Clustering

Basic Concepts and Algorithms 2

Clustering topics

- Hierarchical clustering
- Density-based clustering
- Cluster validity

Proximity measures

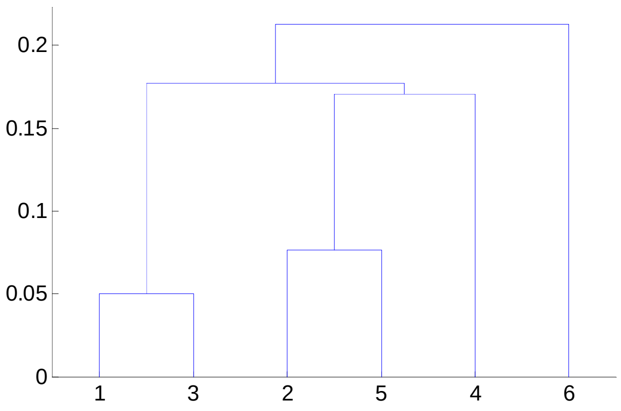
- Proximity is a generic term that refers to either similarity or dissimilarity.
- Similarity
  - Numerical measure of how *alike* two data objects are.
  - Measure is *higher* when objects are *more alike*.
  - Often falls in the range [0, 1].
- Dissimilarity
  - Numerical measure of how *different* two data objects are.
  - Measure is *lower* when objects are *more alike*.
  - Minimum dissimilarity often 0, upper limit varies.
  - Distance sometimes used as a synonym, usually for specific classes of dissimilarities.

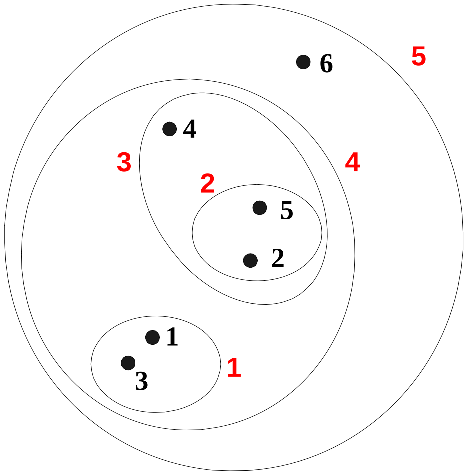
Approaches to clustering

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional clustering
  - Partitional: data points divided into finite number of *partitions* (non-overlapping subsets)
    - each data point is assigned to exactly one subset
  - Hierarchical: data points placed into a set of nested clusters, organized into a *hierarchical tree*
    - tree expresses a continuum of similarities and clustering

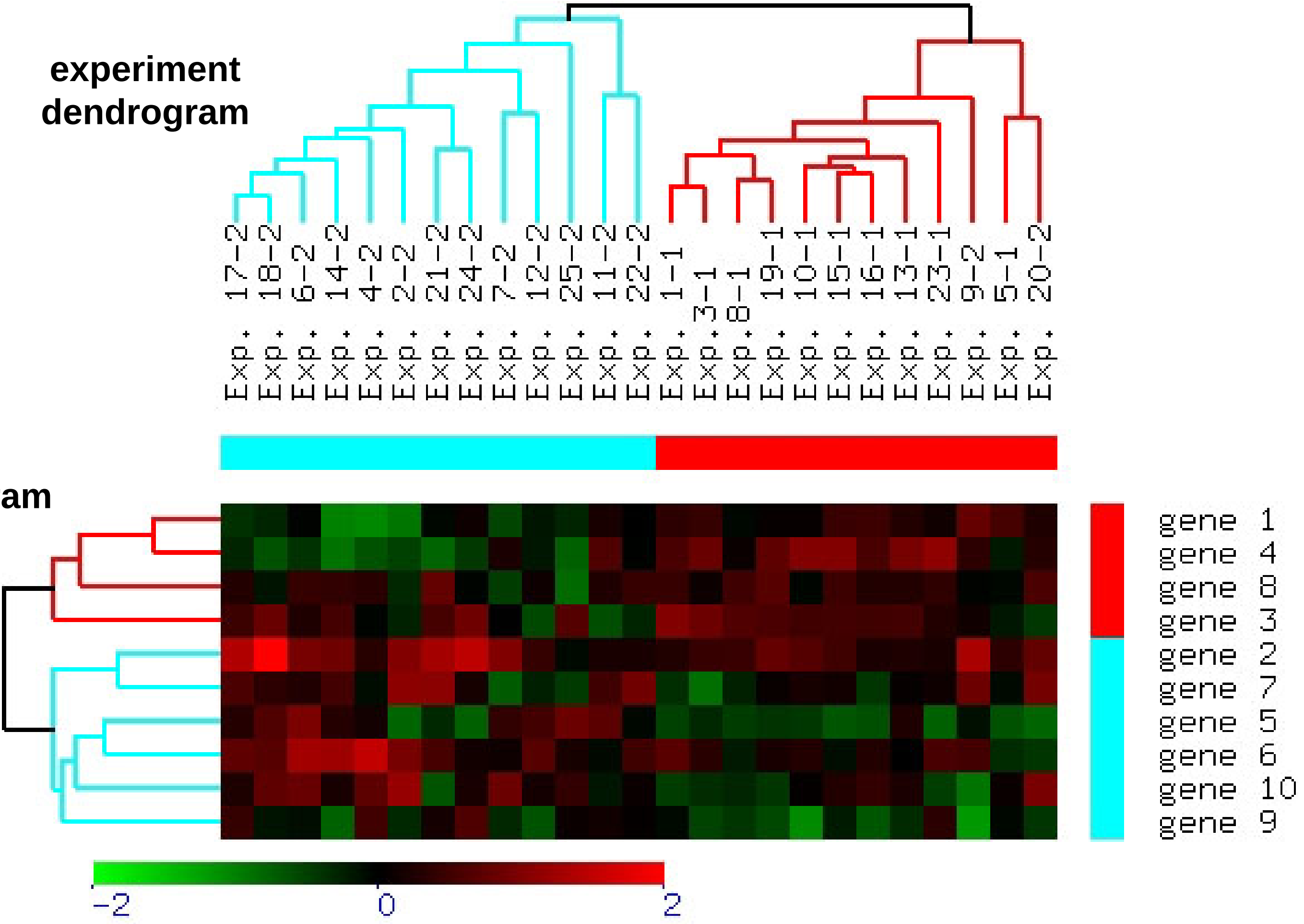
Hierarchical clustering

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





Microarray data analysis

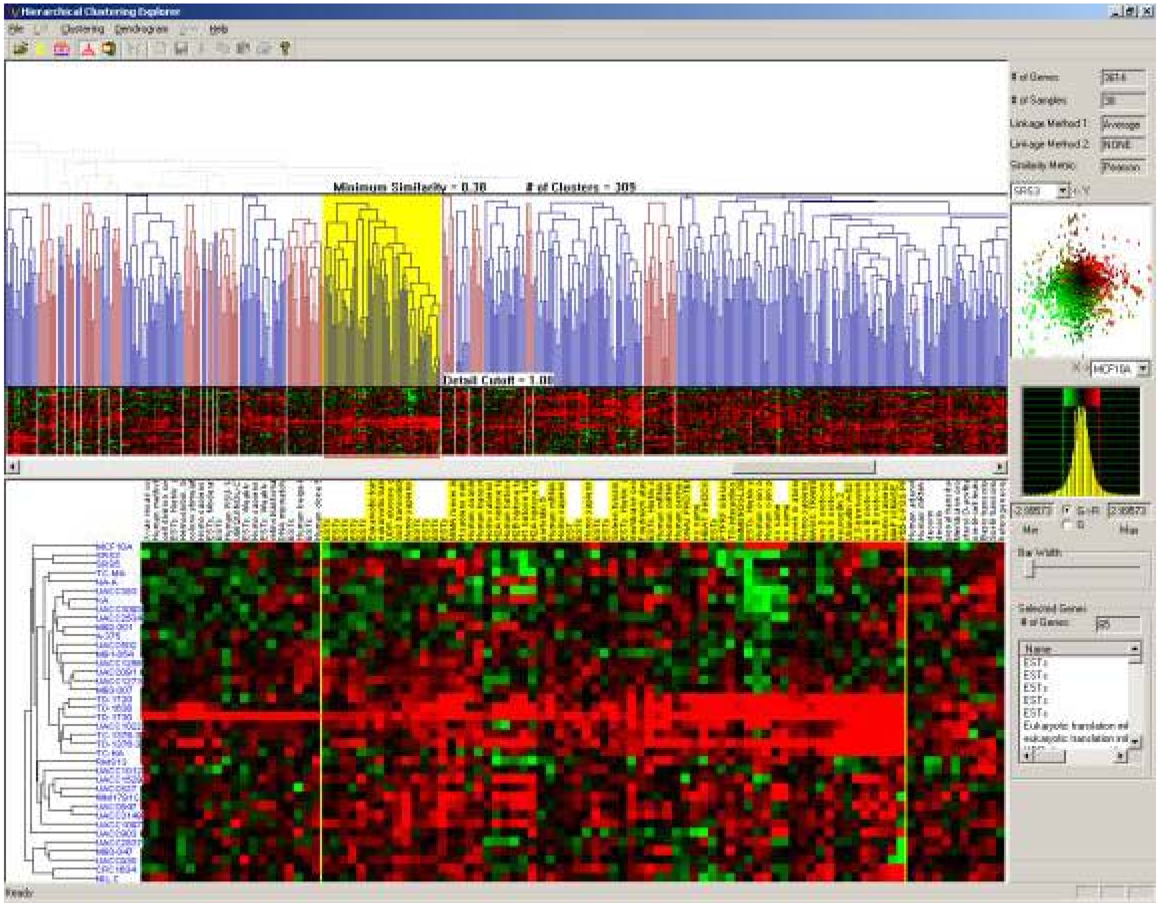


experiment dendrogram

gene dendrogram

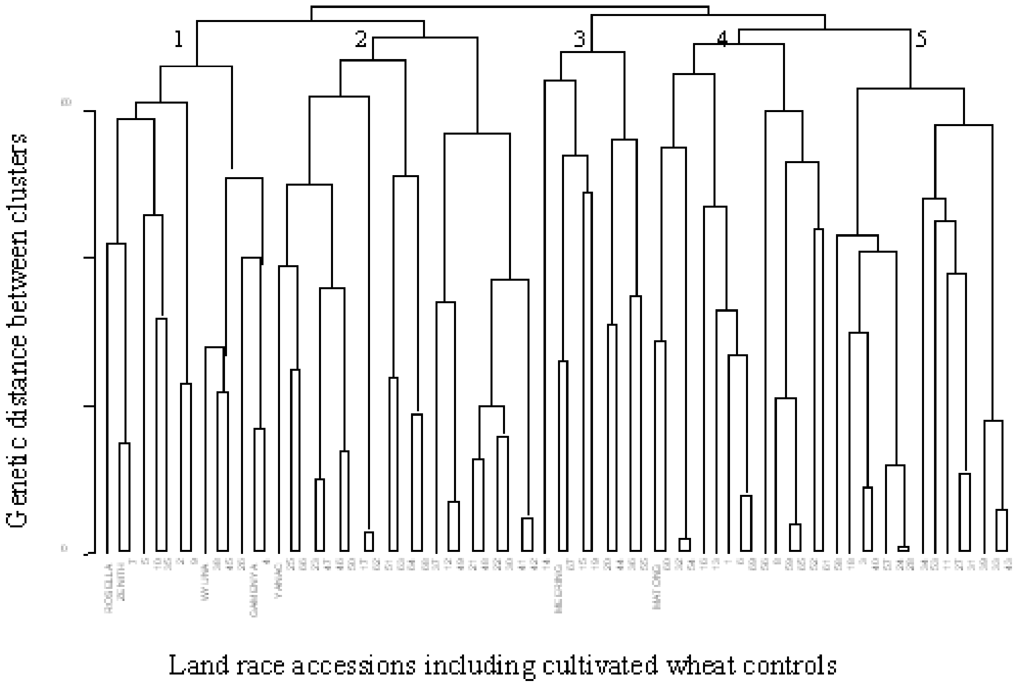
NIH Center for Information Technology

Melanoma gene expression profiles



Univ. of Maryland, Human-Computer Interaction Lab

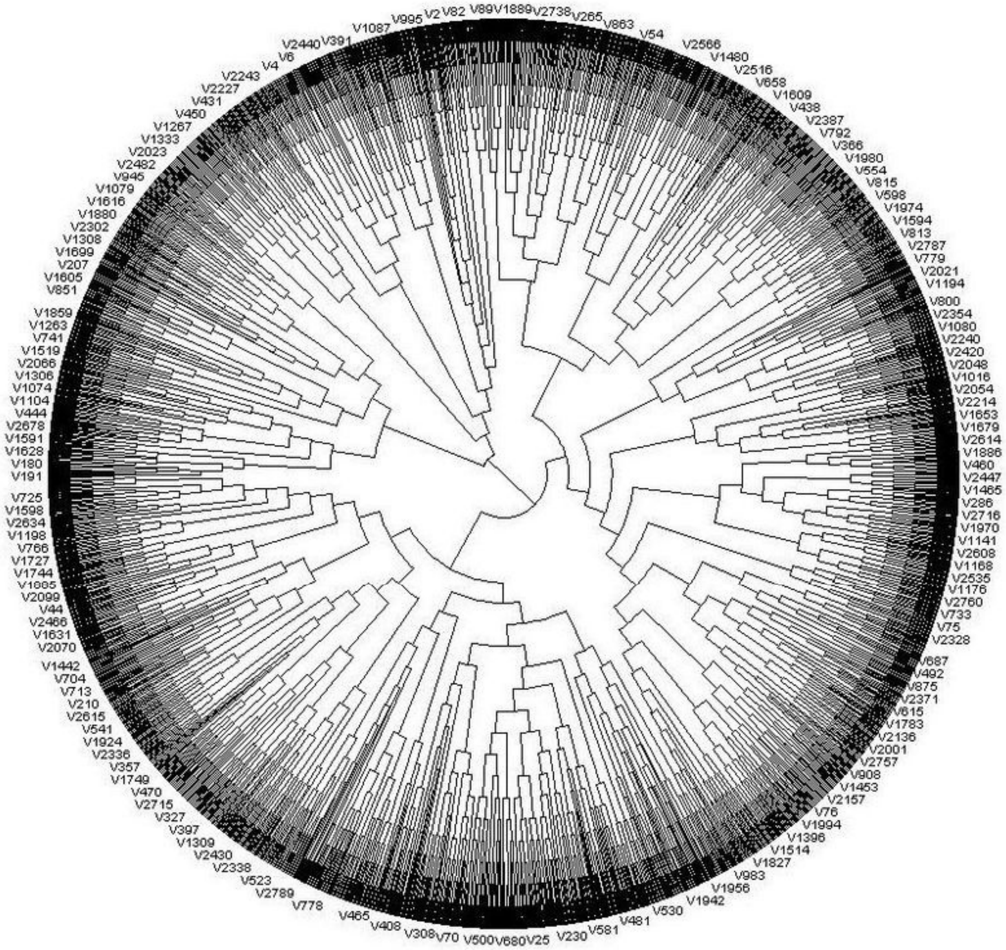
Genetic distance among wheat cultivars



Hierarchical clustering based on 13 quality traits of 75 wheat landraces including seven wheat cultivars.

Australian Society of Agronomy, The Regional Institute Ltd.

Circular cladogram

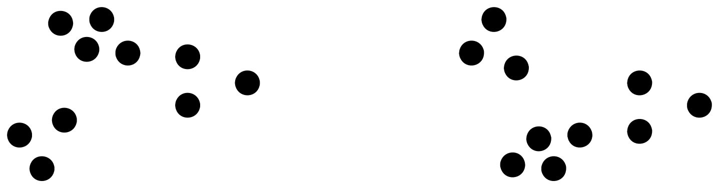


revolutionanalytics.com

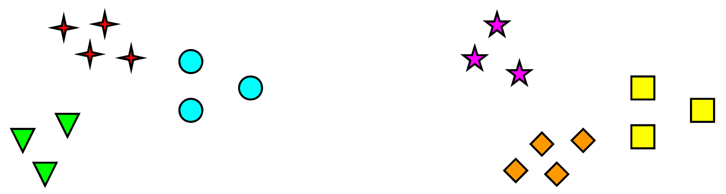
Strengths of hierarchical clustering

- 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, …)

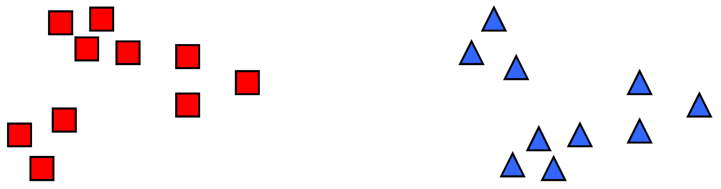
Notion of a cluster can be ambiguous



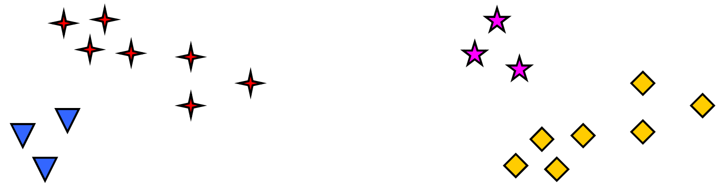
How many clusters?



Six Clusters



Two Clusters



Four Clusters

Hierarchical clustering

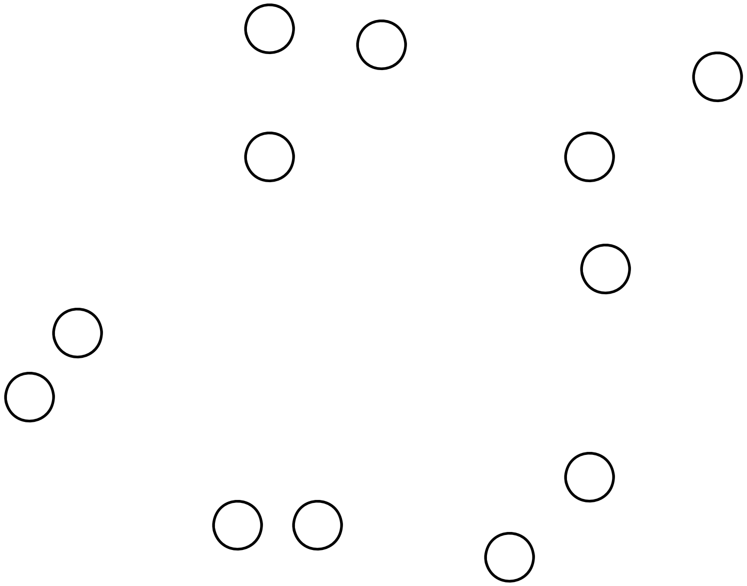
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a proximity 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
- Key operation is the computation of proximities between cluster pairs
  - Different approaches to defining the distance between clusters distinguish the different algorithms

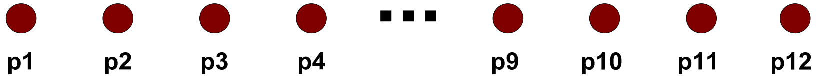
Starting situation

- Start with clusters of individual points and a proximity matrix



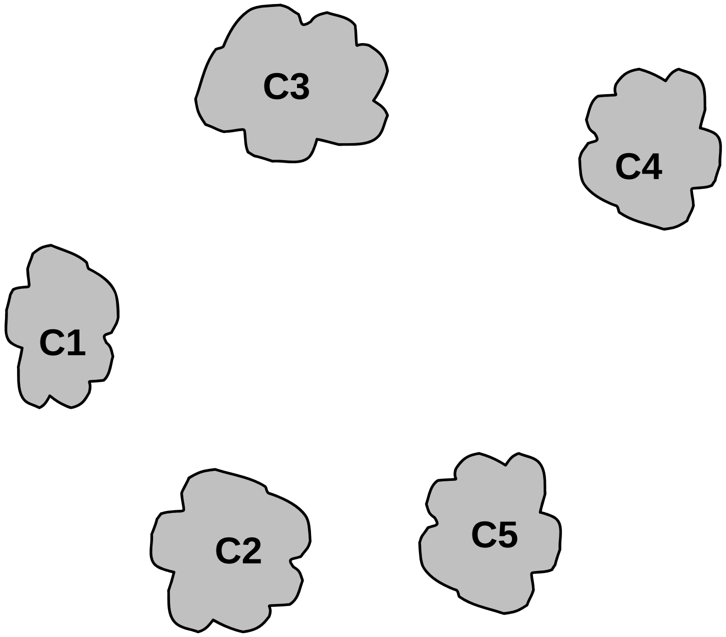
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

proximity matrix



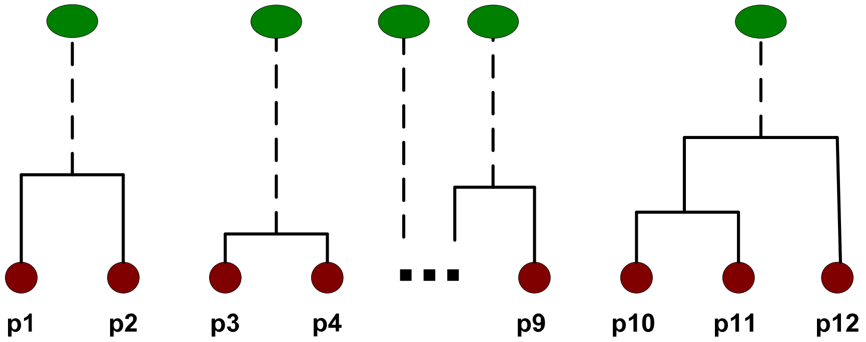
Intermediate situation

- After some merging steps, we have some clusters.



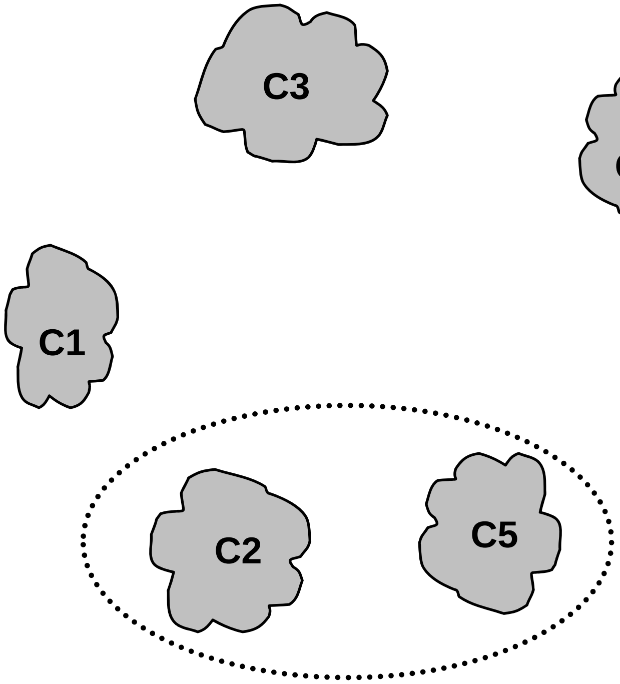
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

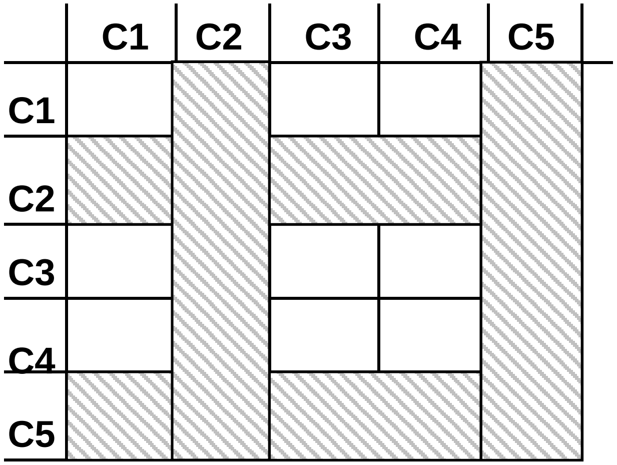
proximity matrix



Intermediate situation

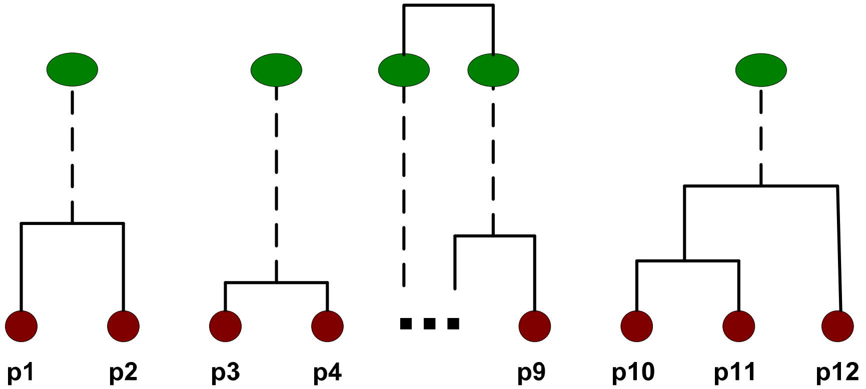
- We decide to merge the two closest clusters (C2 and C5) and update the proximity matrix.





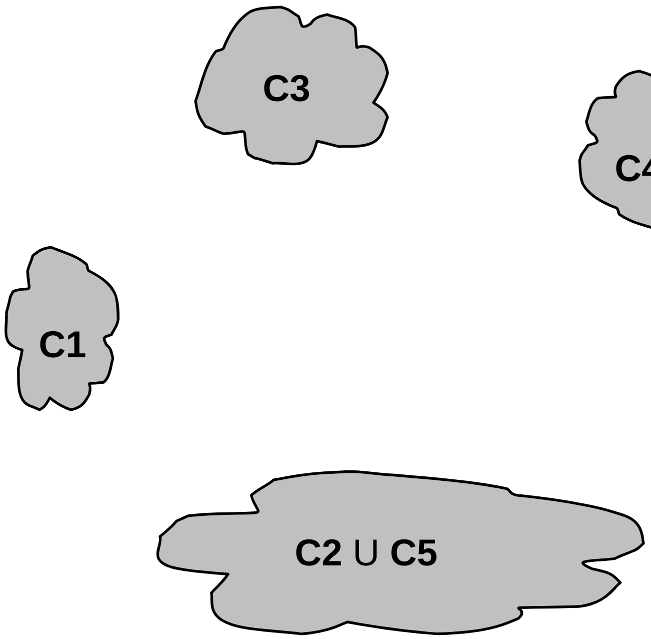
	C1	C2	C3	C4	C5
C1		▨			▨
C2	▨	▨	▨	▨	▨
C3		▨			▨
C4		▨			▨
C5	▨	▨	▨	▨	▨

proximity matrix



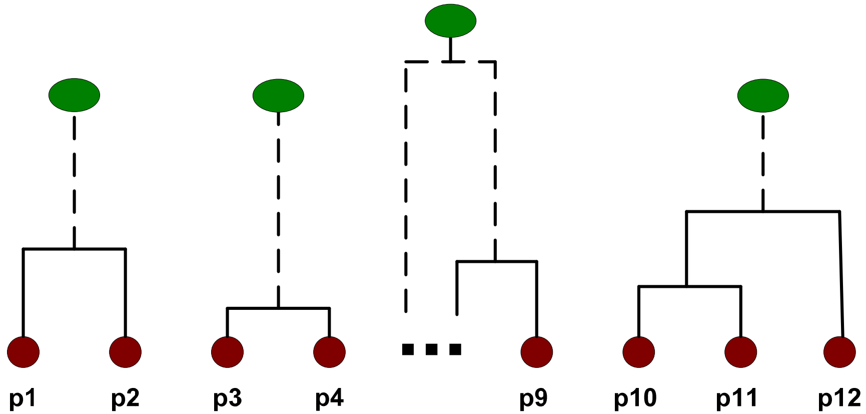
After merging

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

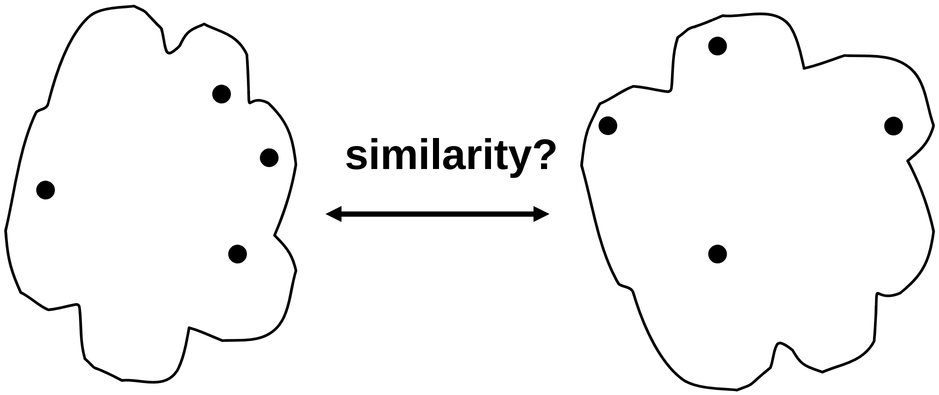


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

proximity matrix



Defining inter-cluster similarity

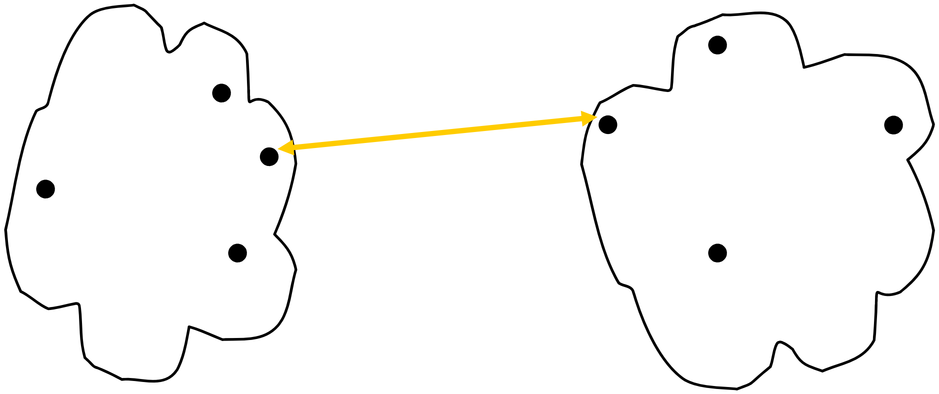


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

proximity matrix

- MIN
- MAX
- Group average
- Distance between centroids
- Other methods driven by an objective function
  - Ward’s method uses squared error

Defining inter-cluster similarity



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

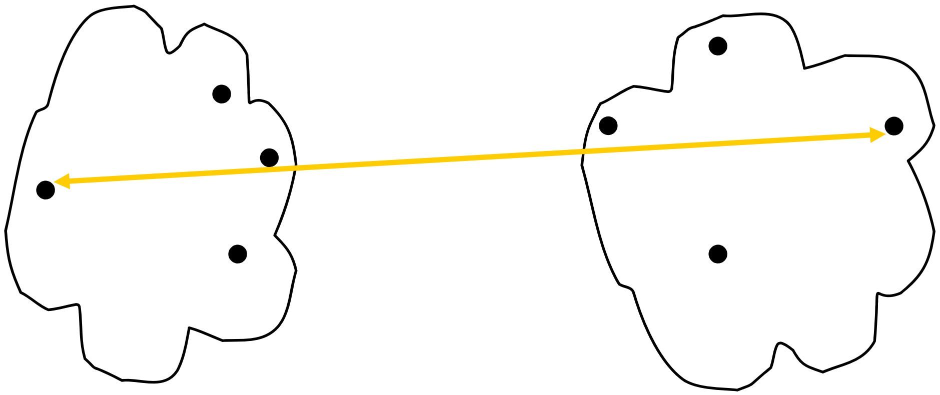
.

.

proximity matrix

- MIN
- MAX
- Group average
- Distance between centroids
- Other methods driven by an objective function
  - Ward’s method uses squared error

Defining inter-cluster similarity

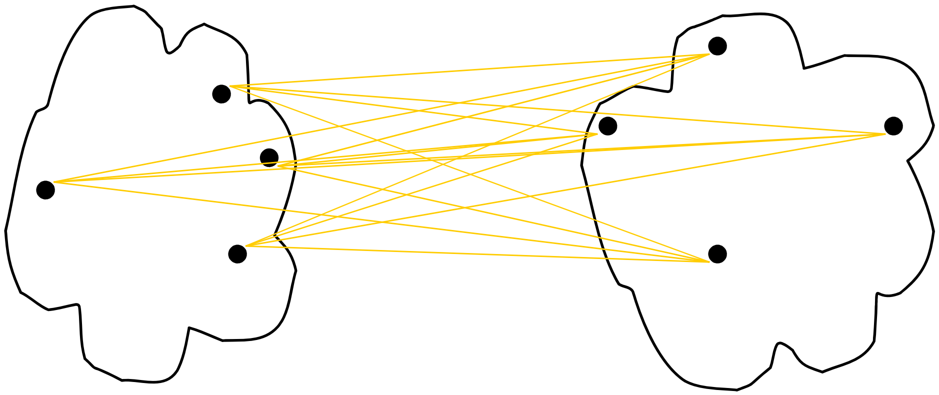


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

proximity matrix

- MIN
- MAX
- Group average
- Distance between centroids
- Other methods driven by an objective function
  - Ward’s method uses squared error

Defining inter-cluster similarity

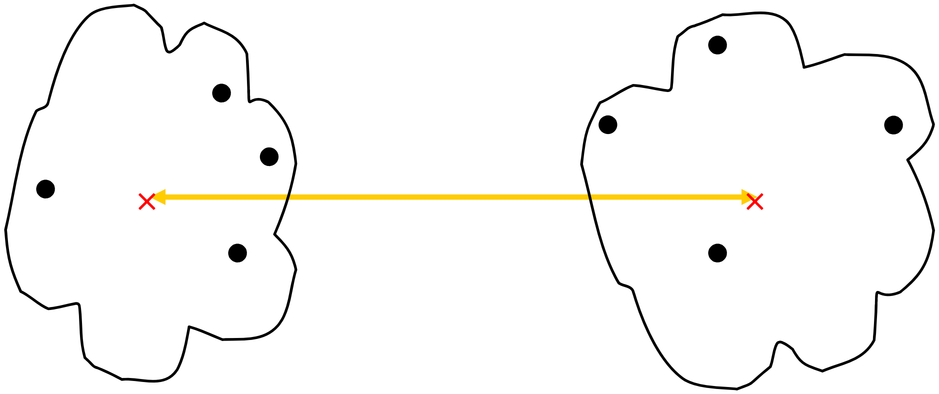


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

proximity matrix

- MIN
- MAX
- Group average
- Distance between centroids
- Other methods driven by an objective function
  - Ward's method uses squared error

Defining inter-cluster similarity



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

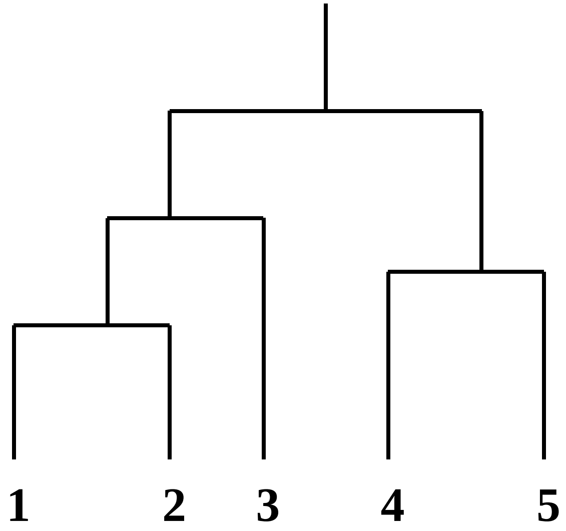
proximity matrix

- MIN
- MAX
- Group average
- Distance between centroids
- Other methods driven by an objective function
  - Ward's method uses squared error

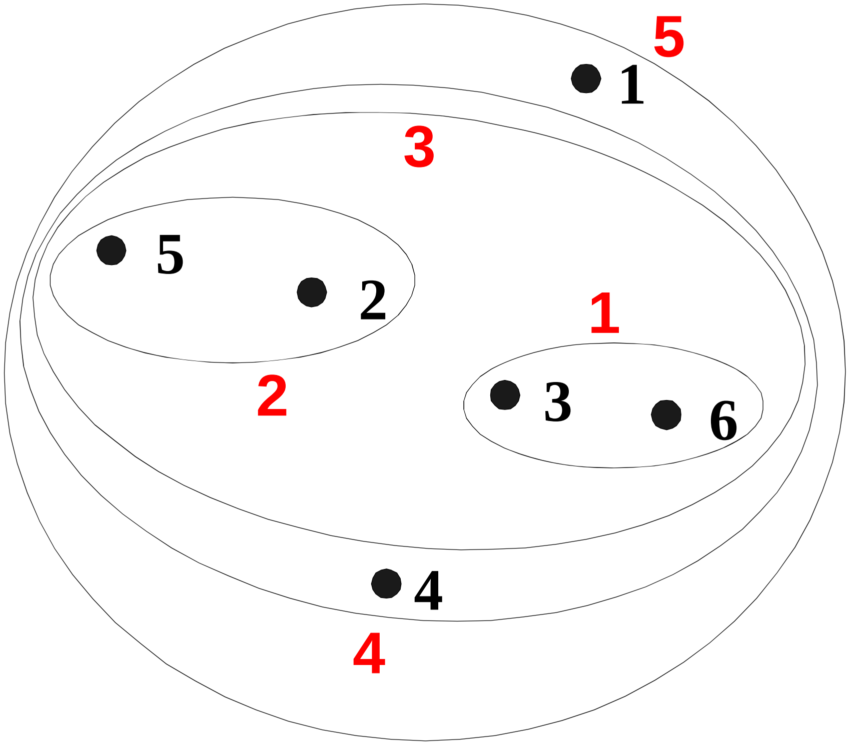
Cluster similarity: MIN or single link

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

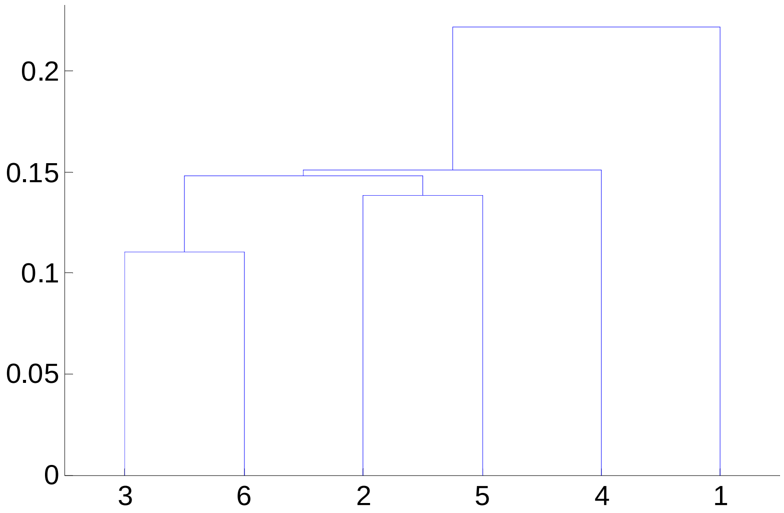
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical clustering: MIN

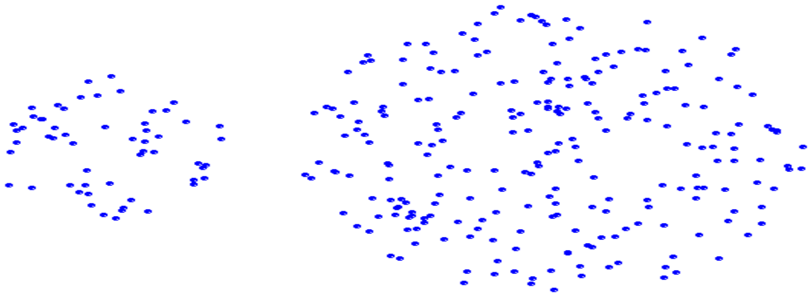


nested clusters

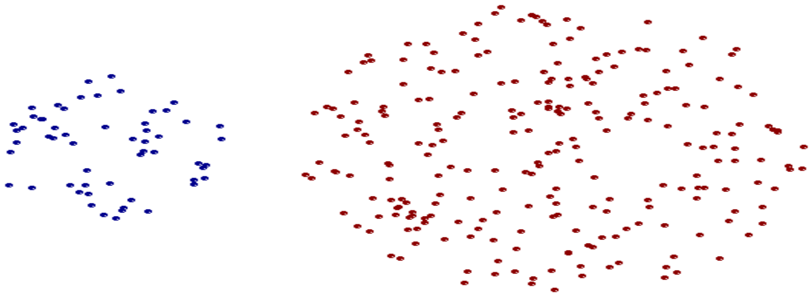


dendrogram

Strength of MIN



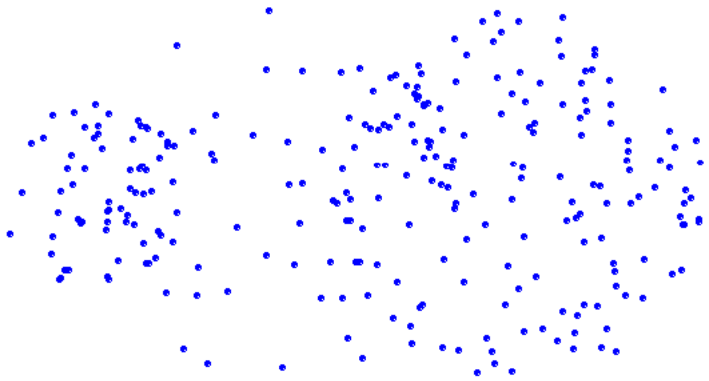
original points



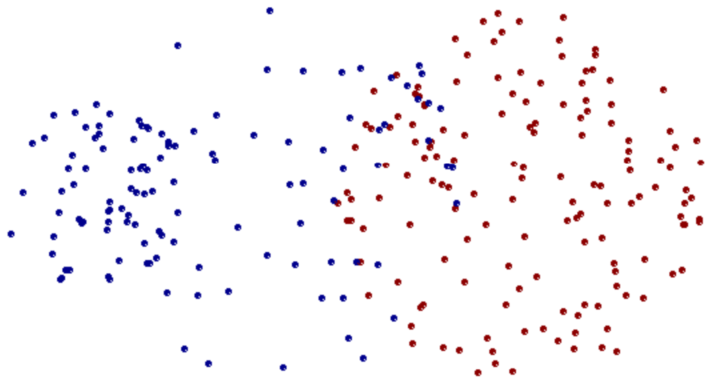
two clusters

- Can handle non-elliptical shapes

Limitations of MIN



original points



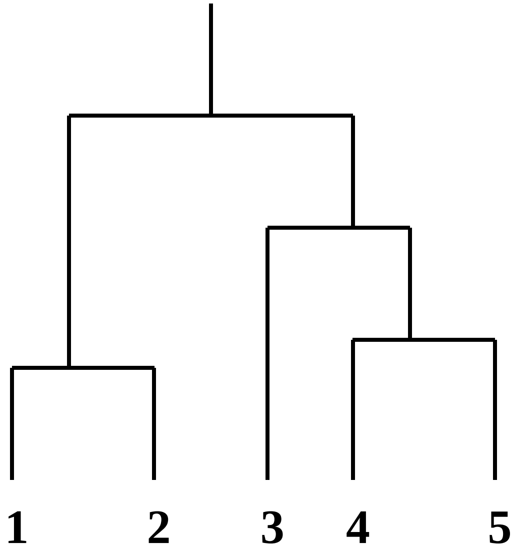
two clusters

- Sensitive to noise and outliers

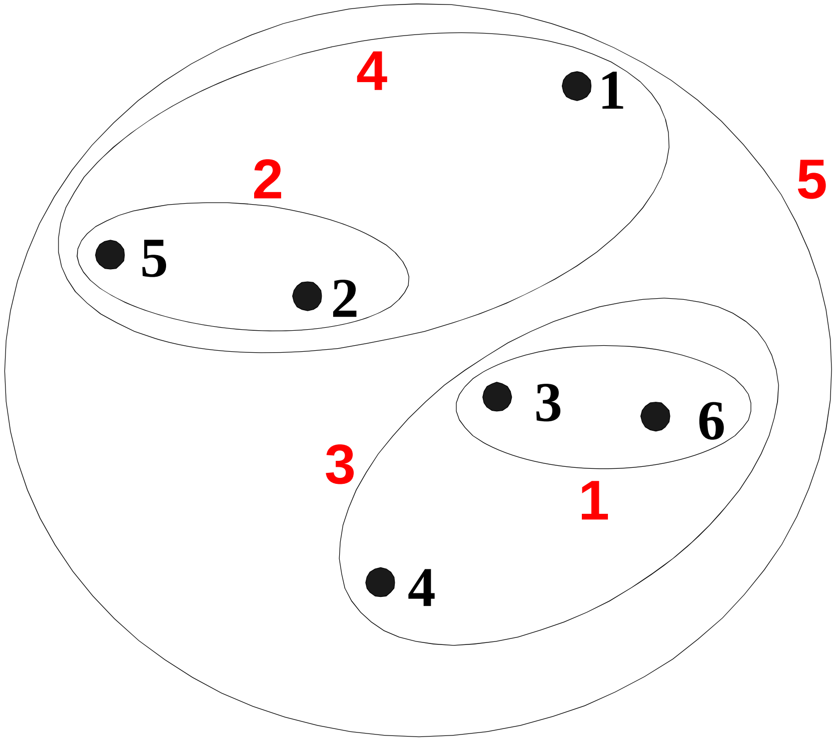
Cluster similarity: MAX or complete link

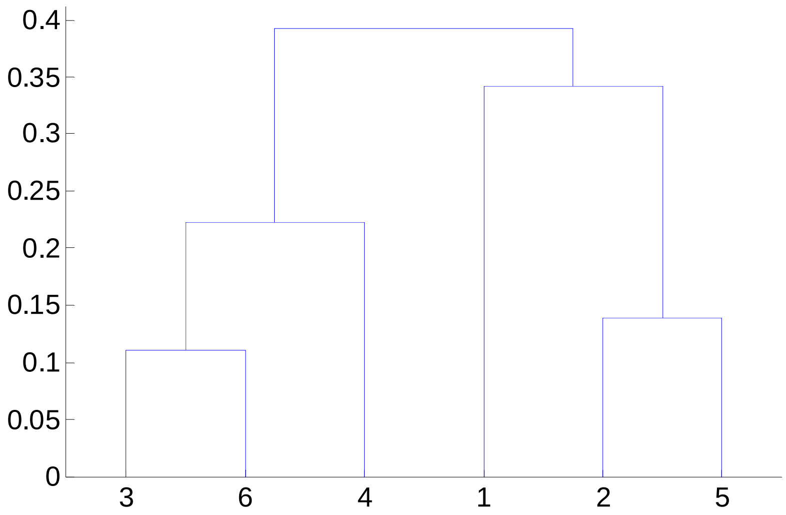
- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

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



Hierarchical clustering: MAX

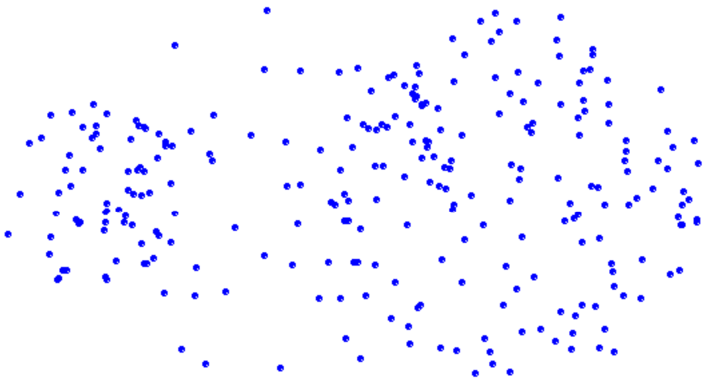




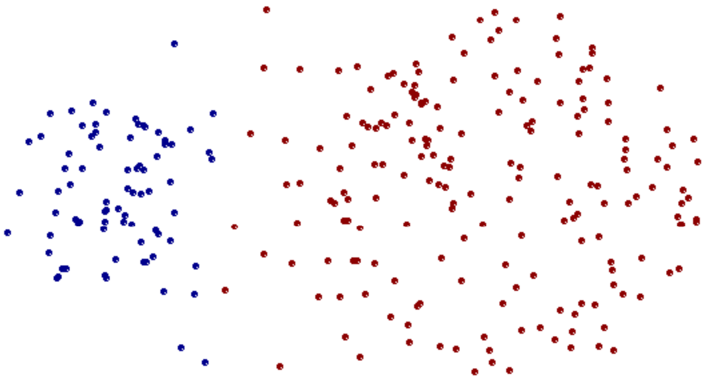
nested clusters

dendrogram

Strength of MAX



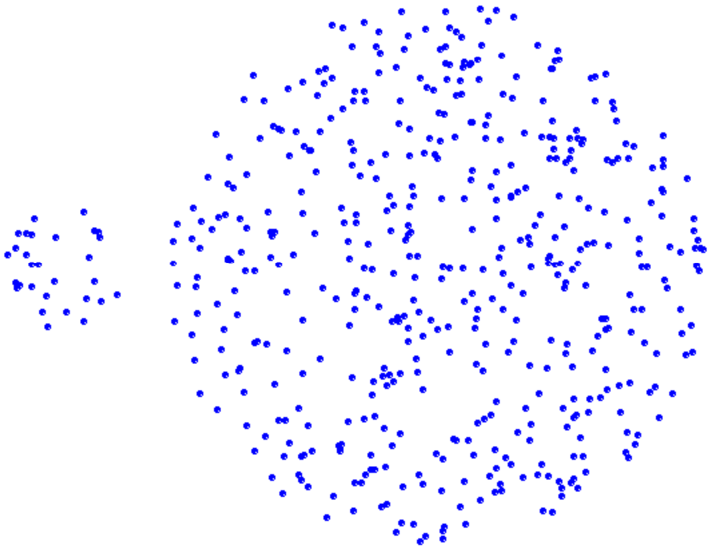
original points



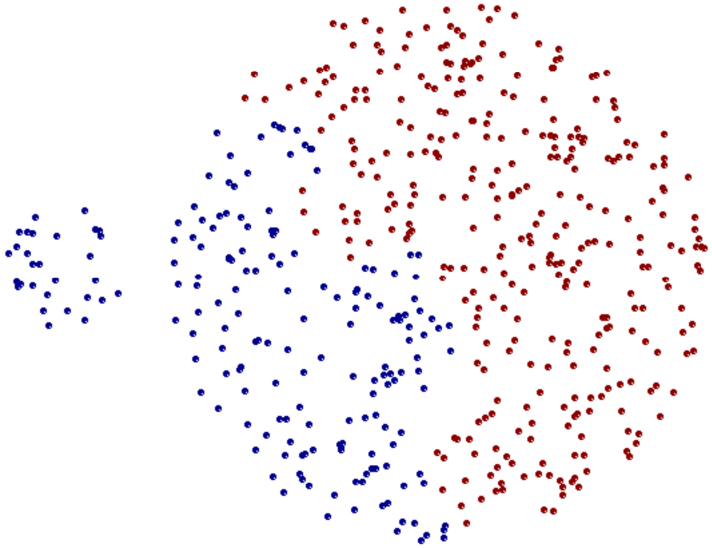
two clusters

- Less susceptible to noise and outliers

Limitations of MAX



original points



two clusters

- Tends to break large clusters
- Biased towards globular clusters

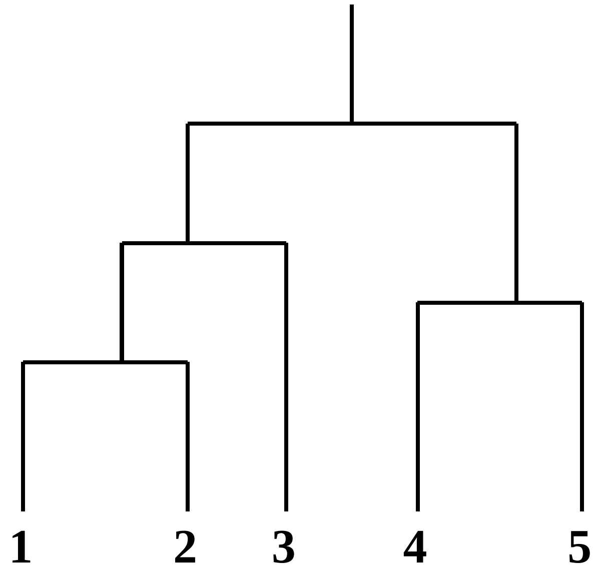
Cluster similarity: group average

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

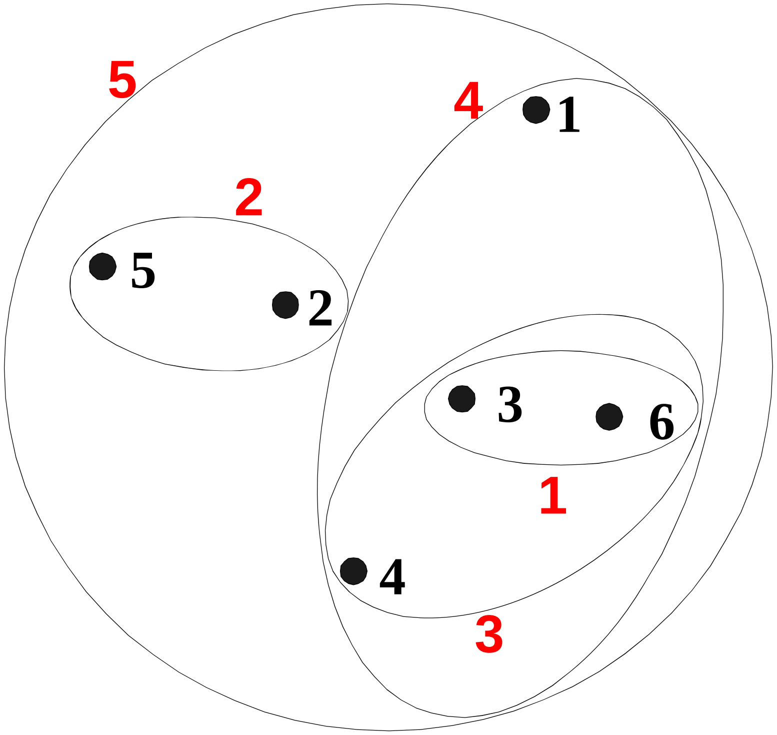
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum\limits_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

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

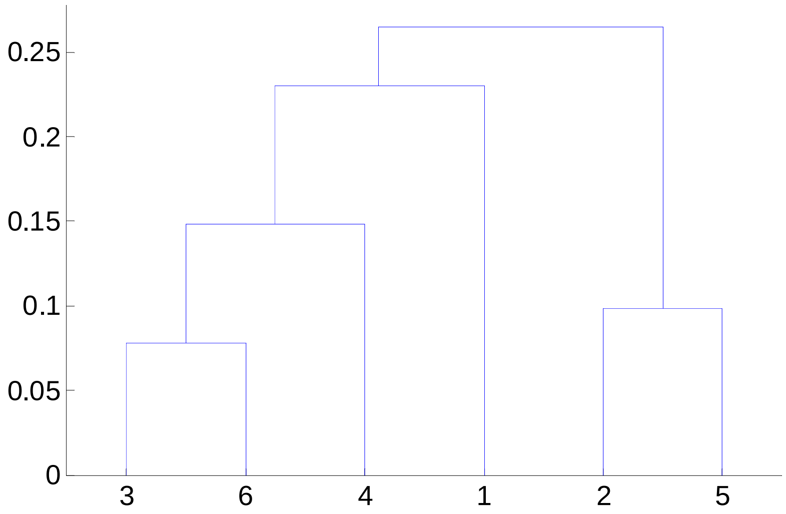
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical clustering: group average



nested clusters



dendrogram

Hierarchical clustering: group average

- Compromise between single and complete link

- Strengths:
  - Less susceptible to noise and outliers

- Limitations:
  - Biased towards globular clusters

Cluster similarity: Ward's method

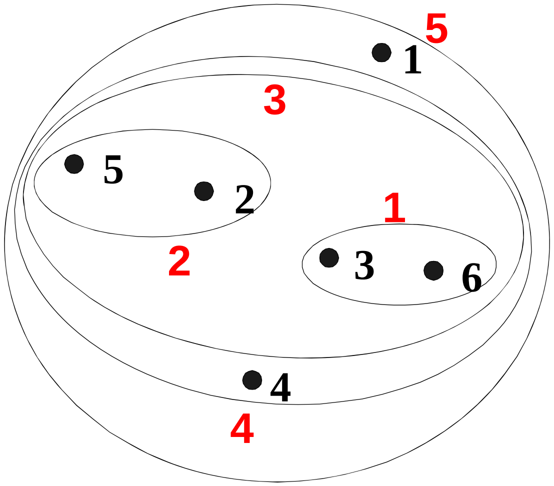
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared

- Less susceptible to noise and outliers

- Biased towards globular clusters

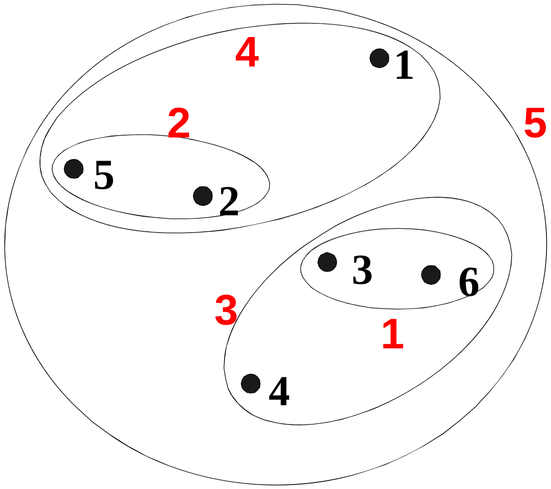
- Hierarchical analogue of k-means
  - Can be used to initialize k-means

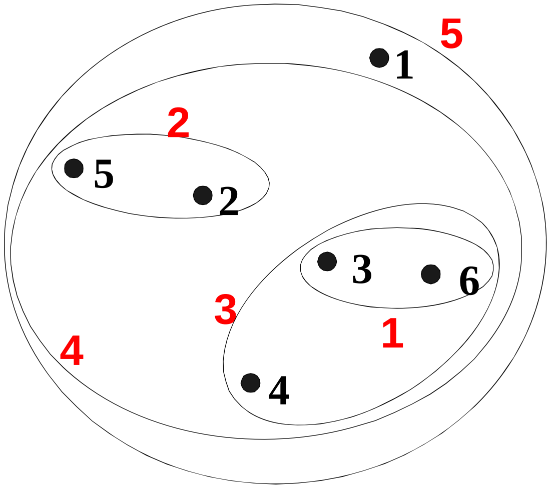
Hierarchical clustering comparison



MIN

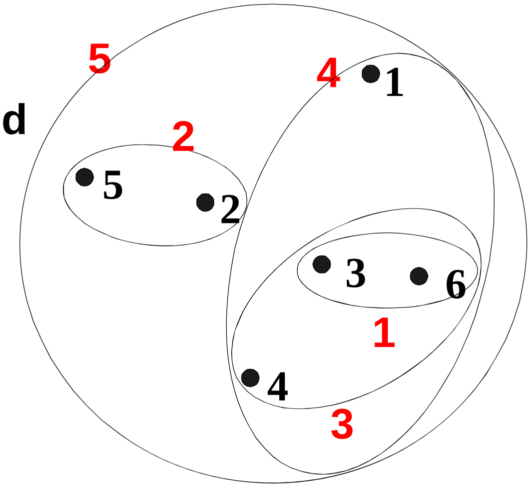
MAX





group average

Ward's method



Hierarchical clustering

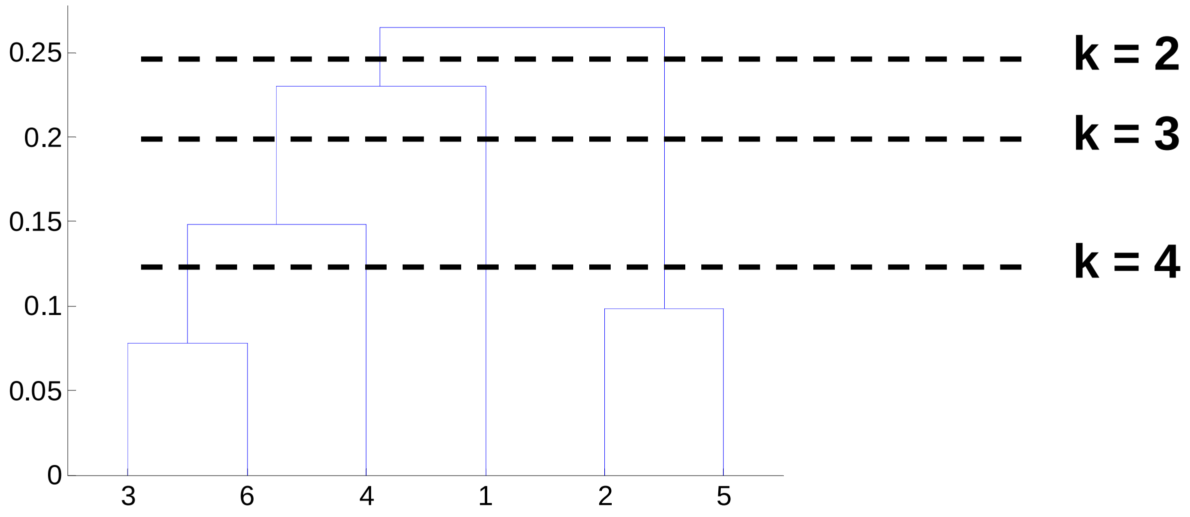
- Time and space complexity
  - n = number of datapoints or objects
  - Space requirement ~ O(n^2) since it uses the proximity matrix.
  - Time complexity ~ O(n^3) many cases.
    - There are n steps and at each step the proximity matrix (size n^2) must be searched and updated.
    - Can be reduced to O(n^2 log(n)) time for some approaches.

Hierarchical clustering

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

From hierarchical to partitional clustering

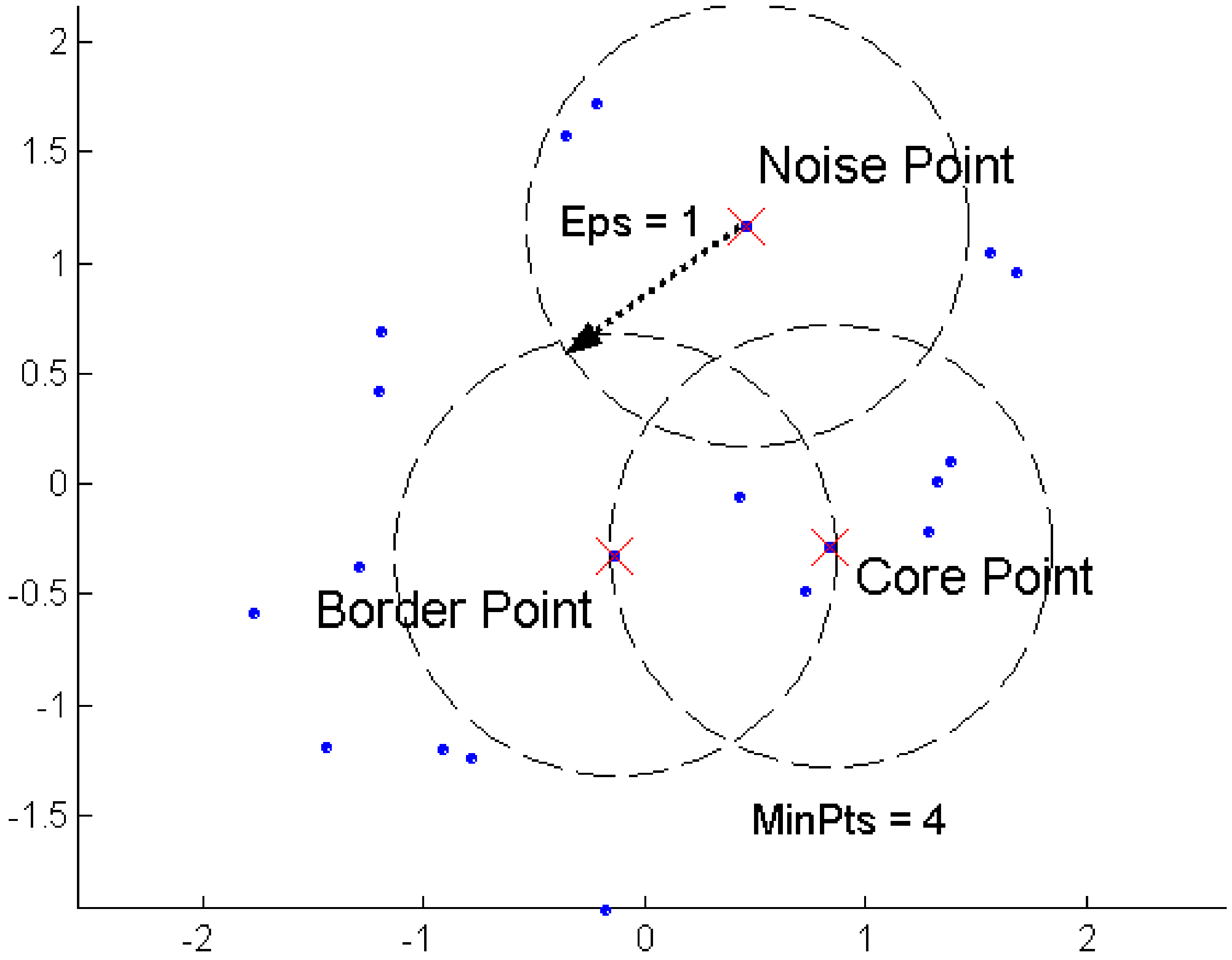
- Cut tree at some height to get desired number of partitions k



DBSCAN

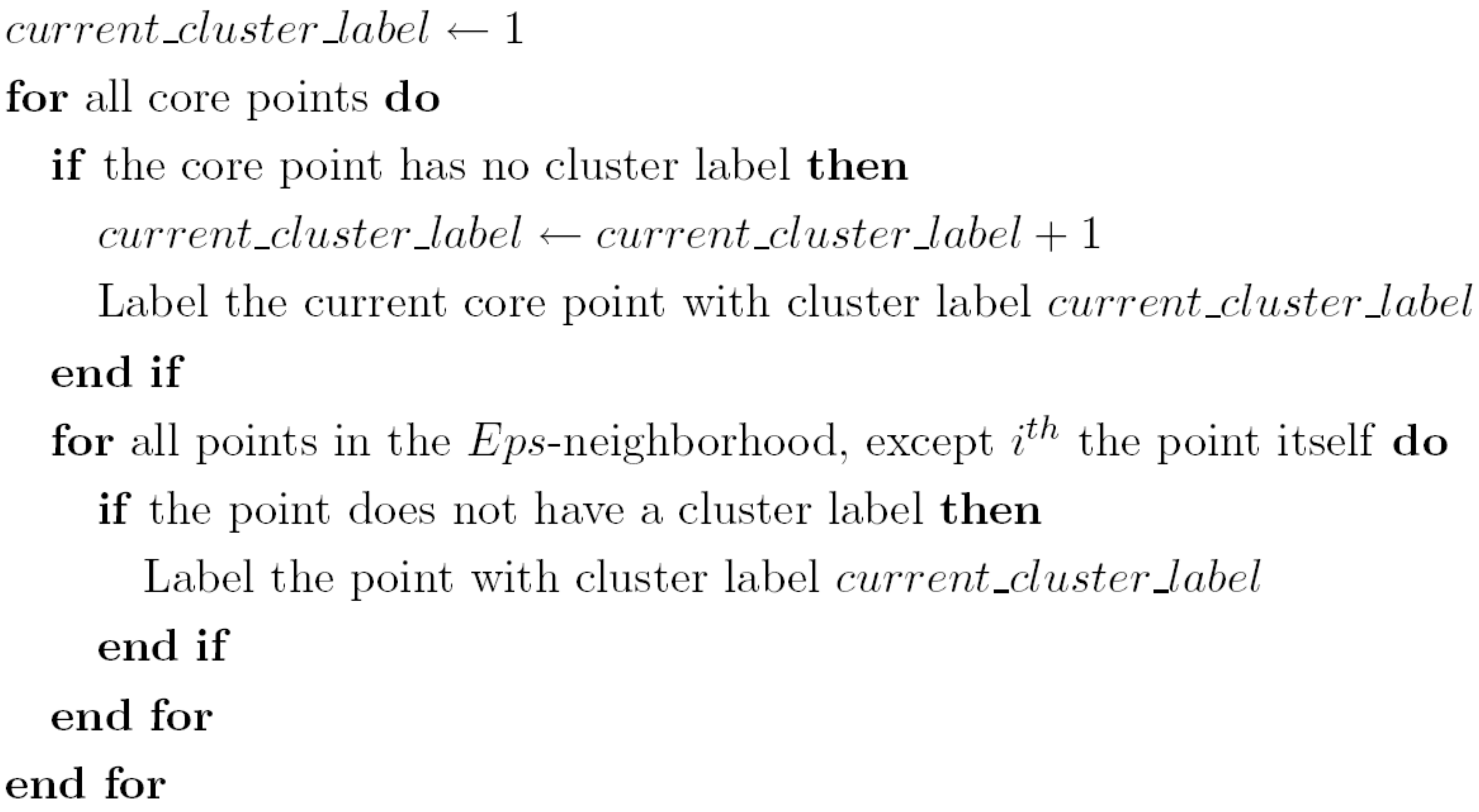
- DBSCAN is a density-based algorithm.
  - Density = number of points within a specified radius (Eps)
  - A point is a core point if it has more than a specified number of points (MinPts) within Eps.
    - These points are in the interior of a cluster.
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.
  - A noise point is any point that is not a core point or a border point.

DBSCAN: core, border, and noise points



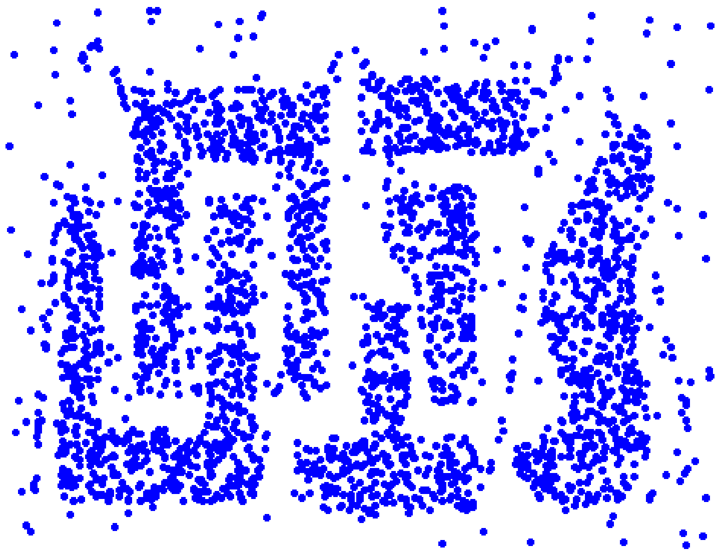
DBSCAN algorithm

- Eliminate noise points
- Perform clustering on the remaining points

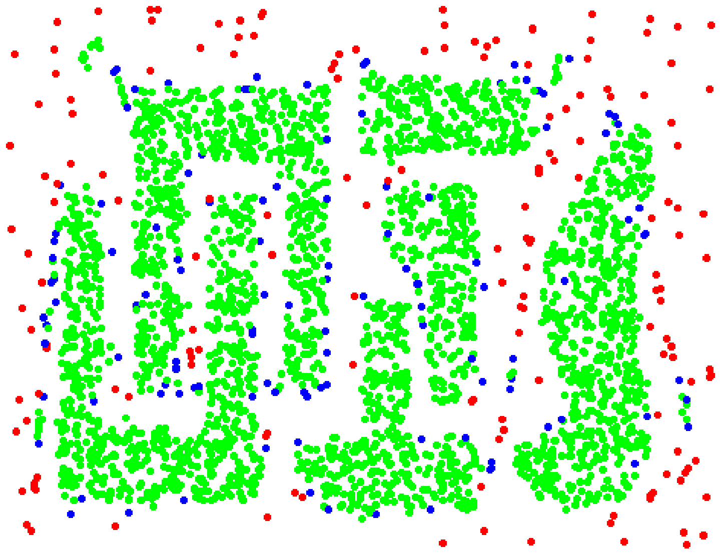


```
$current\_cluster\_label \leftarrow 1$
for all core points do
    if the core point has no cluster label then
        $current\_cluster\_label \leftarrow current\_cluster\_label + 1$
        Label the current core point with cluster label $current\_cluster\_label$
    end if
    for all points in the $Eps$-neighborhood, except $i^{th}$ the point itself do
        if the point does not have a cluster label then
            Label the point with cluster label $current\_cluster\_label$
        end if
    end for
end for
```

DBSCAN: core, border, and noise points



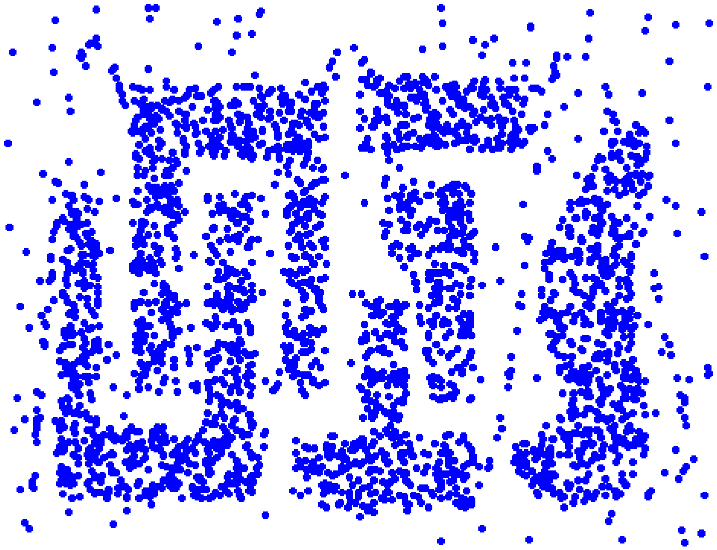
original points



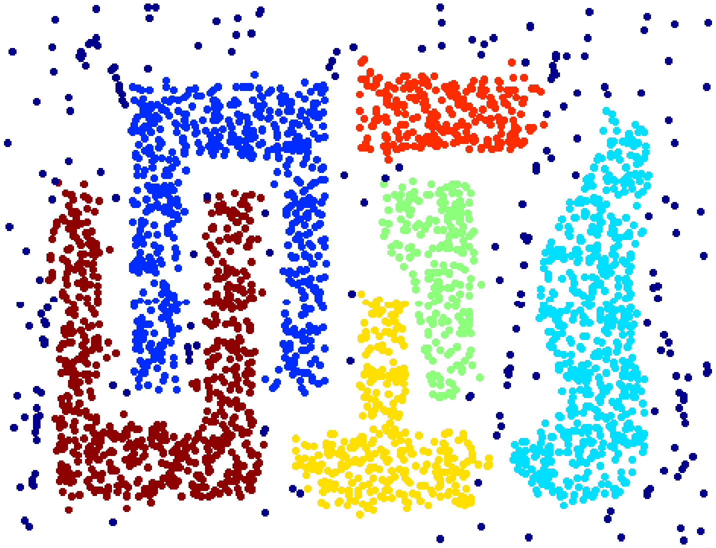
**point types: core,
border and noise**

Eps = 10, MinPts = 4

When DBSCAN works well



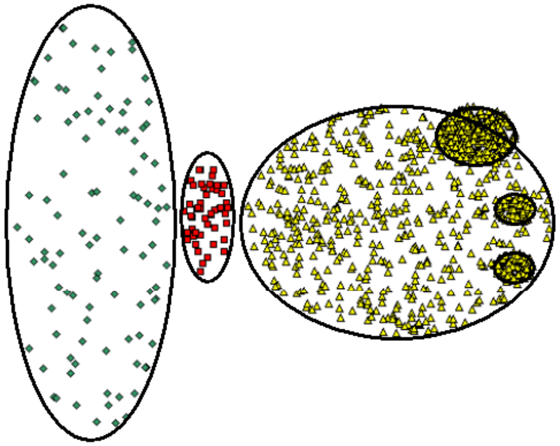
original points



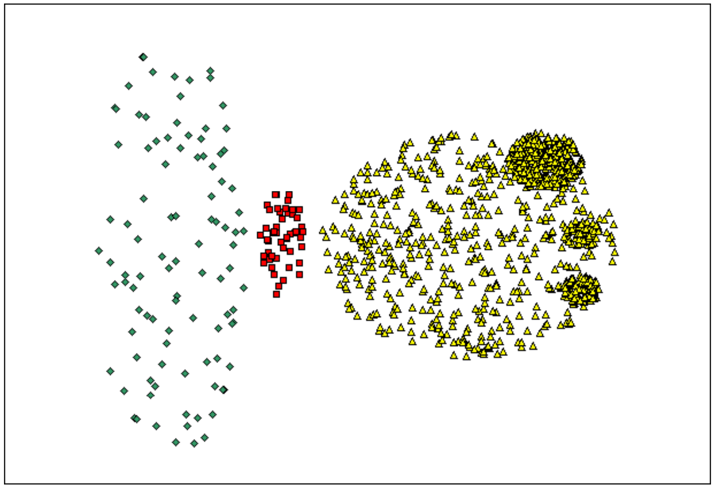
clusters

- resistant to noise
- can handle clusters of different shapes and sizes

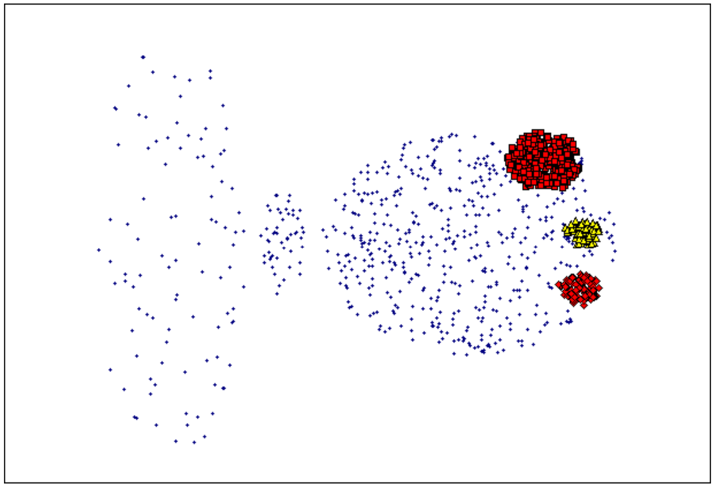
When DBSCAN does NOT work well



original points



(MinPts=4, Eps=9.75).

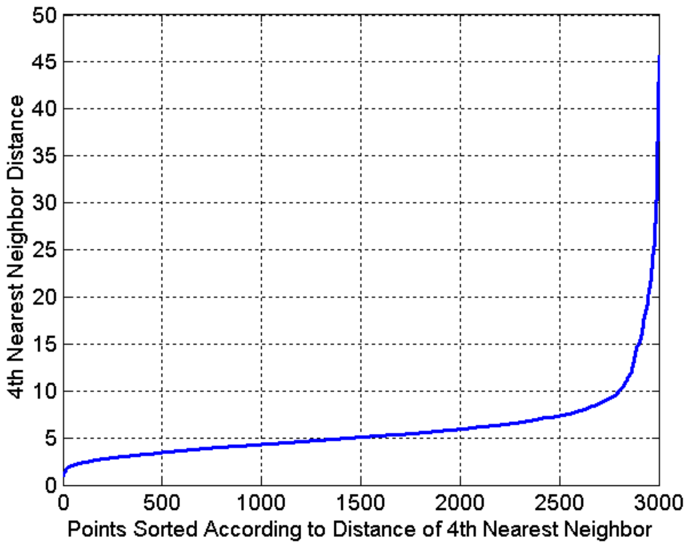


(MinPts=4, Eps=9.92)

- varying densities
- high-dimensional data

DBSCAN: determining EPS and MinPts

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



Cluster validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall, squared error

- For clustering, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- But cluster quality is often in the eye of the beholder!

- It’s still important to try and measure cluster quality
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Different types of cluster validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.

   - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of cluster validity

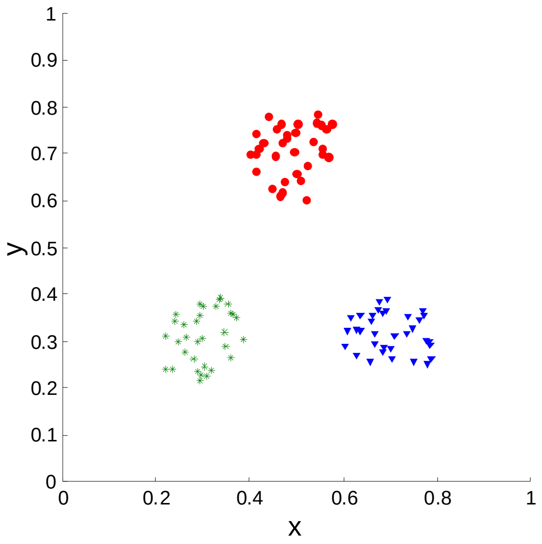
- Numerical measures used to judge various aspects of cluster validity are classified into the following three types:
  - External index: Measures extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal index: Measures the "goodness" of a clustering structure *without* respect to external information.
    - Correlation
    - Visualize similarity matrix
    - Sum of Squared Error (SSE)
  - Relative index: Compares two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy.

Measuring cluster validity via correlation

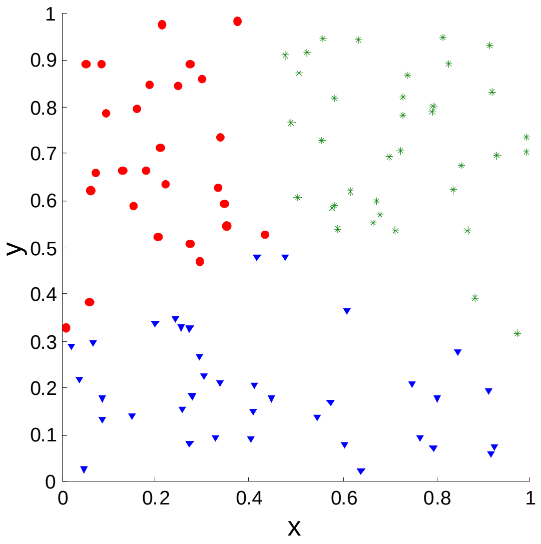
- Two matrices
  - Proximity matrix
  - “Incidence” matrix
    - One row and one column for each data point.
    - An entry is 1 if the associated pair of points belong to same cluster.
    - An entry is 0 if the associated pair of points belongs to different clusters.
- Compute the correlation between 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.

Measuring cluster validity via correlation

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



corr = -0.9235

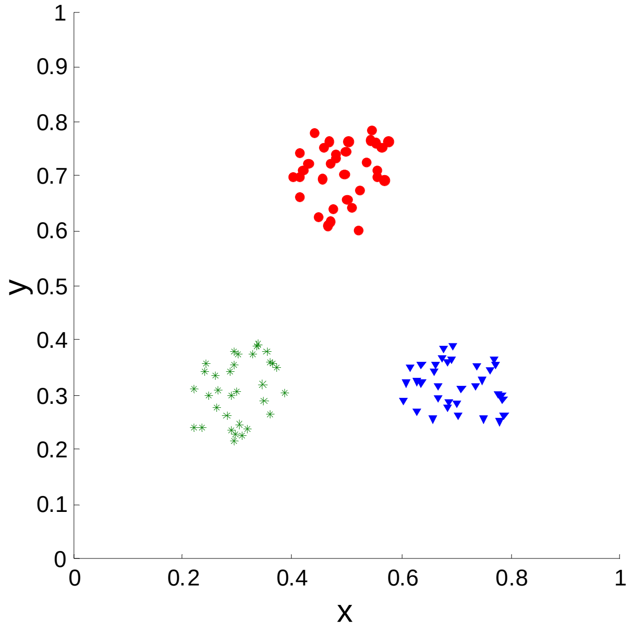


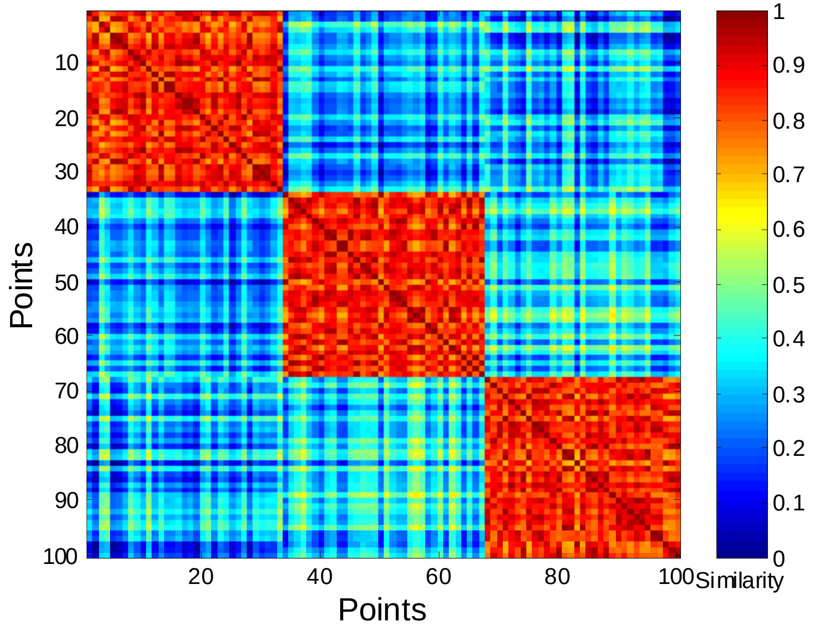
corr = -0.5810

NOTE: correlation will be positive if proximity defined as similarity,
negative if proximity defined as dissimilarity or distance.

Visualizing similarity matrix for cluster validation

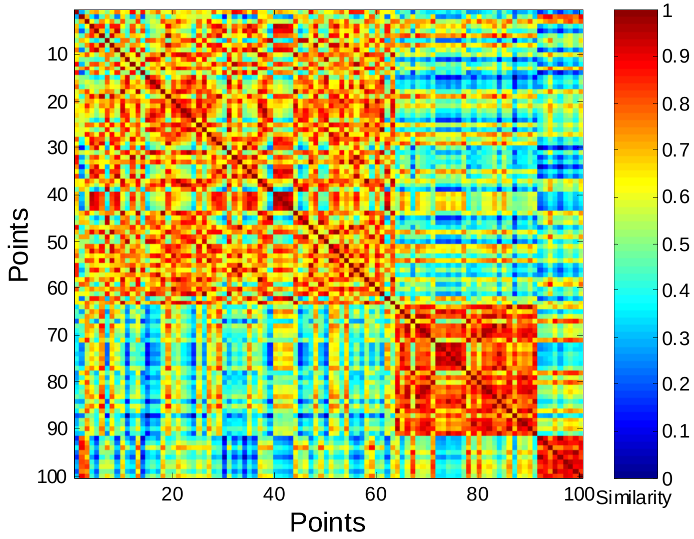
- Order the similarity matrix with respect to cluster indices and inspect visually.

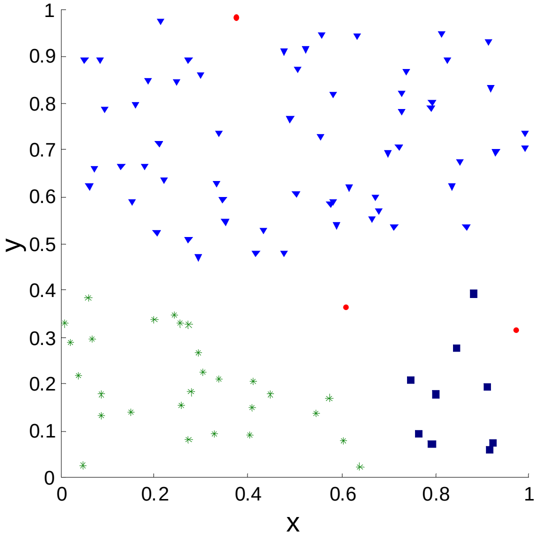




Visualizing similarity matrix for cluster validation

- Clusters in random data are not so crisp

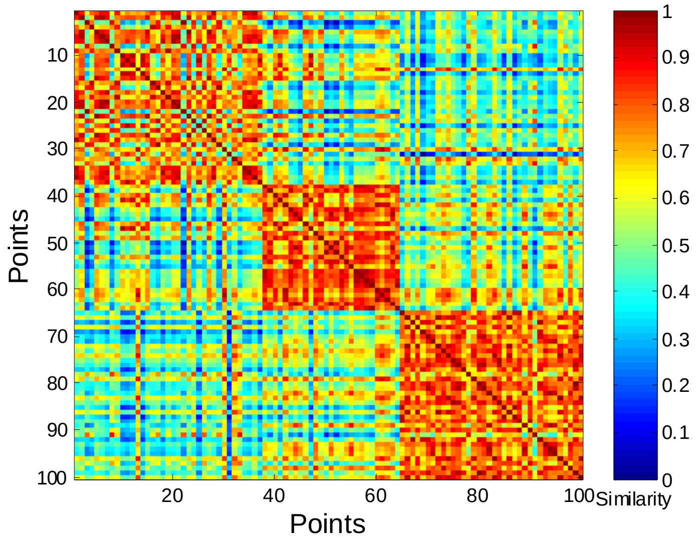


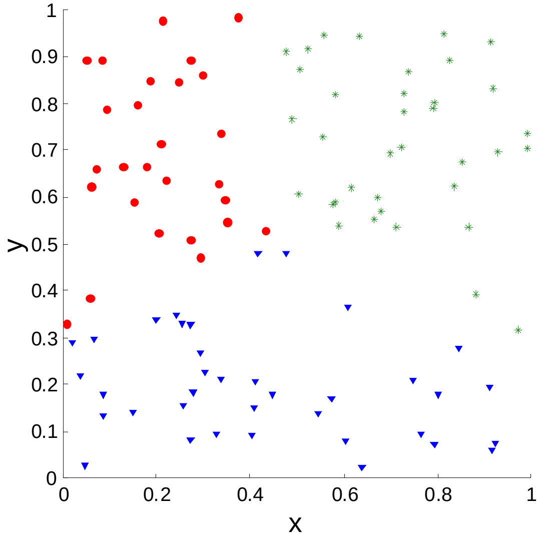


DBSCAN

Visualizing similarity matrix for cluster validation

- Clusters in random data are not so crisp

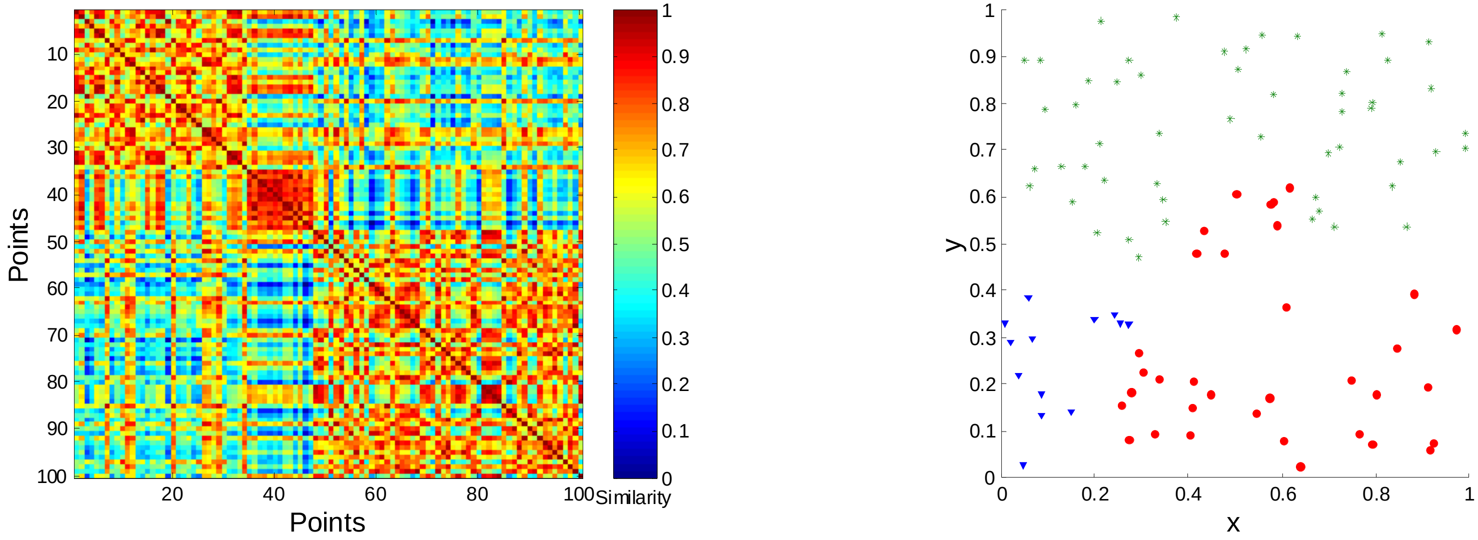




k-means

Visualizing similarity matrix for cluster validation

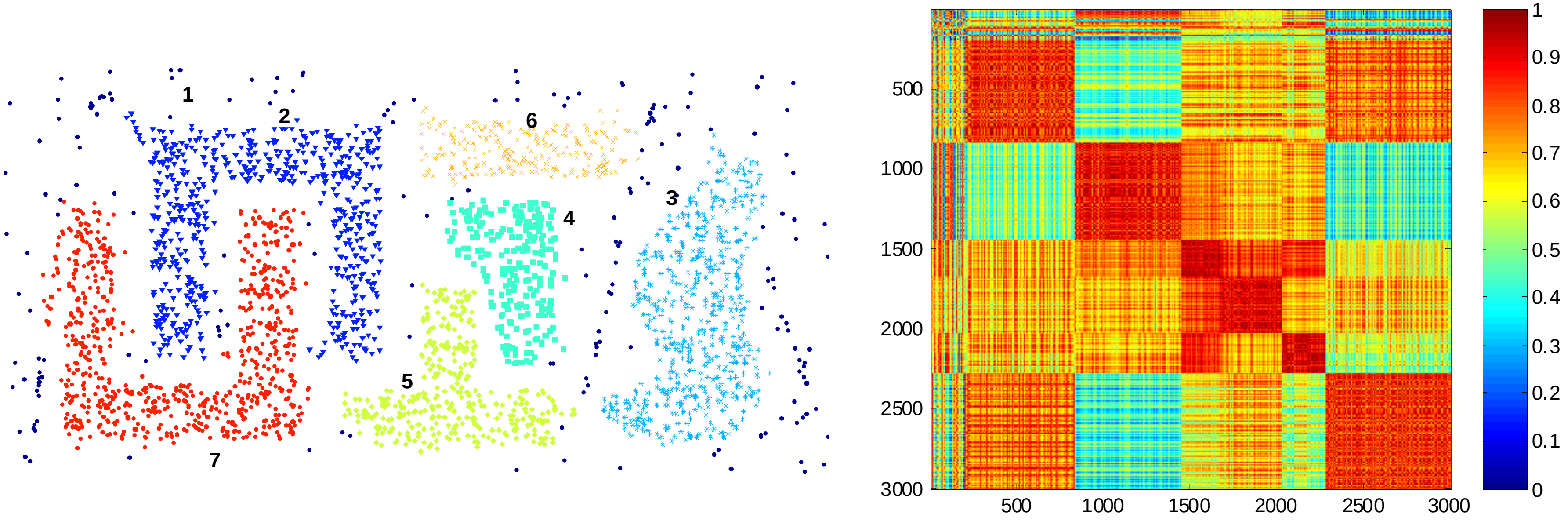
- Clusters in random data are not so crisp



complete link

Visualizing similarity matrix for cluster validation

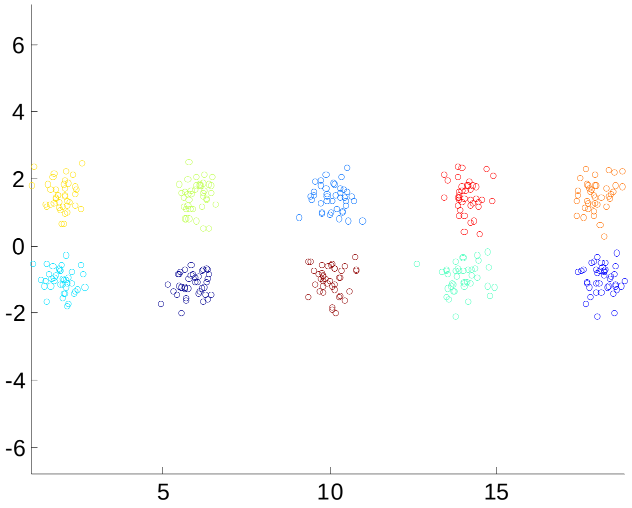
- Not as useful when clusters are non-globular

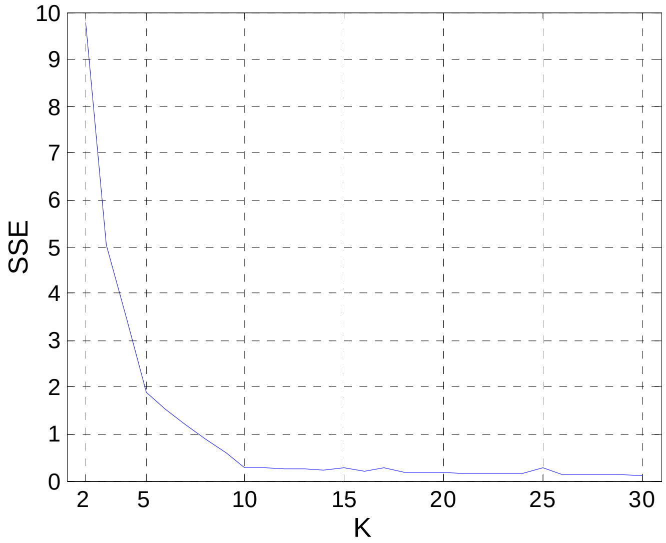


DBSCAN

Internal measures: SSE

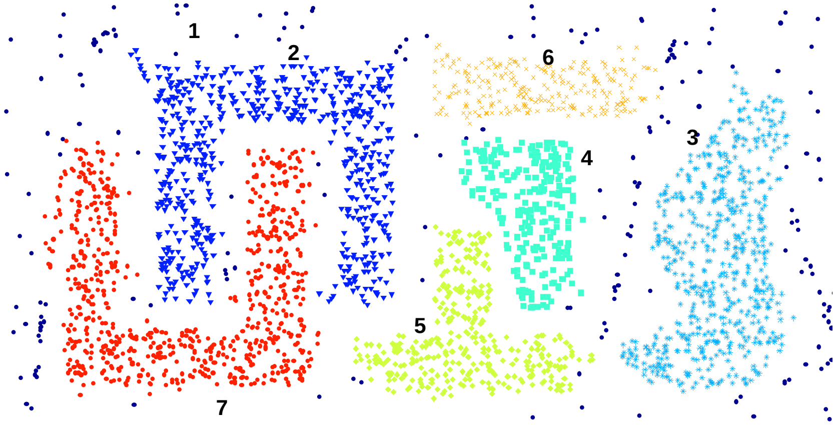
- Clusters in more complicated figures often aren't well separated
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to choose the number of clusters

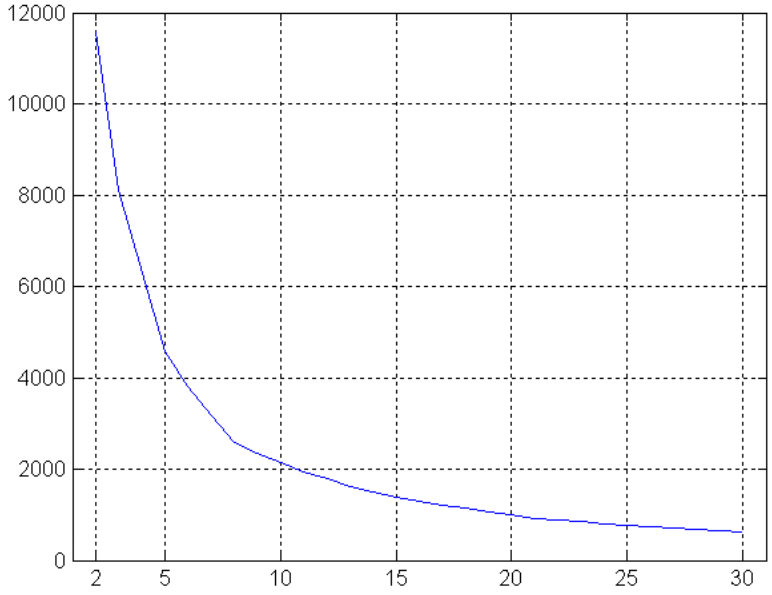




Internal measures: SSE

- SSE curve for a more complicated data set





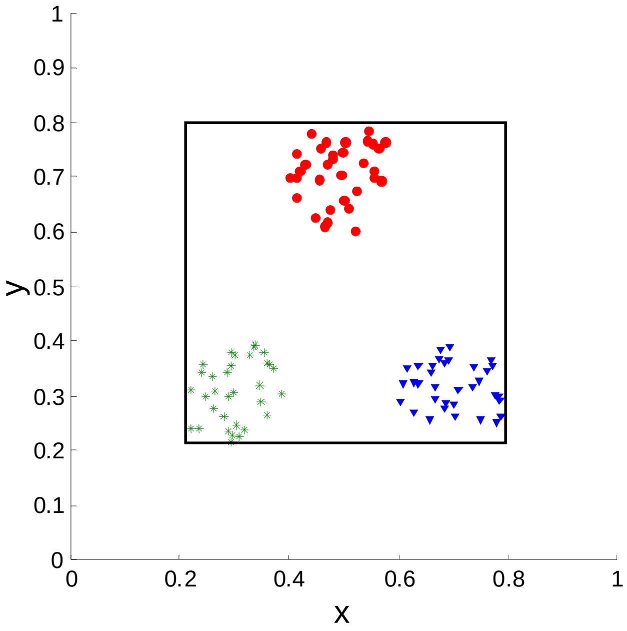
SSE of clusters found using k-means

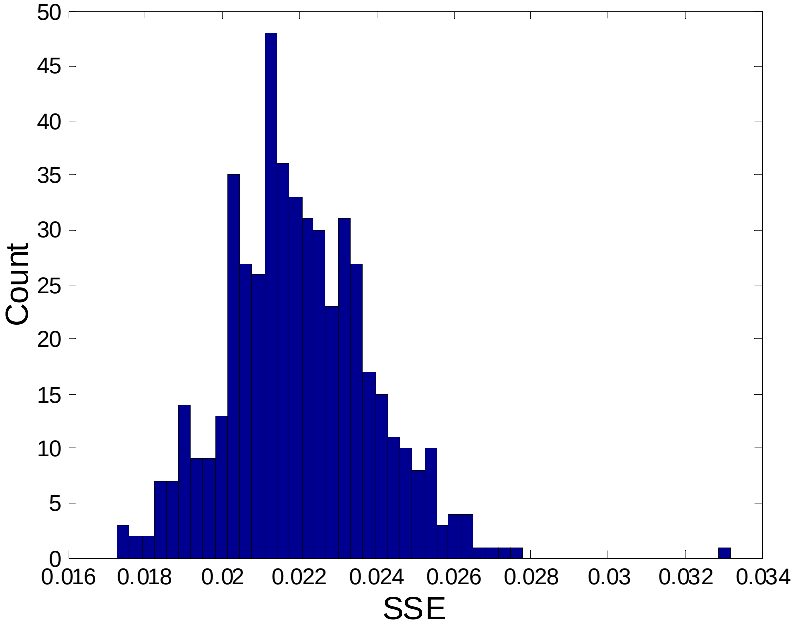
Framework for cluster validity

- Need a framework to interpret any measure.
  - For example, if our measure of evaluation has the value 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
  - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
    - If the value of the index is unlikely, then the cluster results are valid
  - These approaches are more complicated and harder to understand.
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is significant

Statistical framework for SSE

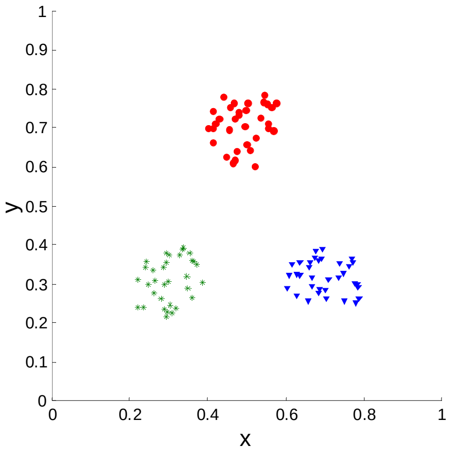
- Example
  - Compare SSE of 0.005 for three true clusters against SSEs for three clusters in random data
  - Histogram shows distributions of SSEs for 500 sets of three clusters in random data points (100 data points randomly placed in range 0.2 - 0.8 for x and y)



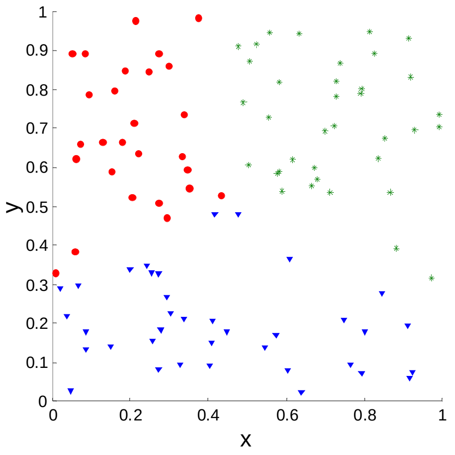


Statistical framework for correlation

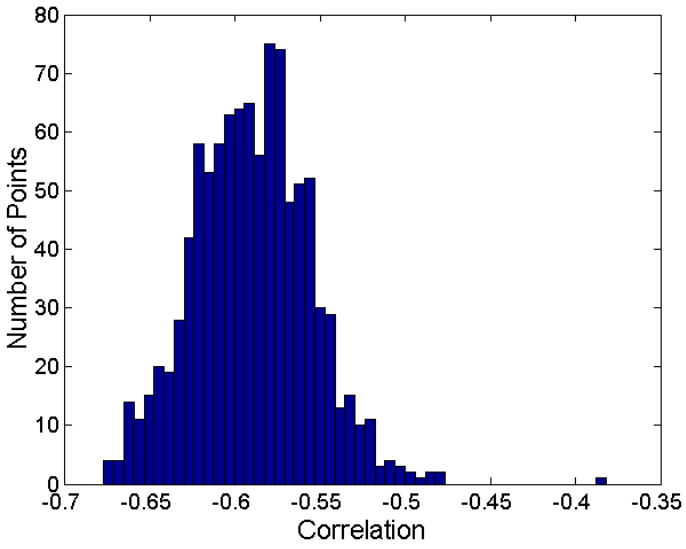
- Correlation of incidence and proximity matrices for the k-means clusterings of the following two data sets.



corr = -0.9235



corr = -0.5810



Final comment on cluster validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

Algorithms for Clustering Data, Jain and Dubes

MATLAB interlude

matlab_demo_11.m