Artificial Intelligence for Medicine II

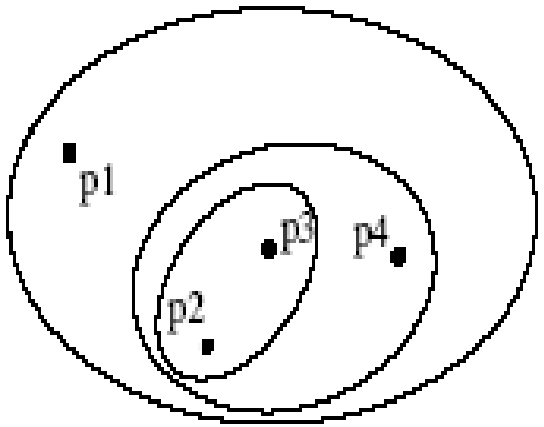
Spring 2025

Lecture 91: Unsupervised Learning Other Clustering Methods

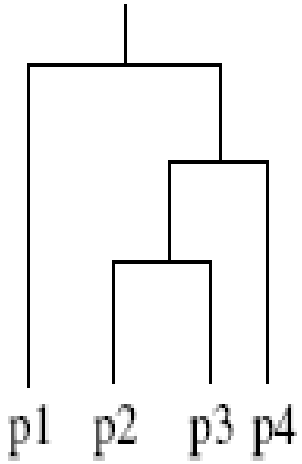
(Many slides adapted from Bing Liu, Han, Kamber & Pei; Tan, Steinbach, Kumar and the web)

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree.
- Can be visualized as a dendrogram
  - Tree like diagram
  - Records the sequences of merges or splits



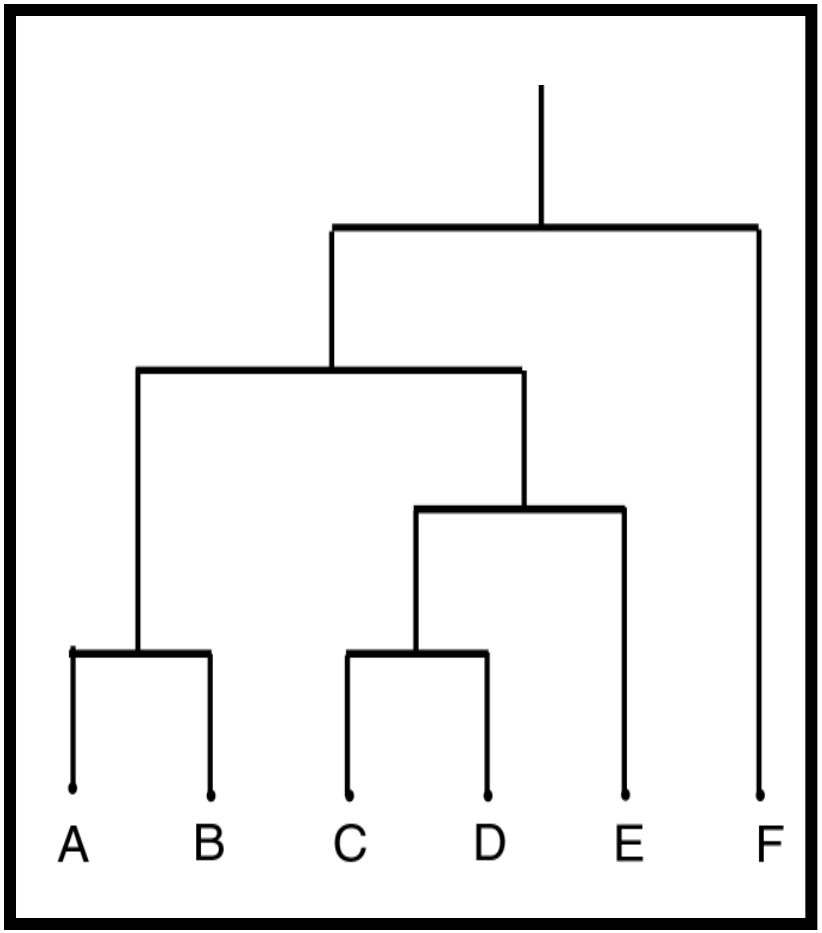
Traditional Hierarchical Clustering



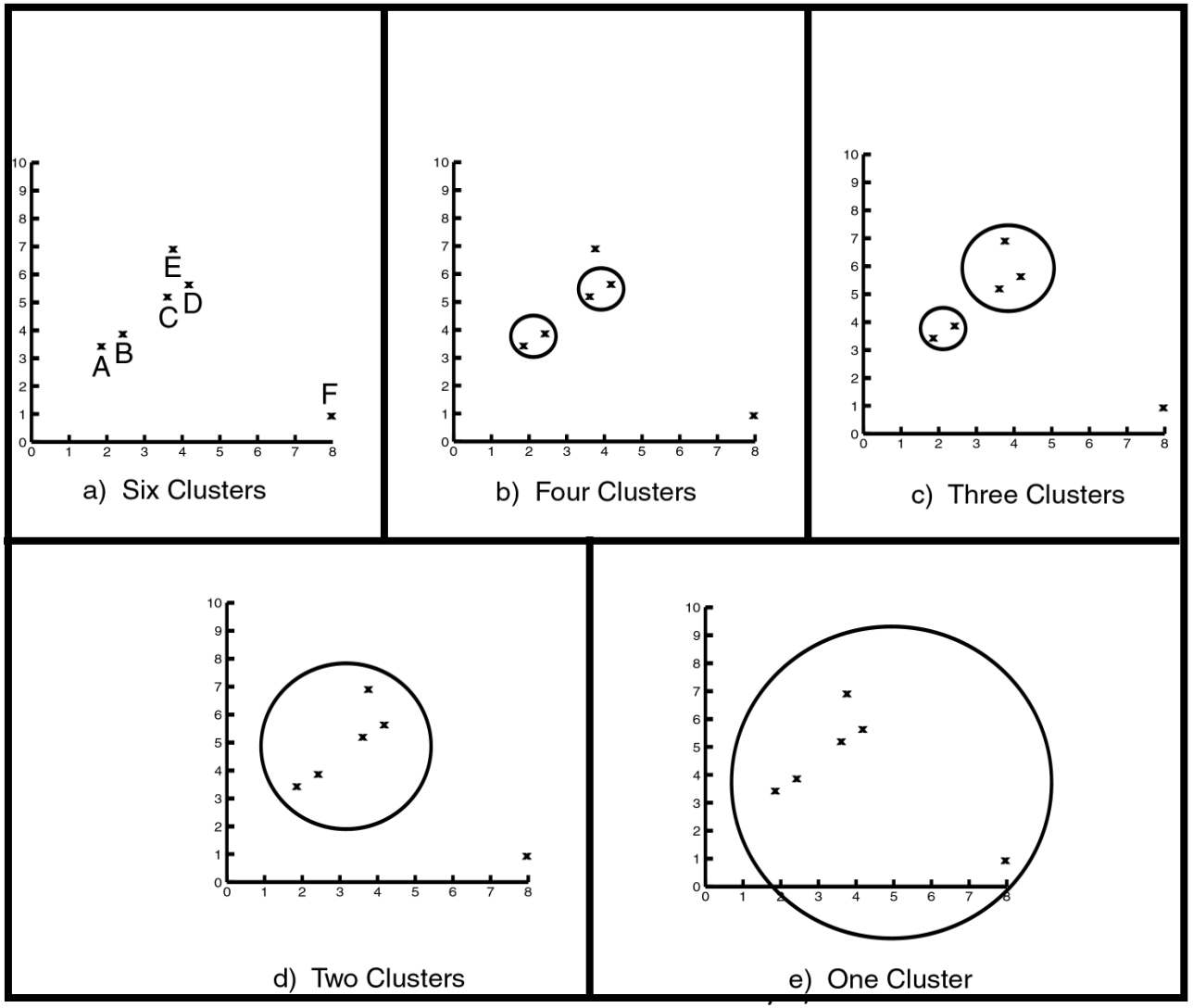
Traditional Dendrogram

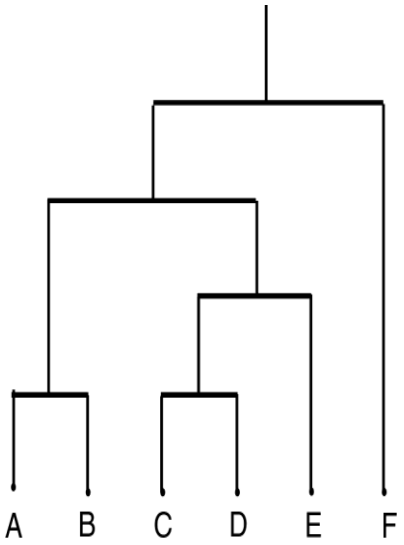
Dendrograms

- ***Dendrogram:*** a tree data structure which illustrates hierarchical clustering techniques.
- Each level shows clusters for that level.
  - Leaf – individual clusters
  - Root – one cluster
- A cluster at level i is the union of its children clusters at level i+1.
- each level is typically associated with a distance threshold: sub-clusters of the clusters at that level were combined because they had a distance between them of less than the distance threshold.

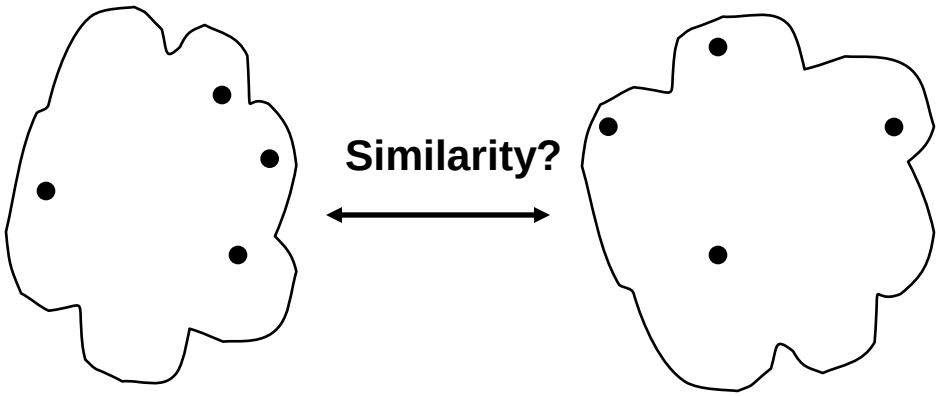


Dendrograms





How to Define Inter-Cluster Distance



- MIN
- MAX
- Distance Between Centroids
- Other methods

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

.

.

Proximity Matrix

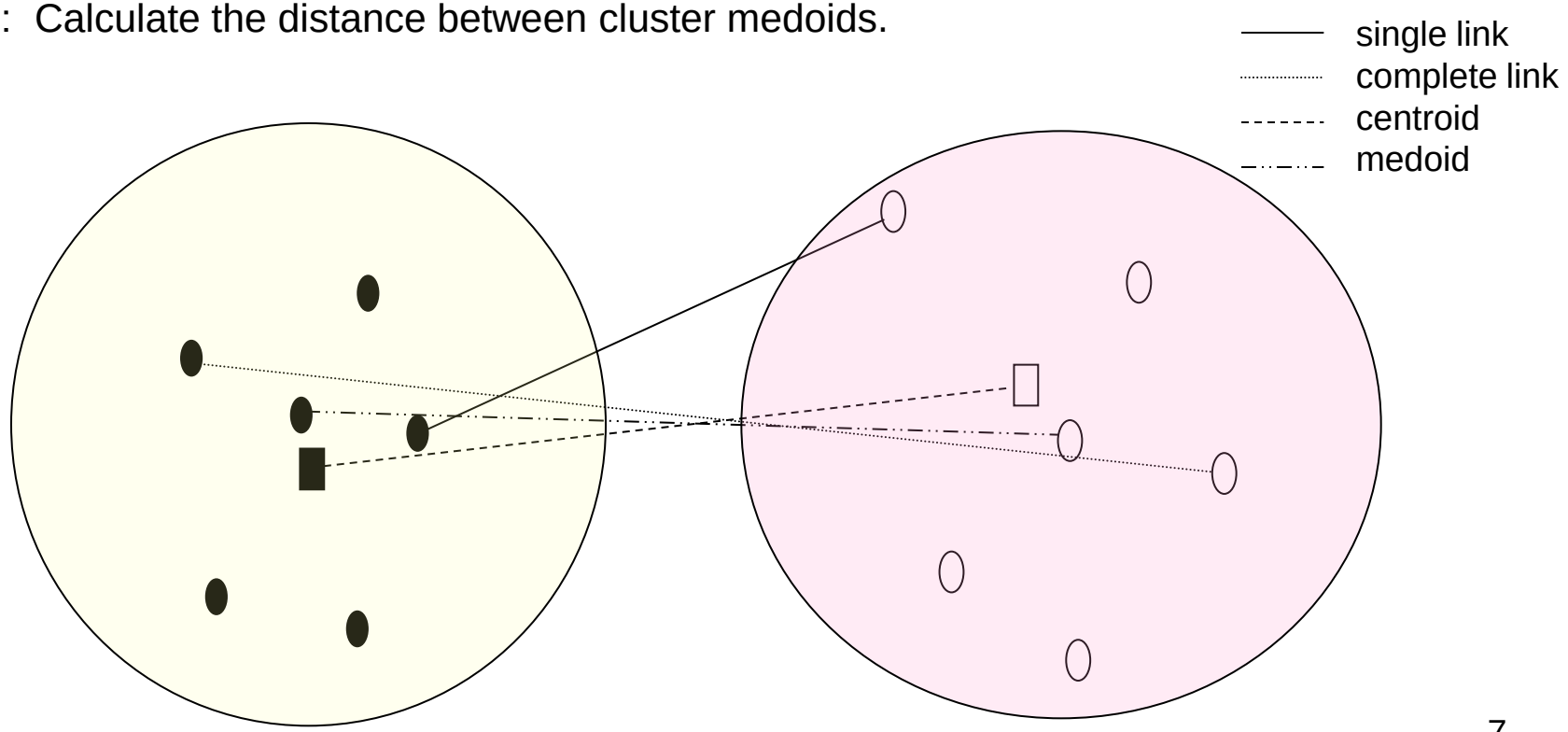
Inter-cluster distance (similarity) metrics

- Many hierarchical clustering algorithms require that the distance (or inter-cluster similarity) between clusters to be determined.
- Let K_i, K_j be two clusters, $i \in K_i$ and $j \in K_{j,,}$, there is a variety of distance metrics to calculate the distance between clusters

  - **Single link**: Smallest distance between two points, one in K_i and the other in K_j : $dist(K_i,K_j) = min(dist(i, j))$.
  - **Complete link**: Largest distance between two points, one in K_i and the other in K_j : $dist(K_i,K_j) = max(dist(i, j))$.
  - **Average link**: Average distance between two points, one in K_i and the other in K_j : $dist(K_i,K_j) = avg(dist(i, j))$.
  - **Centroid**: Distance between the centroid: $dist(K_i,K_j) = dist(C_{Ki},C_{Kj})$.
  - **Medoid**: Distance between the medoids: $dist(K_i,K_j) = dist(M_{Ki},M_{Kj})$.

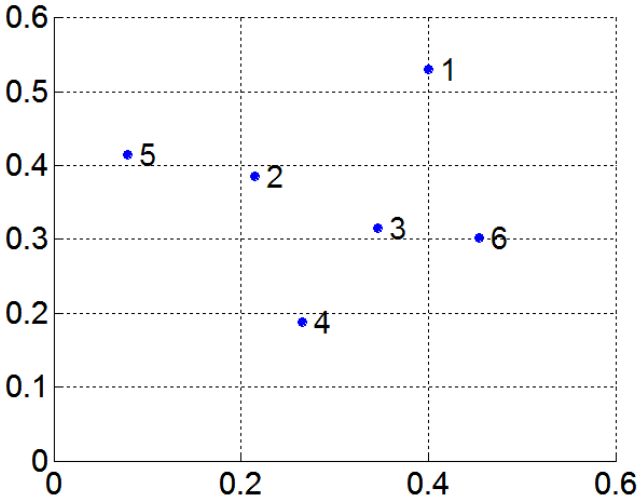
Inter-cluster distance metrics (cont.)

•**Single link**: Calculate the smallest distance between an element in one cluster and an element in the other cluster.
•**Complete link (Farthest neighbor)**: Calculate the largest distance between an element in one cluster and an element in the other cluster.
•**Average link**: Calculate the average distance between each element in one cluster and all elements in the other cluster.
•**Centroid**: Calculate the distance between cluster centroids.
•**Medoid**: Calculate the distance between cluster medoids.



MIN or Single Link

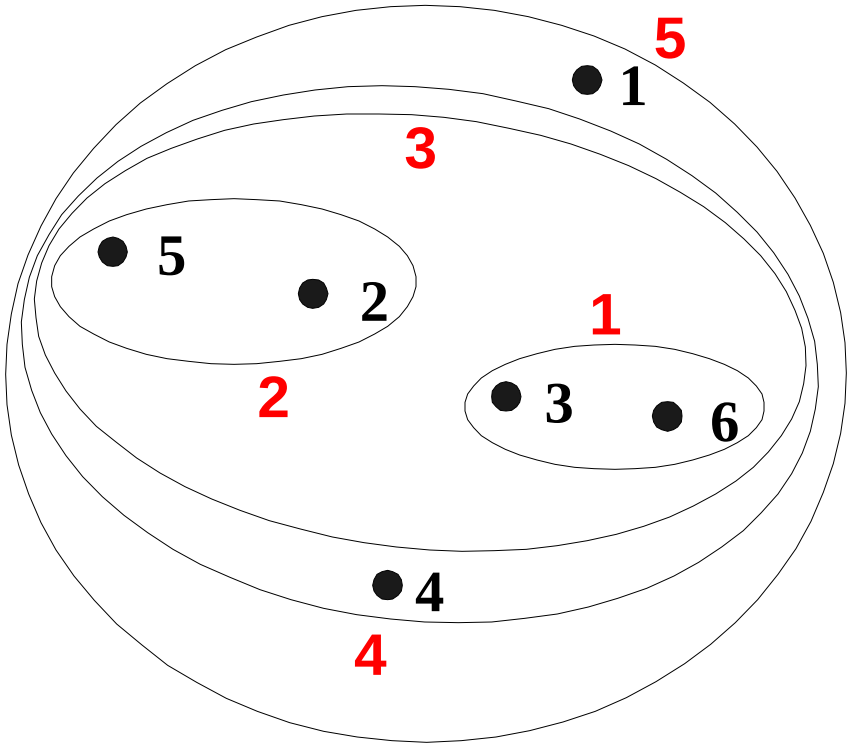
- Proximity of two clusters is based on the two closest points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph
- Example:



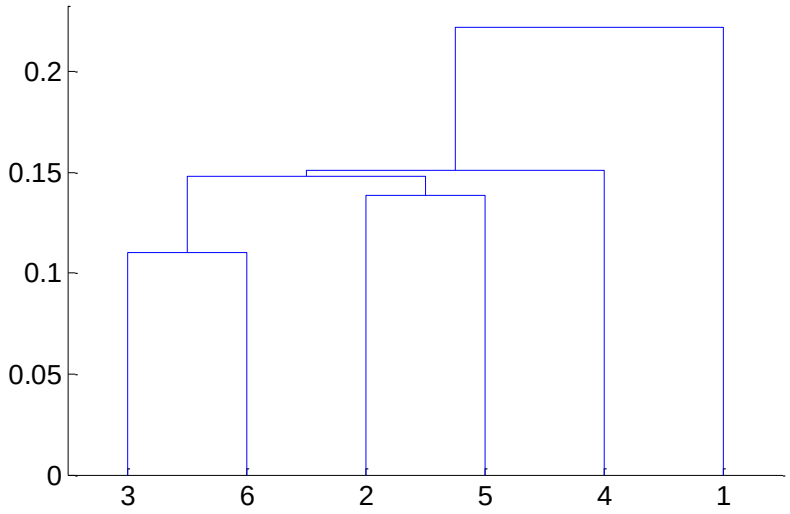
Distance Matrix:

	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

Hierarchical Clustering: MIN



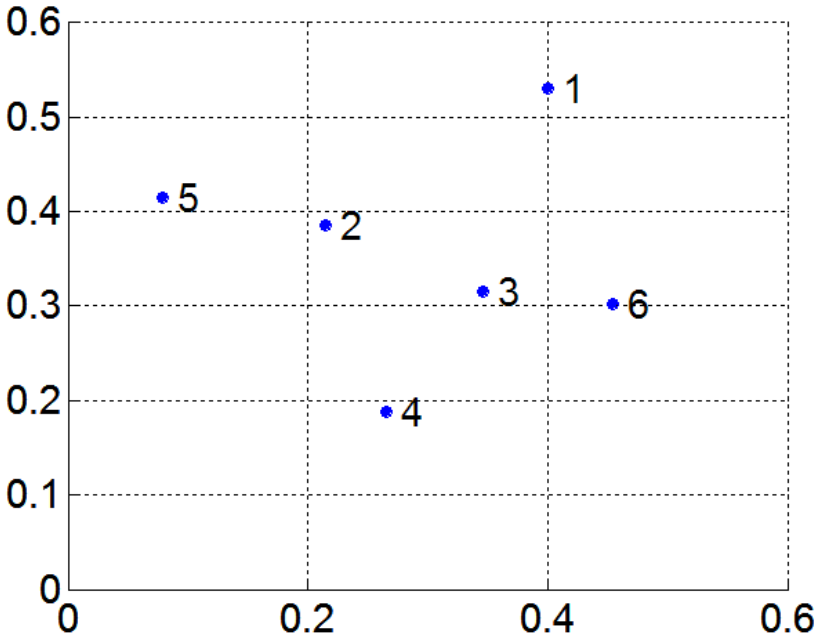
Nested Clusters



Dendrogram

MAX or Complete Linkage

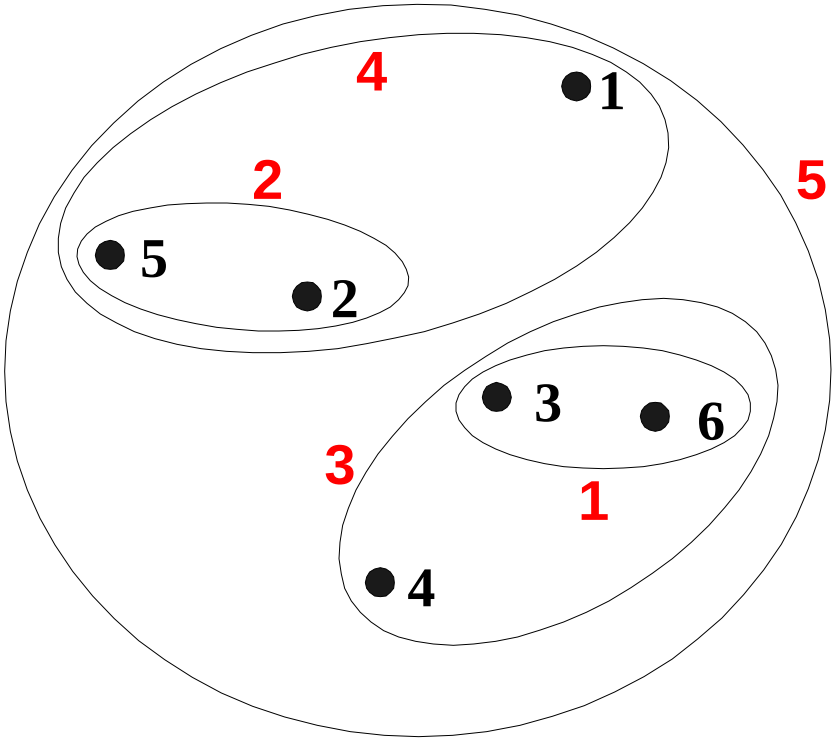
- Proximity of two clusters is based on the two most distant points in the different clusters
  - Determined by all pairs of points in the two clusters



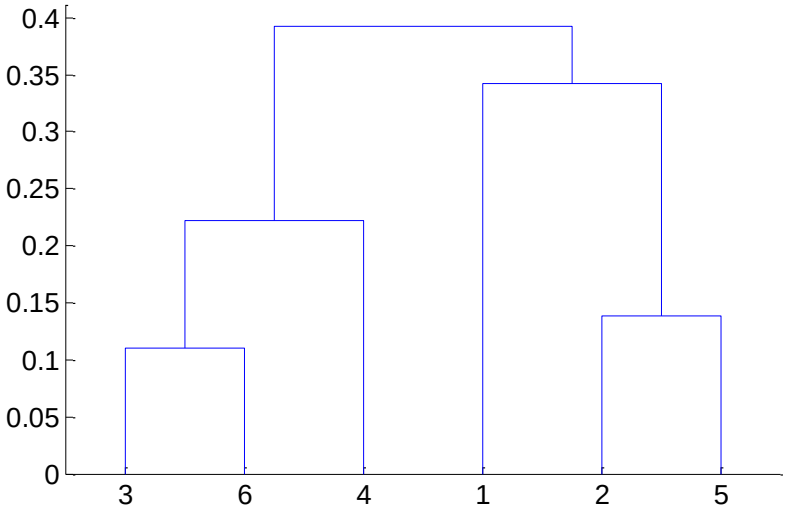
Distance Matrix:

	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

Hierarchical Clustering: MAX



Nested Clusters



Dendrogram

Hierarchical Clustering

Two main types of algorithms:

– Agglomerative (bottom-up merging)

- Start with the points as individual clusters
- Merge clusters until only one is left

– Divisive (top-down splitting)

- Start with all the points as one cluster
- Split clusters until only singleton clusters remain

– Agglomerative is more popular

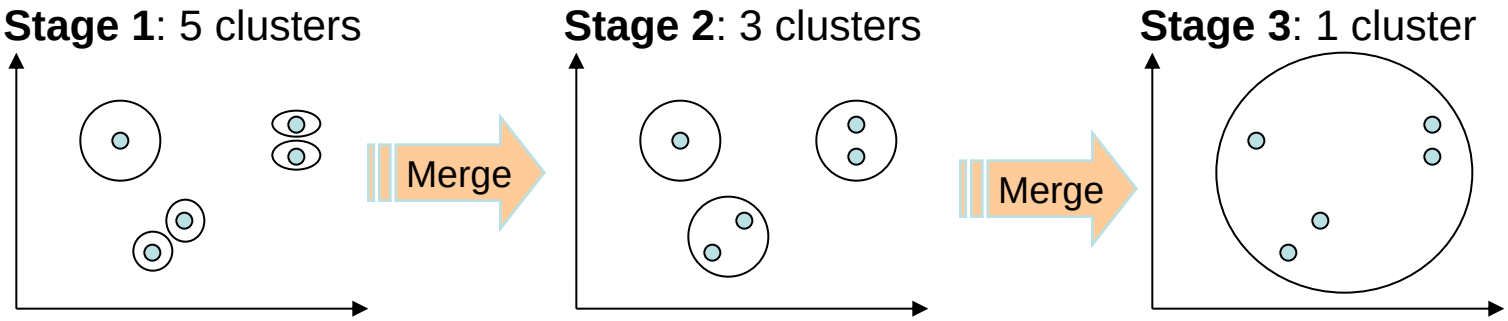
• Traditional hierarchical algorithms use a similarity or distance matrix, and merge or split one cluster at a time

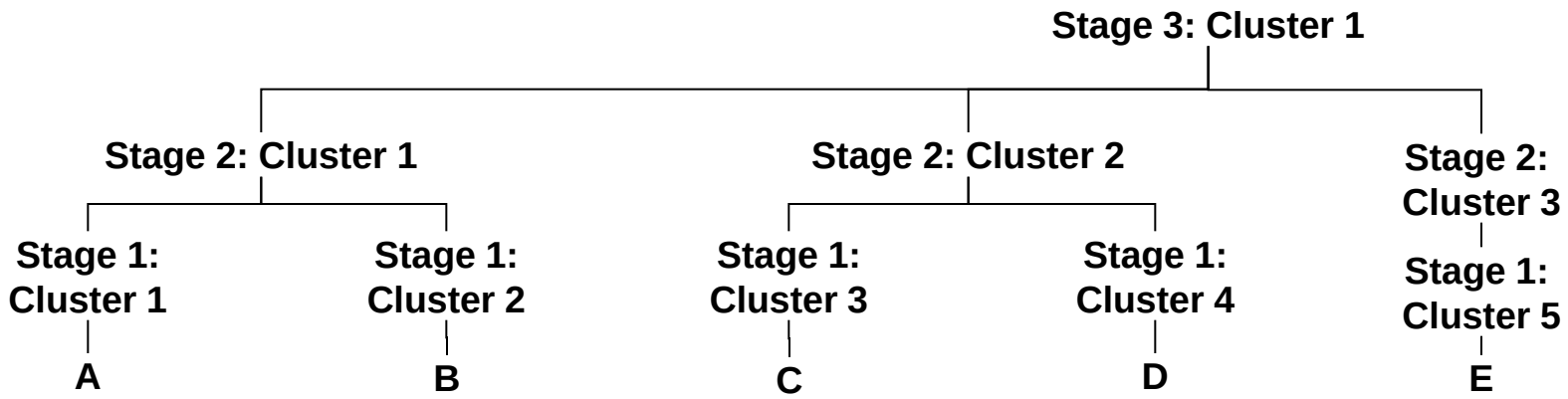
Agglomerative Clustering (Bottom-Up Merging)

- Bottom-up merging techniques are called agglomerative as they agglomerate (merge) smaller clusters into bigger ones.
- Typically, clusters are merged if the distance metric between the two clusters is less than a certain threshold.
- The threshold increases at each stage of merging.
- A variety of distance metrics can be used to calculate the distance (inter-cluster similarity) between clusters

Agglomerative Clustering (Bottom-Up Merging)

The sequence of graphs below show how a bottom-up merging techniques may proceed. Each example starts off in its own cluster. At each stage, we use the distance threshold to decide which clusters to merge, until eventually we have only one super-cluster. (We may also stop at some other termination condition - e.g. we may stop when we have less than a certain number of super-clusters.)

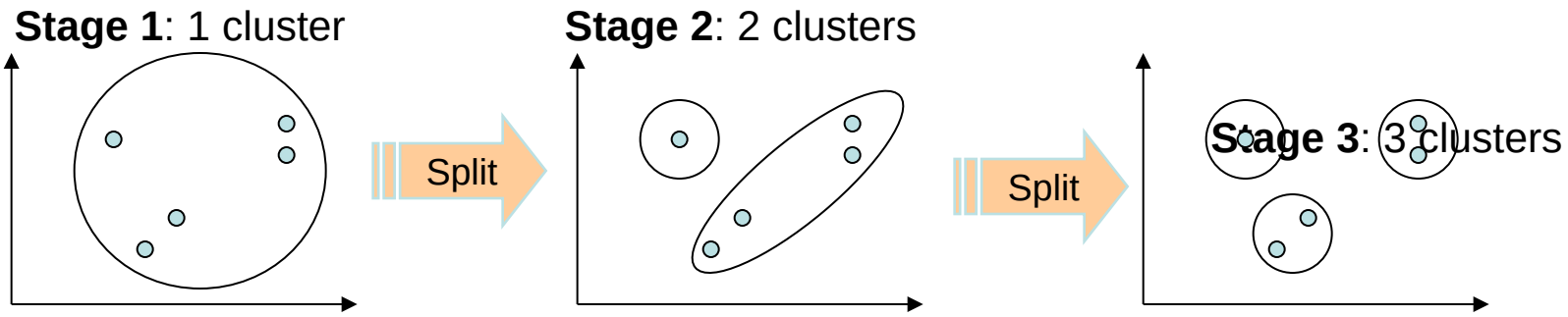


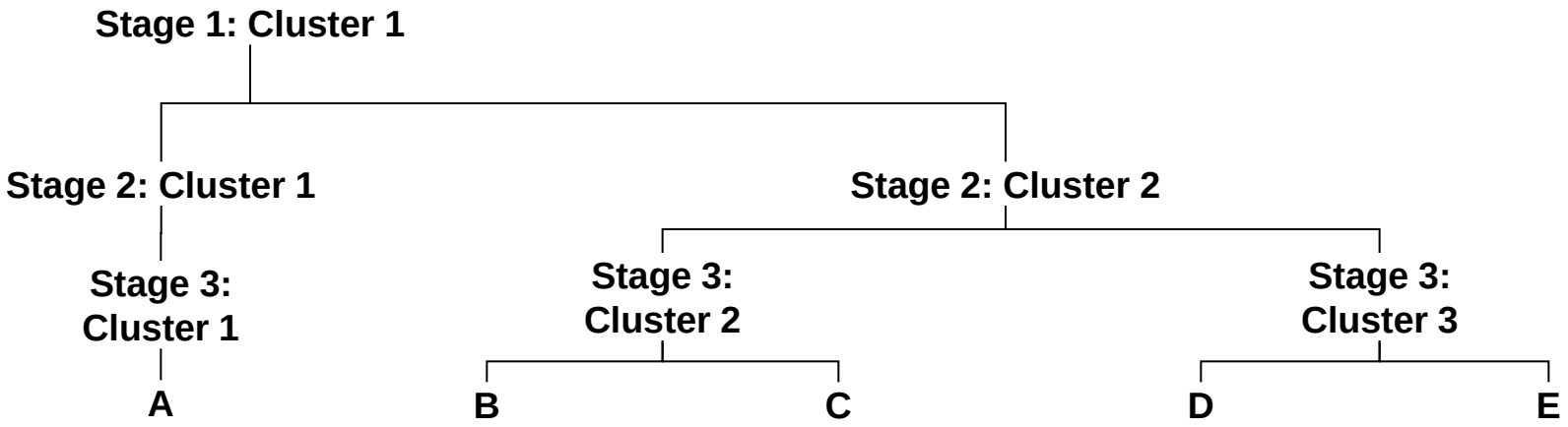


Source: Alan Abrahams

Divisive Clustering (Top-down Splitting)

The sequence of graphs below show how a **top-down splitting** techniques may proceed. We start off with one super-cluster. At each stage, we decide where to split, until eventually we have reasonable sub-clusters.





Source: Alan Abrahams

Agglomerative Algorithms

- Agglomerative algorithms start with each inidividual item in its own cluster and iterative merge clusters until all items belong in on cluster.
- Key operation is the computation of the proximity of two clusters.
- Different approaches to defining the distance between clusters distinguishes the different algorithms.
  - **Single link**
  - **Complete link (Farthest neighbor)**
  - **Average link**
  - **Centroid**
  - **Medoid**

Agglomerative Algorithms

- In the algorithm given on the next slide:
  - d: the threshold distance
  - k: the number of clusters
  - K: the set of clusters
- The procedure *NewClusters* determine how to form the next level clusters from the previous level. This is where the different types of agglomerative algorithms differ.
- Different approaches to defining the distance between clusters results in the different algorithms.
  - **Single link**
  - **Complete link (Farthest neighbor)**
  - **Average link**
  - **Centroid**
  - **Medoid**

Agglomerative Algorithm

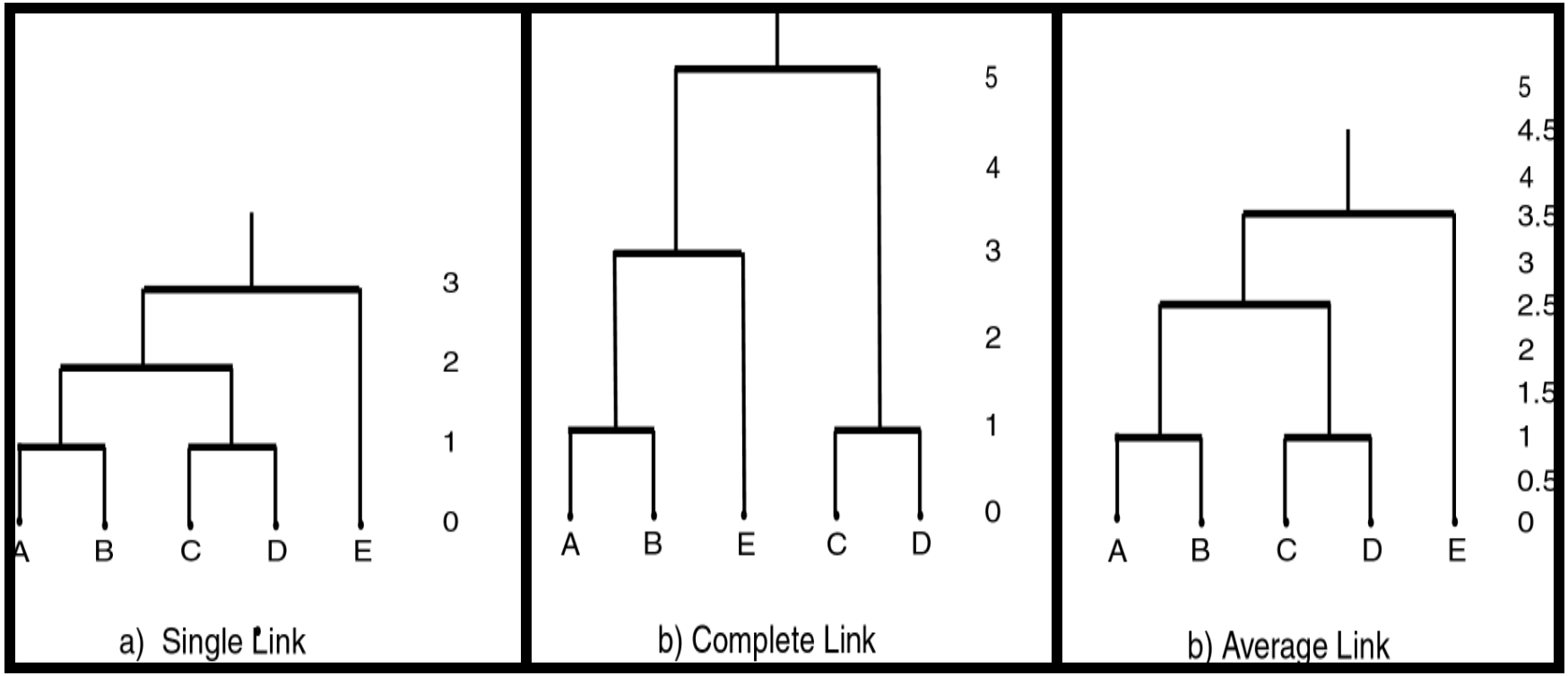
Input:
  $D = \{t_1, t_2, ..., t_n\}$ // Set of elements
  A // Adjacency matrix showing distance between elements.
Output:
  DE // Dendrogram represented as a set of ordered triples.
Agglomerative Algorithm:
  $d = 0$;
  $k = n$;
  $K = \{\{t_1\}, ..., \{t_n\}\}$;
  $DE = \{< d, k, K >\}$; // Initially dendrogram contains each element in its own cluster
  repeat
    $oldk = k$;
    $d = d + 1$;
    A_d = Vertex adjacency matrix for graph with threshold distance of d;
    $< k, K >= NewClusters(A_d, D)$;
    if $oldk \neq k$ then
      $DE = DE \cup < d, k, K >$; // New set of clusters added to dendrogram.
  until $k = 1$

Basic agglomerative algorithms

- Agglomerative algorithm using complete-link (max) technique
  - Merged if the maximum distance is less than or equal to the distance threshold.
  - Clusters found using the complete link tend to be more compact than the single link tehnique
  - Tends to break large clusters.
  - Less susceptible to noise and outliers.

- Agglomerative algorithm using average-link technique
  - Merged if the average distance is less than or equal to the distance threshold
  - Compromise between Single and Complete Link.
  - Need to use average connectivity for scalability since total connectivity favors large clusters.
  - Less susceptible to noise and outliers.

Agglomerative Clustering



Density-Based Clustering Methods

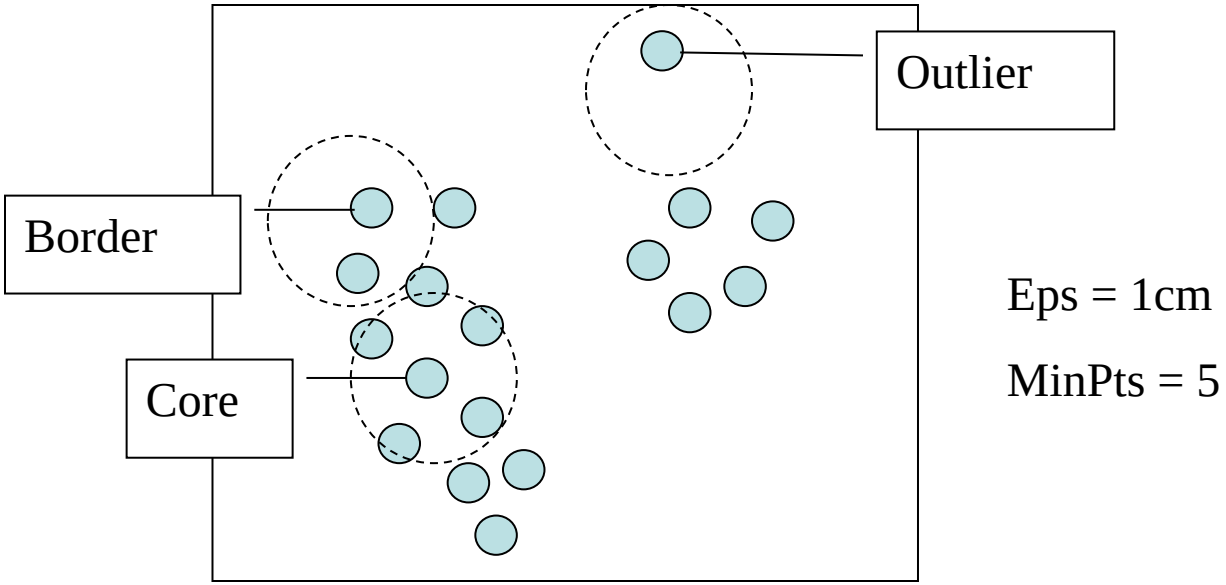
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

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 at least a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
    - Counts the point itself
  - A border point is not a core point, 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

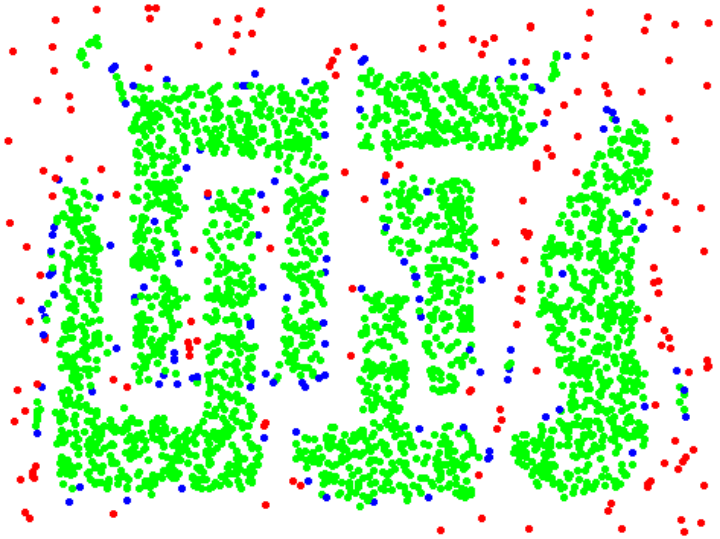
DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: Core, Border and Noise Points

Original Points



Point types: core, border and noise

Eps = 10, MinPts = 4

DBSCAN Algorithm

- Form clusters using core points, and assign border points to one of its neighboring clusters

1: Label all points as core, border, or noise points.

2: Eliminate noise points.

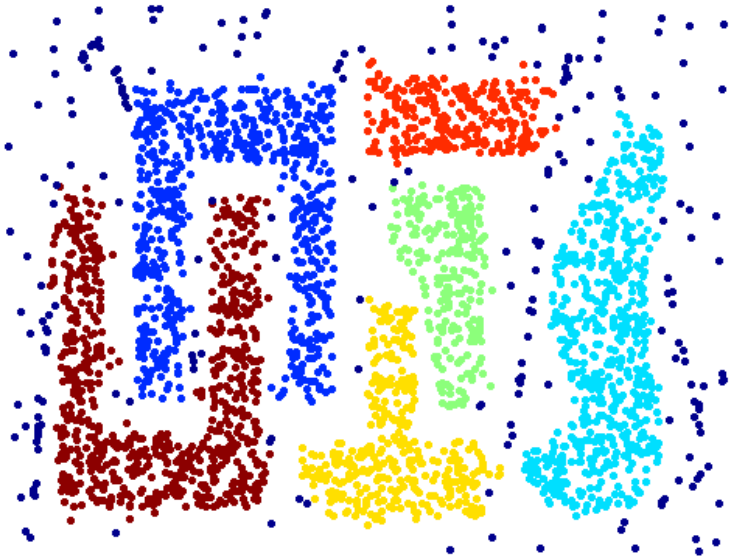
3: Put an edge between all core points within a distance *Eps* of each other.

4: Make each group of connected core points into a separate cluster.

5: Assign each border point to one of the clusters of its associated core points

When DBSCAN Works Well

Original Points



Clusters (dark blue points indicate noise)

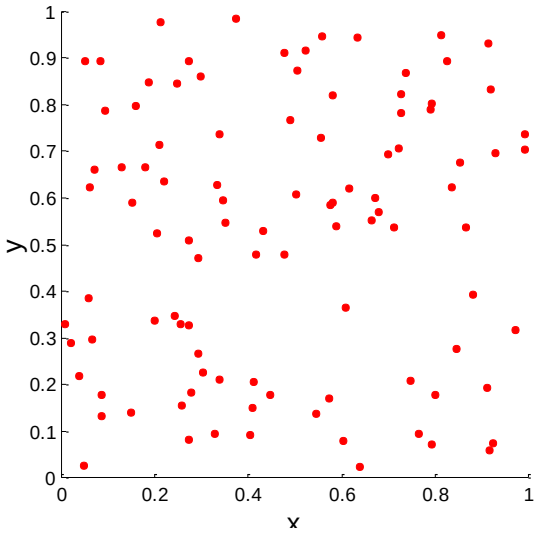
- Can handle clusters of different shapes and sizes
- Resistant to noise

Cluster Validity

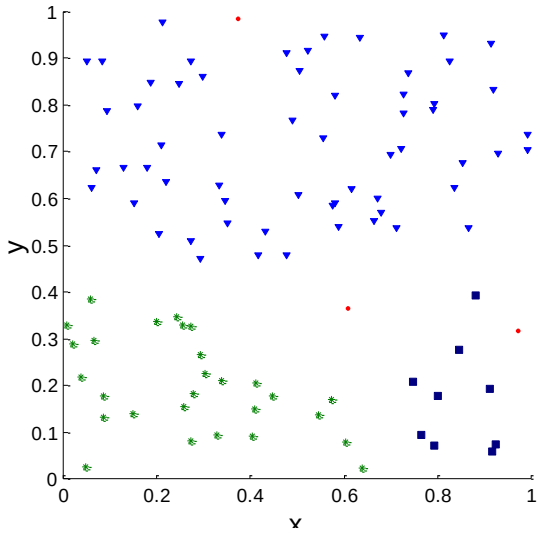
- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?
- But “clusters are in the eye of the beholder”!
  - In practice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

Clusters found in Random Data

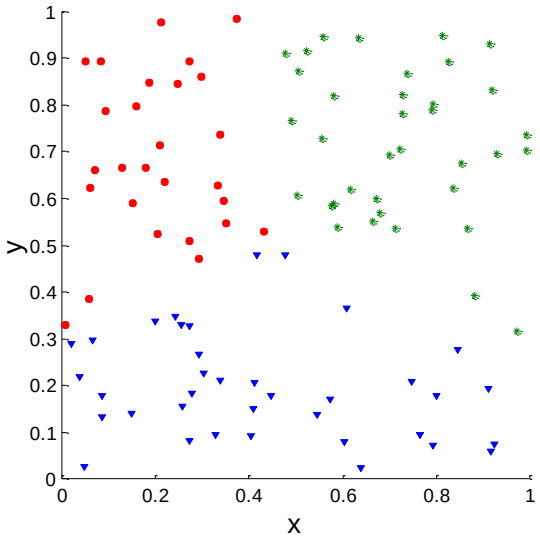
Random Points



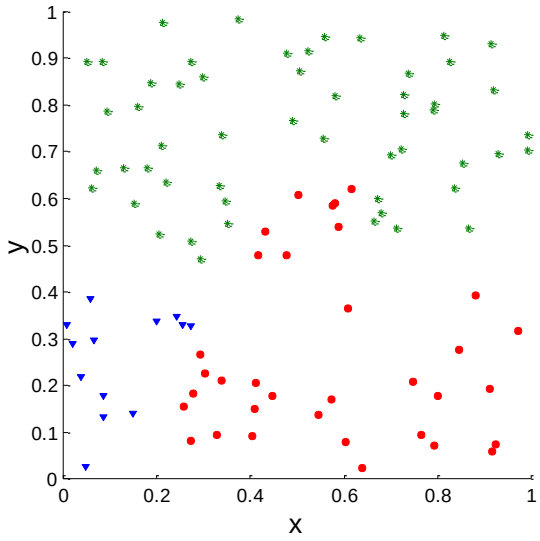
DBSCAN



K-means



Complete Link



Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following two types.
  - Supervised: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
    - Often called *external indices* because they use information external to the data
  - Unsupervised: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
    - Often called *internal indices* because they only use information in the data

- You can use supervised or unsupervised measures to compare clusters or clusterings

Unsupervised Measures: Cohesion and Separation

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

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

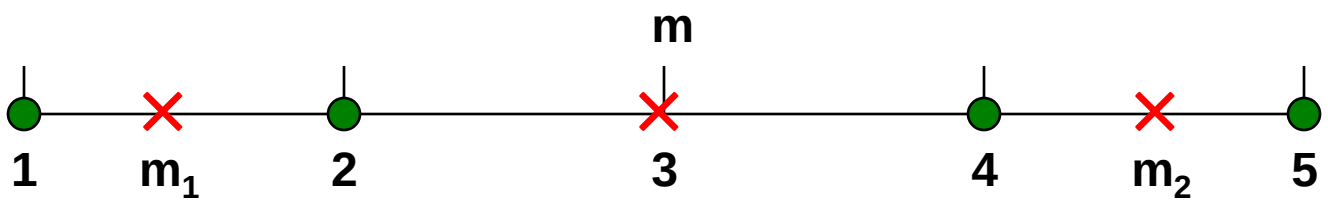
  - Separation is measured by the between cluster sum of squares

$$SSB = \sum_{i} |C_i| (m - m_i)^2$$

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

Unsupervised Measures: Cohesion and Separation

- Example: SSE
  - SSB + SSE = constant



K=1 cluster:

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

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

$$Total = 10 + 0 = 10$$

K=2 clusters:

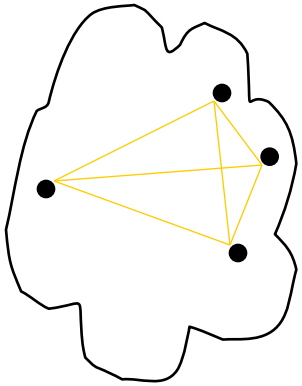
$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

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

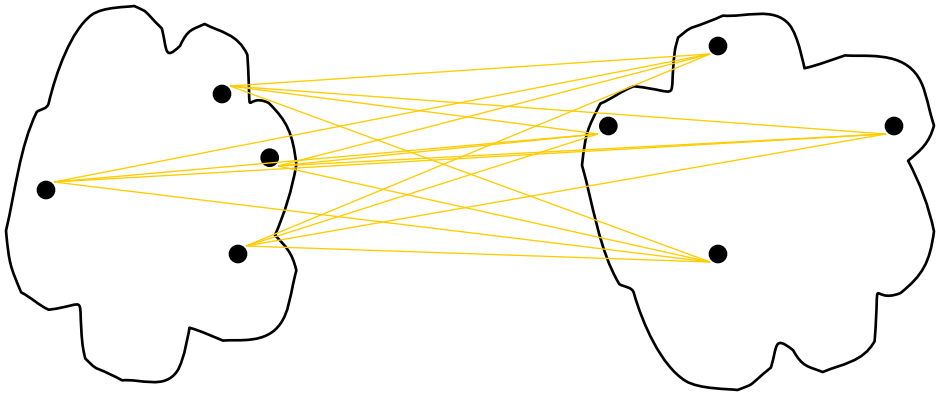
$$Total = 1 + 9 = 10$$

Unsupervised Measures: Cohesion and Separation

- A proximity graph-based approach can also be used for cohesion and separation.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



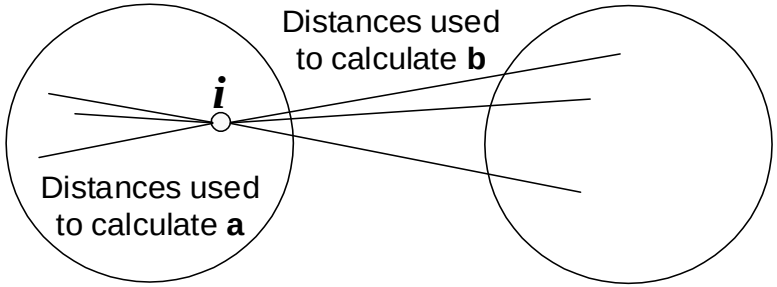
separation

Unsupervised Measures: Silhouette Coefficient

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

    s = (b – a) / max(a,b)

  - Value can vary between -1 and 1
  - Typically ranges between 0 and 1.
  - The closer to 1 the better.



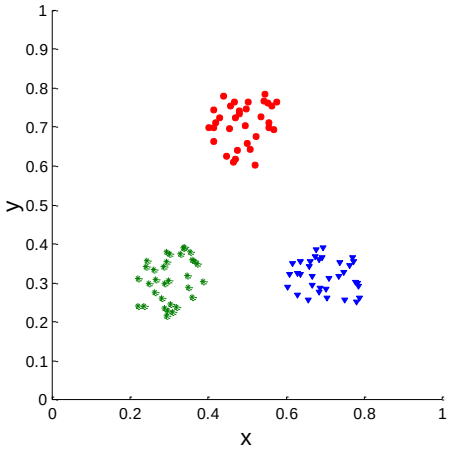
- Can calculate the average silhouette coefficient for a cluster or a clustering

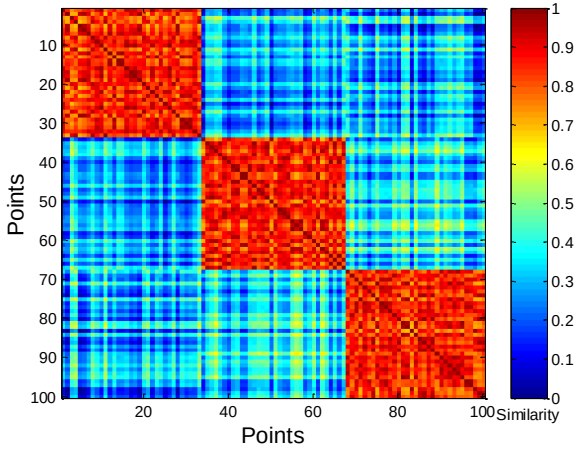
Measuring Cluster Validity Via Correlation

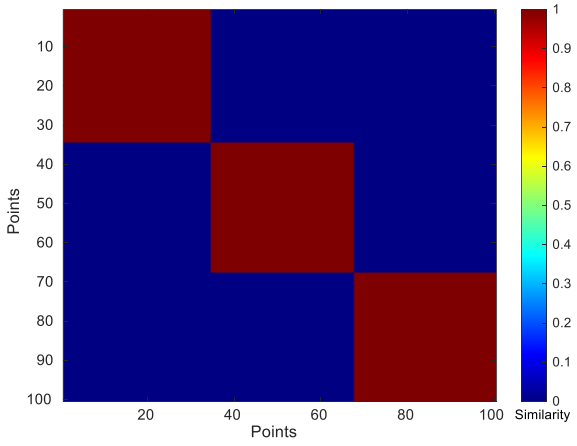
- Two matrices
  - Proximity Matrix
  - Ideal Similarity Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation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

Measuring Cluster Validity Via Correlation

- Correlation of ideal similarity and proximity matrices for the K-means clusterings of the following well-clustered data set.



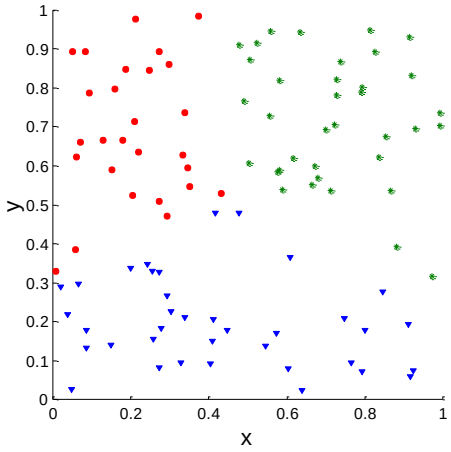


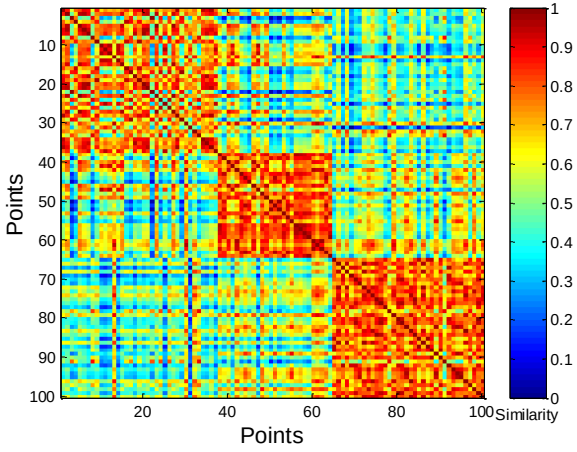


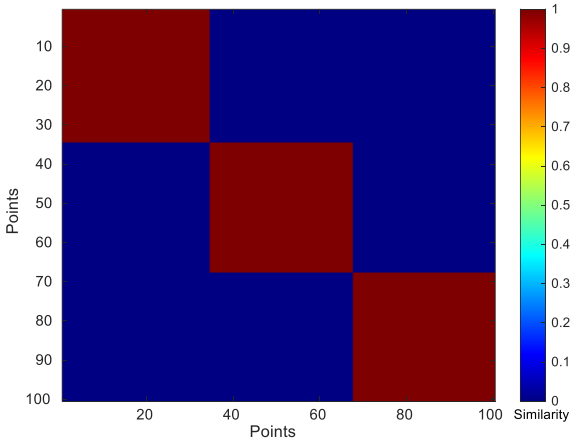
Corr = 0.9235

Measuring Cluster Validity Via Correlation

- Correlation of ideal similarity and proximity matrices for the K-means clusterings of the following random data set.







K-means

Corr = 0.5810

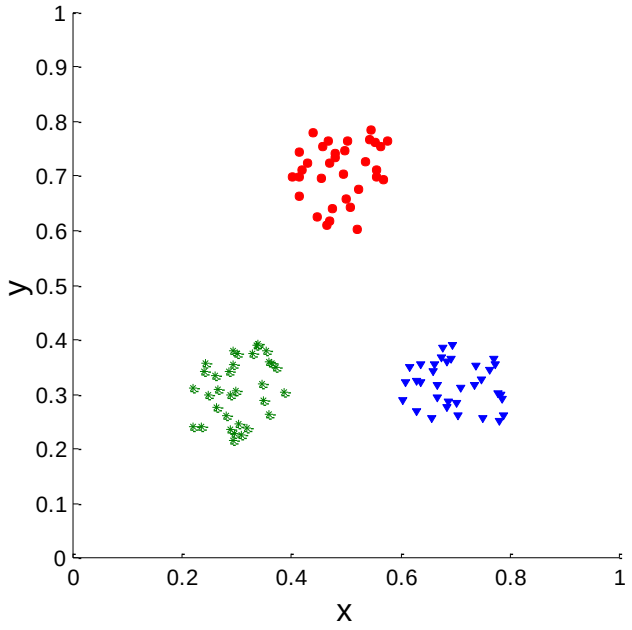
Judging clustering quality visually using a similarity matrix

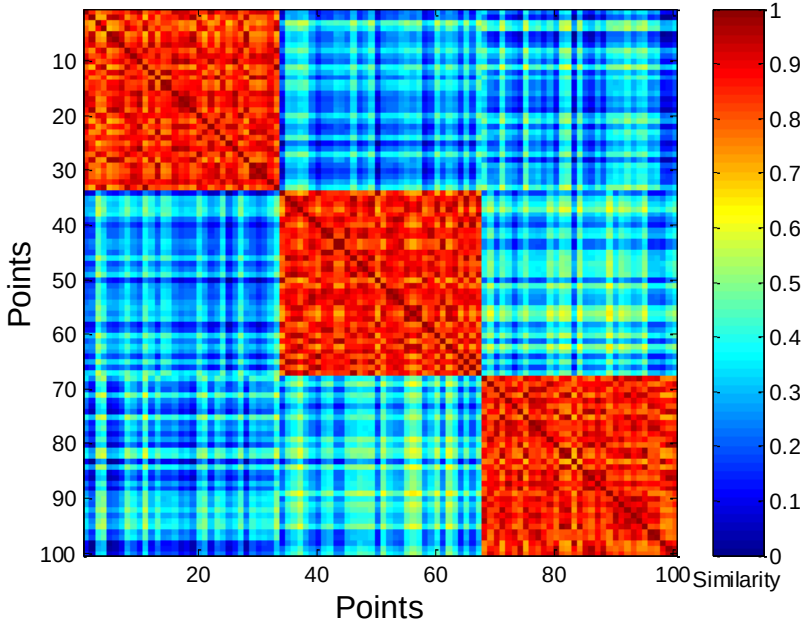
- A similarity matrix represents pairwise similarities between data points.

- Visualizing the Matrix
  - The matrix is typically displayed as a heatmap, where colors indicate similarity levels.
  - Well-defined clusters appear as blocks along the diagonal, showing high intra-cluster similarity.
  - If clusters are poorly separated, the heatmap may show blurred boundaries or scattered similarities.
- Interpretation
  - Clear diagonal blocks → Strong clustering structure.
  - Scattered similarities → Overlapping or weak clusters.
  - Uniform color → No meaningful clustering.
- Applications
  - Used in hierarchical clustering to validate dendrogram structures.
  - Helps assess spectral clustering, where similarity matrices drive the clustering process.
  - Useful in medical imaging and bioinformatics for pattern recognition.

Judging a Clustering Visually by its Similarity Matrix

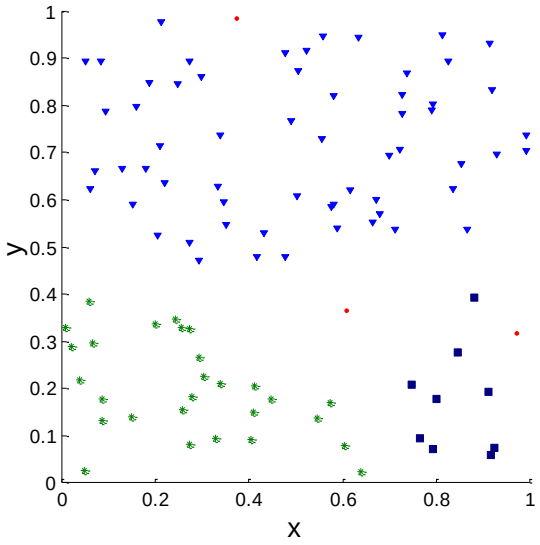
- Order the similarity matrix with respect to cluster labels and inspect visually.

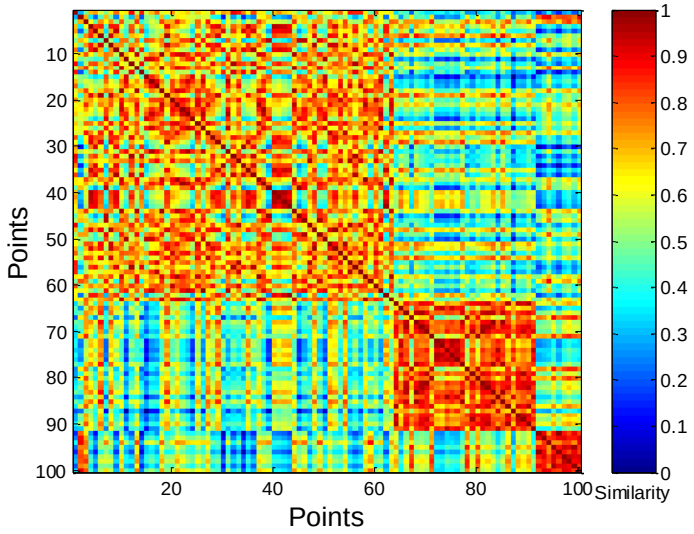




Judging a Clustering Visually by its Similarity Matrix

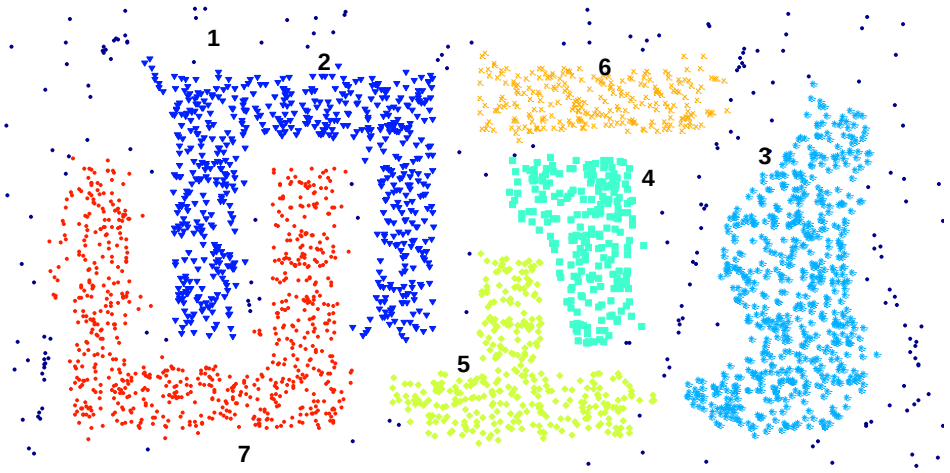
- Clusters in random data are not so crisp

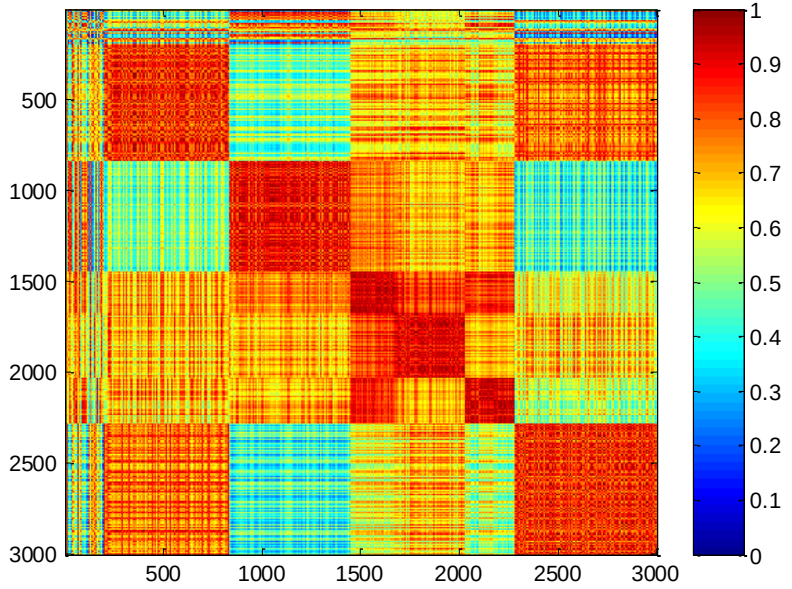




DBSCAN

Judging a Clustering Visually by its Similarity Matrix





DBSCAN

Determining the Correct Number of Clusters Using SSE curve

The **Sum of Squared Errors (SSE) curve**, often called the **Elbow Method**, is a popular technique for determining the optimal number of clusters in **K-Means clustering**.

Here's how it works:

1. Compute SSE for Different Cluster Counts

- SSE measures the total squared distance between each data point and its assigned cluster centroid.
- As the number of clusters **(K)** increases, SSE decreases because clusters become more refined.

2. Plot the SSE Curve

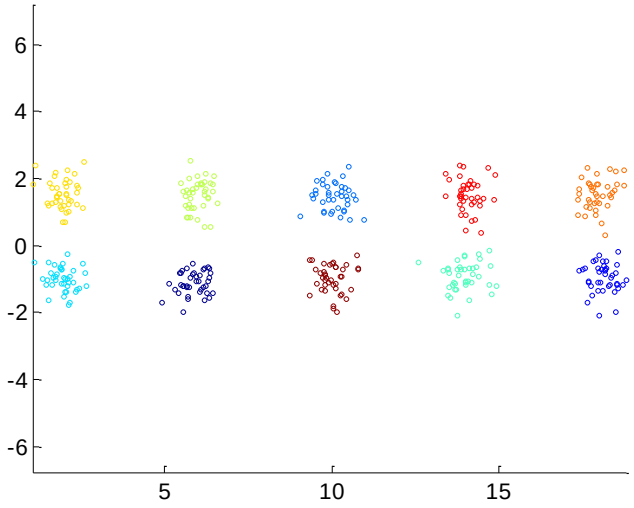
- The SSE values are plotted against different values of **K**.
- The curve typically shows a **sharp drop initially**, then levels off.

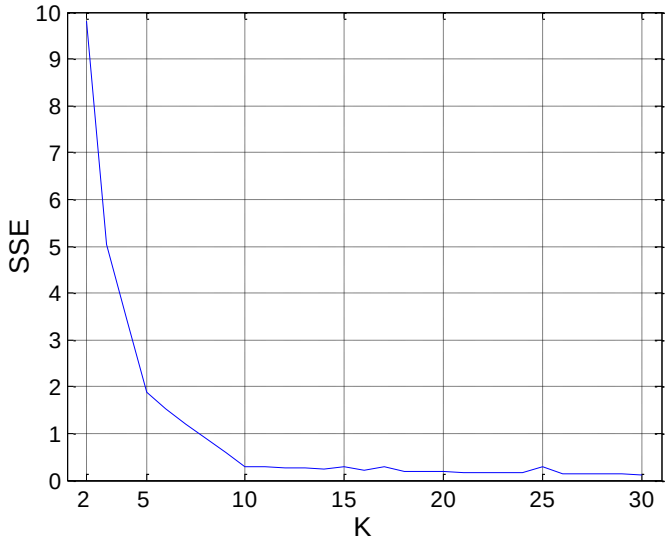
3. Identify the "Elbow" Point

- The **elbow** is where the SSE curve bends, indicating the optimal number of clusters.
- Beyond this point, adding more clusters **does not significantly reduce SSE**, meaning additional clusters may not provide meaningful separation.

Determining the Correct Number of Clusters

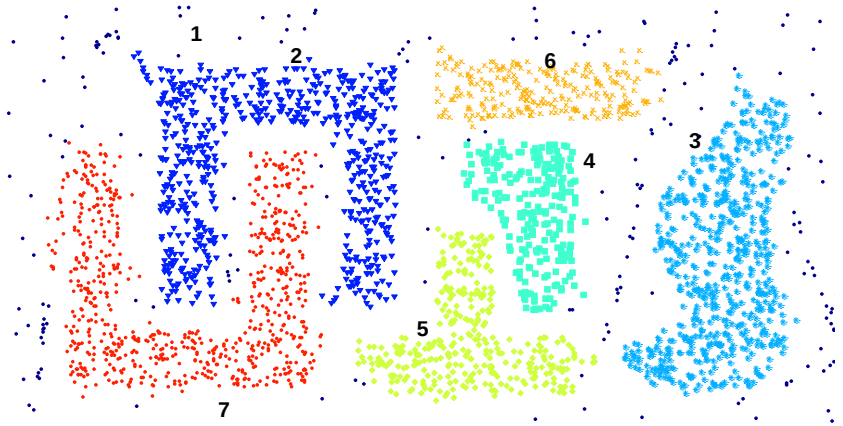
- SSE is good for comparing two clusterings or two clusters
- SSE can also be used to estimate the number of clusters

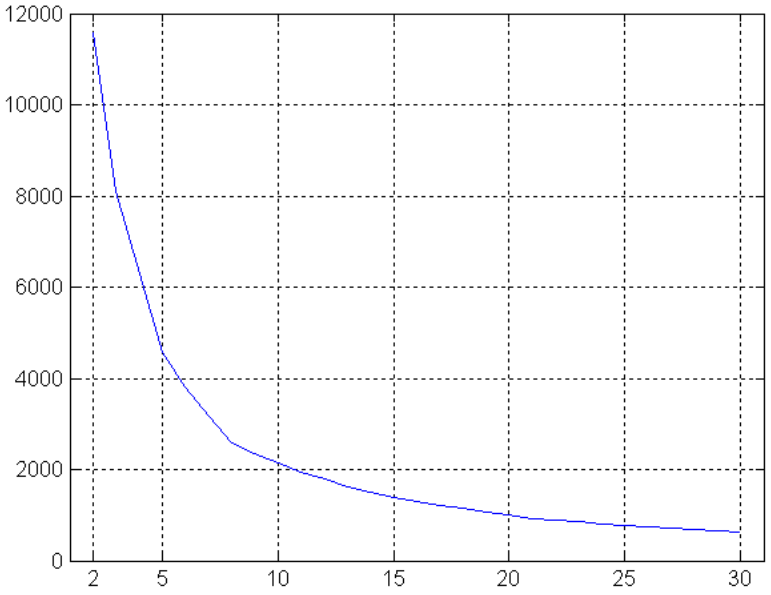




Determining the Correct Number of Clusters

- SSE curve for a more complicated data set





SSE of clusters found using K-means

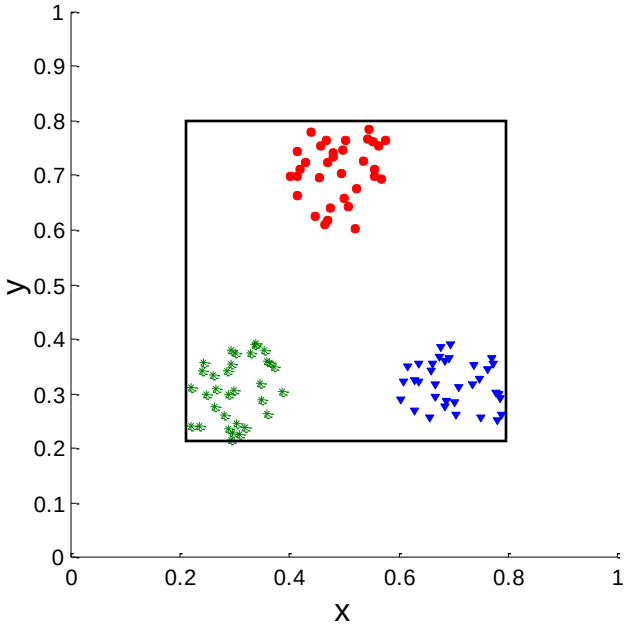
Assessing the Significance of Cluster Validity Measures

- 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
  - Compare the value of an index obtained from the given data with those resulting from random data.
    - Compute the evaluation index (e.g., Silhouette Score, Davies-Bouldin Index) for both.

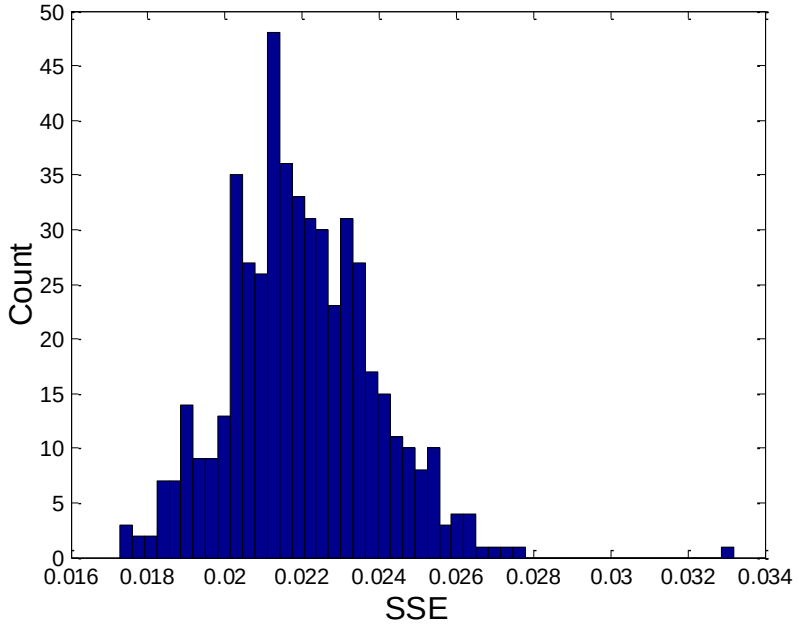
- Try to judge how likely it is that our observed value was achieved by random chance.
- If the index value from the real data is significantly different from those obtained from random data, it suggests that the clustering captures real structure.

Statistical Framework for SSE

- Example
  - Compare SSE of three cohesive clusters against three clusters in random data



SSE = 0.005

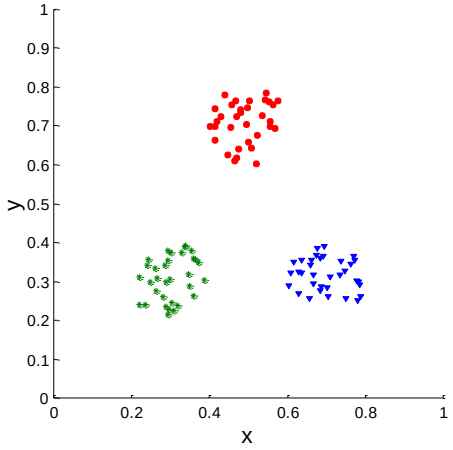


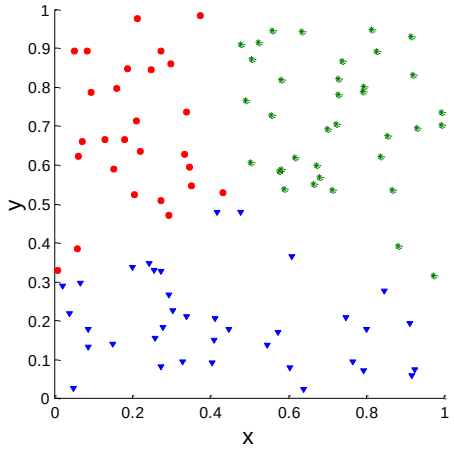
Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

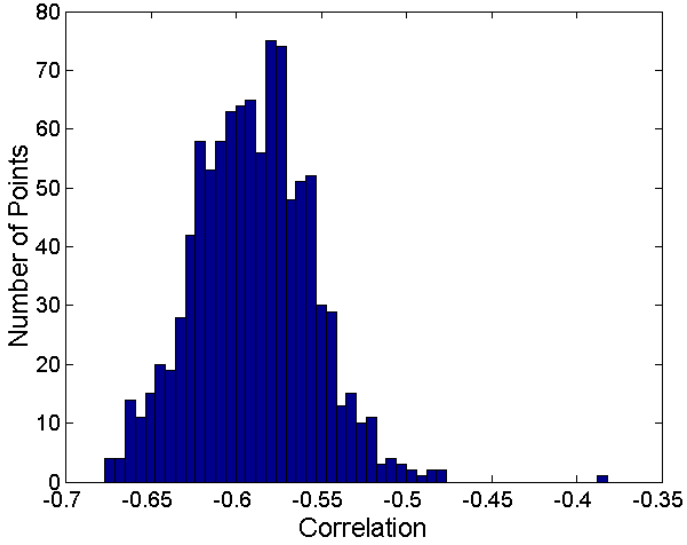
- The lowest SSE shown in histogram is 0.0173. For the three clusters, the SSE is 0.0050. We could therefore conservatively claim that there is less than a 1% chance that a clustering such as that of these three clusters could occur by chance.

Statistical Framework for Correlation

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







Corr = -0.9235

Corr = -0.5810

Correlation is negative because it is calculated between a distance matrix and the ideal similarity matrix. Higher magnitude is better.

Histogram of correlation for 500 random data sets of size 100 with *x* and *y* values of points between 0.2 and 0.8.

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**

- H. Xiong and Z. Li. *Clustering Validation Measures*. In C. C. Aggarwal and C. K. Reddy, editors, Data Clustering: Algorithms and Applications, pages 571–605. Chapman & Hall/CRC, 2013.