Lecture notes

Knowledge Discovery in Databases

Summer Semester 2012

Lecture 8: Clustering II

Lecture: Dr. Eirini Ntoutsi
Tutorials: Erich Schubert

http://www.dbs.ifi.lmu.de/cms/Knowledge_Discovery_in_Databases_I_(KDD_I)

Sources

- Previous KDD I lectures on LMU (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Jörg Sander, Matthias Schubert, Arthur Zimek)

- Tan P.-N., Steinbach M., Kumar V., *Introduction to Data Mining*, Addison-Wesley, 2006

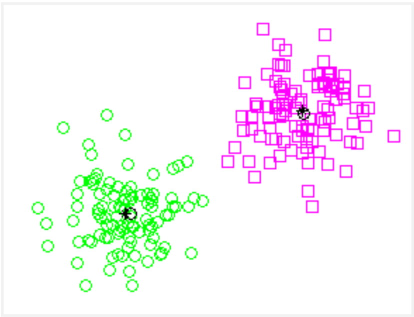
- Jiawei Han, Micheline Kamber and Jian Pei, *Data Mining: Concepts and Techniques, 3rd ed.,* Morgan Kaufmann, 2011.

Outline

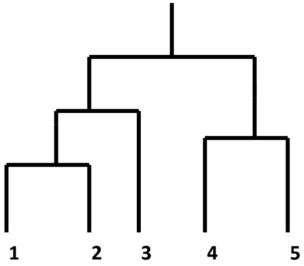
- Introduction
- A categorization of major clustering methods
- Hierarchical methods
- Density based methods
- Grid based methods (next lecture)
- Model-based methods (next lecture)
- Things you should know
- Homework/tutorial

Major clustering methods I

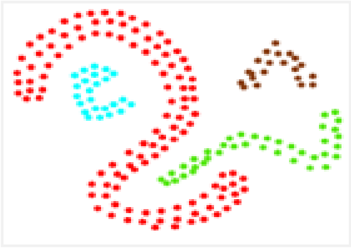
- Partitioning approaches:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS



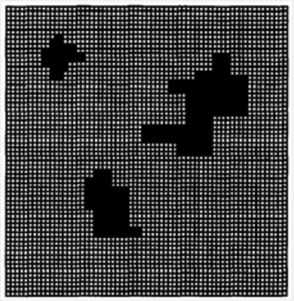
- Hierarchical approaches:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CHAMELEON



- Density-based approaches:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue



Major clustering methods II



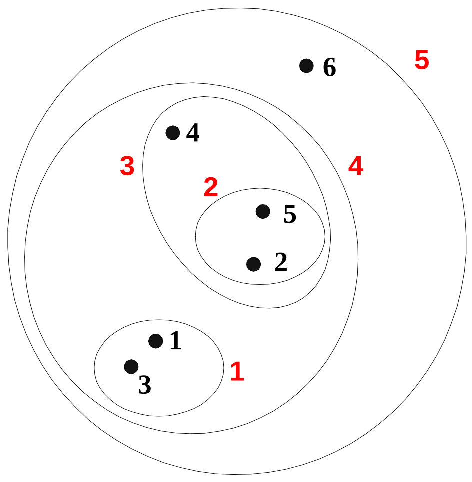
- Grid-based approaches:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE
- Model-based approaches:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based approaches:
  - Based on the analysis of frequent patterns
  - Typical methods: pCluster
- User-guided or constraint-based approaches:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering

Outline

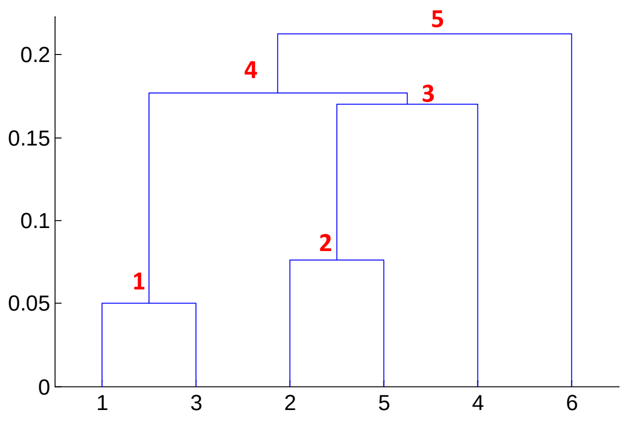
- Introduction
- A categorization of major clustering methods
- Hierarchical methods
- Density based methods
- Grid based methods (next lecture)
- Model-based methods (next lecture)
- Things you should know
- Homework/tutorial

Hierarchical methods idea

- 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
  - The height at which two clusters are merged in the dendrogram reflects their distance



Nested clusters



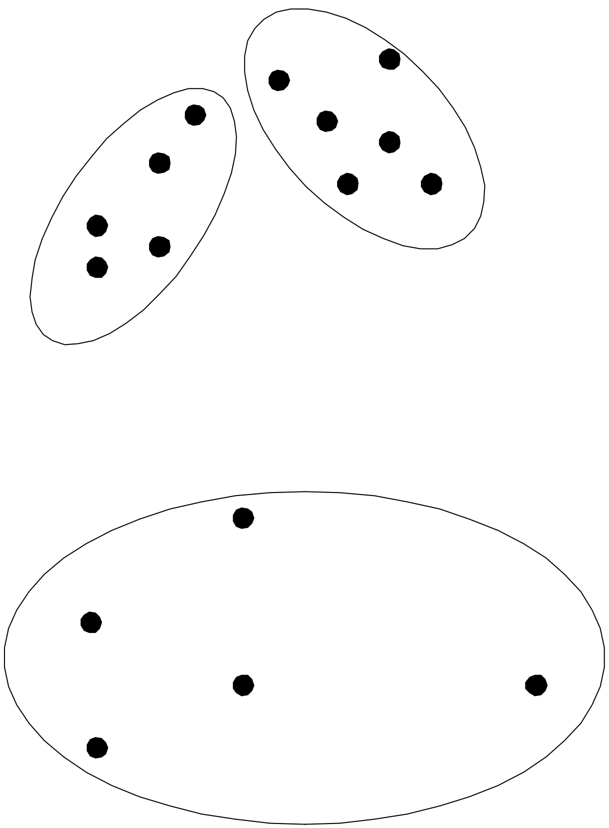
Dendrogram

Strengths of Hierarchical Clustering

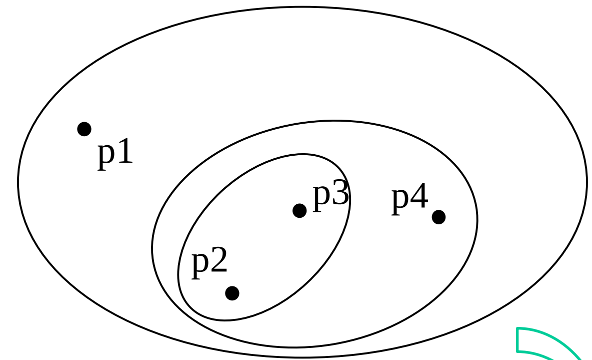
- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level

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

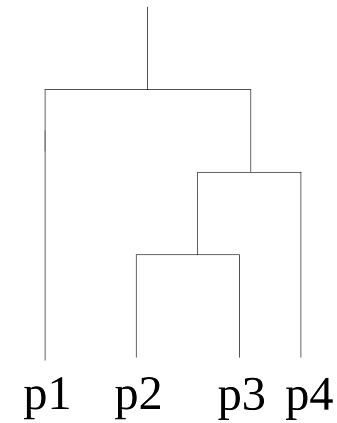
Hierarchical vs Partitioning



Partitioning clustering



Nested clusters



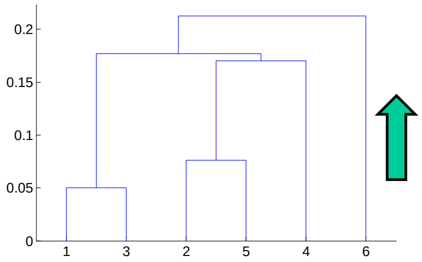
Dendrogram

Partitioning algorithms typically have global objectives

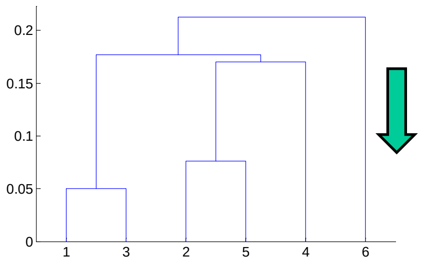
Hierarchical clustering algorithms typically have local objectives

Hierarchical clustering methods

- 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
    - e.g., AGNES



  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are *k* clusters)
    - e.g., DIANA



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

Agglomerative clustering algorithm

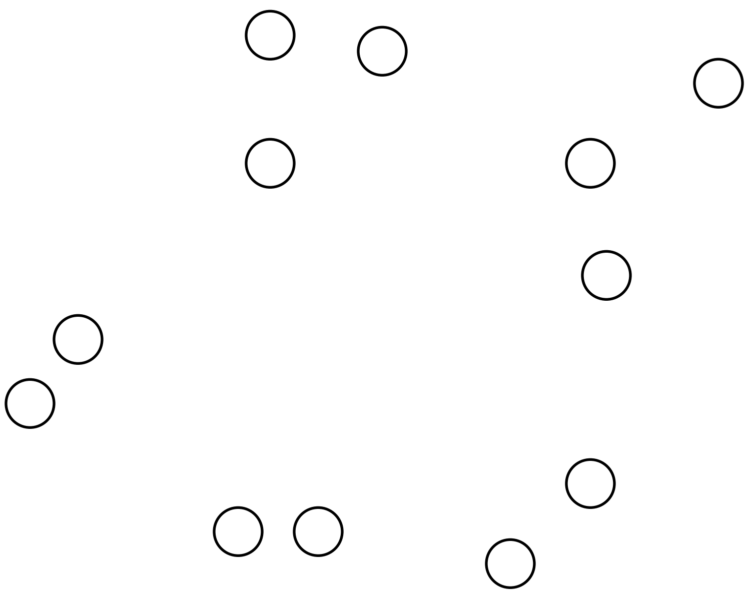
- More popular hierarchical clustering technique
- Basic algorithm is straightforward

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

- Key points:
  - the computation of the proximity of two clusters
    - Different approaches to defining the distance between clusters distinguish the different algorithms (single link, complete link, …..)
  - the update of the proximity matrix due to cluster merges

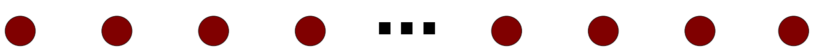
Starting situation

- Start with clusters of individual points and a proximity matrix



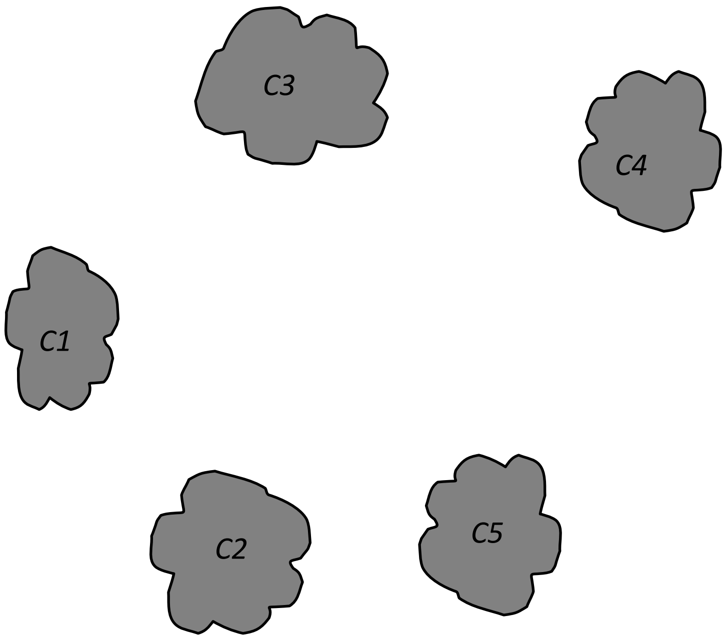
	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix



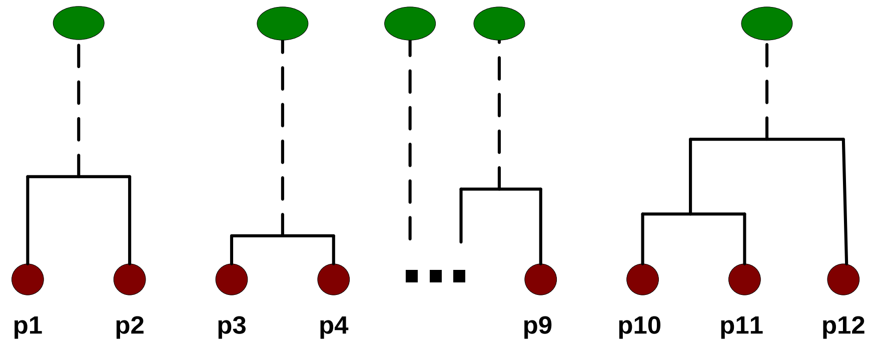
Intermediate situation I

- After some merging steps, we have some clusters



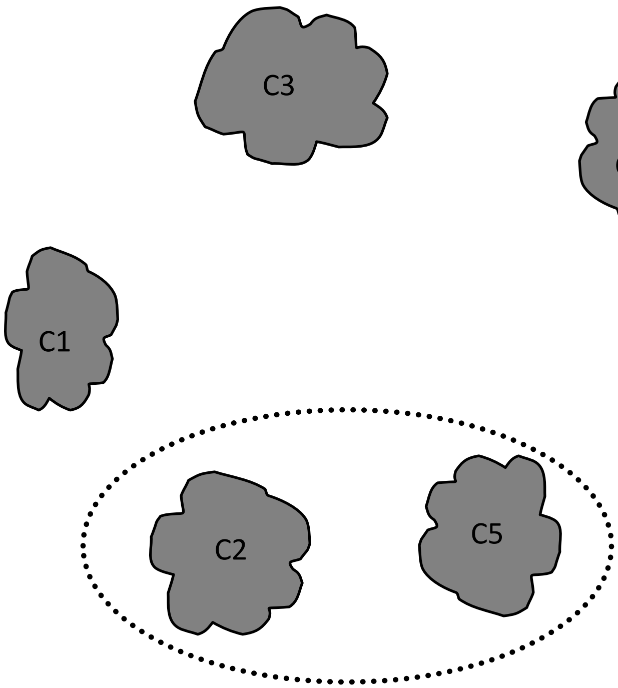
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

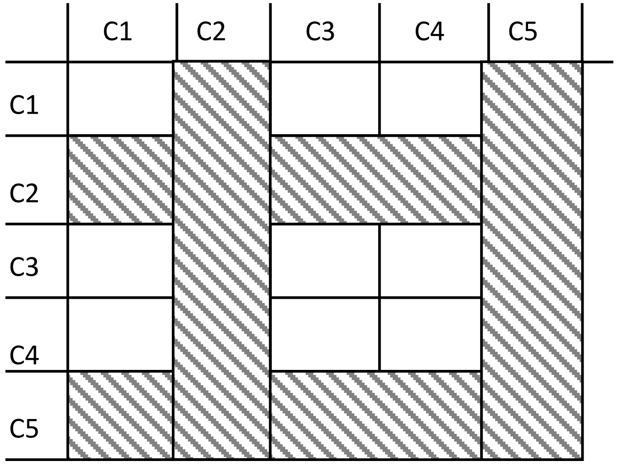
Proximity matrix



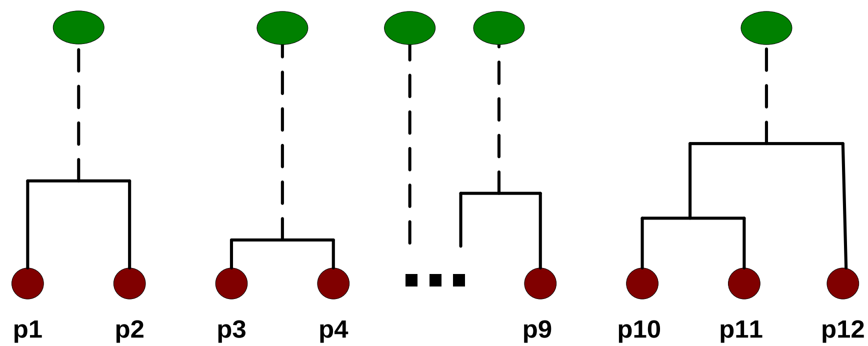
Intermediate situation II

- We want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix.



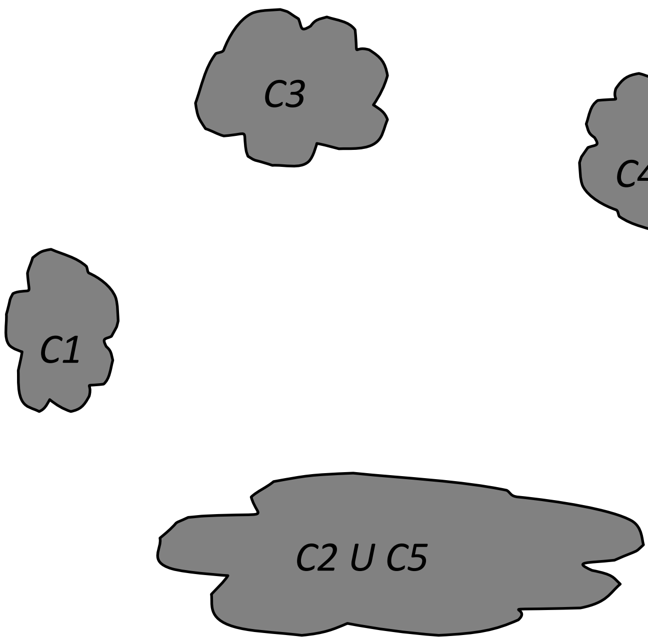


Proximity matrix



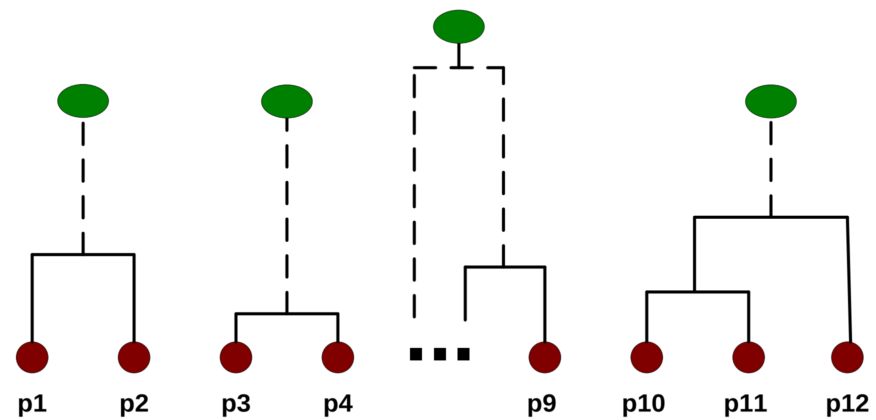
After merging

- The question is "How do we update the proximity matrix?" Or, in other words, what is the similarity between two clusters?

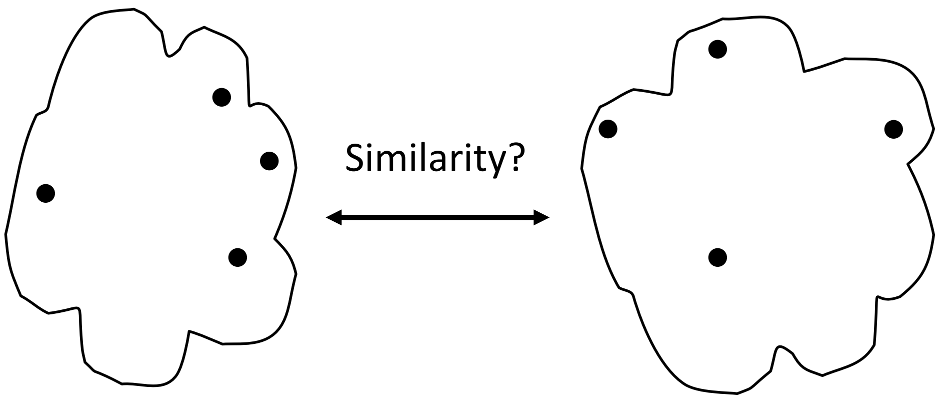


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity matrix



Measures of inter-cluster similarity I



	p1	p2	p3	…	p12	
p1						
p2						
p3						
…						
p12						

Proximity matrix

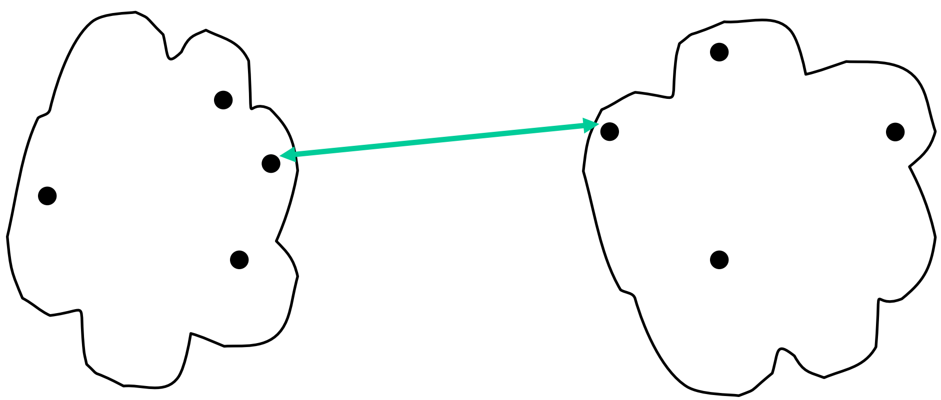
A variety of different measures:

- Single link (or MIN)
- Complete link (or MAX)
- Group average
- Distance between centroids
- Distance between medoids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Measures of inter-cluster similarity II

- Single link (or MIN): smallest distance between an element in one cluster and an element in the other, i.e.,

$$dis_{sl}(C_i, C_j) = \min_{x,y} \{ d(x,y) \mid x \in C_i, y \in C_j \}$$



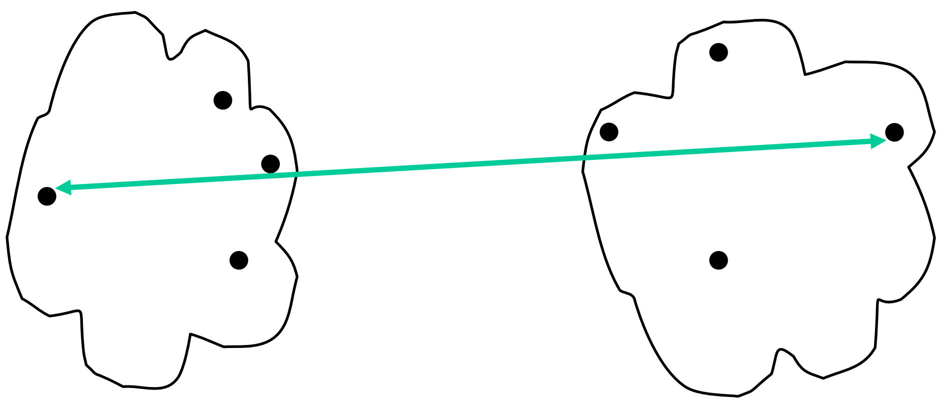
	p1	p2	p3	…	p12	
p1						
p2						
p3						
…						
p12						

Proximity matrix

Measures of inter-cluster similarity III

- Complete link (or MAX): largest distance between an element in one cluster and an element in the other, i.e.,

$$dis_{cl}\left(C_i, C_j\right) = \max_{x,y}\left\{d(x,y) \,\middle|\, x \in C_i, y \in C_j\right\}$$



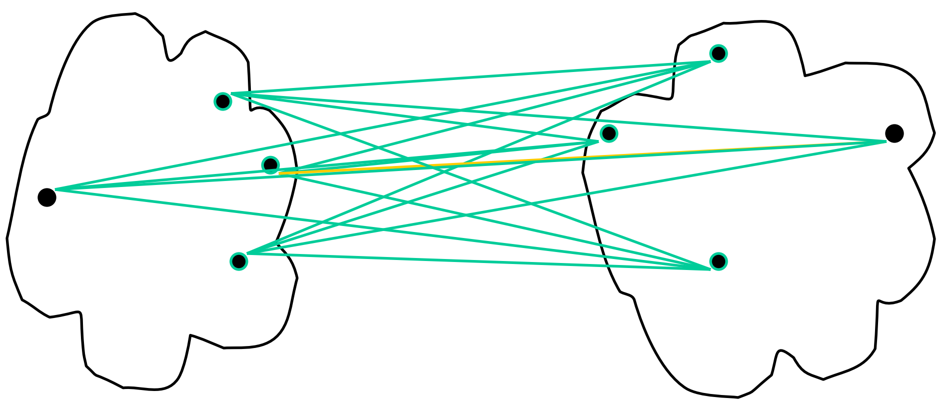
	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix

Measures of inter-cluster similarity IV

- Group average: avg distance between an element in one cluster and an element in the other, i.e.,

$$dis_{avg}\left(C_i, C_j\right) = \frac{\sum_{x \in C_i, y \in C_j} d(x, y)}{\left|C_i\right|\left|C_j\right|}$$



	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix

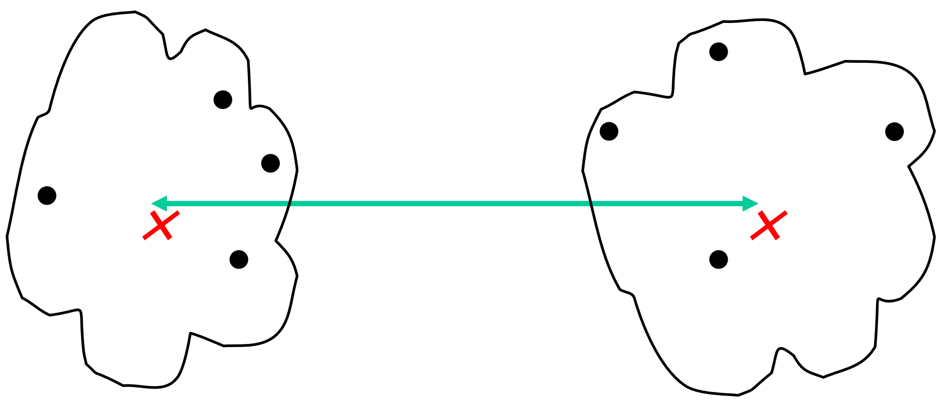
Measures of inter-cluster similarity V

- Centroid: distance between the centroids of two clusters, i.e.,

$$dis_{centroids}(C_i, C_j) = d(c_i, c_j)$$

$$c_m = \frac{\sum_{i=1}^{n} p_i}{n}$$

centroid



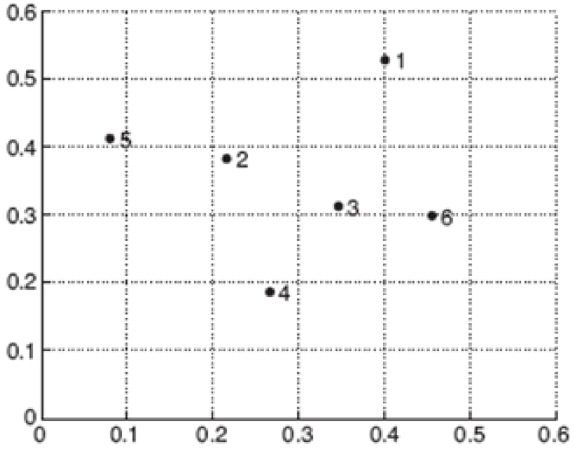
	p1	p2	p3	...	p12
p1					
p2					
p3					
...					
p12					

Proximity matrix

Example

Dataset (6 2D points)

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30



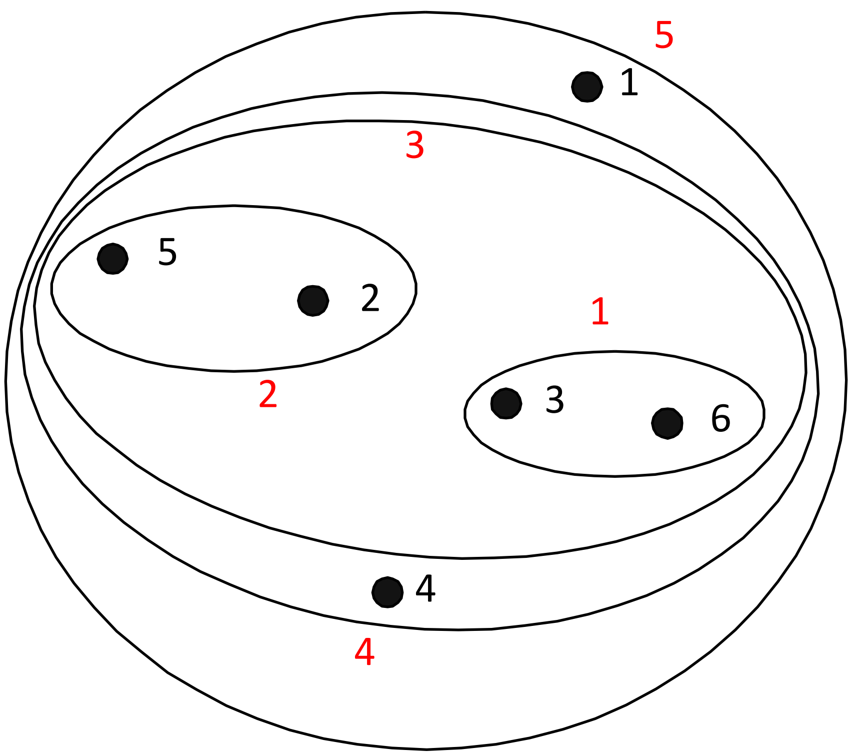
Distance matrix (Euclidean distance)

	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

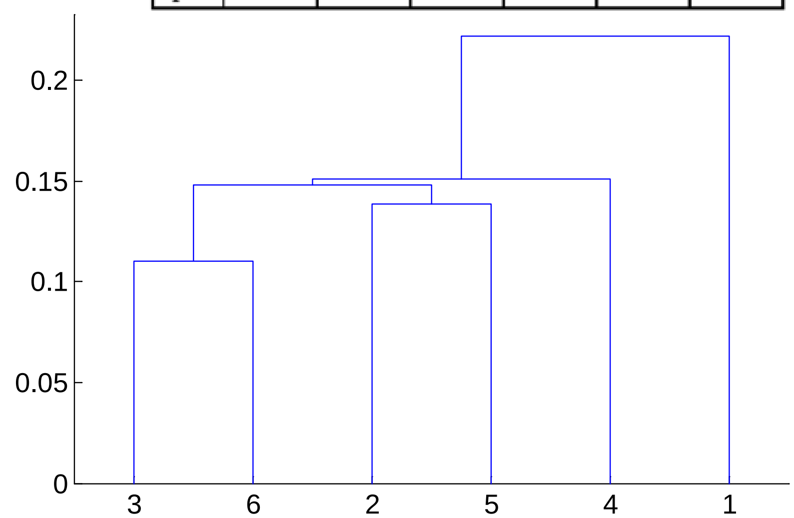
Single link distance (MIN): discussion

- 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

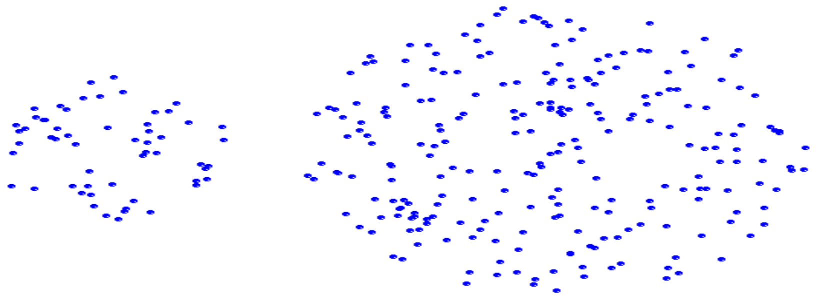


Nested clusters

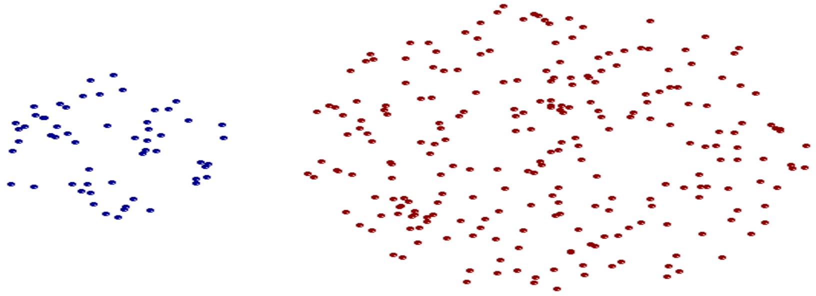


Dendrogram

Single link distance (MIN): strengths



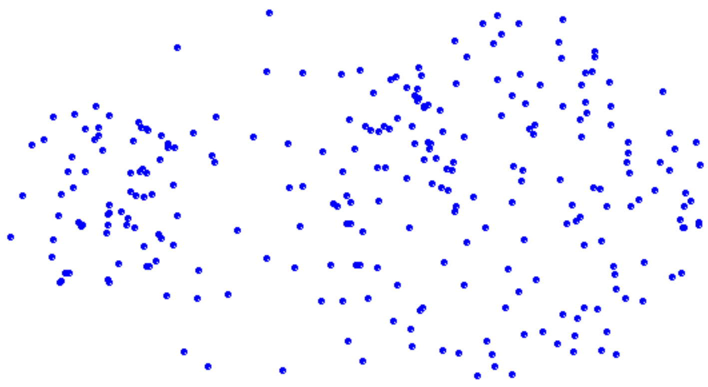
Original points



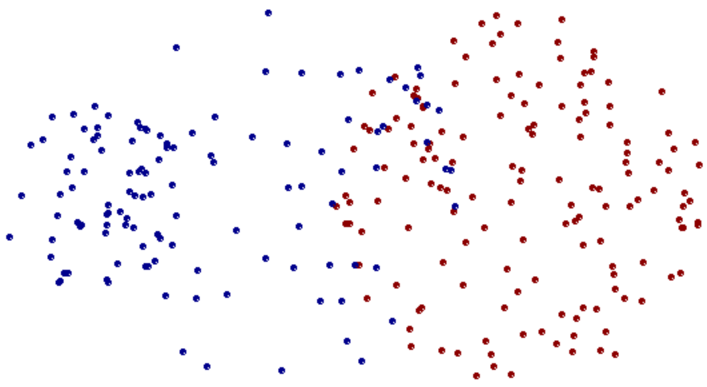
Two clusters

- Can handle non-elliptical shapes

Single link distance (MIN): limitations

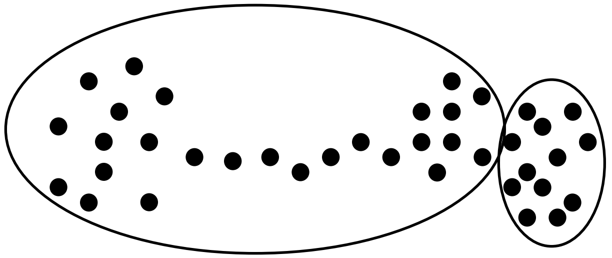


Original points



Two clusters

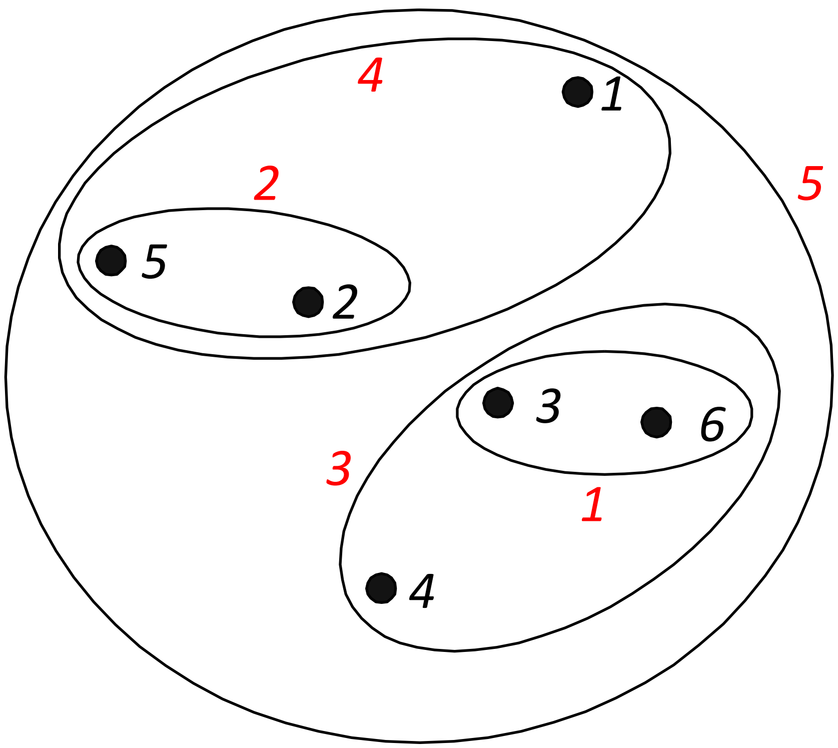
- Sensitive to noise and outliers
- Chain like clusters



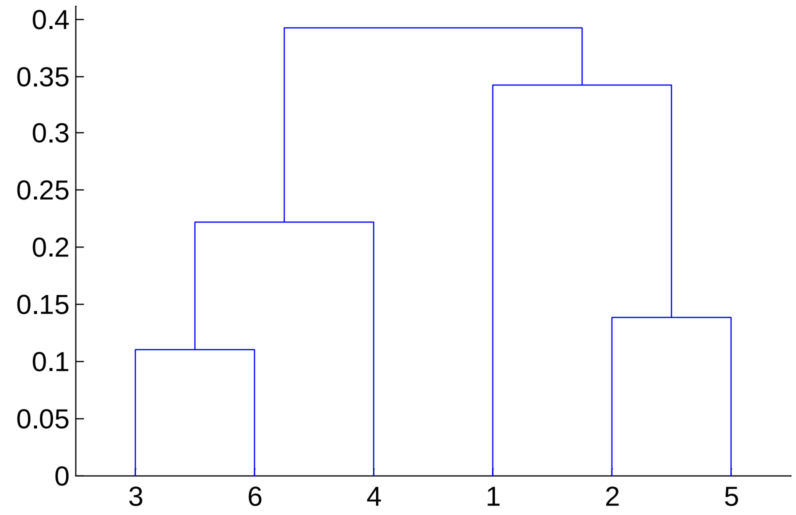
Complete link distance (MAX): discussion

- Similarity of two clusters is based on the two least similar (most distant) 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

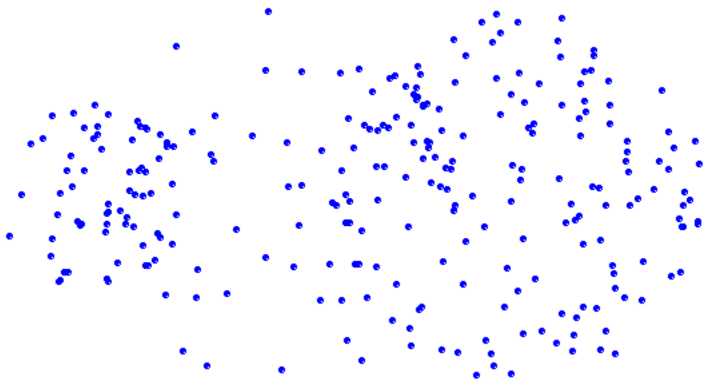


Nested clusters

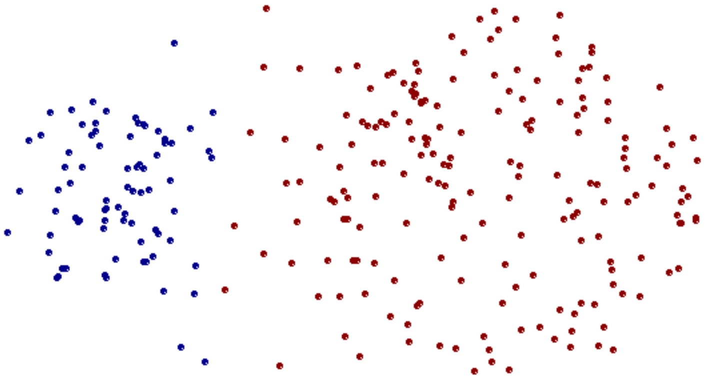


Dendrogram

Complete link distance (MAX): strengths



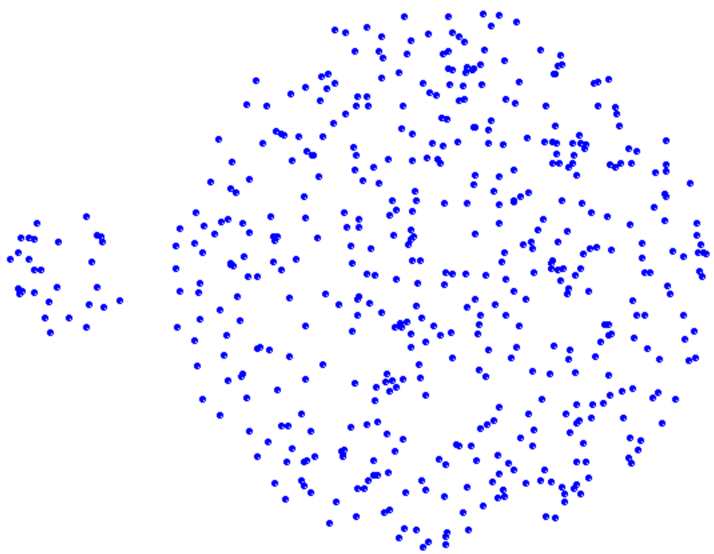
Original points



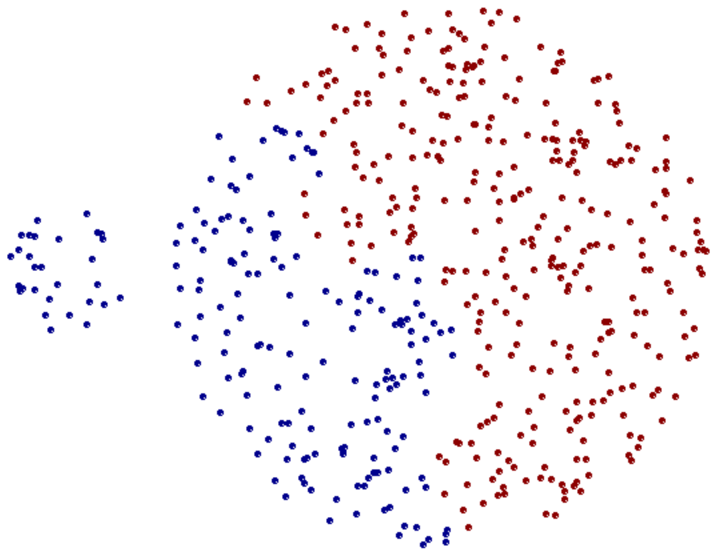
Two clusters

- Less susceptible to noise and outliers

Complete link distance (MAX): limitations



Original points



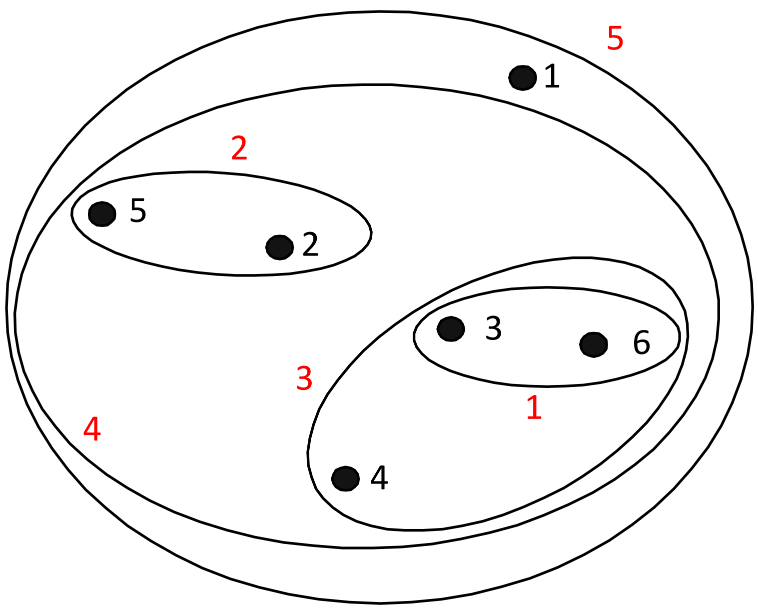
Two clusters

- Tends to break large clusters
- Biased towards spherical clusters

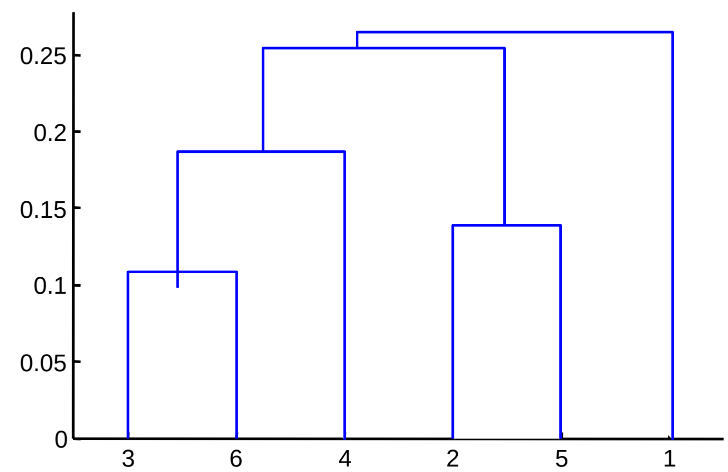
Group average: discussion

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



Nested clusters



Dendrogram

Group average: strengths and limitations

- Compromise between Single and Complete Link

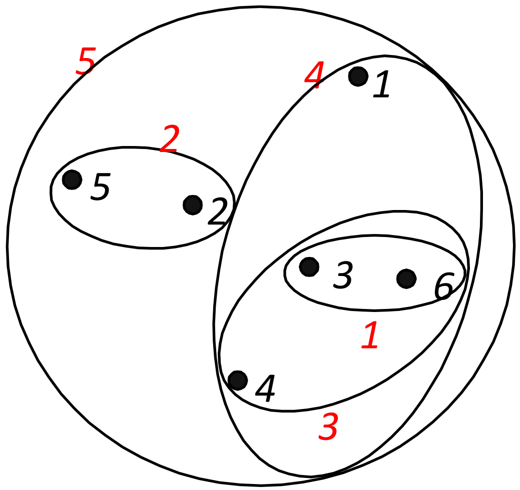
- Strengths
  - Less susceptible to noise and outliers

- Limitations
  - Biased towards spherical clusters

Ward's method

- Ward's method or Ward's minimum variance method
- The proximity between two clusters is measured in terms of the increase in SSE that results from merging the two clusters
  - At each step, merge the pair of clusters that leads to minimum increase in total inter-cluster variance after merging.
  - Similarly to k-Means, tries to minimize the sum of square distances of points from their cluster centroids
- Similar to group average if distance between points is distance squared

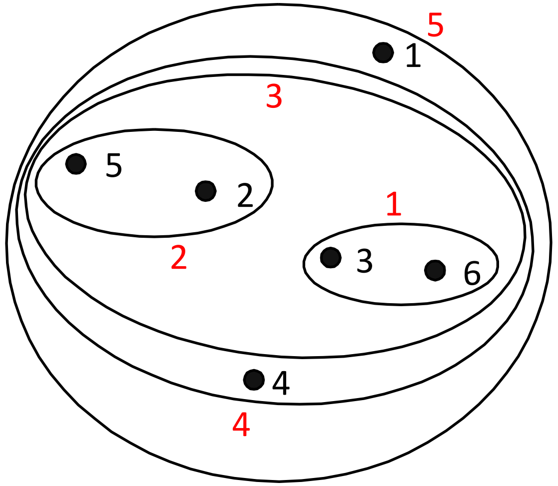
	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00



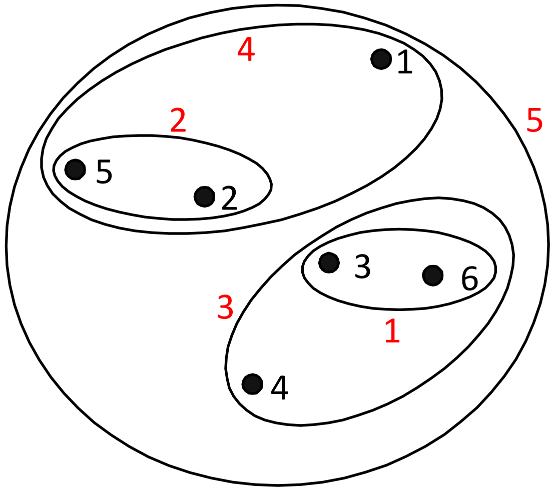
Nested clusters

- Less susceptible to noise and outliers
- Biased towards spherical clusters

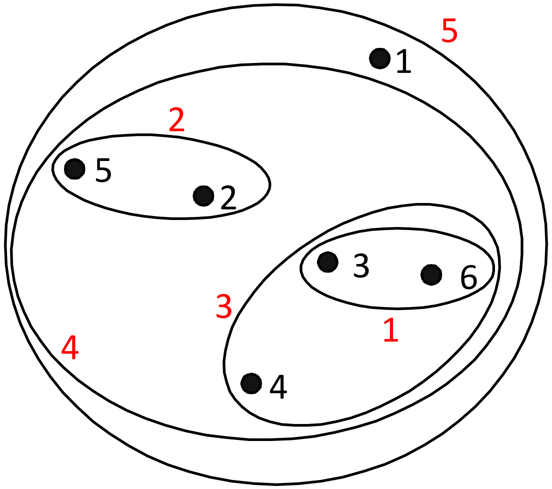
Comparison of the different methods



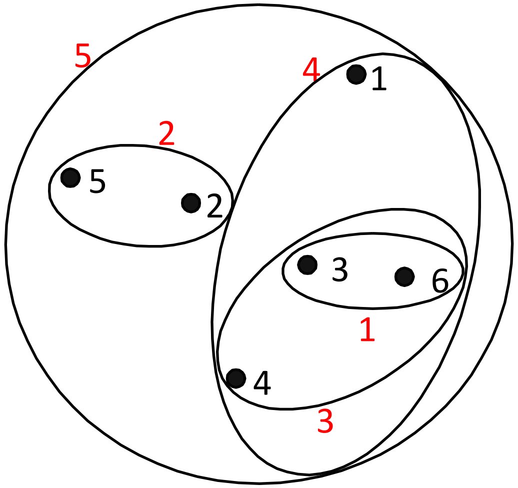
Single link (MIN)



Complete link (MAX)



Group average



Ward's method

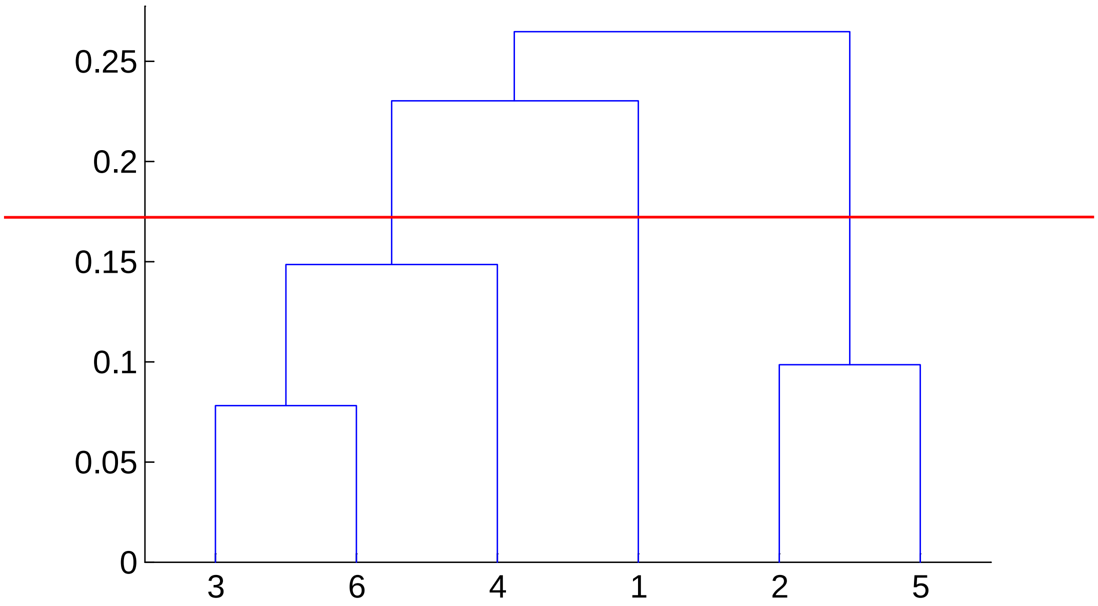
Hierarchical methods: complexity

- $O(N^2)$ 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

How to get a clustering from a dendrogram

- A dendrogram is a tree of clusters.
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.



Hierarchical clustering: overview

- No knowledge on the number of clusters
- Produces a hierarchy of clusters, not a flat clustering
- A single clustering can be obtained from the dendrogram

- Merging decisions are final
  - Once a decision is made to combine two clusters, it cannot be undone
- Lack of a global objective function
  - Decisions are local, at each step
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Breaking large clusters
  - Difficulty handling different sized clusters and convex shapes
- Inefficiency, especially for large datasets

Bisecting k-Means

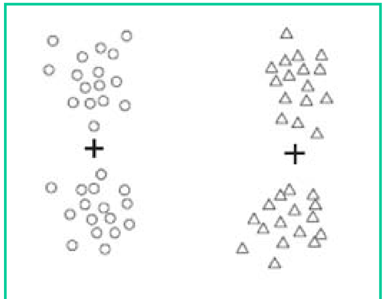
- Hybrid methods: k-Means and hierarchical clustering
- Idea: first split the set of points into two clusters, select one of these clusters for further splitting, and so on, until k clusters.
- Pseudocode:

1. All data constitute one cluster ROOT.
2. The ROOT is partitioned in two clusters, its children, using K-Means for K=2.
3. In each subsequent iteration
    2.1. Choose among the leaf clusters the most inhomogeneous one,
    2.2. Partition it into two clusters with K-Means, K=2,

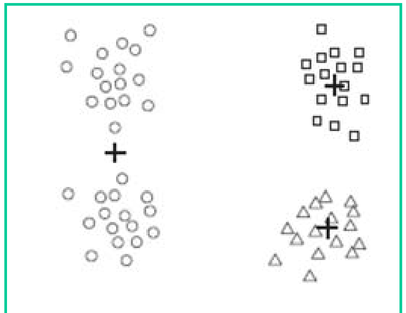
   until K leaf clusters are built.

Which cluster to split?
- e.g., the one with the largest SSE
- e.g., based on SSE and size

- Example:



(a) Iteration 1.



(b) Iteration 2.

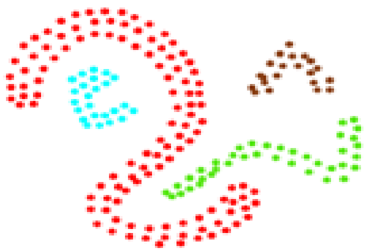
(c) Iteration 3.

Outline

- Introduction
- A categorization of major clustering methods
- Hierarchical methods
- Density based methods
- Grid based methods (next lecture)
- Model-based methods (next lecture)
- Things you should know
- Homework/tutorial

Density based clustering

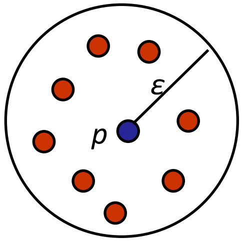
- Clusters are regions of high density surrounded by regions of low density (noise)
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition



- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

The notion of density

- Density:
  - Density is measured locally in the Eps-neighborhood (or ε-neighborhood) of each point
  - Density = number of points within a specified radius Eps (point itself included)



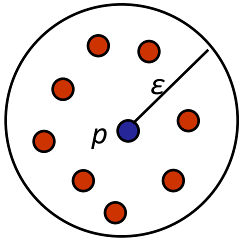
The e-neighborhood of p: 9 points

- Density depends on the specified radius
  - In an extreme small radius, all points will have a density of 1 (only themselves)
  - In an extreme large radius, all points will have a density of N (the size of the dataset)

DBSCAN basic concepts

- Consider a dataset D of objects to be clustered
- Two parameters*:*
  - Eps (or ε): Maximum radius of the neighbourhood
  - MinPts: Minimum number of points in an Eps-neighbourhood of that point

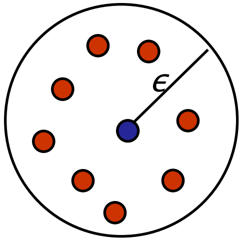
- Eps-neighborhood of a point p in D
  - $N_{Eps}(p)$: {q belongs to D | dist(p,q) <= Eps}



The Eps-neighborhood of p

Core points vs border points vs noise points

- Let D be a dataset. Given a radius parameter Eps and a density parameter MinPts we can distinguish between:

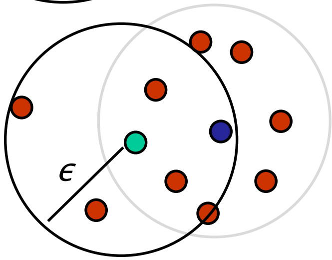


  - Core points

    A point is a core point if it has more than a specified number of points (MinPts) within a specified radius Eps, i.e.,:

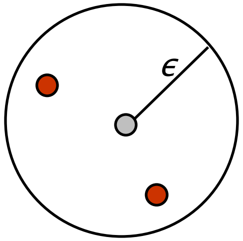
    $|N_{Eps}(p)=\{q \mid dist(p,q) <= Eps\}| \geq MinPts$

    - These are points that are at the interior of a cluster



  - Border points

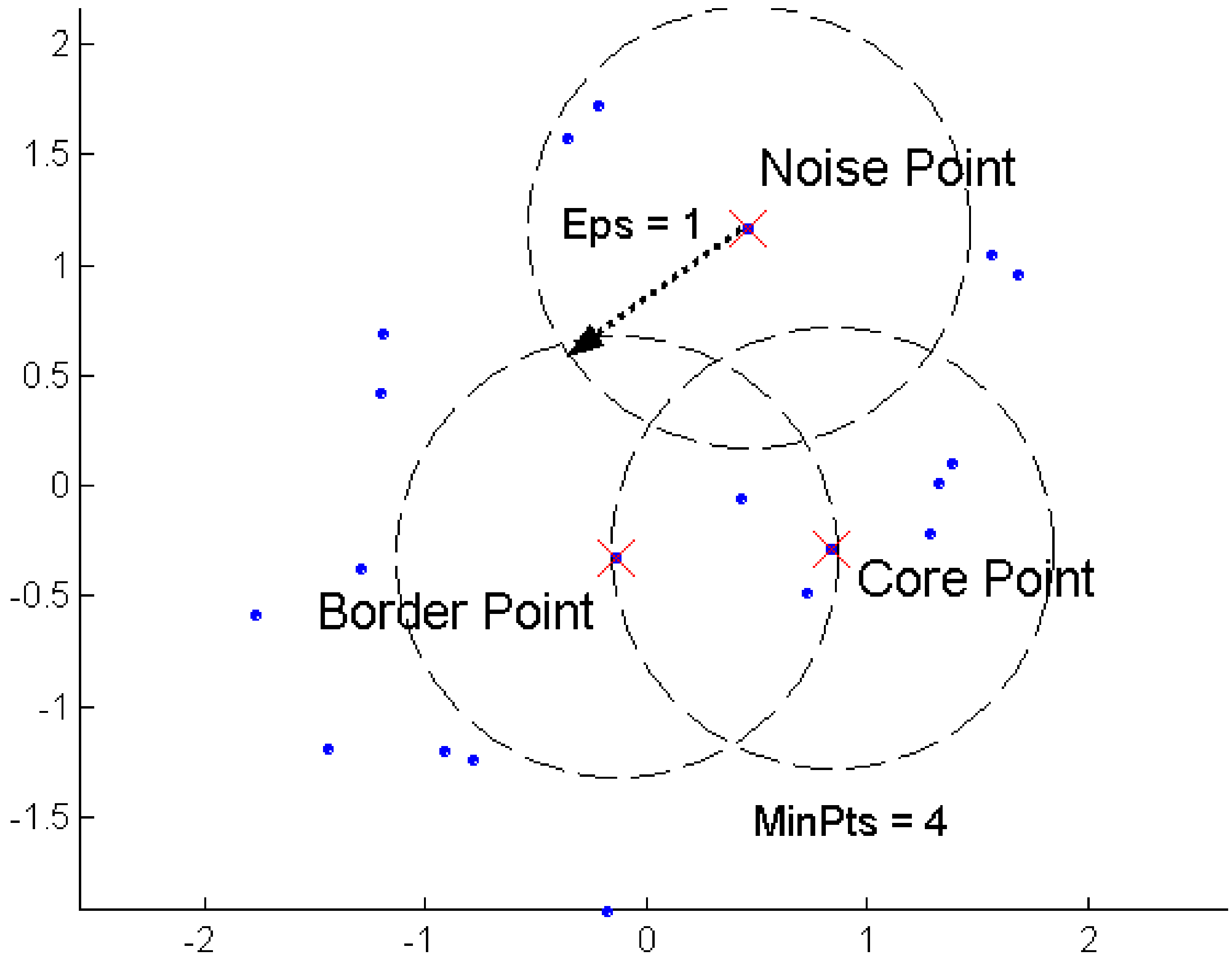
    A border point has fewer than MinPts within Eps, but it is in the neighborhood of a core point



  - Noise points

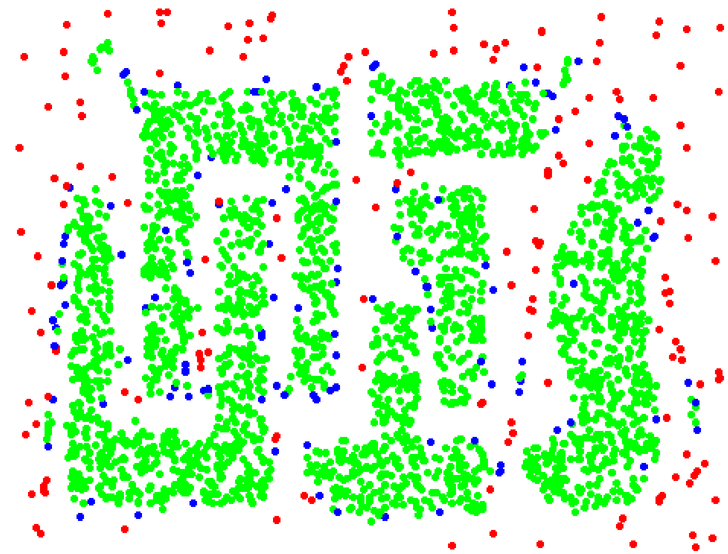
    not a core point nor a border point.

Example


Noise Point
Eps = 1
Border Point
Core Point
MinPts = 4
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
0
1
2

Core, Border and Noise points

Original points

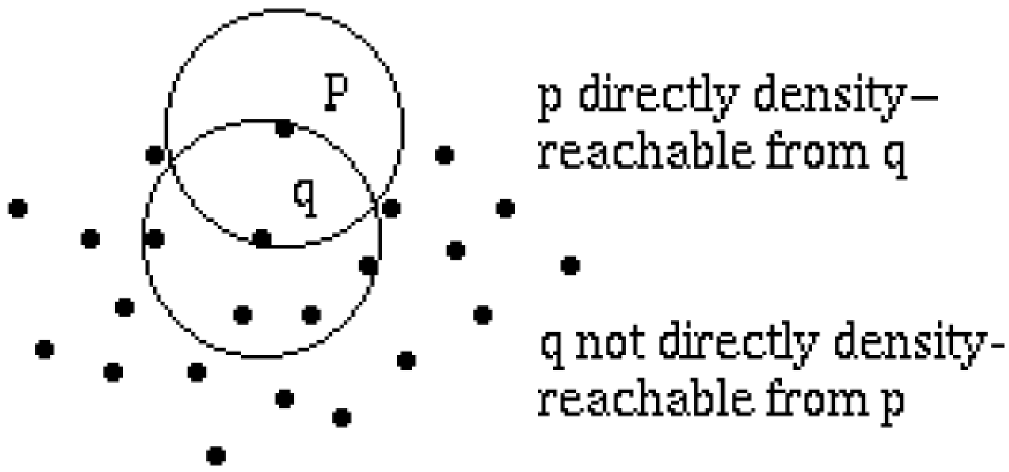


Point types: core, border and noise

Eps = 10, MinPts = 4

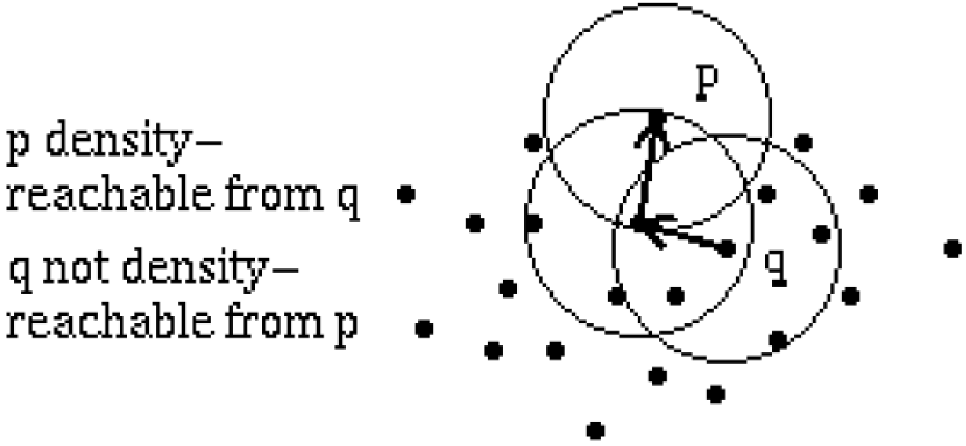
Direct reachability

- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - q is a core point, i.e.,: $|N_{Eps}(q)| >= MinPts$



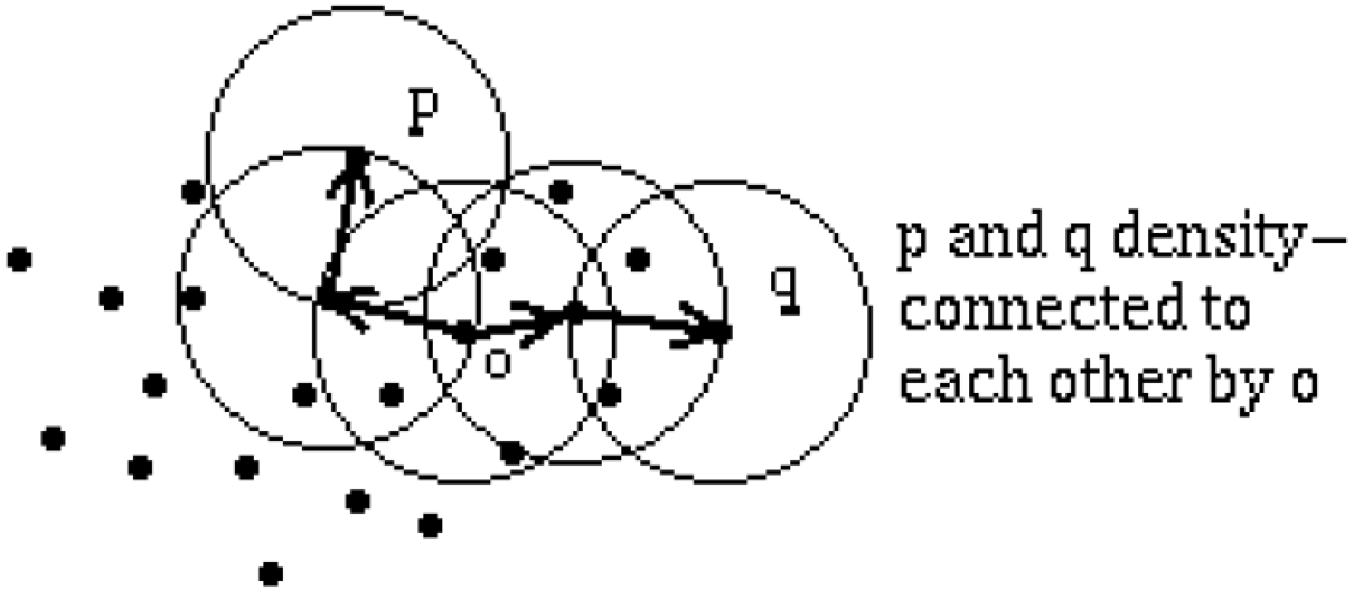
Reachability

- Density-reachable:
  - A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



Connectivity

- Density-connected
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



Cluster

- A cluster is a maximal set of density-connected points

DBSCAN algorithm

- Arbitrary select a point *p*
- Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*.
- If *p* is a core point, a cluster is formed.
- If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

DBSCAN pseudocode I


```
DBSCAN(Dataset DB, Real Eps, Integer MinPts)
   // initially all objects are unclassified,
   // o.ClId = unclassified for all o ∈ DB

   ClusterId := nextId(NOISE);
   for i from 1 to |DB| do
      Object := DB.get(i);
      if Object.ClId = unclassified then
         if ExpandCluster(DB, Object, ClusterId, Eps, MinPts)
         then ClusterId:=nextId(ClusterId);
```

DBSCAN pseudocode II

```
ExpandCluster(DB, StartObject, ClusterId, Eps, MinPts): Boolean
 seeds:= RQ(StartObjekt, Eps);
 if |seeds| < MinPts then // StartObject is not a core object
    StartObject.ClId := NOISE;
     return false;
 else // else: StartObject is a core object
     forall o ∈ seeds do o.ClId := ClusterId;
    remove StartObject from seeds;
    while seeds ≠ Empty do
        select an object o from the set of seeds;
         Neighborhood := RQ(o, Eps);
         if |Neighborhood| ≥ MinPts then // o is a core object
             for i from 1 to |Neighborhood| do
                 p := Neighborhood.get(i);
                 if p.ClId in {UNCLASSIFIED, NOISE} then
                    if p.ClId = UNCLASSIFIED then
                       add p to the seeds;
                    p.ClId := ClusterId;
                end if;
            end for;
         end if;
         remove o from the seeds;
    end while;
 end if
return true;
```

Complexity

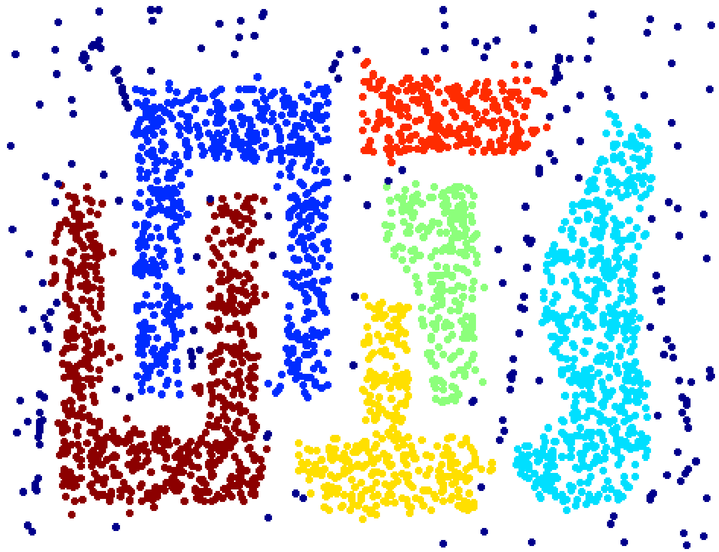
- For a dataset D consisting of n points, the time complexity of DBSCAN is O(n x time to find points in the Eps-neighborhood)

- Worst case $O(n^2)$

- In low-dimensional spaces O(nlogn);
  - efficient data structures (e.g., *kd-trees*) allow for efficient retrieval of all points within a given distance of a specified point

When DBSCAN works well?

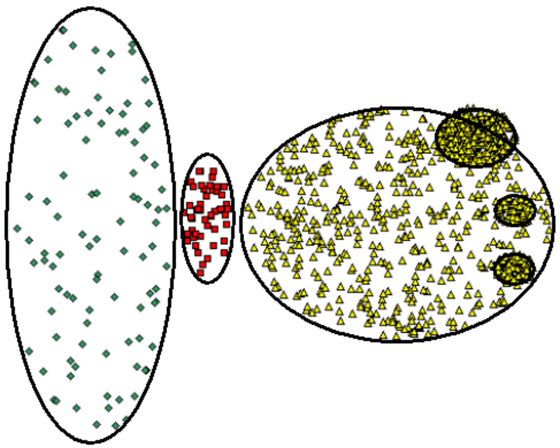
Original points



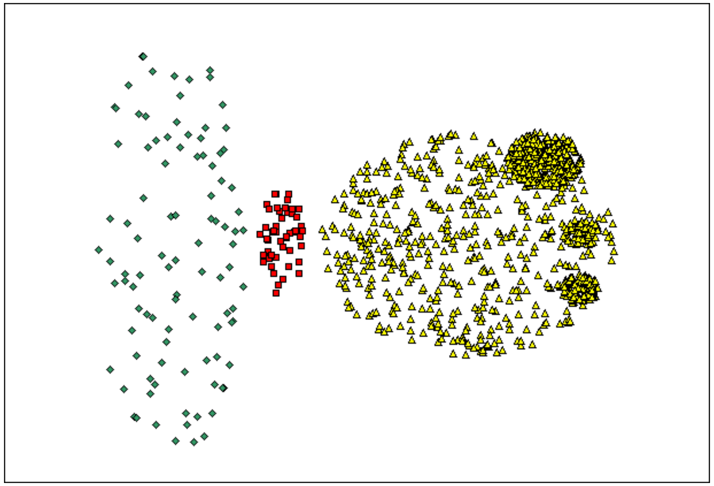
Clusters

- Resistant to noise
- Can handle clusters of different shapes and sizes

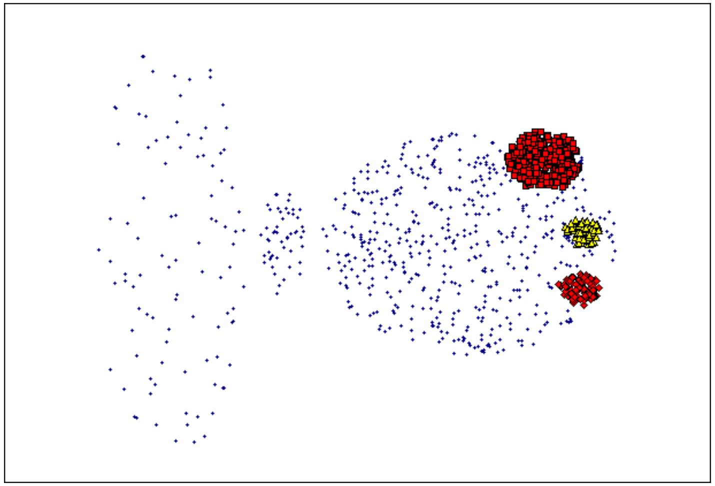
When DBSCAN does not work well?



Original points



(MinPts=4, Eps=9.75).

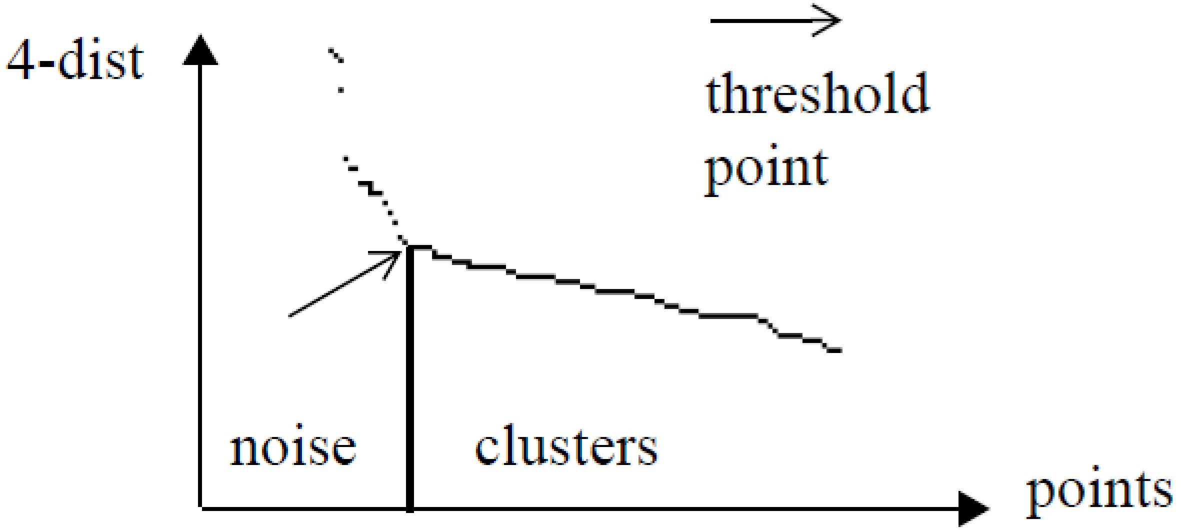


(MinPts=4, Eps=9.92)

- 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



OPTICS

We will discuss OPTICS next time ….

Outline

- Introduction
- A categorization of major clustering methods
- Hierarchical methods
- Density based methods
- Grid based methods (next lecture)
- Model-based methods (next lecture)
- Things you should know
- Homework/tutorial

Things you should know

- Hierarchical methods
  - Agglomerative, divisive
  - Cluster comparison measures
- Bisecting k-Means
- Density based methods
  - DBSCAN

Homework/ Tutorial

Tutorial: Tutorial this Thursday on clustering

Homework:

- Try hierarchical clustering in Weka, Elki
- Implement your own hierarchical clusterer
  - Try the different cluster similarity measures
- Try density based clustering in Elki, Weka
- Implement your own DBSCAN
  - Experiment with different Eps, MinPts parameters

Suggested reading:

- Tan P.-N., Steinbach M., Kumar V., *Introduction to Data Mining*, Addison-Wesley, 2006 (Chapter 8).
- Han J., Kamber M., Pei J. *Data Mining: Concepts and Techniques 3rd ed., Morgan Kaufmann, 2011* (Chapter 10)