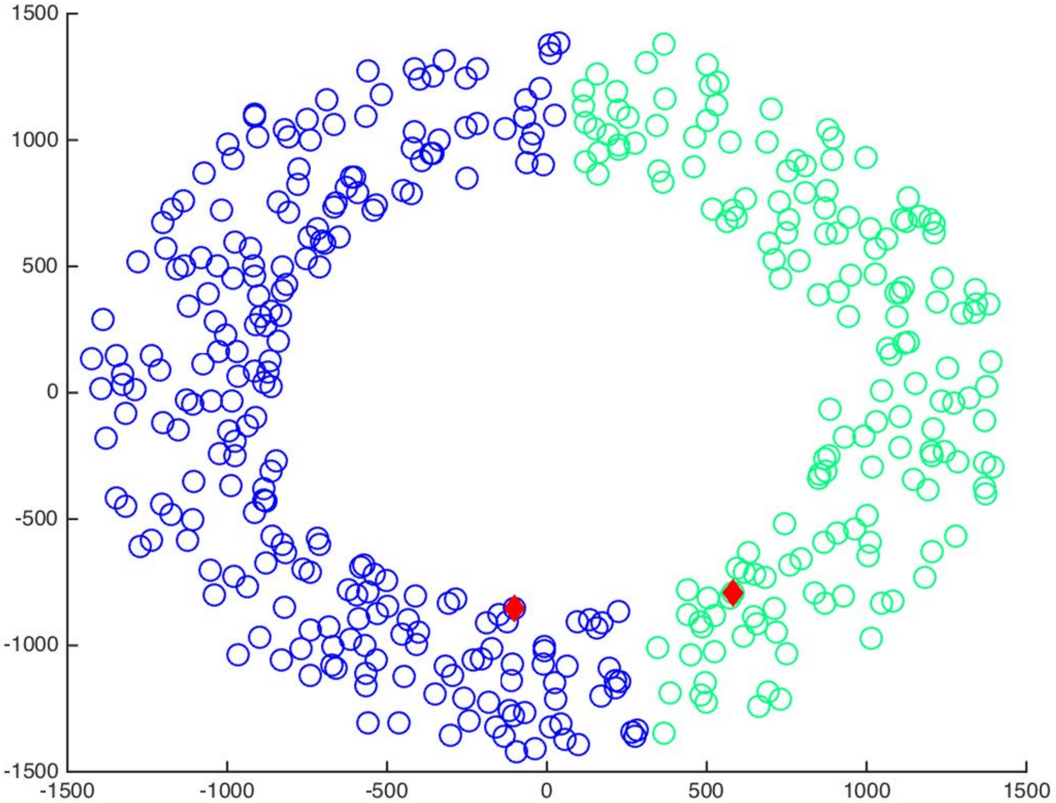
k=6

Clustering - k-means - Limitations

Limitation 2:

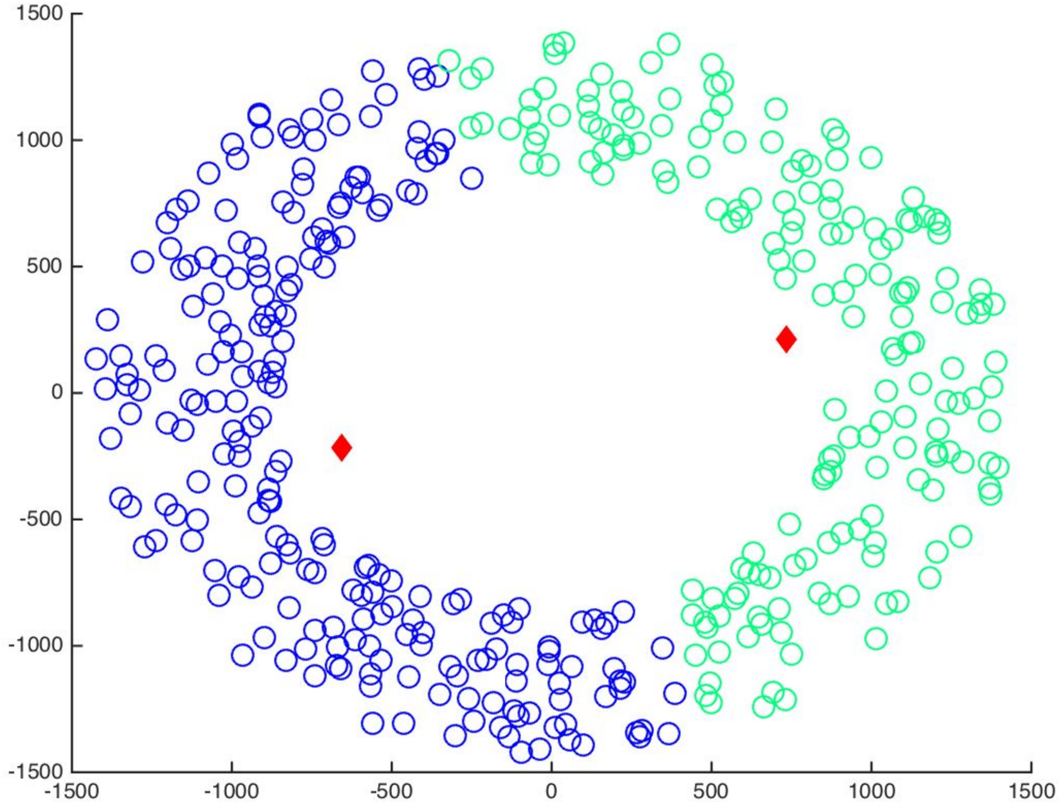
k-means is initialization-dependent

Clustering - k-means - Initializations 1.a



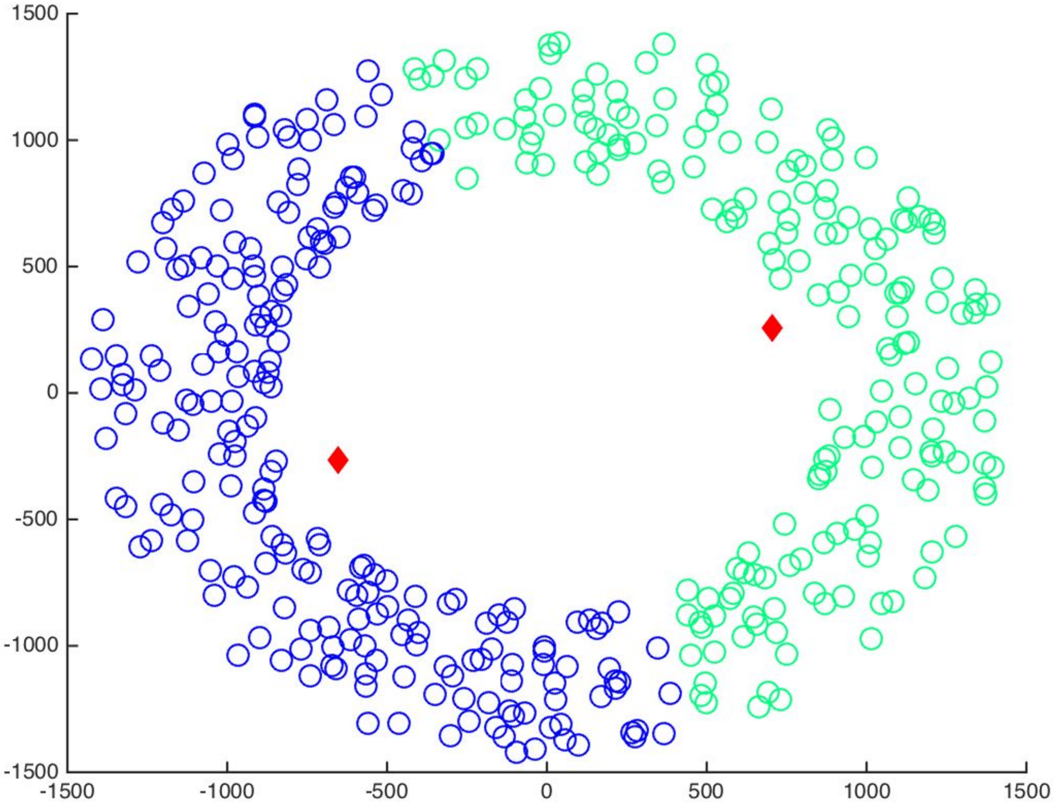
Iteration 1

Clustering - k-means - Initializations 1.b



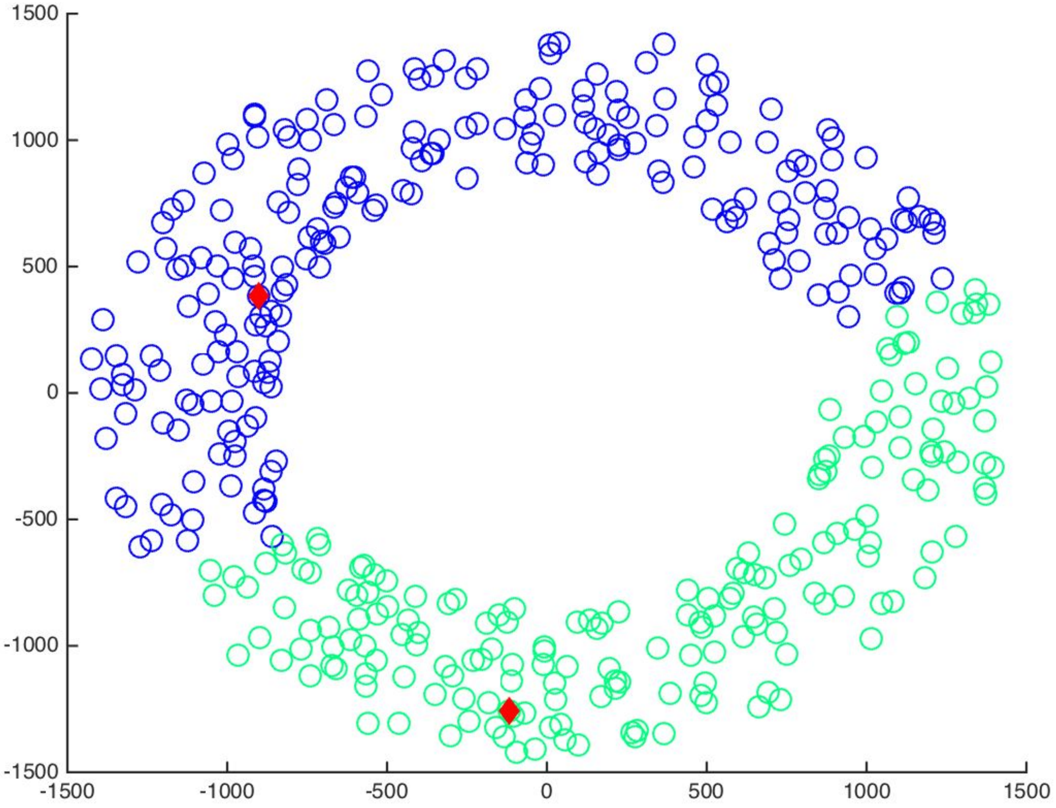
Iteration 5

Clustering - k-means - Initializations 1.c



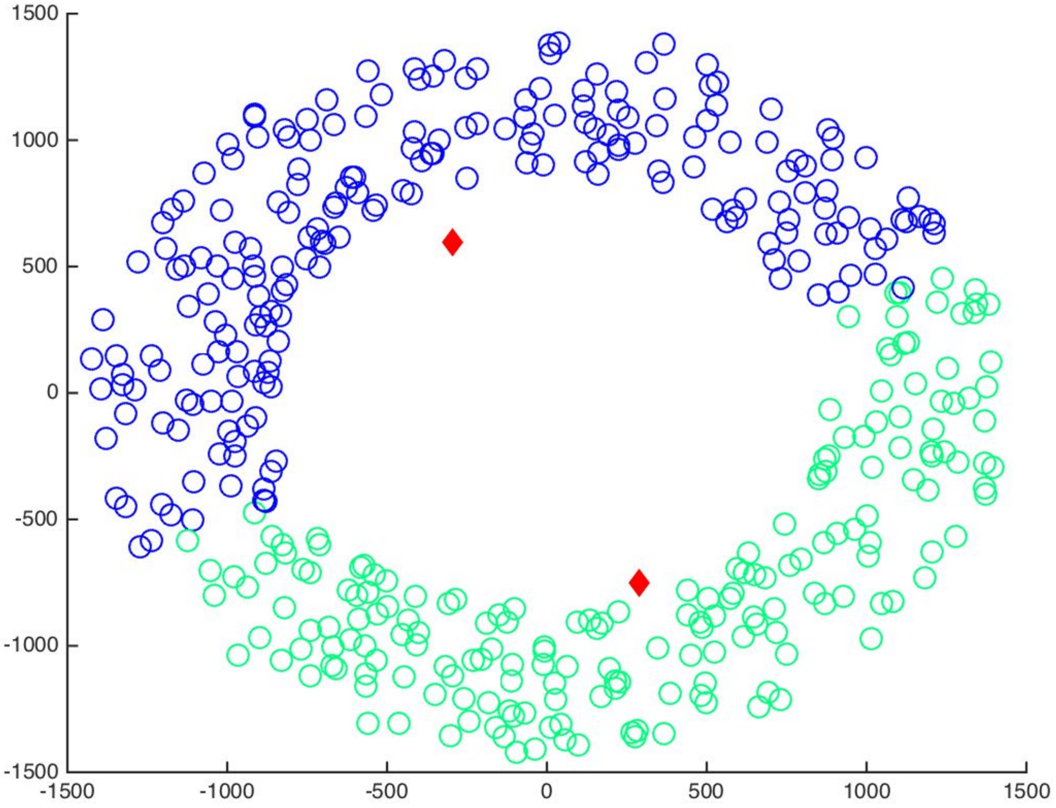
Iteration 10

Clustering - k-means - Initializations 2.a



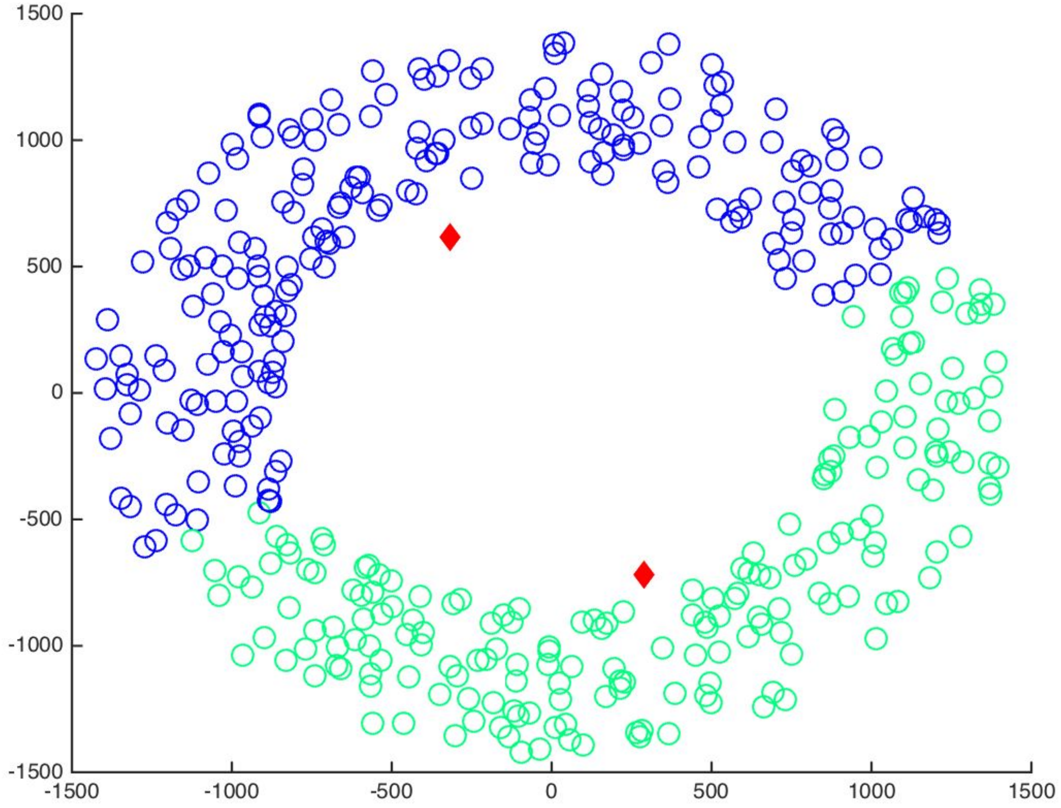
Iteration 1

Clustering - k-means - Initializations 2.b



Iteration 2

Clustering - k-means - Initializations 2.c



Iteration 4

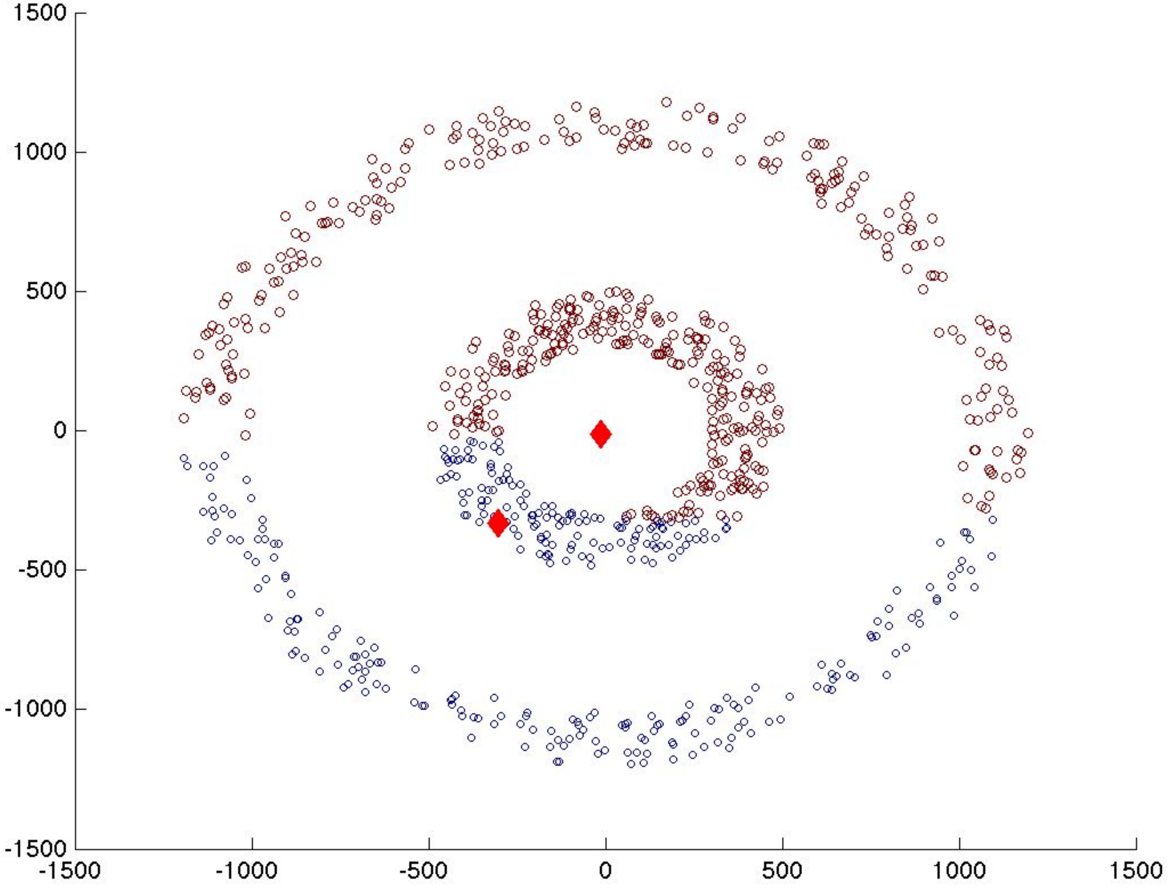
Clustering - k-means - Limitations

Limitation 3:

k-means will miss clusters of particular (non-spherical) shapes

Clustering - k-means - Non-spherical clusters

k means caannot recognize non spherical clusters and fails to give an appropriate answer

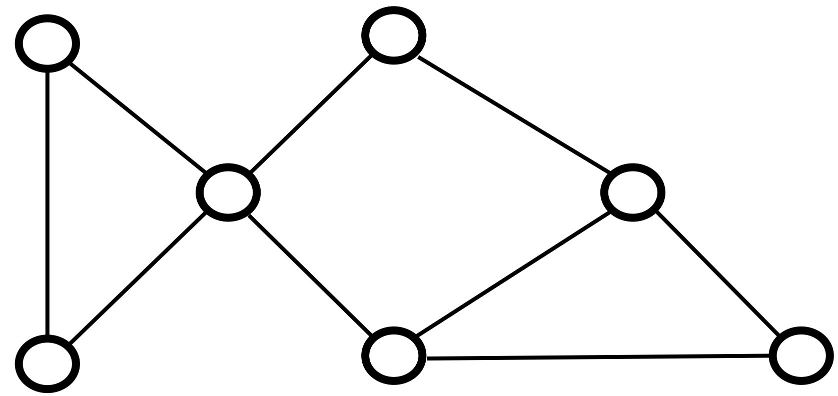


Clustering - k-means

Disadvantages

- Number of clusters k has to be prespecified
- Initialization-dependent
- Solution of Lloyd's algorithm is local optimum (better solutions may exist)
- Often misses non-spherical clusters

Clustering - Graph-based clustering



Assumptions:

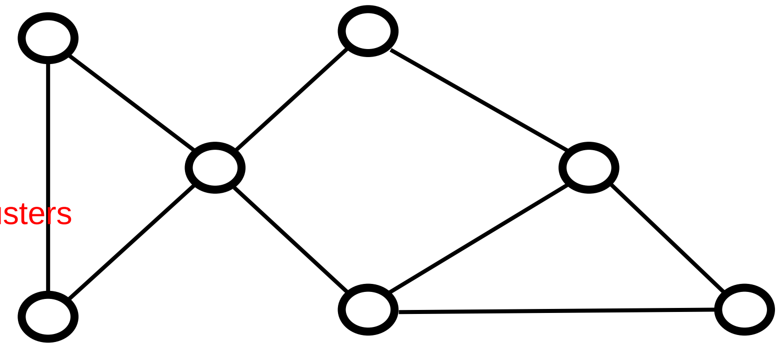
- The data is given in form of a network/graph
- Each node is an object
- Edges connect related objects
- Edge weights represent distances between objects

Clustering - Graph-based clustering

Graph-based clustering:

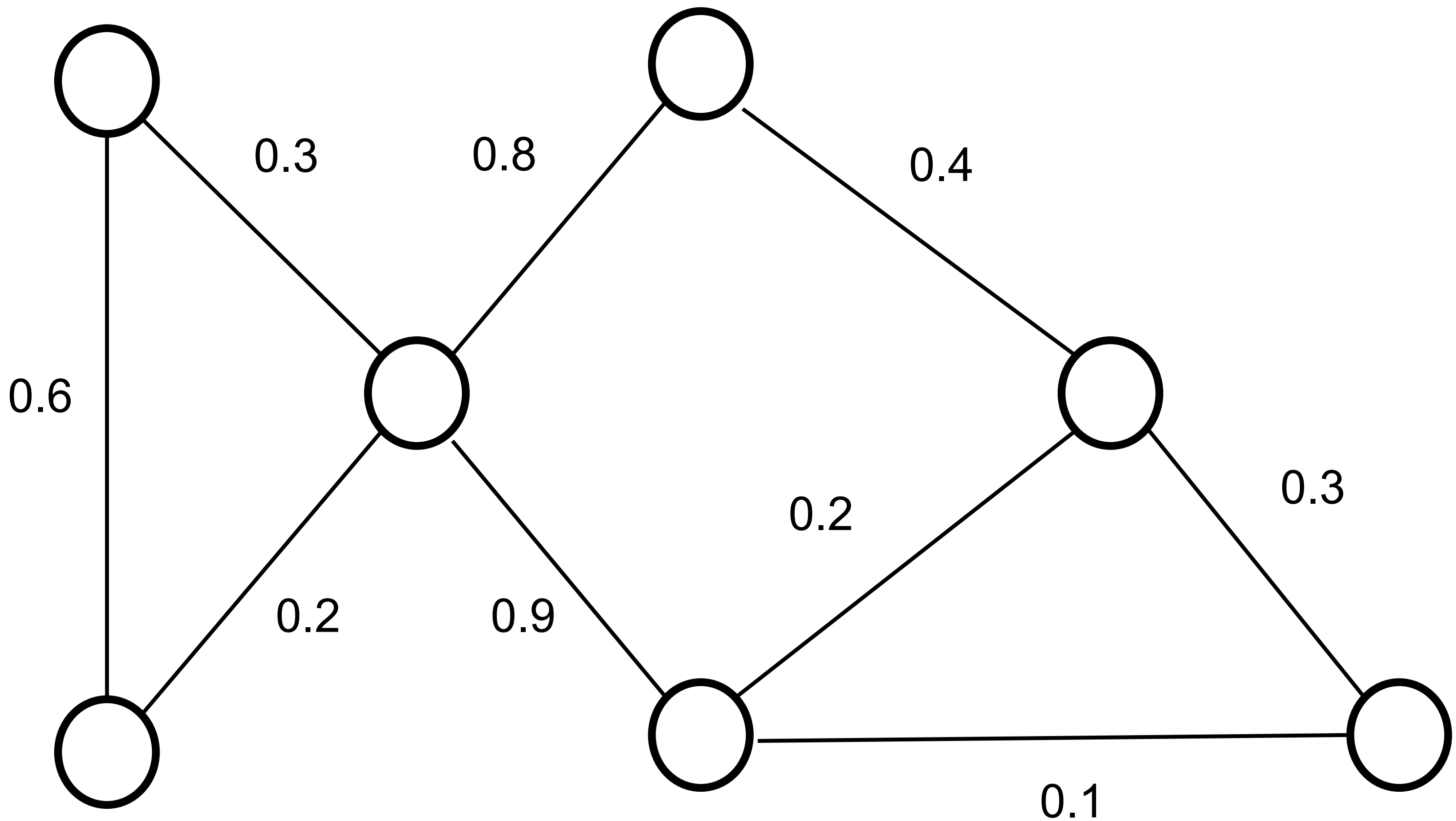
1. Remove all edges with weight > user-defined threshold θ
2. Find all connected components in the resulting graph
3. Each component is one cluster

we delete those edges
and get subgraphs -> those are our clusters



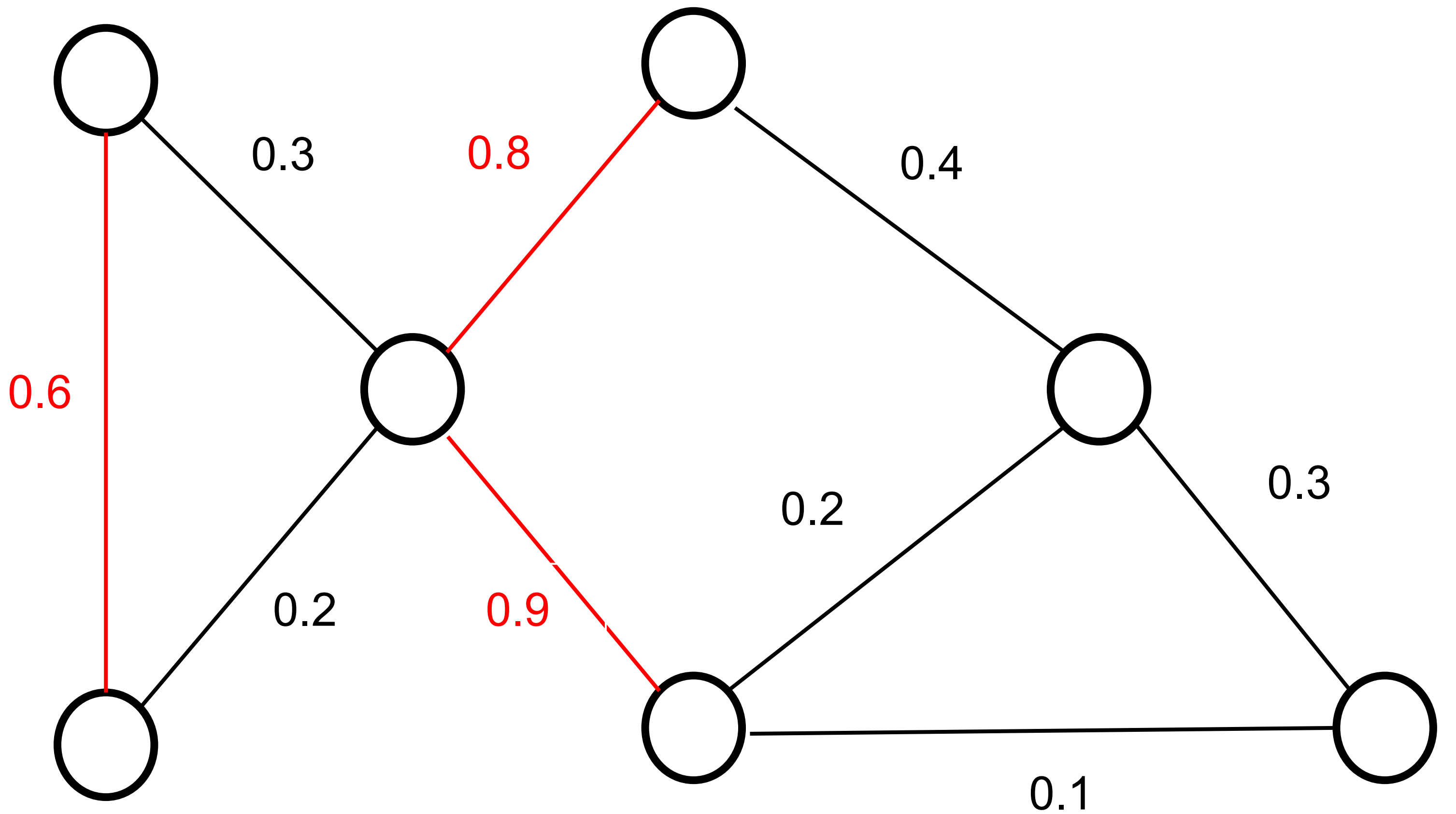
Graph component:
Two nodes belong to the same *graph component* if there is a path between them.

Clustering - Graph-based clustering



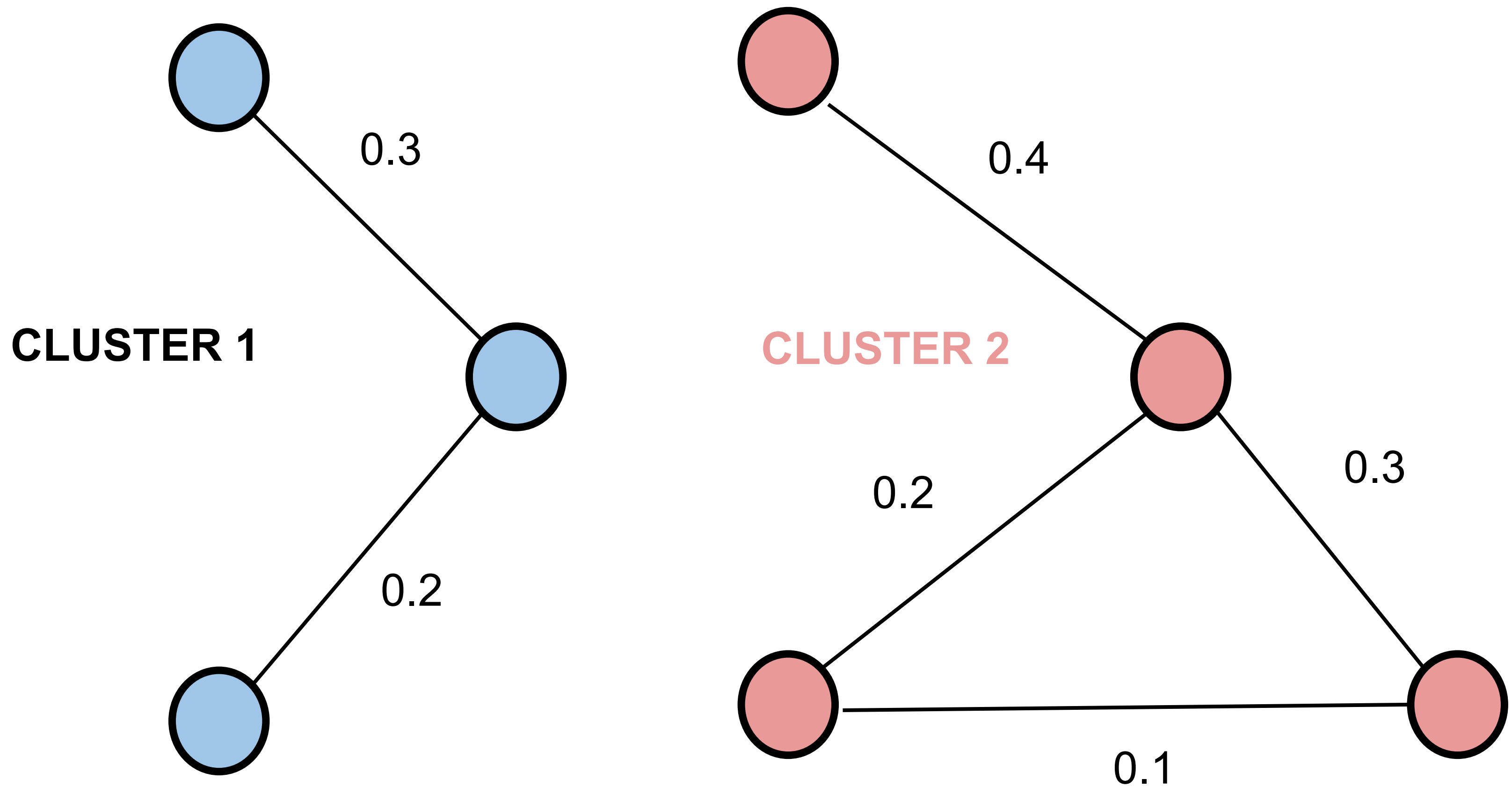
Clustering - Graph-based clustering

Remove all edges with weight > θ (here θ = 0.5)



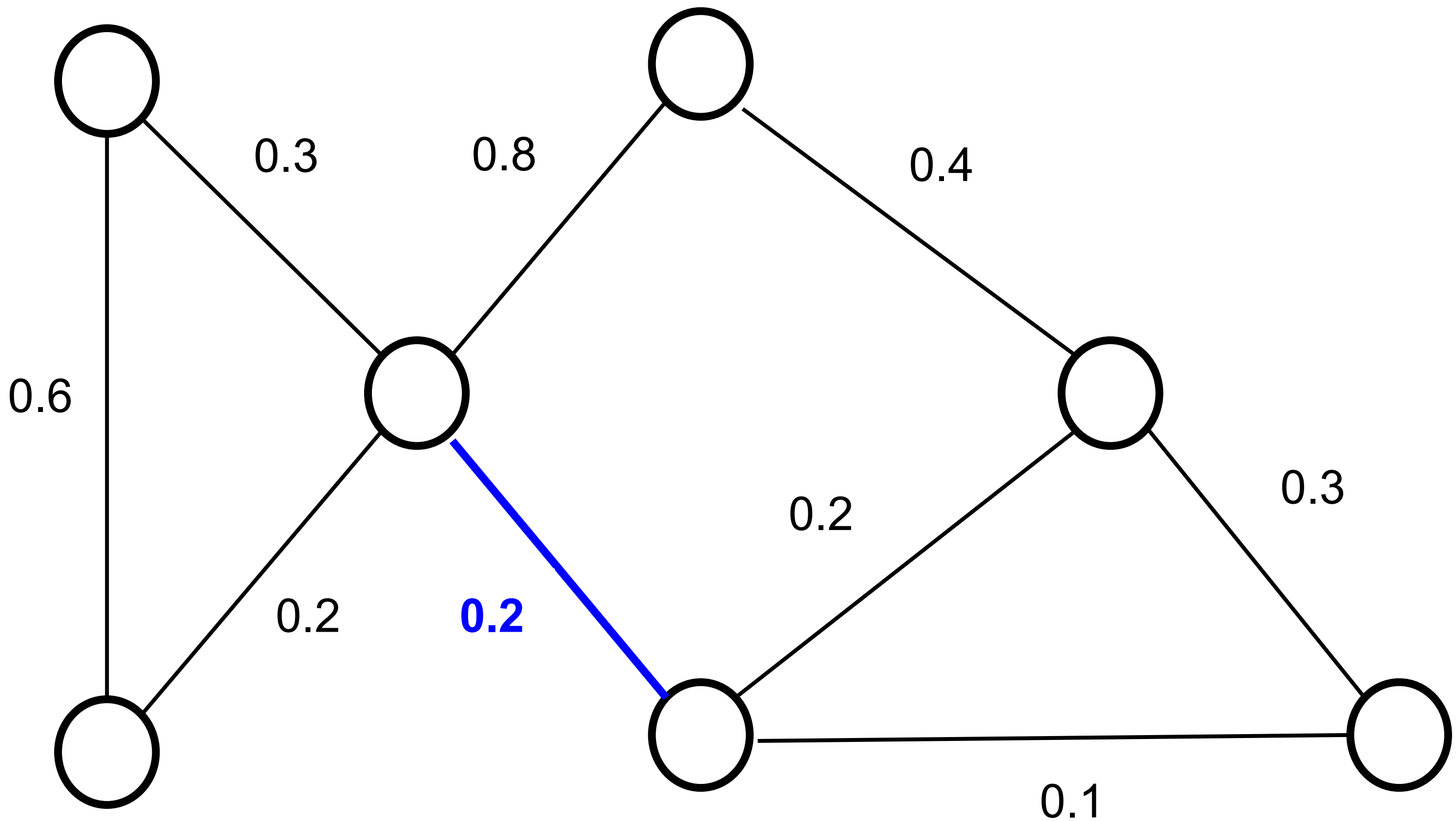
Clustering - Graph-based clustering

after deleting, those clusters have maximally a distance of theta (=0.5)



Clustering - Graph-based clustering

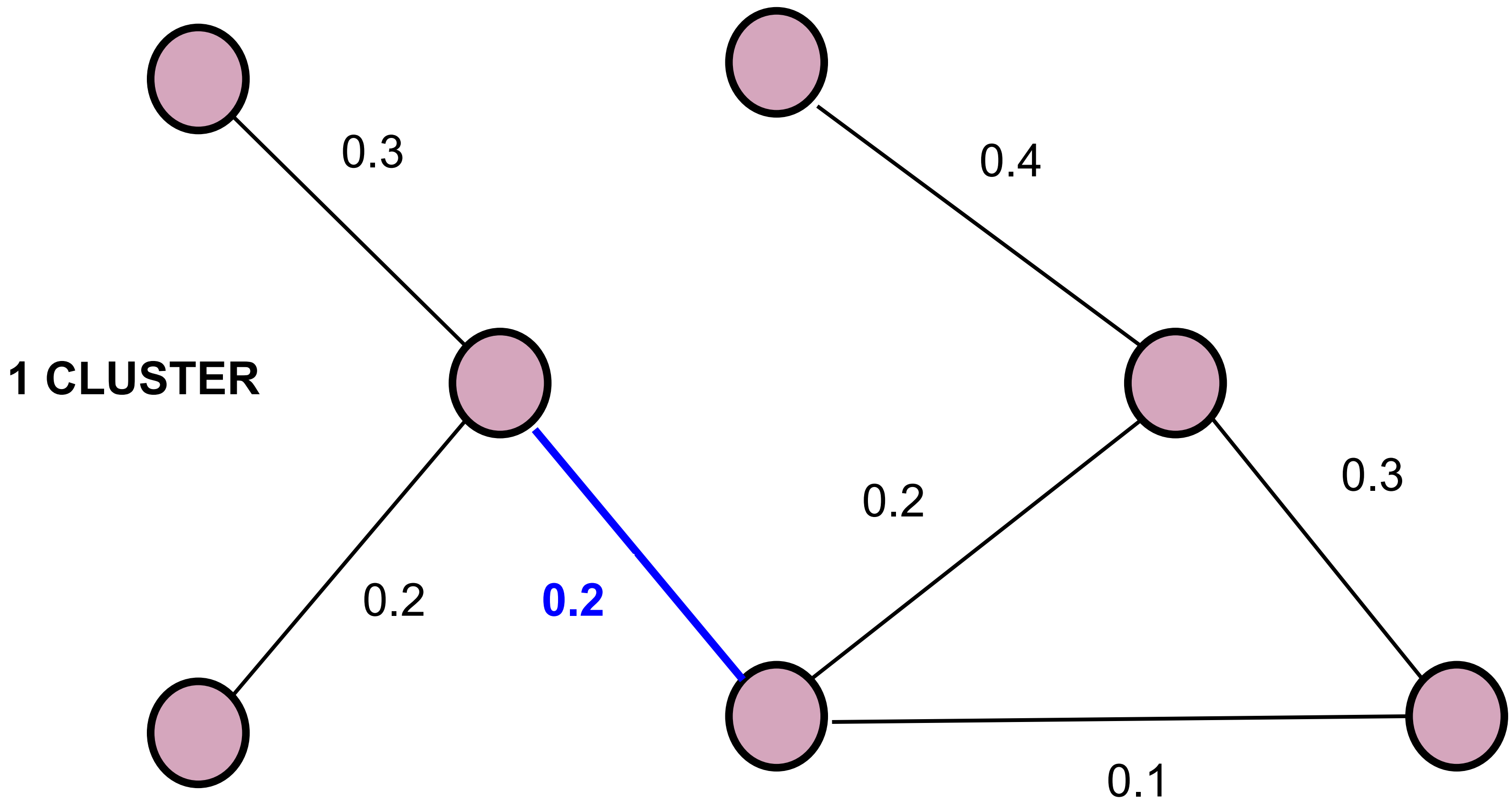
Single link effect



Clustering - Graph-based clustering

if we get noise at one point in the graph the whole cluster is different
=> graph clusters are not robust to noise data. So, we defined DBSCAN

Single link effect



this graph clustering is robust to noise data

Clustering - DBSCAN

Noise-robust variant of graph-based clustering: **D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise (DBSCAN)

In **DBSCAN** (Ester et al., 1996), there are three classes of points:

X is core object <=> there are y>minpoints in r_epsilon

- **Core object:** a point is a core object, if there are (MinPts) points within a distance of (epsilon) from this point. Both (MinPts) and (epsilon) are user-defined parameters.
- **Border point:** a point that is not a core object, but in the epsilon-neighborhood of a core object
- **Noise:** All points that are neither a core object nor a border point.

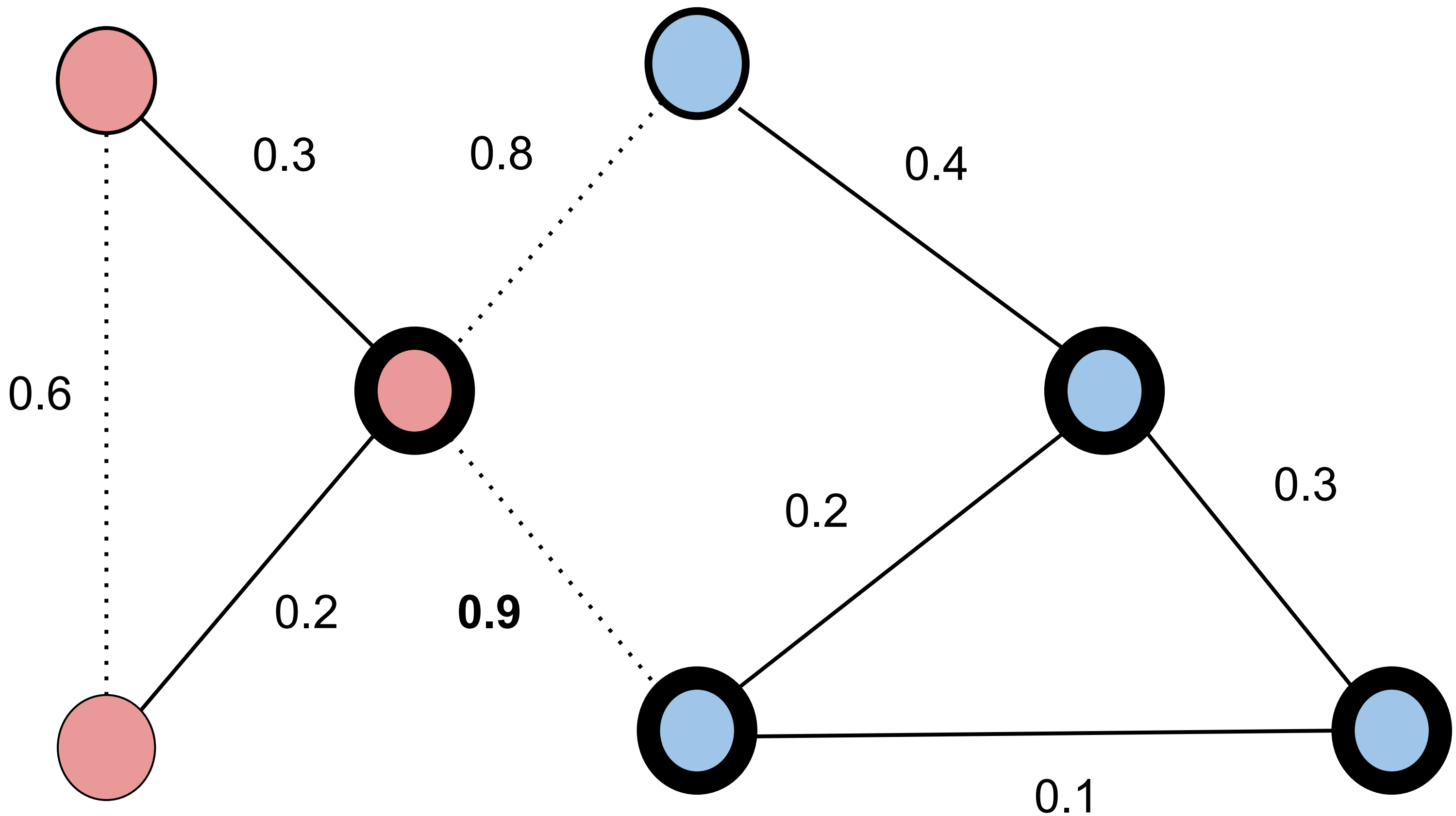
Clustering - DBSCAN - Algorithm

Core function: **Expand cluster(p,C)**

1. Pick a point that has not been assigned to a cluster yet
2. If it is a core object, add it to a new cluster C (if not, label it as “noise”)
3. Add its neighbors to the same cluster C
4. Check for each of the neighbors whether it is a core object
   a. If yes, assign its neighbors to C and perform Step 4 for these neighbors
   b. If no, do not further extend the cluster from this point.
5. Return to Step 1 until all points have been clustered.

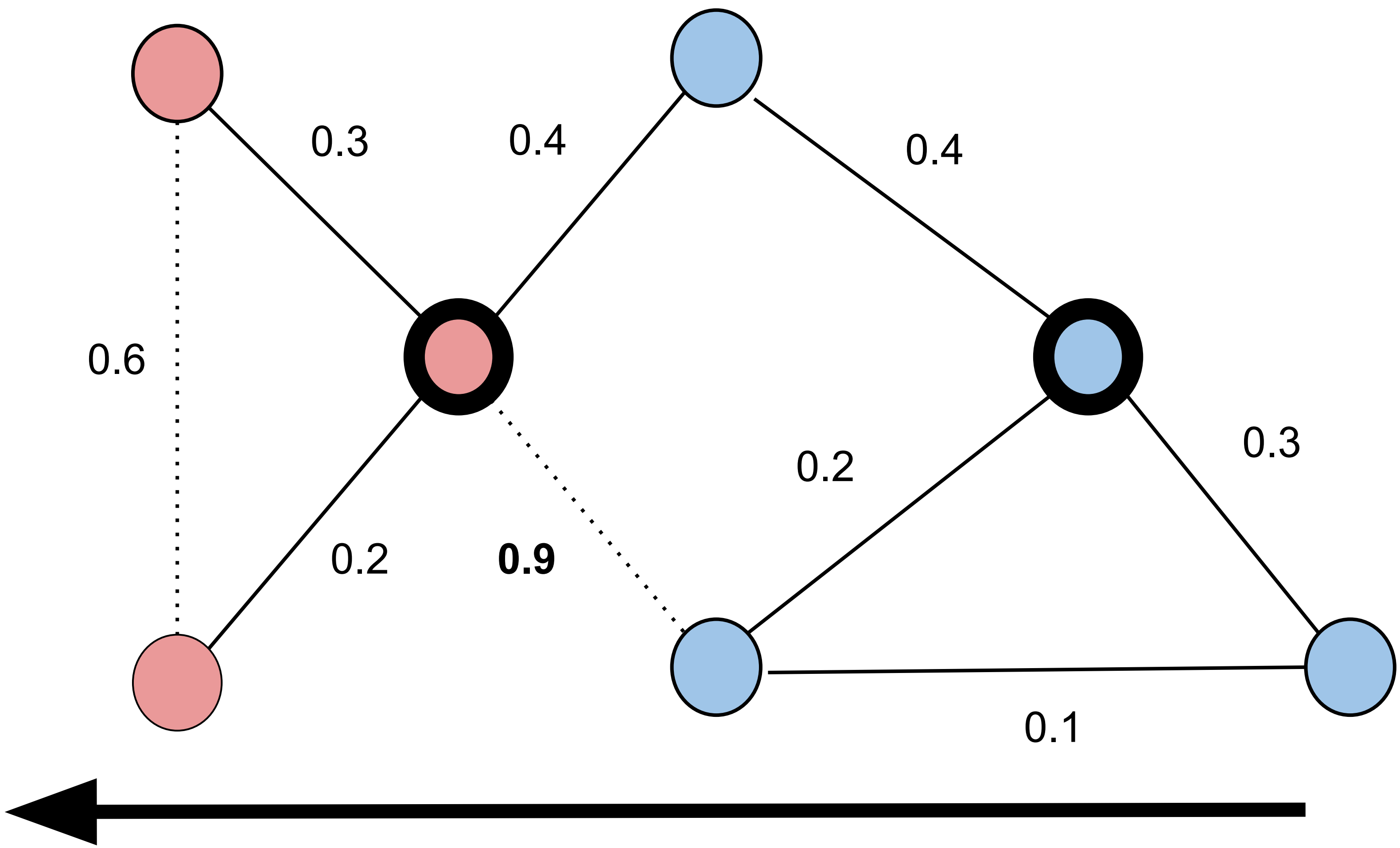
Clustering - DBSCAN - Example 1

MinPts = 2, epsilon = 0.5



Clustering - DBSCAN - Example 2.a

MinPts = 3, epsilon = 0.5

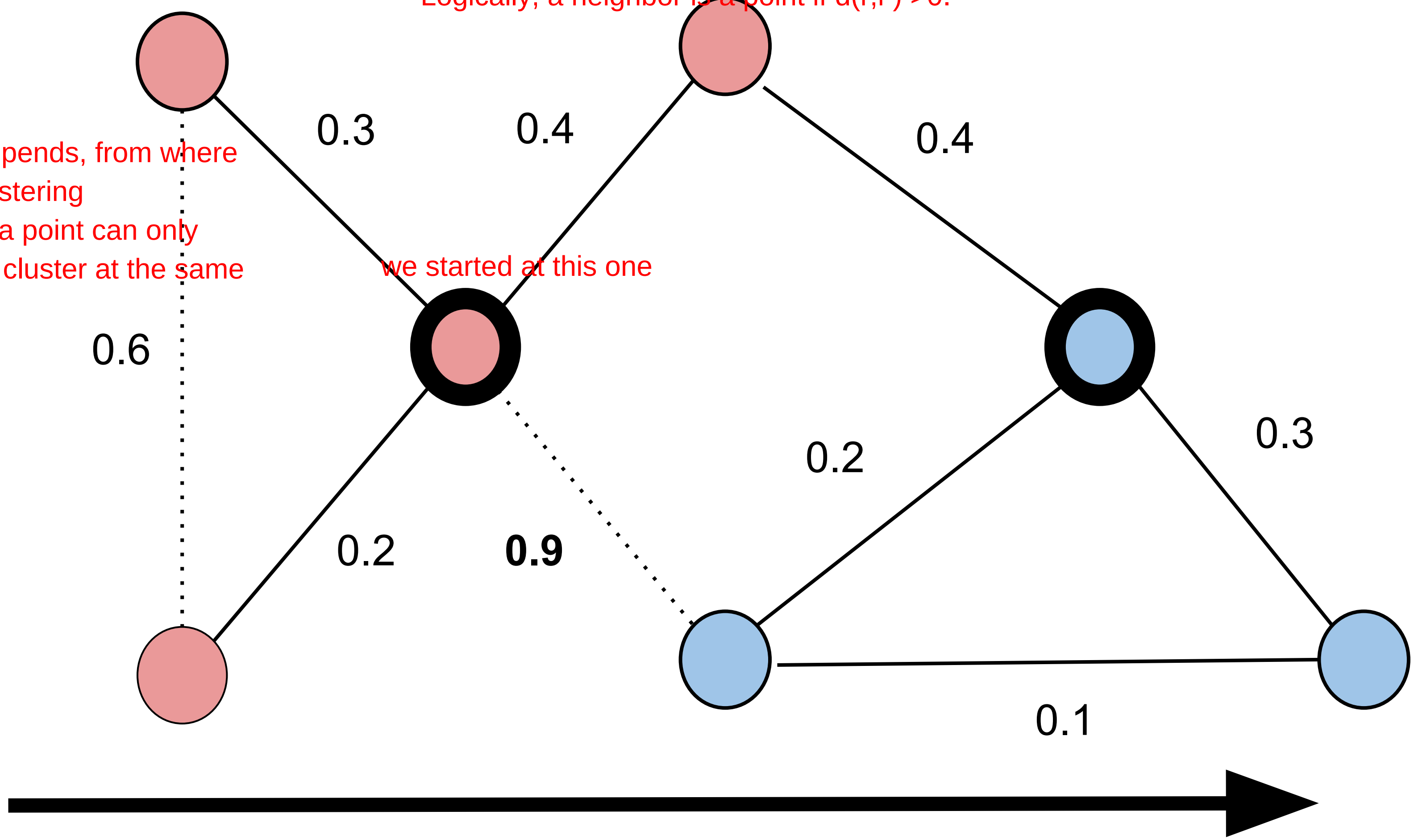


Clustering - DBSCAN - Example 2.b

MinPts = 3, epsilon = 0.5

when implementing, we have to define whether we count the first point also to the minpoints or not. in the example below, it was not counted to minpoints. Logically, a neighbor is a point if d(r,r') >0.

here it also depends, from where we started clustering to begin with. a point can only belong to one cluster at the same time.



Clustering - DBSCAN

Advantages

- No need to set the number of clusters as in k-means
- Able to find clusters missed by k-means
- More robust to noise than graph-based clustering
- Successful in many clustering applications

Disadvantages

- Two parameters have to be set
- Initialisation-dependent results
- If density varies, which is typical in high-dimensional datasets, many clusters will remain undetected.

Clustering - Hierarchical clustering

Motivation

- Graph- and centroid-based clustering are “flat” - the data is partitioned into clusters.
- In real data, clusters often contain clusters themselves - a hierarchy of clusters exists.
- Hierarchical clustering - unlike flat clustering - tries to find a hierarchy of clusters in a given dataset.

this type of clustering is more like 3-D so that it is not flat like kmeans and DBSCAN

Clustering - Hierarchical clustering

Idea of hierarchical clustering

- We initialize each point to be a cluster of its own.
- We iteratively join the two most similar points in the dataset.
- We stop when only we have combined all points into 1 cluster.

Needed: A **similarity measure between clusters** to decide which clusters are most similar.

Clustering - Hierarchical clustering

Single link (Florek et al., 1951)

$$d_{single} = \min\{d(\mathbf{x},\mathbf{x}')|\mathbf{x} \in C, \mathbf{x}' \in C'\}$$

Average link

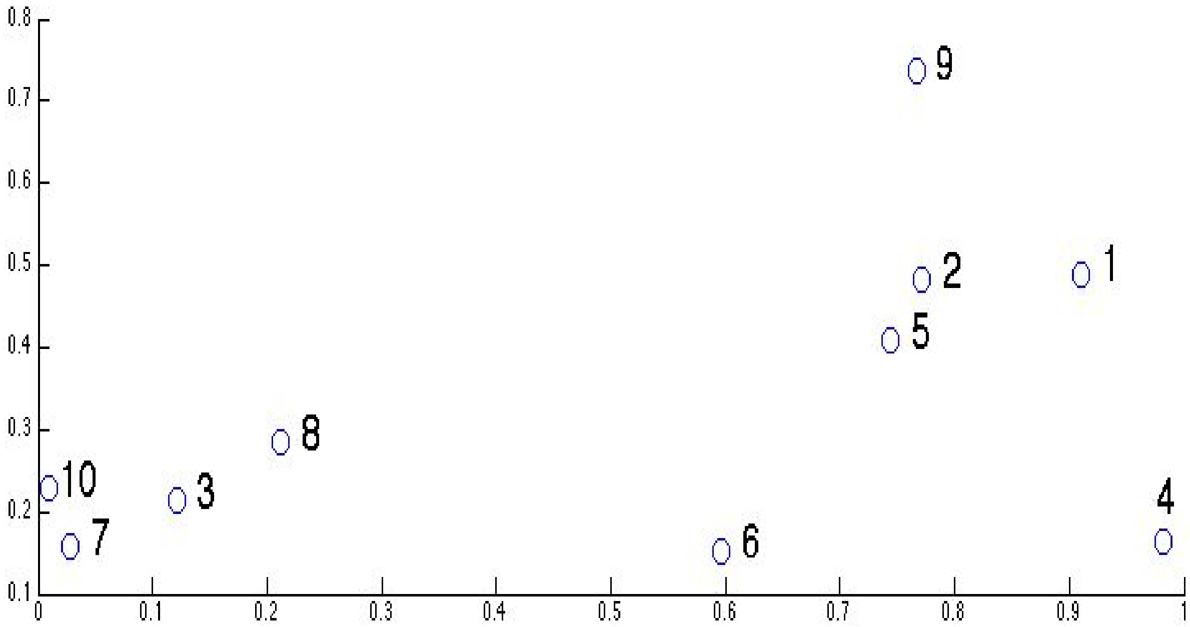
$$d_{average} = \text{mean}\{d(\mathbf{x},\mathbf{x}')|\mathbf{x} \in C, \mathbf{x}' \in C'\}$$

Complete link

$$d_{complete} = \max\{d(\mathbf{x},\mathbf{x}')|\mathbf{x} \in C, \mathbf{x}' \in C'\}$$

Clustering - Hierarchical clustering

Example



Clustering - Hierarchical clustering

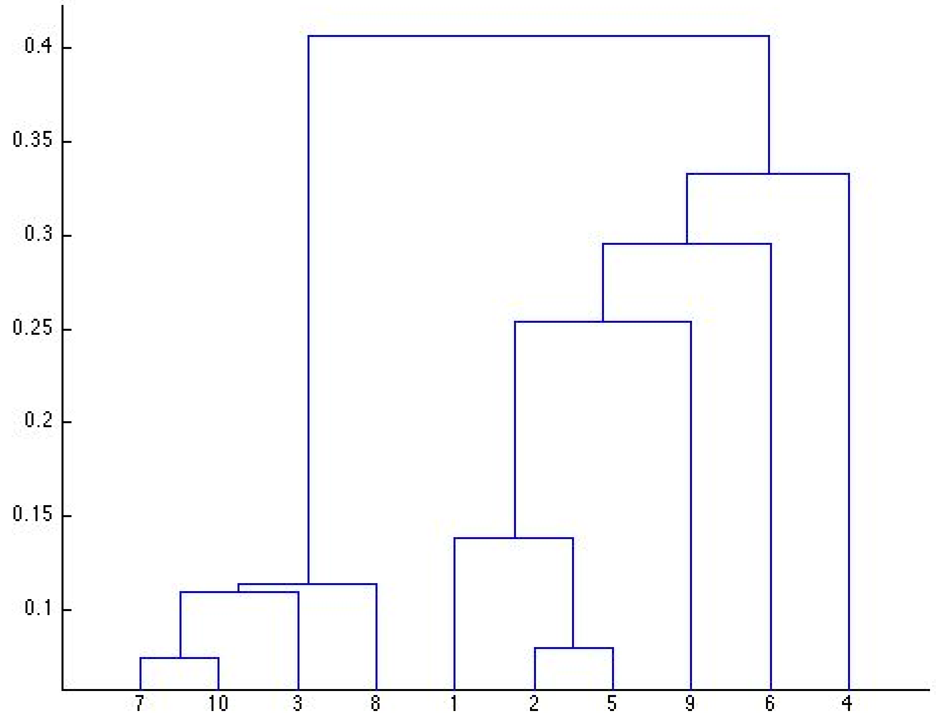
MATLAB

z =rand(10,2)

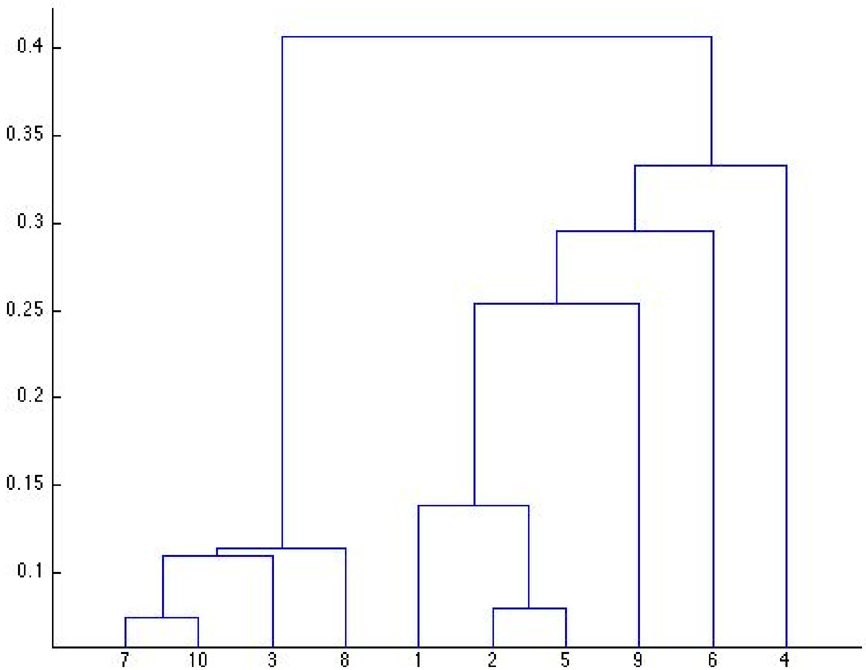
d = pdist(z)

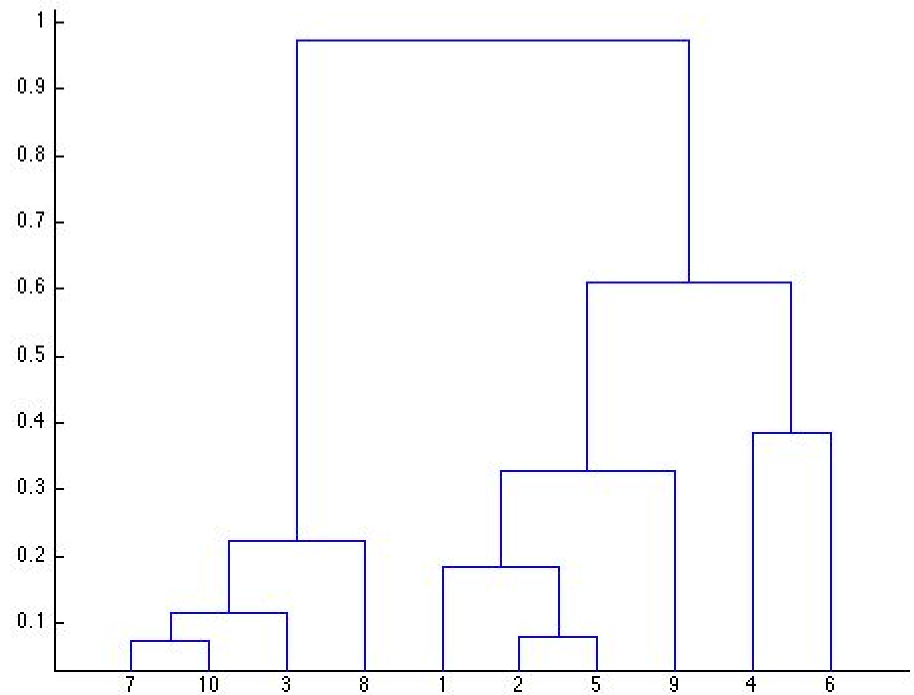
l =linkage(d)

dendrogram(l)



Clustering - Hierarchical clustering





l =linkage(d)

l =linkage(d,'complete')

Clustering - Hierarchical clustering

Advantages

More insight into data structure: non-flat clustering, full hierarchy of clusters

Disadvantages

Desired output for further use is often a flat clustering - where to cut the hierarchy is unclear

Clustering: Summary

- Clustering finds groups of similar objects in a given dataset.

- The three most popular families of clustering algorithms are
  - centroid-based clustering
  - graph-based clustering (including density-based clustering)
  - hierarchical clustering

- When applying these algorithms, it is essential:
  - to be aware of the strengths and weaknesses of these algorithms
  - and to report the exact parameter settings used (e.g. number of clusters, distance function used)

References

Ester, Martin, Hans-Peter Kriegel, Jörg S, und Xiaowei Xu (1996). „A density-based algorithm for discovering clusters in large spatial databases with noise“, SIGKDD 1996, 226–31..

Florek, K., Łukaszewicz, J., Perkal, J., Steinhaus, H., & Zubrzycki, S. (1951). Sur la liaison et la division des points d'un ensemble fini. In Colloquium Mathematicae (Vol. 2, No. 3-4, pp. 282-285). Institute of Mathematics Polish Academy of Sciences.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, et al (1999). „Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring“. *Science (New York, N.Y.)* 286, Nr. 5439 (October 15, 1999): 531–37.

Lloyd, S. P. (1957). "Least square quantization in PCM". *Bell Telephone Laboratories Paper*. Journal version: Lloyd., S. P. (1982). "Least squares quantization in PCM". *IEEE Transactions on Information Theory* **28** (2): 129–137

Rousseeuw, P.J. (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". *Computational and Applied Mathematics* **20**: 53–65.

Steinhaus, H. (1957). "Sur la division des corps matériels en parties". Bull. Acad. Polon. Sci. (in French) 4 (12): 801–804.

Appendix: k-means - Number of clusters

Silhouette Plots

One strategy to select k is to examine **silhouette coefficients**:

A **silhouette coefficient s(p)** (Rousseeuw, 1987) relates the average distance between a point p and and all others points from its cluster C, *d(p,C)*, to the average distance between a point p and the other points from the second nearest cluster C', *d(p,C')*:

$$s(\mathbf{p}) = \frac{d(\mathbf{p}, C') - d(\mathbf{p}, C)}{\max(d(\mathbf{p}, C), d(\mathbf{p}, C'))}$$

s(p) is close to 1, if a point is clearly located in its cluster C.

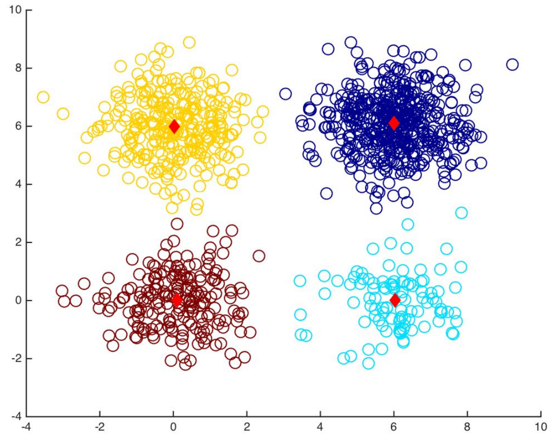
s(p) is close to 0, if a point is located between two clusters.

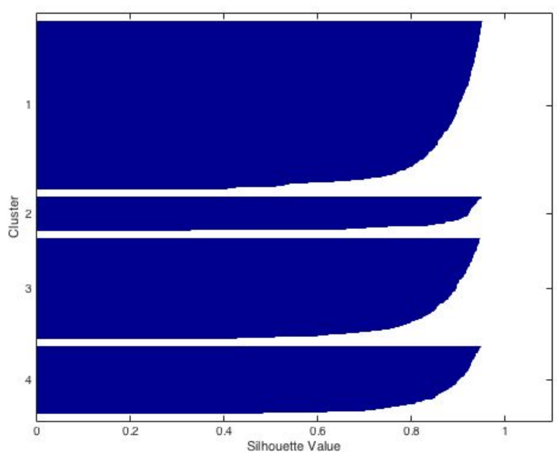
s(p) is negative, if it is closer to another than its current cluster.

Appendix: k-means - Number of clusters

Silhouette Plots(matlab function: silhouette):

k=4





k=5