Machine Learning & Data Mining

What is Machine Learning?

- a branch of artificial intelligence, concerns the construction and study of systems that can learn from data.
- The core of machine learning deals with representation and generalization: Representation of data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the property that the system will perform well on unseen data instances
- Tom M. Mitchell: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E”

From Wikipedia (Machine Learning)

Machine Learning Types

- **Supervised learning**
  - Classification
  - Regression/Prediction
- **Unsupervised learning**
  - Clustering
- **Semi-supervised learning**
- **Association Analysis**
- **Reinforcement learning**

Growth of Machine Learning

- **Machine learning is preferred approach to**
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
- **This trend is accelerating**
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment
  - It turns out to be difficult to extract knowledge from human experts→*failure of expert systems in the 1980's.*

Data Mining/KDD

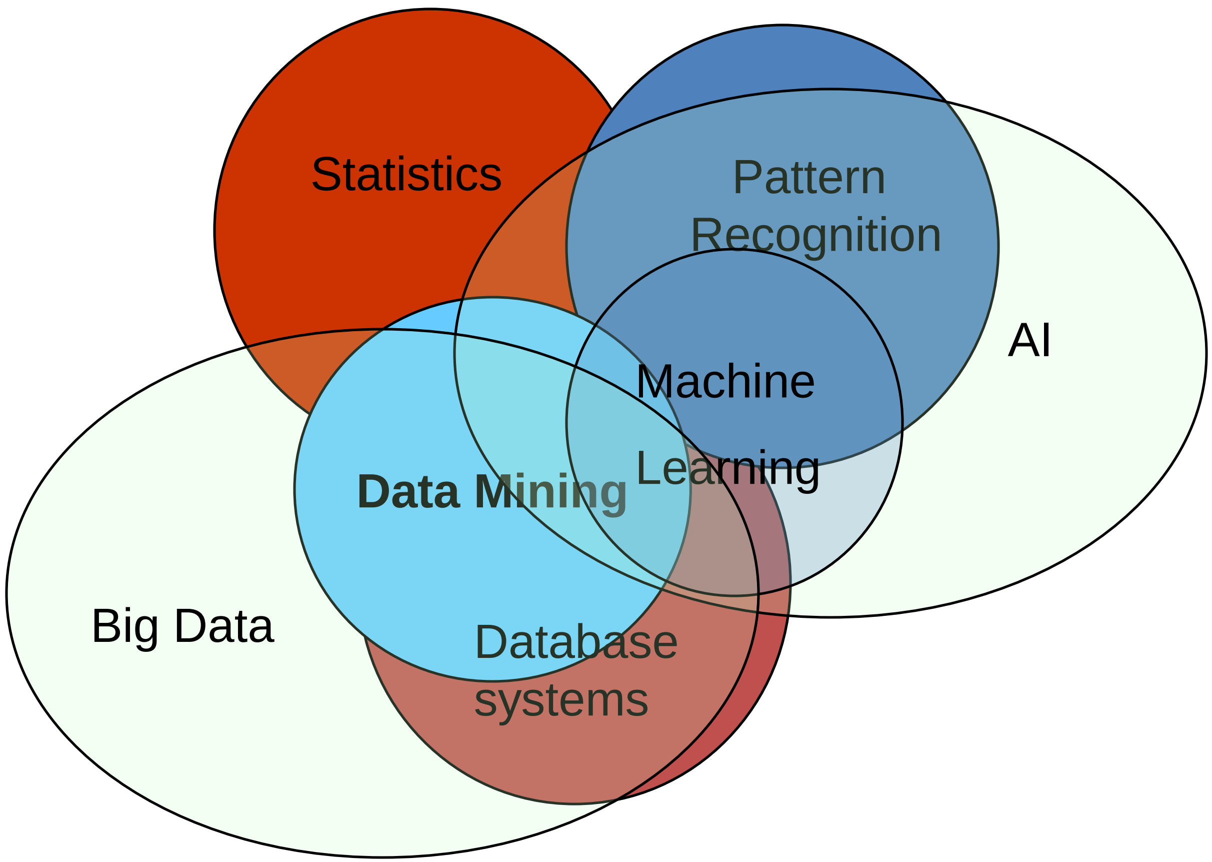
Definition := *"KDD is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"* (Fayyad)

Applications:

- Retail: Market basket analysis, Customer relationship management (CRM)
- Finance: Credit scoring, fraud detection
- Manufacturing: Optimization, troubleshooting
- Medicine: Medical diagnosis
- Telecommunications: Quality of service optimization
- Bioinformatics: Motifs, alignment
- ...

Machine Learning & Data Mining

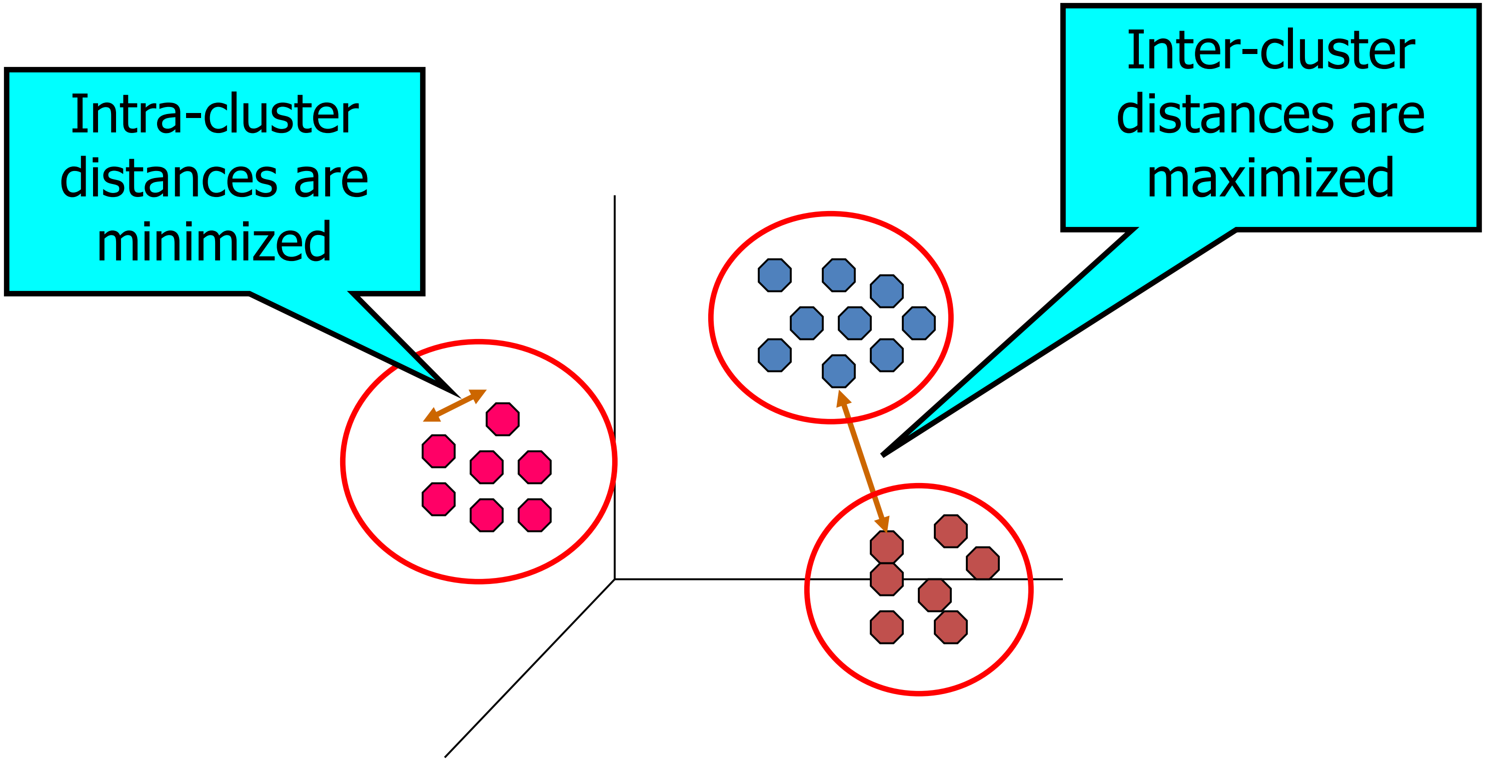
- Machine learning focuses on prediction, based on known properties learned from the training data.
- Data mining focuses on the discovery of (previously) unknown properties in the data. This is the analysis step of Knowledge Discovery in Databases.
- Data mining uses many machine learning methods, but often with a slightly different goal in mind
- Machine learning also employs data mining methods as "unsupervised learning" or as a preprocessing step to improve learner accuracy.


Statistics
Pattern Recognition
AI
Machine Learning
Data Mining
Big Data
Database systems

Unsupervised Learning: Cluster Analysis

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

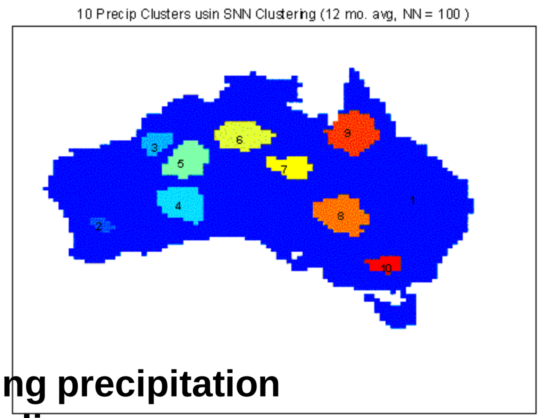


Applications of Cluster Analysis

- **Understanding**
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- **Summarization**
  - Reduce the size of large data sets



Clustering precipitation in Australia

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

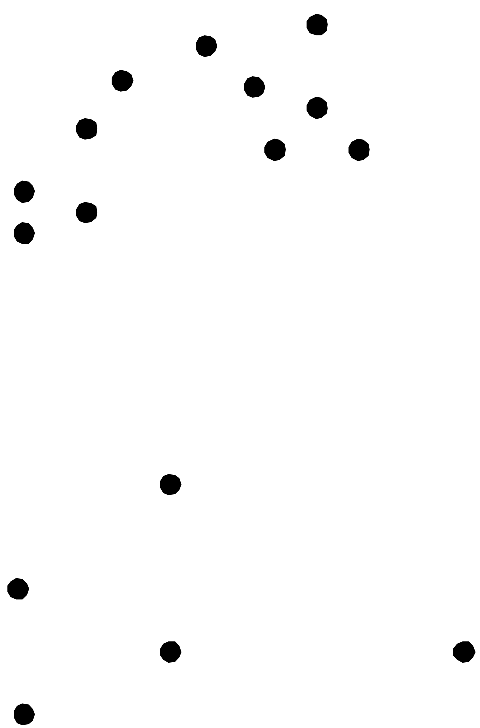
Two Clusters

Four Clusters

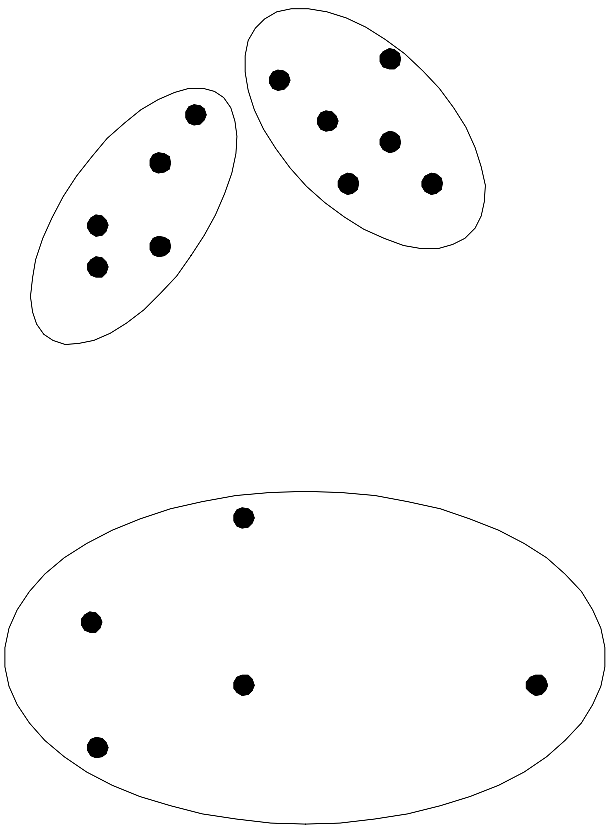
Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

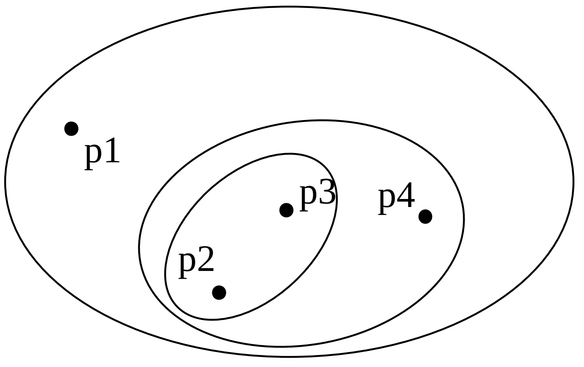


Original Points

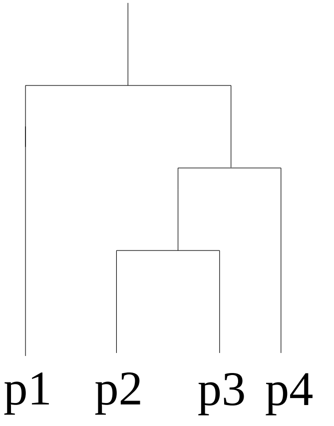


A Partitional Clustering

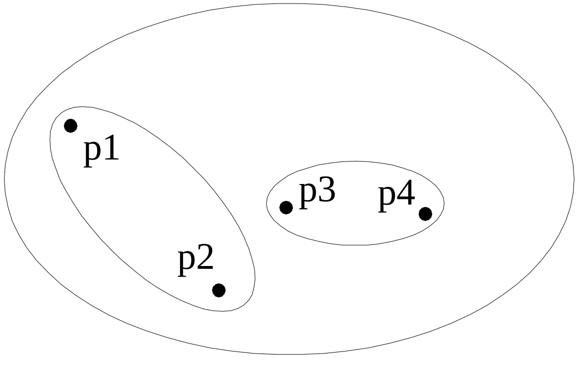
Hierarchical Clustering



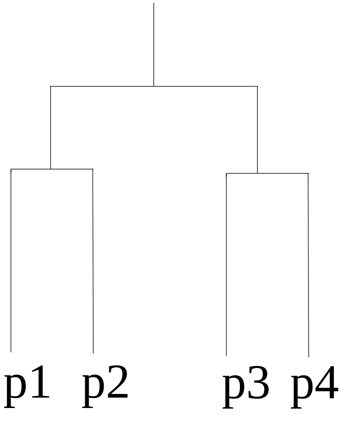
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster of widely different sizes, shapes, and densities

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Density-based clustering

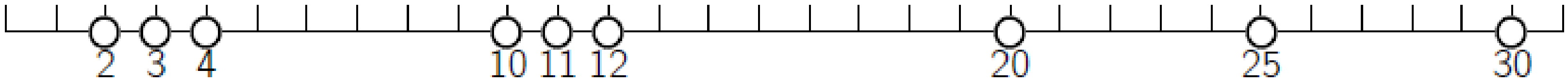
K-means Clustering

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

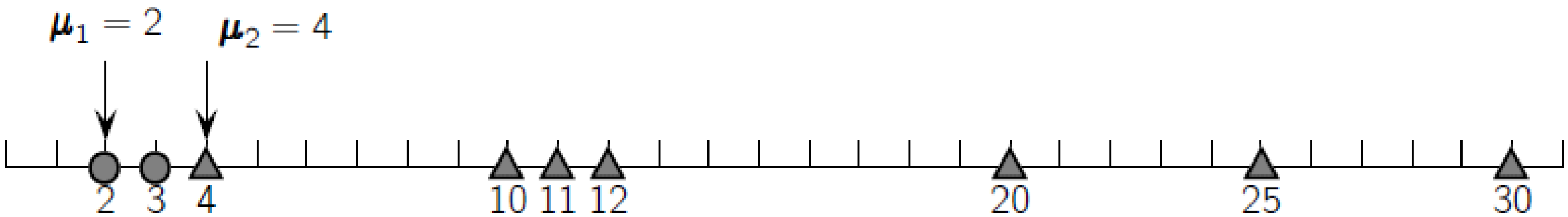
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change

K-means Clustering – Details

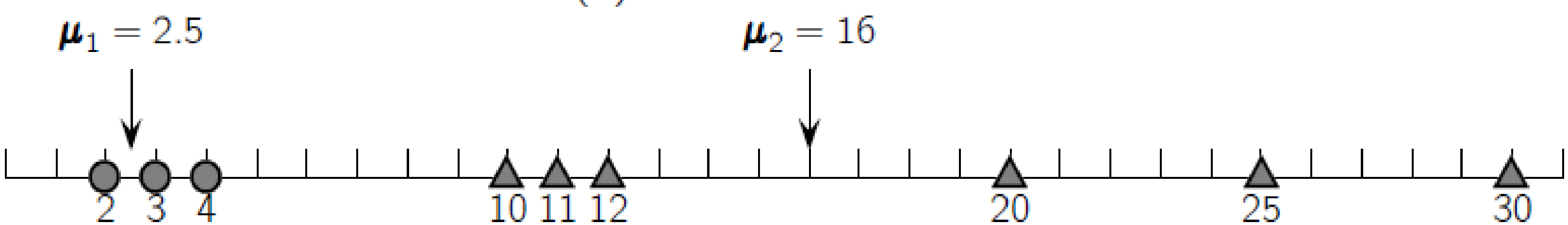
- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.



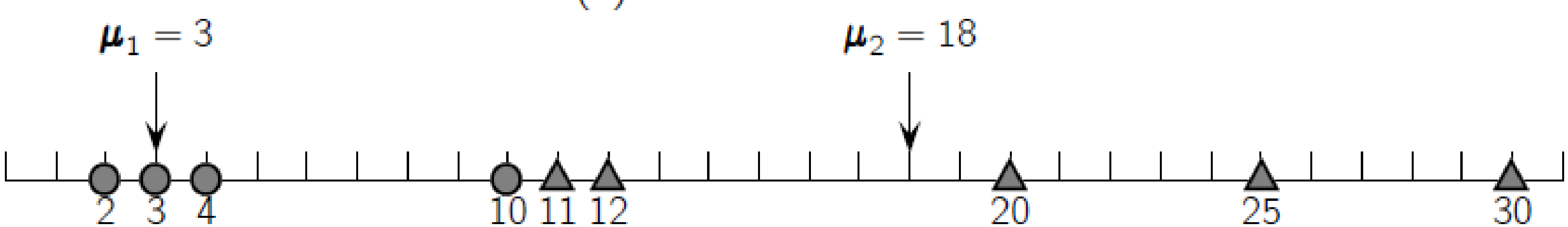
(a) Initial dataset



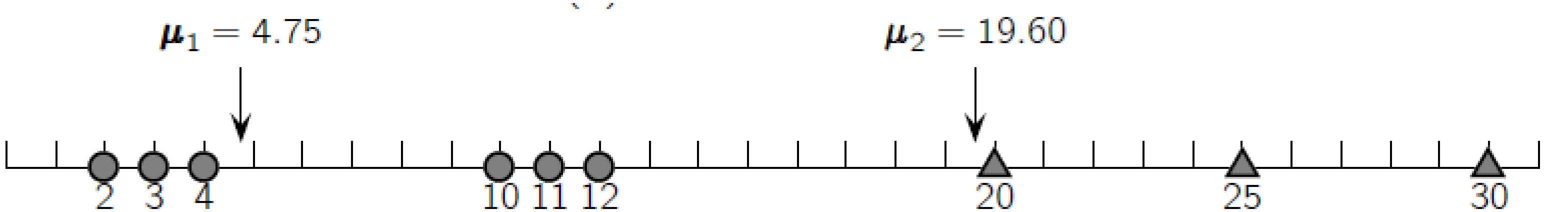
(b) Iteration: $t = 1$



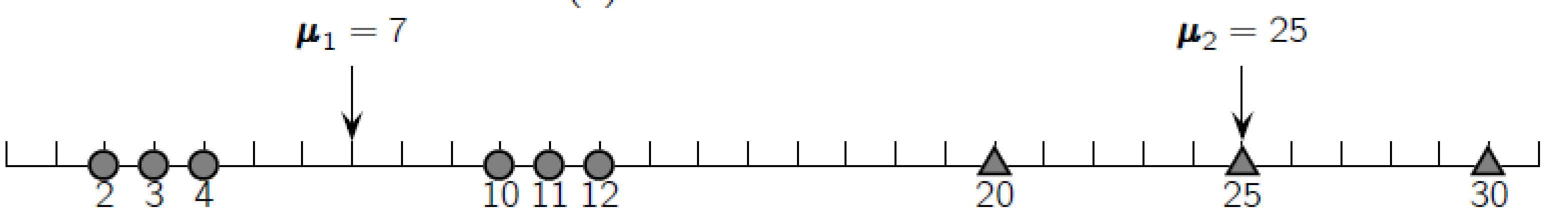
(c) Iteration: $t = 2$



(d) Iteration: $t = 3$



(e) Iteration: $t = 4$



(f) Iteration: $t = 5$ (converged)

K-means Clustering – Details

- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

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

  - *x* is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Algorithm 16.1: K-means Algorithm

K-means $(\boldsymbol{D}, k, \epsilon)$:

$t = 0$

Randomly initialize k centroids: $\boldsymbol{\mu}_1^t, \boldsymbol{\mu}_2^t, \ldots, \boldsymbol{\mu}_k^t$

repeat

  $t = t + 1$

  `// Cluster Assignment Step`

  foreach $\boldsymbol{x}_j \in \boldsymbol{D}$ **do**

    $j^* = \arg\min_i \{\|\boldsymbol{x}_j - \boldsymbol{\mu}_i^t\|^2\}$ `// Assign` $\boldsymbol{x}_j$ `to closest centroid`

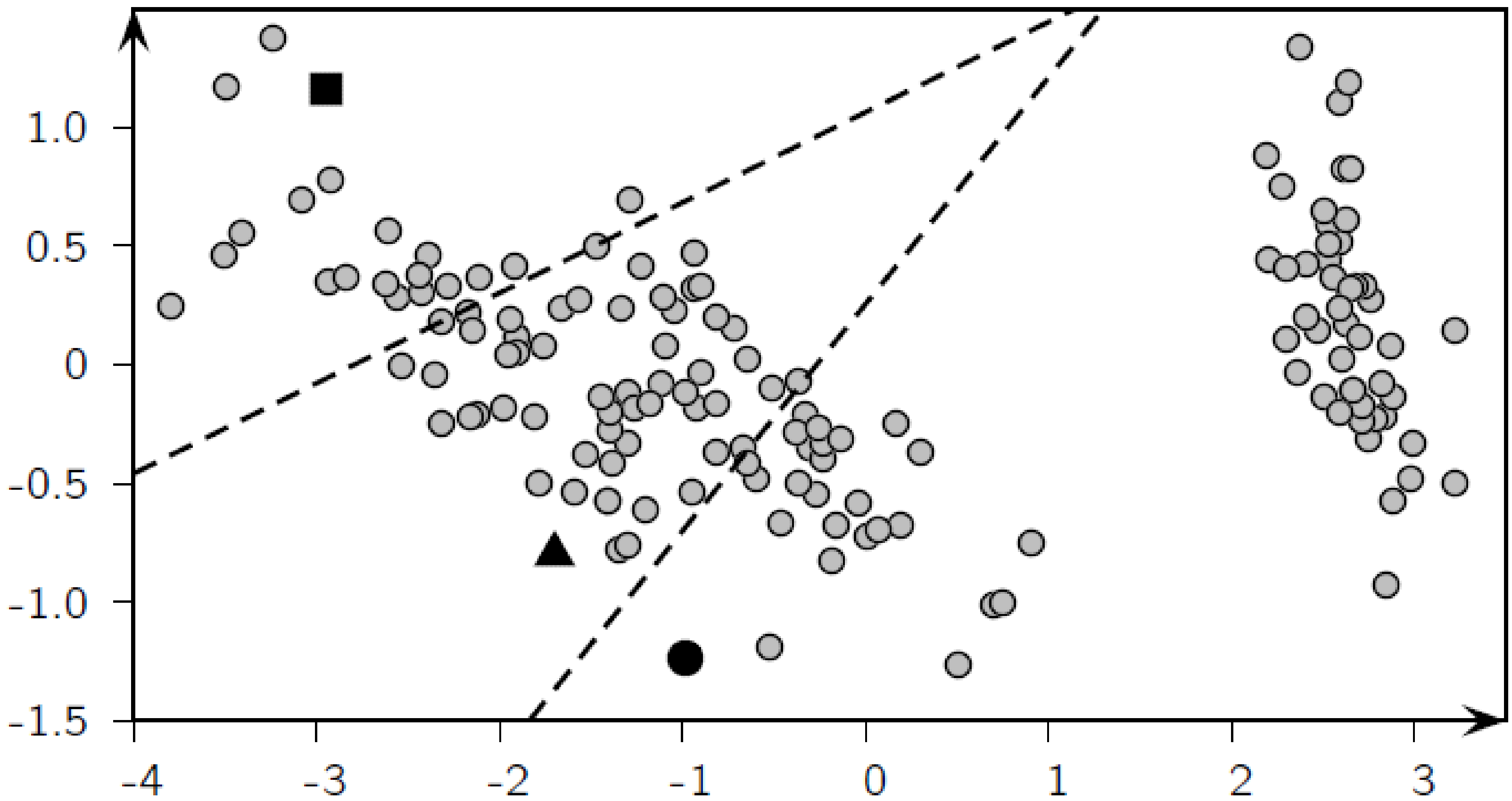
    $C_{j^*} = C_{j^*} \cup \{\boldsymbol{x}_j\}$

  `// Centroid Update Step`

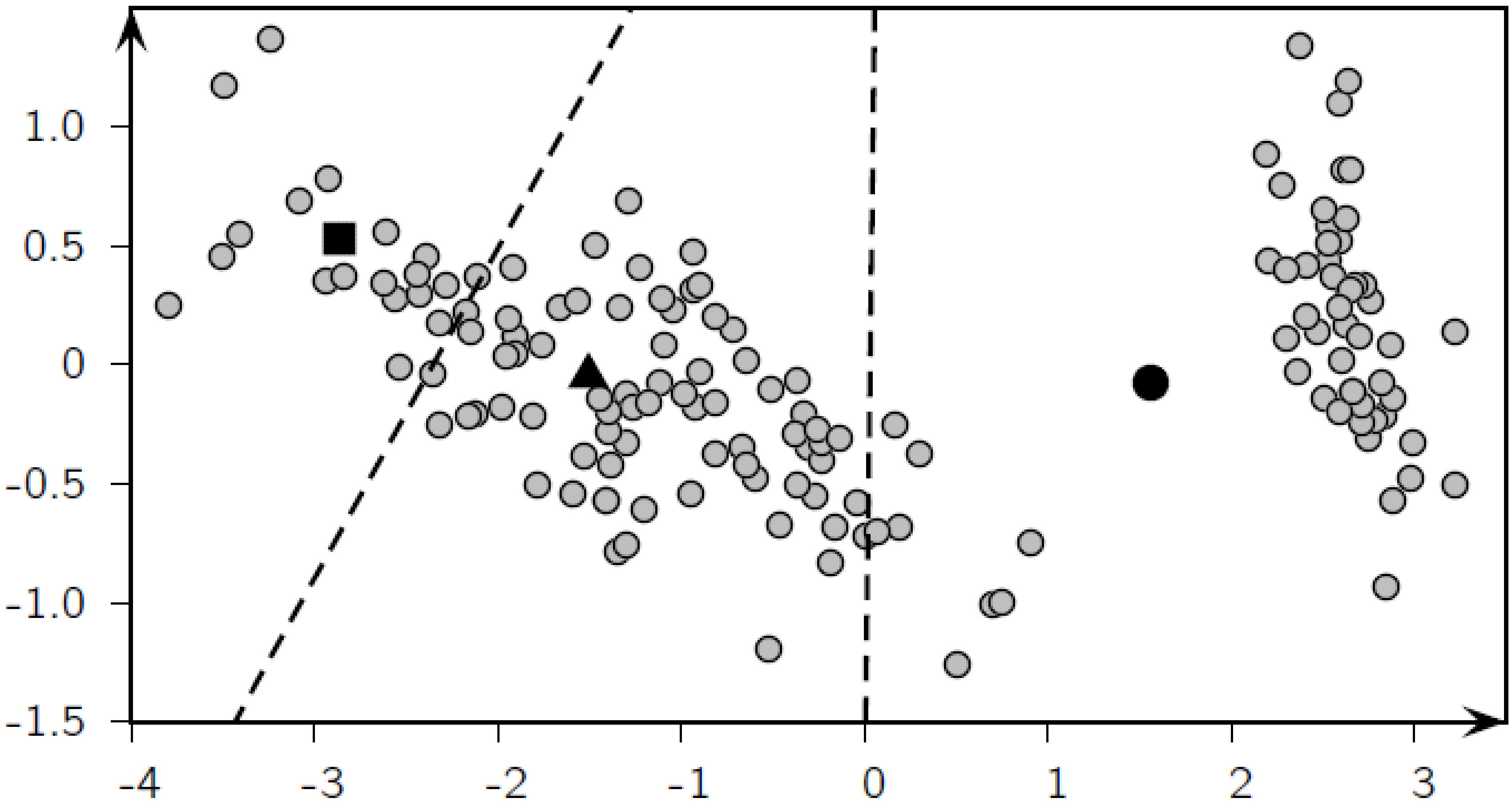
  foreach $i = 1$ *to* k **do**

    $\boldsymbol{\mu}_i^t = \frac{1}{|C_i|} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j$

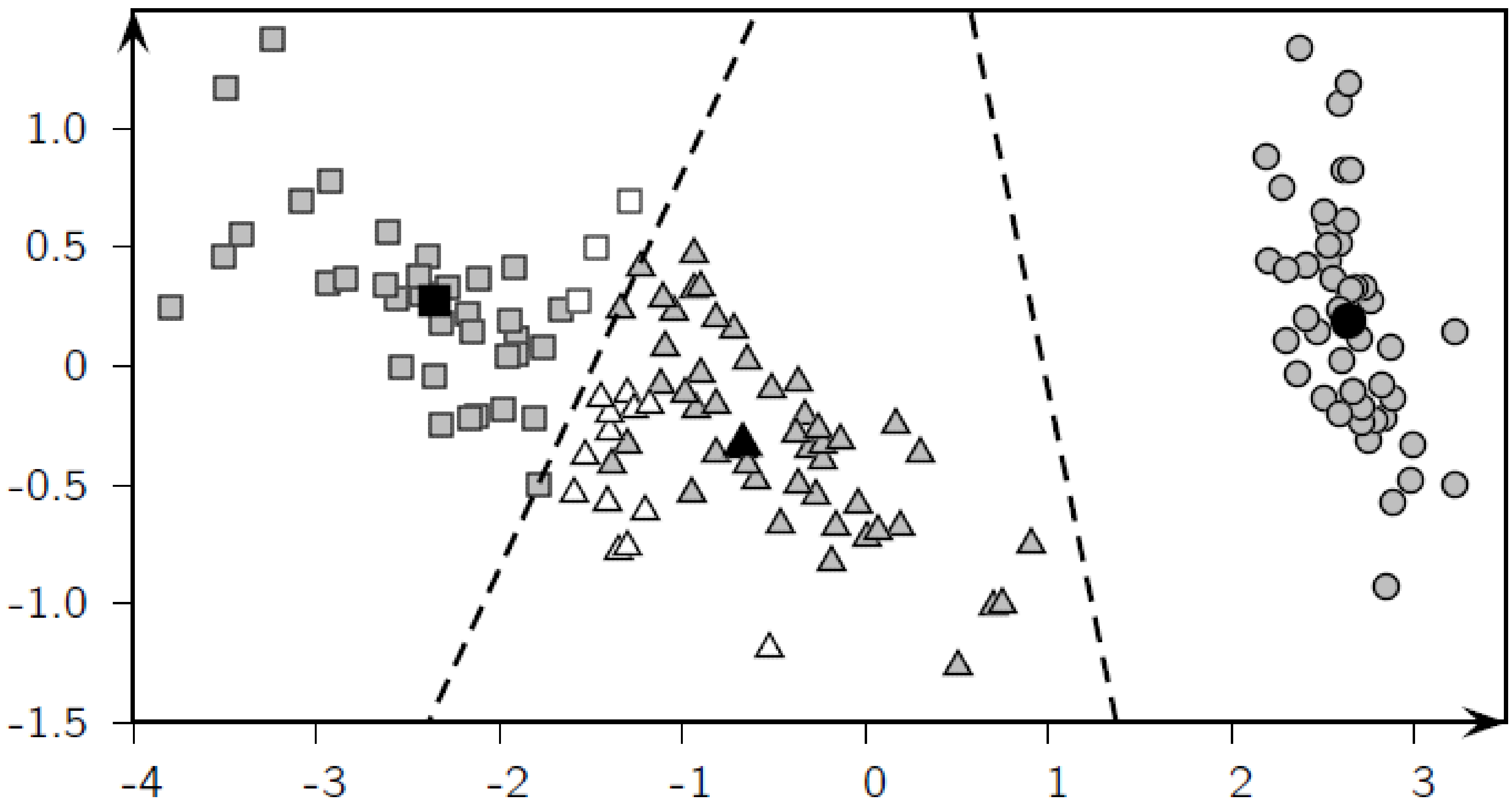
until $\sum_{i=1}^{k} \|\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1}\|^2 \leq \epsilon$



(a) Random Initialization: $t = 0$



(b) Iteration: $t = 1$



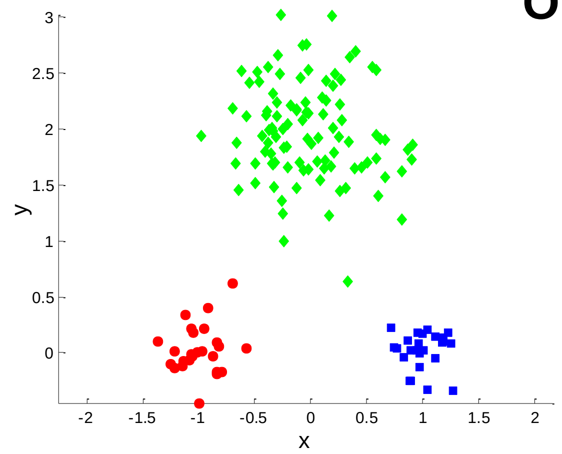
(c) Iteration: $t = 8$ (converged)

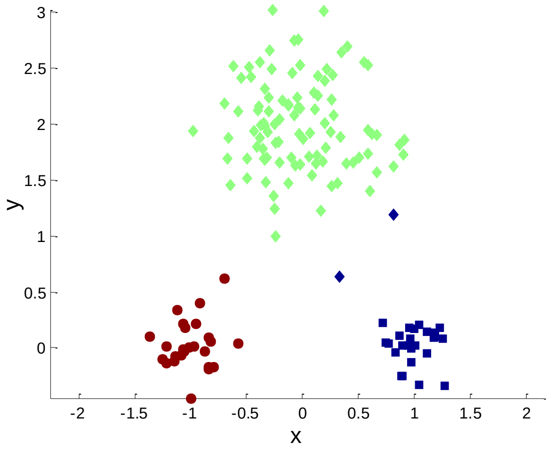
Issues and Limitations for K-means

- How to choose initial centers?
- How to choose K?
- How to handle Outliers?
- Clusters different in
  - Shape
  - Density
  - Size

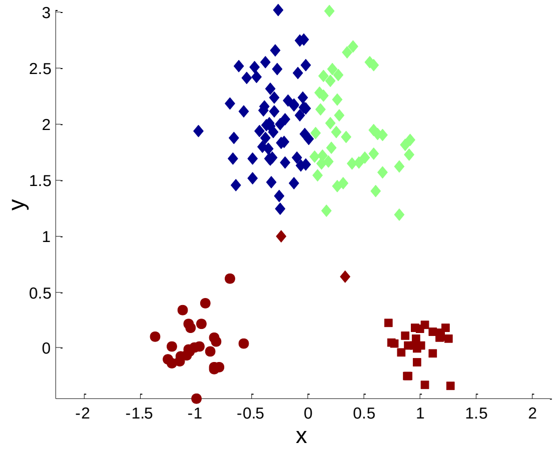
Two different K-means Clusterings

Original Points



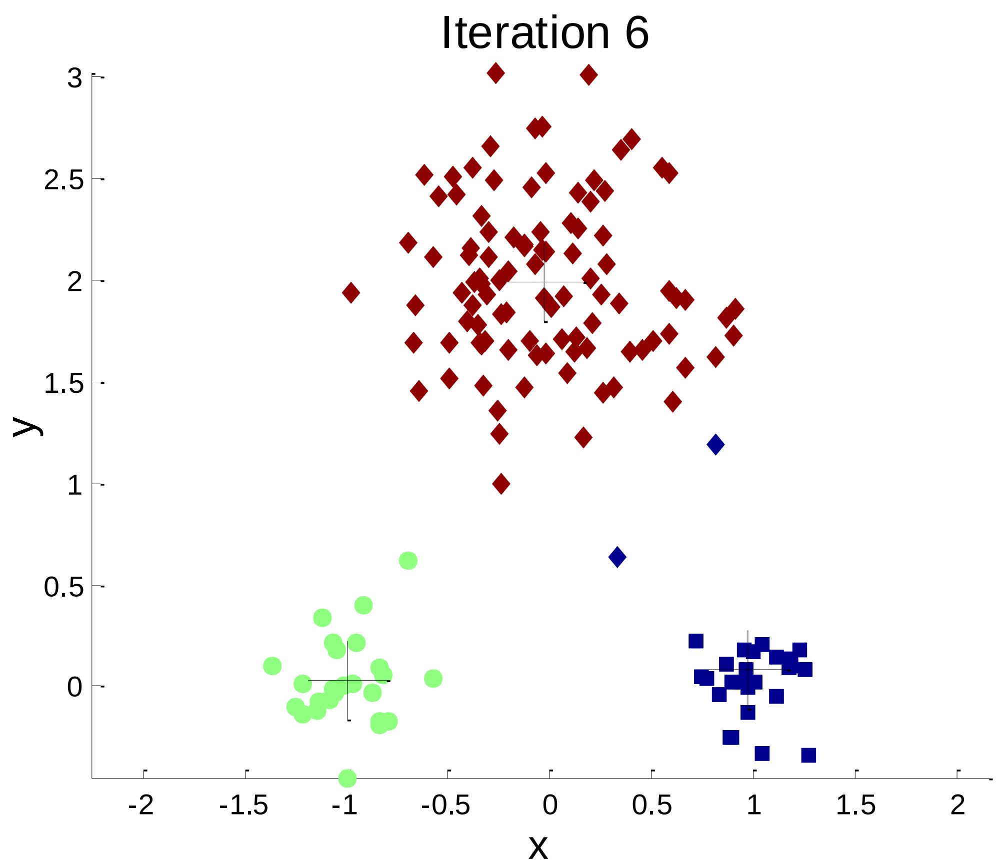


Optimal Clustering

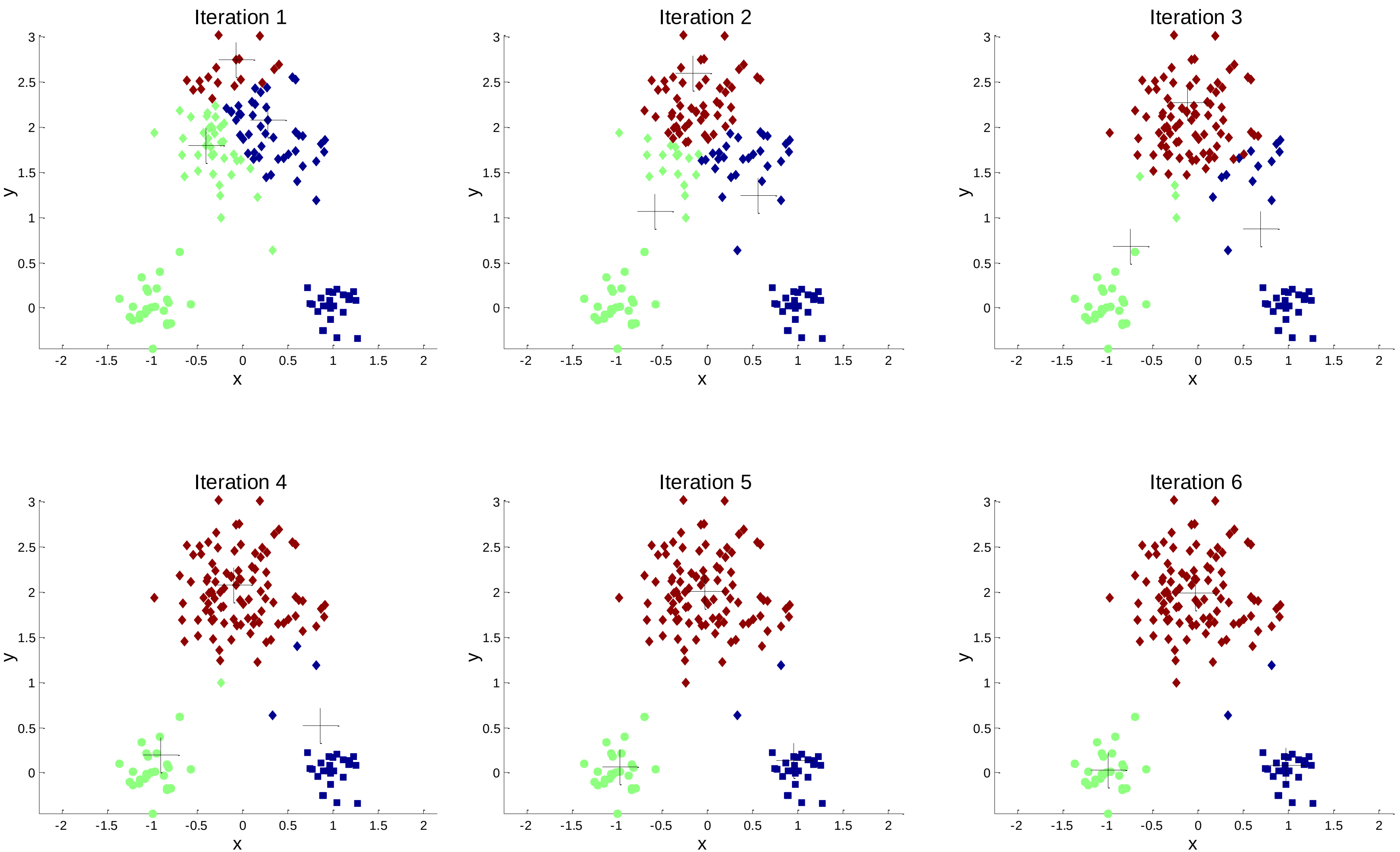


Sub-optimal Clustering

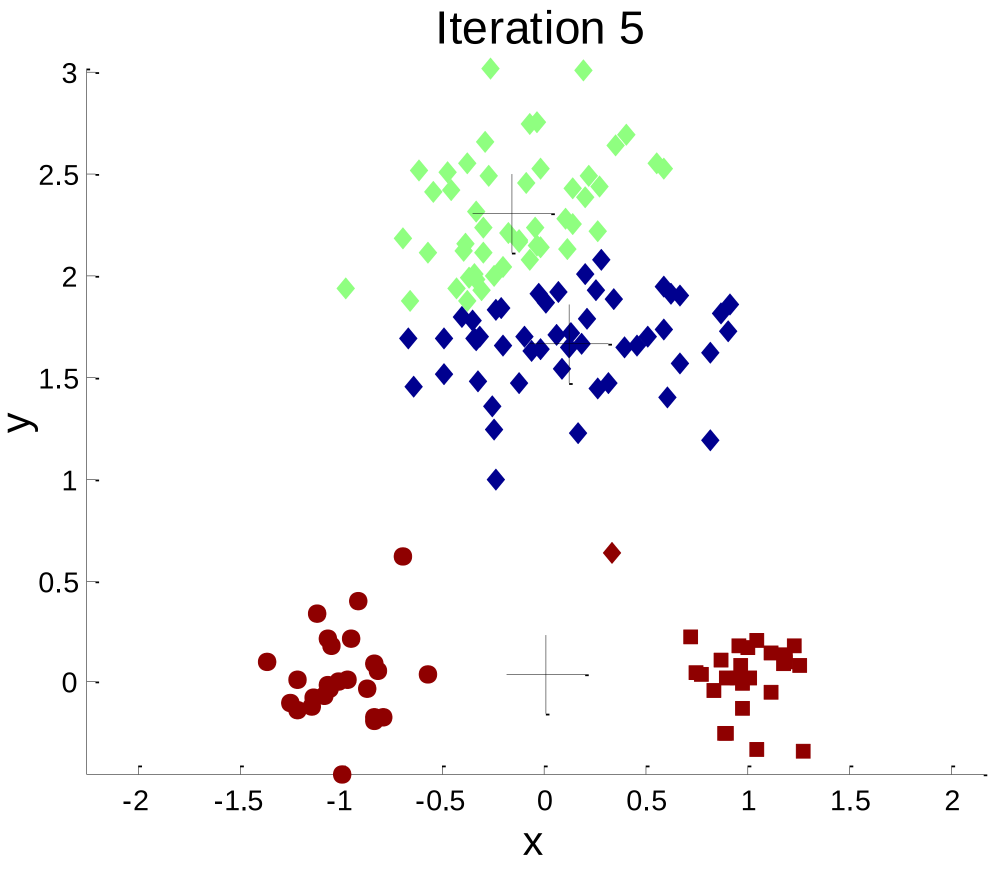
Importance of Choosing Initial Centroids



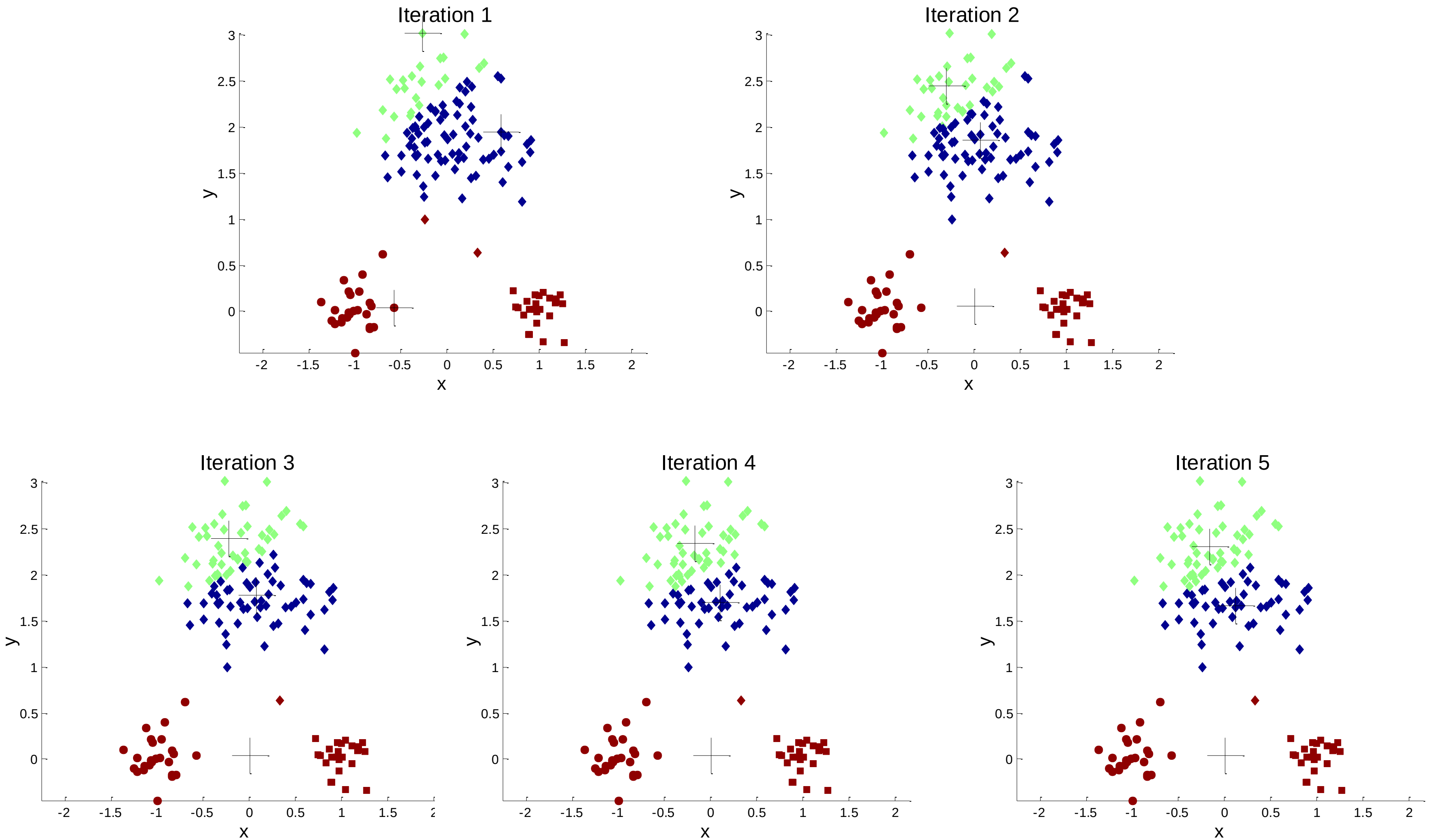
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



Problems with Selecting Initial Points

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

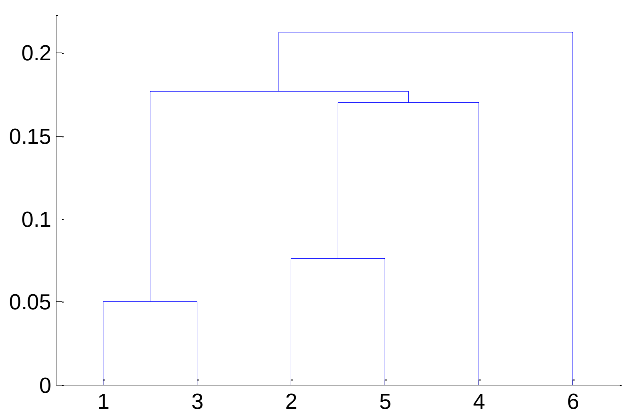
  - For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

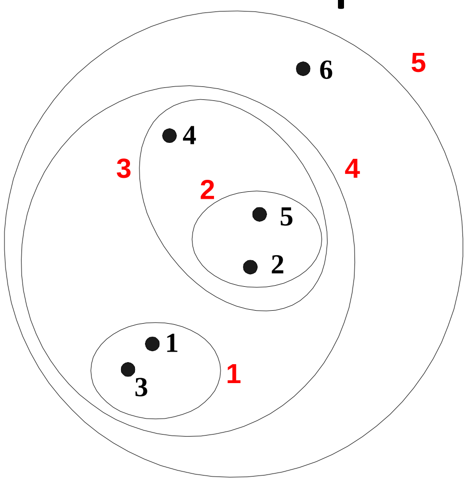
Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- Postprocessing
- Bisecting K-means
  - Not as susceptible to initialization issues

Hierarchical Clustering

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





Strengths of Hierarchical Clustering

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

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

Hierarchical Clustering

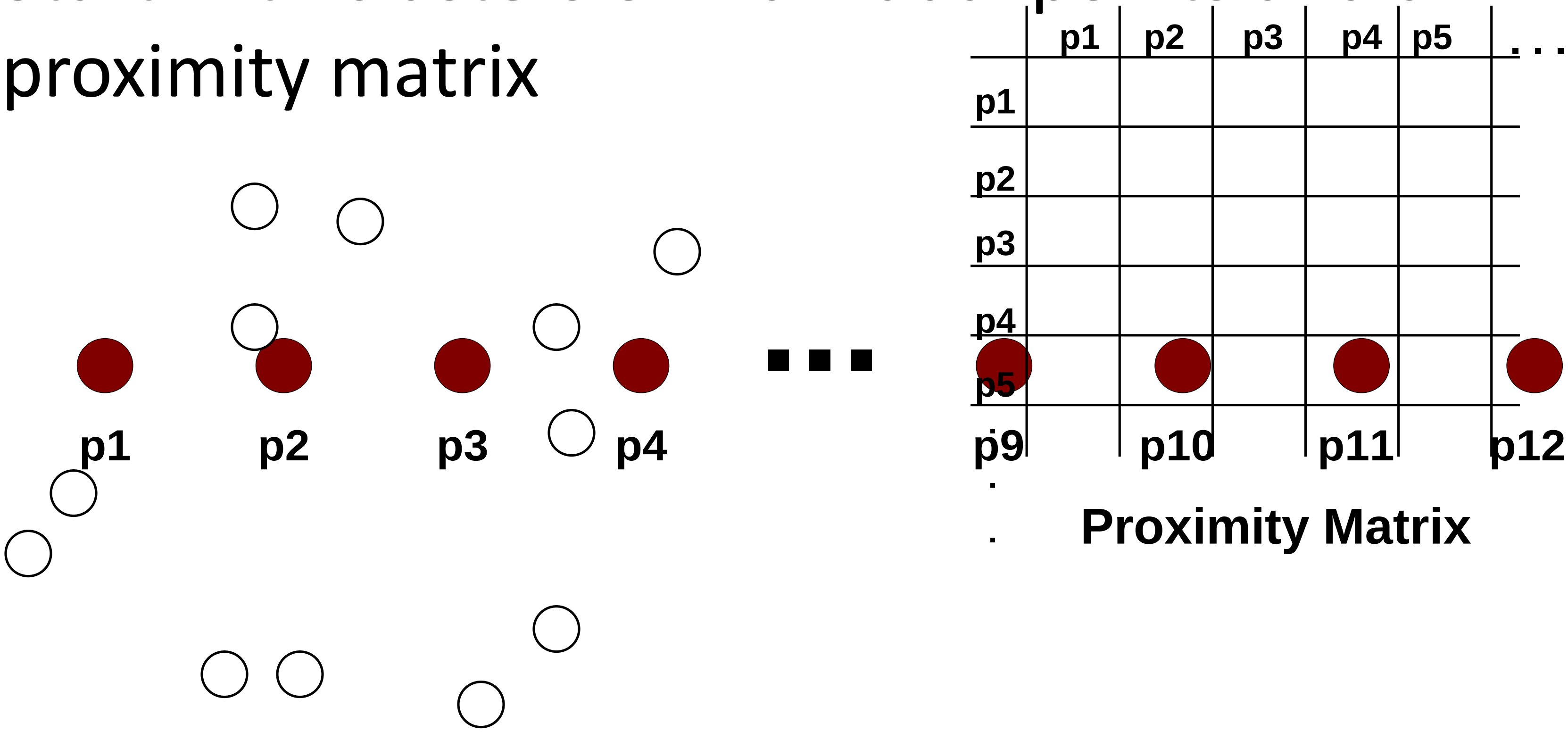
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

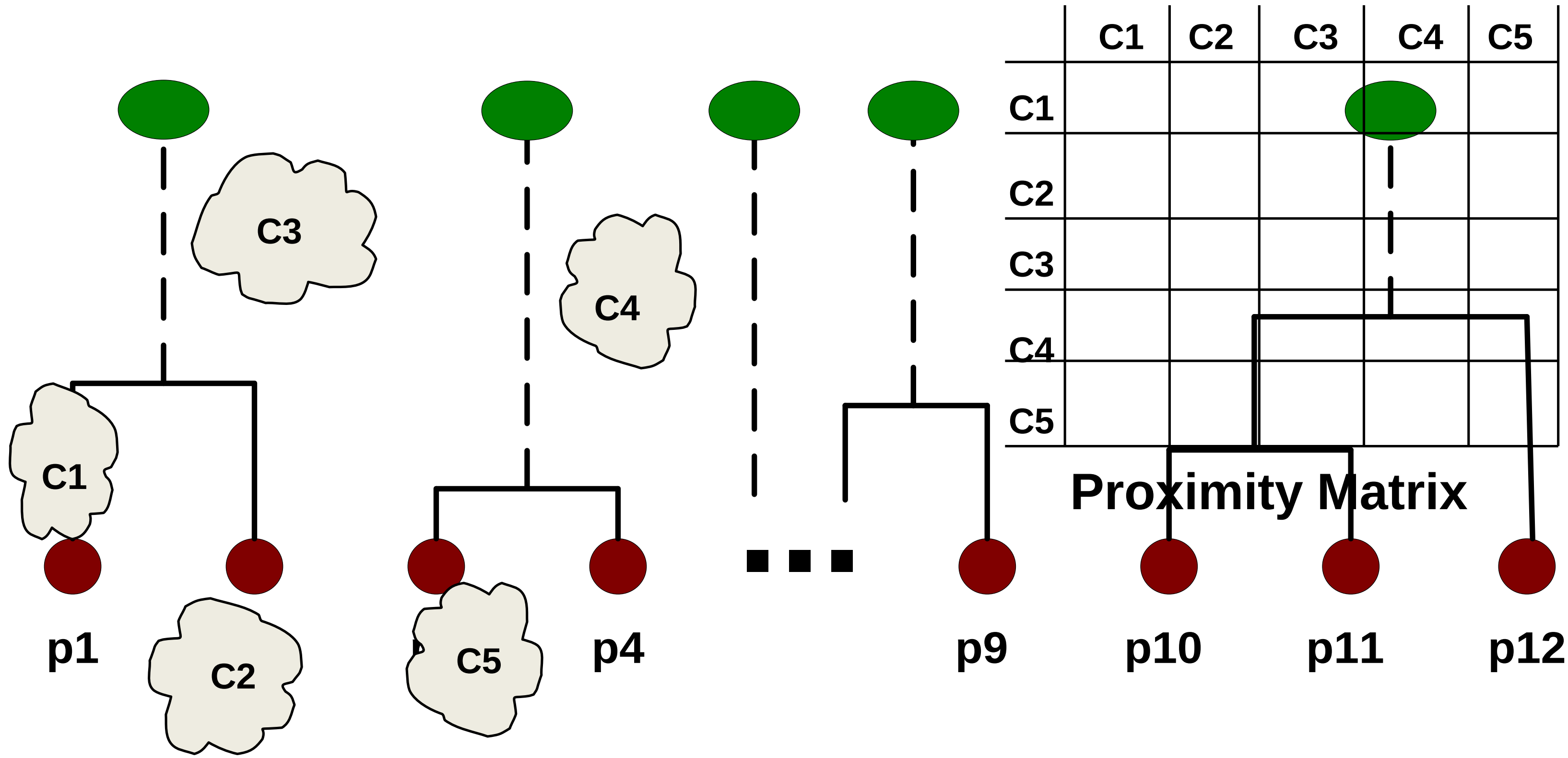
Starting Situation

- Start with clusters of individual points and a proximity matrix



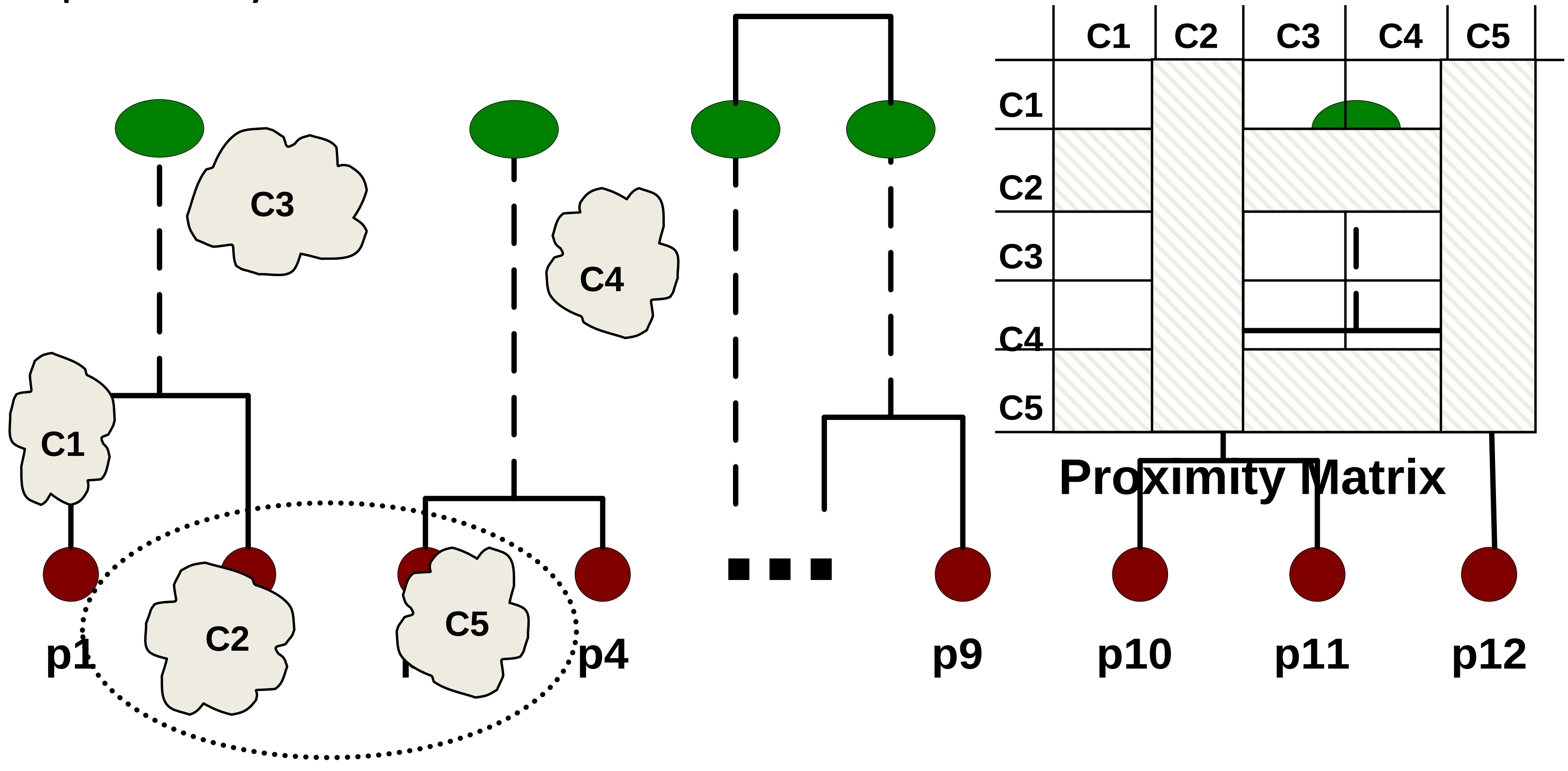
Intermediate Situation

- After some merging steps, we have some clusters



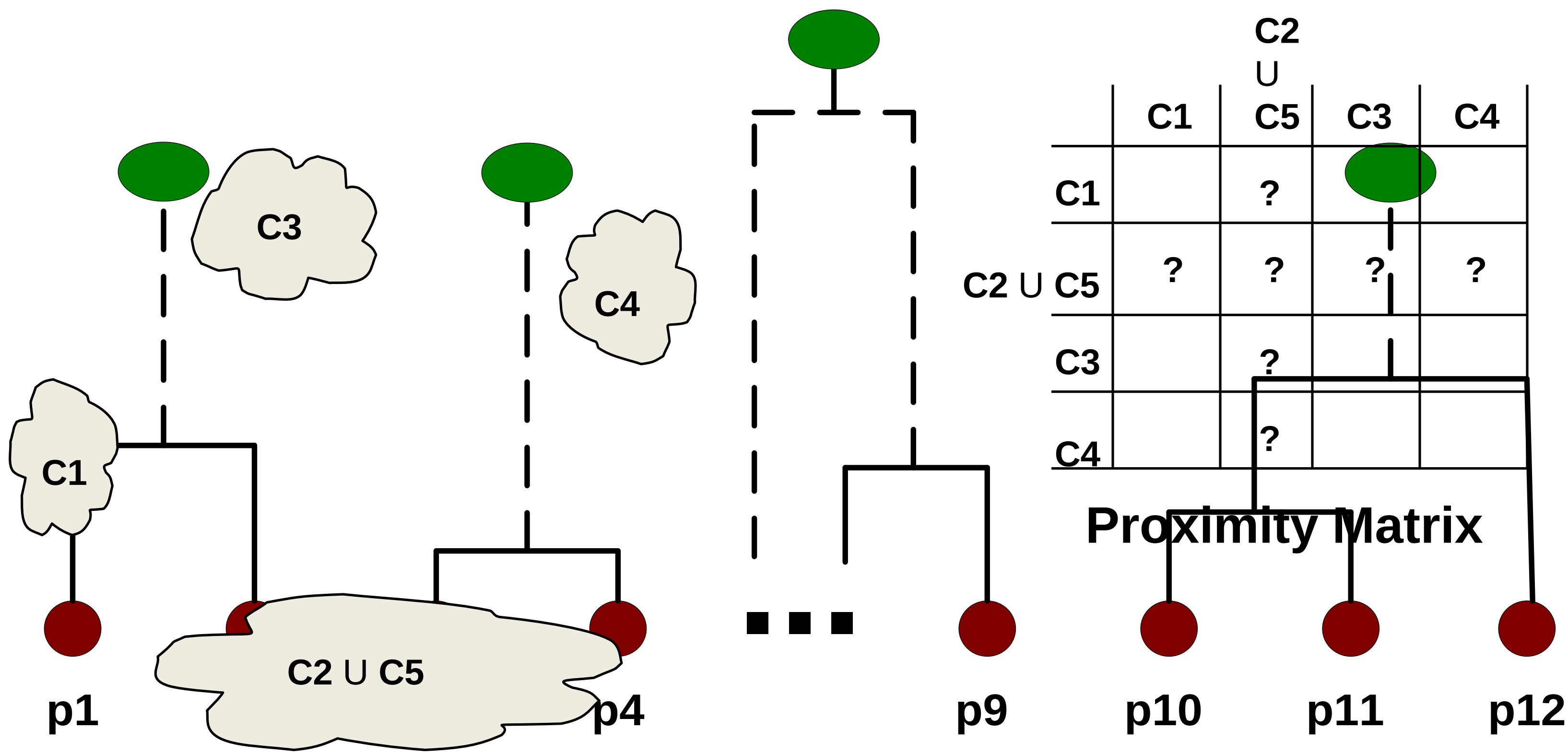
Intermediate Situation

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

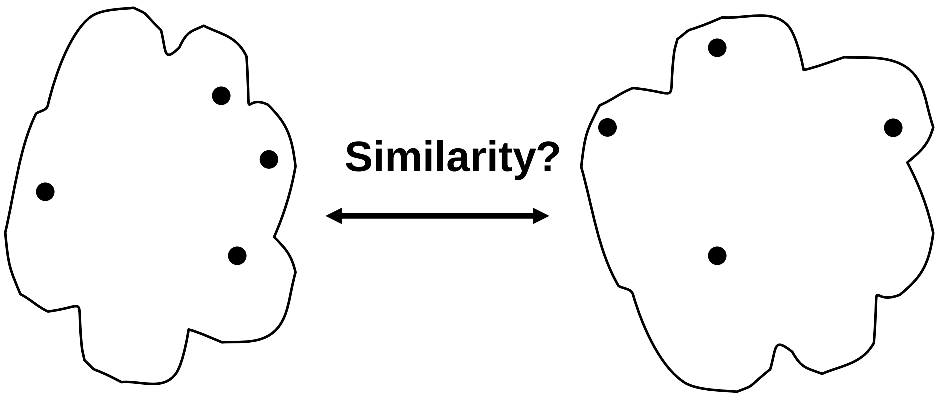


After Merging

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



How to Define Inter-Cluster Similarity

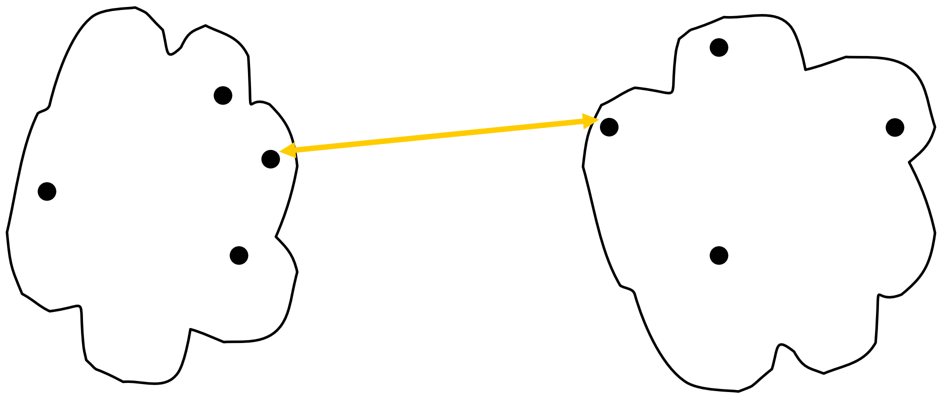


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

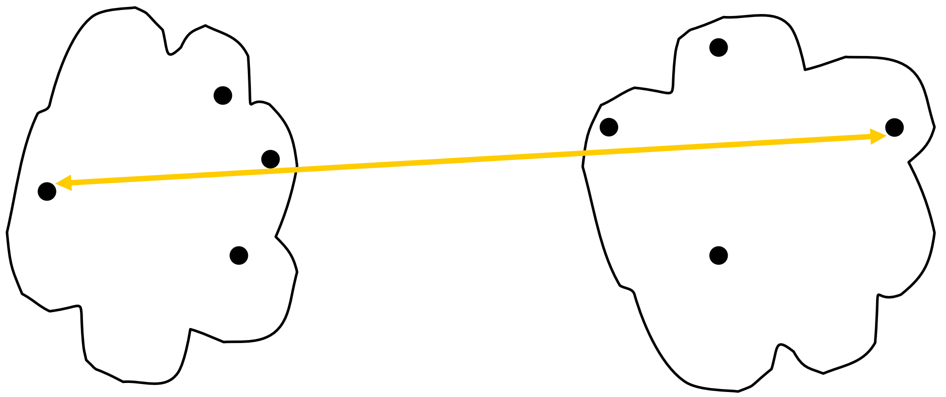


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

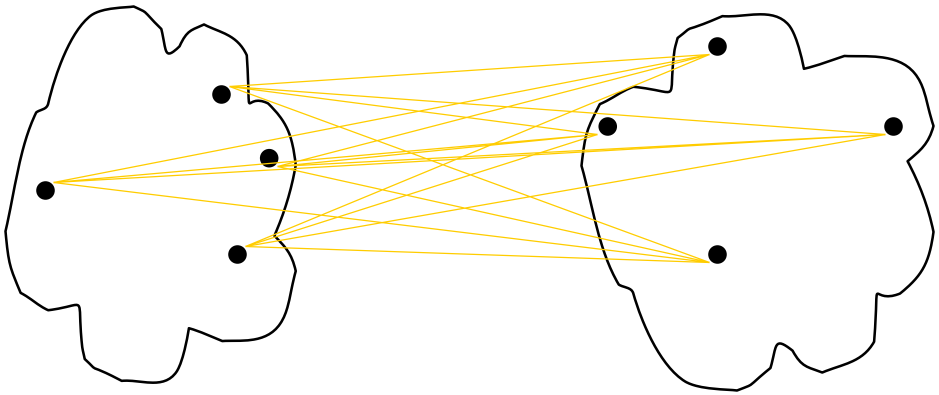


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity

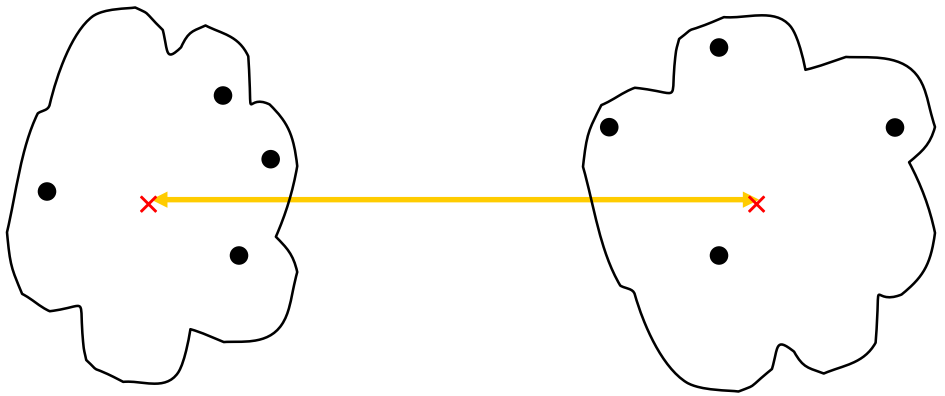


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

How to Define Inter-Cluster Similarity



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

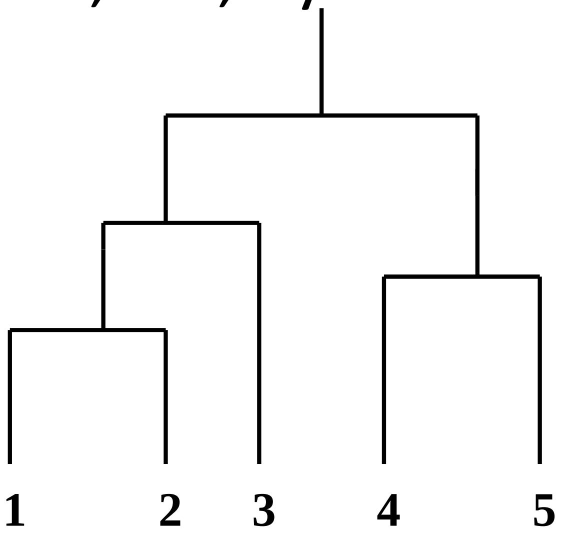
Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

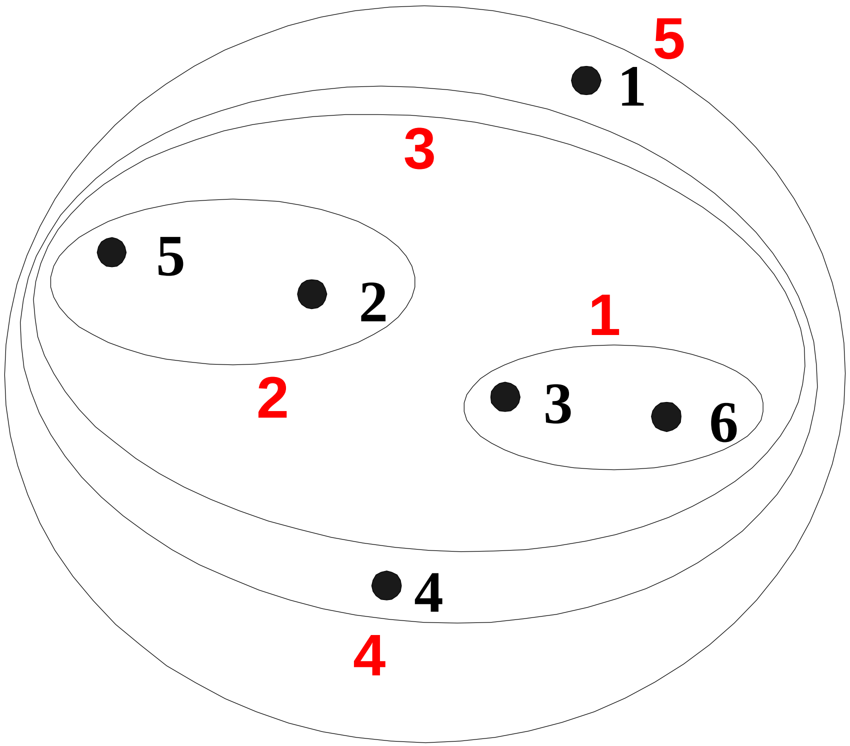
Cluster Similarity: MIN or Single Link

- 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.

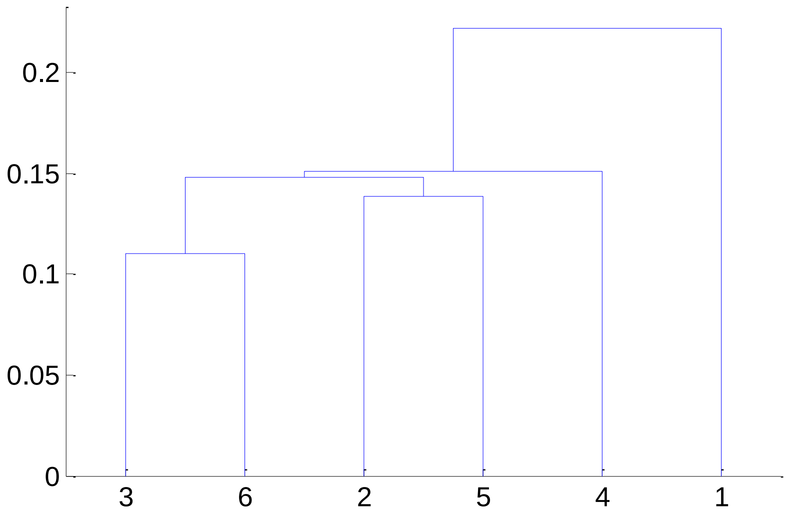
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN

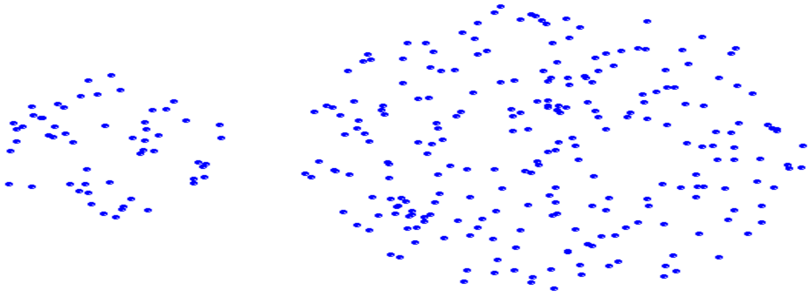


Nested Clusters

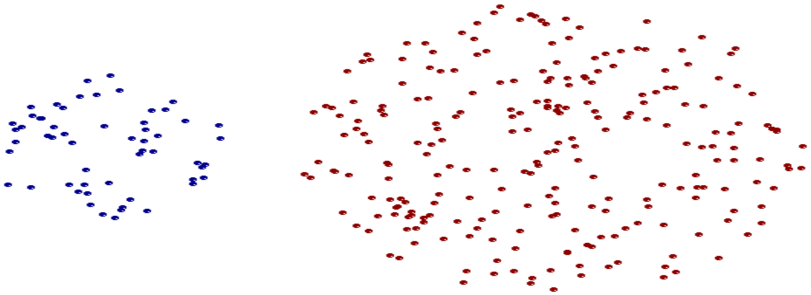


Dendrogram

Strength of MIN



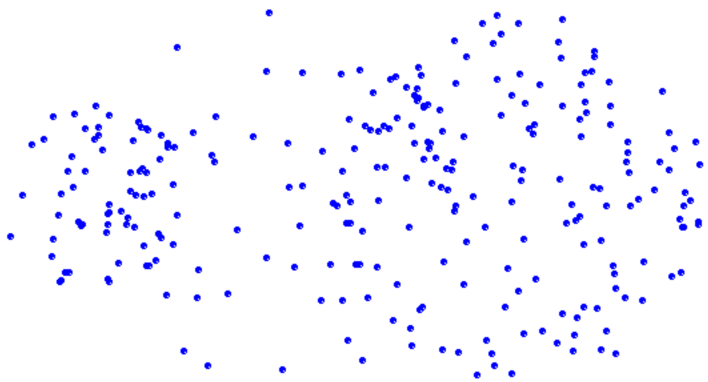
Original Points



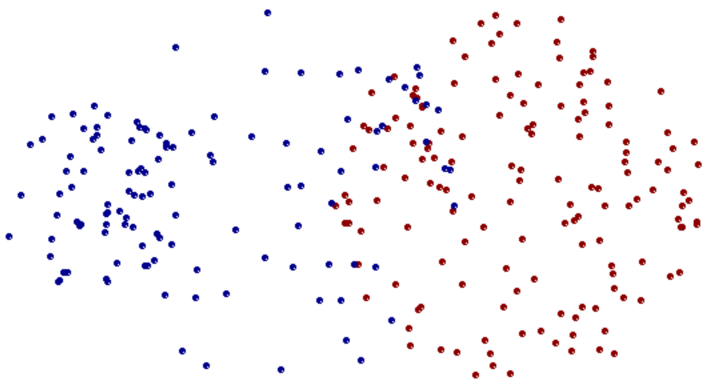
Two Clusters

- **Can handle non-elliptical shapes**

Limitations of MIN



Original Points



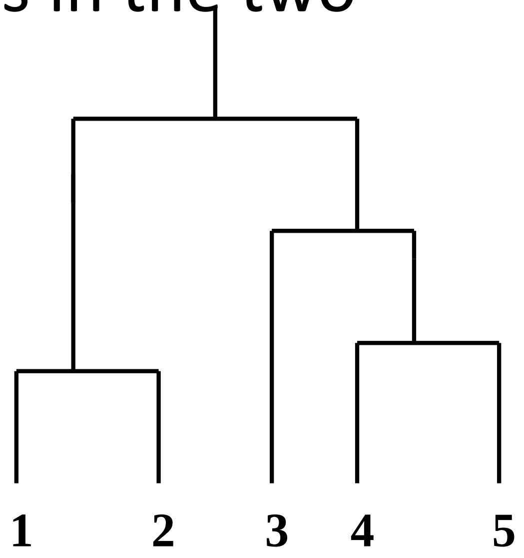
Two Clusters

- **Sensitive to noise and outliers**

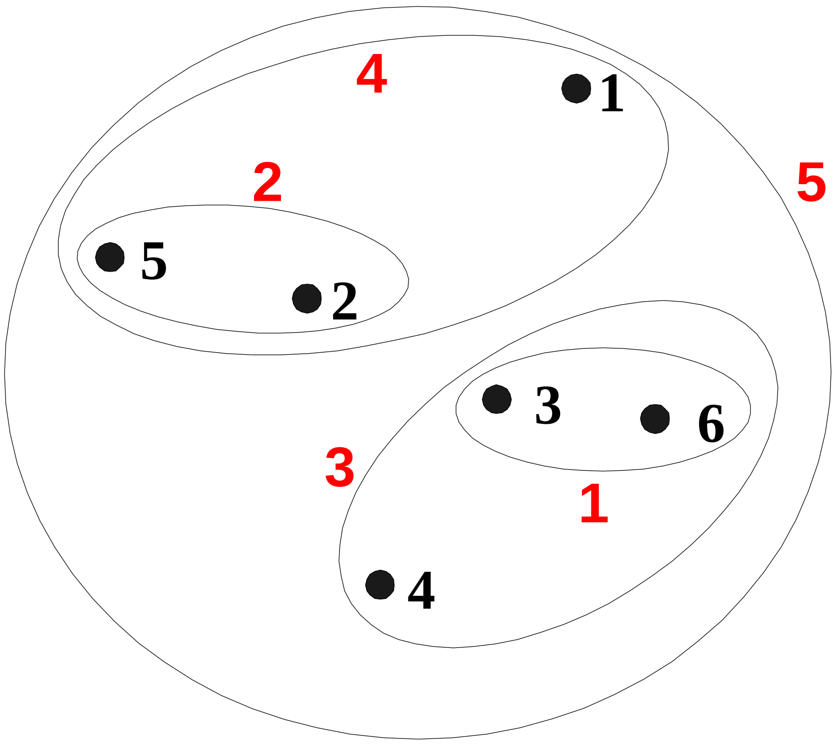
Cluster Similarity: MAX or Complete Linkage

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

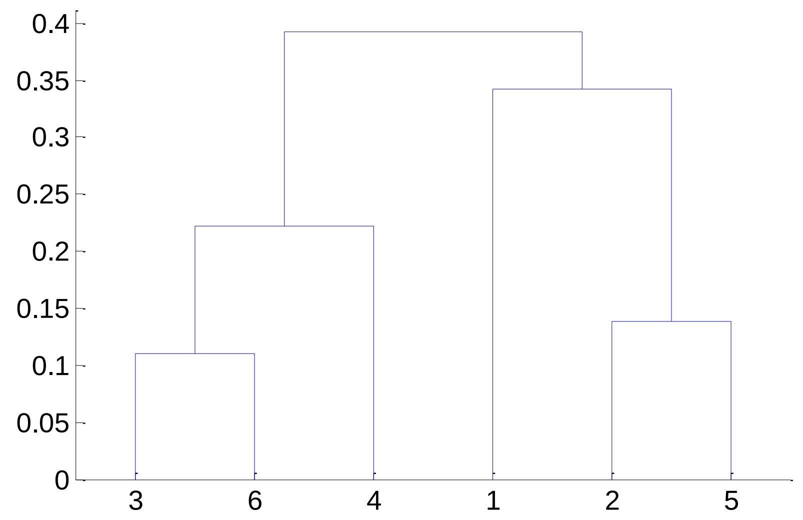
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

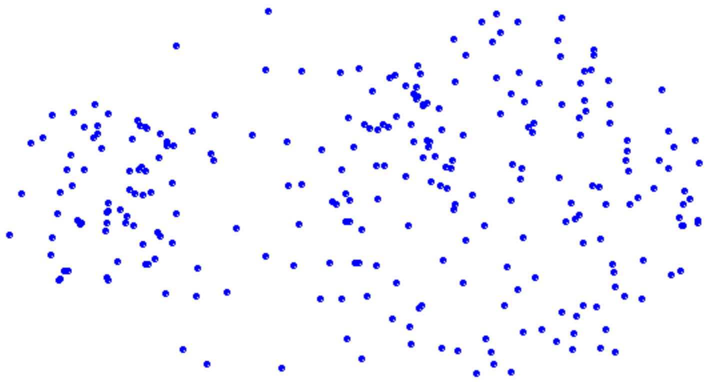


Nested Clusters

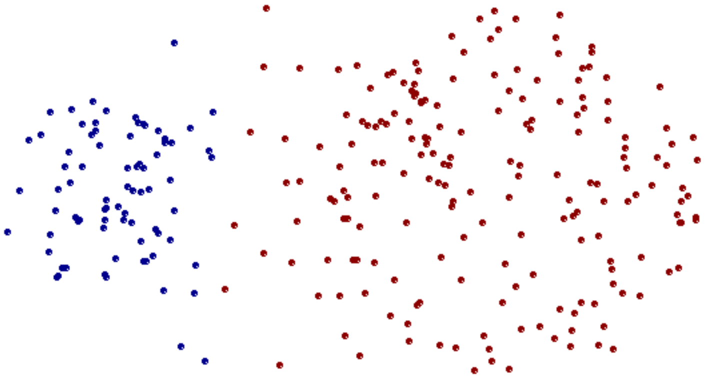


Dendrogram

Strength of MAX



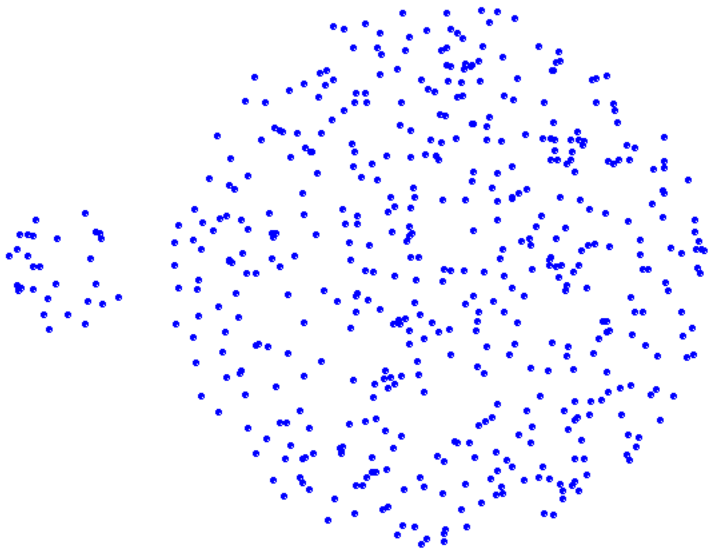
Original Points



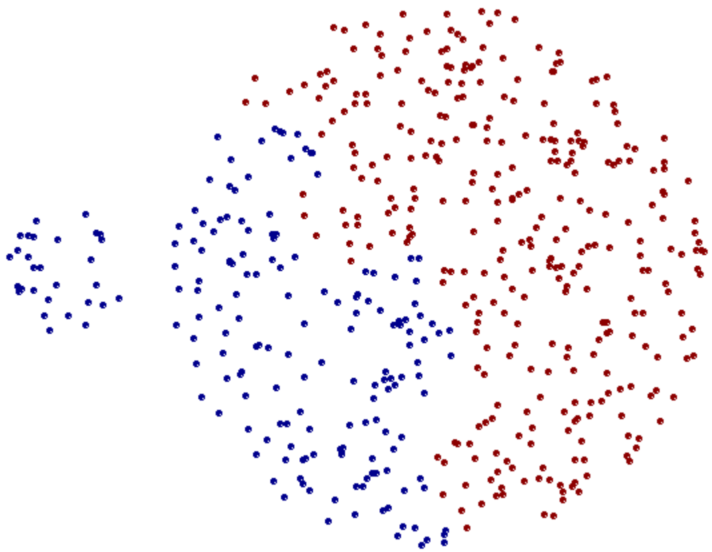
Two Clusters

- **Less susceptible to noise and outliers**

Limitations of MAX



Original Points



Two Clusters

•**Tends to break large clusters**

•**Biased towards globular clusters**

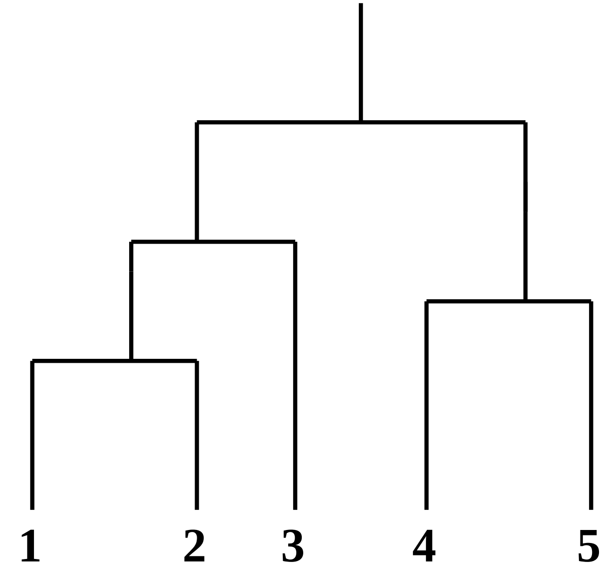
Cluster Similarity: Group Average

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

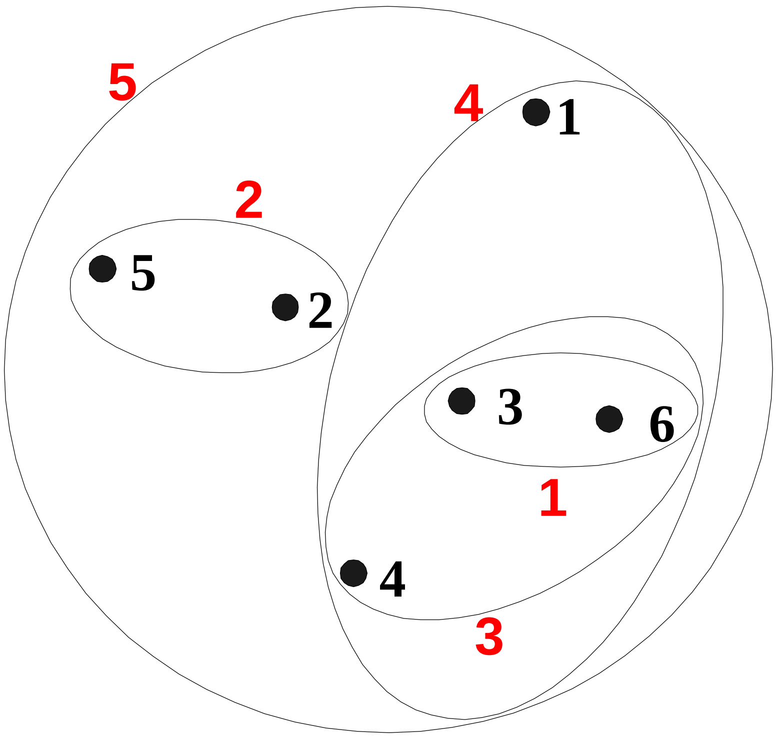
$$\mathbf{proximity}(\mathbf{Cluster_i}, \mathbf{Cluster_j}) = \frac{\sum\limits_{\substack{p_i \in \mathrm{Cluster_i} \\ p_j \in \mathrm{Cluster_j}}} \mathbf{proximity}(\mathbf{p_i}, \mathbf{p_j})}{|\mathbf{Cluster_i}| * |\mathbf{Cluster_j}|}$$

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

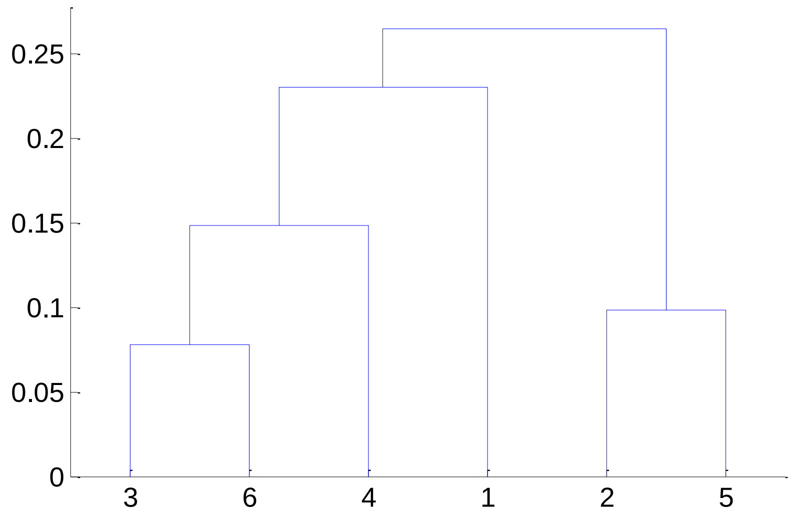
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link

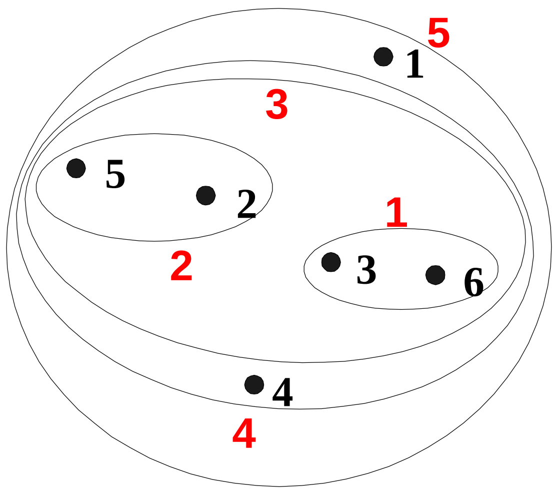
- Strengths
  - Less susceptible to noise and outliers

- Limitations
  - Biased towards globular clusters

Cluster Similarity: Ward's Method

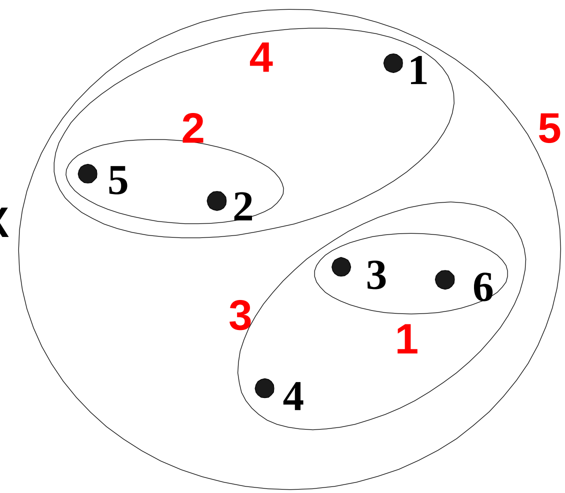
- Similarity of two clusters is based on the increase in squared error when two clusters are merged
  - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
  - Can be used to initialize K-means

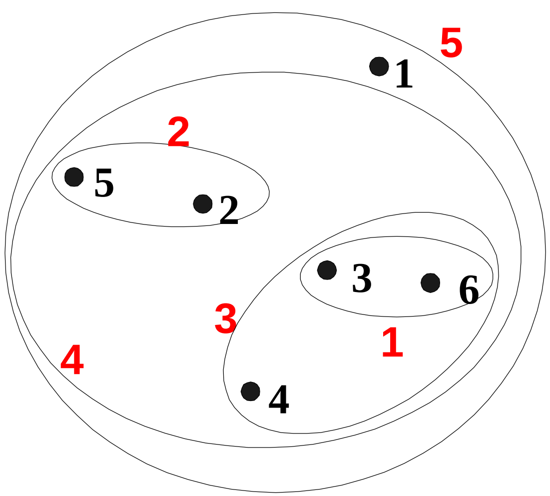
Hierarchical Clustering: Comparison



MIN

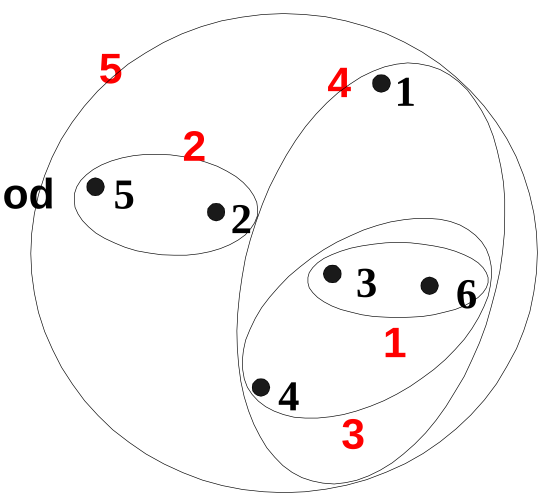
MAX





Group Average

Ward’s Method



Hierarchical Clustering: Time and Space requirements

- $O(N^2)$ 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

Hierarchical Clustering: Problems and Limitations

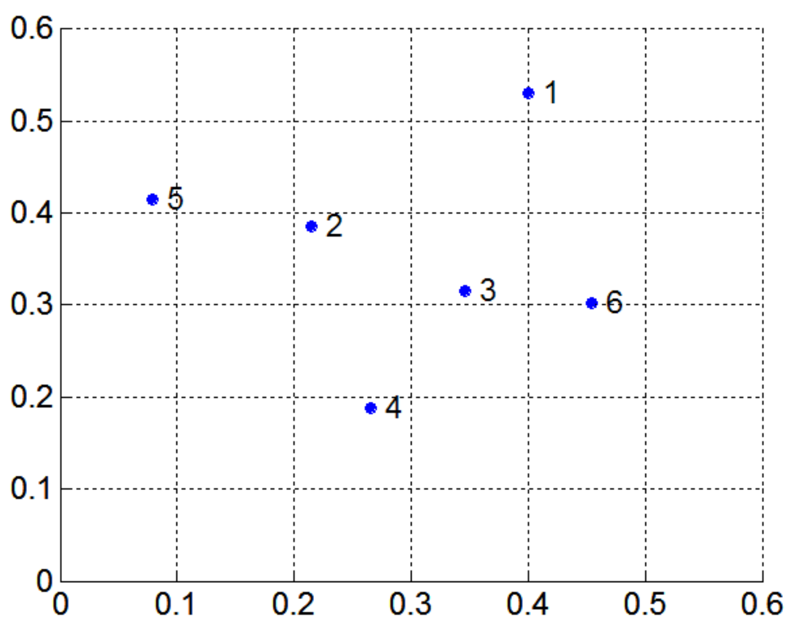
- Once a decision is made to combine two clusters, it cannot be undone

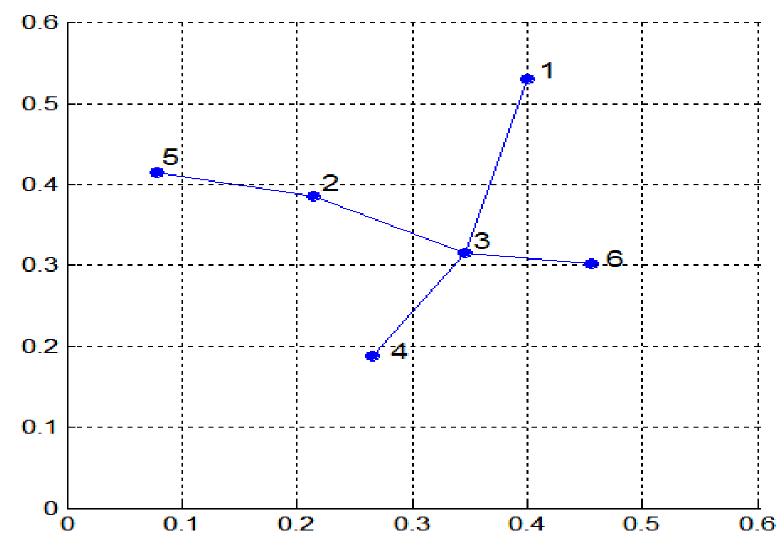
- No objective function is directly minimized

- 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

MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)
  - Start with a tree that consists of any point
  - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
  - Add q to the tree and put an edge between p and q





MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

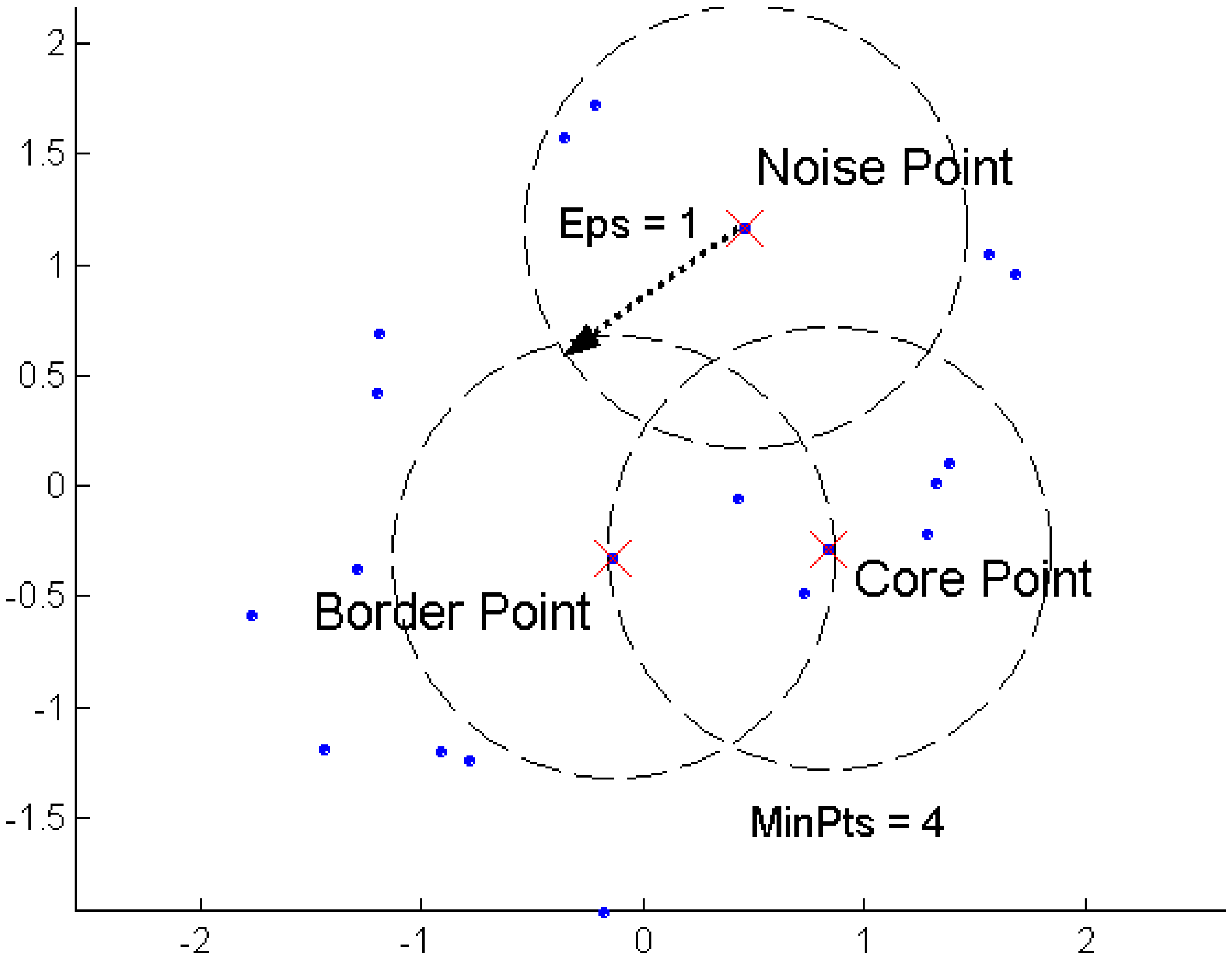
Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

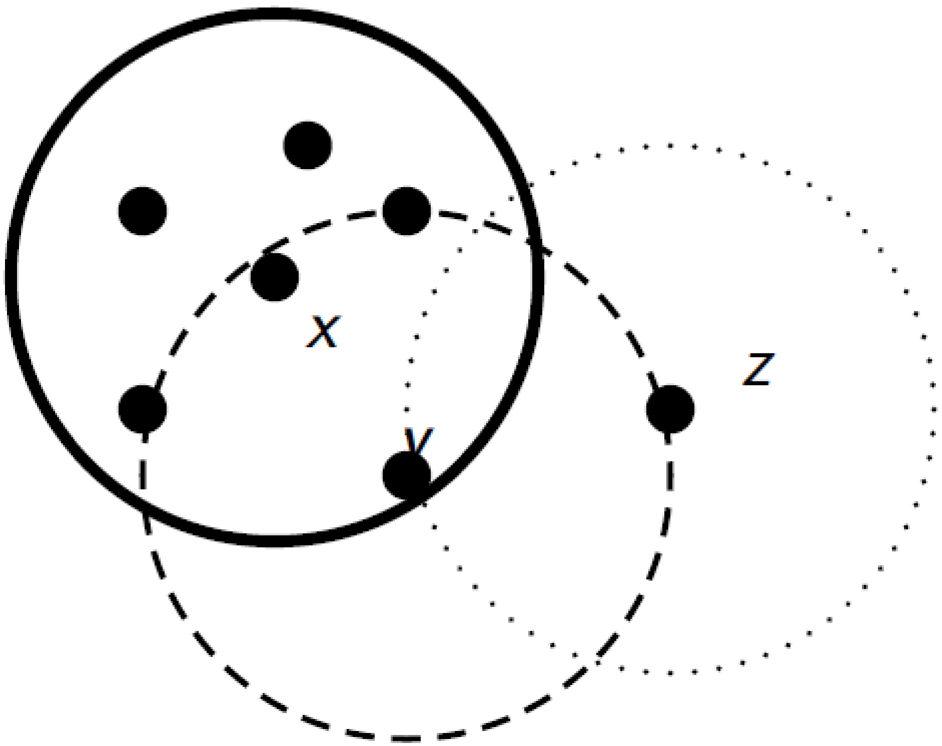
1: Compute a minimum spanning tree for the proximity graph.
2: **repeat**
3: Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).
4: **until** Only singleton clusters remain

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 are points that are at 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




x
y
z

Density Reachable

- (Directly) density reachable
  - A point x is directly density reachable from another point y, if $x \in N_\varepsilon(y)$ and y is a core point
  - A point x is density reachable from y, if there exists a chain of points, $x=x_0,x_1,x_2,\ldots x_l=y$, such that x_i is directly density reachable from x_{i-1}
- Density Connected
  - Two points x and y are density connected if there exists a core point z, such that both x and y are density reachable from z

Algorithm 18.1: Density-based Clustering Algorithm

dbscan $(\mathcal{D}, \epsilon, minpts)$:

foreach $\mathbf{x} \in \mathcal{D}$ **do**
  Compute $N_\epsilon(\mathbf{x})$
  Classify $\mathbf{x}$ as core, border, or noise

$id = 0$

foreach $\mathbf{x} \in \mathcal{D}$, *such that* $\mathbf{x}$ *is core and unmarked* **do**
  $id = id + 1$
  DensityConnected$(\mathbf{x}, id)$

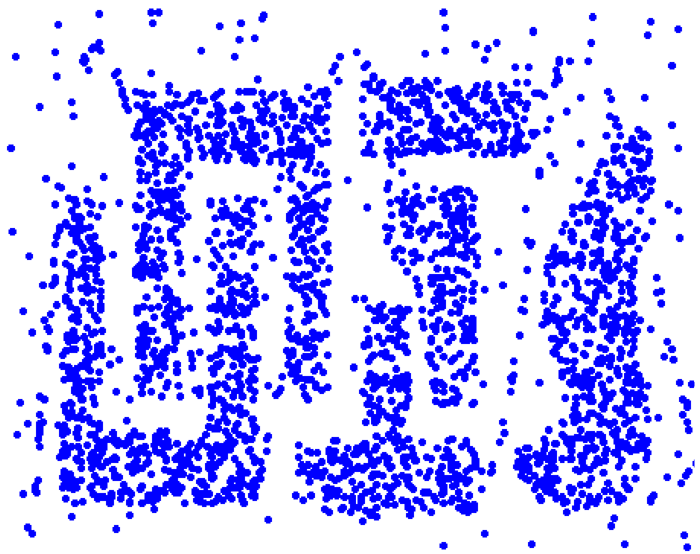
return *Clustering* $\{\mathcal{D}_i\}_{i=1}^{id}$, *where* $\mathcal{D}_i = \{\mathbf{x} \in \mathcal{D} : \mathbf{x}$ *has label* $i\}$

DensityConnected $(\mathbf{x}, id)$:

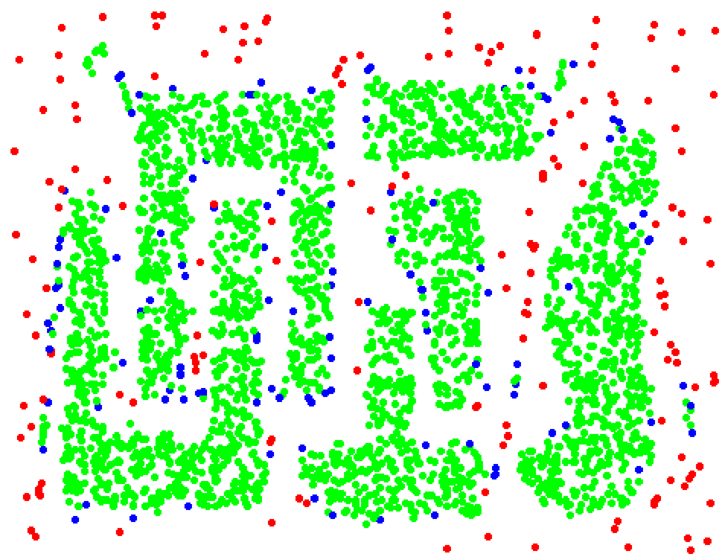
Mark $\mathbf{x}$ with current cluster id

foreach $\mathbf{y} \in N_\epsilon(\mathbf{x})$ **do**
  Mark $\mathbf{y}$ with current cluster id
  if $\mathbf{y}$ *is core* **then**
    DensityConnected$(\mathbf{y}, id)$

DBSCAN: Core, Border and Noise Points



Original Points

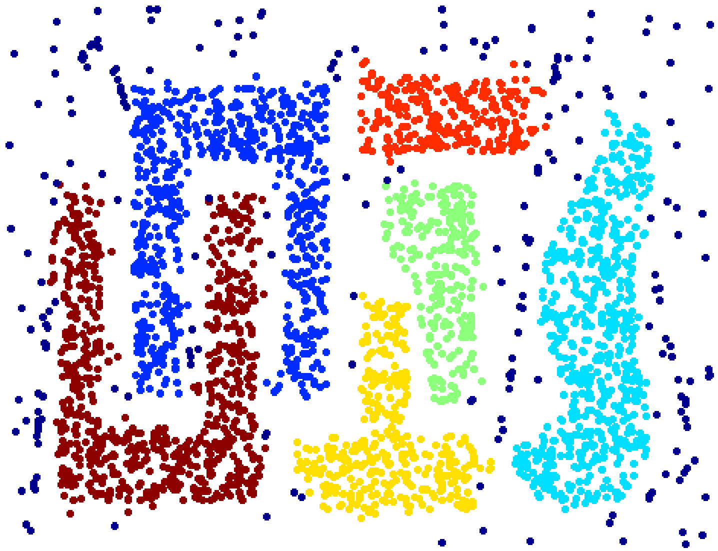


Point types: core, border and noise

Eps = 10, MinPts = 4

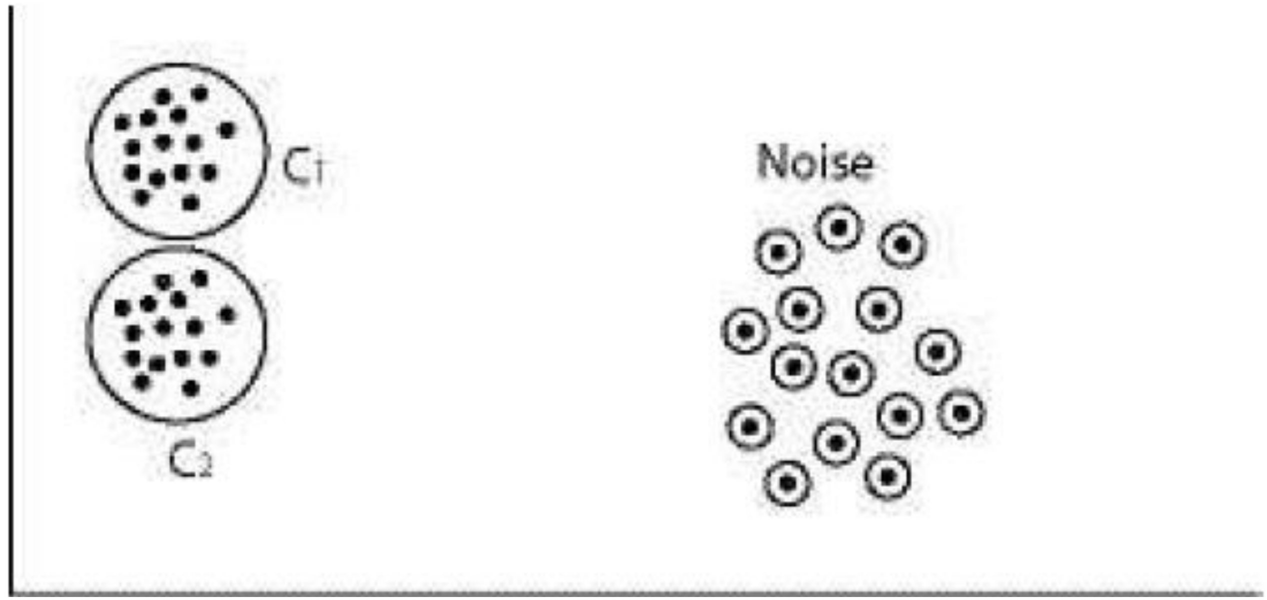
When DBSCAN Works Well

Original Points

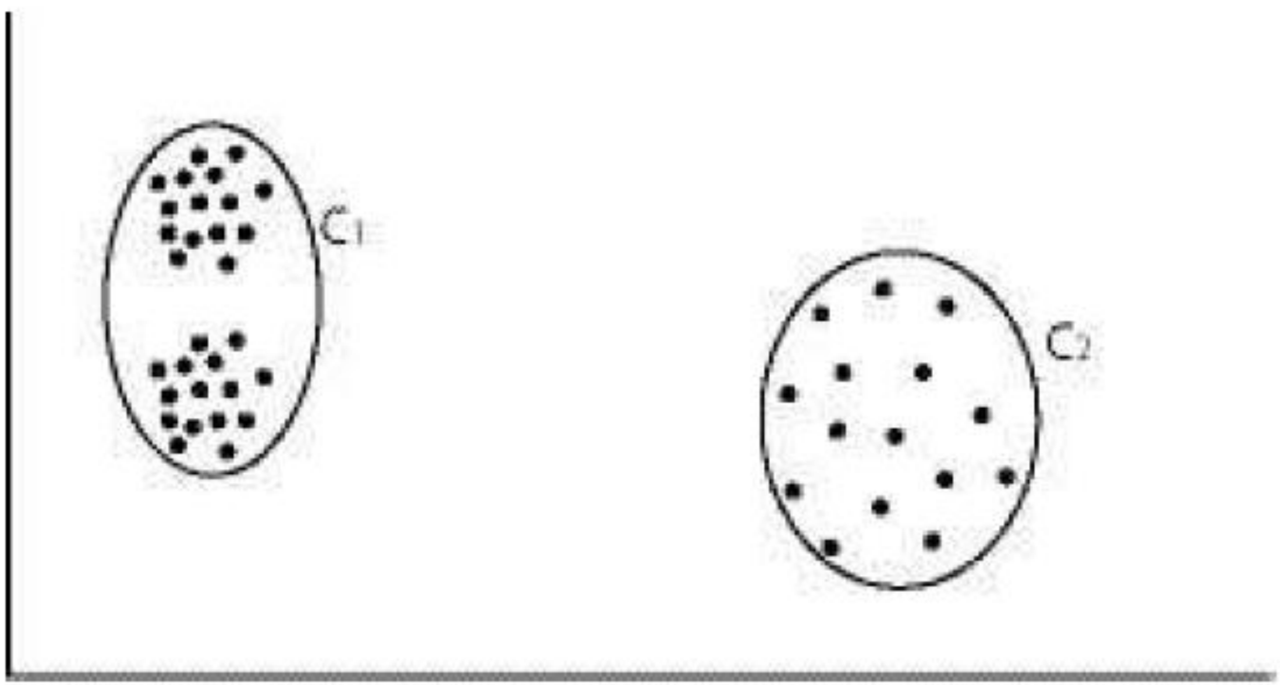


Clusters

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

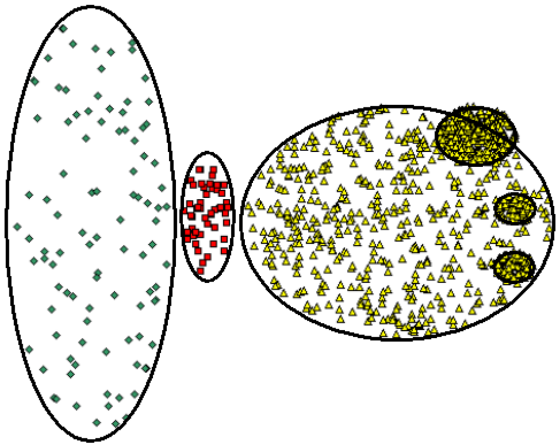


C1
C2
Noise

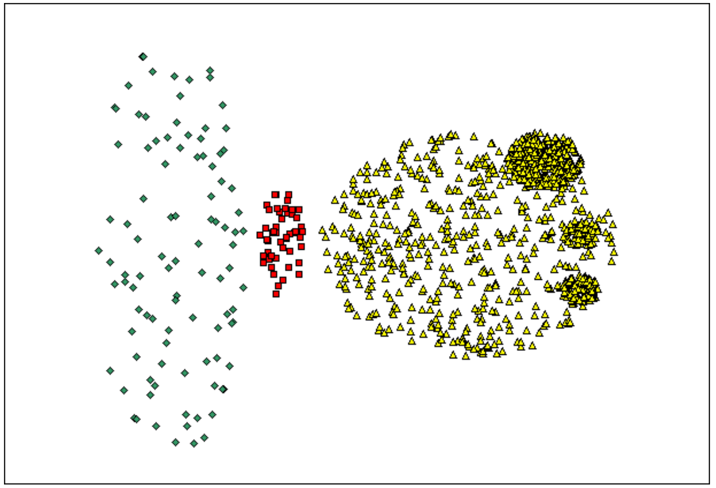


C1
C2

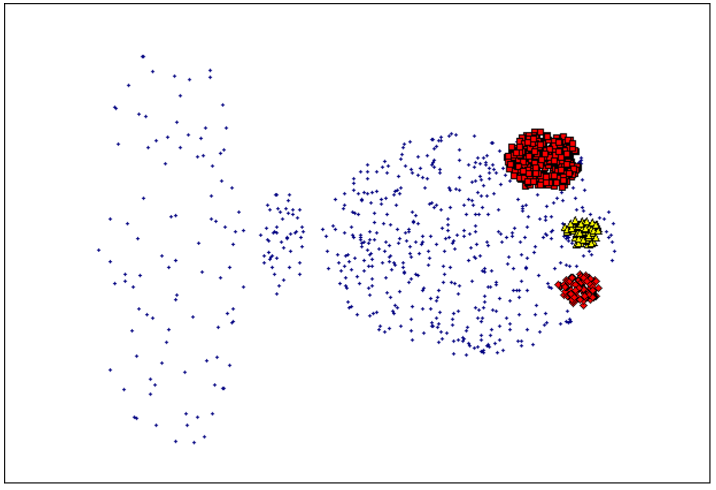
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.75).

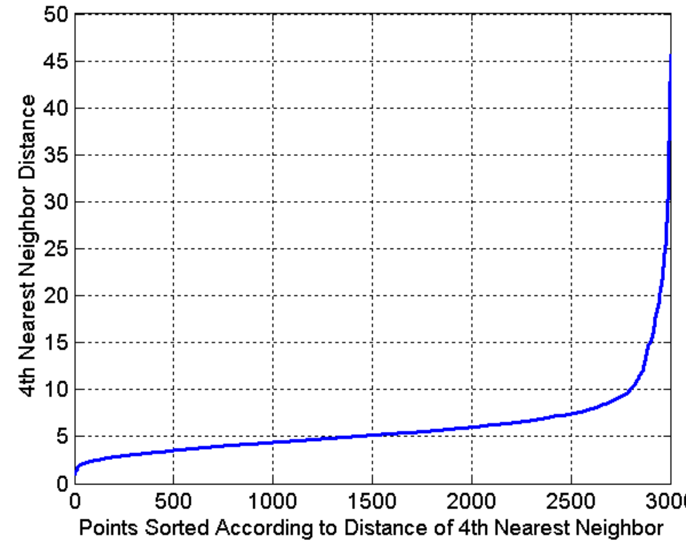


(MinPts=4, Eps=9.92)

- **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



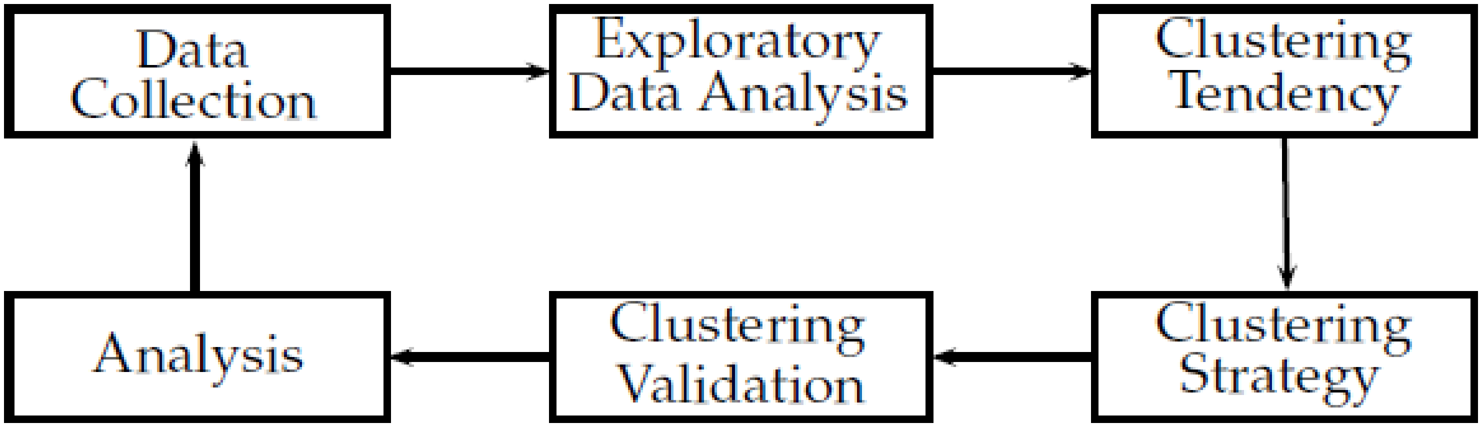
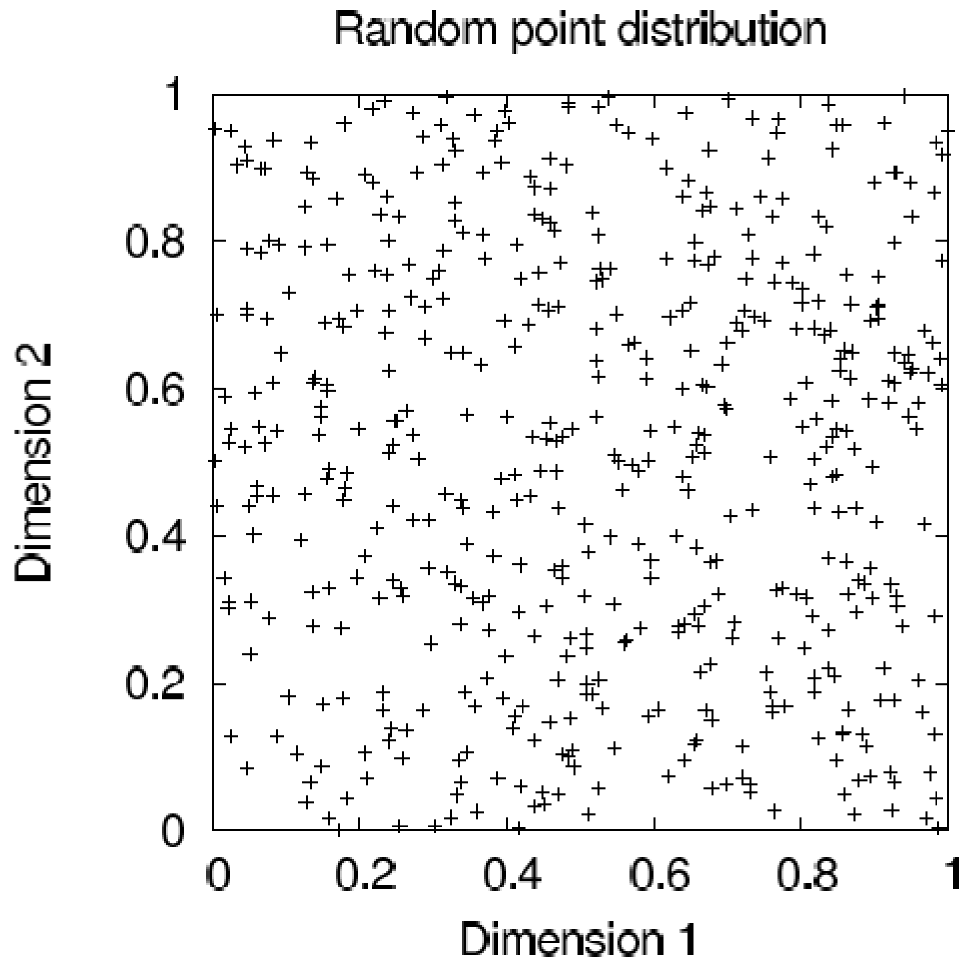
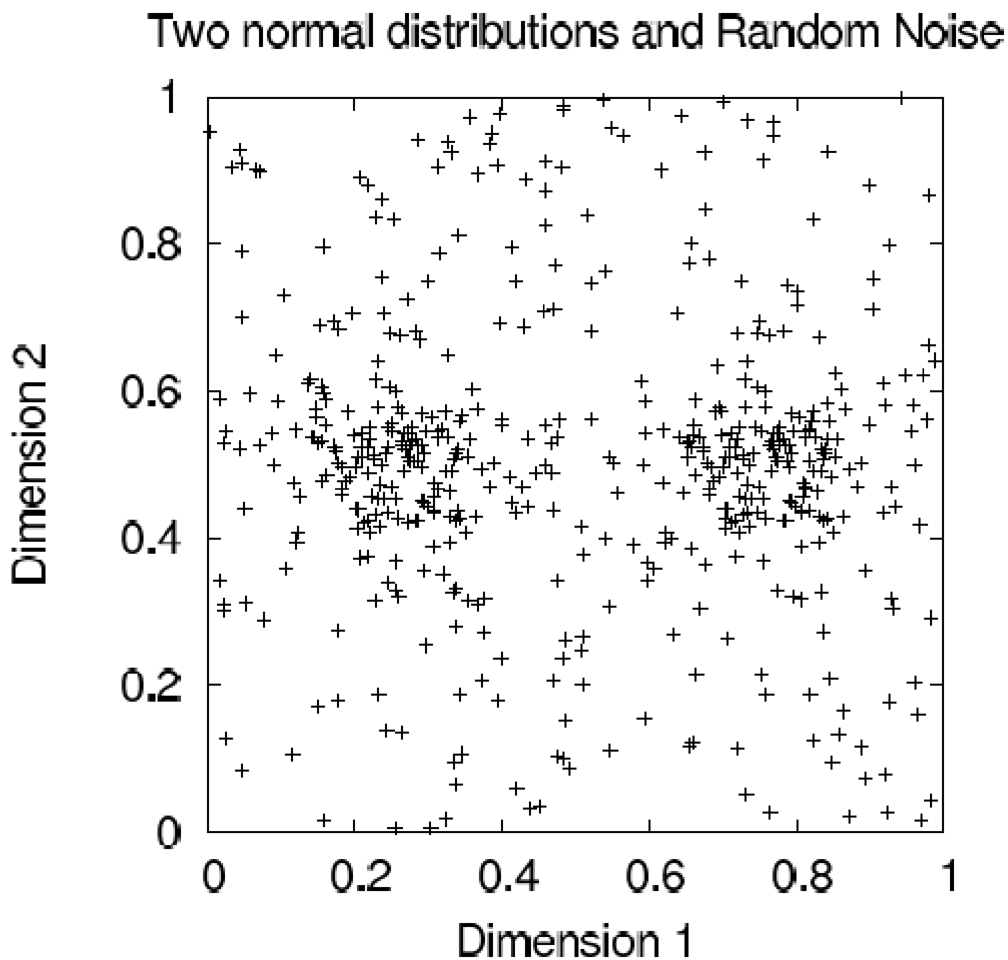
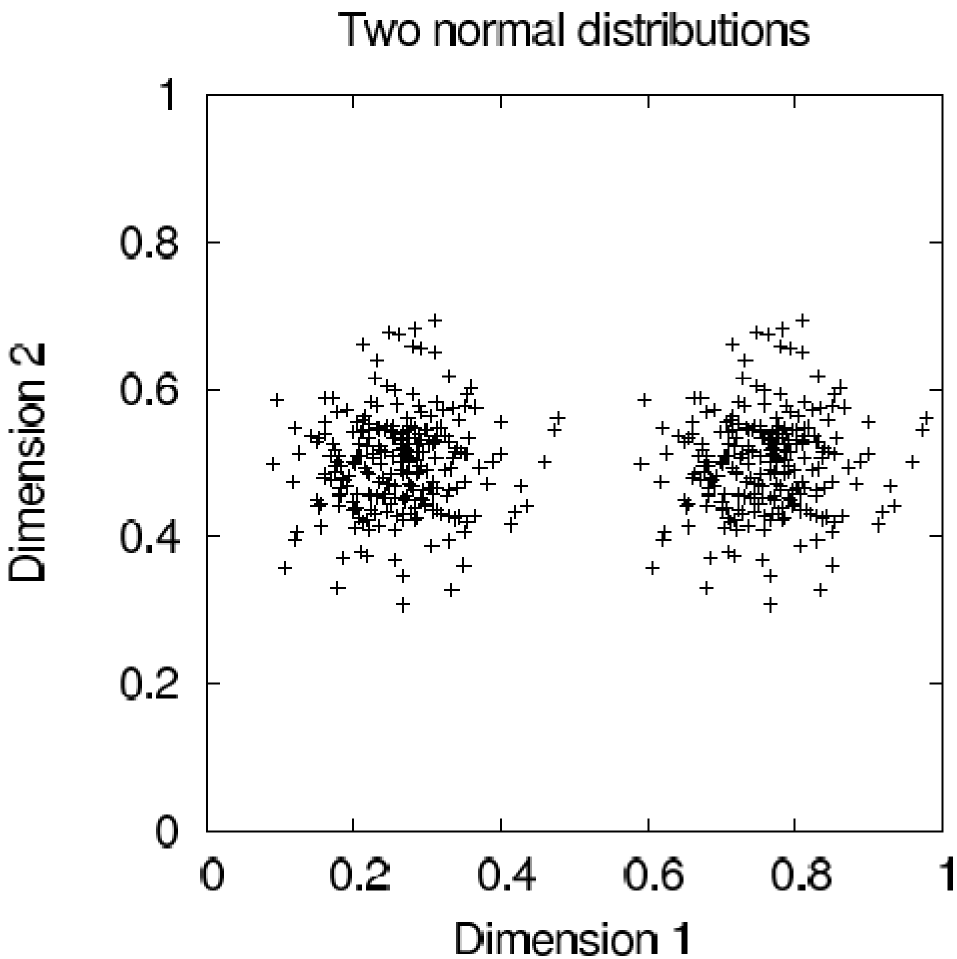
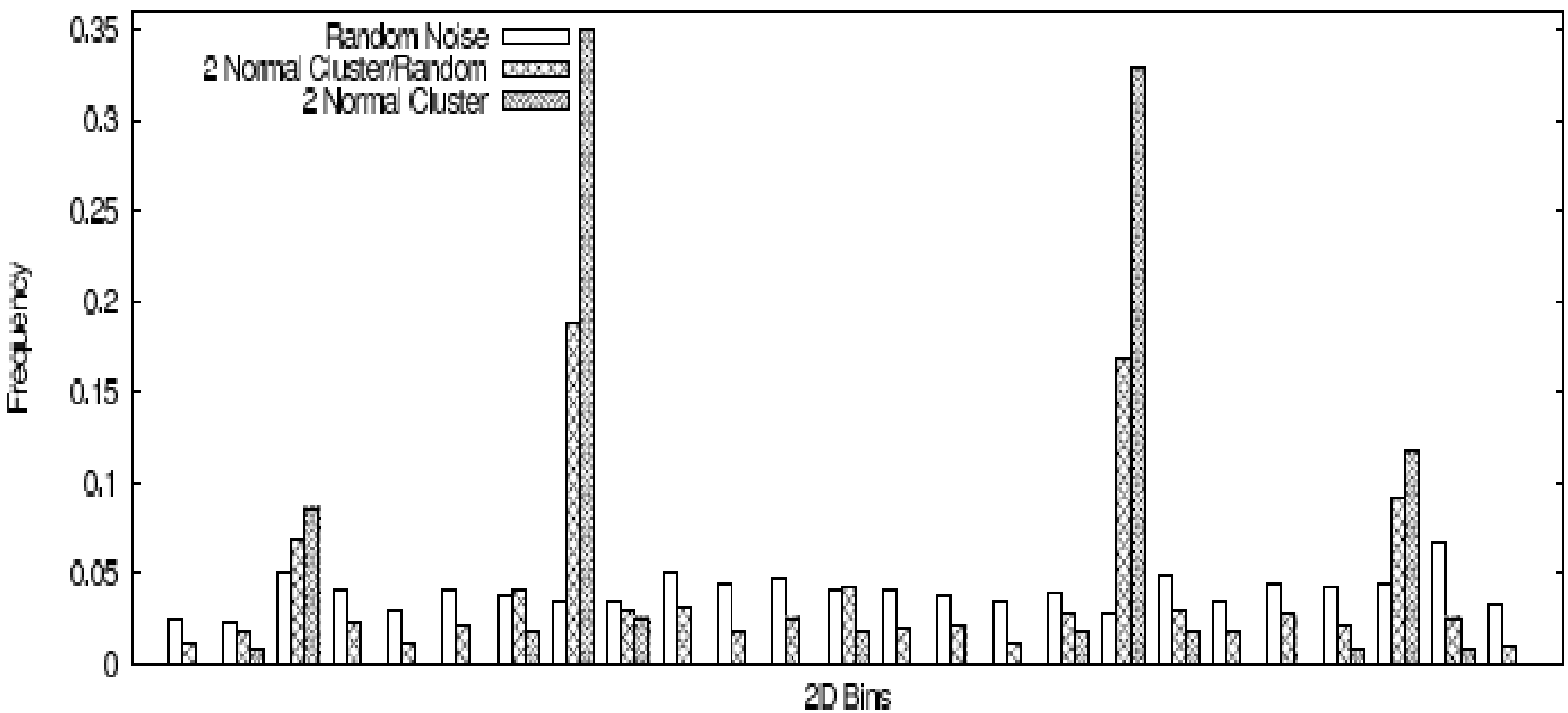


Figure 21.1: Clustering Methodology


Random point distribution
Dimension 2
Dimension 1
0
0.2
0.4
0.6
0.8
1


Two normal distributions and Random Noise
Dimension 2
Dimension 1
0
0.2
0.4
0.6
0.8
1


Two normal distributions
Dimension 2
Dimension 1
0
0.2
0.4
0.6
0.8
1



Random Noise
2 Normal Cluster/Random
2 Normal Cluster
Frequency
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
2D Bins

Cluster Validation

Cluster Validity

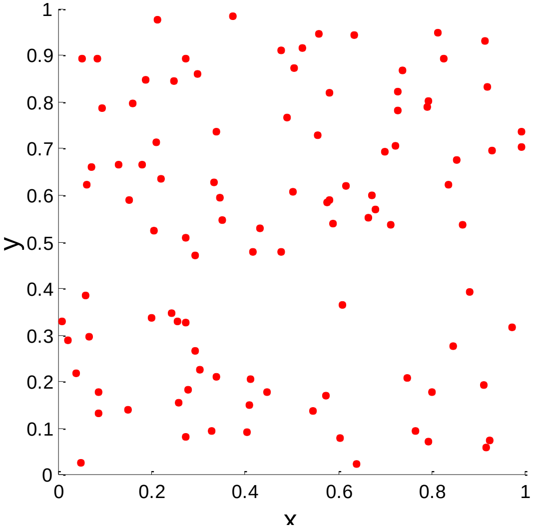
- For cluster analysis, the question is how to evaluate the "goodness" of the resulting clusters?

- But "clusters are in the eye of the beholder"!

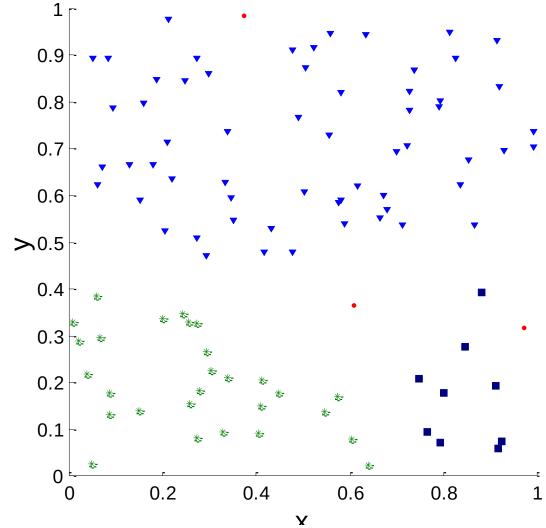
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Clusters found in Random Data

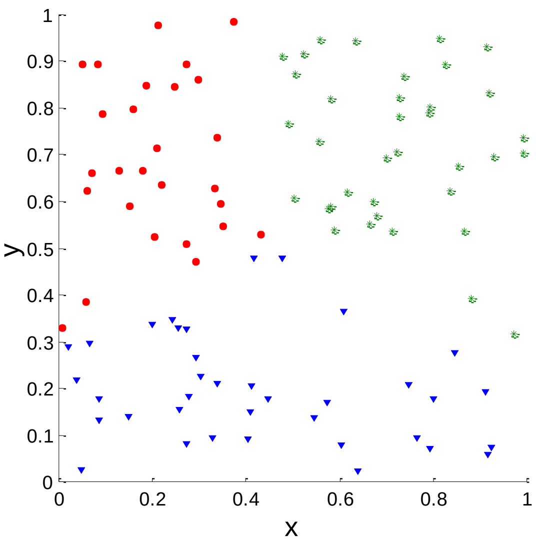
Random Points



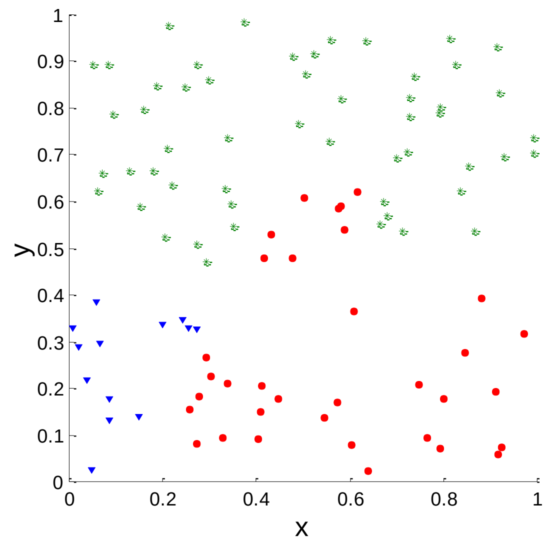
DBSCAN



K-means



Complete Link



Different Aspects 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.

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

Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

External Validation

Algorithm 21.4: Algorithm for matching partitions and clusters

MatchPartitionCluster **(**P**,**C**,**$match$**)**:

foreach $p \in P$ **do**

  $match(p) \leftarrow \emptyset$

  foreach $c \in C$ **do**

    $overlap(p, c) \leftarrow \frac{|p \cap c|}{|p|}$

while $overlap \neq \emptyset$ **do**

  $(p_{max}, c_{max}) \leftarrow GetMaxOverlap(overlap)$

  $match(p_{max}) \leftarrow c_{max}$

  $overlap \leftarrow overlap - \{overlap(p_{max}, *), overlap(*, c_{max})\}$

Purity-Based Measure

- Purity
  - $\frac{|c_i \cap p_j|}{|c_i|}$ $\quad \max_j \rho_{ij} \quad$ $purity_C = \sum_r \frac{|c_i|}{|c|} purity_i,$
- Precision/Recall/F-Measure
  prec(i,j), recall(i,j), $F(i,j) = \frac{2 \times prec(i,j) \times rec(i,j)}{prec(i,j) + rec(i,j)}$
- Entropy

$$e_i = -\sum_q \rho_{ij} \log_2 \rho_{ij}.$$

$$e_C = \sum_r \frac{|c_i|}{|c|} e_i,$$

Matching Measure

1. $x_C = y_C \wedge x_P = y_P$
2. $x_C = y_C \wedge x_P \neq y_P$
3. $x_C \neq y_C \wedge x_P = y_P$
4. $x_C \neq y_C \wedge x_P \neq y_P$

- Rand Statistic:

$$Rand_{P,C} = \frac{CP + \overline{CP}}{m}$$

- Jaccard Coefficient:

$$Jaccard_{P,C} = \frac{CP}{CP + C\overline{P} + \overline{C}P}$$

Correlation Measure

- Hubert's Tau Statistics:

$$\Gamma = \frac{1}{m}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} X_P(i,j)X_C(i,j)$$

- Normalized Tau Statistics:

$$\hat{\Gamma} = \frac{\frac{1}{m}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(X_P(i,j)-\mu_P)(X_C(i,j)\mu_C)}{\sigma_P\sigma_C}$$

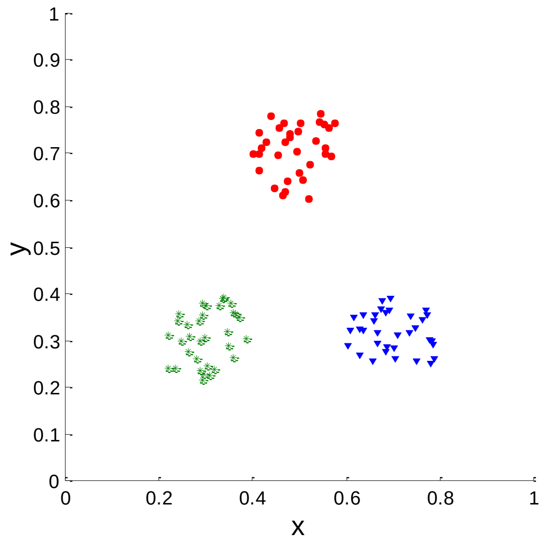
where μ_P and μ_C are the means and σ_P and σ_C are the variances of the matrices X_C and X_P.

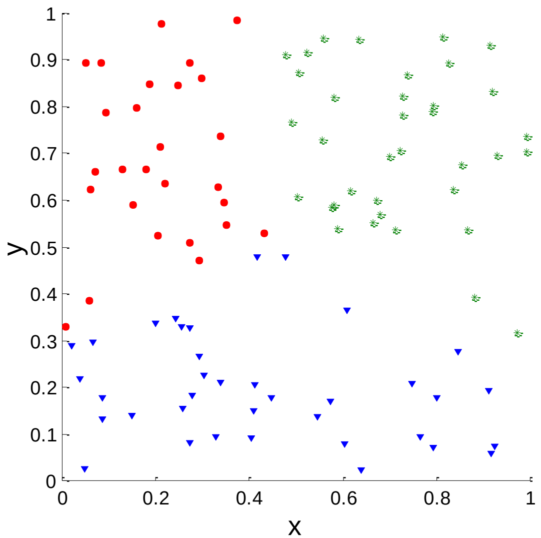
Measuring Cluster Validity Via Correlation

- 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 the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

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

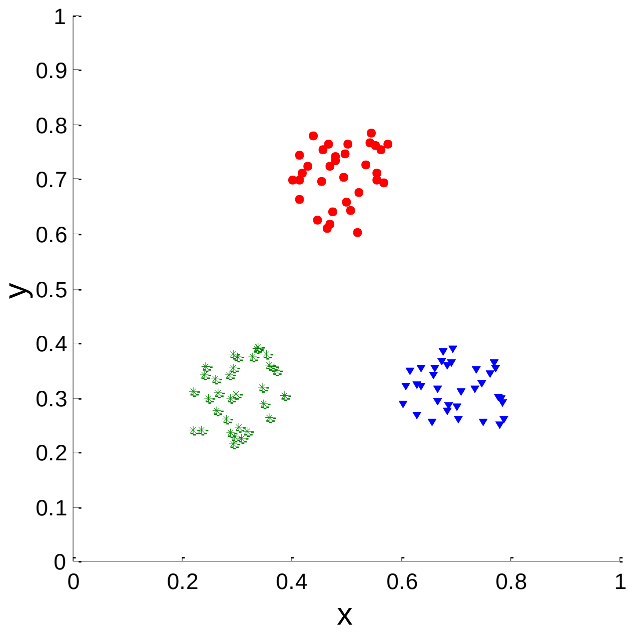


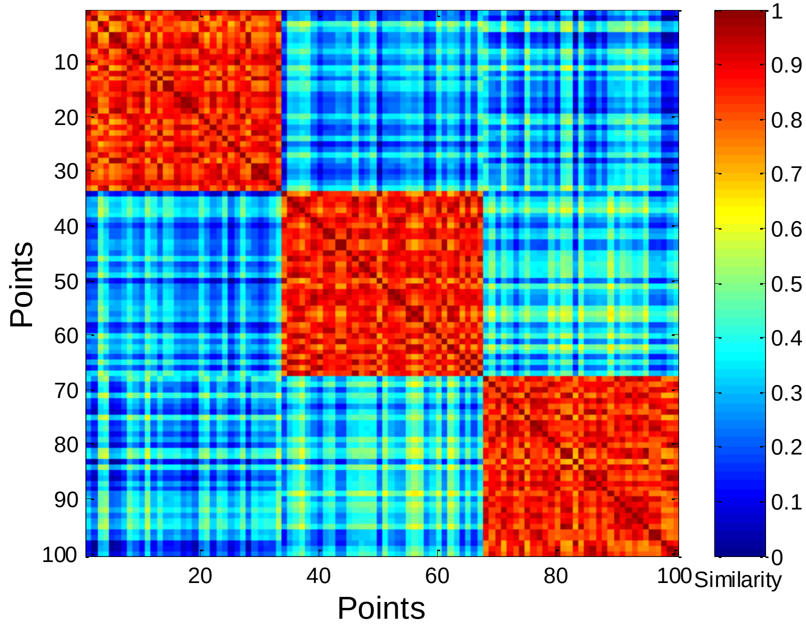


Corr = -0.9235 **Corr = -0.5810**

Using Similarity Matrix for Cluster Validation

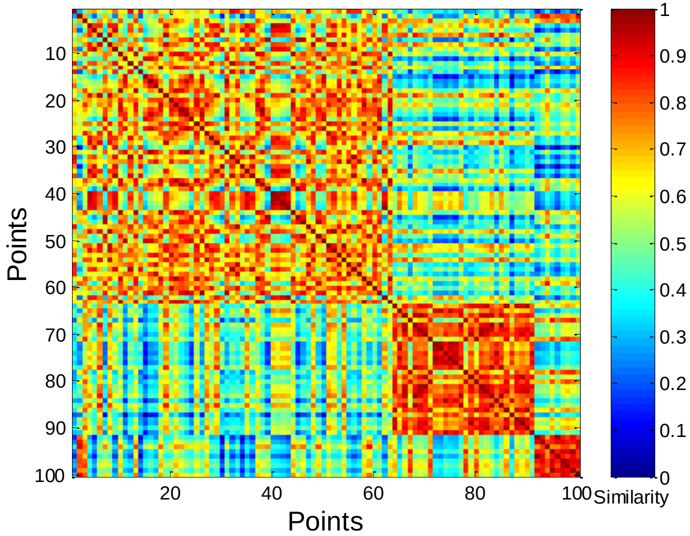
- Order the similarity matrix with respect to cluster labels and inspect visually.

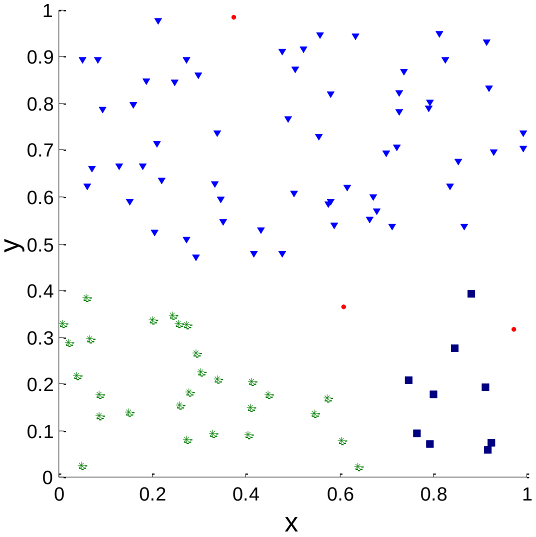




Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

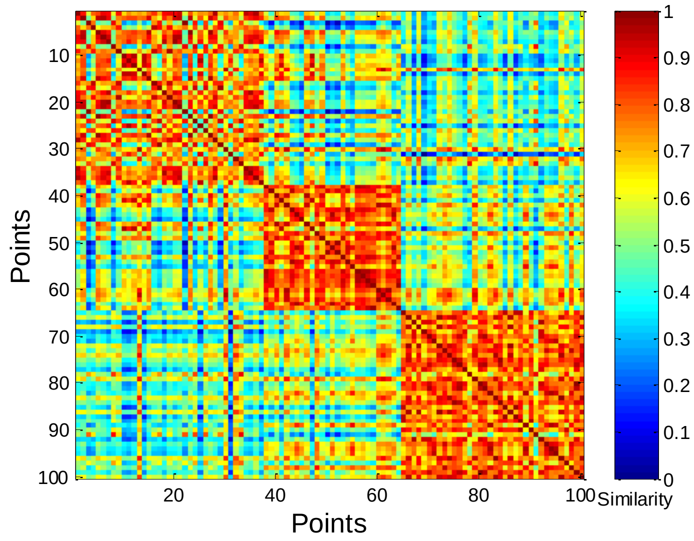


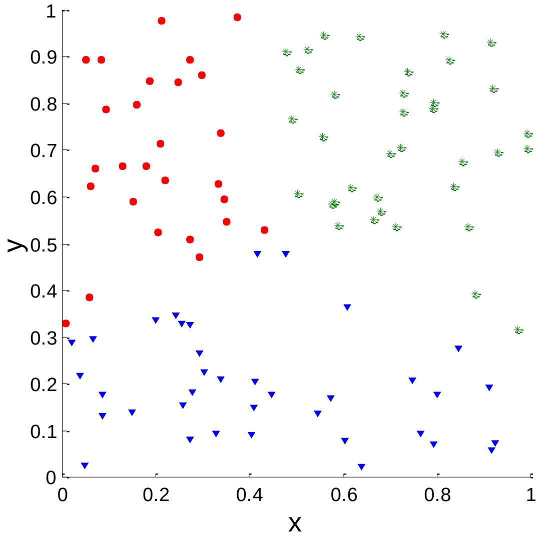


DBSCAN

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

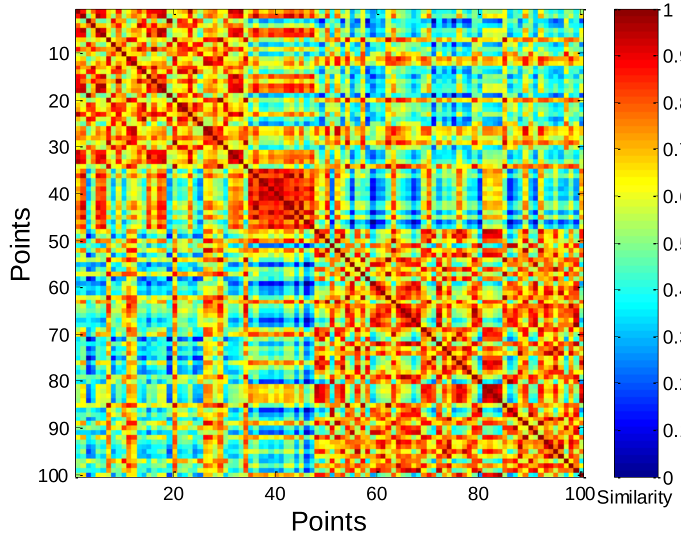


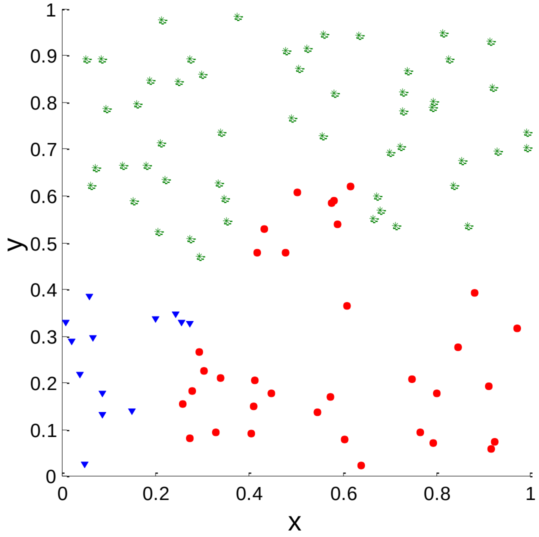


K-means

Using Similarity Matrix for Cluster Validation

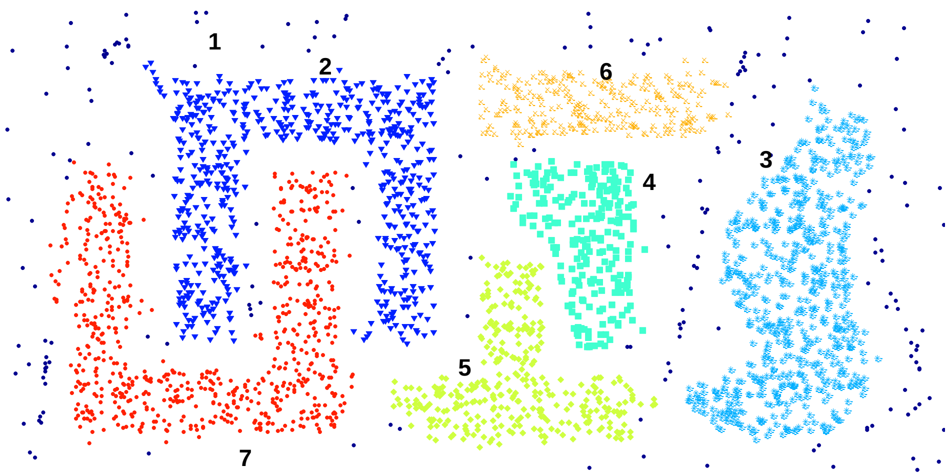
- Clusters in random data are not so crisp

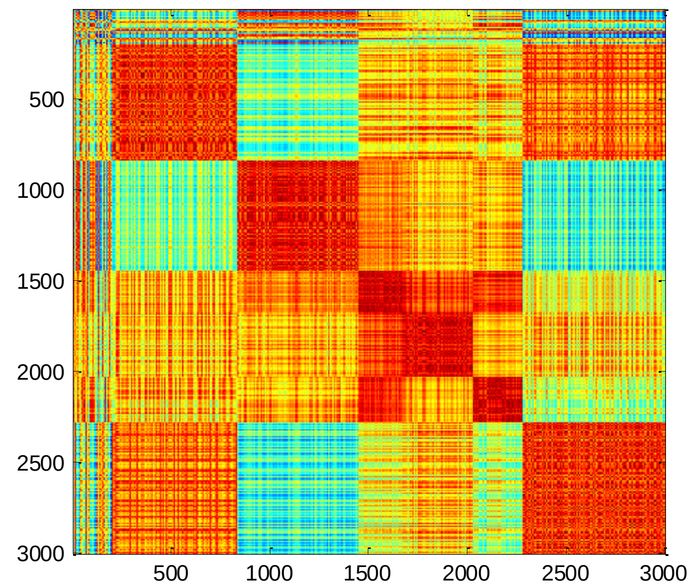




Complete Link

Using Similarity Matrix for Cluster Validation

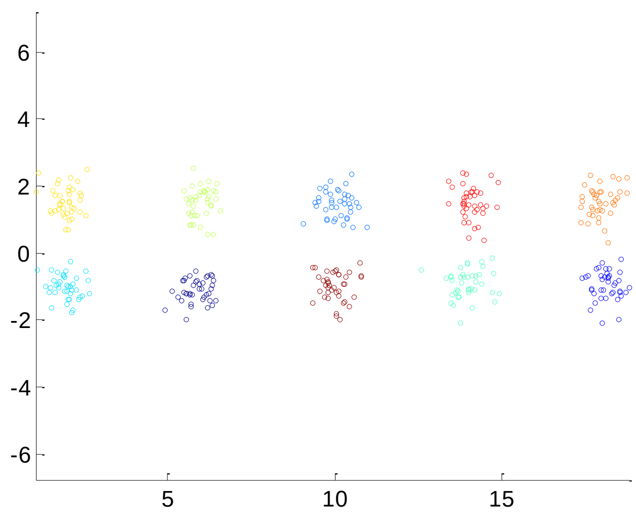


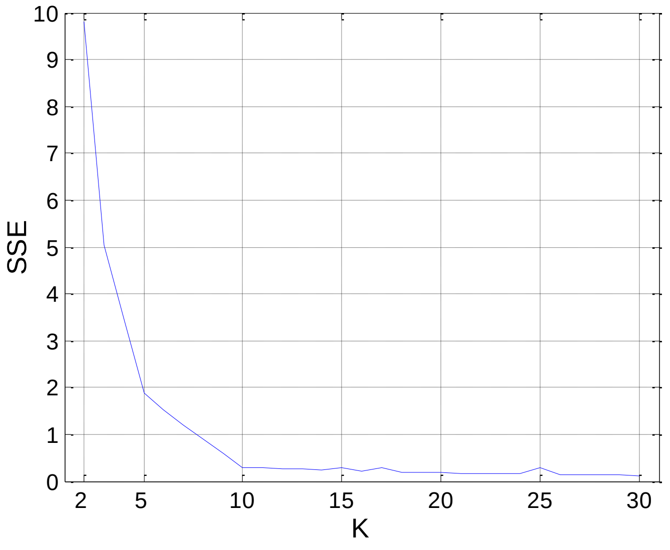


DBSCAN

Internal Measures: SSE

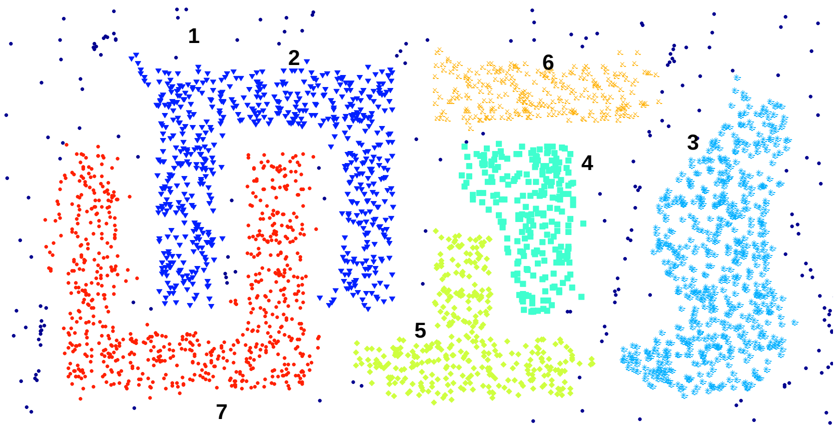
- Clusters in more complicated figures aren't well separated
- Internal Index: Used to measure the goodness of a clustering structure without respect to external information
  - SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters

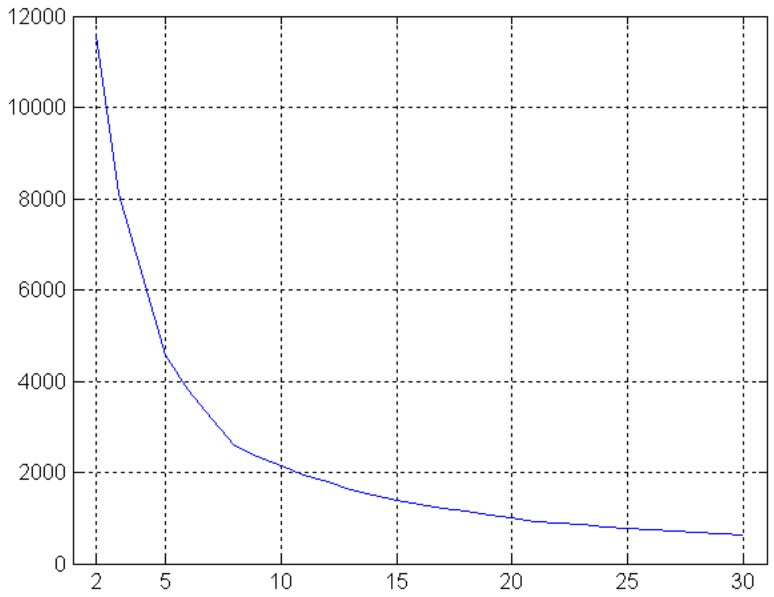




Internal Measures: SSE

- SSE curve for a more complicated data set





SSE of clusters found using K-means

Internal 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)

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

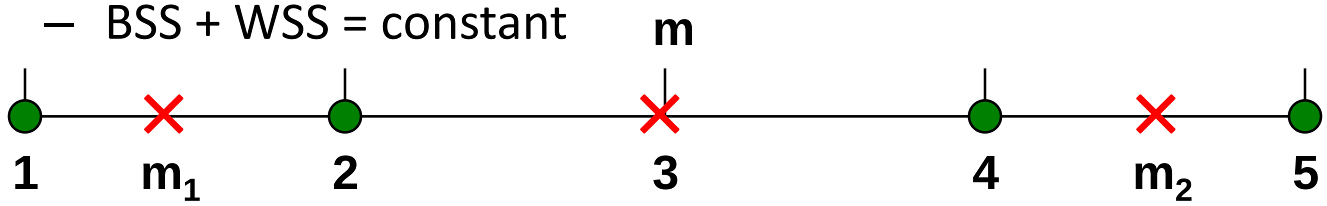
  - Separation is measured by the between cluster sum of squares

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

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

Internal Measures: Cohesion and Separation

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

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

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

$$Total = 10 + 0 = 10$$

K=2 clusters:

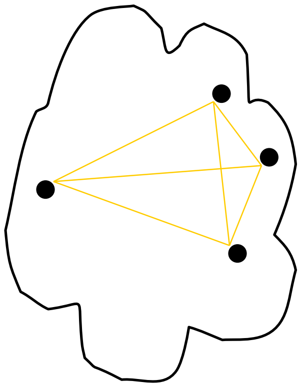
$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

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

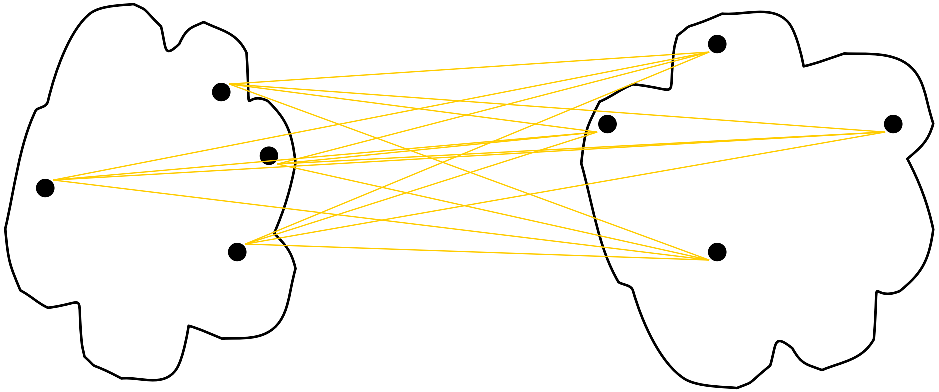
$$Total = 1 + 9 = 10$$

Internal 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

BetaCV

$$BetaCV = \frac{d_{intra}}{d_{inter}}$$

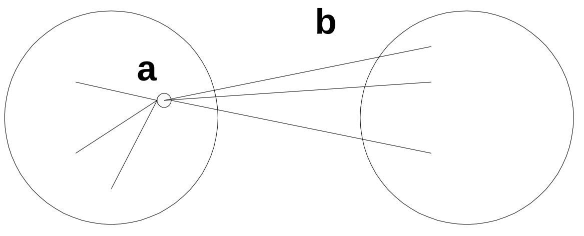
$$d_{intra} = avgd(i,j)|C_i = C_j d_{inter} = avgd(i,j)|C_i \neq C_j$$

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine 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

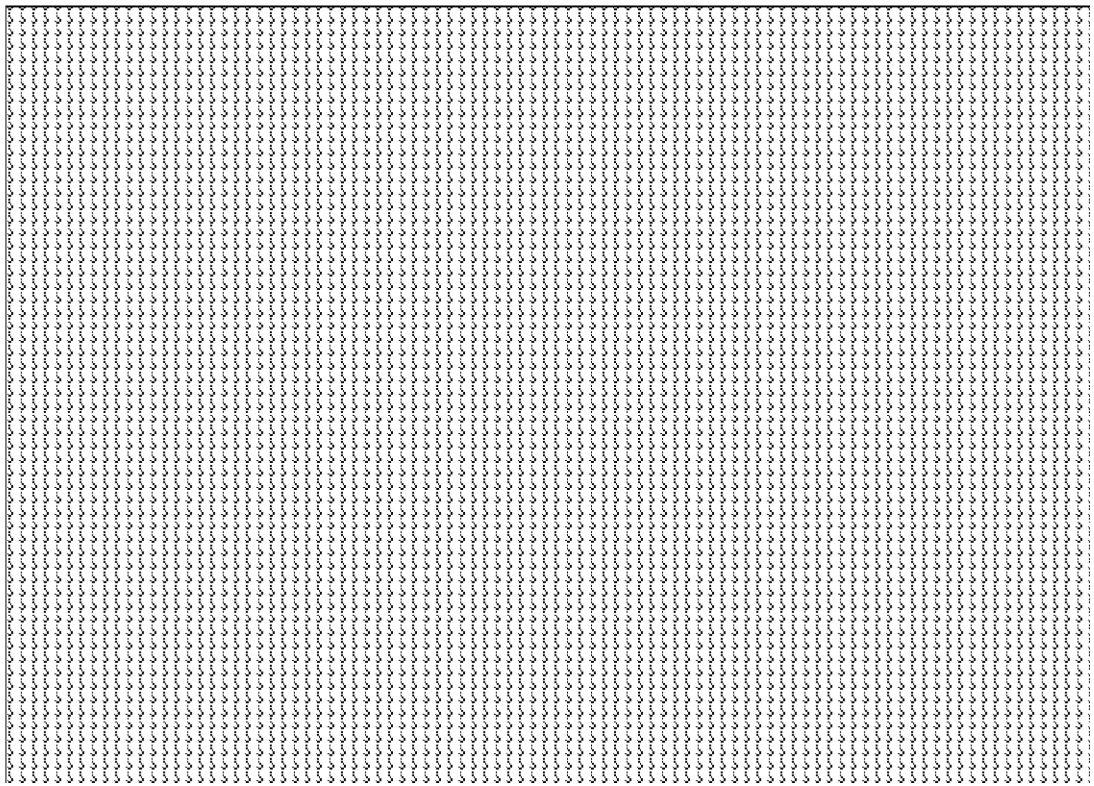
    $s = 1 – a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)

  - Typically between 0 and 1.
  - The closer to 1 the better.



- Can calculate the Average Silhouette width for a cluster or a clustering

External Measures of Cluster Validity: Entropy and Purity



Final Comment on Cluster Validity

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

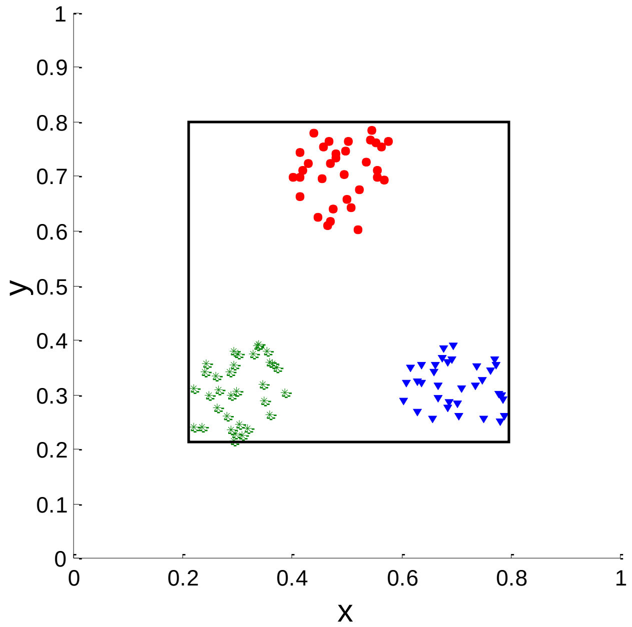
Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

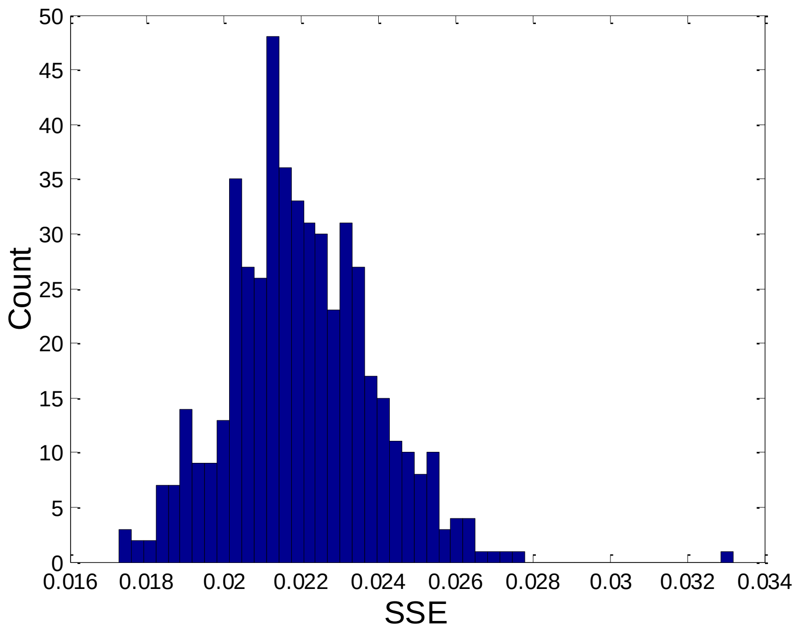
Algorithms for Clustering Data, Jain and Dubes

Extra Slides

Statistical Framework for SSE

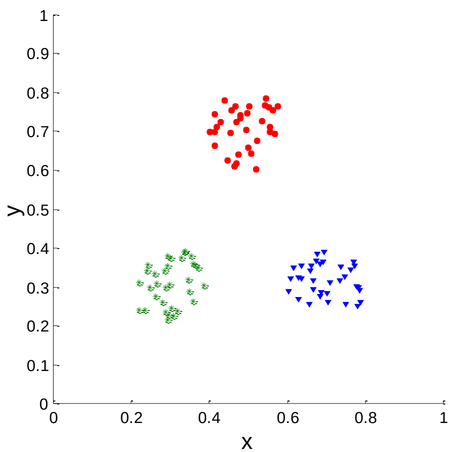
- Example
  - Compare SSE of 0.005 against three clusters in random data
  - Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

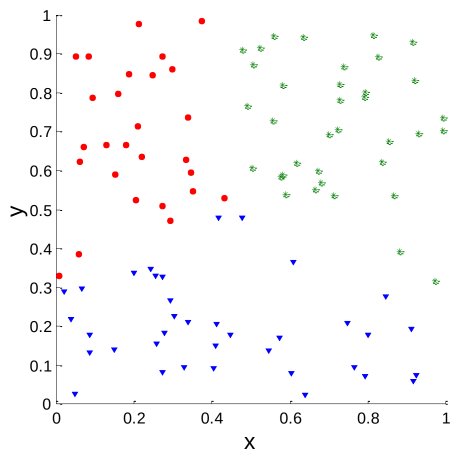


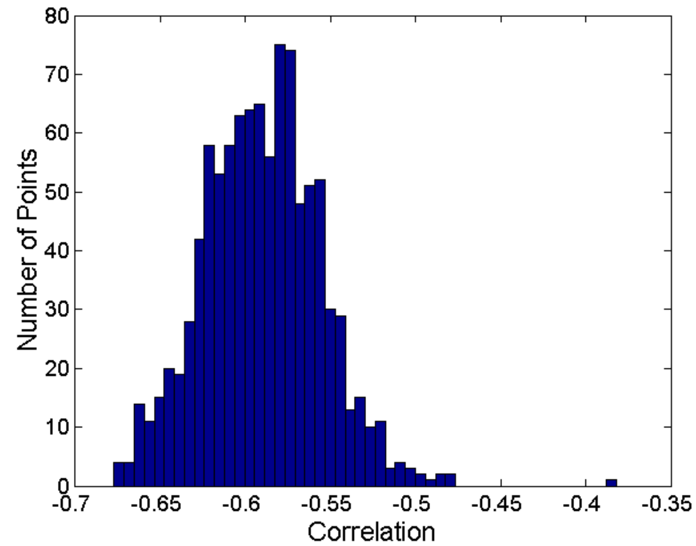


Statistical Framework for Correlation

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







Corr = -0.9235

Corr = -0.5810